The Finite Element Method

Fifth edition

Volume 1: The Basis

Professor O.C. Zienkiewicz, CBE, FRS, FREng is Professor Emeritus and Director of the Institute for Numerical Methods in Engineering at the University of Wales, Swansea, UK. He holds the UNESCO Chair of Numerical Methods in Engineering at the Technical University of Catalunya, Barcelona, Spain. He was the head of the Civil Engineering Department at the University of Wales Swansea between 1961 and 1989. He established that department as one of the primary centres of finite element research. In 1968 he became the Founder Editor of the *International Journal for Numerical Methods in Engineering* which still remains today the major journal in this field. The recipient of 24 honorary degrees and many medals, Professor Zienkiewicz is also a member of five academies – an honour he has received for his many contributions to the fundamental developments of the finite element method. In 1978, he became a Fellow of the Royal Society and the Royal Academy of Engineering. This was followed by his election as a foreign member to the U.S. Academy of Engineering (1981), the Polish Academy of Science (1985), the Chinese Academy of Sciences (1998), and the National Academy of Science, Italy (Academia dei Lincei) (1999). He published the first edition of this book in 1967 and it remained the only book on the subject until 1971.

Professor R.L. Taylor has more than 35 years' experience in the modelling and simulation of structures and solid continua including two years in industry. In 1991 he was elected to membership in the U.S. National Academy of Engineering in recognition of his educational and research contributions to the field of computational mechanics. He was appointed as the T.Y. and Margaret Lin Professor of Engineering in 1992 and, in 1994, received the Berkeley Citation, the highest honour awarded by the University of California, Berkeley. In 1997, Professor Taylor was made a Fellow in the U.S. Association for Computational Mechanics and recently he was elected Fellow in the International Association of Computational Mechanics, and was awarded the USACM John von Neumann Medal. Professor Taylor has written several computer programs for finite element analysis of structural and non-structural systems, one of which, FEAP, is used world-wide in education and research environments. FEAP is now incorporated more fully into the book to address non-linear and finite deformation problems.

Front cover image: A Finite Element Model of the world land speed record (765.035 mph) car THRUST SSC. The analysis was done using the finite element method by K. Morgan, O. Hassan and N.P. Weatherill at the Institute for Numerical Methods in Engineering, University of Wales Swansea, UK. (see K. Morgan, O. Hassan and N.P. Weatherill, 'Why didn't the supersonic car fly?', *Mathematics Today*, *Bulletin of the Institute of Mathematics and Its Applications,* Vol. 35, No. 4, 110–114, Aug. 1999).

The Finite Element Method

Fifth edition

Volume 1: The Basis

O.C. Zienkiewicz, CBE, FRS, FREng
UNESCO Professor of Numerical Methods in Engineering
International Centre for Numerical Methods in Engineering, Barcelona
Emeritus Professor of Civil Engineering and Director of the Institute for Numerical Methods in Engineering, University of Wales, Swansea

R.L. Taylor
Professor in the Graduate School
Department of Civil and Environmental Engineering
University of California at Berkeley
Berkeley, California

BUTTERWORTH HEINEMANN

OXFORD AUCKLAND BOSTON JOHANNESBURG MELBOURNE NEW DELHI

Butterworth-Heinemann
Linacre House, Jordan Hill, Oxford OX2 8DP
225 Wildwood Avenue, Woburn, MA 01801-2041
A division of Reed Educational and Professional Publishing Ltd

A member of the Reed Elsevier plc group

First published in 1967 by McGraw-Hill
Fifth edition published by Butterworth-Heinemann 2000

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication Data
A catalogue record for this book is available from the Library of Congress

ISBN 0 7506 5049 4

**Published with the cooperation of CIMNE,
the International Centre for Numerical Methods in Engineering,
Barcelona, Spain (www.cimne.upc.es)**

Typeset by Academic & Technical Typesetting, Bristol
Printed and bound by MPG Books Ltd

Dedication

This book is dedicated to our wives Helen and Mary Lou and our families for their support and patience during the preparation of this book, and also to all of our students and colleagues who over the years have contributed to our knowledge of the finite element method. In particular we would like to mention Professor Eugenio Oñate and his group at CIMNE for their help, encouragement and support during the preparation process.

Contents

Volume 2: Solid and structural mechanics

1. General problems in solid mechanics and non-linearity
2. Solution of non-linear algebraic equations
3. Inelastic materials
4. Plate bending approximation: thin (Kirchhoff) plates and C_1 continuity requirements
5. 'Thick' Reissner–Mindlin plates – irreducible and mixed formulations
6. Shells as an assembly of flat elements
7. Axisymmetric shells
8. Shells as a special case of three-dimensional analysis – Reissner–Mindlin assumptions
9. Semi-analytical finite element processes – use of orthogonal functions and 'finite strip' methods
10. Geometrically non-linear problems – finite deformation
11. Non-linear structural problems – large displacement and instability
12. Pseudo-rigid and rigid–flexible bodies
13. Computer procedures for finite element analysis

Appendix A: Invariants of second-order tensors

Volume 3: Fluid dynamics

1. Introduction and the equations of fluid dynamics
2. Convection dominated problems – finite element approximations
3. A general algorithm for compressible and incompressible flows – the characteristic based split (CBS) algorithm
4. Incompressible laminar flow – newtonian and non-newtonian fluids
5. Free surfaces, buoyancy and turbulent incompressible flows
6. Compressible high speed gas flow
7. Shallow-water problems
8. Waves
9. Computer implementation of the CBS algorithm

Appendix A. Non-conservative form of Navier–Stokes equations
Appendix B. Discontinuous Galerkin methods in the solution of the convection–diffusion equation
Appendix C. Edge-based finite element formulation
Appendix D. Multi grid methods
Appendix E. Boundary layer – inviscid flow coupling

Preface

It is just over thirty years since *The Finite Element Method in Structural and Continuum Mechanics* was first published. This book, which was the first dealing with the finite element method, provided the base from which many further developments occurred. The expanding research and field of application of finite elements led to the second edition in 1971, the third in 1977 and the fourth in 1989 and 1991. The size of each of these volumes expanded geometrically (from 272 pages in 1967 to the fourth edition of 1455 pages in two volumes). This was necessary to do justice to a rapidly expanding field of professional application and research. Even so, much filtering of the contents was necessary to keep these editions within reasonable bounds.

It seems that a new edition is necessary every decade as the subject is expanding and many important developments are continuously occurring. The present fifth edition is indeed motivated by several important developments which have occurred in the 90s. These include such subjects as adaptive error control, meshless and point based methods, new approaches to fluid dynamics, etc. However, we feel it is important not to increase further the overall size of the book and we therefore have eliminated some redundant material.

Further, the reader will notice the present subdivision into three volumes, in which the first volume provides the general basis applicable to linear problems in many fields whilst the second and third volumes are devoted to more advanced topics in solid and fluid mechanics, respectively. This arrangement will allow a general student to study Volume 1 whilst a specialist can approach their topics with the help of Volumes 2 and 3. Volumes 2 and 3 are much smaller in size and addressed to more specialized readers.

It is hoped that Volume 1 will help to introduce postgraduate students, researchers and practitioners to the modern concepts of finite element methods. In Volume 1 we stress the relationship between the finite element method and the more classic finite difference and boundary solution methods. We show that all methods of numerical approximation can be cast in the same format and that their individual advantages can thus be retained.

Although Volume 1 is not written as a course text book, it is nevertheless directed at students of postgraduate level and we hope these will find it to be of wide use. Mathematical concepts are stressed throughout and precision is maintained, although little use is made of modern mathematical symbols to ensure wider understanding amongst engineers and physical scientists.

In Volumes 1, 2 and 3 the chapters on computational methods are much reduced by transferring the computer source programs to a web site.[1] This has the very substantial advantage of not only eliminating errors in copying the programs but also in ensuring that the reader has the benefit of the most recent set of programs available to him or her at all times as it is our intention from time to time to update and expand the available programs.

The authors are particularly indebted to the International Center of Numerical Methods in Engineering (CIMNE) in Barcelona who have allowed their pre- and post-processing code (GiD) to be accessed from the publisher's web site. This allows such difficult tasks as mesh generation and graphic output to be dealt with efficiently. The authors are also grateful to Dr J.Z. Zhu for his careful scrutiny and help in drafting Chapters 14 and 15. These deal with error estimation and adaptivity, a subject to which Dr Zhu has extensively contributed. Finally, we thank Peter and Jackie Bettess for writing the general subject index.

OCZ and RLT

[1] Complete source code for all programs in the three volumes may be obtained at no cost from the publisher's web page: http://www.bh.com/companions/fem

1

Some preliminaries: the standard discrete system

1.1 Introduction

The limitations of the human mind are such that it cannot grasp the behaviour of its complex surroundings and creations in one operation. Thus the process of subdividing all systems into their individual components or 'elements', whose behaviour is readily understood, and then rebuilding the original system from such components to study its behaviour is a natural way in which the engineer, the scientist, or even the economist proceeds.

In many situations an adequate model is obtained using a finite number of well-defined components. We shall term such problems *discrete*. In others the subdivision is continued indefinitely and the problem can only be defined using the mathematical fiction of an infinitesimal. This leads to differential equations or equivalent statements which imply an infinite number of elements. We shall term such systems *continuous*.

With the advent of digital computers, *discrete* problems can generally be solved readily even if the number of elements is very large. As the capacity of all computers is finite, *continuous* problems can only be solved exactly by mathematical manipulation. Here, the available mathematical techniques usually limit the possibilities to oversimplified situations.

To overcome the intractability of realistic types of continuum problems, various methods of *discretization* have from time to time been proposed both by engineers and mathematicians. All involve an *approximation* which, hopefully, approaches in the limit the true continuum solution as the number of discrete variables increases.

The discretization of continuous problems has been approached differently by mathematicians and engineers. Mathematicians have developed general techniques applicable directly to differential equations governing the problem, such as finite difference approximations,[1,2] various weighted residual procedures,[3,4] or approximate techniques for determining the stationarity of properly defined 'functionals'. The engineer, on the other hand, often approaches the problem more intuitively by creating an analogy between real discrete elements and finite portions of a continuum domain. For instance, in the field of solid mechanics McHenry,[5] Hrenikoff,[6] Newmark[7], and indeed Southwell[9] in the 1940s, showed that reasonably good solutions to an elastic continuum problem can be obtained by replacing small portions

of the continuum by an arrangement of simple elastic bars. Later, in the same context, Argyris[8] and Turner *et al.*[9] showed that a more direct, but no less intuitive, substitution of properties can be made much more effectively by considering that small portions or 'elements' in a continuum behave in a simplified manner.

It is from the engineering 'direct analogy' view that the term 'finite element' was born. Clough[10] appears to be the first to use this term, which implies in it a direct use of a *standard methodology applicable to discrete systems*. Both conceptually and from the computational viewpoint, this is of the utmost importance. The first allows an improved understanding to be obtained; the second offers a unified approach to the variety of problems and the development of standard computational procedures.

Since the early 1960s much progress has been made, and today the purely mathematical and 'analogy' approaches are fully reconciled. It is the object of this text to present a view of the finite element method as *a general discretization procedure of continuum problems posed by mathematically defined statements*.

In the analysis of problems of a discrete nature, a standard methodology has been developed over the years. The civil engineer, dealing with structures, first calculates force–displacement relationships for each element of the structure and then proceeds to assemble the whole by following a well-defined procedure of establishing local equilibrium at each 'node' or connecting point of the structure. The resulting equations can be solved for the unknown displacements. Similarly, the electrical or hydraulic engineer, dealing with a network of electrical components (resistors, capacitances, etc.) or hydraulic conduits, first establishes a relationship between currents (flows) and potentials for individual elements and then proceeds to assemble the system by ensuring continuity of flows.

All such analyses follow a standard pattern which is universally adaptable to discrete systems. It is thus possible to define a *standard discrete system*, and this chapter will be primarily concerned with establishing the processes applicable to such systems. Much of what is presented here will be known to engineers, but some reiteration at this stage is advisable. As the treatment of elastic solid structures has been the most developed area of activity this will be introduced first, followed by examples from other fields, before attempting a complete generalization.

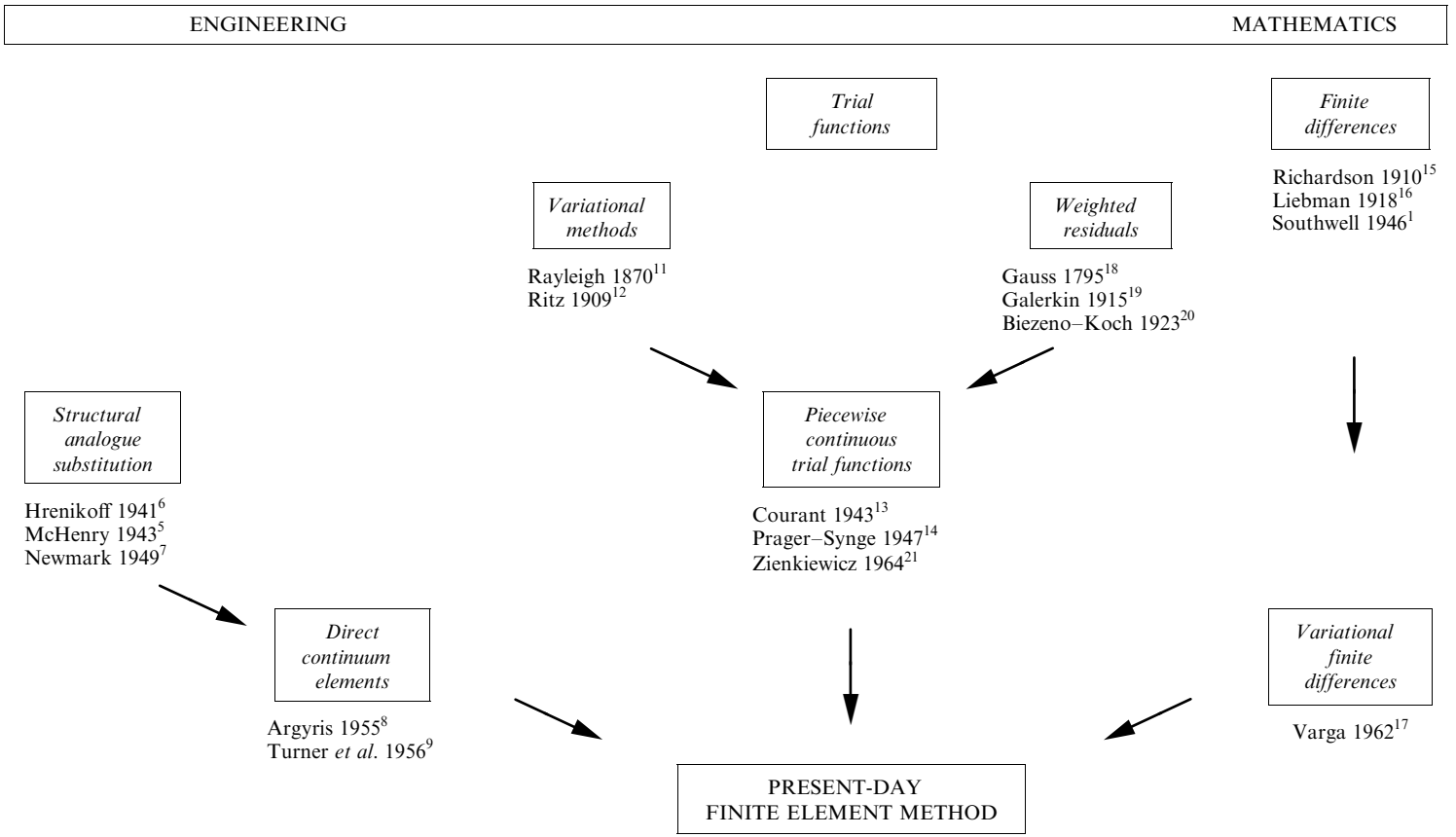
The existence of a unified treatment of 'standard discrete problems' leads us to the first definition of the finite element process as a method of approximation to continuum problems such that

(a) the continuum is divided into a finite number of parts (elements), the behaviour of which is specified by a finite number of parameters, and
(b) the solution of the complete system as an assembly of its elements follows precisely the same rules as those applicable to *standard discrete problems*.

It will be found that most classical mathematical approximation procedures as well as the various direct approximations used in engineering fall into this category. It is thus difficult to determine the origins of the finite element method and the precise moment of its invention.

Table 1.1 shows the process of evolution which led to the present-day concepts of finite element analysis. Chapter 3 will give, in more detail, the mathematical basis which emerged from these classical ideas.[11–20]

Table 1.1


ENGINEERING
MATHEMATICS
Trial functions
Finite differences
Variational methods
Weighted residuals
Richardson 1910[15]
Liebman 1918[16]
Southwell 1946[1]
Rayleigh 1870[11]
Ritz 1909[12]
Gauss 1795[18]
Galerkin 1915[19]
Biezeno–Koch 1923[20]
Structural analogue substitution
Piecewise continuous trial functions
Hrenikoff 1941[6]
McHenry 1943[5]
Newmark 1949[7]
Courant 1943[13]
Prager–Synge 1947[14]
Zienkiewicz 1964[21]
Direct continuum elements
Variational finite differences
Argyris 1955[8]
Turner et al. 1956[9]
Varga 1962[17]
PRESENT-DAY
FINITE ELEMENT METHOD

1.2 The structural element and the structural system

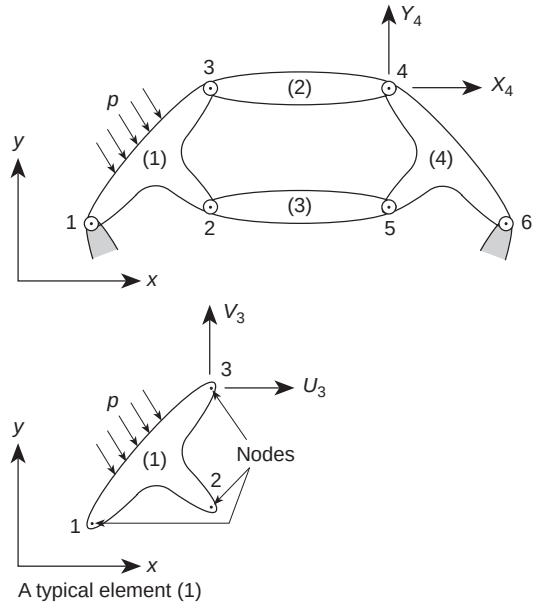


Fig. 1.1 A typical structure built up from interconnected elements.

To introduce the reader to the general concept of discrete systems we shall first consider a structural engineering example of linear elasticity.

Figure 1.1 represents a two-dimensional structure assembled from individual components and interconnected at the nodes numbered 1 to 6. The joints at the nodes, in this case, are pinned so that moments cannot be transmitted.

As a starting point it will be assumed that by separate calculation, or for that matter from the results of an experiment, the characteristics of each element are precisely known. Thus, if a typical element labelled (1) and associated with nodes 1, 2, 3 is examined, the forces acting at the nodes are uniquely defined by the displacements of these nodes, the distributed loading acting on the element (p), and its initial strain. The last may be due to temperature, shrinkage, or simply an initial 'lack of fit'. The forces and the corresponding displacements are defined by appropriate components (U, V and u, v) in a common coordinate system.

Listing the forces acting on all the nodes (three in the case illustrated) of the element (1) as a matrix† we have

$$\mathbf{q}^1 = \begin{Bmatrix} \mathbf{q}_1^1 \\ \mathbf{q}_2^1 \\ \mathbf{q}_3^1 \end{Bmatrix} \qquad \mathbf{q}_1^1 = \begin{Bmatrix} U_1 \\ V_1 \end{Bmatrix}, \qquad \text{etc.} \tag{1.1}$$

† A limited knowledge of matrix algebra will be assumed throughout this book. This is necessary for reasonable conciseness and forms a convenient book-keeping form. For readers not familiar with the subject a brief appendix (Appendix A) is included in which sufficient principles of matrix algebra are given to follow the development intelligently. Matrices (and vectors) will be distinguished by bold print throughout.

and for the corresponding nodal displacements

$$\mathbf{a}^1 = \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{Bmatrix} \qquad \mathbf{a}_1 = \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix}, \qquad \text{etc.} \tag{1.2}$$

Assuming linear elastic behaviour of the element, the characteristic relationship will always be of the form

$$\mathbf{q}^1 = \mathbf{K}^1\mathbf{a}^1 + \mathbf{f}_p^1 + \mathbf{f}_{\varepsilon_0}^1 \tag{1.3}$$

in which $\mathbf{f}_p^1$ represents the nodal forces required to balance any distributed loads acting on the element and $\mathbf{f}_{\varepsilon_0}^1$ the nodal forces required to balance any initial strains such as may be caused by temperature change if the nodes are not subject to any displacement. The first of the terms represents the forces induced by displacement of the nodes.

Similarly, a preliminary analysis or experiment will permit a unique definition of stresses or internal reactions at any specified point or points of the element in terms of the nodal displacements. Defining such stresses by a matrix $\boldsymbol{\sigma}^1$ a relationship of the form

$$\boldsymbol{\sigma}^1 = \mathbf{Q}^1\mathbf{a}^1 + \boldsymbol{\sigma}_{\varepsilon_0}^1 \tag{1.4}$$

is obtained in which the two term gives the stresses due to the initial strains when no nodal displacement occurs.

The matrix $\mathbf{K}^e$ is known as the element stiffness matrix and the matrix $\mathbf{Q}^e$ as the element stress matrix for an element (e).

Relationships in Eqs (1.3) and (1.4) have been illustrated by an example of an element with three nodes and with the interconnection points capable of transmitting only two components of force. Clearly, the same arguments and definitions will apply generally. An element (2) of the hypothetical structure will possess only two points of interconnection; others may have quite a large number of such points. Similarly, if the joints were considered as rigid, three components of generalized force and of generalized displacement would have to be considered, the last of these corresponding to a moment and a rotation respectively. For a rigidly jointed, three-dimensional structure the number of individual nodal components would be six. Quite generally, therefore,

$$\mathbf{q}^e = \begin{Bmatrix} \mathbf{q}_1^e \\ \mathbf{q}_2^e \\ \vdots \\ \mathbf{q}_m^e \end{Bmatrix} \qquad \text{and} \qquad \mathbf{a}^e = \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{Bmatrix} \tag{1.5}$$

with each $\mathbf{q}_i^e$ and $\mathbf{a}_i$ possessing the same number of components or *degrees of freedom*. These quantities are conjugate to each other.

The stiffness matrices of the element will clearly always be square and of the form

$$\mathbf{K}^e = \begin{bmatrix} \mathbf{K}_{ii}^e & \mathbf{K}_{ij}^e & \cdots & \mathbf{K}_{im}^e \\ \vdots & \vdots & & \vdots \\ \mathbf{K}_{mi}^e & \cdots & \cdots & \mathbf{K}_{mm}^e \end{bmatrix} \tag{1.6}$$

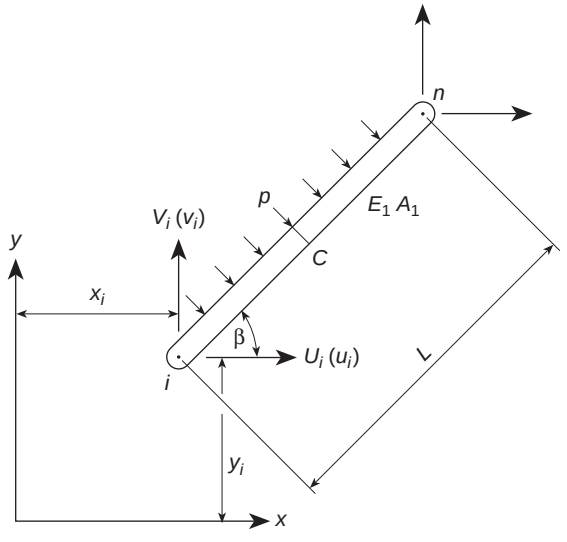


Fig. 1.2 A pin-ended bar.

in which $\mathbf{K}_{ii}^e$, etc., are submatrices which are again square and of the size $l \times l$, where l is the number of force components to be considered at each node.

As an example, the reader can consider a pin-ended bar of uniform section A and modulus E in a two-dimensional problem shown in Fig. 1.2. The bar is subject to a uniform lateral load p and a uniform thermal expansion strain

$$\varepsilon_0 = \alpha T$$

where α is the coefficient of linear expansion and T is the temperature change.

If the ends of the bar are defined by the coordinates x_i, y_i and x_n, y_n its length can be calculated as

$$L = \sqrt{[(x_n - x_i)^2 + (y_n - y_i)^2]}$$

and its inclination from the horizontal as

$$\beta = \tan^{-1} \frac{y_n - y_i}{x_n - x_i}$$

Only two components of force and displacement have to be considered at the nodes.

The nodal forces due to the lateral load are clearly

$$\mathbf{f}_p^e = \begin{Bmatrix} U_i \\ V_i \\ U_n \\ V_n \end{Bmatrix}_p = -\begin{Bmatrix} -\sin\beta \\ \cos\beta \\ -\sin\beta \\ \cos\beta \end{Bmatrix} \frac{pL}{2}$$

and represent the appropriate components of simple reactions, $pL/2$. Similarly, to restrain the thermal expansion ε_0 an axial force $(E\alpha TA)$ is needed, which gives the

components

$$\mathbf{f}^e_{\varepsilon_0} = \begin{Bmatrix} U_i \\ V_i \\ U_n \\ V_n \end{Bmatrix}_{\varepsilon_0} = -\begin{Bmatrix} -\cos\beta \\ -\sin\beta \\ \cos\beta \\ \sin\beta \end{Bmatrix} (E\alpha TA)$$

Finally, the element displacements

$$\mathbf{a}^e = \begin{Bmatrix} u_i \\ v_i \\ u_n \\ v_n \end{Bmatrix}$$

will cause an elongation $(u_n - u_i)\cos\beta + (v_n - v_i)\sin\beta$. This, when multiplied by EA/L, gives the axial force whose components can again be found. Rearranging these in the standard form gives

$$\mathbf{K}^e\mathbf{a}^e = \begin{Bmatrix} U_i \\ V_i \\ U_n \\ V_n \end{Bmatrix}_\delta = \frac{EA}{L}\left[\begin{array}{cc:cc} \cos^2\beta & \sin\beta\cos\beta & -\cos^2\beta & -\sin\beta\cos\beta \\ \sin\beta\cos\beta & \sin^2\beta & -\sin\beta\cos\beta & -\sin^2\beta \\ \hdashline -\cos^2\beta & -\sin\beta\cos\beta & \cos^2\beta & \sin\beta\cos\beta \\ -\sin\beta\cos\beta & -\sin^2\beta & \sin\beta\cos\beta & \sin^2\beta \end{array}\right] \begin{Bmatrix} u_i \\ v_i \\ u_n \\ v_n \end{Bmatrix}$$

The components of the general equation (1.3) have thus been established for the elementary case discussed. It is again quite simple to find the stresses at any section of the element in the form of relation (1.4). For instance, if attention is focused on the mid-section C of the bar the average stress determined from the axial tension to the element can be shown to be

$$\boldsymbol{\sigma}^e \approx \sigma = \frac{E}{L}[-\cos\beta, -\sin\beta, \cos\beta, \sin\beta]\mathbf{a}^e - E\alpha T$$

where all the bending effects of the lateral load p have been ignored.

For more complex elements more sophisticated procedures of analysis are required but the results are of the same form. The engineer will readily recognize that the so-called 'slope–deflection' relations used in analysis of rigid frames are only a special case of the general relations.

It may perhaps be remarked, in passing, that the complete stiffness matrix obtained for the simple element in tension turns out to be symmetric (as indeed was the case with some submatrices). This is by no means fortuitous but follows from the principle of energy conservation and from its corollary, the well-known Maxwell–Betti reciprocal theorem.

The element properties were assumed to follow a simple linear relationship. In principle, similar relationships could be established for non-linear materials, but discussion of such problems will be held over at this stage.

The calculation of the stiffness coefficients of the bar which we have given here will be found in many textbooks. Perhaps it is worthwhile mentioning here that the first use of bar assemblies for large structures was made as early as 1935 when Southwell proposed his classical relaxation method.[22]

1.3 Assembly and analysis of a structure

Consider again the hypothetical structure of Fig. 1.1. To obtain a complete solution the two conditions of

(a) displacement compatibility and
(b) equilibrium

have to be satisfied throughout.

Any system of nodal displacements $\mathbf{a}$:

$$\mathbf{a} = \begin{Bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_n \end{Bmatrix} \tag{1.7}$$

listed now for the whole structure in which all the elements participate, automatically satisfies the first condition.

As the conditions of overall equilibrium have already been satisfied *within* an element, all that is necessary is to establish equilibrium conditions at the nodes of the structure. The resulting equations will contain the displacements as unknowns, and once these have been solved the structural problem is determined. The internal forces in elements, or the stresses, can easily be found by using the characteristics established *a priori* for each element by Eq. (1.4).

Consider the structure to be loaded by external forces $\mathbf{r}$:

$$\mathbf{r} = \begin{Bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{Bmatrix} \tag{1.8}$$

applied at the nodes in addition to the distributed loads applied to the individual elements. Again, any one of the forces $\mathbf{r}_i$ must have the same number of components as that of the element reactions considered. In the example in question

$$\mathbf{r}_i = \begin{Bmatrix} X_i \\ Y_i \end{Bmatrix} \tag{1.9}$$

as the joints were assumed pinned, but at this stage the general case of an arbitrary number of components will be assumed.

If now the equilibrium conditions of a typical node, i, are to be established, each component of $\mathbf{r}_i$ has, in turn, to be equated to the sum of the component forces contributed by the elements meeting at the node. Thus, considering *all* the force

components we have

$$\mathbf{r}_i = \sum_{e=1}^{m} \mathbf{q}_i^e = \mathbf{q}_i^1 + \mathbf{q}_i^2 + \cdots \tag{1.10}$$

in which $\mathbf{q}_i^1$ is the force contributed to node i by element 1, $\mathbf{q}_i^2$ by element 2, etc. Clearly, only the elements which include point i will contribute non-zero forces, but for tidiness all the elements are included in the summation.

Substituting the forces contributing to node i from the definition (1.3) and noting that nodal variables $\mathbf{a}_i$ are common (thus omitting the superscript e), we have

$$\mathbf{r}_i = \left(\sum_{e=1}^{m} \mathbf{K}_{i1}^e \right) \mathbf{a}_1 + \left(\sum_{e=1}^{m} \mathbf{K}_{i2}^e \right) \mathbf{a}_2 + \cdots + \sum_{e=1}^{m} \mathbf{f}_i^e \tag{1.11}$$

where

$$\mathbf{f}^e = \mathbf{f}_p^e + \mathbf{f}_{\varepsilon_0}^e$$

The summation again only concerns the elements which contribute to node i. If all such equations are assembled we have simply

$$\mathbf{K}\mathbf{a} = \mathbf{r} - \mathbf{f} \tag{1.12}$$

in which the submatrices are

$$\begin{aligned} \mathbf{K}_{ij} &= \sum_{e=1}^{m} \mathbf{K}_{ij}^e \\ \mathbf{f}_i &= \sum_{e=1}^{m} \mathbf{f}_i^e \end{aligned} \tag{1.13}$$

with summations including all elements. This simple rule for assembly is very convenient because as soon as a coefficient for a particular element is found it can be put immediately into the appropriate 'location' specified in the computer. *This general assembly process can be found to be the common and fundamental feature of all finite element calculations and should be well understood by the reader*.

If different types of structural elements are used and are to be coupled it must be remembered that the rules of matrix summation permit this to be done only if these are of identical size. The individual submatrices to be added have therefore to be built up of the same number of individual components of force or displacement. Thus, for example, if a member capable of transmitting moments to a node is to be coupled at that node to one which in fact is hinged, it is necessary to complete the stiffness matrix of the latter by insertion of appropriate (zero) coefficients in the rotation or moment positions.

1.4 The boundary conditions

The system of equations resulting from Eq. (1.12) can be solved once the prescribed support displacements have been substituted. In the example of Fig. 1.1, where both components of displacement of nodes 1 and 6 are zero, this will mean

the substitution of

$$\mathbf{a}_1 = \mathbf{a}_6 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

which is equivalent to reducing the number of equilibrium equations (in this instance 12) by deleting the first and last pairs and thus reducing the total number of unknown displacement components to eight. It is, nevertheless, always convenient to assemble the equation according to relation (1.12) so as to include all the nodes.

Clearly, without substitution of a minimum number of prescribed displacements to prevent rigid body movements of the structure, it is impossible to solve this system, because the displacements cannot be uniquely determined by the forces in such a situation. This physically obvious fact will be interpreted mathematically as the matrix $\mathbf{K}$ being singular, i.e., not possessing an inverse. The prescription of appropriate displacements after the assembly stage will permit a unique solution to be obtained by deleting appropriate rows and columns of the various matrices.

If all the equations of a system are assembled, their form is

$$\begin{aligned} &\mathbf{K}_{11}\mathbf{a}_1 + \mathbf{K}_{12}\mathbf{a}_2 + \cdots = \mathbf{r}_1 - \mathbf{f}_1 \\ &\mathbf{K}_{21}\mathbf{a}_1 + \mathbf{K}_{22}\mathbf{a}_2 + \cdots = \mathbf{r}_2 - \mathbf{f}_2 \\ &\text{etc.} \end{aligned} \tag{1.14}$$

and it will be noted that if any displacement, such as $\mathbf{a}_1 = \bar{\mathbf{a}}_1$, is prescribed then the external 'force' $\mathbf{r}_1$ cannot be simultaneously specified and remains unknown. The first equation could then be *deleted* and substitution of known values of $\mathbf{a}_1$ made in the remaining equations. This process is computationally cumbersome and the same objective is served by adding a large number, $\alpha\mathbf{I}$, to the coefficient $\mathbf{K}_{11}$ and replacing the right-hand side, $\mathbf{r}_1 - \mathbf{f}_1$, by $\bar{\mathbf{a}}_1\alpha$. If α is very much larger than other stiffness coefficients this alteration effectively replaces the first equation by the equation

$$\alpha\mathbf{a}_1 = \alpha\bar{\mathbf{a}}_1 \tag{1.15}$$

that is, the required prescribed condition, but the whole system remains symmetric and minimal changes are necessary in the computation sequence. A similar procedure will apply to any other prescribed displacement. The above artifice was introduced by Payne and Irons.[23] An alternative procedure avoiding the assembly of equations corresponding to nodes with prescribed boundary values will be presented in Chapter 20.

When all the boundary conditions are inserted the equations of the system can be solved for the unknown displacements and stresses, and the internal forces in each element obtained.

1.5 Electrical and fluid networks

Identical principles of deriving element characteristics and of assembly will be found in many non-structural fields. Consider, for instance, the assembly of electrical resistances shown in Fig. 1.3.

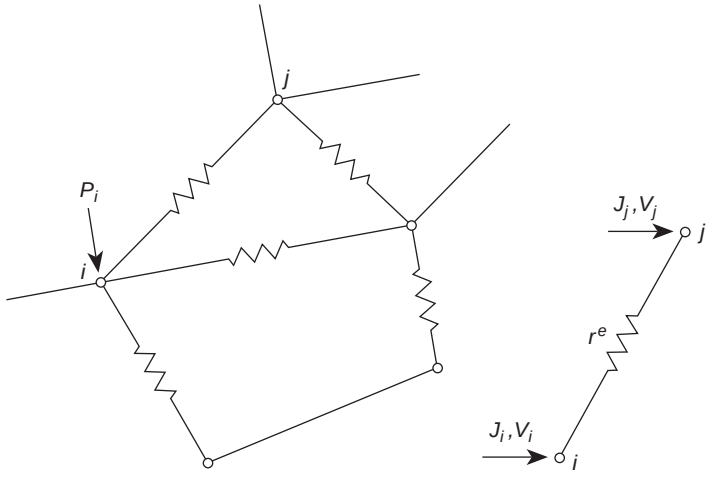


Fig. 1.3 A network of electrical resistances.

If a typical resistance element, *ij*, is isolated from the system we can write, by Ohm's law, the relation between the currents *entering* the element at the ends and the end voltages as

$$J_i^e = \frac{1}{r^e}(V_i - V_j)$$

$$J_j^e = \frac{1}{r^e}(V_j - V_i)$$

or in matrix form

$$\begin{Bmatrix} J_i^e \\ J_j^e \end{Bmatrix} = \frac{1}{r^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} V_i \\ V_j \end{Bmatrix}$$

which in our standard form is simply

$$\mathbf{J}^e = \mathbf{K}^e\mathbf{V}^e \tag{1.16}$$

This form clearly corresponds to the stiffness relationship (1.3); indeed if an external current were supplied along the length of the element the element 'force' terms could also be found.

To assemble the whole network the continuity of the potential (V) at the nodes is assumed and a current balance imposed there. If P_i now stands for the external input of current at node i we must have, with complete analogy to Eq. (1.11),

$$P_i = \sum_{j=1}^{n}\sum_{e=1}^{m} K_{ij}^e V_j \tag{1.17}$$

where the second summation is over all 'elements', and once again for all the nodes

$$\mathbf{P} = \mathbf{K}\mathbf{V} \tag{1.18}$$

in which

$$K_{ij} = \sum_{e=1}^{m} K_{ij}^e$$

Matrix notation in the above has been dropped since the quantities such as voltage and current, and hence also the coefficients of the 'stiffness' matrix, are scalars.

If the resistances were replaced by fluid-carrying pipes in which a laminar regime pertained, an identical formulation would once again result, with V standing for the hydraulic head and J for the flow.

For pipe networks that are usually encountered, however, the linear laws are in general not valid. Typically the flow–head relationship is of a form

$$J_i = c(V_i - V_j)^{\gamma} \tag{1.19}$$

where the index γ lies between 0.5 and 0.7. Even now it would still be possible to write relationships in the form (1.16) noting, however, that the matrices $\mathbf{K}^e$ are no longer arrays of constants but are known functions of $\mathbf{V}$. The final equations can once again be assembled but their form will be non-linear and in general iterative techniques of solution will be needed.

Finally it is perhaps of interest to mention the more general form of an electrical network subject to an alternating current. It is customary to write the relationships between the current and voltage in *complex form* with the resistance being replaced by complex impedance. Once again the standard forms of (1.16)–(1.18) will be obtained but with each quantity divided into real and imaginary parts.

Identical solution procedures can be used if the equality of the real and imaginary quantities is considered at each stage. Indeed with modern digital computers it is possible to use standard programming practice, making use of facilities available for dealing with complex numbers. Reference to some problems of this class will be made in the chapter dealing with vibration problems in Chapter 17.

1.6 The general pattern

An example will be considered to consolidate the concepts discussed in this chapter. This is shown in Fig. 1.4(*a*) where five discrete elements are interconnected. These may be of structural, electrical, or any other linear type. In the solution:

The first step is the determination of element properties from the geometric material and loading data. For each element the 'stiffness matrix' as well as the corresponding 'nodal loads' are found in the form of Eq. (1.3). Each element has its own identifying number and specified nodal connection. For example:

element	1	connection	1	3	4	
	2		1	4	2	
	3		2	5		
	4		3	6	7	4
	5		4	7	8	5

Assuming that properties are found in global coordinates we can enter each 'stiffness' or 'force' component in its position of the global matrix as shown in Fig. 1.4(*b*), Each shaded square represents a single coefficient or a submatrix of type $\mathbf{K}_{ij}$ if more than one quantity is being considered at the nodes. Here the separate contribution of each element is shown and the reader can verify the position of

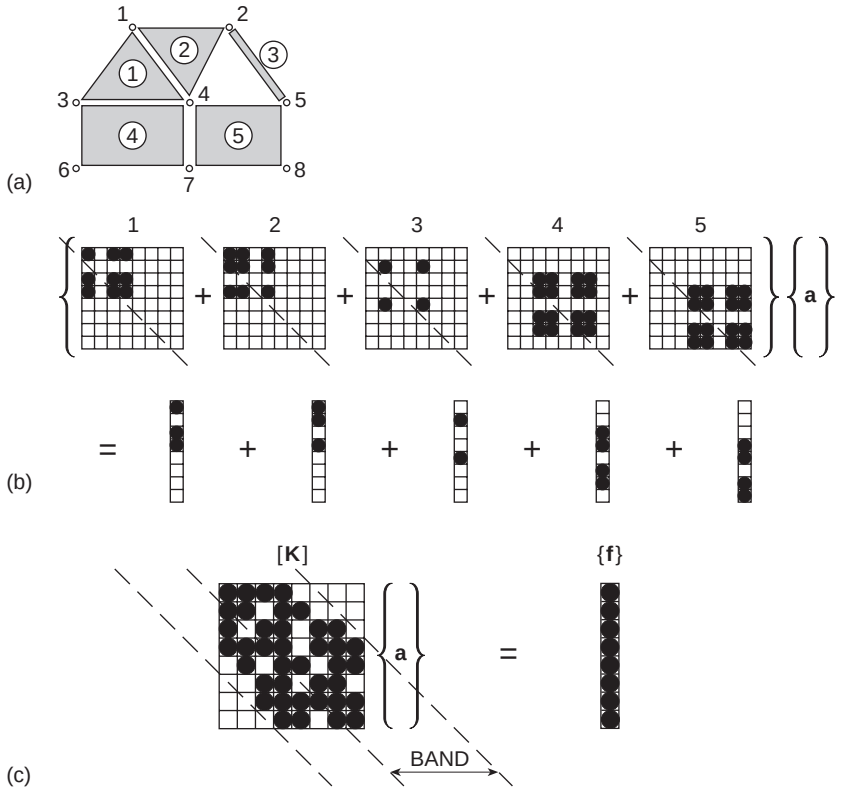


Fig. 1.4 The general pattern.

the coefficients. Note that the various types of 'elements' considered here present no difficulty in specification. (All 'forces', including nodal ones, are here associated with elements for simplicity.)

The second step is the assembly of the final equations of the type given by Eq. (1.12). This is accomplished according to the rule of Eq. (1.13) by *simple addition* of all numbers in the appropriate space of the global matrix. The result is shown in Fig. 1.4(*c*) where the non-zero coefficients are indicated by shading.

As the matrices are symmetric only the half above the diagonal shown needs, in fact, to be found.

All the non-zero coefficients are confined within a *band* or *profile* which can be calculated *a priori* for the nodal connections. Thus in computer programs only the storage of the elements within the upper half of the profile is necessary, as shown in Fig. 1.4(*c*).

The third step is the insertion of prescribed boundary conditions into the final assembled matrix, as discussed in Sec. 1.3. This is followed by the final step.

The final step solves the resulting equation system. Here many different methods can be employed, some of which will be discussed in Chapter 20. The general

subject of equation solving, though extremely important, is in general beyond the scope of this book.

The final step discussed above can be followed by substitution to obtain stresses, currents, or other desired *output* quantities.

All operations involved in structural or other network analysis are thus of an extremely simple and repetitive kind.

We can now define *the standard discrete system* as one in which such conditions prevail.

1.7 The standard discrete system

In the *standard discrete system*, whether it is structural or of any other kind, we find that:

1. A set of discrete parameters, say $\mathbf{a}_i$, can be identified which describes simultaneously the behaviour of each element, e, and of the whole system. We shall call these the *system parameters*.
2. For each element a set of quantities $\mathbf{q}_i^e$ can be computed in terms of the system parameters $\mathbf{a}_i$. The general function relationship can be non-linear
$$\mathbf{q}_i^e = \mathbf{q}_i^e(\mathbf{a}) \tag{1.20}$$
but in many cases a linear form exists giving
$$\mathbf{q}_i^e = \mathbf{K}_{i1}^e \mathbf{a}_1 + \mathbf{K}_{i2}^e \mathbf{a}_2 + \cdots + \mathbf{f}_i^e \tag{1.21}$$
3. The *system equations* are obtained by a simple addition
$$\mathbf{r}_i = \sum_{e=1}^{m} \mathbf{q}_i^e \tag{1.22}$$
where $\mathbf{r}_i$ are system quantities (often prescribed as zero).

   In the linear case this results in a system of equations
$$\mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{r} \tag{1.23}$$
such that
$$\mathbf{K}_{ij} = \sum_{e=1}^{m} \mathbf{K}_{ij}^e \qquad \mathbf{f}_i = \sum_{e=1}^{m} \mathbf{f}_i^e \tag{1.24}$$
from which the solution for the system variables $\mathbf{a}$ can be found after imposing necessary boundary conditions.

The reader will observe that this definition includes the structural, hydraulic, and electrical examples already discussed. However, it is broader. In general neither linearity nor symmetry of matrices need exist – although in many problems this will arise naturally. Further, the narrowness of interconnections existing in usual elements is not essential.

While much further detail could be discussed (we refer the reader to specific books for more exhaustive studies in the structural context[24–26]), we feel that the general exposé given here should suffice for further study of this book.

Only one further matter relating to the change of discrete parameters need be mentioned here. The process of so-called transformation of coordinates is vital in many contexts and must be fully understood.

1.8 Transformation of coordinates

It is often convenient to establish the characteristics of an individual element in a coordinate system which is different from that in which the external forces and displacements of the assembled structure or system will be measured. A different coordinate system may, in fact, be used for every element, to ease the computation. It is a simple matter to transform the coordinates of the displacement and force components of Eq. (1.3) to any other coordinate system. Clearly, it is necessary to do so before an assembly of the structure can be attempted.

Let the local coordinate system in which the element properties have been evaluated be denoted by a prime suffix and the common coordinate system necessary for assembly have no embellishment. The displacement components can be transformed by a suitable matrix of direction cosines $\mathbf{L}$ as

$$\mathbf{a}' = \mathbf{L}\mathbf{a} \tag{1.25}$$

As the corresponding force components must perform the same amount of work in either system†

$$\mathbf{q}^{\mathrm{T}}\mathbf{a} = \mathbf{q}'^{\mathrm{T}}\mathbf{a}' \tag{1.26}$$

On inserting (1.25) we have

$$\mathbf{q}^{\mathrm{T}}\mathbf{a} = \mathbf{q}'^{\mathrm{T}}\mathbf{L}\mathbf{a}$$

or

$$\mathbf{q} = \mathbf{L}^{\mathrm{T}}\mathbf{q}' \tag{1.27}$$

The set of transformations given by (1.25) and (1.27) is called *contravariant*.

To transform 'stiffnesses' which may be available in local coordinates to global ones note that if we write

$$\mathbf{q}' = \mathbf{K}'\mathbf{a}' \tag{1.28}$$

then by (1.27), (1.28), and (1.25)

$$\mathbf{q} = \mathbf{L}^{\mathrm{T}}\mathbf{K}'\mathbf{L}\mathbf{a}$$

or in global coordinates

$$\mathbf{K} = \mathbf{L}^{\mathrm{T}}\mathbf{K}'\mathbf{L} \tag{1.29}$$

The reader can verify the usefulness of the above transformations by reworking the sample example of the pin-ended bar, first establishing its stiffness in its length coordinates.

† With $(\ \ )^{\mathrm{T}}$ standing for the transpose of the matrix.

In many complex problems an external constraint of some kind may be imagined, enforcing the requirement (1.25) with the number of degrees of freedom of $\mathbf{a}$ and $\mathbf{a}'$ being quite different. Even in such instances the relations (1.26) and (1.27) continue to be valid.

An alternative and more general argument can be applied to many other situations of discrete analysis. We wish to replace a set of parameters $\mathbf{a}$ in which the system equations have been written by another one related to it by a transformation matrix $\mathbf{T}$ as

$$\mathbf{a} = \mathbf{T}\mathbf{b} \tag{1.30}$$

In the linear case the system equations are of the form

$$\mathbf{K}\mathbf{a} = \mathbf{r} - \mathbf{f} \tag{1.31}$$

and on the substitution we have

$$\mathbf{K}\mathbf{T}\mathbf{b} = \mathbf{r} - \mathbf{f} \tag{1.32}$$

The new system can be premultiplied simply by $\mathbf{T}^{\mathrm{T}}$, yielding

$$(\mathbf{T}^{\mathrm{T}}\mathbf{K}\mathbf{T})\mathbf{b} = \mathbf{T}^{\mathrm{T}}\mathbf{r} - \mathbf{T}^{\mathrm{T}}\mathbf{f} \tag{1.33}$$

which will preserve the symmetry of equations if the matrix $\mathbf{K}$ is symmetric. However, occasionally the matrix $\mathbf{T}$ is not square and expression (1.30) represents in fact *an approximation* in which a larger number of parameters $\mathbf{a}$ is *constrained*. Clearly the system of equations (1.32) gives more equations than are necessary for a solution of the reduced set of parameters $\mathbf{b}$, and the final expression (1.33) presents a reduced system which in some sense approximates the original one.

We have thus introduced the basic idea of approximation, which will be the subject of subsequent chapters where infinite sets of quantities are reduced to finite sets.

A historical development of the subject of finite element methods has been presented by the author.[27,28]

References

1. R.V. Southwell. *Relaxation Methods in Theoretical Physics*. Clarendon Press, 1946.
2. D.N. de G. Allen. *Relaxation Methods*. McGraw-Hill, 1955.
3. S.H. Crandall. *Engineering Analysis*. McGraw-Hill, 1956.
4. B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, 1972.
5. D. McHenry. A lattice analogy for the solution of plane stress problems. *J. Inst. Civ. Eng.*, **21**, 59–82, 1943.
6. A. Hrenikoff. Solution of problems in elasticity by the framework method. *J. Appl. Mech.*, **A8**, 169–75, 1941.
7. N.M. Newmark. Numerical methods of analysis in bars, plates and elastic bodies, in *Numerical Methods in Analysis in Engineering* (ed. L.E. Grinter), Macmillan, 1949.
8. J.H. Argyris. *Energy Theorems and Structural Analysis*. Butterworth, 1960 (reprinted from *Aircraft Eng.*, 1954–5).
9. M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *J. Aero. Sci.*, **23**, 805–23, 1956.

10. R.W. Clough. The finite element in plane stress analysis. *Proc. 2nd ASCE Conf. on Electronic Computation*. Pittsburgh, Pa., Sept. 1960.
11. Lord Rayleigh (J.W. Strutt). On the theory of resonance. *Trans. Roy. Soc.* (*London*), **A161**, 77–118, 1870.
12. W. Ritz. Über eine neue Methode zur Lösung gewissen Variations – Probleme der mathematischen Physik. *J. Reine Angew. Math.*, **135**, 1–61, 1909.
13. R. Courant. Variational methods for the solution of problems of equilibrium and vibration. *Bull. Am. Math. Soc.*, **49**, 1–23, 1943.
14. W. Prager and J.L. Synge. Approximation in elasticity based on the concept of function space. *Q. J. Appl. Math.*, **5**, 241–69, 1947.
15. L.F. Richardson. The approximate arithmetical solution by finite differences of physical problems. *Trans. Roy. Soc.* (*London*), **A210**, 307–57, 1910.
16. H. Liebman. Die angenäherte Ermittlung: harmonischen, functionen und konformer Abbildung. *Sitzber. Math. Physik Kl. Bayer Akad. Wiss. München*. **3**, 65–75, 1918.
17. R.S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.
18. C.F. Gauss, See *Carl Friedrich Gauss Werks*. Vol. VII, Göttingen, 1871.
19. B.G. Galerkin. Series solution of some problems of elastic equilibrium of rods and plates' (Russian). *Vestn. Inzh. Tech.*, **19**, 897–908, 1915.
20. C.B. Biezeno and J.J. Koch. Over een Nieuwe Methode ter Berekening van Vlokke Platen. *Ing. Grav.*, **38**, 25–36, 1923.
21. O.C. Zienkiewicz and Y.K. Cheung. The finite element method for analysis of elastic isotropic and orthotropic slabs. *Proc. Inst. Civ. Eng.*, **28**, 471–488, 1964.
22. R.V. Southwell. Stress calculation in frame works by the method of systematic relaxation of constraints, Part I & II. *Proc. Roy. Soc. London* (*A*), **151**, 56–95, 1935.
23. N.A. Payne and B.M. Irons, Private communication, 1963.
24. R.K. Livesley. *Matrix Methods in Structural Analysis*. 2nd ed., Pergamon Press, 1975.
25. J.S. Przemieniecki. *Theory of Matrix Structural Analysis*. McGraw-Hill, 1968.
26. H.C. Martin. *Introduction to Matrix Methods of Structural Analysis*. McGraw-Hill, 1966.
27. O.C. Zienkiewicz. Origins, milestones and directions of the finite element method. *Arch. Comp. Methods Eng.*, **2**, 1–48, 1995.
28. O.C. Zienkiewicz. Origins, milestones and directions of the finite element method – A personal view. *Handbook of Numerical Analysis*, **IV**, 3–65. Editors P.C. Ciarlet and J.L. Lions, North-Holland, 1996.

2

A direct approach to problems in elasticity

2.1 Introduction

The process of approximating the behaviour of a continuum by 'finite elements' which behave in a manner similar to the real, 'discrete', elements described in the previous chapter can be introduced through the medium of particular physical applications or as a general mathematical concept. We have chosen here to follow the first path, narrowing our view to a set of problems associated with structural mechanics which historically were the first to which the finite element method was applied. In Chapter 3 we shall generalize the concepts and show that the basic ideas are widely applicable.

In many phases of engineering the solution of stress and strain distributions in elastic continua is required. Special cases of such problems may range from two-dimensional plane stress or strain distributions, axisymmetric solids, plate bending, and shells, to fully three-dimensional solids. In all cases the number of interconnections between any 'finite element' isolated by some imaginary boundaries and the neighbouring elements is infinite. It is therefore difficult to see at first glance how such problems may be discretized in the same manner as was described in the preceding chapter for simpler structures. The difficulty can be overcome (and the approximation made) in the following manner:

1. The continuum is separated by imaginary lines or surfaces into a number of 'finite elements'.
2. The elements are assumed to be interconnected at a discrete number of nodal points situated on their boundaries and occasionally in their interior. In Chapter 6 we shall show that this limitation is not necessary. The displacements of these nodal points will be the basic unknown parameters of the problem, just as in simple, discrete, structural analysis.
3. A set of functions is chosen to define uniquely the state of displacement within each 'finite element' and on its boundaries in terms of its nodal displacements.
4. The displacement functions now define uniquely the state of strain within an element in terms of the nodal displacements. These strains, together with any initial strains and the constitutive properties of the material, will define the state of stress throughout the element and, hence, also on its boundaries.

5. A system of 'forces' concentrated at the nodes and equilibrating the boundary stresses and any distributed loads is determined, resulting in a stiffness relationship of the form of Eq. (1.3).

Once this stage has been reached the solution procedure can follow the standard discrete system pattern described earlier.

Clearly a series of approximations has been introduced. Firstly, it is not always easy to ensure that the chosen displacement functions will satisfy the requirement of displacement continuity between adjacent elements. Thus, the compatibility condition on such lines may be violated (though within each element it is obviously satisfied due to the uniqueness of displacements implied in their continuous representation). Secondly, by concentrating the equivalent forces at the nodes, equilibrium conditions are satisfied in the overall sense only. Local violation of equilibrium conditions within each element and on its boundaries will usually arise.

The choice of element shape and of the form of the displacement function for specific cases leaves many opportunities for the ingenuity and skill of the engineer to be employed, and obviously the degree of approximation which can be achieved will strongly depend on these factors.

The approach outlined here is known as the *displacement formulation*.[1,2]

So far, the process described is justified only intuitively, but what in fact has been suggested is equivalent to the minimization of the total potential energy of the system in terms of a prescribed displacement field. If this displacement field is defined in a suitable way, then convergence to the correct result must occur. The process is then equivalent to the well-known Rayleigh–Ritz procedure. This equivalence will be proved in a later section of this chapter where also a discussion of the necessary convergence criteria will be presented.

The recognition of the equivalence of the finite element method to a minimization process was late.[2,3] However, Courant in 1943[4]† and Prager and Synge[5] in 1947 proposed methods that are in essence identical.

This broader basis of the finite element method allows it to be extended to other continuum problems where a variational formulation is possible. Indeed, general procedures are now available for a finite element discretization of any problem defined by a properly constituted set of differential equations. Such generalizations will be discussed in Chapter 3, and throughout the book application to non-structural problems will be made. It will be found that the processes described in this chapter are essentially an application of trial-function and Galerkin-type approximations to a particular case of solid mechanics.

2.2 Direct formulation of finite element characteristics

The 'prescriptions' for deriving the characteristics of a 'finite element' of a continuum, which were outlined in general terms, will now be presented in more detailed mathematical form.

† It appears that Courant had anticipated the essence of the finite element method in general, and of a triangular element in particular, as early as 1923 in a paper entitled 'On a convergence principle in the calculus of variations.' Kön. Gesellschaft der Wissenschaften zu Göttingen, Nachrichten, Berlin, 1923. He states: 'We imagine a mesh of triangles covering the domain . . . the convergence principles remain valid for each triangular domain.'

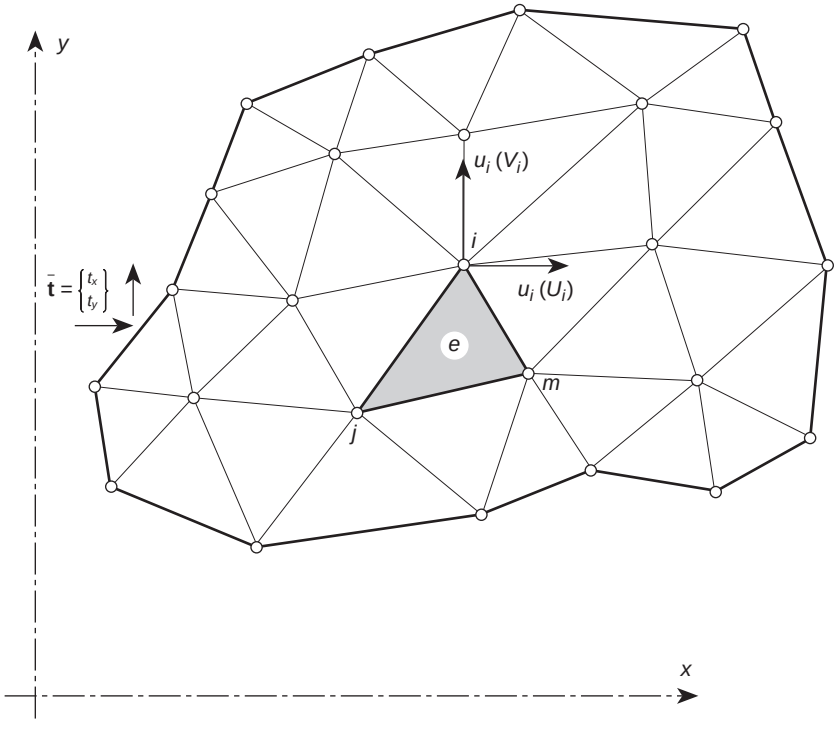


Fig. 2.1 A plane stress region divided into finite elements.

It is desirable to obtain results in a general form applicable to any situation, but to avoid introducing conceptual difficulties the general relations will be illustrated with a very simple example of plane stress analysis of a thin slice. In this a division of the region into triangular-shaped elements is used as shown in Fig. 2.1. Relationships of general validity will be placed in a box. Again, matrix notation will be implied.

2.2.1 Displacement function

A typical finite element, e, is defined by nodes, i, j, m, etc., and straight line boundaries. Let the displacements $\mathbf{u}$ at any point within the element be approximated as a column vector, $\hat{\mathbf{u}}$:

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_k \mathbf{N}_k \mathbf{a}_k^e = [\mathbf{N}_i, \mathbf{N}_j, \ldots] \begin{Bmatrix} \mathbf{a}_i \\ \mathbf{a}_j \\ \vdots \end{Bmatrix}^e = \mathbf{N}\mathbf{a}^e \tag{2.1}$$

in which the components of $\mathbf{N}$ are prescribed functions of position and $\mathbf{a}^e$ represents a listing of nodal displacements for a particular element.

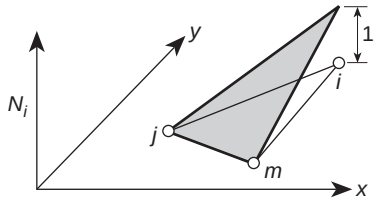


Fig. 2.2 Shape function N_i for one element.

In the case of plane stress, for instance,

$$\mathbf{u} = \begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix}$$

represents horizontal and vertical movements of a typical point within the element and

$$\mathbf{a}_i = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}$$

the corresponding displacements of a node i.

The functions $\mathbf{N}_i$, $\mathbf{N}_j$, $\mathbf{N}_m$ have to be chosen so as to give appropriate nodal displacements when the coordinates of the corresponding nodes are inserted in Eq. (2.1). Clearly, in general,

$$\mathbf{N}_i(x_i, y_i) = \mathbf{I} \quad \text{(identity matrix)}$$

while

$$\mathbf{N}_i(x_j, y_j) = \mathbf{N}_i(x_m, y_m) = \mathbf{0}, \qquad \text{etc.}$$

which is simply satisfied by suitable linear functions of x and y.

If both the components of displacement are specified in an identical manner then we can write

$$\mathbf{N}_i = N_i \mathbf{I}$$

and obtain N_i from Eq. (2.1) by noting that $N_i = 1$ at x_i, y_i but zero at other vertices.

The most obvious linear function in the case of a triangle will yield the shape of N_i of the form shown in Fig. 2.2. Detailed expressions for such a linear interpolation are given in Chapter 4, but at this stage can be readily derived by the reader.

The functions $\mathbf{N}$ will be called *shape functions* and will be seen later to play a paramount role in finite element analysis.

2.2.2 Strains

With displacements known at all points within the element the 'strains' at any point can be determined. These will always result in a relationship that can be written in

matrix notation as†

$$\boxed{\boldsymbol{\varepsilon} \approx \hat{\boldsymbol{\varepsilon}} = \mathbf{S}\mathbf{u}} \tag{2.2}$$

where $\mathbf{S}$ is a suitable linear operator. Using Eq. (2.1), the above equation can be approximated as

$$\boxed{\boldsymbol{\varepsilon} \approx \hat{\boldsymbol{\varepsilon}} = \mathbf{B}\mathbf{a}} \tag{2.3}$$

with

$$\boxed{\mathbf{B} = \mathbf{S}\mathbf{N}} \tag{2.4}$$

For the plane stress case the relevant strains of interest are those occurring in the plane and are defined in terms of the displacements by well-known relations[6] which define the operator $\mathbf{S}$:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x}, & 0 \\ 0, & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y}, & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix}$$

With the shape functions $\mathbf{N}_i$, $\mathbf{N}_j$, and $\mathbf{N}_m$ already determined, the matrix $\mathbf{B}$ can easily be obtained. If the linear form of these functions is adopted then, in fact, the strains will be constant throughout the element.

2.2.3 Stresses

In general, the material within the element boundaries may be subjected to initial strains such as may be due to temperature changes, shrinkage, crystal growth, and so on. If such strains are denoted by $\boldsymbol{\varepsilon}_0$ then the stresses will be caused by the difference between the actual and initial strains.

In addition it is convenient to assume that at the outset of the analysis the body is stressed by some known system of initial residual stresses $\boldsymbol{\sigma}_0$ which, for instance, could be measured, but the prediction of which is impossible without the full knowledge of the material's history. These stresses can simply be added on to the general definition. Thus, assuming general linear elastic behaviour, the relationship between stresses and strains will be linear and of the form

$$\boxed{\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0} \tag{2.5}$$

where $\mathbf{D}$ is an elasticity matrix containing the appropriate material properties.

† It is known that strain is a second-rank tensor by its transformation properties; however, in this book we will normally represent quantities using matrix (Voigt) notation. The interested reader is encouraged to consult Appendix B for the relations between tensor forms and matrix quantities.

Again, for the particular case of plane stress three components of stress corresponding to the strains already defined have to be considered. These are, in familiar notation

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}$$

and the **D** matrix may be simply obtained from the usual isotropic stress–strain relationship[6]

$$\varepsilon_x - (\varepsilon_x)_0 = \frac{1}{E}\sigma_x - \frac{\nu}{E}\sigma_y$$

$$\varepsilon_y - (\varepsilon_y)_0 = -\frac{\nu}{E}\sigma_x + \frac{1}{E}\sigma_y$$

$$\gamma_{xy} - (\gamma_{xy})_0 = \frac{2(1+\nu)}{E}\tau_{xy}$$

i.e., on solving,

$$\mathbf{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}$$

2.2.4 Equivalent nodal forces

Let

$$\mathbf{q}^e = \begin{Bmatrix} \mathbf{q}_i^e \\ \mathbf{q}_j^e \\ \vdots \end{Bmatrix}$$

define the nodal forces which are statically equivalent to the boundary stresses and distributed body forces on the element. Each of the forces $\mathbf{q}_i^e$ must contain the same number of components as the corresponding nodal displacement $\mathbf{a}_i$ and be ordered in the appropriate, corresponding directions.

The distributed body forces **b** are defined as those acting on a unit volume of material within the element with directions corresponding to those of the displacements **u** at that point.

In the particular case of plane stress the nodal forces are, for instance,

$$\mathbf{q}_i^e = \begin{Bmatrix} U_i \\ V_i \end{Bmatrix}^e$$

with components U and V corresponding to the directions of u and v displacements, and the distributed body forces are

$$\mathbf{b} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix}$$

in which b_x and b_y are the 'body force' components.

To make the nodal forces statically equivalent to the actual boundary stresses and distributed body forces, the simplest procedure is to impose an arbitrary (virtual) nodal displacement and to equate the external and internal work done by the various forces and stresses during that displacement.

Let such a virtual displacement be $\delta\mathbf{a}^e$ at the nodes. This results, by Eqs (2.1) and (2.2), in displacements and strains within the element equal to

$$\delta\mathbf{u} = \mathbf{N}\,\delta\mathbf{a}^e \quad \text{and} \quad \delta\boldsymbol{\varepsilon} = \mathbf{B}\,\delta\mathbf{a}^e \tag{2.6}$$

respectively.

The work done by the nodal forces is equal to the sum of the products of the individual force components and corresponding displacements, i.e., in matrix language

$$\delta\mathbf{a}^{e\mathrm{T}}\mathbf{q}^e \tag{2.7}$$

Similarly, the internal work per unit volume done by the stresses and distributed body forces is

$$\delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma} - \delta\mathbf{u}^{\mathrm{T}}\mathbf{b} \tag{2.8}$$

or†

$$\delta\mathbf{a}^{\mathrm{T}}(\mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma} - \mathbf{N}^{\mathrm{T}}\mathbf{b}) \tag{2.9}$$

Equating the external work with the total internal work obtained by integrating over the volume of the element, V^e, we have

$$\delta\mathbf{a}^{e\mathrm{T}}\mathbf{q}^e = \delta\mathbf{a}^{e\mathrm{T}}\left(\int_{V^e}\mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d(vol)} - \int_{V^e}\mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d(vol)}\right) \tag{2.10}$$

As this relation is valid for any value of the virtual displacement, the multipliers must be equal. Thus

$$\boxed{\mathbf{q}^e = \int_{V^e}\mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d(vol)} - \int_{V^e}\mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d(vol)}} \tag{2.11}$$

This statement is valid quite generally for any stress–strain relation. With the linear law of Eq. (2.5) we can write Eq. (2.11) as

$$\boxed{\mathbf{q}^e = \mathbf{K}^e\mathbf{a}^e + \mathbf{f}^e} \tag{2.12}$$

where

$$\boxed{\mathbf{K}^e = \int_{V^e}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d(vol)}} \tag{2.13a}$$

and

$$\boxed{\mathbf{f}^e = -\int_{V^e}\mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d(vol)} - \int_{V^e}\mathbf{B}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}_0\,\mathrm{d(vol)} + \int_{V^e}\mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}_0\,\mathrm{d(vol)}} \tag{2.13b}$$

† Note that by the rules of matrix algebra for the transpose of products

$$(\mathbf{A}\mathbf{B})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$

In the last equation the three terms represent forces due to body forces, initial strain, and initial stress respectively. The relations have the characteristics of the discrete structural elements described in Chapter 1.

If the initial stress system is self-equilibrating, as must be the case with normal residual stresses, then the forces given by the initial stress term of Eq. (2.13b) are identically zero after assembly. Thus frequent evaluation of this force component is omitted. However, if for instance a machine part is manufactured out of a block in which residual stresses are present or if an excavation is made in rock where known tectonic stresses exist a removal of material will cause a force imbalance which results from the above term.

For the particular example of the plane stress triangular element these characteristics will be obtained by appropriate substitution. It has already been noted that the **B** matrix in that example was not dependent on the coordinates; hence the integration will become particularly simple.

The interconnection and solution of the whole assembly of elements follows the simple structural procedures outlined in Chapter 1. In general, external concentrated forces may exist at the nodes and the matrix

$$\mathbf{r} = \begin{Bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_n \end{Bmatrix} \tag{2.14}$$

will be added to the consideration of equilibrium at the nodes.

A note should be added here concerning elements near the boundary. If, at the boundary, displacements are specified, no special problem arises as these can be satisfied by specifying some of the nodal parameters **a**. Consider, however, the boundary as subject to a distributed external loading, say $\bar{\mathbf{t}}$ per unit area. A loading term on the nodes of the element which has a boundary face A^e will now have to be added. By the virtual work consideration, this will simply result in

$$\boxed{\mathbf{f}^e = -\int_{A^e} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d(area)}} \tag{2.15}$$

with the integration taken over the boundary area of the element. It will be noted that $\bar{\mathbf{t}}$ must have the same number of components as **u** for the above expression to be valid.

Such a boundary element is shown again for the special case of plane stress in Fig. 2.1. An integration of this type is sometimes not carried out explicitly. Often by 'physical intuition' the analyst will consider the boundary loading to be represented simply by concentrated loads acting on the boundary nodes and calculate these by direct static procedures. In the particular case discussed the results will be identical.

Once the nodal displacements have been determined by solution of the overall 'structural' type equations, the stresses at any point of the element can be found from the relations in Eqs (2.3) and (2.5), giving

$$\boxed{\boldsymbol{\sigma} = \mathbf{D}\mathbf{B}\mathbf{a}^e - \mathbf{D}\boldsymbol{\varepsilon}_0 + \boldsymbol{\sigma}_0} \tag{2.16}$$

in which the typical terms of the relationship of Eq. (1.4) will be immediately recognized, the element stress matrix being

$$\mathbf{Q}^e = \mathbf{DB} \tag{2.17}$$

To this the stresses

$$\boldsymbol{\sigma}_{\varepsilon_0} = -\mathbf{D}\boldsymbol{\varepsilon}_0 \quad \text{and} \quad \boldsymbol{\sigma}_0 \tag{2.18}$$

have to be added.

2.2.5 Generalized nature of displacements, strains, and stresses

The meaning of displacements, strains, and stresses in the illustrative case of plane stress was obvious. In many other applications, shown later in this book, this terminology may be applied to other, less obvious, quantities. For example, in considering plate elements the 'displacement' may be characterized by the lateral deflection and the slopes of the plate at a particular point. The 'strains' will then be defined as the curvatures of the middle surface and the 'stresses' as the corresponding internal bending moments (see Volume 2).

All the expressions derived here are generally valid provided the sum product of displacement and corresponding load components truly represents the external work done, while that of the 'strain' and corresponding 'stress' components results in the total internal work.

2.3 Generalization to the whole region – internal nodal force concept abandoned

In the preceding section the virtual work principle was applied to a single element and the concept of equivalent nodal force was retained. The assembly principle thus followed the conventional, direct equilibrium, approach.

The idea of nodal forces contributed by elements replacing the continuous interaction of stresses between elements presents a conceptual difficulty. However, it has a considerable appeal to 'practical' engineers and does at times allow an interpretation which otherwise would not be obvious to the more rigorous mathematician. There is, however, no need to consider each element individually and the reasoning of the previous section may be applied directly to the whole continuum.

Equation (2.1) can be interpreted as applying to the whole structure, that is,

$$\mathbf{u} = \tilde{\mathbf{N}}\mathbf{a} \tag{2.19}$$

in which $\mathbf{a}$ lists all the nodal points and

$$\tilde{\mathbf{N}}_i = \mathbf{N}_i^e \tag{2.20}$$

when the point concerned is within a particular element e and i is a point associated with that element. If point i does not occur within the element (see Fig. 2.3)

$$\tilde{\mathbf{N}}_i = \mathbf{0} \tag{2.21}$$

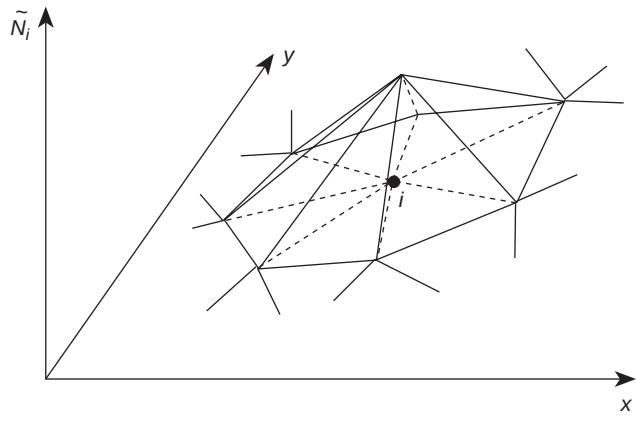


Fig. 2.3. A 'global' shape function – $\bar{N}_i$

Matrix $\bar{\mathbf{B}}$ can be similarly defined and we shall drop the bar superscript, considering simply that the shape functions, etc., are always defined over the whole region V.

For any virtual displacement $\delta\mathbf{a}$ we can now write the sum of internal and external work for the whole region as

$$-\delta\mathbf{a}^{\mathrm{T}}\mathbf{r} = \int_V \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}V + \int_A \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}A - \int_V \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}V \tag{2.22}$$

In the above equation $\delta\mathbf{a}$, $\delta\mathbf{u}$, and $\delta\boldsymbol{\varepsilon}$ can be completely arbitrary, providing they stem from a continuous displacement assumption. If for convenience we assume they are simply variations linked by the relations (2.19) and (2.3) we obtain, on substitution of the constitutive relation (2.5), a system of algebraic equations

$$\boxed{\mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{r}} \tag{2.23}$$

where

$$\boxed{\mathbf{K} = \int_V \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d}V} \tag{2.24a}$$

and

$$\boxed{\mathbf{f} = -\int_V \mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}V - \int_A \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}A - \int_V \mathbf{B}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}_0\,\mathrm{d}V + \int_V \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}_0\,\mathrm{d}V} \tag{2.24b}$$

The integrals are taken over the whole volume V and over the whole surface area A on which the tractions are given.

It is immediately obvious from the above that

$$\boxed{\mathbf{K}_{ij} = \sum \mathbf{K}_{ij}^e} \qquad \boxed{\mathbf{f}_i = \sum \mathbf{f}_i^e} \tag{2.25}$$

by virtue of the property of definite integrals requiring that the total be the sum of the parts:

$$\int_V (\quad)\,\mathrm{d}V = \sum \int_{V^e} (\quad)\,\mathrm{d}V \tag{2.26}$$

The same is obviously true for the surface integrals in Eq. (2.25). We thus see that the 'secret' of the approximation possessing the required behaviour of a 'standard discrete system of Chapter 1' lies simply in the requirement of writing the relationships in integral form.

The assembly rule as well as the whole derivation has been achieved without involving the concept of 'interelement forces' (i.e., $\mathbf{q}^e$). In the remainder of this book the element superscript will be dropped unless specifically needed. Also no differentiation between element and system shape functions will be made.

However, an important point arises immediately. In considering the virtual work for the whole system [Eq. (2.22)] and equating this to the sum of the element contributions it is implicitly assumed that no discontinuity in displacement between adjacent elements develops. If such a discontinuity developed, a contribution equal to the work done by the stresses in the separations would have to be added.

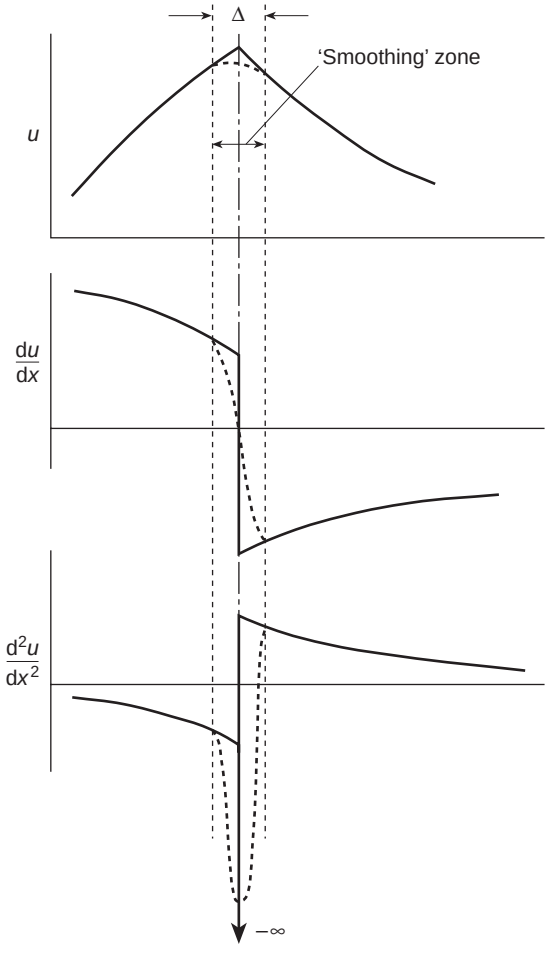


Fig. 2.4 Differentiation of a function with slope discontinuity (C_0 continuous).

Put in other words, we require that the terms integrated in Eq. (2.26) be finite. These terms arise from the shape functions N_i used in defining the displacement $\mathbf{u}$ [by Eq. (2.19)] and its derivatives associated with the definition of strain [viz. Eq. (2.3)]. If, for instance, the 'strains' are defined by first derivatives of the functions $\mathbf{N}$, the displacements must be continuous. In Fig. 2.4 we see how first derivatives of continuous functions may involve a 'jump' but are still finite, while second derivatives may become infinite. Such functions we call C_0 continuous.

In some problems the 'strain' in a generalized sense may be defined by second derivatives. In such cases we shall obviously require that both the function $\mathbf{N}$ and its slope (first derivative) be continuous. Such functions are more difficult to derive but we shall make use of them in plate and shell problems (see Volume 2). The continuity involved now is called C_1 continuity.

2.4 Displacement approach as a minimization of total potential energy

The principle of virtual displacements used in the previous sections ensured satisfaction of equilibrium conditions within the limits prescribed by the assumed displacement pattern. Only if the virtual work equality for all, arbitrary, variations of displacement was ensured would the equilibrium be complete.

As the number of parameters of $\mathbf{a}$ which prescribes the displacement increases without limit then ever closer approximation of all equilibrium conditions can be ensured.

The virtual work principle as written in Eq. (2.22) can be restated in a different form if the virtual quantities $\delta\mathbf{a}$, $\delta\mathbf{u}$, and $\delta\boldsymbol{\varepsilon}$ are considered as *variations* of the real quantities.

Thus, for instance, we can write

$$\delta\left(\mathbf{a}^{\mathrm{T}}\mathbf{r} + \int_V \mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}V + \int_A \mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}A\right) = -\delta W \tag{2.27}$$

for the first three terms of Eq. (2.22), where *W is the potential energy of the external loads*. The above is certainly true if $\mathbf{r}$, $\mathbf{b}$, and $\bar{\mathbf{t}}$ are conservative (or independent of displacement).

The last term of Eq. (2.22) can, for elastic materials, be written as

$$\delta U = \int_V \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}V \tag{2.28}$$

where U is the 'strain energy' of the system. For the elastic, linear material described by Eq. (2.5) the reader can verify that

$$U = \frac{1}{2}\int_V \boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}\,\mathrm{d}V - \int_V \boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}_0\,\mathrm{d}V + \int_V \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}_0\,\mathrm{d}V \tag{2.29}$$

will, after differentiation, yield the correct expression providing $\mathbf{D}$ is a symmetric matrix. (This is indeed a necessary condition for a single-valued U to exist.)

Thus instead of Eq. (2.22) we can write simply

$$\delta(U + W) = \delta(\Pi) = 0 \tag{2.30}$$

in which the quantity Π is called the *total potential energy*.

The above statement means that for equilibrium to be ensured the *total potential energy must be stationary* for variations of admissible displacements. The finite element equations derived in the previous section [Eqs (2.23)–(2.25)] are simply the statements of this variation with respect to displacements constrained to a finite number of parameters **a** and could be written as

$$\frac{\partial \Pi}{\partial \mathbf{a}} = \begin{Bmatrix} \dfrac{\partial \Pi}{\partial \mathbf{a}_1} \\ \dfrac{\partial \Pi}{\partial \mathbf{a}_2} \\ \vdots \end{Bmatrix} = \mathbf{0} \tag{2.31}$$

It can be shown that in stable elastic situations the total potential energy is not only stationary but is a minimum.[7] *Thus the finite element process seeks such a minimum within the constraint of an assumed displacement pattern.*

The greater the degrees of freedom, the more closely will the solution approximate the true one, ensuring complete equilibrium, providing the true displacement can, in the limit, be represented. The necessary convergence conditions for the finite element process could thus be derived. Discussion of these will, however, be deferred to subsequent sections.

It is of interest to note that if true equilibrium requires an absolute minimum of the total potential energy, Π, a finite element solution by the displacement approach will always provide an approximate Π greater than the correct one. *Thus a bound on the value of the total potential energy is always achieved.*

If the functional Π could be specified, *a priori*, then the finite element equations could be derived directly by the differentiation specified by Eq. (2.31).

The well-known Rayleigh[8]–Ritz[9] process of approximation frequently used in elastic analysis uses precisely this approach. The total potential energy expression is formulated and the displacement pattern is assumed to vary with a finite set of undetermined parameters. A set of simultaneous equations minimizing the total potential energy with respect to these parameters is set up. Thus the finite element process as described so far can be considered to be the Rayleigh–Ritz procedure. The difference is only in the manner in which the assumed displacements are prescribed. In the Ritz process traditionally used these are usually given by expressions valid throughout the whole region, thus leading to simultaneous equations in which no banding occurs and the coefficient matrix is full. In the finite element process this specification is usually piecewise, each nodal parameter influencing only adjacent elements, and thus a sparse and usually banded matrix of coefficients is found.

By its nature the conventional Ritz process is limited to relatively simple geometrical shapes of the total region while this limitation only occurs in finite element analysis in the element itself. Thus complex, realistic, configurations can be assembled from relatively simple element shapes.

A further difference in kind is in the usual association of the undetermined parameter with a particular nodal displacement. This allows a simple physical interpretation invaluable to an engineer. Doubtless much of the popularity of the finite element process is due to this fact.

2.5 Convergence criteria

The assumed shape functions limit the infinite degrees of freedom of the system, and the true minimum of the energy may never be reached, irrespective of the fineness of subdivision. To ensure convergence to the correct result certain simple requirements must be satisfied. Obviously, for instance, the displacement function should be able to represent the true displacement distribution as closely as desired. It will be found that this is not so if the chosen functions are such that straining is possible when the element is subjected to rigid body displacements. Thus, the first criterion that the displacement function must obey is as follows:

> *Criterion* 1. The displacement function chosen should be such that it does not permit straining of an element to occur when the nodal displacements are caused by a rigid body motion.

This self-evident condition can be violated easily if certain types of function are used; care must therefore be taken in the choice of displacement functions.

A second criterion stems from similar requirements. Clearly, as elements get smaller nearly constant strain conditions will prevail in them. If, in fact, constant strain conditions exist, it is most desirable for good accuracy that a finite size element is able to reproduce these exactly. It is possible to formulate functions that satisfy the first criterion but at the same time require a strain variation throughout the element when the nodal displacements are compatible with a constant strain solution. Such functions will, in general, not show good convergence to an accurate solution and cannot, even in the limit, represent the true strain distribution. The second criterion can therefore be formulated as follows:

> *Criterion* 2. The displacement function has to be of such a form that if nodal displacements are compatible with a constant strain condition such constant strain will in fact be obtained. (In this context again a generalized 'strain' definition is implied.)

It will be observed that Criterion 2 in fact incorporates the requirement of Criterion 1, as rigid body displacements are a particular case of constant strain – with a value of zero. This criterion was first stated by Bazeley *et al.*[10] in 1965. *Strictly, both criteria need only be satisfied in the limit as the size of the element tends to zero*. However, the imposition of these criteria on elements of finite size leads to improved accuracy, although in certain situations (such as illustrated by the axisymmetric analysis of Chapter 5) the imposition of the second one is not possible or essential.

Lastly, as already mentioned in Sec. 2.3, it is implicitly assumed in this derivation that no contribution to the virtual work arises at element interfaces. It therefore appears necessary that the following criterion be included:

> *Criterion* 3. The displacement functions should be chosen such that the strains at the interface between elements are finite (even though they may be discontinuous).

This criterion implies a certain continuity of displacements between elements. In the case of strains being defined by first derivatives, as in the plane stress example quoted here, the displacements only have to be continuous. If, however, as in the

plate and shell problems, the 'strains' are defined by second derivatives of deflections, first derivatives of these have also to be continuous.[2]

The above criteria are included mathematically in a statement of 'functional completeness' and the reader is referred elsewhere for full mathematical discussion.[11–16] The 'heuristic' proof of the convergence requirements given here is sufficient for practical purposes in all but the most pathological cases and we shall generalize all of the above criteria in Section 3.6 and more fully in Chapter 10, where we shall present a universal test which justifies convergence even if some of the above criteria are violated.

2.6 Discretization error and convergence rate

In the foregoing sections we have assumed that the approximation to the displacement as represented by Eq. (2.1) will yield the exact solution in the limit as the size h of elements decreases. The arguments for this are simple: if the expansion is capable, in the limit, of exactly reproducing any displacement form conceivable in the continuum, then as the solution of each approximation is unique it must approach, in the limit of $h \to 0$, the unique exact solution. In some cases the exact solution is indeed obtained with a finite number of subdivisions (or even with one element only) if the *polynomial expansion is used in that element and if this can fit exactly the correct solution*. Thus, for instance, if the exact solution is of the form of a quadratic polynomial *and* the shape functions include all the polynomials of that order, the approximation will yield the exact answer.

The last argument helps in determining the order of convergence of the finite element procedure as the exact solution can always be expanded in the vicinity of any point (or node) i as a polynomial:

$$\mathbf{u} = \mathbf{u}_i + \left(\frac{\partial \mathbf{u}}{\partial x}\right)_i (x - x_i) + \left(\frac{\partial \mathbf{u}}{\partial y}\right)_i (y - y_i) + \cdots \tag{2.32}$$

If within an element of 'size' h a polynomial expansion of degree p is employed, this can fit locally the Taylor expansion up to that degree and, as $x - x_i$ and $y - y_i$ are of the order of magnitude h, the error in $\mathbf{u}$ will be of the order $O(h^{p+1})$. Thus, for instance, in the case of the plane elasticity problem discussed, we used a linear expansion and $p = 1$. We should therefore expect a *convergence* rate of order $O(h^2)$, i.e., the error in displacement being reduced to $\frac{1}{4}$ for a halving of the mesh spacing.

By a similar argument the strains (or stresses) which are given by the mth derivatives of displacement should converge with an error of $O(h^{p+1-m})$, i.e., as $O(h)$ in the example quoted, where $m = 1$. The strain energy, being given by the square of the stresses, will show an error of $O(h^{2(p+1-m)})$ or $O(h^2)$ in the plane stress example.

The arguments given here are perhaps a trifle 'heuristic' from a mathematical viewpoint – they are, however, true[15,16] and correctly give the orders of convergence, which can be expected to be achieved asymptotically as the element size tends to zero and if the exact solution does not contain singularities. Such singularities may result in infinite values of the coefficients in terms omitted in the Taylor expansion of Eq. (2.32) and invalidate the arguments. However, in many well-behaved problems the mere determination of the order of convergence often suffices to extrapolate the

solution to the correct result. Thus, for instance, if the displacement converges at $O(h^2)$ and we have two approximate solutions u^1 and u^2 obtained with meshes of size h and $h/2$, we can write, with u being the exact solution,

$$\frac{u^1 - u}{u^2 - u} = \frac{O(h^2)}{O(h/2)^2} = 4 \tag{2.33}$$

From the above an (almost) exact solution u can be predicted. This type of extrapolation was first introduced by Richardson[17] and is of use if convergence is monotonic and nearly asymptotic.

We shall return to the important question of estimating errors due to the discretization process in Chapter 14 and will show that much more precise methods than those arising from convergence rate considerations are possible today. Indeed automatic mesh refinement processes are being introduced so that the specified accuracy can be achieved (viz. Chapter 15).

Discretization error is not the only error possible in a finite element computation. In addition to obvious mistakes which can occur when using computers, errors due to *round-off* are always possible. With the computer operating on numbers rounded off to a finite number of digits, a reduction of accuracy occurs every time differences between 'like' numbers are being formed. In the process of equation solving many subtractions are necessary and accuracy decreases. Problems of matrix conditioning, etc., enter here and the user of the finite element method must at all times be aware of accuracy limitations which simply do not allow the exact solution ever to be obtained. Fortunately in many computations, by using modern machines which carry a large number of significant digits, these errors are often small.

The question of errors arising from the algebraic processes will be stressed in Chapter 20 dealing with computation procedures.

2.7 Displacement functions with discontinuity between elements – non-conforming elements and the patch test

In some cases considerable difficulty is experienced in finding displacement functions for an element which will automatically be continuous along the whole interface between adjacent elements.

As already pointed out, the discontinuity of displacement will cause infinite strains at the interfaces, a factor ignored in this formulation because the energy contribution is limited to the elements themselves.

However, if, in the limit, as the size of the subdivision decreases continuity is restored, then the formulation already obtained will still tend to the correct answer. This condition is always reached if

(a) a constant strain condition automatically ensures displacement continuity and
(b) the constant strain criterion of the previous section is satisfied.

To test that such continuity is achieved for any mesh configuration when using such *non-conforming* elements it is necessary to impose, on an arbitrary patch of elements, nodal displacements corresponding to any state of constant strain. *If*

nodal equilibrium is simultaneously achieved without the imposition of external, nodal, forces and if a state of constant stress is obtained, then clearly no external work has been lost through interelement discontinuity.

Elements which pass such a *patch test* will converge, and indeed at times non-conforming elements will show a superior performance to conforming elements.

The patch test was first introduced by Irons[10] and has since been demonstrated to give a sufficient condition for convergence.[16,18–22] The concept of the patch test can be generalized to give information on the rate of convergence which can be expected from a given element.

We shall return to this problem in detail in Chapter 10 where the test will be fully discussed.

2.8 Bound on strain energy in a displacement formulation

While the approximation obtained by the finite element displacement approach always overestimates the true value of Π, the total potential energy (the absolute minimum corresponding to the exact solution), this is not directly useful in practice. It is, however, possible to obtain a more useful limit in special cases.

Consider in particular the problem in which no 'initial' strains or initial stresses exist. Now by the principle of energy conservation the strain energy will be equal to the work done by the external loads which increase uniformly from zero.[23] This work done is equal to $-\frac{1}{2}W$ where W is the potential energy of the loads.

Thus

$$U + \tfrac{1}{2}W = 0 \tag{2.34}$$

or

$$\Pi = U + W = -U \tag{2.35}$$

whether an exact or approximate displacement field is assumed.

Thus in the above case the approximate solution always *underestimates* the value of U and a displacement solution is frequently referred to as the *lower bound solution*.

If only one external concentrated load R is present the strain energy bound immediately informs us that the deflection under this load has been underestimated (as $U = -\frac{1}{2}W = \frac{1}{2}\mathbf{r}^{\mathrm{T}}\mathbf{a}$). In more complex loading cases the usefulness of this bound is limited as neither local deflections nor stresses, i.e., the quantities of real engineering interest, can be bounded.

It is important to remember that this bound on strain energy is only valid in the absence of any initial stresses or strains.

The expression for U in this case can be obtained from Eq. (2.29) as

$$U = \tfrac{1}{2}\int_V \boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}\,\mathrm{d(vol)} \tag{2.36}$$

which becomes by Eq. (2.2) simply

$$U = \tfrac{1}{2}\mathbf{a}^{\mathrm{T}}\left[\int_V \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d(vol)}\right]\mathbf{a} = \tfrac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} \tag{2.37}$$

a 'quadratic' matrix form in which $\mathbf{K}$ is the 'stiffness' matrix previously discussed.

The above energy expression is always positive from physical considerations. It follows therefore that the matrix **K** occurring in all the finite element assemblies is not only symmetric but is 'positive definite' (a property defined in fact by the requirements that the quadratic form should always be greater than or equal to zero).

This feature is of importance when the numerical solution of the simultaneous equations involved is considered, as simplifications arise in the case of 'symmetric positive definite' equations.

2.9 Direct minimization

The fact that the finite element approximation reduces to the problem of minimizing the total potential energy Π defined in terms of a finite number of nodal parameters led us to the formulation of the simultaneous set of equations given symbolically by Eq. (2.31). This is the most usual and convenient approach, especially in linear solutions, but other search procedures, now well developed in the field of optimization, could be used to estimate the lowest value of Π. In this text we shall continue with the simultaneous equation process but the interested reader could well bear the alternative possibilities in mind.[24,25]

2.10 An example

The concepts discussed and the general formulation cited are a little abstract and readers may at this stage seek to test their grasp of the nature of the approximations derived. While detailed computations of a two-dimensional element system are performed using the computer, we can perform a simple hand calculation on a one-dimensional finite element of a beam. Indeed, this example will allow us to introduce the concept of generalized stresses and strains in a simple manner.

Consider the beam shown in Fig. 2.5. The generalized 'strain' here is the curvature. Thus we have

$$\boldsymbol{\varepsilon} \equiv \kappa = -\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}$$

where w is the deflection, which is the basic unknown. The generalized stress (in the absence of shear deformation) will be the bending moment M, which is related to the 'strain' as

$$\boldsymbol{\sigma} \equiv M = -EI\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}$$

Thus immediately we have, using the general notation of previous sections,

$$\mathbf{D} \equiv EI$$

If the displacement w is discretized we can write

$$w \equiv \mathbf{Na}$$

for the whole system or, for an individual element, ij.

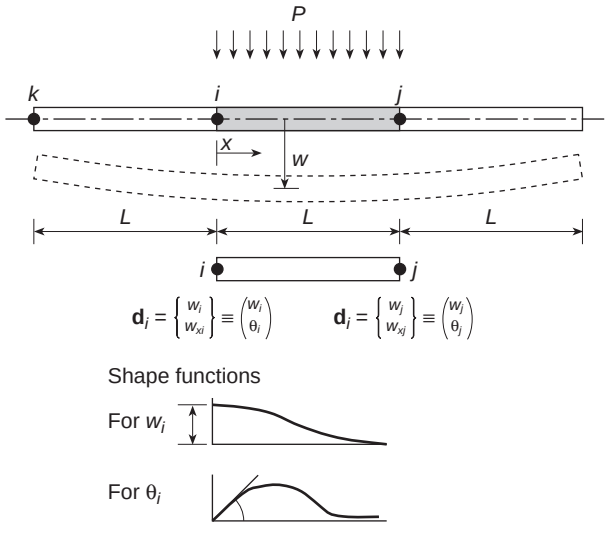


Fig. 2.5 A beam element and its shape functions.

In this example the strains are expressed as the second derivatives of displacement and it is necessary to ensure that both w and its slope

$$w_x \equiv \frac{\mathrm{d}w}{\mathrm{d}x} = \theta$$

be continuous between elements. This is easily accomplished if the nodal parameters are taken as the values of w and the slope, θ. Thus,

$$\mathbf{a}_i = \begin{Bmatrix} w_i \\ \theta_i \end{Bmatrix}$$

The shape functions will now be derived. If we accept that in an element two nodes (i.e., four variables) define the deflected shape we can assume this to be given by a cubic

$$w = \alpha_1 + \alpha_2 s + \alpha_3 s^2 + \alpha_4 s^3 \qquad \text{where } s = \frac{x}{L}.$$

This will define the shape functions corresponding to w_i and θ_i by taking for each a cubic giving unity for the appropriate points ($x = 0$, L or $s = 0, 1$) and zero for other quantities, as shown in Fig. 2.5.

The expressions for the shape functions can be written for the element shown as

$$\mathbf{N}_i = [1 - 3s^2 + 2s^3, L(s - 2s^2 + s^3)]$$

$$\mathbf{N}_j = [3s^2 - 2s^3, L(-s^2 + s^3)]$$

Immediately we can write

$$\mathbf{B}_i = -\frac{\mathrm{d}^2\mathbf{N}_i}{\mathrm{d}x^2} = \frac{1}{L^2}[6 - 12s, L(4 - 6s)]$$

$$\mathbf{B}_i = -\frac{\mathrm{d}^2\mathbf{N}_j}{\mathrm{d}x^2} = \frac{1}{L^2}[-6 + 12s, L(2 - 6s)]$$

and the stiffness matrices for the element can be written as

$$\mathbf{K}_{ij}^{e} = \int_0^L \mathbf{B}_i^{\mathrm{T}} EI\, \mathbf{B}_j \, \mathrm{d}x = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}$$

We shall leave the detailed calculation of this and the 'forces' corresponding to a uniformly distributed load p (assumed constant on ij and zero elsewhere) to the reader. It will be observed that the final assembled equations for a node i are of the form linking three nodal displacements i,j,k. Explicitly these equations are for elements of equal length L:

$$EI \begin{bmatrix} -12/L^3, & -6/L^2 \\ 6/L^2, & 2/L \end{bmatrix} \begin{Bmatrix} w_k \\ \theta_k \end{Bmatrix} + EI \begin{bmatrix} 24/L^3, & 0 \\ 0, & 8/L \end{bmatrix} \begin{Bmatrix} w_i \\ \theta_i \end{Bmatrix}$$
$$+ EI \begin{bmatrix} -12/L^3, & +6/L^2 \\ -6/L^2, & 2/L \end{bmatrix} \begin{Bmatrix} w_j \\ \theta_j \end{Bmatrix} + \begin{Bmatrix} pL/2 \\ -pL^2/12 \end{Bmatrix} = 0$$

It is of interest to compare these with the *exact* form represented by the so-called 'slope–deflection' equations which can be found in standard texts on structural analysis.

Here it will be found that the finite element approximation has achieved the exact solution at nodes for a uniform load. We show in Chapter 3 and in Appendix H reasons for this unexpected result.

2.11 Concluding remarks

The 'displacement' approach to the analysis of elastic solids is still undoubtedly the most popular and easily understood procedure. In many of the following chapters we shall use the general formulae developed here in the context of linear elastic analysis (Chapters 4, 5, and 6). These are also applicable in the context of non-linear analysis, the main variants being the definitions of the stresses, generalized strains, and other associated quantities. It is thus convenient to summarize the essential formulae, and this is done in Appendix C.

In Chapter 3 we shall show that the procedures developed here are but a particular case of finite element discretization applied to the governing equilibrium equations written in terms of displacements.[26] Clearly, alternative starting points are possible. Some of these will be mentioned in Chapters 11 and 12.

References

1. R.W. Clough. The finite element in plane stress analysis. *Proc. 2nd ASCE Conf. on Electronic Computation*. Pittsburgh, Pa., Sept. 1960.
2. R.W. Clough. The finite element method in structural mechanics. Chapter 7 of *Stress Analysis* (eds O.C. Zienkiewicz and G.S. Holister), Wiley, 1965.

3. J. Szmelter. The energy method of networks of arbitrary shape in problems of the theory of elasticity. *Proc. IUTAM Symposium on Non-Homogeneity in Elasticity and Plasticity* (ed. W. Olszak), Pergamon Press, 1959.
4. R. Courant. Variational methods for the solution of problems of equilibrium and vibration. *Bull. Am. Math. Soc.*, **49**, 1–23, 1943.
5. W. Prager and J.L. Synge. Approximation in elasticity based on the concept of function space. *Quart. Appl. Math.*, **5**, 241–69, 1947.
6. S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. 2nd ed., McGraw-Hill, 1951.
7. K. Washizu. *Variational Methods in Elasticity and Plasticity*. 2nd ed., Pergamon Press, 1975.
8. J.W. Strutt (Lord Rayleigh). On the theory of resonance. *Trans. Roy. Soc.* (*London*), **A161**, 77–118, 1870.
9. W. Ritz. Über eine neue Methode zur Lösung gewissen Variations – Probleme der mathematischen Physik. *J. Reine angew. Math.*, **135**, 1–61, 1909.
10. G.P. Bazeley, Y.K. Cheung, B.M. Irons, and O.C. Zienkiewicz. Triangular elements in bending – conforming and non-conforming solutions. *Proc. Conf. Matrix Methods in Structural Mechanics*. Air Force Inst. Tech., Wright-Patterson AF Base, Ohio, 1965.
11. S.C. Mikhlin. *The Problem of the Minimum of a Quadratic Functional*. Holden-Day, 1966.
12. M.W. Johnson and R.W. McLay. Convergence of the finite element method in the theory of elasticity. *J. Appl. Mech.. Trans. Am. Soc. Mech. Eng.*, 274–8, 1968.
13. P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.
14. T.H.H. Pian and Ping Tong. The convergence of finite element method in solving linear elastic problems. *Int. J. Solids Struct.*, **3**, 865–80, 1967.
15. E.R. de Arrantes Oliveira. Theoretical foundations of the finite element method. *Int. J. Solids Struct.*, **4**, 929–52, 1968.
16. G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. p. 106, Prentice-Hall, 1973.
17. L.F. Richardson. The approximate arithmetical solution by finite differences of physical problems. *Trans. Roy. Soc.* (*London*), **A210**, 307–57, 1910.
18. B. N. Irons and A. Razzaque. Experience with the patch test, in *Mathematical Foundations of the Finite Element Method* (ed. A.R. Aziz), pp. 557–87, Academic Press, 1972.
19. B. Fraeijs de Veubeke. Variational principles and the patch test. *Int. J. Num. Meth. Eng.*, **8**, 783–801, 1974.
20. R.L. Taylor, O.C. Zienkiewicz, J.C. Simo, and A.H.C. Chan. The patch test – a condition for assessing FEM convergence. *Int. J. Numer. Methods Engrg.*, **22**, 39–62, 1986.
21. O.C. Zienkiewicz, S. Qu, R.L. Taylor, and S. Nakazawa. The patch test for mixed formulations. *Int. J. Numer. Methods Engrg.*, **23**, 1873–83, 1986.
22. O.C. Zienkiewicz and R.L. Taylor. The finite element patch test revisited. A computer test for convergence, validation and error estimates. *Comp. Meth. Appl. Mech. and Engrg.*, **149**, 223–54, 1997.
23. B. Fraeijs de Veubeke. Displacement and equilibrium models in the finite element method. Chapter 9 of *Stress Analysis* (eds O.C. Zienkiewicz and G.S. Holister), Wiley, 1965.
24. R.L. Fox and E.L. Stanton. Developments in structural analysis by direct energy minimization. *JAIAA*. **6**, 1036–44, 1968.
25. F.K. Bogner, R.H. Mallett, M.D. Minich, and L.A. Schmit. Development and evaluation of energy search methods in non-linear structural analysis. *Proc. Conf. Matrix Methods in Structural Mechanics*. Air Force Inst. Tech., Wright-Patterson AF Base, Ohio, 1965.
26. O.C. Zienkiewicz and K. Morgan. *Finite Elements and Approximation*. Wiley, 1983.

3

Generalization of the finite element concepts. Galerkin-weighted residual and variational approaches

3.1 Introduction

We have so far dealt with one possible approach to the approximate solution of the particular problem of linear elasticity. Many other continuum problems arise in engineering and physics and usually these problems are posed by appropriate differential equations and boundary conditions to be imposed on the unknown function or functions. It is the object of this chapter to show that all such problems can be dealt with by the finite element method.

Posing the problem to be solved in its most general terms we find that we seek an unknown function $\mathbf{u}$ such that it satisfies a certain differential equation set

$$\mathbf{A}(\mathbf{u}) = \begin{Bmatrix} A_1(\mathbf{u}) \\ A_2(\mathbf{u}) \\ \vdots \end{Bmatrix} = \mathbf{0} \tag{3.1}$$

in a 'domain' (volume, area, etc.) Ω (Fig. 3.1), together with certain boundary conditions

$$\mathbf{B}(\mathbf{u}) = \begin{Bmatrix} B_1(\mathbf{u}) \\ B_2(\mathbf{u}) \\ \vdots \end{Bmatrix} = \mathbf{0} \tag{3.2}$$

on the boundaries Γ of the domain (Fig. 3.1).

The function sought may be a scalar quantity or may represent a vector of several variables. Similarly, the differential equation may be a single one or a set of simultaneous equations and does not need to be linear. It is for this reason that we have resorted to matrix notation in the above.

The finite element process, being one of approximation, will seek the solution in the approximate form

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_{i=1}^{n} \mathbf{N}_i \mathbf{a}_i = \mathbf{N}\mathbf{a} \tag{3.3}$$

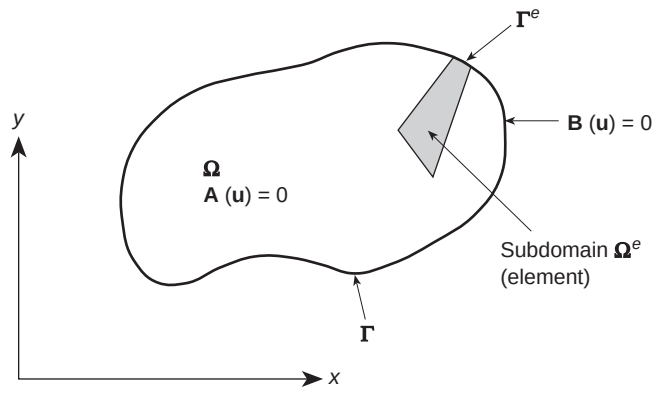


Fig. 3.1 Problem domain Ω and boundary Γ.

where $\mathbf{N}_i$ are shape functions prescribed in terms of independent variables (such as the coordinates x, y, etc.) and all or most of the parameters $\mathbf{a}_i$ are unknown.

We have seen that precisely the same form of approximation was used in the displacement approach to elasticity problems in the previous chapter. We also noted there that (*a*) the shape functions were usually defined locally for elements or subdomains and (*b*) the properties of discrete systems were recovered if the approximating eqations were cast in *an integral form* [viz. Eqs (2.22)–(2.26)].

With this object in mind we shall seek to cast the equation from which the unknown parameters $\mathbf{a}_i$ are to be obtained in the integral form

$$\int_\Omega \mathbf{G}_j(\hat{\mathbf{u}})\,\mathrm{d}\Omega + \int_\Gamma \mathbf{g}_j(\hat{\mathbf{u}})\,\mathrm{d}\Gamma = \mathbf{0} \qquad j = 1 \text{ to } n \tag{3.4}$$

in which $\mathbf{G}_j$ and $\mathbf{g}_j$ prescribe known functions or operators.

These integral forms will permit the approximation to be obtained element by element and an assembly to be achieved by the use of the procedures developed for *standard discrete systems* in Chapter 1, since, providing the functions $\mathbf{G}_j$ and $\mathbf{g}_j$ are integrable, we have

$$\int_\Omega \mathbf{G}_j\,\mathrm{d}\Omega + \int_\Gamma \mathbf{g}_j\,\mathrm{d}\Gamma = \sum_{e=1}^{m} \left(\int_{\Omega^e} \mathbf{G}_j\,\mathrm{d}\Omega + \int_{\Gamma^e} \mathbf{g}_j\,\mathrm{d}\Gamma \right) = \mathbf{0} \tag{3.5}$$

where Ω^e is the domain of each element and Γ^e its part of the boundary.

Two distinct procedures are available for obtaining the approximation in such integral forms. The first is the *method of weighted residuals* (known alternatively as the Galerkin procedure); the second is the determination of *variational functionals* for which stationarity is sought. We shall deal with both approaches in turn.

If the differential equations are linear, i.e., if we can write (3.1) and (3.2) as

$$\mathbf{A}(\mathbf{u}) \equiv \mathbf{L}\mathbf{u} + \mathbf{p} = \mathbf{0} \qquad \text{in } \Omega \tag{3.6}$$

$$\mathbf{B}(\mathbf{u}) \equiv \mathbf{M}\mathbf{u} + \mathbf{t} = \mathbf{0} \qquad \text{on } \Gamma \tag{3.7}$$

then the approximating equation system (3.4) will yield a set of linear equations of the form

$$\mathbf{Ka} + \mathbf{f} = \mathbf{0} \tag{3.8}$$

with

$$\mathbf{K}_{ij} = \sum_{e=1}^{m} \mathbf{K}_{ij}^{e} \qquad \mathbf{f}_i = \sum_{e=1}^{m} \mathbf{f}_i^{e} \tag{3.9}$$

The reader not used to abstraction may well now be confused about the meaning of the various terms. We shall introduce here some typical sets of differential equations for which we will seek solutions (and which will make the problems a little more definite).

Example 1. Steady-state heat conduction equations in a two-dimensional domain:

$$\begin{aligned} A(\phi) &= \frac{\partial}{\partial x}\left(k\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial \phi}{\partial y}\right) + Q = 0 \\ B(\phi) &= \phi - \bar{\phi} = 0 \qquad \text{on } \Gamma_\phi \\ \text{or} \quad B(\phi) &= k\frac{\partial \phi}{\partial n} + \bar{q} = 0 \qquad \text{on } \Gamma_q \end{aligned} \tag{3.10}$$

where $\mathbf{u} \equiv \phi$ indicates temperature, k is the conductivity, Q is a heat source, $\bar{\phi}$ and $\bar{q}$ are the prescribed values of temperature and heat flow on the boundaries and n is the direction normal to Γ.

In the above problem k and Q can be functions of position and, if the problem is non-linear, of ϕ or its derivatives.

Example 2. Steady-state heat conduction–convection equation in two dimensions:

$$A(\phi) = \frac{\partial}{\partial x}\left(k\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial \phi}{\partial y}\right) + u_x\frac{\partial \phi}{\partial x} + u_y\frac{\partial \phi}{\partial y} + Q = 0 \tag{3.11}$$

with boundary conditions as in the first example. Here u_x and u_y are known functions of position and represent velocities of an incompressible fluid in which heat transfer occurs.

Example 3. A system of three first order equations equivalent to Example 1:

$$\mathbf{A}(\mathbf{u}) = \begin{Bmatrix} \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} + Q \\ q_x + k\dfrac{\partial \phi}{\partial x} \\ q_y + k\dfrac{\partial \phi}{\partial y} \end{Bmatrix} = 0 \tag{3.12}$$

in Ω and

$$\begin{aligned}\mathbf{B}(\mathbf{u}) &= \phi - \bar{\phi} = 0 \qquad \text{on } \Gamma_\phi \\ &= q_n - \bar{q} = 0 \qquad \text{on } \Gamma_q\end{aligned}$$

where q_n is the flux normal to the boundary.

Here the unknown function vector $\mathbf{u}$ corresponds to the set

$$\mathbf{u} = \begin{Bmatrix} \phi \\ q_x \\ q_y \end{Bmatrix}$$

This last example is typical of a so-called *mixed formulation*. In such problems the number of dependent unknowns can always be reduced in the governing equations by suitable algebraic operations, still leaving a solvable problem [e.g., obtaining Eq. (3.10) from (3.12) by eliminating q_x and q_y].

If this cannot be done [viz. Eq. (3.10)] we have an *irreducible formulation*.

Problems of mixed form present certain complexities in their solution which we shall discuss in Chapters 11–13.

In Chapter 7 we shall return to detailed examples of the above field problems, and other examples will be introduced throughout the book. The three sets of problems will, however, be useful in their full form or reduced to one dimension (by suppressing the y variable) to illustrate the various approaches used in this chapter.

Weighted residual methods

3.2 Integral or 'weak' statements equivalent to the differential equations

As the set of differential equations (3.1) has to be zero at each point of the domain Ω, it follows that

$$\int_\Omega \mathbf{v}^{\mathrm{T}} \mathbf{A}(\mathbf{u})\, \mathrm{d}\Omega \equiv \int_\Omega [v_1 A_1(\mathbf{u}) + v_2 A_2(\mathbf{u}) + \cdots]\, \mathrm{d}\Omega \equiv 0 \tag{3.13}$$

where

$$\mathbf{v} = \begin{Bmatrix} v_1 \\ v_2 \\ \vdots \end{Bmatrix} \tag{3.14}$$

is a set of arbitrary functions equal in number to the number of equations (or components of $\mathbf{u}$) involved.

The statement is, however, more powerful. *We can assert that if* (*3.13*) *is satisfied for all* $\mathbf{v}$ *then the differential equations* (*3.1*) *must be satisfied at all points of the domain*. The proof of the validity of this statement is obvious if we consider the possibility that $\mathbf{A}(\mathbf{u}) \neq \mathbf{0}$ at any point or part of the domain. Immediately, a function $\mathbf{v}$ can be found which makes the integral of (3.13) non-zero, and hence the point is proved.

If the boundary conditions (3.12) are to be simultaneously satisfied, then we require that

$$\int_\Gamma \bar{\mathbf{v}}^{\mathrm{T}}\mathbf{B}(\mathbf{u})\,\mathrm{d}\Gamma \equiv \int_\Gamma [\bar{v}_1 B_1(\mathbf{u}) + \bar{v}_2 B_2(\mathbf{u}) + \cdots]\,\mathrm{d}\Gamma = 0 \tag{3.15}$$

for any set of functions $\bar{\mathbf{v}}$.

Indeed, the integral statement that

$$\int_\Omega \mathbf{v}^{\mathrm{T}}\mathbf{A}(\mathbf{u})\,\mathrm{d}\Omega + \int_\Gamma \bar{\mathbf{v}}^{\mathrm{T}}\mathbf{B}(\mathbf{u})\,\mathrm{d}\Gamma = 0 \tag{3.16}$$

is satisfied for all $\mathbf{v}$ and $\bar{\mathbf{v}}$ is equivalent to the satisfaction of the differential equations (3.1) and their boundary conditions (3.2).

In the above discussion it was implicitly assumed that integrals such as those in Eq. (3.16) are capable of being evaluated. This places certain restrictions on the possible families to which the functions $\mathbf{v}$ or $\mathbf{u}$ must belong. *In general we shall seek to avoid functions which result in any term in the integrals becoming infinite.*

Thus, in Eq. (3.16) we generally limit the choice of $\mathbf{v}$ and $\bar{\mathbf{v}}$ to bounded functions without restricting the validity of previous statements.

What restrictions need to be placed on the functions? The answer obviously depends on the order of differentiation implied in the equations $\mathbf{A}(\mathbf{u})$ [or $\mathbf{B}(\mathbf{u})$]. Consider, for instance, a function $\mathbf{u}$ which is continuous but has a discontinuous slope in the x-direction, as shown in Fig. 3.2 which is identical to Fig. 2.4 but is reproduced here for clarity. We imagine this discontinuity to be replaced by a continuous variation in a very small distance Δ (a process known as 'molification') and study the behaviour of the derivatives. It is easy to see that although the first derivative is not defined here, it has finite value and can be integrated easily but the second derivative tends to infinity. This therefore presents difficulties if integrals are to be evaluated numerically by simple means, even though the integral is finite. If such derivatives are multiplied by each other the integral does not exist and the function is known as *non-square integrable. Such a function is said to be C_0 continuous.*

In a similar way it is easy to see that if nth-order derivatives occur in any term of $\mathbf{A}$ or $\mathbf{B}$ then the function has to be such that its $n-1$ derivatives are continuous (C_{n-1} continuity).

On many occasions it is possible to perform an integration by parts on Eq. (3.16) and replace it by an alternative statement of the form

$$\int_\Omega \mathbf{C}(\mathbf{v})^{\mathrm{T}}\mathbf{D}(\mathbf{u})\,\mathrm{d}\Omega + \int_\Gamma \mathbf{E}(\bar{\mathbf{v}})^{\mathrm{T}}\mathbf{F}(\mathbf{u})\,\mathrm{d}\Gamma = 0 \tag{3.17}$$

In this the operators $\mathbf{C}$ to $\mathbf{F}$ usually contain lower order derivatives than those occurring in operators $\mathbf{A}$ or $\mathbf{B}$. Now a lower order of continuity is required in the choice of the $\mathbf{u}$ function at a price of higher continuity for $\mathbf{v}$ and $\bar{\mathbf{v}}$.

The statement (3.17) is now more 'permissive' than the original problem posed by Eqs (3.1), (3.2), or (3.16) and is called a *weak form* of these equations. It is a somewhat surprising fact that often this weak form is more realistic physically than the original differential equation which implied an excessive 'smoothness' of the true solution.

Integral statements of the form of (3.16) and (3.17) will form the basis of finite element approximations, and we shall discuss them later in fuller detail. Before doing so we shall apply the new formulation to an example.

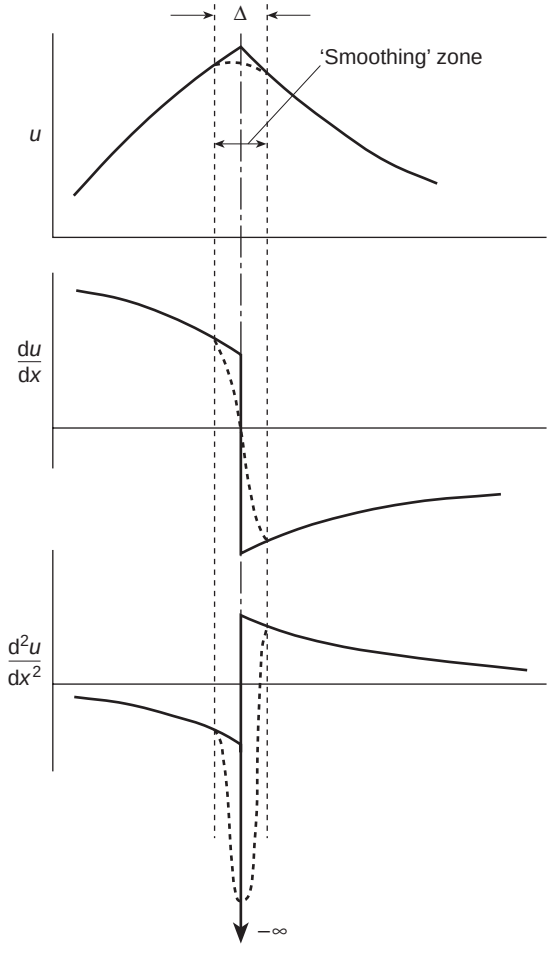


Fig. 3.2 Differentiation of function with slope discontinuity (C_0 continuous).

Example. Weak form of the heat conduction equation – forced and natural boundary conditions. Consider now the integral form of Eq. (3.10). We can write the statement (3.16) as

$$\int_\Omega v\left[\frac{\partial}{\partial x}\left(k\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial\phi}{\partial y}\right)+Q\right]\mathrm{d}x\,\mathrm{d}y+\int_{\Gamma_q}\bar{v}\left[k\frac{\partial\phi}{\partial n}+\bar{q}\right]\mathrm{d}\Gamma=0 \qquad (3.18)$$

noting that v and $\bar{v}$ are scalar functions and presuming that one of the boundary conditions, i.e.,

$$\phi-\bar{\phi}=0$$

is automatically satisfied by the choice of the functions ϕ on Γ_ϕ.

Equation (3.18) can now be integrated by parts to obtain a weak form similar to Eq. (3.17). We shall make use here of general formulae for such integration (Green's formulae) which we derive in Appendix G and which on many occasions will be

useful, i.e.

$$\begin{aligned}\int_\Omega v\frac{\partial}{\partial x}\left(k\frac{\partial\phi}{\partial x}\right)\mathrm{d}x\,\mathrm{d}y &\equiv -\int_\Omega \frac{\partial v}{\partial x}\left(k\frac{\partial\phi}{\partial x}\right)\mathrm{d}x\,\mathrm{d}y+\oint_\Gamma v\left(k\frac{\partial\phi}{\partial x}\right)n_x\,\mathrm{d}\Gamma\\ \int_\Omega v\frac{\partial}{\partial y}\left(k\frac{\partial\phi}{\partial y}\right)\mathrm{d}x\,\mathrm{d}y &\equiv -\int_\Omega \frac{\partial v}{\partial y}\left(k\frac{\partial\phi}{\partial y}\right)\mathrm{d}x\,\mathrm{d}y+\oint_\Gamma v\left(k\frac{\partial\phi}{\partial y}\right)n_y\,\mathrm{d}\Gamma\end{aligned} \tag{3.19}$$

We have thus in place of Eq. (3.18)

$$\begin{aligned}&-\int_\Omega\left(\frac{\partial v}{\partial x}k\frac{\partial\phi}{\partial x}+\frac{\partial v}{\partial y}k\frac{\partial\phi}{\partial y}-vQ\right)\mathrm{d}x\,\mathrm{d}y+\oint_\Gamma vk\left(\frac{\partial\phi}{\partial x}n_x+\frac{\partial\phi}{\partial y}n_y\right)\mathrm{d}\Gamma\\ &\quad+\int_{\Gamma_q}\bar{v}\left[k\frac{\partial\phi}{\partial n}+\bar{q}\right]\mathrm{d}\Gamma=0\end{aligned} \tag{3.20}$$

Noting that the derivative along the normal is given as

$$\frac{\partial\phi}{\partial n}\equiv\frac{\partial\phi}{\partial x}n_x+\frac{\partial\phi}{\partial y}n_y \tag{3.21}$$

and, further, making

$$\bar{v}=-v\qquad\text{on }\Gamma \tag{3.22}$$

without loss of generality (as both functions are arbitrary), we can write Eq. (3.20) as

$$\int_\Omega\boldsymbol{\nabla}^{\mathrm{T}}v\,k\boldsymbol{\nabla}\phi\,\mathrm{d}\Omega-\int_\Omega v\,Q\,\mathrm{d}\Omega-\int_{\Gamma_q}v\,\bar{q}\,\mathrm{d}\Gamma-\int_{\Gamma_\phi}v\,k\frac{\partial\phi}{\partial n}\,\mathrm{d}\Gamma=0 \tag{3.23}$$

where the operator $\boldsymbol{\nabla}$ is simply

$$\boldsymbol{\nabla}=\begin{Bmatrix}\dfrac{\partial}{\partial x}\\[2ex]\dfrac{\partial}{\partial y}\end{Bmatrix}$$

We note that

(*a*) the variable ϕ has disappeared from the integrals taken along the boundary Γ_q and that the boundary condition

$$B(\phi)=k\frac{\partial\phi}{\partial n}+\bar{q}=0$$

on that boundary is automatically satisfied – such a condition is known as a *natural boundary condition* – and

(*b*) if the choice of ϕ is restricted so as to satisfy the *forced boundary conditions* $\phi-\bar{\phi}=0$, we can omit the last term of Eq. (3.23) by restricting the choice of v to functions which give $v=0$ on Γ_ϕ.

The form of Eq. (3.23) is the *weak form* of the heat conduction statement equivalent to Eq. (3.17). It admits discontinuous conductivity coefficients k and temperature ϕ which show discontinuous first derivatives, a real possibility not admitted in the differential form.

3.3 Approximation to integral formulations: the weighted residual Galerkin method

If the unknown function $\mathbf{u}$ is approximated by the expansion (3.3), i.e.,

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_{i=1}^{n} \mathbf{N}_i \mathbf{a}_i = \mathbf{N}\mathbf{a}$$

then it is clearly impossible to satisfy both the differential equation and the boundary conditions in the general case. The integral statements (3.16) or (3.17) allow an approximation to be made if, in place of *any function* $\mathbf{v}$, we put a finite set of approximate functions

$$\mathbf{v} = \sum_{j=1}^{n} \mathbf{w}_j \, \delta\mathbf{a}_j \qquad \bar{\mathbf{v}} = \sum_{j=1}^{n} \bar{\mathbf{w}}_j \, \delta\mathbf{a}_j \tag{3.24}$$

in which $\delta\mathbf{a}_j$ are arbitrary parameters and n is the number of unknowns entering the problem.

Inserting the above approximations into Eq. (3.16) we have

$$\delta\mathbf{a}_j^{\mathrm{T}} \left[\int_{\Omega} \mathbf{w}_j^{\mathrm{T}} \mathbf{A}(\mathbf{N}\mathbf{a}) \, \mathrm{d}\Omega + \int_{\Gamma} \bar{\mathbf{w}}_j^{\mathrm{T}} \mathbf{B}(\mathbf{N}\mathbf{a}) \, \mathrm{d}\Gamma \right] = 0$$

and since $\delta\mathbf{a}_j$ is arbitrary we have a set of equations which is sufficient to determine the parameters $\mathbf{a}_j$ as

$$\int_{\Omega} \mathbf{w}_j^{\mathrm{T}} \mathbf{A}(\mathbf{N}\mathbf{a}) \, \mathrm{d}\Omega + \int_{\Gamma} \bar{\mathbf{w}}_j^{\mathrm{T}} \mathbf{B}(\mathbf{N}\mathbf{a}) \, \mathrm{d}\Gamma = 0 \qquad j = 1 \text{ to } n \tag{3.25}$$

or, from Eq. (3.17),

$$\int_{\Omega} \mathbf{C}(\mathbf{w}_j)^{\mathrm{T}} \mathbf{D}(\mathbf{N}\mathbf{a}) \, \mathrm{d}\Omega + \int_{\Gamma} \mathbf{E}(\bar{\mathbf{w}}_j)^{\mathrm{T}} \mathbf{F}(\mathbf{N}\mathbf{a}) \, \mathrm{d}\Gamma = 0 \qquad j = 1 \text{ to } n \tag{3.26}$$

If we note that $\mathbf{A}(\mathbf{N}\mathbf{a})$ represents the *residual or error* obtained by substitution of the approximation into the differential equation [and $\mathbf{B}(\mathbf{N}\mathbf{a})$, the residual of the boundary conditions], then Eq. (3.25) is a *weighted integral of such residuals*. The approximation may thus be called the *method of weighted residuals*.

In its classical sense it was first described by Crandall,[1] who points out the various forms used since the end of the last century. More recently a very full exposé of the method has been given by Finlayson.[2] Clearly, almost any set of independent functions $\mathbf{w}_j$ could be used for the purpose of weighting and, according to the choice of function, a different name can be attached to each process. Thus the various common choices are:

1. *Point collocation*.[3] $\mathbf{w}_j = \boldsymbol{\delta}_j$, where $\boldsymbol{\delta}_j$ is such that for $x \neq x_j$; $y \neq y_j$, $\mathbf{w}_j = 0$ but $\int_{\Omega} \mathbf{w}_j \, \mathrm{d}\Omega = \mathbf{I}$ (unit matrix). This procedure is equivalent to simply making the residual zero at n points within the domain and integration is 'nominal' (incidentally although $\mathbf{w}_j$ defined here does not satisfy all the criteria of Sec. 3.2, it is nevertheless admissible in view of its properties).
2. *Subdomain collocation*.[4] $\mathbf{w}_j = \mathbf{I}$ in Ω_j and zero elsewhere. This essentially makes the integral of the error zero over the specified subdomains.

3. *The Galerkin method* (Bubnov–Galerkin).[5,6] $\mathbf{w}_j = \mathbf{N}_j$. Here simply the original shape (or basis) functions are used as weighting. This method, as we shall see, frequently (but by no means always) leads to symmetric matrices and for this and other reasons will be adopted in our finite element work almost exclusively.

The name of 'weighted residuals' is clearly much older than that of the 'finite element method'. The latter uses mainly locally based (element) functions in the expansion of Eq. (3.3) but the general procedures are identical. As the process always leads to equations which, being of integral form, can be obtained by summation of contributions from various subdomains, we choose to embrace all weighted residual approximations under the name of *generalized finite element method*. Frequently, simultaneous use of both local and 'global' trial functions will be found to be useful.

In the literature the names of Petrov and Galerkin[5] are often associated with the use of weighting functions such that $\mathbf{w}_j \neq \mathbf{N}_j$. It is important to remark that the well-known *finite difference method* of approximation is a particular case of collocation with locally defined basis functions and is thus a case of a Petrov–Galerkin scheme. We shall return to such unorthodox definitions in more detail in Chapter 16.

To illustrate the procedure of weighted residual approximation and its relation to the finite element process let us consider some specific examples.

Example 1. One-dimensional equation of heat conduction (Fig. 3.3)**.** The problem here will be a one-dimensional representation of the heat conduction equation [Eq. (3.10)] with unit conductivity. (This problem could equally well represent many other physical situations, e.g., deformation of a loaded string.) Here we have

$$A(\phi) = \frac{\mathrm{d}^2\phi}{\mathrm{d}x^2} + Q = 0 \quad (0 \leqslant x \leqslant L) \tag{3.27}$$

with $Q = Q(x)$ given by $Q = 1$ $(0 \leqslant x < L/2)$ and $Q = 0$ $(L/2 \leqslant x \leqslant L)$. The boundary conditions assumed will be simply $\phi = 0$ at $x = 0$ and $x = L$.

In the first case we shall consider a one- or two-term approximation of the Fourier series form, i.e.,

$$\phi \approx \hat{\phi} = \sum a_i \sin\frac{\pi x_i}{L} \qquad N_i = \sin\frac{\pi x_i}{L} \tag{3.28}$$

with $i = 1$ and $i = 1$ and 2. These satisfy the boundary conditions exactly and are continuous throughout the domain. We can thus use either Eq. (3.16) or Eq. (3.17) for the approximation with equal validity. We shall use the former, which allows various weighting functions to be adopted. In Fig. 3.3 we present the problem and its solution using point collocation, subdomain collocation, and the Galerkin method.†

As the chosen expansion satisfies *a priori* the boundary conditions there is no need to introduce them into the formulation, which is given simply by

$$\int_0^L w_j \left[\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(\sum N_i a_i\right) + Q\right] \mathrm{d}x = 0 \tag{3.29}$$

The full working out of this problem is left as an exercise to the reader.

† In the case of point collocation using $i = 1$ $(x_i = L/2)$ a difficulty arises about the value of Q (as this is either zero or one). The value of $\frac{1}{2}$ was therefore used for the example.

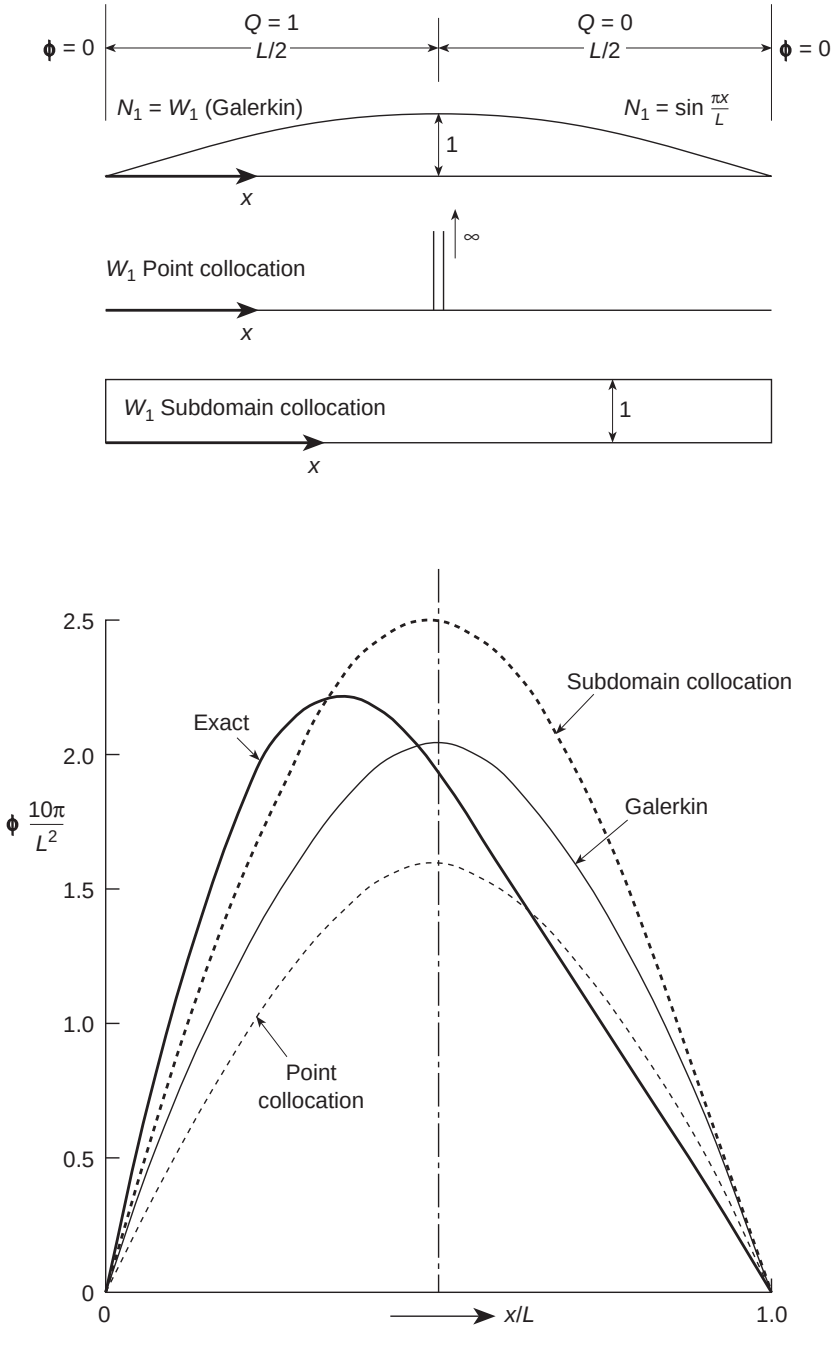
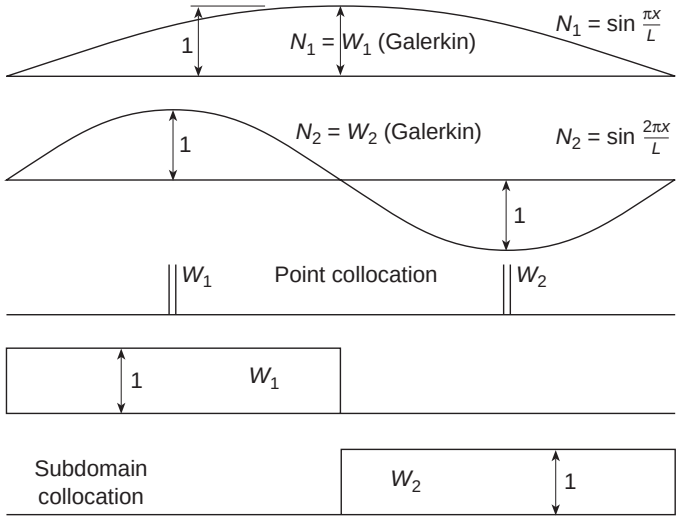


Fig. 3.3 One-dimensional heat conduction. (a) One-term solution using different weighting procedures.



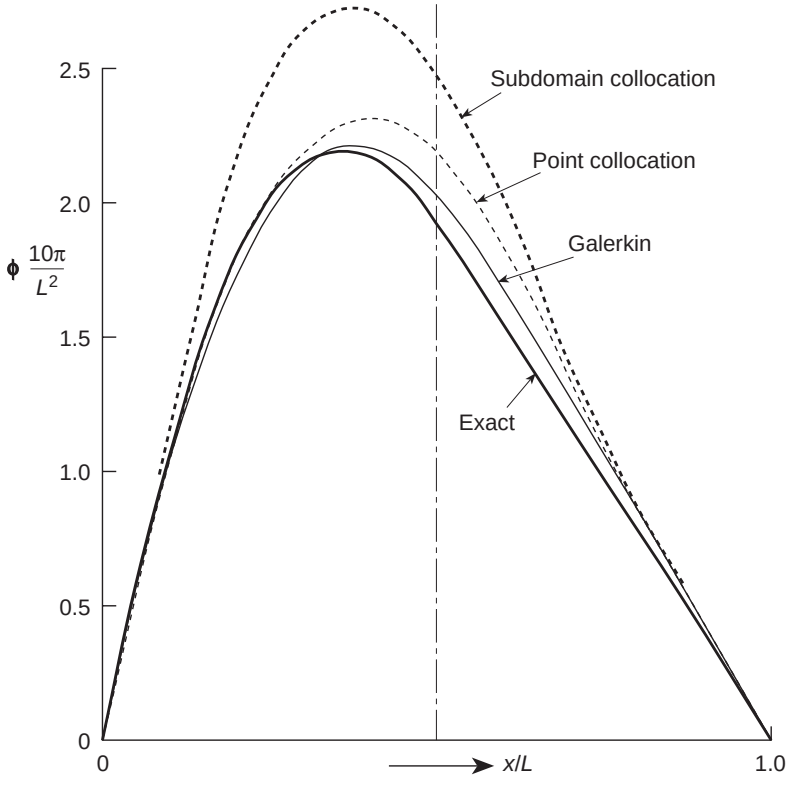


Fig. 3.3 (*cont.*) (b) Two-term solutions using different weighting procedures.

Of more interest to the standard finite element field is the use of piecewise defined (locally based) functions in place of the global functions of Eq. (3.28). Here, to avoid imposing slope continuity, we shall use the equivalent of Eq. (3.17) obtained by integrating Eq. (3.29) by parts. This yields

$$\int_0^L \left[\frac{\mathrm{d}w_j}{\mathrm{d}x} \left(\sum_i \frac{\mathrm{d}N_i}{\mathrm{d}x} a_i \right) - w_j Q \right] \mathrm{d}x = 0 \tag{3.30}$$

The boundary terms disappear identically if $w_j = 0$ at the two ends.

The above equations can be written as

$$\mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{3.31}$$

where for each 'element' of length L^e,

$$\begin{aligned} K_{ji}^e &= \int_0^{L^e} \frac{\mathrm{d}w_j}{\mathrm{d}x} \frac{\mathrm{d}N_i}{\mathrm{d}x} \mathrm{d}x \\ f_j^e &= -\int_0^{L^e} w_j Q \, \mathrm{d}x \end{aligned} \tag{3.32}$$

with the usual rules of addition pertaining, i.e.,

$$K_{ji} = \sum_e K_{ji}^e \qquad f_j = \sum_e f_j^e \tag{3.33}$$

In the computation we shall use the Galerkin procedure, i.e. $w_j = N_j$, and the reader will observe that the matrix $\mathbf{K}$ is then symmetric, i.e., $K_{ij} = K_{ji}$.

As the shape functions need only be of C_0 continuity, a piecewise linear approximation is conveniently used, as shown in Fig. 3.4. Considering a typical element *ij* shown, we can write (moving the origin of x to point i)

$$N_j = \frac{x}{L^e} \qquad N_i = \frac{L^e - x}{L^e} \tag{3.34}$$

giving, for a typical element,

$$\begin{aligned} K_{ii} &= K_{jj} = \frac{1}{L^e} = -K_{ji} = -K_{ij} \\ f_j^e &= -Q^e L^e/2 = f_i^e \end{aligned} \tag{3.35}$$

where Q^e is the value for element e.

Assembly of a typical equation at a node i is left to the reader, who is well advised to carry out the calculations leading to the results shown in Fig. 3.4 for a two- and four-element subdivision.

Some points of interest immediately arise if the results of Figs 3.3 and 3.4 are compared. With smooth global shape functions the Galerkin method gives better overall results than those achieved for the same number of unknown parameters **a** with locally based functions. This we shall find to be the general case with higher order approximations, yielding better accuracy. Further, it will be observed that the linear approximation has given the exact answers at the interelement nodal points. This is a property of the particular equation being solved and unfortunately does not carry over to general problems.[7] (See also Appendix H.) Lastly, the

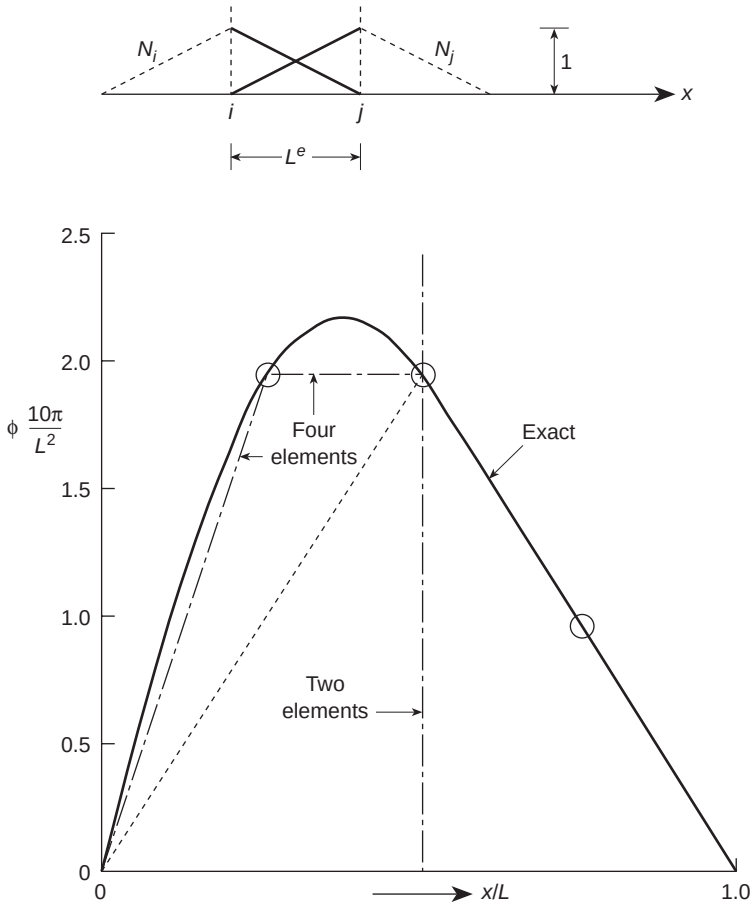


Fig. 3.4 Galerkin finite element solution of problem of Fig. 3.3 using linear locally based shaped functions.

reader will observe how easy it is to create equations with any degree of subdivision once the element properties [Eq. (3.35)] have been derived. This is not the case with global approximation where new integrations have to be carried out for each new parameter introduced. It is this repeatability feature that is one of the advantages of the finite element method.

Example 2. Steady-state heat conduction–convection in two dimensions. The Galerkin formulation. We have already introduced the problem in Sec. 3.1 and defined it by Eq. (3.11) with appropriate boundary conditions. The equation differs only in the convective terms from that of simple heat conduction for which the weak form has already been obtained in Eq. (3.23). We can write the weighted residual equation immediately from this, substituting $v = w_j \delta a_j$ and adding the convective terms. Thus we have

$$\int_\Omega \nabla^{\mathrm{T}} w_j k \nabla \hat{\phi}\, \mathrm{d}\Omega - \int_\Omega w_j \left(u_x \frac{\partial \hat{\phi}}{\partial x} + u_y \frac{\partial \hat{\phi}}{\partial y} \right) \mathrm{d}\Omega - \int_\Omega w_j Q\, \mathrm{d}\Omega - \int_{\Gamma_q} w_j \bar{q}\, \mathrm{d}\Gamma = 0 \quad (3.36)$$

with $\hat{\phi} = \sum N_i a_i$ being such that the prescribed values of $\bar{\phi}$ are given on the boundary Γ_ϕ and that $\delta a_j = 0$ on that boundary (ignoring that term in (3.36)).

Specializing to the Galerkin approximation, i.e., putting $w_j = N_j$, we have immediately a set of equations of the form

$$\mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{3.37}$$

with

$$\begin{aligned} K_{ji} &= \int_\Omega \boldsymbol{\nabla}^{\mathrm{T}} N_j k \boldsymbol{\nabla} N_i \, \mathrm{d}\Omega - \int_\Omega \left(N_j u_x \frac{\partial N_i}{\partial x} + N_j u_y \frac{\partial N_i}{\partial y} \right) \mathrm{d}\Omega \\ &= \int_\Omega \left(\frac{\partial N_j}{\partial x} k \frac{\partial N_i}{\partial x} + \frac{\partial N_j}{\partial y} k \frac{\partial N_i}{\partial y} \right) \mathrm{d}\Omega \\ &\quad - \int_\Omega \left(N_j u_x \frac{\partial N_i}{\partial x} + N_j u_y \frac{\partial N_i}{\partial y} \right) \mathrm{d}\Omega \end{aligned} \tag{3.38a}$$

$$f_j = -\int_\Omega N_j Q \, \mathrm{d}\Omega - \int_{\Gamma_q} N_j \bar{q} \, \mathrm{d}\Gamma \tag{3.38b}$$

Once again the components K_{ji} and f_j can be evaluated for a typical element or subdomain and systems of equations built up by standard methods.

At this point it is important to mention that to satisfy the boundary conditions some of the parameters $\mathbf{a}_i$ have to be prescribed and the number of approximation equations must be equal to the number of unknown parameters. It is nevertheless often convenient to form all equations for all parameters and prescribe the fixed values at the end using precisely the same techniques as we have described in Chapter 1 for the insertion of prescribed boundary conditions in standard discrete problems.

A further point concerning the coefficients of the matrix $\mathbf{K}$ should be noted here. The first part, corresponding to the pure heat conduction equation, is symmetric ($K_{ij} = K_{ji}$) but the second is not and thus a system of non-symmetric equations needs to be solved. There is a basic reason for such non-symmetries which will be discussed in Sec. 3.9.

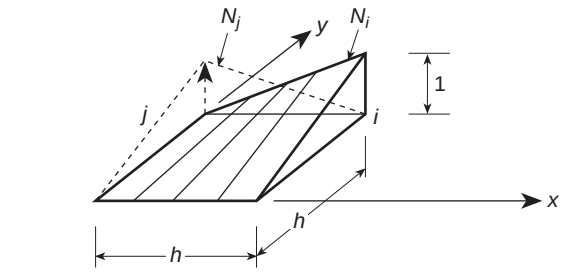
To make the problem concrete consider the domain Ω to be divided into regular square elements of side h (Fig. 3.5). To preserve C_0 continuity with nodes placed at corners, shape functions given as the product of the linear expansions can be written. For instance, for node i, as shown in Fig. 3.5,

$$N_i = \frac{x}{h}\frac{y}{h}$$

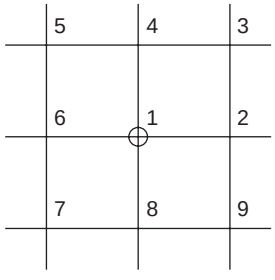
and for node j,

$$N_j = \frac{(h-x)}{h}\frac{y}{h}, \qquad \text{etc.}$$

With these shape functions the reader is invited to evaluate typical element contributions and to assemble the equations for point 1 of the mesh numbered as



(a)



(b)

Fig. 3.5 A linear square element of C_0 continuity. (a) Shape functions for a square element. (b) 'Connected' equation for node 1.

shown in Fig. 3.5. The result will be (if no boundary of type Γ_q is present and Q is assumed to be constant)

$$\begin{aligned}\frac{8}{3}a_1 &- \left(\frac{1}{3}-\frac{u_x h}{3k}-\frac{u_y h}{6k}\right)a_2 - \left(\frac{1}{3}-\frac{u_x h}{12k}-\frac{u_y h}{12k}\right)a_3 - \left(\frac{1}{3}-\frac{u_x h}{6k}-\frac{u_y h}{3k}\right)a_4 \\ &- \left(\frac{1}{3}+\frac{u_x h}{12k}-\frac{u_y h}{12k}\right)a_5 - \left(\frac{1}{3}+\frac{u_x h}{3k}-\frac{u_y h}{6k}\right)a_6 - \left(\frac{1}{3}+\frac{u_x h}{12k}+\frac{u_y h}{12k}\right)a_7 \\ &- \left(\frac{1}{3}-\frac{u_x h}{6k}+\frac{u_y h}{3k}\right)a_8 - \left(\frac{1}{3}+\frac{u_x h}{12k}+\frac{u_y h}{12k}\right)a_9 = 4h^2 Q \end{aligned} \tag{3.39}$$

This equation is similar to those that would be obtained by using finite difference approximations to the same equations in a fairly standard manner.[8,9] In the example discussed some difficulties arise when the convective terms are large. In such cases the Galerkin weighting is not acceptable and other forms have to be used. This is discussed in detail in Chapter 2 of the third volume.

3.4 Virtual work as the 'weak form' of equilibrium equations for analysis of solids or fluids

In Chapter 2 we introduced a finite element by way of an application to the solid mechanics problem of linear elasticity. The integral statement necessary for

formulation in terms of the finite element approximation was supplied via the principle of *virtual work*, which was assumed to be so basic as not to merit proof. Indeed, to many this is so, and the virtual work principle is considered as a statement of mechanics more fundamental than the traditional equilibrium conditions of Newton's laws of motion. Others will argue with this view and will point out that all work statements are derived from the classical laws pertaining to the equilibrium of the particle. We shall therefore show in this section that the virtual work statement is simply a 'weak form' of equilibrium equations.

In a general three-dimensional continuum the equilibrium equations of an elementary volume can be written in terms of the components of the symmetric cartesian stress tensor as[10]

$$\begin{Bmatrix} A_1 \\ A_2 \\ A_3 \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial \sigma_x}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + \dfrac{\partial \tau_{xz}}{\partial z} \\ \dfrac{\partial \sigma_y}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + \dfrac{\partial \tau_{yz}}{\partial z} \\ \dfrac{\partial \sigma_z}{\partial z} + \dfrac{\partial \tau_{xz}}{\partial x} + \dfrac{\partial \tau_{yz}}{\partial y} \end{Bmatrix} + \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} = 0 \tag{3.40}$$

where $\mathbf{b}^{\mathrm{T}} = [b_x, b_y, b_z]$ stands for the body forces acting per unit volume (which may well include acceleration effects).

In solid mechanics the six stress components will be some general functions of the components of the displacement

$$\mathbf{u} = [u, v, w]^{\mathrm{T}} \tag{3.41}$$

and in fluid mechanics of the velocity vector $\mathbf{u}$, which has identical components. Thus Eq. (3.40) can be considered as a general equation of the form Eq. (3.1), i.e., $\mathbf{A}(\mathbf{u}) = \mathbf{0}$. To obtain a weak form we shall proceed as before, introducing an arbitrary weighting function vector, defined as

$$\mathbf{v} \equiv \delta\mathbf{u} = [\delta u, \delta v, \delta w]^{\mathrm{T}} \tag{3.42}$$

We can now write the integral statement of Eq. (3.13) as

$$\int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{A}(\mathbf{u})\,\mathrm{d}V = \int_\Omega \left[\delta u\left(\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + b_x\right) + \delta v(A_2) + \delta w(A_3)\right]\mathrm{d}\Omega \tag{3.43}$$

where V, the volume, is the problem domain.

Integrating each term by parts and rearranging we can write this as

$$\begin{aligned} &-\int_\Omega \left[\sigma_x \frac{\partial}{\partial x}(\delta u) + \tau_{xy}\left(\frac{\partial}{\partial y}(\delta u) + \frac{\partial}{\partial x}(\delta v)\right) + \cdots - \delta u b_x - \delta v b_y - \delta w b_z\right]\mathrm{d}\Omega \\ &\quad + \int_\Gamma [\delta u(\sigma_x n_x + \tau_{xy} n_y + \tau_{xz} n_z) + \delta v(\cdots) + \delta w(\cdots)]\,\mathrm{d}\Gamma = 0 \end{aligned} \tag{3.44}$$

where Γ is the surface area of the solid (here again Green's formulae of Appendix G are used).

In the first set of bracketed terms we can recognize immediately the small strain operators acting on $\delta\mathbf{u}$, which can be termed a virtual displacement (or virtual velocity). We can therefore introduce a virtual strain (or strain rate) defined as

$$\delta\boldsymbol{\varepsilon} = \begin{Bmatrix} \dfrac{\partial}{\partial x}(\delta u) \\ \dfrac{\partial}{\partial y}(\delta v) \\ \dfrac{\partial}{\partial z}(\delta w) \\ \vdots \end{Bmatrix} = \mathbf{S}\,\delta\mathbf{u} \tag{3.45}$$

where the strain operators are defined as in Chapter 2 [Eqs (2.2)–(2.4)].

Similarly, the terms in the second integral will be recognized as forces $\mathbf{t}$:

$$\mathbf{t} = [t_x, t_y, t_z]^{\mathrm{T}} \tag{3.46}$$

acting per unit area of the surface A. Arranging the six stress components in a vector $\boldsymbol{\sigma}$ and similarly the six virtual strain (or rate of virtual strain) components in a vector $\delta\boldsymbol{\varepsilon}$, we can write Eq. (3.44) simply as

$$\int_\Omega \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_\Gamma \delta\mathbf{u}^{\mathrm{T}}\mathbf{t}\,\mathrm{d}\Gamma = 0 \tag{3.47}$$

which is the three dimensional equivalent virtual work statement used in Eqs (2.10) and (2.22) of Chapter 2.

We see from the above that the virtual work statement is precisely the weak form of the equilibrium equations and is valid for non-linear as well as linear stress–strain (*or stress–rate of strain*) *relations*.

The finite element approximation which we have derived in Chapter 2 *is in fact a Galerkin formulation of the weighted residual process applied to the equilibrium equation*. Thus, if we take $\delta\mathbf{u}$ as the shape function times arbitrary parameters

$$\delta\mathbf{u} = \mathbf{N}\,\delta\mathbf{a} \tag{3.48}$$

where the displacement field is discretized, i.e.,

$$\mathbf{u} = \sum \mathbf{N}_i\mathbf{a}_i \tag{3.49}$$

together with the constitutive relation of Eq. (2.5), we shall determine once again all the basic expressions of Chapter 2 which are so essential to the solution of elasticity problems.

Similar expressions are vital to the formulation of equivalent fluid mechanics problems as discussed further in the third volume.

3.5 Partial discretization

In the approximation to the problem of solving the differential equation (3.1) by an expression of the standard form of Eq. (3.3), we have assumed that the shape functions $\mathbf{N}$ included in them are *all* independent coordinates of the problem

and that **a** was simply a set of constants. The final approximation equations were thus always of an algebraic form, from which a unique set of parameters could be determined.

In some problems it is convenient to proceed differently. Thus, for instance, if the independent variables are x, y and z we could allow the parameters **a** to be functions of z and do the approximate expansion only in the domain of x, y, say $\bar{\Omega}$. Thus, in place of Eq. (3.3) we would have

$$\begin{aligned} \mathbf{u} &= \mathbf{N}\mathbf{a} \\ \mathbf{N} &= \mathbf{N}(x, y) \\ \mathbf{a} &= \mathbf{a}(z) \end{aligned} \tag{3.50}$$

Clearly the derivatives of **a** with respect to z will remain in the final discretization and the result will be a set of *ordinary differential equations* with z as the independent variable. In linear problems such a set will have the appearance

$$\mathbf{K}\mathbf{a} + \mathbf{C}\dot{\mathbf{a}} + \cdots + \mathbf{f} = \mathbf{0} \tag{3.51}$$

where $\dot{\mathbf{a}} \equiv \mathrm{d}\mathbf{a}/\mathrm{d}z$, etc.

Such a partial discretization can obviously be used in different ways, but is particularly useful when the domain $\bar{\Omega}$ is not dependent on z, i.e., when the *problem is prismatic*. In such a case the coefficient matrices of the ordinary differential equations, (3.51), are independent of z and the solution of the system can frequently be carried out efficiently by standard analytical methods.

This type of partial discretization has been applied extensively by Kantorovitch[11] and is frequently known by his name. In the second volume we shall discuss such semi-analytical treatments in the context of prismatic solids where the final solution is obtained in terms of Fourier series. However, the most frequently encountered 'prismatic' problem is one involving the time variable, where the space domain $\bar{\Omega}$ is not subject to change. We shall address such problems in Chapter 17 of this volume. It is convenient by way of illustration to consider here heat conduction in a two-dimensional equation in its transient state. This is obtained from Eq. (3.10) by addition of the heat storage term $c(\partial\phi/\partial t)$, where c is the specific heat per unit volume. We now have a problem posed in a domain $\Omega(x, y, t)$ in which the following equation holds:

$$A(\phi) \equiv \frac{\partial}{\partial x}\left(k\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial\phi}{\partial y}\right) + Q - c\frac{\partial\phi}{\partial t} = 0 \tag{3.52}$$

with boundary conditions identical to those of Eq. (3.10) and the temperature is zero at time zero. Taking

$$\phi \approx \hat{\phi} = \sum N_i a_i \tag{3.53}$$

with $a_i = a_i(t)$ and $N_i = N_i(x, y)$ and using the Galerkin weighting procedure we follow precisely the steps outlined in Eqs (3.36)–(3.38) and arrive at a system of ordinary differential equations

$$\mathbf{K}\mathbf{a} + \mathbf{C}\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} + \mathbf{f} = \mathbf{0} \tag{3.54}$$

Here the expression for K_{ij} is identical with that of Eq. (3.38a) (convective terms neglected), f_i identical to Eq. (3.38b), and the reader can verify that the matrix $\mathbf{C}$ is defined by

$$C_{ij} = \int_{\Omega} N_i c N_j \, \mathrm{d}x \, \mathrm{d}y \tag{3.55}$$

Once again the matrix $\mathbf{C}$ can be assembled from its element contributions. Various analytical and numerical procedures can be applied simply to the solution of such transient, ordinary, differential equations which, again, we shall discuss in detail in Chapters 17 and 18. However, to illustrate the detail and the possible advantage of the process of partial discretization, we shall consider a very simple problem.

Example. Consider a square prism of size L in which the transient heat conduction equation (3.52) applies and assume that the rate of heat generation varies with time as

$$Q = Q_0 \, \mathrm{e}^{-\alpha t} \tag{3.56}$$

(this approximates a problem of heat development due to hydration of concrete). We assume that at $t = 0$, $\phi = 0$ throughout. Further, we shall take $\phi = 0$ on all boundaries throughout all times.

As a first approximation a shape function for a one-parameter solution is taken:

$$\begin{aligned} \phi &= N_1 a_1 \\ N_1 &= \cos\frac{\pi x}{L} \cos\frac{\pi y}{L} \end{aligned} \tag{3.57}$$

with x and y measured from the centre (Fig. 3.6). Evaluating the coefficients, we have

$$\begin{aligned} K_{11} &= \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} \left[k\left(\frac{\partial N_1}{\partial x}\right)^2 + k\left(\frac{\partial N_1}{\partial y}\right)^2 \right] \mathrm{d}x \, \mathrm{d}y = \frac{\pi^2 k}{2} \\ C_{11} &= \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} c N_1^2 \, \mathrm{d}x \, \mathrm{d}y = \frac{L^2 c}{4} \\ f_1 &= \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} N_1 Q_0 \, \mathrm{e}^{-\alpha t} \, \mathrm{d}x \, \mathrm{d}y = \frac{4 Q_0 L^2}{\pi^2} \, \mathrm{e}^{-\alpha t} \end{aligned} \tag{3.58}$$

Thus leads to an ordinary differential equation with one parameter a_1:

$$C_{11} \frac{\mathrm{d}a_1}{\mathrm{d}t} + K_{11} a_1 + f_1 = 0 \tag{3.59}$$

with $a_1 = 0$ when $t = 0$. The exact solution of this is easy to obtain, as is shown in Fig. 3.6 for specific values of the parameters α and $k/L^2 c$.

On the same figure we show a two-parameter solution with

$$N_2 = \cos\frac{3\pi x}{L} \cos\frac{3\pi y}{L} \tag{3.60}$$

which readers can pursue to test their grasp of the problem. The second component of the Fourier series is here omitted due to the required symmetry of solution.

The remarkable accuracy of the one-term approximation in this example should be noted.

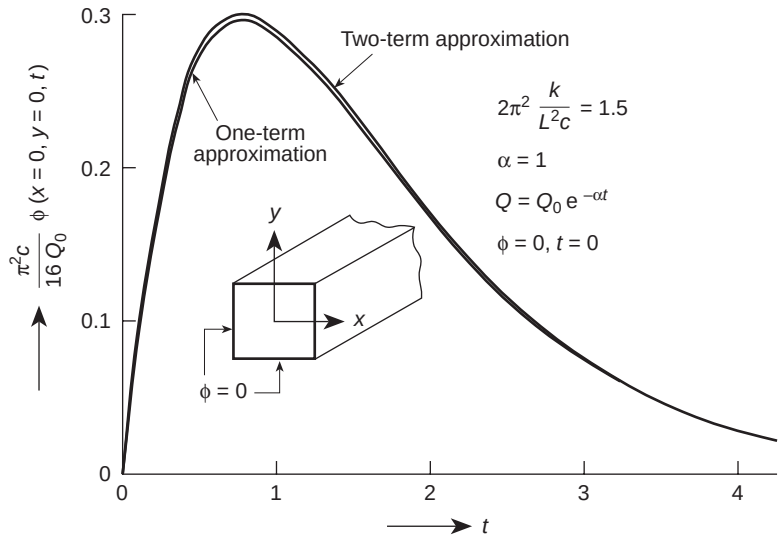


Fig. 3.6 Two-dimensional transient heat development in a square prism: plot of temperature at centre.

3.6 Convergence

In the previous sections we have discussed how approximate solutions can be obtained by use of an expansion of the unknown function in terms of trial or shape functions. Further, we have stated the necessary conditions that such functions have to fulfil in order that the various integrals can be evaluated over the domain. Thus if various integrals contain only the values of N or its first derivatives then N has to be C_0 continuous. If second derivatives are involved, C_1 continuity is needed, etc. The problem to which we have not yet addressed ourselves consists of the questions of *just how good the approximation is* and *how it can be systematically improved to approach the exact answer*. The first question is more difficult to answer and presumes knowledge of the exact solution (see Chapter 14). The second is more rational and can be answered if we consider some systematic way in which the number of parameters $\mathbf{a}$ in the standard expansion of Eq. (3.3),

$$\hat{\mathbf{u}} = \sum_{1}^{n} \mathbf{N}_i \mathbf{a}_i$$

is presumed to increase.

In some of the examples we have assumed, in effect, a trigonometric Fourier-type series limited to a finite number of terms with a single form of trial function assumed over the whole domain. Here addition of new terms would be simply an extension of the number of terms in the series included in the analysis, and as the Fourier series is known to be able to represent any function within any accuracy desired as the number of terms increases, we can talk about *convergence* of the approximation to the true solution as the number of terms increases.

In other examples of this chapter we have used locally based functions which are fundamental in the finite element analysis. Here we have tacitly assumed that *convergence occurs as the size of elements decreases and, hence, the number of* **a** *parameters specified at nodes increases*. It is with such convergence that we need to be concerned and we have already discussed this in the context of the analysis of elastic solids in Chapter 2 (Sec. 2.6).

We have now to determine

(*a*) that, as the number of elements increases, the unknown functions can be approximated as closely as required, and
(*b*) how the error decreases with the size, h, of the element subdivisions (h is here some typical dimension of an element).

The first problem is that of *completeness* of the expansion and we shall here assume that all trial functions are polynomials (or at least include certain terms of a polynomial expansion).

Clearly, as the approximation discussed here is to the weak, integral form typified by Eqs (3.13) or (3.17) it is necessary that every term occurring under the integral be in the limit capable of being approximated as nearly as possible and, in particular, giving a single constant value over an infinitesimal part of the domain Ω.

If a derivative of order m exists in any such term, then it is obviously necessary for the local polynomial to be at least of the order m so that, in the limit, such a constant value can be obtained.

We will thus state that a necessary condition for the expansion to be covergent is the *criterion of completeness*: that a constant value of the mth derivative be attainable in the element domain (if mth derivatives occur in the integral form) when the size of any element tends to zero.

This criterion is automatically ensured if the polynomials used in the shape function N are complete to mth order. This criterion is also equivalent to the one of constant strain postulated in Chapter 2 (Sec. 2.5). This, however, has to be satisfied only in the limit $h \to 0$.

If the actual order of a complete polynomial used in the finite element expansion is $p \geqslant m$, then *the order of convergence* can be ascertained by seeing how closely such a polynomial can follow the local Taylor expansion of the unknown **u**. Clearly the order of error will be simply $O(h^{p+1})$ since only terms of order p can be rendered correctly.

Knowledge of the order of convergence helps in ascertaining how good the approximation is if studies on several decreasing mesh sizes are conducted. Though, in Chapter 15, we shall see this asymptotic convergence rate is seldom reached if singularities occur in the problem. Once again we have reestablished some of the conditions discussed in Chapter 2.

We shall not discuss, at this stage, approximations which do not satisfy the postulated continuity requirements except to remark that once again, in many cases, convergence and indeed improved results can be obtained (see Chapter 10).

In the above we have referred to the convergence of a given element type as its size is reduced. This is sometimes referred to as *h convergence*.

On the other hand, it is possible to consider a subdivision into elements of a given size and to obtain convergence to the exact solution by increasing the polynomial order p of each element. This is referred to as *p convergence*, which is obviously

assured. In general *p* convergence is more rapid per degree of freedom introduced. We shall discuss both types further in Chapter 15.

Variational principles

3.7 What are 'variational principles'?

What are variational principles and how can they be useful in the approximation to continuum problems? It is to these questions that the following sections are addressed.

First a definition: a 'variational principle' specifies a scalar quantity (functional) Π, which is defined by an integral form

$$\Pi = \int_{\Omega} F\left(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial x}, \dots\right) \mathrm{d}\Omega + \int_{\Gamma} E\left(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial x}, \dots\right) \mathrm{d}\Gamma \tag{3.61}$$

in which $\mathbf{u}$ is the unknown function and F and E are specified differential operators. The solution to the continuum problem is a function $\mathbf{u}$ which makes Π *stationary* with respect to arbitrary changes $\delta\mathbf{u}$. Thus, for a solution to the continuum problem, the 'variation' is

$$\delta\Pi = 0 \tag{3.62}$$

for any $\delta\mathbf{u}$, which defines the condition of stationarity.[12]

If a 'variational principle' can be found, then means are immediately established for obtaining approximate solutions in the standard, integral form suitable for finite element analysis.

Assuming a trial function expansion in the usual form [Eq. (3.3)]

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_{1}^{n} \mathbf{N}_i \mathbf{a}_i$$

we can insert this into Eq. (3.61) and write

$$\delta\Pi = \frac{\partial \Pi}{\partial \mathbf{a}_1}\delta\mathbf{a}_1 + \frac{\partial \Pi}{\partial \mathbf{a}_2}\delta\mathbf{a}_2 + \cdots + \frac{\partial \Pi}{\partial \mathbf{a}_n}\delta\mathbf{a}_n = 0 \tag{3.63}$$

This being true for any variations $\delta\mathbf{a}$ yields a set of equations

$$\frac{\partial \Pi}{\partial \mathbf{a}} = \begin{Bmatrix} \dfrac{\partial \Pi}{\partial \mathbf{a}_1} \\ \vdots \\ \dfrac{\partial \Pi}{\partial \mathbf{a}_n} \end{Bmatrix} = \mathbf{0} \tag{3.64}$$

from which parameters $\mathbf{a}_i$ are found. The equations are of an integral form necessary for the finite element approximation as the original specification of Π was given in terms of domain and boundary integrals.

The process of finding stationarity with respect to trial function parameters $\mathbf{a}$ is an old one and is associated with the names of Rayleigh[13] and Ritz.[14] It has become

extremely important in finite element analysis which, to many investigators, is typified as a 'variational process'.

If the functional Π is 'quadratic', i.e., if the function $\mathbf{u}$ and its derivatives occur in powers not exceeding 2, then Eq. (3.64) reduces to a standard linear form similar to Eq. (3.8), i.e.,

$$\frac{\partial \Pi}{\partial \mathbf{a}} \equiv \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{3.65}$$

It is easy to show that the matrix $\mathbf{K}$ will now always be symmetric. To do this let us consider a linearization of the vector $\partial\Pi/\partial\mathbf{a}$. This we can write as

$$\Delta\left(\frac{\partial \Pi}{\partial \mathbf{a}}\right) = \left[\begin{array}{c} \dfrac{\partial}{\partial \mathbf{a}_1}\left(\dfrac{\partial \Pi}{\partial \mathbf{a}_1}\right)\Delta\mathbf{a}_1, \dfrac{\partial}{\partial \mathbf{a}_2}\left(\dfrac{\partial \Pi}{\partial \mathbf{a}_1}\right)\Delta\mathbf{a}_2, \ldots \\ \vdots \end{array}\right] \equiv \mathbf{K}_{\mathrm{T}}\,\Delta\mathbf{a} \tag{3.66}$$

in which $\mathbf{K}_{\mathrm{T}}$ is generally known as the tangent matrix, of significance in non-linear analysis, and $\Delta\mathbf{a}_j$ are small incremental changes to $\mathbf{a}$. Now it is easy to see that

$$\mathbf{K}_{\mathrm{T}ij} = \frac{\partial^2 \Pi}{\partial \mathbf{a}_i\, \partial \mathbf{a}_j} = \mathbf{K}_{\mathrm{T}ji}^{\mathrm{T}} \tag{3.67}$$

Hence $\mathbf{K}_{\mathrm{T}}$ is symmetric.

For a quadratic functional we have, from Eq. (3.65),

$$\Delta\left(\frac{\partial \Pi}{\partial \mathbf{a}}\right) = \mathbf{K}\,\Delta\mathbf{a} \qquad \text{or} \qquad \mathbf{K} = \mathbf{K}^{\mathrm{T}} \tag{3.68}$$

and hence symmetry must exist.

The fact that *symmetric matrices will arise whenever a variational principle exists is one of the most important merits of variational approaches for discretization*. However, symmetric forms will frequently arise directly from the Galerkin process. In such cases we simply conclude that the variational principle exists but we shall not need to use it directly.

How then do 'variational principles' arise and is it always possible to construct these for continuous problems?

To answer the first part of the question we note that frequently the physical aspects of the problem can be stated directly in a variational principle form. Theorems such as minimization of total potential energy to achieve equilibrium in mechanical systems, least energy dissipation principles in viscous flow, etc., may be known to the reader and are considered by many as the basis of the formulation. We have already referred to the first of these in Sec. 2.4 of Chapter 2.

Variational principles of this kind are 'natural' ones but unfortunately they do not exist for all continuum problems for which well-defined differential equations may be formulated.

However, there is another category of variational principles which we may call 'contrived'. Such contrived principles can always be constructed for any differentially specified problems either by extending the number of unknown functions $\mathbf{u}$ by additional variables known as Lagrange multipliers, or by procedures imposing a higher degree of continuity requirements such as least square problems. In subsequent

sections we shall discuss, respectively, such 'natural' and 'contrived' variational principles.

Before proceeding further it is worth noting that, in addition to symmetry occurring in equations derived by variational means, sometimes further motivation arises. When 'natural' variational principles exist the quantity Π may be of specific interest itself. If this arises a variational approach possesses the merit of easy evaluation of this functional.

The reader will observe that if the functional is 'quadratic' and yields Eq. (3.65), then we can write the approximate 'functional' Π simply as

$$\Pi = \tfrac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} + \mathbf{a}^{\mathrm{T}}\mathbf{f} \tag{3.69}$$

By simple differentiation

$$\delta\Pi = \tfrac{1}{2}\delta(\mathbf{a}^{\mathrm{T}})\mathbf{K}\mathbf{a} + \tfrac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\,\delta\mathbf{a} + \delta\mathbf{a}^{\mathrm{T}}\mathbf{f}$$

As $\mathbf{K}$ is symmetric,

$$\delta\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} \equiv \mathbf{a}^{\mathrm{T}}\mathbf{K}\,\delta\mathbf{a}$$

Hence

$$\delta\Pi = \delta\mathbf{a}^{\mathrm{T}}(\mathbf{K}\mathbf{a} + \mathbf{f}) = 0$$

which is true for all $\delta\mathbf{a}$ and hence

$$\mathbf{K}\mathbf{a} + \mathbf{f} = 0$$

3.8 'Natural' variational principles and their relation to governing differential equations

3.8.1 Euler equations

If we consider the definitions of Eqs (3.61) and (3.62) we observe that for stationarity we can write, after performing some differentiations,

$$\delta\Pi = \int_{\Omega} \delta\mathbf{u}^{\mathrm{T}}\mathbf{A}(\mathbf{u})\,\mathrm{d}\Omega + \int_{\Gamma} \delta\mathbf{u}^{\mathrm{T}}\mathbf{B}(\mathbf{u})\,\mathrm{d}\Gamma = 0 \tag{3.70}$$

As the above has to be true for any variations $\delta\mathbf{u}$, we must have

$$\begin{aligned} &\mathbf{A}(\mathbf{u}) = \mathbf{0} \qquad \text{in } \Omega \\ \text{and} \qquad &\\ &\mathbf{B}(\mathbf{u}) = \mathbf{0} \qquad \text{on } \Gamma \end{aligned} \tag{3.71}$$

If $\mathbf{A}$ corresponds precisely to the differential equations governing the problem and $\mathbf{B}$ to its boundary conditions, then the variational principle is a *natural* one. Equations (3.71) are known as the Euler differential equations corresponding to the variational principle requiring the stationarity of Π. It is easy to show that for any variational principle a corresponding set of Euler equations can be established. The reverse is unfortunately not true, i.e., only certain forms of differential equations are Euler

equations of a variational functional. In the next section we shall consider the conditions necessary for the existence of variational principles and give a prescription for the establishment of Π from a set of suitable linear differential equations. In this section we shall continue to assume that the form of the variational principle is known.

To illustrate the process let us now consider a specific example. Suppose we specify the problem by requiring the stationarity of a functional

$$\Pi = \int_\Omega \left[\frac{1}{2} k \left(\frac{\partial \phi}{\partial x} \right)^2 + \frac{1}{2} k \left(\frac{\partial \phi}{\partial y} \right)^2 - Q\phi \right] \mathrm{d}\Omega - \int_{\Gamma_q} \bar{q}\phi \, \mathrm{d}\Gamma \tag{3.72}$$

in which k and Q depend only on position and $\delta\phi$ is defined such that $\delta\phi = 0$ on Γ_ϕ, where Γ_ϕ and Γ_q bound the domain Ω.

We now perform the variation.[12] This can be written following the rules of differentiation as

$$\delta\Pi = \int_\Omega \left[k \frac{\partial \phi}{\partial x} \delta \left(\frac{\partial \phi}{\partial x} \right) + k \frac{\partial \phi}{\partial y} \delta \left(\frac{\partial \phi}{\partial y} \right) - Q\,\delta\phi \right] \mathrm{d}\Omega - \int_{\Gamma_q} (\bar{q}\,\delta\phi) \, \mathrm{d}\Gamma \tag{3.73}$$

As

$$\delta \left(\frac{\partial \phi}{\partial x} \right) = \frac{\partial}{\partial x} (\delta\phi) \tag{3.74}$$

we can integrate by parts (as in Sec. 3.3) and, noting that $\delta\phi = 0$ on Γ_ϕ, obtain

$$\begin{aligned} \delta\Pi = & - \int_\Omega \delta\phi \left[\frac{\partial}{\partial x} \left(k \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial \phi}{\partial y} \right) + Q \right] \mathrm{d}\Omega \\ & + \int_{\Gamma_q} \delta\phi \left(k \frac{\partial \phi}{\partial n} - \bar{q} \right) \mathrm{d}\Gamma = 0 \end{aligned} \tag{3.75a}$$

This is of the form of Eq. (3.70) and we immediately observe that the Euler equations are

$$\begin{aligned} A(\phi) &= \frac{\partial}{\partial x} \left(k \frac{\partial \phi}{\partial y} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial \phi}{\partial y} \right) + Q \qquad \text{in } \Omega \\ B(\phi) &= k \frac{\partial \phi}{\partial n} - \bar{q} = 0 \qquad \text{on } \Gamma_q \end{aligned} \tag{3.75b}$$

If ϕ is prescribed so that $\phi = \bar{\phi}$ on Γ_ϕ and $\delta\phi = 0$ on that boundary, then the problem is precisely the one we have already discussed in Sec. 3.2 and the functional (3.72) specifies the *two-dimensional heat conduction* problem in an alternative way.

In this case we have 'guessed' the functional but the reader will observe that the variation operation could have been carried out for any functional specified and corresponding *Euler* equations could have been established.

Let us continue the process to obtain an approximate solution of the linear heat conduction problem. Taking, as usual,

$$\phi \approx \hat{\phi} = \sum N_i a_i = \mathbf{Na} \tag{3.76}$$

we substitute this approximation into the expression for the functional Π [Eq. (3.72)] and obtain

$$\Pi = \int_\Omega \frac{1}{2} k \left(\sum \frac{\partial N_i}{\partial x} a_i \right)^2 \mathrm{d}\Omega + \int_\Omega \frac{1}{2} k \left(\sum \frac{\partial N_i}{\partial y} a_i \right)^2 \mathrm{d}\Omega - \int_\Omega Q \sum N_i a_i \, \mathrm{d}\Omega - \int_{\Gamma_q} \bar{q} \sum N_i a_i \, \mathrm{d}\Gamma \tag{3.77}$$

On differentiation with respect to a typical parameter a_j we have

$$\frac{\partial \Pi}{\partial a_j} = \int_\Omega k \left(\sum \frac{\partial N_i}{\partial x} a_i \right) \frac{\partial N_j}{\partial x} \, \mathrm{d}\Omega + \int k \left(\sum \frac{\partial N_i}{\partial y} a_i \right) \frac{\partial N_j}{\partial y} \, \mathrm{d}\Omega - \int_\Omega Q N_j \, \mathrm{d}\Omega - \int_{\Gamma_q} \bar{q} N_j \, \mathrm{d}\Gamma \tag{3.78}$$

and a system of equations for the solution of the problem is

$$\mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{3.79}$$

with

$$\begin{aligned} K_{ij} = K_{ji} &= \int_\Omega k \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} \, \mathrm{d}\Omega + \int_\Omega k \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \, \mathrm{d}\Omega \\ f_j &= -\int_\Omega N_j Q \, \mathrm{d}\Omega - \int_{\Gamma_q} N_j \bar{q} \, \mathrm{d}\Gamma \end{aligned} \tag{3.80}$$

The reader will observe that the approximation equations are here identical with those obtained in Sec. 3.5 for the same problem using the Galerkin process. No special advantage accrues to the variational formulation here, and indeed we can predict now that *Galerkin and variational procedures must give the same answer for cases where natural variational principles exist*.

3.8.2 Relation of the Galerkin method to approximation via variational principles

In the preceding example we have observed that the approximation obtained by the use of a natural variational principle and by the use of the Galerkin weighting process proved identical. That this is the case follows directly from Eq. (3.70), in which the variation was derived in terms of the original differential equations and the associated boundary conditions.

If we consider the usual trial function expansion [Eq. (3.3)]

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}\mathbf{a}$$

we can write the variation of this approximation as

$$\delta \hat{\mathbf{u}} = \mathbf{N} \, \delta \mathbf{a} \tag{3.81}$$

and inserting the above into (3.70) yields

$$\delta\Pi = \delta\mathbf{a}^{\mathrm{T}} \int_{\Omega} \mathbf{N}^{\mathrm{T}}\mathbf{A}(\mathbf{N}\mathbf{a})\,\mathrm{d}\Omega + \delta\mathbf{a}^{\mathrm{T}} \int_{\Gamma} \mathbf{N}^{\mathrm{T}}\mathbf{B}(\mathbf{N}\mathbf{a})\,\mathrm{d}\Gamma = 0 \tag{3.82}$$

The above form, being true for all $\delta\mathbf{a}$, requires that the expression under the integrals should be zero. The reader will immediately recognize this as simply the Galerkin form of the weighted residual statement discussed earlier [Eq. (3.25)], and identity is hereby proved.

We need to underline, however, that this is only true if the Euler equations of the variational principle coincide with the governing equations of the original problem. The Galerkin process thus retains its greater range of applicability.

At this stage another point must be made, however. If we consider a *system* of governing equations [Eq. (3.1)]

$$\mathbf{A}(\mathbf{u}) = \begin{Bmatrix} A_1(\mathbf{u}) \\ A_2(\mathbf{u}) \\ \vdots \end{Bmatrix} = \mathbf{0}$$

with $\hat{\mathbf{u}} = \mathbf{N}\mathbf{a}$, the Galerkin weighted residual equation becomes (disregarding the boundary conditions)

$$\int_{\Omega} \mathbf{N}^{\mathrm{T}}\mathbf{A}(\hat{\mathbf{u}})\,\mathrm{d}\Omega = \mathbf{0} \tag{3.83}$$

This form is not unique as the system of equations $\mathbf{A}$ can be ordered in a number of ways. Only one such ordering will correspond precisely with the Euler equations of a variational principle (if this exists) and the reader can verify that for an equation system weighted in the Galerkin manner at best only one arrangement of the vector $\mathbf{A}$ results in a symmetric set of equations.

As an example, consider, for instance, the one-dimensional heat conduction problem (Example 1, Sec. 3.3) redefined as an equation system with two unknowns, ϕ being the temperature and q the heat flow. Disregarding at this stage the boundary conditions we can write these equations as

$$\mathbf{A}(\mathbf{u}) = \begin{Bmatrix} q - \dfrac{\mathrm{d}\phi}{\mathrm{d}x} \\ \dfrac{\mathrm{d}q}{\mathrm{d}x} + Q \end{Bmatrix} = \mathbf{0} \tag{3.84}$$

or as a linear equation system,

$$\mathbf{A}(\mathbf{u}) \equiv \mathbf{L}\mathbf{u} + \mathbf{b} = \mathbf{0}$$

in which

$$\mathbf{L} \equiv \begin{bmatrix} 1, & -\dfrac{\mathrm{d}}{\mathrm{d}x} \\ \dfrac{\mathrm{d}}{\mathrm{d}x}, & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{Bmatrix} 0 \\ Q \end{Bmatrix} \qquad \mathbf{u} = \begin{Bmatrix} q \\ \phi \end{Bmatrix} \tag{3.85}$$

Writing the trial function in which a different interpolation is used for each function

$$\mathbf{u} = \sum \mathbf{N}_i\mathbf{a}_i \qquad \mathbf{N}_i = \begin{bmatrix} N_i^1 & 0 \\ 0 & N_i^2 \end{bmatrix}$$

and applying the Galerkin process, we arrive at the usual linear equation system with

$$\mathbf{K}_{ij} = \int_\Omega \mathbf{N}_i^{\mathrm{T}} \mathbf{L} \mathbf{N}_j \, \mathrm{d}x = \int_\Omega \begin{bmatrix} N_i^1 N_j^1, & -N_i^1 \dfrac{\mathrm{d}N_j^2}{\mathrm{d}x} \\ N_i^2 \dfrac{\mathrm{d}N_j^1}{\mathrm{d}x}, & 0 \end{bmatrix} \mathrm{d}x \tag{3.86}$$

After integration by parts, this form yields a symmetric equation† system and

$$K_{ij} = K_{ji} \tag{3.87}$$

If the order of equations were simply reversed, i.e., using

$$\mathbf{A}(\mathbf{u}) = \begin{bmatrix} \dfrac{\mathrm{d}q}{\mathrm{d}x} + Q \\ q - \dfrac{\mathrm{d}\phi}{\mathrm{d}x} \end{bmatrix} = 0 \tag{3.88}$$

application of the Galerkin process would now lead to non-symmetric equations quite different from those arising using the variational principle. The second type of Galerkin approximation would clearly be less desirable due to loss of symmetry in the final equations. It is easy to show that the first system corresponds precisely to the Euler equations of the variational functional deduced in the next section.

3.9 Establishment of natural variational principles for linear, self-adjoint differential equations

3.9.1 General theorems

General rules for deriving natural variational principles from non-linear differential equations are complicated and even the tests necessary to establish the existence of such variational principles are not simple. Much mathematical work has been done, however, in this context by Vainberg,[15] Tonti,[16] Oden,[17] and others.

For linear differential equations the situation is much simpler and a thorough study is available in the works of Mikhlin,[18,19] and in this section a brief presentation of such rules is given.

We shall consider here only the establishment of variational principles for a linear system of equations with *forced* boundary conditions, implying only variation of functions which yield $\delta\mathbf{u} = \mathbf{0}$ on their boundaries. The extension to include natural boundary conditions is simple and will be omitted.

Writing a linear system of differential equations as

$$\mathbf{A}(\mathbf{u}) \equiv \mathbf{L}\mathbf{u} + \mathbf{b} = \mathbf{0} \tag{3.89}$$

† As

$$\int N_i^1 \frac{\mathrm{d}N_j^2}{\mathrm{d}x} \, \mathrm{d}x \equiv -\int \frac{\mathrm{d}N_i^1}{\mathrm{d}x} N_j^2 \, \mathrm{d}x + \text{boundary terms}$$

in which $\mathbf{L}$ is a linear differential operator it can be shown that natural variational principles require that the operator $\mathbf{L}$ be such that

$$\int_\Omega \boldsymbol{\psi}^{\mathrm{T}}(\mathbf{L}\boldsymbol{\gamma})\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{\gamma}^{\mathrm{T}}(\mathbf{L}\boldsymbol{\psi})\,\mathrm{d}\Omega + \text{b.t.} \tag{3.90}$$

for any two function sets $\boldsymbol{\psi}$ and $\boldsymbol{\gamma}$. In the above, 'b.t.' stands for boundary terms which we disregard in the present context. The property required in the above operator is called that of *self-adjointness* or *symmetry*.

If the operator $\mathbf{L}$ is self-adjoint, the variational principle can be written immediately as

$$\Pi = \int_\Omega [\tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{L}\mathbf{u} + \mathbf{u}^{\mathrm{T}}\mathbf{b}]\,\mathrm{d}\Omega + \text{b.t.} \tag{3.91}$$

To prove the veracity of the last statement a variation needs to be considered. We thus write

$$\delta\Pi = \int_\Omega [\tfrac{1}{2}\delta\mathbf{u}^{\mathrm{T}}\mathbf{L}\mathbf{u} + \tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\delta(\mathbf{L}\mathbf{u}) + \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}]\,\mathrm{d}\Omega + \text{b.t.} \tag{3.92}$$

Noting that for any linear operator

$$\delta(\mathbf{L}\mathbf{u}) \equiv \mathbf{L}\,\delta\mathbf{u} \tag{3.93}$$

and that $\mathbf{u}$ and $\delta\mathbf{u}$ can be treated as any two independent functions, by identity (3.90) we can write Eq. (3.92) as

$$\delta\Pi = \int_\Omega \delta\mathbf{u}^{\mathrm{T}}[\mathbf{L}\mathbf{u} + \mathbf{b}]\,\mathrm{d}\Omega + \text{b.t.} \tag{3.94}$$

We observe immediately that the term in the brackets, i.e. the Euler equation of the functional, is identical with the original equation postulated, and therefore the variational principle is verified.

The above gives a very simple test and a prescription for the establishment of natural variational principles for differential equations of the problem.

Consider, for instance, two examples.

Example 1. This is a problem governed by the differential equation similar to the heat conduction equation, e.g.,

$$\nabla^2\phi + c\phi + Q = 0 \tag{3.95}$$

with c and Q being dependent on position only.

The above can be written in the general form of Eq. (3.89), with

$$\mathbf{L} \equiv \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + c\right] \qquad \mathbf{b} \equiv Q \tag{3.96}$$

Verifying that self-adjointness applies (which we leave to the reader as an exercise), we immediately have a variational principle

$$\Pi = \int_\Omega \left[\frac{1}{2}\phi\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + c\phi\right) + Q\phi\right]\mathrm{d}x\,\mathrm{d}y \tag{3.97}$$

with ϕ satisfying the forced boundary condition, i.e., $\phi = \bar{\phi}$ on Γ_ϕ. Integration by parts of the first two terms results in

$$\Pi = -\int_\Omega \left[\frac{1}{2}\left(\frac{\partial \phi}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial \phi}{\partial y}\right)^2 - \frac{1}{2}c\phi^2 - Q\phi \right] \mathrm{d}x\,\mathrm{d}y \tag{3.98}$$

on noting that boundary terms with prescribed ϕ do not alter the principle.

Example 2. This problem concerns the equation system discussed in the previous section [Eqs (3.84) and (3.85)]. Again self-adjointness of the operator can be tested, and found to be satisfied. We now write the functional as

$$\begin{aligned}\Pi &= \int_\Omega \left(\frac{1}{2} \begin{Bmatrix} q \\ \phi \end{Bmatrix}^{\mathrm{T}} \begin{bmatrix} 1, & -\dfrac{\mathrm{d}}{\mathrm{d}x} \\ \dfrac{\mathrm{d}}{\mathrm{d}x}, & 0 \end{bmatrix} \begin{Bmatrix} q \\ \phi \end{Bmatrix} + \begin{Bmatrix} q \\ \phi \end{Bmatrix}^{\mathrm{T}} \begin{Bmatrix} 0 \\ Q \end{Bmatrix} \right) \mathrm{d}x \\ &= \int_\Omega \left(\frac{1}{2}q^2 - \frac{1}{2}q\frac{\mathrm{d}\phi}{\partial x} + \frac{1}{2}\phi\frac{\mathrm{d}q}{\mathrm{d}x} + \phi Q \right) \mathrm{d}x \end{aligned} \tag{3.99}$$

The verification of the correctness of the above, by executing a variation, is left to the reader.

These two examples illustrate the simplicity of application of the general expressions. The reader will observe that self-adjointness of the operator will generally exist if even orders of differentiation are present. For odd orders self-adjointness is only possible if the operator is a 'skew'-symmetric matrix such as occurs in the second example.

3.9.2 Adjustment for self-adjointness

On occasion a linear operator which is not self-adjoint can be adjusted so that self-adjointness is achieved without altering the basic equation. Consider, for instance, the problem governed by the following differential equation of a standard linear form:

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}x^2} + \alpha\frac{\mathrm{d}\phi}{\mathrm{d}x} + \beta\phi + Q = 0 \tag{3.100}$$

In this equation α and β are functions of x. It is easy to see that the operator $\mathbf{L}$ is now a scalar:

$$L \equiv \frac{\mathrm{d}^2}{\mathrm{d}x^2} + \alpha\frac{\mathrm{d}}{\mathrm{d}x} + \beta \tag{3.101}$$

and is not self-adjoint.

Let p be some, as yet undetermined, function of x. We shall show that it is possible to convert Eq. (3.100) to a self-adjoint form by multiplying it by this function. The new operator becomes

$$\bar{L} = pL \tag{3.102}$$

To test for symmetry with any two functions ψ and γ we write

$$\int_\Omega \psi(pL\gamma)\,\mathrm{d}x = \int_\Omega \left(\psi p\frac{\mathrm{d}^2\gamma}{\mathrm{d}x^2} + \psi p\alpha\frac{\mathrm{d}\gamma}{\mathrm{d}x} + \psi p\beta\gamma\right)\mathrm{d}x \tag{3.103}$$

On integration of the first term, by parts, we have (b.t. denoting boundary terms)

$$\begin{aligned}\int_\Omega \left(-\frac{\mathrm{d}(\psi p)}{\mathrm{d}x}\frac{\mathrm{d}\gamma}{\mathrm{d}x} + \psi p\alpha\frac{\mathrm{d}\gamma}{\mathrm{d}x} + \beta\psi p\gamma\,\mathrm{d}x\right)\mathrm{d}x + \text{b.t.}\\ = \int_\Omega \left[-\frac{\mathrm{d}\psi}{\mathrm{d}x}p\frac{\mathrm{d}\gamma}{\mathrm{d}x} + \psi\frac{\mathrm{d}\gamma}{\mathrm{d}x}\left(p\alpha - \frac{\mathrm{d}p}{\mathrm{d}x}\right) + \psi p\beta\gamma\right]\mathrm{d}x + \text{b.t.}\end{aligned} \tag{3.104}$$

Symmetry (and therefore self-adjointness) is now achieved in the first and last terms. The middle term will only be symmetric if it disappears, i.e., if

$$p\alpha - \frac{\mathrm{d}p}{\mathrm{d}x} = 0 \tag{3.105}$$

or

$$\frac{\mathrm{d}p}{p} = \alpha\,\mathrm{d}x; \qquad p = \mathrm{e}^{\int \alpha\,\mathrm{d}x} \tag{3.106}$$

By using this value of p the operator is made self-adjoint and a variational principle for the problem of Eq. (3.100) is easily found.

A procedure of this kind has been used by Guymon *et al.*[20] to derive variational principles for a convective diffusion equation which is not self-adjoint. (We have noted such lack of symmetry in the equation in Example 2, Sec. 3.3.)

A similar method for creating variational functionals can be extended to the special case of non-linearity of Eq. (3.89) when

$$\mathbf{b} = \mathbf{b}(\mathbf{u}, x, \ldots) \tag{3.107}$$

If Eq. (3.92) is inspected we note that we could write

$$\delta(\mathbf{u}^{\mathrm{T}}\mathbf{b}) = \delta(\mathrm{g}) \tag{3.108}$$

if

$$\mathrm{g} = \int \mathbf{b}^{\mathrm{T}}\,\mathrm{d}\mathbf{u}$$

This integration is often quite easy to accomplish.

3.10 Maximum, minimum, or a saddle point?

In discussing variational principles so far we have assumed simply that at the solution point $\delta\Pi = 0$, that is the functional is stationary. It is often desirable to know whether Π is at a maximum, minimum, or simply at a 'saddle point'. If a maximum or a minimum is involved, then the approximation will always be 'bounded', i.e., will provide approximate values of Π which are either smaller or larger than the correct ones.† This in itself may be of practical significance.

† Provided all integrals are exactly evaluated.

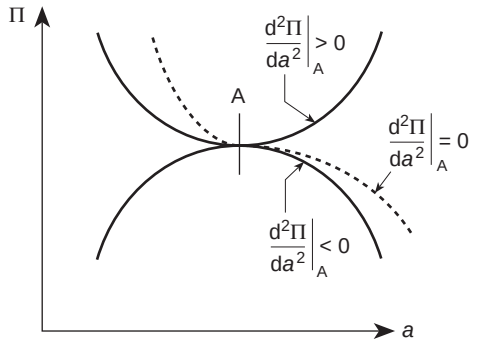


Fig. 3.7 Maximum, minimum, and a 'saddle' point for a functional Π of one variable.

When, in elementary calculus, we consider a stationary point of a function Π of one variable a, we investigate the rate of change of dΠ with da and write

$$\mathrm{d}(\mathrm{d}\Pi) = \mathrm{d}\left(\frac{\partial \Pi}{\partial a}\,\mathrm{d}a\right) = \frac{\partial^2 \Pi}{\partial a^2}(\mathrm{d}a)^2 \tag{3.109}$$

The sign of the second derivative determines whether Π is a minimum, maximum, or simply stationary (saddle point), as shown in Fig. 3.7. By analogy in the calculus of variations we shall consider changes of $\delta\Pi$. Noting the general form of this quantity given by Eq. (3.63) and the notion of the second derivative of Eq. (3.66) we can write, in terms of discrete parameters,

$$\delta(\delta\Pi) \equiv \delta\left(\frac{\partial \Pi}{\partial \mathbf{a}}\right)^{\mathrm{T}} \delta\mathbf{a} = \delta\mathbf{a}^{\mathrm{T}} \delta\left(\frac{\partial \Pi}{\partial \mathbf{a}}\right) = \delta\mathbf{a}^{\mathrm{T}} \mathbf{K}_{\mathrm{T}}\, \delta\mathbf{a} \tag{3.110}$$

If, in the above, $\delta(\delta\Pi)$ is always negative then Π is obviously reaching a maximum, if it is always positive then Π is a minimum, but if the sign is indeterminate this shows only the existence of a saddle point.

As $\delta\mathbf{a}$ is an arbitrary vector this statement is equivalent to requiring the matrix $\mathbf{K}_{\mathrm{T}}$ to be negative definite for a maximum *or* positive definite for a minimum. The form of the matrix $\mathbf{K}_{\mathrm{T}}$ (or in linear problems of $\mathbf{K}$ which is identical to it) is thus of great importance in variational problems.

3.11 Constrained variational principles. Lagrange multipliers and adjoint functions

3.11.1 Lagrange multipliers

Consider the problem of making a functional Π stationary, subject to the unknown $\mathbf{u}$ obeying some set of additional differential relationships

$$\mathbf{C}(\mathbf{u}) = \mathbf{0} \qquad \text{in } \Omega \tag{3.111}$$

We can introduce this constraint by forming another functional

$$\bar{\Pi}(\mathbf{u},\boldsymbol{\lambda}) = \Pi(\mathbf{u}) + \int_\Omega \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}(\mathbf{u})\,\mathrm{d}\Omega \tag{3.112}$$

in which $\boldsymbol{\lambda}$ is some set of functions of the independent coordinates in the domain Ω known as *Lagrange multipliers*. The variation of the new functional is now

$$\delta\bar{\Pi} = \delta\Pi + \int_\Omega \boldsymbol{\lambda}^{\mathrm{T}}\delta\mathbf{C}(\mathbf{u})\,\mathrm{d}\Omega + \int_\Omega \delta\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}(\mathbf{u})\,\mathrm{d}\Omega \tag{3.113}$$

and this is zero providing $\mathbf{C}(\mathbf{u}) = \mathbf{0}$ and, simultaneously,

$$\delta\bar{\Pi} = 0 \tag{3.114}$$

In a similar way, constraints can be introduced at some points or over boundaries of the domain. For instance, if we require that $\mathbf{u}$ obey

$$\mathbf{E}(\mathbf{u}) = \mathbf{0} \qquad \text{on } \Gamma \tag{3.115}$$

we would add to the original functional the term

$$\int_\Gamma \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{E}(\mathbf{u})\,\mathrm{d}\Gamma \tag{3.116}$$

with $\boldsymbol{\lambda}$ now being an unknown function defined only on Γ. Alternatively, if the constraint $\mathbf{C}$ is applicable only at one or more points of the system, then the simple addition of $\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}(\mathbf{u})$ at these points to the general functional Π will introduce a discrete number of constraints.

It appears, therefore, possible to always introduce additional functions $\boldsymbol{\lambda}$ and modify a functional to include any prescribed constraints. In the 'discretization' process we shall now have to use trial functions to describe both $\mathbf{u}$ and $\boldsymbol{\lambda}$.

Writing, for instance,

$$\hat{\mathbf{u}} = \sum \mathbf{N}_i\mathbf{a}_i = \mathbf{N}\mathbf{a} \qquad \hat{\boldsymbol{\lambda}} = \sum \bar{\mathbf{N}}_i\mathbf{b}_i = \bar{\mathbf{N}}\mathbf{b} \tag{3.117}$$

we shall obtain a set of equations

$$\frac{\partial\Pi}{\partial\mathbf{c}} = \begin{Bmatrix} \dfrac{\partial\Pi}{\partial\mathbf{a}} \\ \dfrac{\partial\Pi}{\partial\mathbf{b}} \end{Bmatrix} = \mathbf{0} \qquad \mathbf{c} = \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} \tag{3.118}$$

from which both the sets of parameters $\mathbf{a}$ and $\mathbf{b}$ can be obtained. It is somewhat paradoxical that the 'constrained' problem has resulted in a larger number of unknown parameters than the original one and, indeed, has complicated the solution. We shall, nevertheless, find practical use for Lagrange multipliers in formulating some physical variational principles, and will make use of these in a more general context in Chapters 11 and 12.

Example. The point about increasing the number of parameters to introduce a constraint may perhaps be best illustrated in a simple algebraic situation in which we require a stationary value of a quadratic function of two variables a_1 and a_2:

$$\Pi = 2a_1^2 - 2a_1a_2 + a_2^2 + 18a_1 + 6a_2 \tag{3.119}$$

subject to a constraint

$$a_1 - a_2 = 0 \tag{3.120}$$

The obvious way to proceed would be to insert directly the equality 'constraint' and obtain

$$\Pi = a_1^2 + 24a_1 \tag{3.121}$$

and write, for stationarity,

$$\frac{\partial \Pi}{\partial a_1} = 0 = 2a_1 + 24 \qquad a_1 = a_2 = -12 \tag{3.122}$$

Introducing a Lagrange multiplier λ we can alternatively find the stationarity of

$$\bar{\Pi} = 2a_1^2 - 2a_1a_2 + a_2^2 + 18a_1 + 6a_2 + \lambda(a_1 - a_2) \tag{3.123}$$

and write *three* simultaneous equations

$$\frac{\partial \bar{\Pi}}{\partial a_1} = 0 \qquad \frac{\partial \bar{\Pi}}{\partial a_2} = 0 \qquad \frac{\partial \bar{\Pi}}{\partial \lambda} = 0 \tag{3.124}$$

The solution of the above system again yields the correct answer

$$a_1 = a_2 = -12 \qquad \lambda = 6$$

but at considerably more effort. Unfortunately, in most continuum problems direct elimination of constraints cannot be so simply accomplished.†

Before proceeding further it is of interest to investigate the form of equations resulting from the modified functional Π of Eq. (3.112). If the original functional Π gave as its Euler equations a system

$$\mathbf{A}(\mathbf{u}) = \mathbf{0} \tag{3.125}$$

then we have

$$\delta\bar{\Pi} = \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{A}(\mathbf{u})\,\mathrm{d}\Omega + \int_\Omega \delta\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}(\mathbf{u})\,\mathrm{d}\Omega + \int_\Omega \boldsymbol{\lambda}^{\mathrm{T}}\delta\mathbf{C}\,\mathrm{d}\Omega \tag{3.126}$$

Substituting the trial functions (3.117) we can write for a linear set of constraints

$$\begin{aligned} \mathbf{C}(\mathbf{u}) &= \mathbf{L}_1\mathbf{u} + \mathbf{C}_1 \\ \delta\bar{\Pi} &= \delta\mathbf{a}^{\mathrm{T}}\int_\Omega \mathbf{N}^{\mathrm{T}}\mathbf{A}(\hat{\mathbf{u}})\,\mathrm{d}\Omega + \delta\mathbf{b}^{\mathrm{T}}\int_\Omega \bar{\mathbf{N}}^{\mathrm{T}}(\mathbf{L}_1\hat{\mathbf{u}} + \mathbf{C}_1)\,\mathrm{d}\Omega \\ &\quad + \delta\mathbf{a}^{\mathrm{T}}\int_\Omega (\mathbf{L}_1\mathbf{N})^{\mathrm{T}}\hat{\boldsymbol{\lambda}}\,\mathrm{d}\Omega = 0 \end{aligned} \tag{3.127}$$

As this has to be true for all variations $\delta\mathbf{a}$ and $\delta\mathbf{b}$, we have a system of equations

$$\begin{aligned} \int_\Omega \mathbf{N}^{\mathrm{T}}\mathbf{A}(\hat{\mathbf{u}})\,\mathrm{d}\Omega + \int_\Omega (\mathbf{L}_1\mathbf{N})^{\mathrm{T}}\hat{\boldsymbol{\lambda}}\,\mathrm{d}\Omega &= 0 \\ \int_\Omega \bar{\mathbf{N}}^{\mathrm{T}}(\mathbf{L}_1\hat{\mathbf{u}} + \mathbf{C}_1)\,\mathrm{d}\Omega &= 0 \end{aligned} \tag{3.128}$$

† In the finite element context, Szabo and Kassos[21] use such direct elimination; however, this involves considerable algebraic manipulation.

For linear equations **A**, the first term of the first equation is precisely the ordinary, unconstrained, variational approximation

$$\mathbf{K}_{aa}\mathbf{a} + \mathbf{f}_a \tag{3.129}$$

and inserting again the trial functions (3.117) we can write the approximated Eq. (3.128) as a linear system:

$$\mathbf{K}_c\mathbf{c} = \begin{bmatrix} \mathbf{K}_{aa}, & \mathbf{K}_{ab} \\ \mathbf{K}_{ab}^{\mathrm{T}}, & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} + \begin{Bmatrix} \mathbf{f}_a \\ \mathbf{f}_b \end{Bmatrix} = \mathbf{0} \tag{3.130}$$

with

$$\mathbf{K}_{ab}^{\mathrm{T}} = \int_\Omega \bar{\mathbf{N}}^{\mathrm{T}}\mathbf{L}_1\mathbf{N}\,\mathrm{d}\Omega; \qquad \mathbf{f}_b = -\int_\Omega \bar{\mathbf{N}}^{\mathrm{T}}\mathbf{C}_1\,\mathrm{d}\Omega \tag{3.131}$$

Clearly the system of equations is symmetric but now possesses zeros on the diagonal, and therefore the variational principle Π is merely stationary. Further, computational difficulties may be encountered unless the solution process allows for zero diagonal terms.

3.11.2 Identification of Lagrange multipliers. Forced boundary conditions and modified variational principles

Although the Lagrange multipliers were introduced as a mathematical fiction necessary for the enforcement of certain external constraints required to satisfy the original variational principle, we shall find that in most physical situations they can be identified with certain physical quantities of importance to the original mathematical model. Such an identification will follow immediately from the definition of the variational principle established in Eq. (3.112) and through the second of the Euler equations corresponding to it. The variation $\delta\bar{\Pi}$, written in Eq. (3.113), supplies through its third term the constraint equation. The first two terms can always be rewritten as

$$\int_\Omega \boldsymbol{\lambda}^{\mathrm{T}}\delta\mathbf{C}(\mathbf{u})\,\mathrm{d}\Omega + \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{A}(\mathbf{u})\,\mathrm{d}\Omega + \text{b.t.} = \mathbf{0} \tag{3.132}$$

This supplies the identification of $\boldsymbol{\lambda}$.

In the literature of variational calculation such identification arises frequently and the reader is referred to the excellent text by Washizu[22] for numerous examples.

Example. Here we shall introduce this identification by means of the example considered in Sec. 3.8.1. As we have noted, the variational principle of Eq. (3.72) established the governing equation and the natural boundary conditions of the heat conduction problem providing the forced boundary condition

$$\mathbf{C}(\phi) = \phi - \bar{\phi} = 0 \tag{3.133}$$

was satisfied on Γ_ϕ in the choice of the trial function for ϕ.

The above forced boundary condition can, however, be considered as a constraint on the original problem. We can write the constrained variational principle as

$$\bar{\Pi} = \Pi + \int_{\Gamma_\phi} \lambda(\phi - \bar{\phi})\, \mathrm{d}\Gamma \tag{3.134}$$

where Π is given by Eq. (3.72).

Performing the variation we have

$$\delta\bar{\Pi} = \delta\Pi + \int_{\Gamma_\phi} \delta\lambda(\phi - \bar{\phi})\, \mathrm{d}\Gamma + \int_{\Gamma_\phi} \delta\phi\lambda\, \mathrm{d}\Gamma \tag{3.135}$$

$\delta\Pi$ is now given by the expression (3.75a) augmented by an integral

$$\int_{\Gamma_\phi} \delta\phi k \frac{\partial\phi}{\partial n}\, \mathrm{d}\Gamma \tag{3.136}$$

which was previously disregarded (as we had assumed that $\delta\phi = 0$ on Γ_ϕ). In addition to the conditions of Eq. (3.75b), we now require that

$$\int_{\Gamma_\phi} \delta\lambda(\phi - \bar{\phi})\, \mathrm{d}\Gamma + \int_{\Gamma_\phi} \delta\phi\left(\lambda + k\frac{\partial\phi}{\partial n}\right) \mathrm{d}\Gamma = 0 \tag{3.137}$$

which must be true for all variations $\delta\lambda$ and $\delta\phi$. The first simply reiterates the constraint

$$\phi - \bar{\phi} = 0 \qquad \text{on } \Gamma_\phi \tag{3.138}$$

The second *defines* λ as

$$\lambda = -k\frac{\partial\phi}{\partial n} \tag{3.139}$$

Noting that $k(\partial\phi/\partial n)$ is equal to the flux q_n on the boundary Γ_ϕ, the physical identification of the multiplier has been achieved.

The identification of the Lagrange variable leads to the possible establishment of a modified variational principal in which λ is replaced by the identification.

We could thus write a new principle for the above example:

$$\bar{\Pi} = \Pi - \int_{\Gamma_\phi} k\frac{\partial\phi}{\partial n}(\phi - \bar{\phi})\, \mathrm{d}\Gamma \tag{3.140}$$

in which once again Π is given by the expression (3.72) but ϕ is not constrained to satisfy any boundary conditions. Use of such modified variational principles can be made to restore interelement continuity and appears to have been first introduced for that purpose by Kikuchi and Ando.[23] In general these present interesting new procedures for establishing useful variational principles.

A further extension of such principles has been made use of by Chen and Mei[24] and Zienkiewicz *et al.*[25] Washizu[22] discusses many such applications in the context of structural mechanics. The reader can verify that the variational principle expressed in Eq. (3.140) leads to automatic satisfaction of all the necessary boundary conditions in the example considered.

The use of modified variational principles restores the problem to the original number of unknown functions or parameters and is often computationally advantageous.

3.11.3 A general variational principle: adjoint functions and operators

The Lagrange multiplier method leads to an obvious procedure of 'creating' a variational principle for any set of equations even if the operators are not self-adjoint:

$$\mathbf{A}(\mathbf{u}) = \mathbf{0} \tag{3.141}$$

Treating all the above equations as a set of constraints we can obtain such a general variational functional simply by putting $\Pi = 0$ in Eq. (3.112) and writing

$$\bar{\Pi} = \int_\Omega \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{A}(\mathbf{u})\, \mathrm{d}\Omega \tag{3.142}$$

now requiring stationarity for all variations of $\delta\boldsymbol{\lambda}$ and $\delta\mathbf{u}$. The new variational principle has, however, been introduced at the expense of doubling the number of variables in the discretized situation. Treating the case of linear equations only, i.e.,

$$\mathbf{A}(\mathbf{u}) = \mathbf{L}\mathbf{u} + \mathbf{g} = \mathbf{0} \tag{3.143}$$

and discretizing we note, going through the steps involved in Eqs (3.126) to (3.130), that the final system of equations now takes the form

$$\begin{bmatrix} \mathbf{0} & \mathbf{K}_{ab} \\ \mathbf{K}_{ab}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{f} \end{Bmatrix} = 0 \tag{3.144}$$

with

$$\begin{aligned} \mathbf{K}_{ab}^{\mathrm{T}} &= \int_\Omega \bar{\mathbf{N}}^{\mathrm{T}} \mathbf{L} \mathbf{N}\, \mathrm{d}\Omega \\ \mathbf{f} &= \int_\Omega \bar{\mathbf{N}}^{\mathrm{T}} \mathbf{g}\, \mathrm{d}\Omega \end{aligned} \tag{3.145}$$

The equations are completely decoupled and the second set can be solved independently for all the parameters $\mathbf{a}$ describing the unknowns in which we were originally interested without consideration of the parameters $\mathbf{b}$. It will be observed that this second set of equations is *identical with an, apparently arbitrary, weighted residual process*. We have thus completed the full circle and obtained the weighted residual forms of Sec. 3.3 from a general variational principle.

The function $\boldsymbol{\lambda}$ which appears in the variational principle of Eq. (3.142) is known as the *adjoint function to* $\mathbf{u}$.

By performing a variation on Eq. (3.142) it is easy to show that the Euler equations of the principle are such that

$$\mathbf{A}(\mathbf{u}) = \mathbf{0} \tag{3.146}$$

and

$$\mathbf{A}^*(\mathbf{u}) = \mathbf{0} \tag{3.147}$$

where the operator $\mathbf{A}^*$ is such that

$$\int_\Omega \boldsymbol{\lambda}^{\mathrm{T}} \delta(\mathbf{A}\mathbf{u})\, \mathrm{d}\Omega = \int_\Omega \delta\mathbf{u}^{\mathrm{T}} \mathbf{A}^*(\boldsymbol{\lambda})\, \mathrm{d}\Omega \tag{3.148}$$

The operator $\mathbf{A}^*$ is known as the adjoint operator and will exist only in linear problems (see Appendix H).

For the full significance of the adjoint operator the reader is advised to consult mathematical texts.[26]

3.12 Constrained variational principles. Penalty functions and the least square method

3.12.1 Penalty functions

In the previous section we have seen how the process of introducing Lagrange multipliers allows constrained variational principles to be obtained at the expense of increasing the total number of unknowns. Further, we have shown that even in linear problems the algebraic equations which have to be solved are now complicated by having zero diagonal terms. In this section we shall consider an alternative procedure of introducing constraints which does not possess these drawbacks.

Considering once again the problem of obtaining stationarity of Π with a set of constraint equations $\mathbf{C}(\mathbf{u}) = \mathbf{0}$ in domain Ω, we note that the product

$$\mathbf{C}^{\mathrm{T}}\mathbf{C} = C_1^2 + C_2^2 + \cdots \tag{3.149}$$

where

$$\mathbf{C}^{\mathrm{T}} = [C_1, C_2, \ldots]$$

must always be a quantity which is positive or zero. Clearly, the latter value is found when the constraints are satisfied and clearly the variation

$$\delta(\mathbf{C}^{\mathrm{T}}\mathbf{C}) = 0 \tag{3.150}$$

as the product reaches that minimum.

We can now immediately write a new functional

$$\bar{\bar{\Pi}} = \Pi + \alpha \int_{\Omega} \mathbf{C}^{\mathrm{T}}(\mathbf{u})\mathbf{C}(\mathbf{u})\, \mathrm{d}\Omega \tag{3.151}$$

in which α is a 'penalty number' and then require the stationarity for the constrained solution. If Π is itself a minimum of the solution then α should be a positive number. The solution obtained by the stationarity of the functional $\bar{\bar{\Pi}}$ will satisfy the constraints only approximately. The larger the value of α the better will be the constraints achieved. Further, it seems obvious that the process is best suited to cases where Π is a minimum (or maximum) principle, but success can be obtained even with purely saddle point problems. The process is equally applicable to constrants applied on boundaries or simple discrete constraints. In this latter case integration is dropped.

Example. To clarify ideas let us once again consider the algebraic problem of Sec. 3.11, in which the stationarity of a functional given by Eq. (3.119) was sought subject to a constraint. With the penalty function approach we now seek the

Table 3.1

$\alpha =$	1	2	6	10	100
$a_1 =$	−12.00	−12.00	−12.00	−12.00	−12.00
$a_2 =$	−13.50	−13.00	−12.43	−12.78	−12.03

minimum of a functional

$$\bar{\bar{\Pi}} = 2a_1^2 - 2a_1a_2 + a_2^2 + 18a_1 + 6a_2 + \alpha(a_1 - a_2)^2 \tag{3.152}$$

with respect to the variation of both parameters $a_1$and a_2. Writing the two simultaneous equations

$$\frac{\partial\bar{\bar{\Pi}}}{\partial a_1} = 0 \qquad \frac{\partial\bar{\bar{\Pi}}}{\partial a_2} = 0 \tag{3.153}$$

we find that as α is increased we approach the correct solution. In Table 3.1 the results are set out demonstrating the convergence.

The reader will observe that in a problem formulated in the above manner the constraint introduces no additional unknown parameters – but neither does it decrease their original number. The process will always result in strongly positive definite matrices if the original variational principle is one of a minimum.

In practical applications the method of penalty functions has proved to be quite effective,[27] and indeed is often introduced intuitively. One such 'intuitive' application was already made when we enforced the value of boundary parameters in the manner indicated in Chapter 1, Sec. 1.4.

In the example presented here (and frequently practised in the real assembly of discretized finite element equations), the forced boundary conditions are not introduced *a priori* and the problem gives, on assembly, a singular system of equations

$$\mathbf{Ka} + \mathbf{f} = \mathbf{0} \tag{3.154}$$

which can be obtained from a functional (providing $\mathbf{K}$ is symmetric)

$$\Pi = \tfrac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{Ka} + \mathbf{a}^{\mathrm{T}}\mathbf{f} \tag{3.155}$$

Introducing a prescribed value of a_1, i.e., writing

$$a_1 - \bar{a}_1 = 0 \tag{3.156}$$

the functional can be modified to

$$\bar{\bar{\Pi}} = \Pi + \alpha(a_1 - \bar{a}_1)^2 \tag{3.157}$$

yielding

$$\bar{\bar{K}}_{11} = K_{11} + 2\alpha \qquad \bar{\bar{f}}_1 = f_1 - 2\alpha\bar{a}_1 \tag{3.158}$$

and giving no change in any of the other matrix coefficients. This is precisely the procedure adopted in Chapter 1 (page 10) for modifying the equations, to introduce prescribed values of a_1 (2α here replacing α, the 'large number' of Sec. 1.4). Many applications of such a 'discrete' kind are discussed by Campbell.[28]

It is easy to show in another context[27,29] that the use of a high Poisson's ratio ($\nu \to 0.5$) for the study of incompressible solids or fluids is in fact equivalent to the introduction of a penalty term to suppress any compressibility allowed by an arbitrary displacement variation.

The use of the penalty function in the finite element context presents certain difficulties.

Firstly, the constrained functional of Eq. (3.151) leads to equations of the form

$$(\mathbf{K}_1 + \alpha\mathbf{K}_2)\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{3.159}$$

where $\mathbf{K}_1$ derives from the original functions and $\mathbf{K}_2$ from the constraints. As α increases the above equation degenerates:

$$\mathbf{K}_2\mathbf{a} = -\mathbf{f}/\alpha \to \mathbf{0}$$

and $\mathbf{a} = \mathbf{0}$ unless the matrix $\mathbf{K}_2$ is singular. The phenomenon where $\mathbf{a} \Rightarrow \mathbf{0}$ is known as *locking* and has often been encountered by researchers who failed to recognize its source. This singularity in the equations does not always arise and we shall discuss means of its introduction in Chapters 11 and 12.

Secondly, with large but finite values of α numerical difficulties will be encountered. Noting that discretization errors can be of comparable magnitude to those due to not *satisfying* the constraint, we can make

$$\alpha = \text{constant } (1/h)^n$$

ensuring a limiting convergence to the correct answer. Fried[30,31] discusses this problem in detail.

A more general discussion of the whole topic is given in reference 32 and in Chapter 12 where the relationship between Lagrange constraints and penalty forms is made clear.

3.12.2 Least square approximations

In Sec. 3.11.3 we have shown how a constrained variational principle procedure could be used to construct a general variational principle if the constraints become simply the governing equations of the problem

$$\mathbf{C}(\mathbf{u}) = \mathbf{A}(\mathbf{u}) \tag{3.160}$$

Obviously the same procedure can be used in the context of the penalty function approach by setting $\Pi = 0$ in Eq. (3.151). We can thus write a 'variational principle'

$$\bar{\bar{\Pi}} = \int_\Omega (A_1^2 + A_2^2 + \cdots)\, \mathrm{d}\Omega = \int_\Omega \mathbf{A}^{\mathrm{T}}(\mathbf{u})\mathbf{A}(\mathbf{u})\, \mathrm{d}\Omega \tag{3.161}$$

for any set of differential equations. In the above equation the boundary conditions are assumed to be satisfied by $\mathbf{u}$ (forced boundary condition) and the parameter α is dropped as it becomes a multiplier.

Clearly, the above statement is a requirement that the sum of the squares of the residuals of the differential equations should be a minimum at the correct solution.

This minimum is obviously zero at that point, and the process is simply the well-known *least square method* of approximation.

It is equally obvious that we could obtain the correct solution by minimizing any functional of the form

$$\bar{\bar{\Pi}} = \int_\Omega (p_1 A_1^2 + p_2 A_2^2 + \cdots)\, \mathrm{d}\Omega = \int_\Omega \mathbf{A}^{\mathrm{T}}(\mathbf{u})\mathbf{p}\mathbf{A}(\mathbf{u})\, \mathrm{d}\Omega \tag{3.162}$$

in which $p_1, p_2, \ldots$, etc., are positive valued weighting functions or constants and $\mathbf{p}$ is a diagonal matrix:

$$\mathbf{p} = \begin{bmatrix} p_1 & & & 0 \\ & p_2 & & \\ & & p_3 & \\ 0 & & & \ddots \end{bmatrix} \tag{3.163}$$

The above alternative form is sometimes convenient as it puts different importance on the satisfaction of individual components of the equation and allows additional freedom in the choice of the approximate solution. Once again this weighting function could be chosen so as to ensure a constant ratio of terms contributed by various elements, although this has not yet been put into practice.

Least square methods of the kind shown above are a very powerful alternative procedure for obtaining integral forms from which an approximate solution can be started, and have been used with considerable success.[33,34] As the least square variational principles can be written for *any* set of differential equations without introducing additional variables, we may well enquire what is the difference between these and the *natural variational principles* discussed previously. On performing a variation in a specific case the reader will find that the Euler equations which are obtained no longer give the original differential equations but give higher order derivatives of these. This introduces the possibility of spurious solutions if incorrect boundary conditions are used. Further, higher order continuity of trial function is now generally needed. This may be a serious drawback but frequently can be by-passed by stating the original problem as a set of lower order equations.

We shall now consider the general form of discretized equations resulting from the least square approximation for linear equation sets (again neglecting boundary conditions which are enforced). Thus, if we take

$$\mathbf{A}(\mathbf{u}) = \mathbf{L}\mathbf{u} + \mathbf{b} \tag{3.164}$$

and take the usual trial function approximation

$$\hat{\mathbf{u}} = \mathbf{N}\mathbf{a} \tag{3.165}$$

we can write, substituting into (3.162),

$$\bar{\bar{\Pi}} = \int_\Omega [(\mathbf{L}\mathbf{N})\mathbf{a} + \mathbf{b}]^{\mathrm{T}}\mathbf{p}[(\mathbf{L}\mathbf{N})\mathbf{a} + \mathbf{b}]\, \mathrm{d}\Omega \tag{3.166}$$

and obtain

$$\delta\bar{\bar{\Pi}} = \int_\Omega \delta\mathbf{a}^{\mathrm{T}}(\mathbf{L}\mathbf{N})^{\mathrm{T}}\mathbf{p}[(\mathbf{L}\mathbf{N})\mathbf{a} + \mathbf{b}]\, \mathrm{d}\Omega + \int_\Omega [(\mathbf{L}\mathbf{N})\mathbf{a} + \mathbf{b}]^{\mathrm{T}}\mathbf{p}(\mathbf{L}\mathbf{N})\, \delta\mathbf{a}\, \mathrm{d}\Omega = 0 \tag{3.167}$$

or, as $\mathbf{p}$ is symmetric,

$$\delta\bar{\bar{\Pi}} = 2\delta\mathbf{a}^{\mathrm{T}}\left\{\left[\int_{\Omega}(\mathbf{LN})^{\mathrm{T}}\mathbf{p}(\mathbf{LN})\,\mathrm{d}\Omega\right]\mathbf{a} + \int_{\Omega}(\mathbf{LN})^{\mathrm{T}}\mathbf{pb}\,\mathrm{d}\Omega\right\} = 0 \tag{3.168}$$

This immediately yields the approximation equation in the usual form:

$$\mathbf{Ka} + \mathbf{f} = \mathbf{0} \tag{3.169}$$

and the reader can observe that the matrix $\mathbf{K}$ is symmetric and positive definite.

To illustrate an actual example, consider the problem governed by Eq. (3.95) for which we have already obtained a *natural* variational principle [Eq. (3.98)] in which only first derivatives were involved requiring C_0 continuity for $\mathbf{u}$. Now, if we use the operator $\mathbf{L}$ and term $\mathbf{b}$ defined by Eq. (3.96), we have a set of approximating equations with

$$\begin{aligned} K_{ij} &= \int_{\Omega}(\nabla^2 N_i + cN_i)(\nabla^2 N_j + cN_j)\,\mathrm{d}x\,\mathrm{d}y \\ f_i &= \int_{\Omega}(\nabla^2 N_i + cN_i)Q\,\mathrm{d}x\,\mathrm{d}y \end{aligned} \tag{3.170}$$

The reader will observe that now C_1 continuity is needed for the trial functions $\mathbf{N}$.

An alternative avoiding this difficulty is to write Eq. (3.95) as a first-order system. This can be written as

$$\mathbf{A}(\mathbf{u}) = \begin{Bmatrix} \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} + c\phi + Q \\ \dfrac{\partial \phi}{\partial x} - q_x \\ \dfrac{\partial \phi}{\partial y} - q_y \end{Bmatrix} = 0 \tag{3.171}$$

or, introducing the vector $\mathbf{u}$,

$$\mathbf{u} = [\phi, q_x, q_y]^{\mathrm{T}} = (\mathbf{Na}) \tag{3.172}$$

as the unknown we can write the standard linear form (3.164) as

$$\mathbf{Lu} + \mathbf{b} = \mathbf{0}$$

where

$$\mathbf{L} = \begin{bmatrix} c, & \dfrac{\partial}{\partial x}, & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial x}, & -1, & 0 \\ \dfrac{\partial}{\partial y}, & 0, & -1 \end{bmatrix} \qquad \mathbf{b} = \begin{Bmatrix} Q \\ 0 \\ 0 \end{Bmatrix} \tag{3.173}$$

The reader can now perform the substitution into Eq. (3.168) to obtain the approximation equations in a form requiring only C_0 continuity – introduced,

however, at the expense of additional variables. Use of such forms has been made extensively in the finite element context.[33,34]

3.12.3 Galerkin least squares, stabilization

It is interesting to note that the concept of penalty formulation introduced earlier in this section was anticipated as early as 1943 by Courant[35] in a somewhat different manner. He used the original variational principle augmented by the differential equations of the problem employed as least square constraints. In this manner he claimed, though never proved, that the convergence rate could be accelerated.

The suggestion put forward by Courant has been used effectively by others though in a somewhat different manner. Noting that the Galerkin process is, for self-adjoint equations, equivalent to that of minimizing a functional, the least square formulation using the original equation is simply added to the Galerkin form. Here it allows non-self adjoint operators to be used, for instance, and this feature has been exploited with success. Consider, for instance, the problem which we have discussed in Section 3.9.2 [viz. Eq. (3.100)] with $\beta = 0$. This equation, as we have already pointed out, is non-self adjoint but Galerkin methods have been successfully used in its solution providing the convection term ($\alpha\, \mathrm{d}\phi/\mathrm{d}x$) remains relatively small compared to the second derivative term (the diffusion term). However, it is found that as the convection term increases the solution becomes highly oscillatory. We shall discuss the stabilization of such problems in a general manner exhaustively in Volume 3 as such problems are frequently encountered in fluid mechanics. But here it is easy to consider the problem in a preliminary manner. Suppose in a Galerkin form given by

$$\int_\Omega \left\{ \frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}\phi}{\mathrm{d}x} - v\left(\alpha\frac{\mathrm{d}\phi}{\mathrm{d}x} + Q\right)\right\} \mathrm{d}x = 0 \tag{3.174}$$

we add a multiple of the minimization of the least square of the total equation. The result is

$$\int_\Omega \left\{ \frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}\phi}{\mathrm{d}x} - v\left(\alpha\frac{\mathrm{d}\phi}{\mathrm{d}x} + Q\right)\right\} \mathrm{d}x + \int_\Omega \left(\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} + \alpha\frac{\mathrm{d}v}{\mathrm{d}x}\right)\tau\left(\frac{\mathrm{d}^2\phi}{\mathrm{d}x^2} + \alpha\frac{\mathrm{d}\phi}{\mathrm{d}x} + Q\right) \mathrm{d}x = 0 \tag{3.175}$$

and we see immediately that an additional diffusive term has been added which depends on the parameter τ, though at the expense of having higher derivatives appearing in the integrals. If only linear elements are used and the discontinuities ignored at element interfaces, the process of adding the diffusive terms can *stabilize* the oscillations which would otherwise occur. The idea appears to have first been used by Hughes[36]. This process in the view of the authors is somewhat unorthodox as discontinuity of derivatives is ignored, and alternatives to this will be discussed at length in Chapter two of Volume 3.

It interesting to note also that another application of the same Galerkin least square process can be made to the mixed formulation with two variables $\mathbf{u}$ and p for incompressible problems. We shall discuss such problems in Chapter 12 of this volume and show how this process can be made applicable there.

Finally, it is of interest to note that the simple procedure introduced by Courant can also be effective in the prevention of locking of other problems. The treatment for beams has been studied by Freund and Salonen[40] and it appears that quite an effective process can be reached.

3.13 Concluding remarks – finite difference and boundary methods

This very extensive chapter presents the general possibilities of using the finite element process in almost any mathematical or mathematically modelled physical problem. The essential approximation processes have been given in as simple a form as possible, at the same time presenting a fully comprehensive picture which should allow the reader to understand much of the literature and indeed to experiment with new permutations. In the chapters that follow we shall apply to various physical problems a limited selection of the methods to which allusion has been made. In some we shall show, however, that certain extensions of the process are possible (Chapters 12 and 16) and in another (Chapter 10) how a violation of some of the rules here expounded can be accepted.

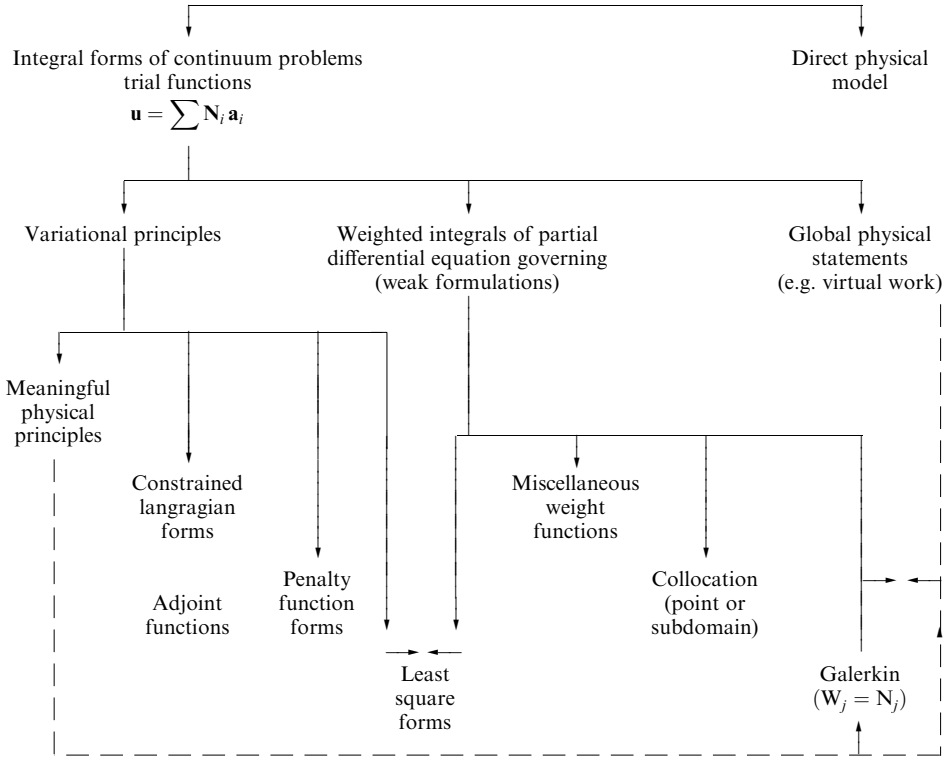
The numerous approximation procedures discussed fall into several categories. To remind the reader of these, we present in Table 3.2 a comprehensive catalogue of the methods used here and in Chapter 2. The only aspect of the finite element process mentioned in that table that has not been discussed here is that of a *direct physical method*. In such models an 'atomic' rather than continuum concept is the starting point. While much interest exists in the possibilities offered by such models, their discussion is outside the scope of this book.

In all the continuum processes discussed the first step is always the choice of suitable shape or trial functions. A few simple forms of such functions have been introduced as the need demanded and many new forms will be introduced in subsequent chapters. Indeed, the reader who has mastered the essence of the present chapter will have little difficulty in applying the finite element method to any suitably defined physical problem. For further reading references 41–45 could be consulted.

The methods listed do not include specifically two well-known techniques, i.e., *finite difference* methods and *boundary solution* methods (sometimes known as boundary elements). In the general sense these belong under the category of the *generalized finite element* method discussed here.[41]

1. Boundary solution methods choose the trial functions such that the governing equation is automatically satisfied in the domain Ω. Thus starting from the general approximation equation (3.25), we note that only boundary terms are retained. We shall return to such approximations in Chapter 13.
2. Finite difference procedures can be interpreted as an approximation based on local, discontinuous, shape functions with collocation weighting applied (although

Table 3.2 Finite element approximation



usually the derivation of the approximation algorithm is based on a Taylor expansion).

As Galerkin or variational approaches give, in the energy sense, the best approximation, this method has only the merit of computational simplicity and occasionally a loss of accuracy.

To illustrate this process we discuss an approximation carried out for the one-dimensional equation (3.27) (viz. p. 47). Here we represent a localized approximation through equally spaced nodal point values by

$$\phi(x) = \left[\frac{1}{2}\left(\frac{(x-x_i)^2}{h^2} - \frac{x-x_i}{h}\right),\ 1 - \frac{(x-x_i)^2}{h^2},\ \frac{1}{2}\left(\frac{(x-x_i)^2}{h^2} + \frac{x-x_i}{h^2}\right)\right] \times \begin{Bmatrix} \phi_{i-1} \\ \phi_i \\ \phi_{i+1} \end{Bmatrix} \tag{3.176}$$

where $h = x_{i+1} - x_i$ (shown in Fig. 3.8). It is clear that adjacent parabolic approximations in this case are discontinuous between the nodes. Values of the function and its

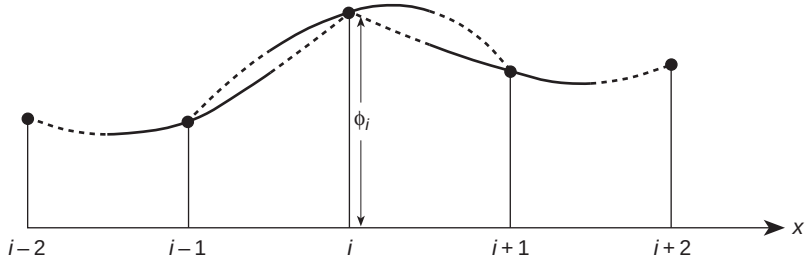


Fig. 3.8. A local, discontinuous shape function by parabolic segments used to obtain a finite difference approximation.

first two derivatives at a typical node i are given by

$$\begin{aligned} \phi(x_i) &= \phi_i \\ \left.\frac{\partial \phi}{\partial x}\right|_{x=x_i} &= \frac{1}{2h}(\phi_{i+1} - \phi_{i-1}) \\ \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{x=x_i} &= \frac{1}{h^2}(\phi_{i+1} - 2\phi_i + \phi_{i-1}) \end{aligned} \tag{3.177}$$

If we insert these into the governing equation at node i, we note immediately that the approximating equation at the node becomes

$$\frac{1}{h^2}(\phi_{i-1} - 2\phi_i + \phi_{i+1}) + Q_i = 0 \tag{3.178}$$

This is identical (within a multiple of h) to the assembled finite element equations (which we did not do explicitly) for the approximation with linear elements discussed in Eq. (3.35). This is indeed one of the cases in which the approximation is identical rather than different. In Chapter 16 we shall be discussing such finite difference and point approximations in more detail. However, the reader will note the present exercise is simply given to underline the similarity of finite element and finite difference processes.

Many textbooks deal exclusively with these types of approximations. References 46–50 discuss finite difference approximation and references 51–54 relate to boundary methods.

References

1. S.H. Crandall. *Engineering Analysis*. McGraw-Hill, 1956.
2. B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, 1972.
3. R.A. Frazer, W.P. Jones, and S.W. Sken. *Approximations to functions and to the solutions of differential equations*. Aero. Research Committee Report 1799, 1937.
4. C.B. Biezeno and R. Grammel. *Technische Dynamik*, p. 142, Springer-Verlag, 1933.
5. B.G. Galerkin. Series solution of some problems of elastic equilibrium of rods and plates (Russian). *Vestn. Inzh. Tech.*, **19**, 897–908, 1915.

6. Also attributed to Bubnov, 1913: see S.C. Mikhlin. *Variational Methods in Mathematical Physics*. Macmillan, 1964.
7. P. Tong. Exact solution of certain problems by the finite element method. *J. AIAA*, **7**, 179–80, 1969.
8. R.V. Southwell. *Relaxation Methods in Theoretical Physics*. Clarendon Press, 1946.
9. R.S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.
10. S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. 2nd ed., McGraw-Hill, 1951.
11. L.V. Kantorovitch and V.I. Krylov. *Approximate Methods of Higher Analysis*. Wiley (International), 1958.
12. F.B. Hildebrand. *Methods of Applied Mathematics*, 2nd edn. Dover Publications, 1992.
13. J.W. Strutt (Lord Rayleigh). On the theory of resonance. *Trans. Roy. Soc.* (*London*), **A161**, 77–118, 1870.
14. W. Ritz. Über eine neue Methode zur Lösung gewissen Variations – Probleme der Mathematischen Physik. *J. Reine angew. Math.*, **135**, 1–61, 1909.
15. M.M. Vainberg. *Variational Methods for the Study of Nonlinear Operators*. Holden-Day, 1964.
16. E. Tonti. Variational formulation of non-linear differential equations. *Bull. Acad. Roy. Belg.* (*Classe Sci.*), **55**, 137–65 and 262–78, 1969.
17. J.T. Oden. A general theory of finite elements – I: Topological considerations, pp. 205–21, and II: Applications, pp. 247–60. *Int. J. Num. Meth. Eng.*, **1**, 1969.
18. S.C. Mikhlin. *Variational Methods in Mathematical Physics*. Macmillan, 1964.
19. S.C. Mikhlin. *The Problems of the Minimum of a Quadratic Functional*. Holden-Day, 1965.
20. G.L. Guymon, V.H. Scott, and L.R. Herrmann. A general numerical solution of the two-dimensional differential–convection equation by the finite element method. *Water Res.*, **6**, 1611–15, 1970.
21. B.A. Szabo and T. Kassos. Linear equation constraints in finite element approximations. *Int. J. Num. Meth. Eng.*, **9**, 563–80, 1975.
22. K. Washizu. *Variational Methods in Elasticity and Plasticity*. 2nd ed., Pergamon Press, 1975.
23. F. Kikuchi and Y. Ando. A new variational functional for the finite element method and its application to plate and shell problems. *Nucl. Eng. Des.*, **21**, 95–113, 1972.
24. H.S. Chen and C.C. Mei. *Oscillations and water forces in an offshore harbour*. Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics, Report 190, Cambridge, Mass., 1974.
25. O.C. Zienkiewicz, D.W. Kelly, and P. Bettess. The coupling of the finite element method and boundary solution procedures. *Int. J. Num. Meth. Eng.*, **11**, 355–75, 1977.
26. I. Stakgold. *Boundary Value Problems of Mathematical Physics*. Macmillan, 1967.
27. O.C. Zienkiewicz. Constrained variational principles and penalty function methods in the finite element analysis. *Lecture Notes in Mathematics*. No. 363, pp. 207–14, Springer-Verlag, 1974.
28. J. Campbell. *A finite element system for analysis and design*. Ph.D. thesis, Swansea, 1974.
29. D.J. Naylor. Stresses in nearly incompressible materials for finite elements with application to the calculation of excess pore pressures. *Int. J. Num. Meth. Eng.*, **8**, 443–60, 1974.
30. I. Fried. Finite element analysis of incompressible materials by residual energy balancing. *Int. J. Solids Struct.*, **10**, 993–1002, 1974.
31. I. Fried. Shear in C^0 and C^1 bending finite elements. *Int. J. Solids Struct.*, **9**, 449–60, 1973.
32. O.C. Zienkiewicz and E. Hinton. Reduced integration, function smoothing and non-conformity in finite element analysis. *J. Franklin Inst.*, **302**, 443–61, 1976.
33. P.P. Lynn and S.K. Arya. Finite elements formulation by the weighted discrete least squares method. *Int. J. Num. Meth. Eng.*, **8**, 71–90, 1974.

34. O.C. Zienkiewicz, D.R.J. Owen, and K.N. Lee. Least square finite element for elasto-static problems – use of reduced integration. *Int. J. Num. Meth. Eng.*, **8**, 341–58, 1974.
35. R. Courant. Variational methods for the solution of problems of equilibrium and vibration. *Bull. Amer Math. Soc.*, **49**, 1–61, 1943.
36. T.J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Comp. Meth. Appl. Mech. Eng.*, **59**, 85–99, 1986.
37. T.J.R. Hughes and L.P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. *Comp. Meth. Appl. Mech. Eng.*, **65**, 85–96, 1987.
38. T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations. *Comp. Meth. Appl. Mech. Eng.*, **73**, 173–189, 1989.
39. R. Codina. A comparison of some finite element methods for solving the diffusion–convection–reaction equation. *Comp. Meth. Appl. Mech. Eng.*, **156**, 185–210, 1998.
40. Jouni Freund and Eero-Matti Salonen. Sensitizing according to Courant the Timoshenko beam finite element solution. *Int. J. Num. Meth. Eng.*, **x**, 129–60, 1999.
41. O.C. Zienkiewicz and K. Morgan. *Finite Elements and Approximation*. Wiley, 1983.
42. E.B. Becker, G.F. Carey, and J.T. Oden. *Finite Elements*: *An Introduction*. Vol. 1, Prentice-Hall, 1981.
43. I. Fried. *Numerical Solution of Differential Equations*. Academic Press, New York, 1979.
44. A.J. Davies. *The Finite Element Method*. Clarendon Press, Oxford, 1980.
45. C.A.T. Fletcher. *Computational Galerkin Methods*. Springer-Verlag, 1984.
46. R.V. Southwell. *Relaxation Methods in Theoretical Physics*. 1st edn., Clarendon Press, Oxford, 1946.
47. R.V. Southwell. *Relaxation Methods in Theoretical Physics*. 2nd edn., Clarendon Press, Oxford, 1956.
48. D.N. de G. Allen. *Relaxation Methods*. McGraw-Hill, London, 1955.
49. F.B. Hildebrand. *Introduction to Numerical Analysis*. 2nd edn., Dover Publications, 1987.
50. A.R. Mitchell and D. Griffiths. *The Finite Difference Method in Partial Differential Equations*. John Wiley & Sons, London, 1980.
51. J. MacKerle and C.A. Brebbia, editors. *The Boundary Element Reference Book*. Computational Mechanics, Southampton, 1988.
52. G. Beer and J.O. Watson. *Introduction to Finite and Boundary Element Methods for Engineers*. John Wiley & Sons, London, 1993.
53. P.K. Banerjee. *The Boundary Element Methods in Engineering*. McGraw-Hill, London, 1994.
54. Prem K. Kythe. *An Introduction to Boundary Element Methods*. CRC Press, 1994.

4

Plane stress and plane strain

4.1 Introduction

Two-dimensional elastic problems were the first successful examples of the application of the finite element method.[1,2] Indeed, we have already used this situation to illustrate the basis of the finite element formulation in Chapter 2 where the general relationships were derived. These basic relationships are given in Eqs (2.1)–(2.5) and (2.23) and (2.24), which for quick reference are summarized in Appendix C.

In this chapter the particular relationships for the plane stress and plane strain problem will be derived in more detail, and illustrated by suitable practical examples, a procedure that will be followed throughout the remainder of the book.

Only the simplest, triangular, element will be discussed in detail but the basic approach is general. More elaborate elements to be discussed in Chapters 8 and 9 could be introduced to the same problem in an identical manner.

The reader not familiar with the applicable basic definitions of elasticity is referred to elementary texts on the subject, in particular to the text by Timoshenko and Goodier,[3] whose notation will be widely used here.

In both problems of plane stress and plane strain the displacement field is uniquely given by the u and v displacement in the directions of the cartesian, orthogonal x and y axes.

Again, in both, the only strains and stresses that have to be considered are the three components in the xy plane. In the case of *plane stress*, by definition, all other components of stress are zero and therefore give no contribution to internal work. In *plane strain* the stress in a direction perpendicular to the xy plane is not zero. However, by definition, the strain in that direction is zero, and therefore no contribution to internal work is made by this stress, which can in fact be explicitly evaluated from the three main stress components, if desired, at the end of all computations.

4.2 Element characteristics

4.2.1 Displacement functions

Figure 4.1 shows the typical triangular element considered, with nodes i, j, m

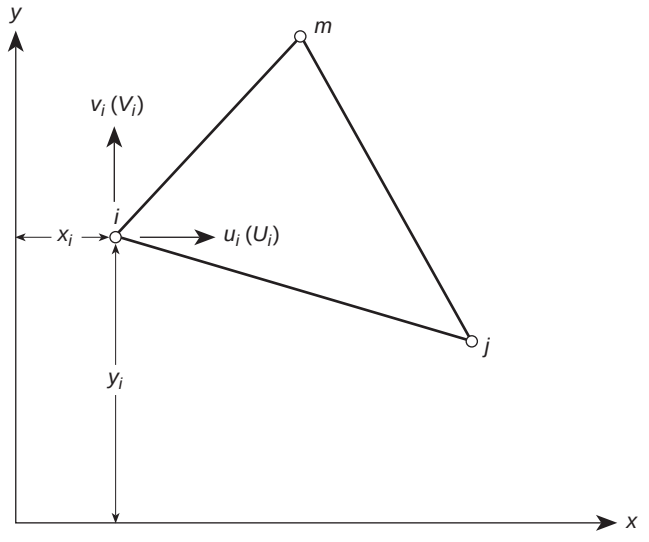


Fig. 4.1 An element of a continuum in plane stress or plane strain.

numbered in an anticlockwise order. The displacements of a node have two components

$$\mathbf{a}_i = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \tag{4.1}$$

and the six components of element displacements are listed as a vector

$$\mathbf{a}^e = \begin{Bmatrix} \mathbf{a}_i \\ \mathbf{a}_j \\ \mathbf{a}_m \end{Bmatrix} \tag{4.2}$$

The displacements within an element have to be uniquely defined by these six values. The simplest representation is clearly given by two linear polynomials

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y \\ v &= \alpha_4 + \alpha_5 x + \alpha_6 y \end{aligned} \tag{4.3}$$

The six constants α can be evaluated easily by solving the two sets of three simultaneous equations which will arise if the nodal coordinates are inserted and the displacements equated to the appropriate nodal displacements. Writing, for example,

$$\begin{aligned} u_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 y_i \\ u_j &= \alpha_1 + \alpha_2 x_j + \alpha_3 y_j \\ u_m &= \alpha_1 + \alpha_2 x_m + \alpha_3 y_m \end{aligned} \tag{4.4}$$

we can easily solve for α_1, α_2, and α_3 in terms of the nodal displacements u_i, u_j, u_m and obtain finally

$$u = \frac{1}{2\Delta}[(a_i + b_i x + c_i y)u_i + (a_j + b_j x + c_j y)u_j + (a_m + b_m x + c_m y)u_m] \tag{4.5a}$$

in which

$$\begin{aligned} a_i &= x_j y_m - x_m y_j \\ b_i &= y_j - y_m \\ c_i &= x_m - x_j \end{aligned} \tag{4.5b}$$

with the other coefficients obtained by a cycle permutation of subscripts in the order, i, j, m, and where†

$$2\Delta = \det \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} = 2 \cdot (\text{area of triangle } ijm) \tag{4.5c}$$

As the equations for the vertical displacement v are similar we also have

$$v = \frac{1}{2\Delta}[(a_i + b_i x + c_i y)v_i + (a_j + b_j x + c_j y)v_j + (a_m + b_m x + c_m y)v_m] \tag{4.6}$$

Though not strictly necessary at this stage we can represent the above relations, Eqs (4.5a) and (4.6), in the standard form of Eq. (2.1):

$$\mathbf{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \mathbf{N}\mathbf{a}^e = [\mathbf{I}N_i, \mathbf{I}N_j, \mathbf{I}N_m]\mathbf{a}^e \tag{4.7}$$

with $\mathbf{I}$ a two by two identity matrix, and

$$N_i = \frac{a_i + b_i x + c_i y}{2\Delta}, \qquad \text{etc.} \tag{4.8}$$

The chosen displacement function automatically guarantees continuity of displacement with adjacent elements because the displacements vary linearly along any side of the triangle and, with identical displacement imposed at the nodes, the same displacement will clearly exist all along an interface.

4.2.2 Strain (total)

The total strain at any point within the element can be defined by its three components which contribute to internal work. Thus

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x}, & 0 \\ 0, & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y}, & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} = \mathbf{S}\mathbf{u} \tag{4.9}$$

† *Note*: If coordinates are taken from the centroid of the element then

$$x_i + x_j + x_m = y_i + y_j + y_m = 0 \quad \text{and} \quad a_i = 2\Delta/3 = a_j = a_m$$

See also Appendix D for a summary of integrals for a triangle.

Substituting Eq. (4.7) we have

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{a}^e = [\mathbf{B}_i, \mathbf{B}_j, \mathbf{B}_m] \begin{Bmatrix} \mathbf{a}_i \\ \mathbf{a}_j \\ \mathbf{a}_m \end{Bmatrix} \tag{4.10a}$$

with a typical matrix $\mathbf{B}_i$ given by

$$\mathbf{B}_i = \mathbf{S}N_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x}, & 0 \\ 0, & \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial y}, & \dfrac{\partial N_i}{\partial x} \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_i, & 0 \\ 0, & c_i \\ c_i, & b_i \end{bmatrix} \tag{4.10b}$$

This defines matrix **B** of Eq. (2.4) explicitly.

It will be noted that in this case the **B** matrix is independent of the position within the element, and hence the strains are constant throughout it. Obviously, the criterion of constant strain mentioned in Chapter 2 is satisfied by the shape functions.

4.2.3 Elasticity matrix

The matrix **D** of Eq. (2.5)

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \mathbf{D} \left(\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - \boldsymbol{\varepsilon}_0 \right) \tag{4.11}$$

can be explicitly stated for any material (excluding here $\boldsymbol{\sigma}_0$ which is simply additive). To consider the special cases in two dimensions it is convenient to start from the form

$$\boldsymbol{\varepsilon} = \mathbf{D}^{-1}\boldsymbol{\sigma} + \boldsymbol{\varepsilon}_0$$

and impose the conditions of plane stress or plane strain.

Plane stress – isotropic material

For plane stress in an isotropic material we have by definition,

$$\begin{aligned} \varepsilon_x &= \frac{\sigma_x}{E} - \frac{\nu\sigma_y}{E} + \varepsilon_{x0} \\ \varepsilon_y &= -\frac{\nu\sigma_x}{E} + \frac{\sigma_y}{E} + \varepsilon_{y0} \\ \gamma_{xy} &= \frac{2(1+\nu)\tau_{xy}}{E} + \gamma_{xy0} \end{aligned} \tag{4.12}$$

Solving the above for the stresses, we obtain the matrix **D** as

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \tag{4.13}$$

and the initial strains as

$$\boldsymbol{\varepsilon}_0 = \begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{x0} \\ \gamma_{xy0} \end{Bmatrix} \tag{4.14}$$

in which E is the elastic modulus and ν is Poisson's ratio.

Plane strain – isotropic material

In this case a normal stress σ_z exists in addition to the other three stress components. Thus we now have

$$\begin{aligned} \varepsilon_x &= \frac{\sigma_x}{E} - \frac{\nu\sigma_y}{E} - \frac{\nu\sigma_z}{E} + \varepsilon_{x0} \\ \varepsilon_y &= -\frac{\nu\sigma_x}{E} + \frac{\sigma_y}{E} - \frac{\nu\sigma_z}{E} + \varepsilon_{y0} \\ \gamma_{xy} &= \frac{2(1+\nu)\tau_{xy}}{E} + \gamma_{xy0} \end{aligned} \tag{4.15}$$

and in addition

$$\varepsilon_z = -\frac{\nu\sigma_x}{E} - \frac{\nu\sigma_y}{E} + \frac{\sigma_z}{E} + \varepsilon_{z0} = 0$$

which yields

$$\sigma_z = \nu(\sigma_x + \sigma_y) - E\varepsilon_{z0}$$

On eliminating σ_z and solving for the three remaining stresses we obtain the matrix $\mathbf{D}$ as

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix} \tag{4.16}$$

and the initial strains

$$\boldsymbol{\varepsilon}_0 = \begin{Bmatrix} \varepsilon_{x0} + \nu\varepsilon_{z0} \\ \varepsilon_{y0} + \nu\varepsilon_{z0} \\ \gamma_{xy0} \end{Bmatrix} \tag{4.17}$$

Anisotropic materials

For a completely anisotropic material, 21 independent elastic constants are necessary to define completely the three-dimensional stress–strain relationship.[4,5]

If two-dimensional analysis is to be applicable a symmetry of properties must exist, implying at most six independent constants in the $\mathbf{D}$ matrix. Thus, it is always possible to write

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ & d_{22} & d_{23} \\ \text{sym.} & & d_{33} \end{bmatrix} \tag{4.18}$$

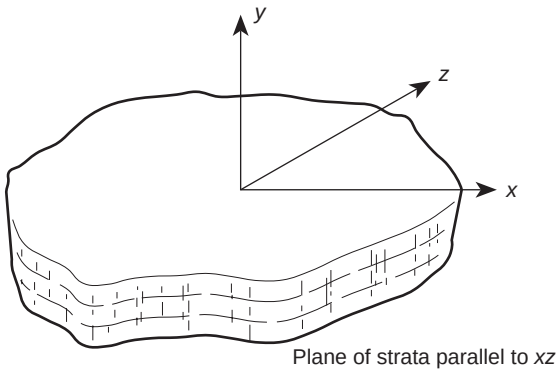


Fig. 4.2 A stratified (transversely isotropic) material.

to describe the most general two-dimensional behaviour. (The necessary symmetry of the **D** matrix follows from the general equivalence of the Maxwell–Betti reciprocal theorem and is a consequence of invariant energy irrespective of the path taken to reach a given strain state.)

A case of particular interest in practice is that of a 'stratified' or transversely isotropic material in which a rotational symmetry of properties exists within the plane of the strata. Such a material possesses only five independent elastic constants.

The general stress–strain relations give in this case, following the notation of Lekhnitskii[4] and taking now the y-axis as perpendicular to the strata (neglecting initial strain) (Fig. 4.2),

$$\begin{aligned}
\varepsilon_x &= \frac{\sigma_x}{E_1} - \frac{\nu_2\sigma_y}{E_2} - \frac{\nu_1\sigma_z}{E_1} \\
\varepsilon_y &= -\frac{\nu_2\sigma_x}{E_2} + \frac{\sigma_y}{E_2} - \frac{\nu_2\sigma_z}{E_2} \\
\varepsilon_z &= -\frac{\nu_1\sigma_x}{E_1} - \frac{\nu_2\sigma_y}{E_2} + \frac{\sigma_z}{E_1} \\
\gamma_{xz} &= \frac{2(1+\nu_1)}{E_1}\tau_{xz} \\
\gamma_{xy} &= \frac{1}{G_2}\tau_{xy} \\
\gamma_{yz} &= \frac{1}{G_2}\tau_{yz}
\end{aligned} \tag{4.19}$$

in which the constants E_1, ν_1 (G_1 is dependent) are associated with the behaviour in the plane of the strata and E_2, G_2, ν_2 with a direction normal to the plane.

The **D** matrix in two dimensions now becomes, taking $E_1/E_2 = n$ and $G_2/E_2 = m$,

$$\mathbf{D} = \frac{E_2}{1-n\nu_2^2}\begin{bmatrix} n & n\nu_2 & 0 \\ n\nu_2 & 1 & 0 \\ 0 & 0 & m(1-n\nu_2^2) \end{bmatrix} \tag{4.20}$$

for plane stress or

$$\mathbf{D} = \frac{E_2}{(1+\nu_1)(1-\nu_1-2n\nu_2^2)} \times \begin{bmatrix} n(1-n\nu_2^2) & n\nu_2(1+\nu_1) & 0 \\ n\nu_2(1+\nu_1) & (1-\nu_1^2) & 0 \\ 0 & 0 & m(1+\nu_1)(1-\nu_1-2n\nu_2^2) \end{bmatrix} \tag{4.21}$$

for plane strain.

When, as in Fig. 4.3, the direction of the strata is inclined to the x-axis then to obtain the $\mathbf{D}$ matrices in universal coordinates a transformation is necessary. Taking $\mathbf{D}'$ as relating the stresses and strains in the inclined coordinate system (x', y') it is easy to show that

$$\mathbf{D} = \mathbf{T}\mathbf{D}'\mathbf{T}^{\mathrm{T}} \tag{4.22}$$

where

$$\mathbf{T} = \begin{bmatrix} \cos^2\beta & \sin^2\beta & -2\sin\beta\cos\beta \\ \sin^2\beta & \cos^2\beta & 2\sin\beta\cos\beta \\ \sin\beta\cos\beta & -\sin\beta\cos\beta & \cos^2\beta - \sin^2\beta \end{bmatrix} \tag{4.23}$$

with β as defined in Fig. 4.3.

If the stress systems $\boldsymbol{\sigma}'$ and $\boldsymbol{\sigma}$ correspond to $\boldsymbol{\varepsilon}'$ and $\boldsymbol{\varepsilon}$ respectively then by equality of work

$$\boldsymbol{\sigma}'^{\mathrm{T}}\boldsymbol{\varepsilon}' = \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon}$$

or

$$\boldsymbol{\varepsilon}'^{\mathrm{T}}\mathbf{D}'\boldsymbol{\varepsilon}' = \boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}$$

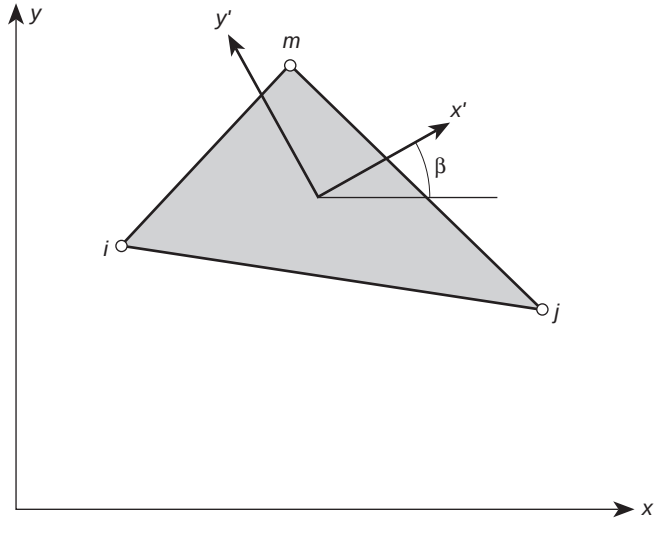


Fig. 4.3 An element of a stratified (transversely isotropic) material.

from which Eq. (4.22) follows on noting (see also Chapter 1)

$$\boldsymbol{\varepsilon}' = \mathbf{T}^{\mathrm{T}}\boldsymbol{\varepsilon} \tag{4.24}$$

4.2.4 Initial strain (thermal strain)

'Initial' strains, i.e., strains which are independent of stress, may be due to many causes. Shrinkage, crystal growth, or, most frequently, temperature change will, in general, result in an initial strain vector:

$$\boldsymbol{\varepsilon}_0 = [\varepsilon_{x0} \quad \varepsilon_{y0} \quad \varepsilon_{z0} \quad \gamma_{xy0} \quad \gamma_{yz0} \quad \gamma_{zx0}]^{\mathrm{T}} \tag{4.25}$$

Although this initial strain may, in general, depend on the position within the element, it will here be defined by average, constant values to be consistent with the constant strain conditions imposed by the prescribed displacement function.

For an isotropic material in an element subject to a temperature rise θ^e with a coefficient of thermal expansion α we will have

$$\boldsymbol{\varepsilon}_0 = \alpha\theta^e[1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}} \tag{4.26}$$

as no shear strains are caused by a thermal dilatation. Thus, for plane stress, Eq. (4.14) yields the initial strains given by

$$\boldsymbol{\varepsilon}_0 = \alpha\theta^e \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} = \alpha\theta^e \mathbf{m} \tag{4.27}$$

In *plane strain* the σ_z stress perpendicular to the xy plane will develop due to the thermal expansion as shown above. Using Eq. (4.17) the initial thermal strains for this case are given by

$$\boldsymbol{\varepsilon}_0 = (1+\nu)\alpha\theta^e \mathbf{m} \tag{4.28}$$

Anisotropic materials present special problems, since the coefficients of thermal expansion may vary with direction. In the general case the thermal strains are given by

$$\boldsymbol{\varepsilon}_0 = \boldsymbol{\alpha}\theta^e \tag{4.29}$$

where $\boldsymbol{\alpha}$ has properties similar to strain. Accordingly, it is always possible to find orthogonal directions for which $\boldsymbol{\alpha}$ is diagonal. If we let x' and y' denote the principal thermal directions of the material, the initial strain due to thermal expansion for a plane stress state becomes (assuming z' is a principal direction)

$$\boldsymbol{\varepsilon}' = \theta^e \begin{Bmatrix} \varepsilon_{x'0} \\ \varepsilon_{y'0} \\ \gamma_{x'y'0} \end{Bmatrix} = \theta^e \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ 0 \end{Bmatrix} \tag{4.30}$$

where α_1 and α_2 are the expansion coefficients referred to the x' and y' axes, respectively.

To obtain strain components in the x, y system it is necessary to use the strain transformation

$$\boldsymbol{\varepsilon}_0' = \mathbf{T}^{\mathrm{T}}\boldsymbol{\varepsilon}_0 \tag{4.31}$$

where $\mathbf{T}$ is again given by Eq. (4.23). Thus, $\boldsymbol{\varepsilon}_0$ can be simply evaluated. It will be noted that the shear component of strain is no longer equal to zero in the x, y coordinates.

4.2.5 The stiffness matrix

The stiffness matrix of the element *ijm* is defined from the general relationship (2.13) with the coefficients

$$\mathbf{K}_{ij}^{e} = \int \mathbf{B}_i^{\mathrm{T}}\mathbf{D}\mathbf{B}_j t \,\mathrm{d}x\,\mathrm{d}y \tag{4.32}$$

where t is the thickness of the element and the integration is taken over the area of the triangle. If the thickness of the element is assumed to be constant, an assumption convergent to the truth as the size of elements decreases, then, as neither of the matrices contains x or y we have simply

$$\mathbf{K}_{ij}^{e} = \mathbf{B}_i^{\mathrm{T}}\mathbf{D}\mathbf{B}_j t\Delta \tag{4.33}$$

where Δ is the area of the triangle [already defined by Eq. (4.5)]. This form is now sufficiently explicit for computation with the actual matrix operations being left to the computer.

4.2.6 Nodal forces due to initial strain

These are given directly by the expression Eq. (2.13b) which, on performing the integration, becomes

$$(\mathbf{f}_i)_{\boldsymbol{\varepsilon}_0}^{e} = -\mathbf{B}_i^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}_0 t\Delta, \quad \text{etc.} \tag{4.34}$$

These 'initial strain' forces contribute to the nodes of an element in an unequal manner and require precise evaluation. Similar expressions are derived for initial stress forces.

4.2.7 Distributed body forces

In the general case of plane stress or strain each element of unit area in the xy plane is subject to forces

$$\mathbf{b} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix}$$

in the direction of the appropriate axes.

Again, by Eq. (2.13b), the contribution of such forces to those at each node is given by

$$\mathbf{f}_i^e = -\int N_i \begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \mathrm{d}x\,\mathrm{d}y$$

or by Eq. (4.7),

$$\mathbf{f}_i^e = -\begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \int N_i \,\mathrm{d}x\,\mathrm{d}y, \quad \text{etc.} \tag{4.35}$$

if the body forces b_x and b_y are constant. As N_i is not constant the integration has to be carried out explicitly. Some general integration formulae for a triangle are given in Appendix D.

In this special case the calculation will be simplified if the origin of coordinates is taken at the centroid of the element. Now

$$\int x\,\mathrm{d}x\,\mathrm{d}y = \int y\,\mathrm{d}x\,\mathrm{d}y = 0$$

and on using Eq. (4.8)

$$\mathbf{f}_i^e = -\begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \int \frac{a_i\,\mathrm{d}x\,\mathrm{d}y}{2\Delta} = -\begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \frac{a_i}{2} = -\begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \frac{\Delta}{3} \tag{4.36}$$

by relations noted on page 89.

Explicitly, for the whole element

$$\mathbf{f}^e = \begin{Bmatrix} \mathbf{f}_i^e \\ \mathbf{f}_j^e \\ \mathbf{f}_m^e \end{Bmatrix} = -\begin{Bmatrix} b_x \\ b_y \\ b_x \\ b_y \\ b_x \\ b_y \end{Bmatrix} \frac{\Delta}{3} \tag{4.37}$$

which means simply that the total forces acting in the x and y directions due to the body forces are distributed to the nodes in three equal parts. This fact corresponds with physical intuition, and was often assumed implicitly.

4.2.8 Body force potential

In many cases the body forces are defined in terms of a body force potential ϕ as

$$b_x = -\frac{\partial \phi}{\partial x} \qquad b_y = -\frac{\partial \phi}{\partial y} \tag{4.38}$$

and this potential, rather than the values of b_x and b_y, is known throughout the region and is specified at nodal points. If $\boldsymbol{\phi}^e$ lists the three values of the potential associated with the nodes of the element, i.e.,

$$\boldsymbol{\phi}^e = \begin{Bmatrix} \phi_i \\ \phi_j \\ \phi_m \end{Bmatrix} \tag{4.39}$$

and has to correspond with constant values of b_x and b_y, ϕ must vary linearly within the element. The 'shape function' of its variation will obviously be given by a procedure identical to that used in deriving Eqs (4.4)–(4.6), and yields

$$\phi = [N_i, N_j, N_m]\boldsymbol{\phi}^e \tag{4.40}$$

Thus,

$$b_x = -\frac{\partial \phi}{\partial x} = -[b_i, b_j, b_m]\frac{\boldsymbol{\phi}^e}{2\Delta}$$

and

$$b_y = -\frac{\partial \phi}{\partial y} = -[c_i, c_j, c_m]\frac{\boldsymbol{\phi}^e}{2\Delta} \tag{4.41}$$

The vector of nodal forces due to the body force potential will now replace Eq. (4.37) by

$$\mathbf{f}^e = \frac{1}{6}\begin{bmatrix} b_i, & b_j, & b_m \\ c_i, & c_j, & c_m \\ b_i, & b_j, & b_m \\ c_i, & c_j, & c_m \\ b_i, & b_j, & b_m \\ c_i, & c_j, & c_m \end{bmatrix}\boldsymbol{\phi}^e \tag{4.42}$$

4.2.9 Evaluation of stresses

The derived formulae enable the full stiffness matrix of the structure to be assembled, and a solution for displacements to be obtained.

The stress matrix given in general terms in Eq. (2.16) is obtained by the appropriate substitutions for each element.

The stresses are, by the basic assumption, constant within the element. It is usual to assign these to the centroid of the element, and in most of the examples in this chapter this procedure is followed. An alternative consists of obtaining stress values at the nodes by averaging the values in the adjacent elements. Some 'weighting' procedures have been used in this context on an empirical basis but their advantage appears small.

It is also usual to calculate the principal stresses and their directions for every element. In Chapter 14 we shall return to the problem of stress recovery and show that better procedures of stress recovery exist.[6,7]

4.3 Examples – an assessment of performance

There is no doubt that the solution to plane elasticity problems as formulated in Sec. 4.2 is, in the limit of subdivision, an exact solution. Indeed at any stage of a

finite subdivision it is an approximate solution as is, say, a Fourier series solution with a limited number of terms.

As explained in Chapter 2, the total strain energy obtained during any stage of approximation will be below the true strain energy of the exact solution. In practice it will mean that the displacements, and hence also the stresses, will be underestimated by the approximation in its *general picture*. However, it must be emphasized that this is not necessarily true at every point of the continuum individually; hence the value of such a bound in practice is not great.

What is important for the engineer to know is the order of accuracy achievable in typical problems with a certain fineness of element subdivision. In any particular case the error can be assessed by comparison with known, exact, solutions or by a study of the convergence, using two or more stages of subdivision.

With the development of experience the engineer can assess *a priori* the order of approximation that will be involved in a specific problem tackled with a given element subdivision. Some of this experience will perhaps be conveyed by the examples considered in this book.

In the first place attention will be focused on some simple problems for which exact solutions are available.

4.3.1 Uniform stress field

If the exact solution is in fact that of a uniform stress field then, whatever the element subdivision, the finite element solution will coincide exactly with the exact one. This is an obvious corollary of the formulation; nevertheless it is useful as a first check of written computer programs.

4.3.2 Linearly varying stress field

Here, obviously, the basic assumption of constant stress within each element means that the solution will be approximate only. In Fig. 4.4 a simple example of a beam subject to constant bending moment is shown with a fairly coarse subdivision. It is readily seen that the axial (σ_y) stress given by the element 'straddles' the exact values and, in fact, if the constant stress values are associated with centroids of the elements and plotted, the best 'fit' line represents the exact stresses.

The horizontal and shear stress components differ again from the exact values (which are simply zero). Again, however, it will be noted that they oscillate by equal, small amounts around the exact values.

At internal nodes, if the average of the stresses of surrounding elements is taken it will be found that the exact stresses are very closely represented. The average at external faces is not, however, so good. The overall improvement in representing the stresses by nodal averages, as shown in Fig. 4.4, is often used in practice for contour plots. However, we shall show in Chapter 14 a method of recovery which gives much improved values at both interior and boundary points.

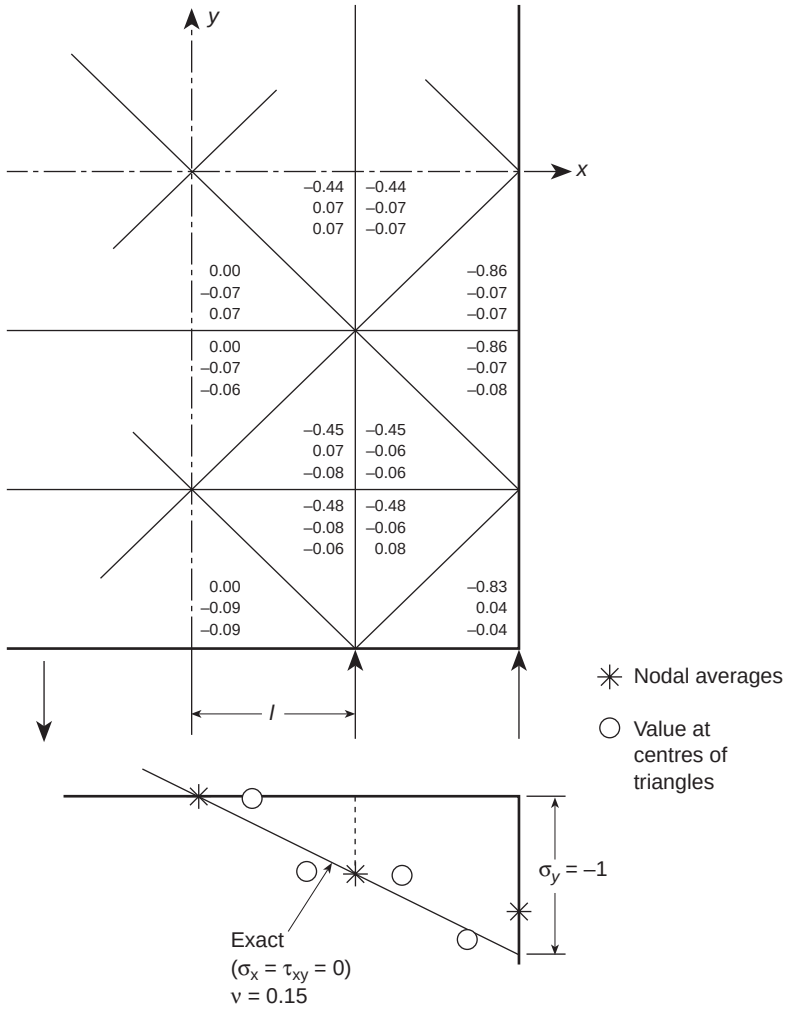


Fig. 4.4 Pure bending of a beam solved by a coarse subdivision into elements of triangular shape. (Values of σ_y, σ_x, and τ_{xy} listed in that order.)

4.3.3 Stress concentration

A more realistic test problem is shown in Figs 4.5 and 4.6. Here the flow of stress around a circular hole in an isotropic and in an anisotropic stratified material is considered when the stress conditions are uniform.[8] A graded division into elements is used to allow a more detailed study in the region where high stress gradients are expected. The accuracy achievable can be assessed from Fig. 4.6 where some of the results are compared against exact solutions.[3,9]

In later chapters we shall see that even more accurate answers can be obtained with the use of more elaborate elements; however, the principles of the analysis remain identical.

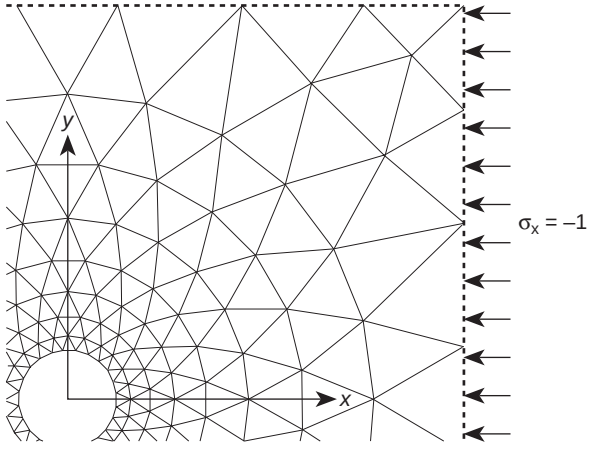


Fig. 4.5 A circular hole in a uniform stress field: (a) isotropic material; (b) stratified (orthotropic) material; $E_x = E_1 = 1$, $E_y = E_2 = 3$, $\nu_1 = 0.1$, $\nu_2 = 0$, $G_{xy} = 0.42$.

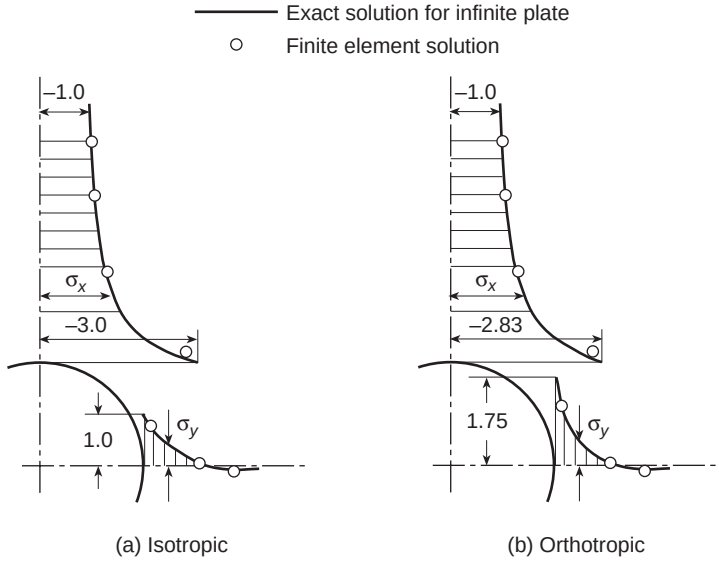


Fig. 4.6 Comparison of theoretical and finite element results for cases (a) and (b) of Fig. 4.5.

4.4 Some practical applications

Obviously, the practical applications of the method are limitless, and the finite element method has superseded experimental technique for plane problems because of its high accuracy, low cost, and versatility. The ease of treatment of material anisotropy, thermal stresses, or body force problems add to its advantages.

A few examples of actual early applications of the finite element method to complex problems of engineering practice will now be given.

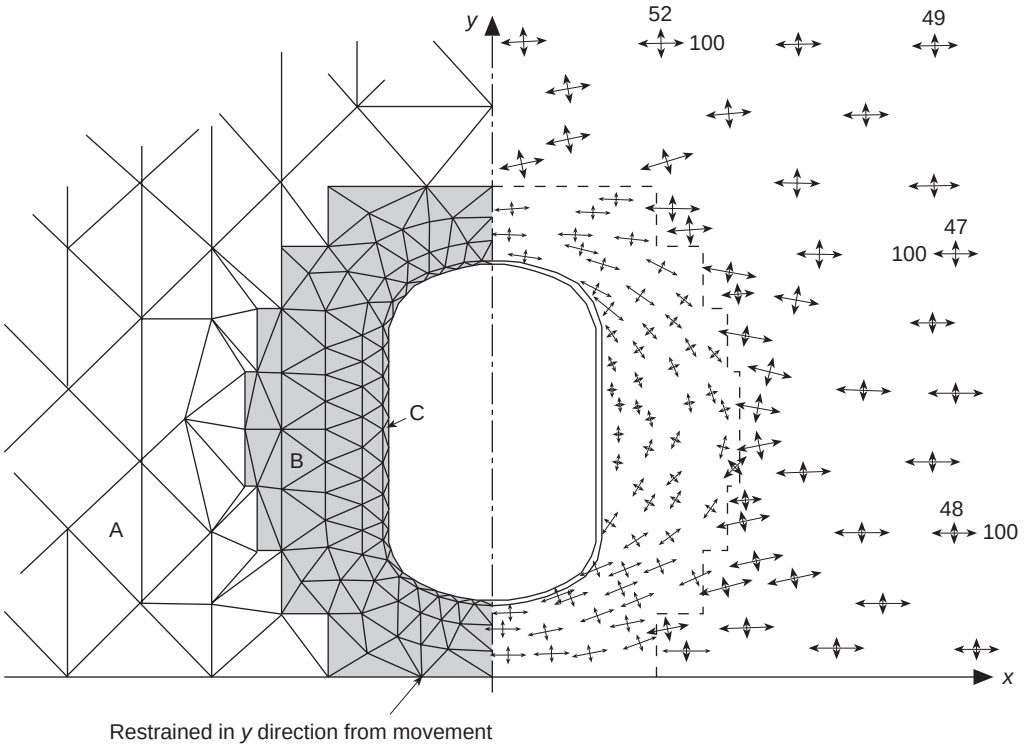


Fig. 4.7 A reinforced opening in a plate. Uniform stress field at a distance from opening $\sigma_x = 100$, $\sigma_y = 50$. Thickness of plate regions A, B, and C is in the ratio of 1 : 3 : 23.

4.4.1 Stress flow around a reinforced opening (Fig. 4.7)

In steel pressure vessels or aircraft structures, openings have to be introduced in the stressed skin. The penetrating duct itself provides some reinforcement round the edge and, in addition, the skin itself is increased in thickness to reduce the stresses due to concentration effects.

Analysis of such problems treated as cases of plane stress present no difficulties. The elements are chosen so as to follow the thickness variation, and appropriate values of this are assigned.

The narrow band of thick material near the edge can be represented either by special bar-type elements, or by very thin triangular elements of the usual type, to which appropriate thickness is assigned. The latter procedure was used in the problem shown in Fig. 4.7 which gives some of the resulting stresses near the opening itself. The fairly large extent of the region introduced in the analysis and the grading of the mesh should be noted.

4.4.2 An anisotropic valley subject to techtonic stress[8] (Fig. 4.8)

A symmetrical valley subject to a uniform horizontal stress is considered. The material is stratified, and hence is 'transversely isotropic', and the direction of strata varies from point to point.

The stress plot shows the tensile region that develops. This phenomenon is of considerable interest to geologists and engineers concerned with rock mechanics. (See reference 10 for additional applications on this topic.)

4.4.3 A dam subject to external and internal water pressures[11,12]

A buttress dam on a somewhat complex rock foundation is shown in Fig. 4.9 and analysed. This dam (completed in 1964) is of particular interest as it is the first to which the finite element method was applied during the design stage. The heterogeneous foundation region is subject to plane strain conditions while the dam itself is considered in a state of plane stress of variable thickness.

With external and gravity loading no special problems of analysis arise.

When pore pressures are considered, the situation, however, requires perhaps some explanation.

It is well known that in a porous material the water pressure is transmitted to the structure as a *body force* of magnitude

$$b_x = -\frac{\partial p}{\partial x} \qquad b_y = -\frac{\partial p}{\partial y} \tag{4.43}$$

and that now the external pressure need not be considered.

The pore pressure p is, in fact, now a body force potential, as defined in Eq. (4.38). Figure 4.9 shows the element subdivision of the region and the outline of the dam. Figure 4.10(*a*) and (*b*) shows the stresses resulting from gravity (applied to the dam only) and due to water pressure assumed to be acting as an external load or, alternatively,

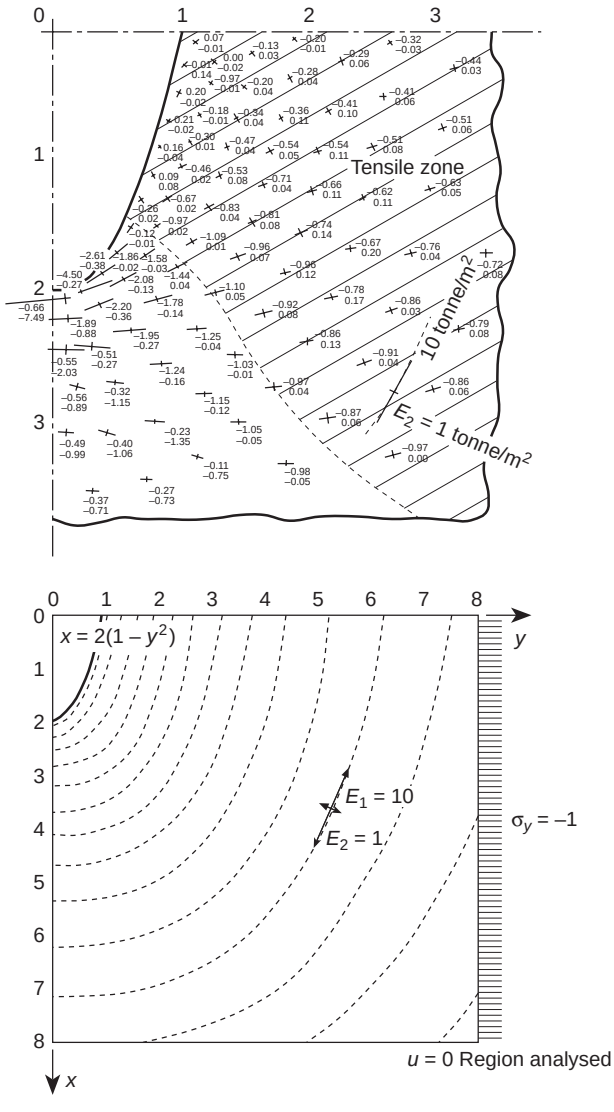


Fig. 4.8 A valley with curved strata subject to a horizontal tectonic stress (plane strain 170 nodes, 298 elements).

as an internal pore pressure. Both solutions indicate large tensile regions, but the increase of stresses due to the second assumption is important.

The stresses calculated here are the so-called 'effective' stresses. These represent the forces transmitted between the solid particles and are defined in terms of the *total* stresses $\boldsymbol{\sigma}$ and the pore pressures p by

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} + \mathbf{m}p \qquad \mathbf{m}^{\mathrm{T}} = [1, 1, 0] \tag{4.44}$$

i.e., simply by removing the hydrostatic pressure component from the *total* stress.[10,13]

The effective stress is of particular importance in the mechanics of porous media such as those that occur in the study of soils, rocks, or concrete. The basic assumption

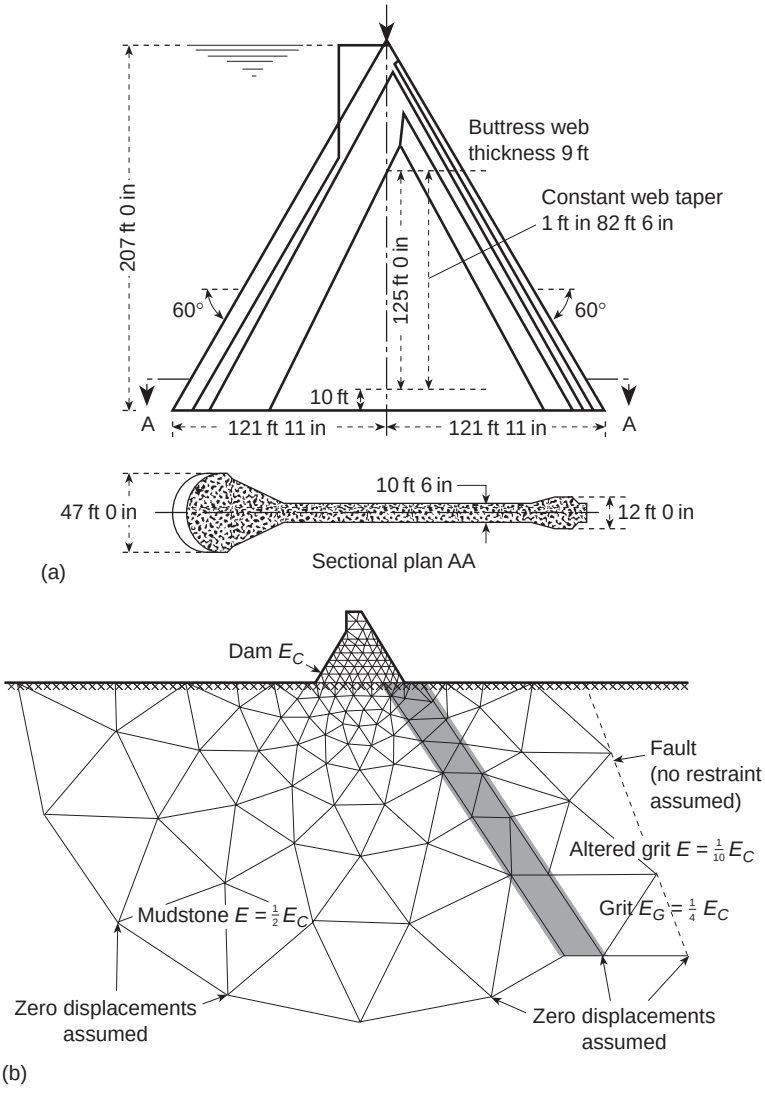


Fig. 4.9 Stress analysis of a buttress dam. A plane stress condition is assumed in the dam and plane strain in the foundation. (a) The buttress section analysed. (b) Extent of foundation considered and division into finite elements.

in deriving the body forces of Eq. (4.43) is that only the effective stress is of any importance in deforming the solid phase. This leads immediately to another possibility of formulation.[14] If we examine the equilibrium conditions of Eq. (2.10) we note that this is written in terms of total stresses. Writing the constitutive relation, Eq. (2.5), in terms of effective stresses, i.e.,

$$\boldsymbol{\sigma}' = \mathbf{D}'(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}'_0 \tag{4.45}$$

and substituting into the equilibrium equation (2.10) we find that Eq. (2.12) is again obtained, with the stiffness matrix using the matrix $\mathbf{D}'$ and the force terms of

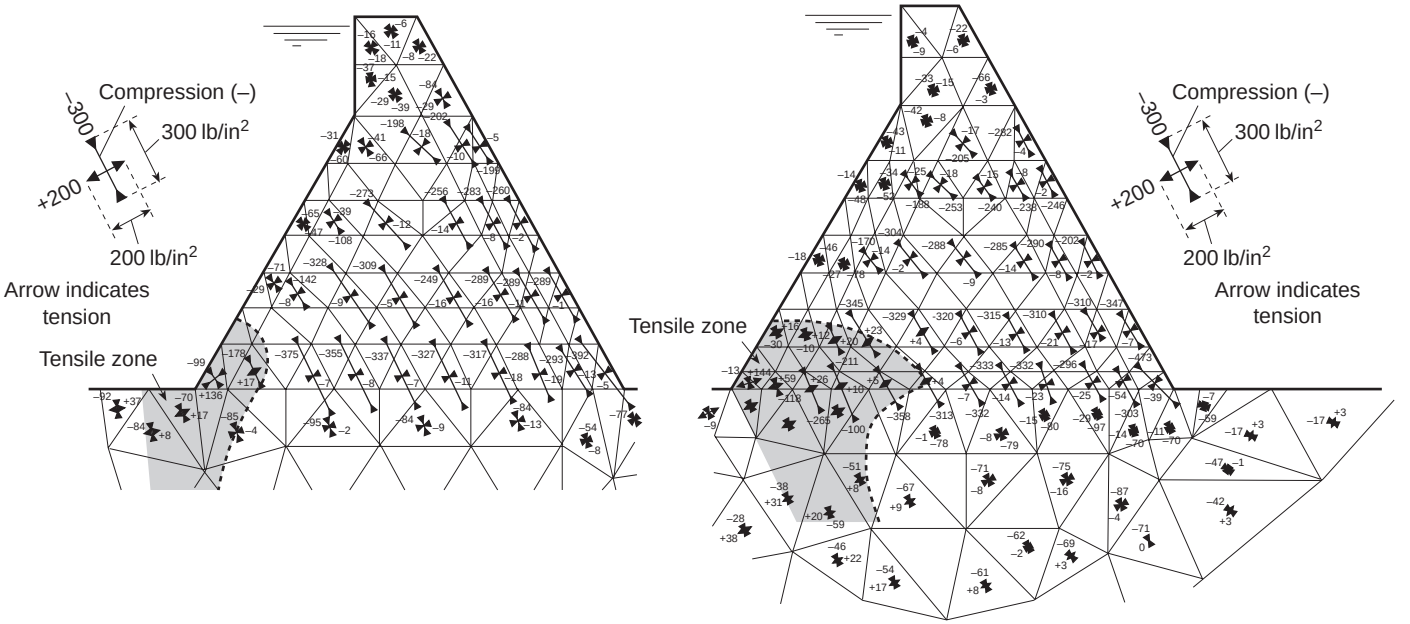


Fig. 4.10 Stress analysis of the buttress dam of Fig. 4.9. Principal stresses for gravity loads are combined with water pressures, which are assumed to act (a) as external loads, (b) as body forces due to pore pressure.

Eq. (2.13b) being augmented by an additional force

$$-\int_{V^e} \mathbf{B}^{\mathrm{T}}\mathbf{m}p\,\mathrm{d(vol)} \tag{4.46}$$

or, if p is interpolated by shape functions N_i', the force becomes

$$-\int_{V^e} \mathbf{B}^{\mathrm{T}}\mathbf{m}\mathbf{N}'\,\mathrm{d(vol)}\bar{\mathbf{p}}^e \tag{4.47}$$

This alternative form of introducing pore pressure effects allows a discontinuous interpolation of p to be used [as in Eq. (4.46) no derivatives occur] and this is now frequently used in practice.

4.4.4 Cracking

The tensile stresses in the previous example will doubtless cause the rock to crack. If a stable situation can develop when such a crack spreads then the dam can be considered safe.

Cracks can be introduced very simply into the analysis by assigning zero elasticity values to chosen elements. An analysis with a wide cracked wedge is shown in Fig. 4.11, where it can be seen that with the extent of the crack assumed no tension within the dam body develops.

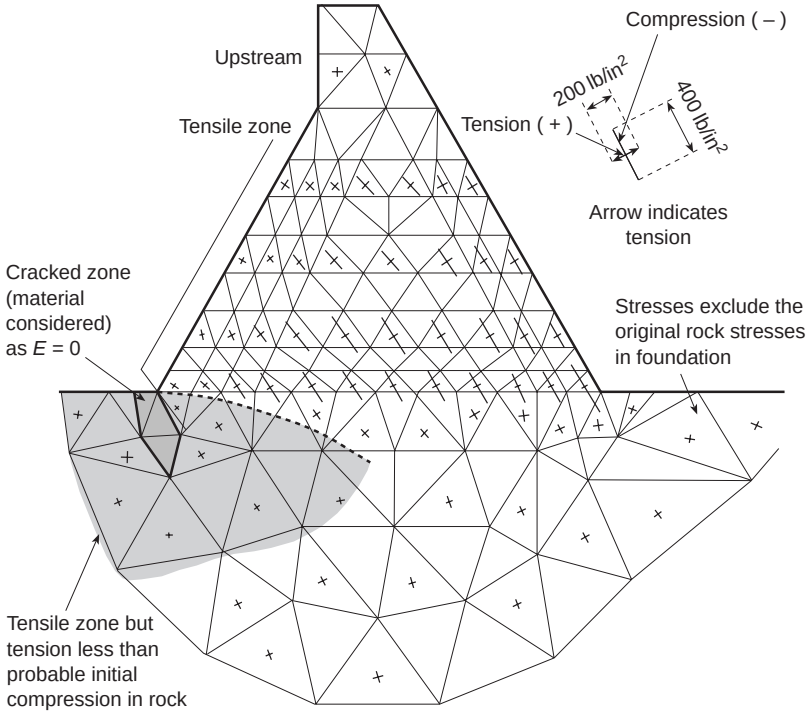


Fig. 4.11 Stresses in a buttress dam. The introduction of a 'crack' modifies the stress distribution [same loading as Fig. 4.10(b)].

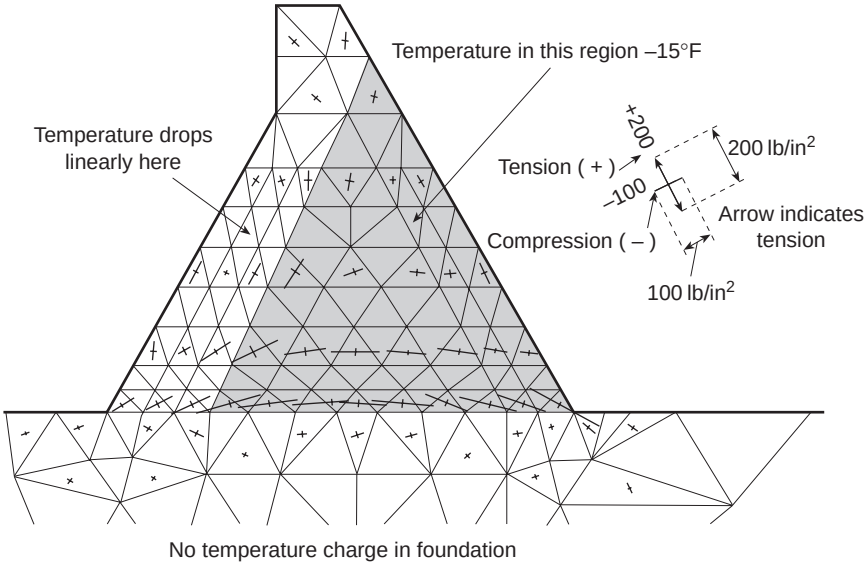


Fig. 4.12 Stress analysis of a buttress dam. Thermal stresses due to cooling of the shaded area by 15 °F ($E = 3 \times 10^6$ lb/in^2, $\alpha = 6 \times 10^{-6}$/°F).

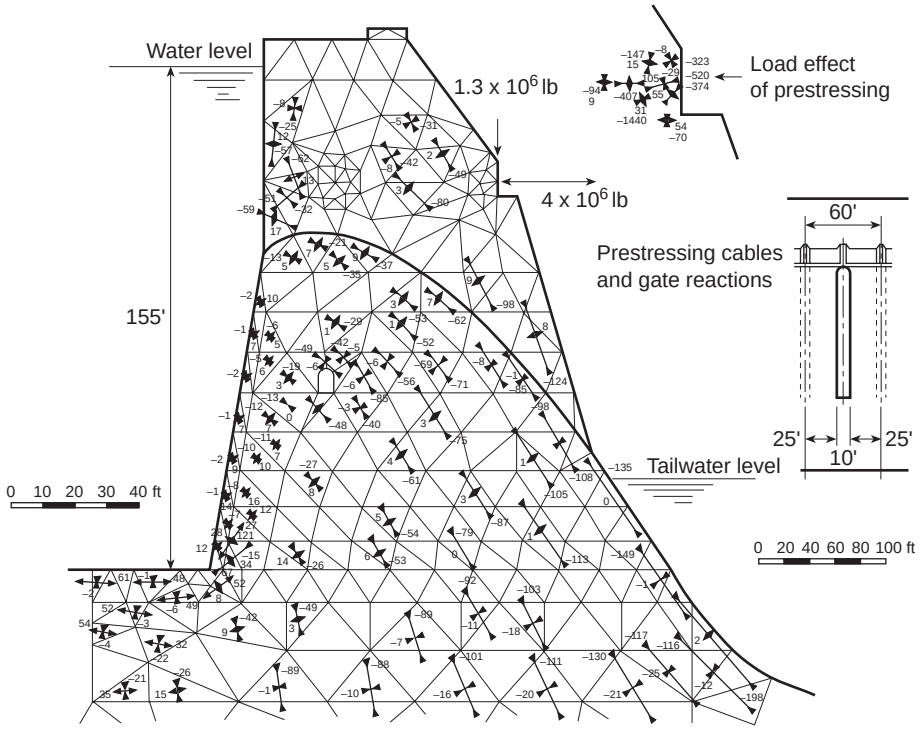


Fig. 4.13 A large barrage with piers and prestressing cables.

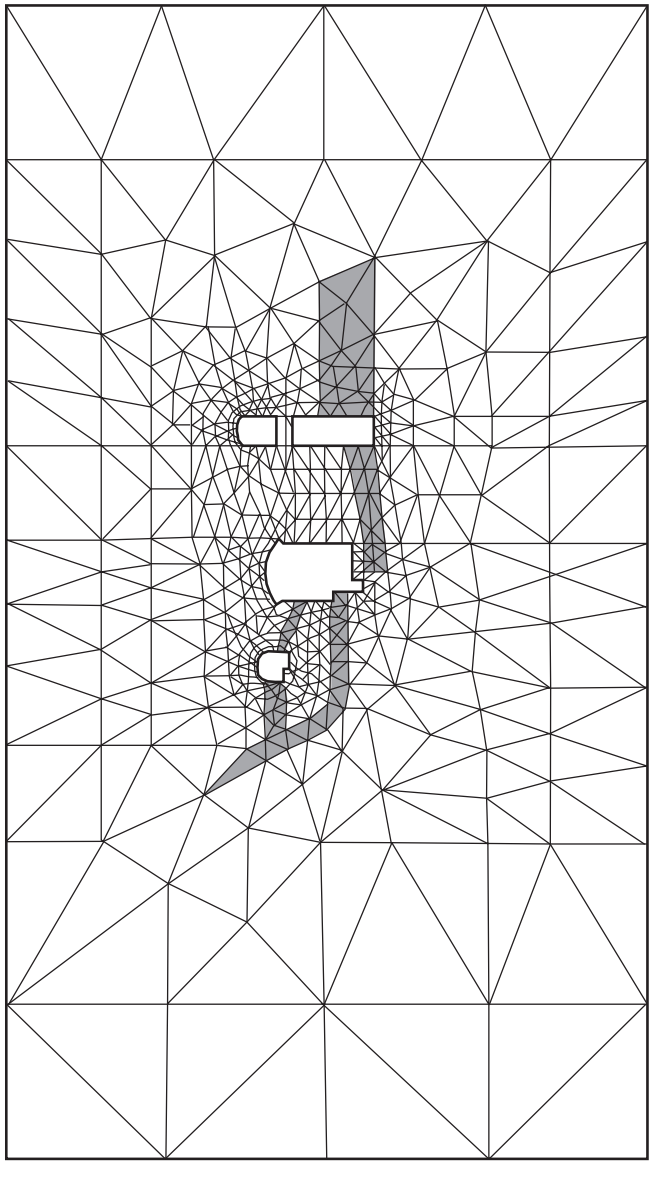


Fig. 4.14 An underground power station. Mesh used in analysis.

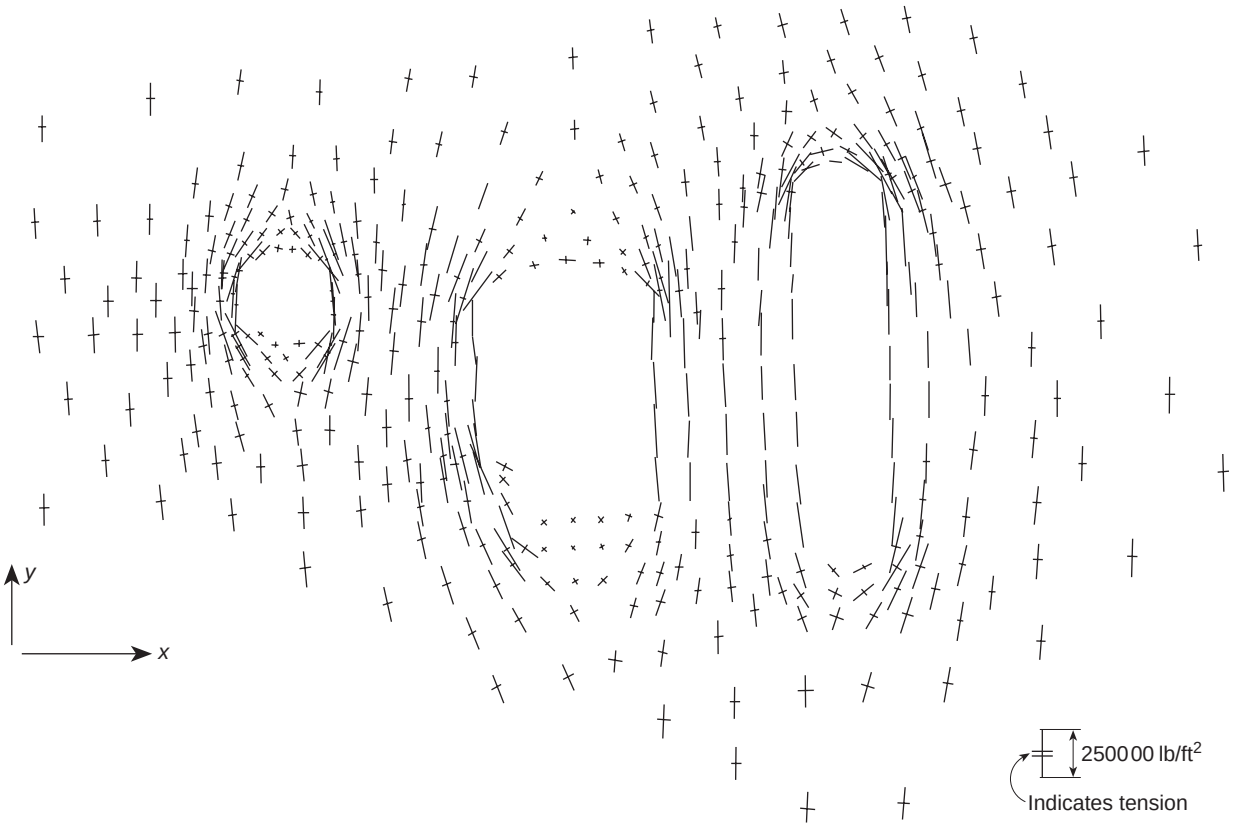


Fig. 4.15 An underground power station. Plot of principal stresses.

A more elaborate procedure for allowing crack propagation and resulting stress redistribution can be developed (see Volume 2).

4.4.5 Thermal stresses

As an example of thermal stress computation the same dam is shown under simple temperature distribution assumptions. Results of this analysis are given in Fig. 4.12.

4.4.6 Gravity dams

A buttress dam is a natural example for the application of finite element methods. Other types, such as gravity dams with or without piers and so on, can also be simply treated. Figure 4.13 shows an analysis of a large dam with piers and crest gates.

In this case the approximation of assuming a two-dimensional treatment in the vicinity of the abrupt change of section, i.e., where the piers join the main body of the dam, is clearly involved, but this leads to localized errors only.

It is important to note here how, in a single solution, the grading of element size is used to study concentration of stress at the cable anchorages, the general stress flow in the dam, and the foundation behaviour. The linear ratio of size of largest to smallest elements is of the order of 30 to 1 (the largest elements occurring in the foundation are not shown in the figure).

4.4.7 Underground power station

This last example, illustrated in Figs 4.14 and 4.15, shows an interesting application. Here principal stresses are plotted automatically. In this analysis many different components of $\boldsymbol{\sigma}_0$, the initial stress, were used due to uncertainty of knowledge about geological conditions. The rapid solution and plot of many results enabled the limits within which stresses vary to be found and an engineering decision arrived at. In this example, the exterior boundaries were taken far enough and 'fixed' ($u = v = 0$). However, a better treatment could be made using infinite elements as described in Sec. 9.13.

4.5 Special treatment of plane strain with an incompressible material

It will have been noted that the relationship (4.16) defining the elasticity matrix $\mathbf{D}$ for an isotropic material breaks down when Poisson's ratio reaches a value of 0.5 as the factor in parentheses becomes infinite. A simple way of side-stepping this difficulty is to use values of Poisson's ratio approximating to 0.5 but not equal to it. Experience shows, however, that if this is done the solution deteriorates unless special formulations such as those discussed in Chapter 12 are used.

4.6 Concluding remark

In subsequent chapters, we shall introduce elements which give much greater accuracy for the same number of degrees of freedom in a particular problem. This has led to the belief that the simple triangle used here is completely superseded. In recent years, however, its very simplicity has led to its revival in practical use in combination with the error estimation and adaptive procedures discussed in Chapters 14 and 15.

References

1. M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *J. Aero. Sci.*, **23**, 805–23, 1956.
2. R.W. Clough. The finite element in plane stress analysis. *Proc. 2nd ASCE Conf. on Electronic Computation*. Pittsburgh, Pa., Sept. 1960.
3. S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. 2nd ed., McGraw-Hill, 1951.
4. S.G. Lekhnitskii. *Theory of Elasticity of an Anisotropic Elastic Body* (Translation from Russian by P. Fern). Holden Day, San Francisco, 1963.
5. R.F.S. Hearmon. *An Introduction to Applied Anisotropic Elasticity*. Oxford University Press, 1961.
6. O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery (SPR) and adaptive finite element refinement. *Comp. Methods Appl. Mech. Eng.*, **101**, 207–24, 1992.
7. B. Boroomand and O.C. Zienkiewicz. Recovery by equilibrium patches (REP). *Internat. J. Num. Meth. Eng.*, **40**, 137–54, 1997.
8. O.C. Zienkiewicz, Y.K. Cheung, and K.G. Stagg. Stresses in anisotropic media with particular reference to problems of rock mechanics. *J. Strain Analysis*, **1**, 172–82, 1966.
9. G.N. Savin. *Stress Concentration Around Holes* (Translation from Russian). Pergamon Press, 1961.
10. O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, B. Schrefler, and T. Shiomi. *Computational Geomechanics*. John Wiley and Sons, Chichester, 1999.
11. O.C. Zienkiewicz and Y.K. Cheung. Buttress dams on complex rock foundations. *Water Power*, **16**, 193, 1964.
12. O.C. Zienkiewicz and Y.K. Cheung. Stresses in buttress dams. *Water Power*, **17**, 69, 1965.
13. K. Terzhagi. *Theoretical Soil Mechanics*. Wiley, 1943.
14. O.C. Zienkiewicz, C. Humpheson, and R.W. Lewis. A unified approach to soil mechanics problems, including plasticity and visco-plasticity. *Int. Symp. on Numerical Methods in Soil and Rock Mechanics*. Karlsruhe, 1975. See also Chapter 4 of *Finite Elements in Geomechanics* (ed. G. Gudehus), pp. 151–78, Wiley, 1977.

5

Axisymmetric stress analysis

5.1 Introduction

The problem of stress distribution in bodies of revolution (axisymmetric solids) under axisymmetric loading is of considerable practical interest. The mathematical problems presented are very similar to those of plane stress and plane strain as, once again, the situation is two dimensional.[1,2] By symmetry, the two components of displacements in any plane section of the body along its axis of symmetry define completely the state of strain and, therefore, the state of stress. Such a cross-section is shown in Fig. 5.1. If r and z denote respectively the radial and axial coordinates of a point, with u and v being the corresponding displacements, it can readily be seen that precisely the same displacement functions as those used in Chapter 4 can be used to define the displacements within the triangular element i, j, m shown.

The volume of material associated with an 'element' is now that of a body of revolution indicated in Fig. 5.1, and all integrations have to be referred to this.

The triangular element is again used mainly for illustrative purposes, the principles developed being completely general.

In plane stress or strain problems it was shown that internal work was associated with three strain components in the coordinate plane, the stress component normal to this plane not being involved due to zero values of either the stress or the strain.

In the axisymmetrical situation any radial displacement automatically induces a strain in the circumferential direction, and as the stresses in this direction are certainly non-zero, this fourth component of strain and of the associated stress has to be considered. Here lies the essential difference in the treatment of the axisymmetric situation.

The reader will find the algebra involved in this chapter somewhat more tedious than that in the previous one but, essentially, identical operations are once again involved, following the general formulation of Chapter 2.

5.2 Element characteristics

5.2.1 Displacement function

Using the triangular shape of element (Fig. 5.1) with the nodes i, j, m numbered in the

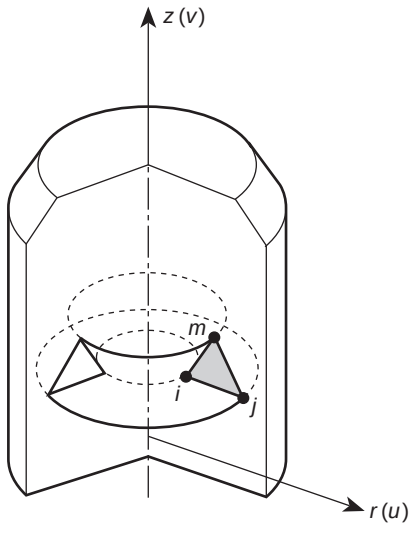


Fig. 5.1 Element of an axisymmetric solid.

anticlockwise sense, we define the nodal displacement by its two components as

$$\mathbf{a}_i = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \tag{5.1}$$

and the element displacements by the vector

$$\mathbf{a}^e = \begin{Bmatrix} \mathbf{a}_i \\ \mathbf{a}_j \\ \mathbf{a}_m \end{Bmatrix} \tag{5.2}$$

Obviously, as in Sec. 4.2.1, a linear polynomial can be used to define uniquely the displacements within the element. As the algebra involved is identical to that of Chapter 4 it will not be repeated here. The displacement field is now given again by Eq. (4.7):

$$\mathbf{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = [\mathbf{I}N_i, \mathbf{I}N_j, \mathbf{I}N_m]\mathbf{a}^e \tag{5.3}$$

with

$$N_i = \frac{a_i + b_i r + c_i z}{2\Delta}, \qquad \text{etc.}$$

and $\mathbf{I}$ a two-by-two identity matrix. In the above

$$\begin{aligned} a_i &= r_j z_m - r_m z_j \\ b_i &= z_j - z_m \\ c_i &= r_m - r_j \end{aligned} \tag{5.4}$$

etc., in cyclic order. Once again Δ is the area of the element triangle.

5.2.2 Strain (total)

As already mentioned, four components of strain have now to be considered. These are, in fact, all the non-zero strain components possible in an axisymmetric deformation. Figure 5.2 illustrates and defines these strains and the associated stresses.

The strain vector defined below lists the strain components involved and defines them in terms of the displacements of a point. The expressions involved are almost self-evident and will not be derived here. The interested reader can consult a standard elasticity textbook[3] for the full derivation. We thus have

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_\theta \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial r} \\ \dfrac{\partial v}{\partial z} \\ \dfrac{u}{r} \\ \dfrac{\partial u}{\partial z} + \dfrac{\partial v}{\partial r} \end{Bmatrix} = \mathbf{S}\mathbf{u} \tag{5.5}$$

Using the displacement functions defined by Eqs (5.3) and (5.4) we have

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{a}^e = [\mathbf{B}_i, \mathbf{B}_j, \mathbf{B}_m]\mathbf{a}^e$$

in which

$$\mathbf{B}_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial r}, & 0 \\ 0, & \dfrac{\partial N_i}{\partial z} \\ \dfrac{N_i}{r}, & 0 \\ \dfrac{\partial N_i}{\partial z}, & \dfrac{\partial N_i}{\partial r} \end{bmatrix} = \begin{bmatrix} b_i, & 0 \\ 0, & c_i \\ \dfrac{a_i}{r} + b_i + \dfrac{c_i z}{r}, & 0 \\ c_i, & b_i \end{bmatrix}, \qquad \text{etc.} \tag{5.6}$$

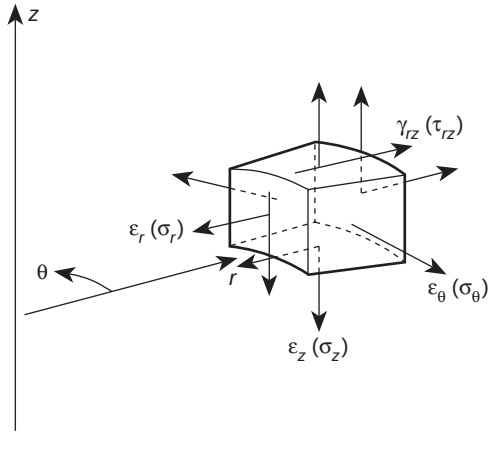


Fig. 5.2 Strains and stresses involved in the analysis of axisymmetric solids.

With the **B** matrix now involving the coordinates r and z, the strains are no longer constant within an element as in the plane stress or strain case. This strain variation is due to the ε_θ term. If the imposed nodal displacements are such that u is proportional to r then indeed the strains will all be constant. In addition, constant ε_z and γ_{rz} strains may be deduced from a linear v displacement. This is the only state of displacement coincident with a constant strain condition and it is clear that the displacement function satisfies the basic criterion of Chapter 2.

5.2.3 Initial strain (thermal strain)

In general, four independent components of the initial strain vector can be envisaged:

$$\boldsymbol{\varepsilon}_0 = \begin{Bmatrix} \varepsilon_{r0} \\ \varepsilon_{z0} \\ \varepsilon_{\theta 0} \\ \gamma_{rz0} \end{Bmatrix} \tag{5.7}$$

Although this can, in general, be variable within the element, it will be convenient to take the initial strain as constant there.

The most frequently encountered case of initial strain will be that due to thermal expansion. For an isotropic material we shall have then

$$\boldsymbol{\varepsilon}_0 = \alpha\theta^e \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{Bmatrix} = \alpha\theta^e \mathbf{m} \tag{5.8}$$

where θ^e is the average temperature rise in an element and α is the coefficient of thermal expansion.

The general case of anisotropy need not be considered since axial symmetry would be impossible to achieve under such circumstances. A case of some interest in practice is that of a 'stratified' material, similar to the one discussed in Chapter 4, in which the plane of isotropy is normal to the axis of symmetry (Fig. 5.3). Here, two different expansion coefficients are possible: one in the axial direction α_z and another in the plane normal to it, α_r.

Now the initial thermal strain becomes

$$\boldsymbol{\varepsilon}_0 = \theta^e \begin{Bmatrix} \alpha_r \\ \alpha_z \\ \alpha_r \\ 0 \end{Bmatrix} \tag{5.9}$$

Practical cases of such 'stratified' anisotropy often arise in laminated or fibreglass construction of machine components.

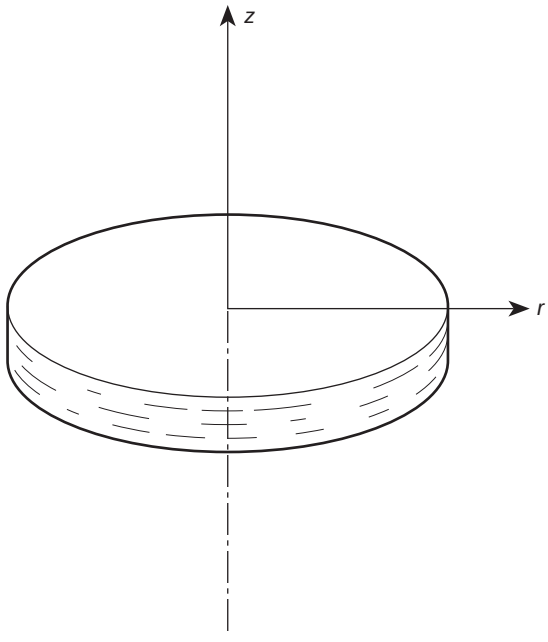


Fig. 5.3 Axisymmetrically stratified material.

5.2.4 Elasticity matrix

The elasticity matrix $\mathbf{D}$ linking the strains $\boldsymbol{\varepsilon}$ and the stresses $\boldsymbol{\sigma}$ in the standard form [Eq. (2.5)],

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_r \\ \sigma_z \\ \sigma_\theta \\ \tau_{rz} \end{Bmatrix} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0$$

needs now to be derived.

The anisotropic 'stratified' material will be considered first, as the isotropic case can be simply presented as a special form.

Anisotropic, stratified, material (Fig. 5.3)

With the z-axis representing the normal to the planes of stratification we can rewrite Eqs (4.19) (again ignoring the initial strains and stresses for convenience) as

$$\begin{aligned} \varepsilon_r &= \frac{\sigma_r}{E_1} - \frac{\nu_2\sigma_z}{E_2} - \frac{\nu_1\sigma_\theta}{E_1} \qquad \varepsilon_z = -\frac{\nu_2\sigma_r}{E_2} + \frac{\sigma_z}{E_2} - \frac{\nu_2\sigma_\theta}{E_2} \\ \varepsilon_\theta &= -\frac{\nu_1\sigma_r}{E_1} - \frac{\nu_2\sigma_z}{E_2} + \frac{\sigma_z}{E_1} \qquad \gamma_{rz} = \frac{\tau_{rz}}{G_2} \end{aligned} \tag{5.10}$$

Writing again

$$\frac{E_1}{E_2} = n; \qquad \frac{G_2}{D_2} = m \qquad \text{and} \qquad d = (1+\nu_1)(1-\nu_1-2n\nu_2^2)$$

we have on solving for the stresses, that

$$\mathbf{D} = \frac{E_2}{d}\begin{bmatrix} n(1-n\nu_2^2), & n\nu_2(1+\nu_1), & n(\nu_1+n\nu_2^2), & 0 \\ & 1-\nu_1^2, & n\nu_2(1+\nu), & 0 \\ & & n(1-n\nu_2^2), & 0 \\ \text{sym.} & & , & md \end{bmatrix} \tag{5.11}$$

Isotropic material

For an isotropic material we can obtain the **D** matrix by taking

$$E_1 = E_2 = E \qquad \text{or} \qquad n = 1$$

and

$$\nu_1 = \nu_2 = \nu$$

and using the well-known relationship between isotropic elastic constants

$$\frac{G_2}{E_2} = \frac{G}{E} = m = \frac{1}{2(1+\nu)}$$

Substituting in Eq. (5.11) we now have

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu, & \nu, & \nu, & 0 \\ \nu, & 1-\nu, & \nu, & 0 \\ \nu, & \nu, & 1-\nu, & 0 \\ 0, & 0, & 0, & (1-2\nu)/2 \end{bmatrix} \tag{5.12}$$

5.2.5 The stiffness matrix

The stiffness matrix of the element *ijm* can now be computed according to the general relationship (2.13). Remembering that the volume integral has to be taken over the whole ring of material we have

$$\mathbf{K}_{ij}^e = 2\pi \int \mathbf{B}_i^{\mathrm{T}} \mathbf{D} \mathbf{B}_j r \, \mathrm{d}r \, \mathrm{d}z \tag{5.13}$$

with **B** given by Eq. (5.6) and **D** by either Eq. (5.11) or Eq. (5.12), depending on the material.

The integration cannot now be performed as simply as was the case in the plane stress problem because the **B** matrix depends on the coordinates. Two possibilities exist: the first is that of numerical integration and the second of an explicit multiplication and term-by-term integration.

The simplest numerical integration procedure is to evaluate all quantities for a centroidal point

$$\bar{r} = \frac{r_i + r_j + r_m}{3} \qquad \text{and} \qquad \bar{z} = \frac{z_i + z_j + z_m}{3}$$

In this case we have simply as a first approximation

$$\mathbf{K}_{ij}^{e} = 2\pi \bar{\mathbf{B}}_i^{\mathrm{T}} \mathbf{D} \bar{\mathbf{B}}_j \bar{r} \Delta \tag{5.14}$$

with Δ being the triangle area and $\bar{\mathbf{B}}$ the value of the strain-displacement matrix at the centroidal point.

More elaborate numerical integration schemes could be used by evaluating the integrand at several points of the triangle. Such methods will be discussed in detail in Chapter 9. However, it can be shown that if the numerical integration is of such an order that the volume of the element is exactly determined by it, then in the limit of subdivision, the solution will converge to the exact answer.[4] The 'one point' integration suggested here is of this type, as it is well known that the volume of a body of revolution is given exactly by the product of the area and the path swept around by its centroid. With the simple triangular element used here a fairly fine subdivision is in any case needed for accuracy and most practical programs use the simple approximation which, surprisingly perhaps, is in fact usually superior to exact integration (see Chapter 10). One reason for this is the occurrence of logarithmic terms in the exact formulation. These involve ratios of the type r_i/r_m and, when the element is at a large distance from the axis, such terms tend to unity and evaluation of the logarithm is inaccurate.

5.2.6 External nodal forces

In the case of the two-dimensional problems of the previous chapter the question of assigning of the external loads was so obvious as not to need further comment. In the present case, however, it is important to realize that the nodal forces represent a combined effect of the force acting along the whole circumference of the circle forming the element 'node'. This point was already brought out in the integration of the expressions for the stiffness of an element, such integrations being conducted over the whole ring.

Thus, if $\bar{R}$ represents the radial component of force per unit length of the circumference of a node at a radius r, the external 'force' which will have to be introduced in the computation is

$$2\pi r \bar{R}$$

In the axial direction we shall, similarly, have

$$2\pi r \bar{Z}$$

to represent the combined effect of axial forces.

5.2.7 Nodal forces due to initial strain

Again, by Eq. (2.13),

$$\mathbf{f}^{e} = -2\pi \int \mathbf{B}^{\mathrm{T}} \mathbf{D} \boldsymbol{\varepsilon}_0 r \, \mathrm{d}r \, \mathrm{d}z \tag{5.15}$$

or noting that $\boldsymbol{\varepsilon}_0$ is constant,

$$\mathbf{f}_i^e = -2\Pi\left(\int \mathbf{B}_i^{\mathrm{T}} r \,\mathrm{d}r \,\mathrm{d}z\right)\mathbf{D}\boldsymbol{\varepsilon}_0 \tag{5.16}$$

The integration should be performed in a similar manner to that used in the determination of the stiffness.

It will readily be seen that, again, an approximate expression using a centroidal value is

$$\mathbf{f}_i^e = -2\pi\bar{\mathbf{B}}_i^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}_0\bar{r}\Delta \tag{5.17}$$

Initial stress forces are treated in an identical manner.

5.2.8 Distributed body forces

Distributed body forces, such as those due to gravity (if acting along the z-axis), centrifugal force in rotating machine parts, or pore pressure, often occur in axisymmetric problems.

Let such forces be denoted by

$$\mathbf{b} = \begin{Bmatrix} b_r \\ b_z \end{Bmatrix} \tag{5.18}$$

per unit volume of material in the directions of r and z respectively. By the general equation (2.13) we have

$$\mathbf{f}_i^e = -2\pi\int \mathbf{I}N_i \begin{Bmatrix} b_r \\ b_z \end{Bmatrix} r \,\mathrm{d}r \,\mathrm{d}z \tag{5.19}$$

Using a coordinate shift similar to that of Sec. 4.2.7 it is easy to show that the first approximation, if the body forces are constant, results in

$$\mathbf{f}_i^e = -2\pi \begin{Bmatrix} b_r \\ b_z \end{Bmatrix} \frac{\bar{r}\Delta}{3} \tag{5.20}$$

Although this is not exact the error term will be found to decrease with reduction of element size and, as it is also self-balancing, it will not introduce inaccuracies. Indeed, as will be shown in Chapter 10, the convergence rate is maintained.

If the body forces are given by a potential similar to that defined in Sec. 4.2.8, i.e.,

$$b_r = -\frac{\partial\phi}{\partial r} \qquad b_z = -\frac{\partial\phi}{\partial z} \tag{5.21}$$

and if this potential is defined linearly by its nodal values, an expression equivalent to Eq. (4.42) can again be determined.

In many problems the body forces vary proportionately to r. For example in rotating machinery we have centrifugal forces

$$b_r = \omega^2\rho r \tag{5.22}$$

where ω is the angular velocity and ρ the density of the material.

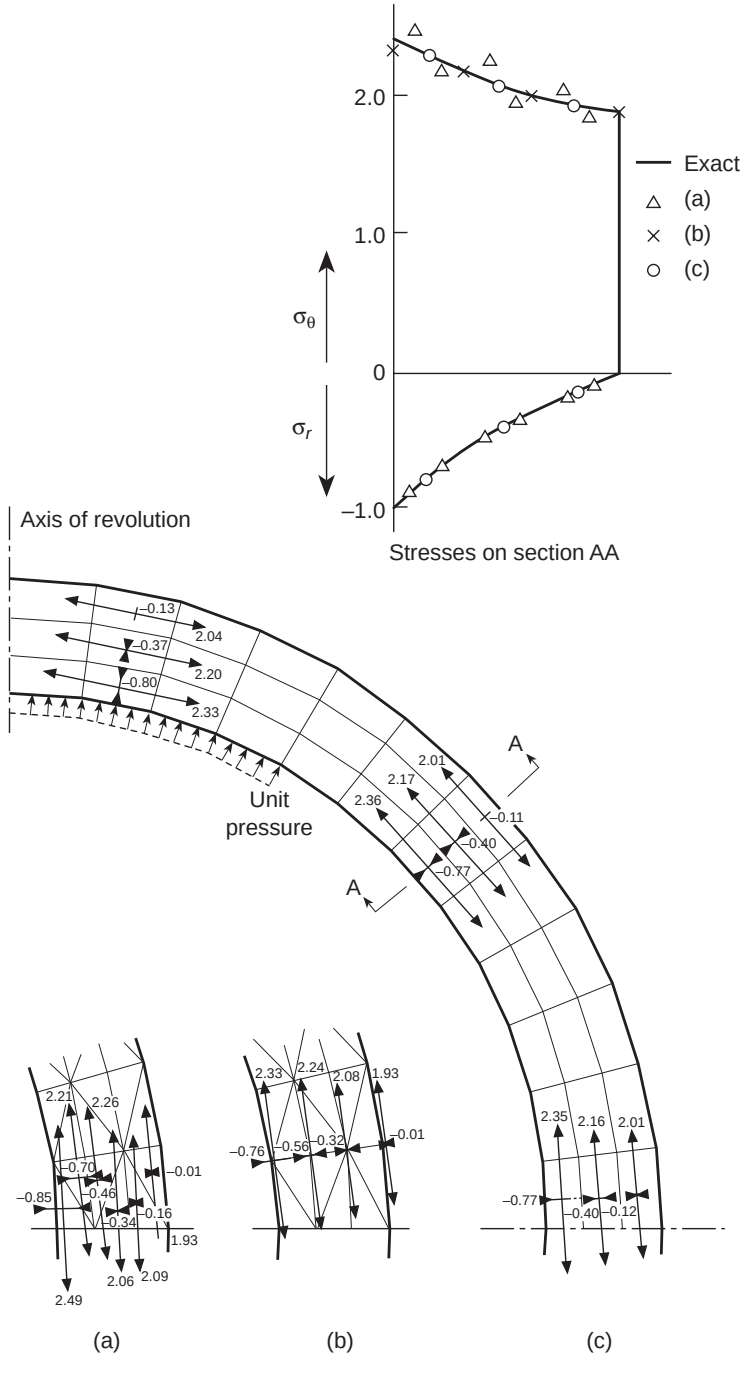


Fig. 5.4 Stresses in a sphere subject to an internal pressure (Poisson's ratio $\nu = 0.3$: (a) triangular mesh – centroidal values; (b) triangular mesh – nodal averages; (c) quadrilateral mesh obtained by averaging adjacent triangles.

5.2.9 Evaluation of stresses

The stresses now vary throughout the element, as will be appreciated from Eqs (4.5) and (4.6). It is convenient now to evaluate the average stress at the centroid of the element. The stress matrix resulting from Eqs (5.6) and (2.3) gives there, as usual,

$$\bar{\boldsymbol{\sigma}}^e = \mathbf{D}\bar{\mathbf{B}}\mathbf{a}^e - \mathbf{D}\boldsymbol{\varepsilon}_0 + \boldsymbol{\sigma}_0 \tag{5.23}$$

It will be found that a certain amount of oscillation of stress values between elements occurs and better approximation can be achieved by averaging nodal stresses or recovery procedures of Chapter 14.

5.3 Some illustrative examples

Test problems such as those of a cylinder under constant axial or radial stress give, as indeed would be expected, solutions which correspond to exact ones. This is again an obvious corollary of the ability of the displacement function to reproduce constant strain conditions.

A problem for which an exact solution is available and in which almost linear stress gradients occur is that of a sphere subject to internal pressure. Figure 5.4(*a*) shows the centroidal stresses obtained using rather a coarse mesh, and the stress oscillation around the exact values should be noted. (This oscillation becomes even more pronounced at larger values of Poisson's ratio although the exact solution is independent of it.) In Fig. 5.4(*b*) the very much better approximation obtained by averaging the stresses at nodal points is shown, and in Fig. 5.4(*c*) a further improvement is given by element averaging. The close agreement with the exact solution even for the very coarse subdivision used here shows the accuracy achievable. The displacements at nodes compared with the exact solution are given in Fig. 5.5.

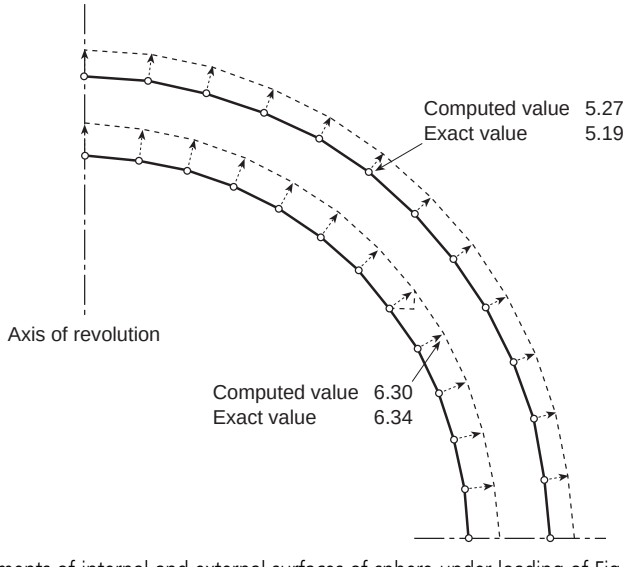


Fig. 5.5 Displacements of internal and external surfaces of sphere under loading of Fig. 5.4.

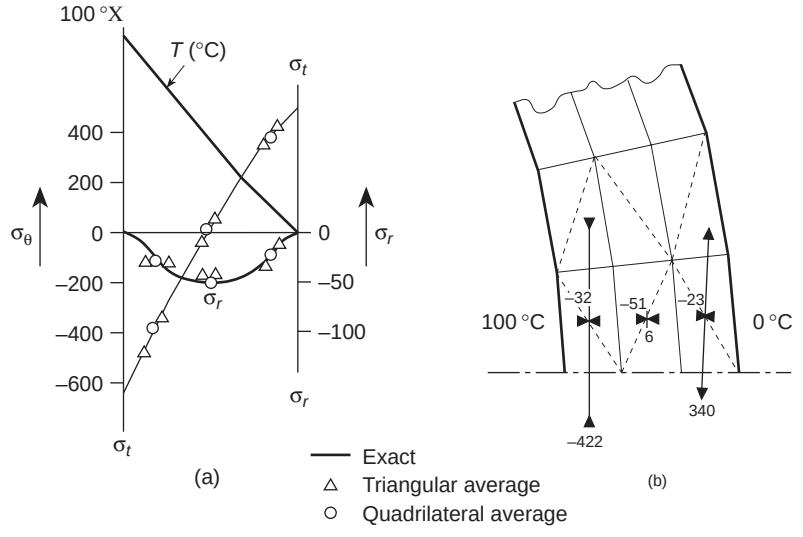


Fig. 5.6 Sphere subject to steady-state heat flow (100 °C internal temperature, 0 °C external temperature): (a) temperature and stress variation on radial section; (b) 'quadrilateral' averages.

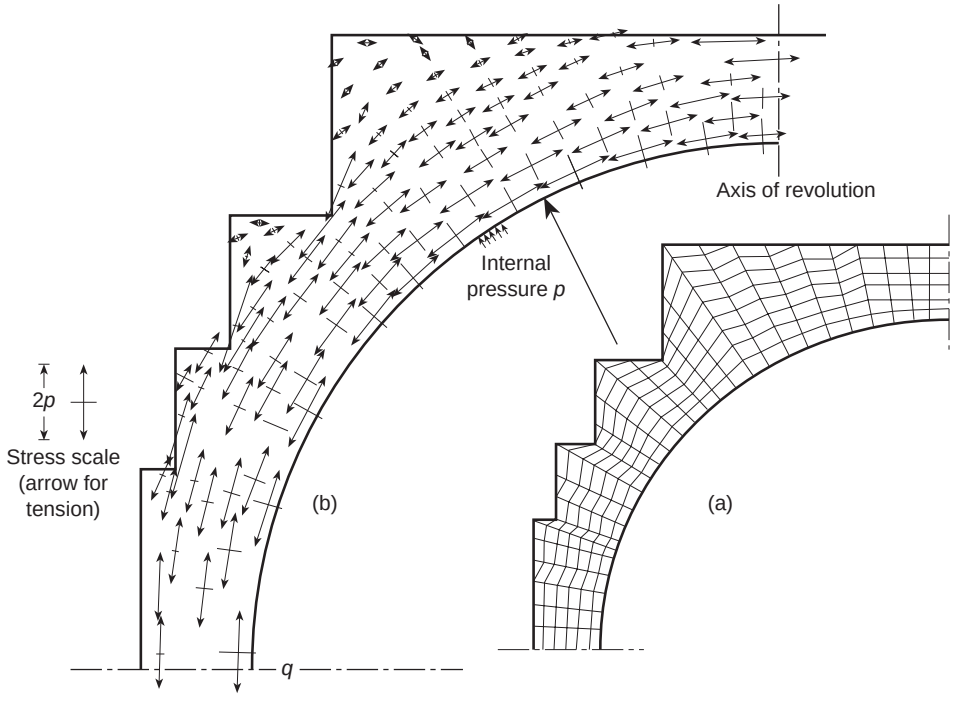


Fig. 5.7 A reactor pressure vessel. (a) 'Quadrilateral' mesh used in analysis; this was generated automatically by a computer. (b) Stresses due to a uniform internal pressure (automatic computer plot). Solution based on quadrilateral averages. (Poisson's ratio $\nu = 0.15$).

In Fig. 5.6 thermal stresses in the same sphere are computed for the steady-state temperature variation shown. Again, excellent accuracy is demonstrated by comparison with the exact solution.

5.4 Early practical applications

Two examples of practical applications of the programs available for axisymmetrical stress distribution are given here.

5.4.1 A prestressed concrete reactor pressure vessel

Figure 5.7 shows the stress distribution in a relatively simple prototype pressure vessel. Due to symmetry only one-half of the vessel is analysed, the results given here referring to the components of stress due to internal pressure.

In Fig. 5.8 contours of equal major principal stresses caused by temperature are shown. The thermal state is due to steady-state heat conduction and was itself found by the finite element method in a way described in Chapter 7.

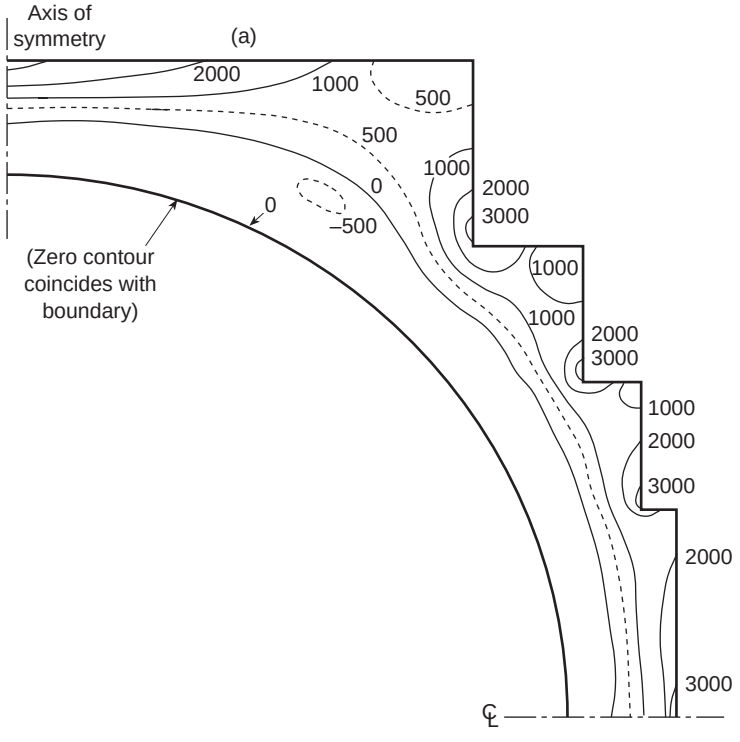


Fig. 5.8 A reactor pressure vessel. Thermal stresses due to steady-state heat conduction. Contours of major principal stress in pounds per square inch. (Interior temperature 400 °C, exterior temperature 0 °C, $\alpha = 5 \times 10^{-6}/°C$. $E = 2.58 \times 10^{6}$ lb/in^2, $\nu = 0.15$).

5.4.2 Foundation pile

Figure 5.9 shows the stress distribution around a foundation pile penetrating two different strata. This non-homogeneous problem presents no difficulties and is treated by the standard approach given in this chapter in which the 'quadrilateral' elements shown are assemblies of two triangles and the results are averaged.

5.5 Non-symmetrical loading

The method described in the present chapter can be extended to deal with non-symmetrical loading. If the circumferential loading variation is expressed in circular harmonics then it is still possible to focus attention on one axial section although the nodal degrees of freedom are now increased to three.

Details of this process are described in references 5 and 6 and in Chapter 9 of Volume 2.

5.6 Axisymmetry – plane strain and plane stress

In the previous chapter we noted that plane stress and strain analysis was done in terms of three stress and strain components and, indeed, both cases could be generally incorporated in a single program with an indicator changing appropriate constants in the matrix **D**. Doing this loses track of the σ_z component in the plane strain case which has to be separately evaluated. Further, special expressions [viz. Eq. 4.28] had to be used to introduce initial strains. This is inconvenient (especially when non-linear constitutive laws are used), and an alternative of writing the plane strain case in terms of four stress–strain components as a special case of axisymmetric analysis is highly recommended.

If the axisymmetric strain definition of Eq. (5.5) is examined, we note that $r = \infty$ gives $\varepsilon_\theta \equiv 0$ and plane strain conditions are obtained. Thus, if we ignore the terms in **B** associated with ε_θ, replace the coordinates

$$r \text{ and } z \qquad \text{by} \qquad x \text{ and } y$$

and further change the volume of integration

$$2\pi r \qquad \text{to} \qquad 1$$

the plane strain formulation becomes available from the axisymmetric plane strain directly.

Plane stress conditions can similarly be incorporated, requiring in addition substitution of the axisymmetric **D** matrix by Eqs (4.13) or (4.19) augmented by an appropriate zero row and column. Thus, at the cost of additional storage of the fourth stress and strain component, all the cases discussed can be incorporated in a single format.

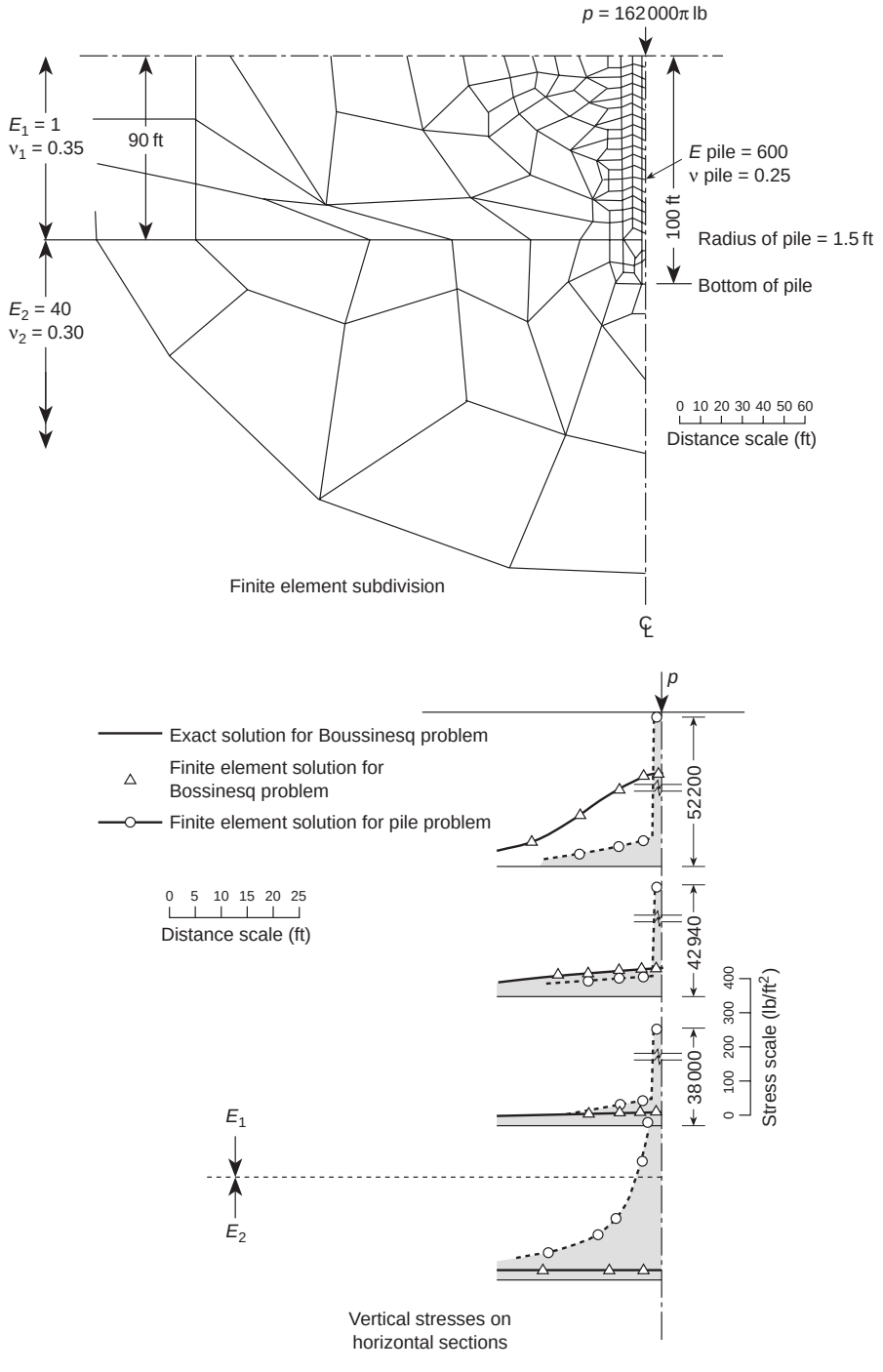


Fig. 5.9 (a) A pile in stratified soil. Irregular mesh and data for the problem. (b) A pile in stratified soil. Plot of vertical stresses on horizontal sections. Solution also plotted for Boussinesq problem obtained by making $E_1 = E_2 = E_{\text{pile}}$, and this is compared with exact values.

References

1. R.W. Clough. Chapter 7 of *Stress Analysis* (eds O.C. Zienkiewicz and G.S. Holister), Wiley, 1965.
2. R.W. Clough and Y.R. Rashid. Finite element analysis of axi-symmetric solids. *Proc. ASCE*, **91**, EM.1, 71, 1965.
3. S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. 2nd ed., McGraw-Hill, 1951.
4. B.M. Irons. Comment on 'Stiffness matrices for section element' by I.R. Raju and A.K. Rao. *JAIAA*, **7**, 156–7, 1969.
5. E.L. Wilson. Structural analysis of axisymmetric solids. *JAIAA*, **3**, 2269–74, 1965.
6. O.C. Zienkiewicz. *The Finite Element Method*. 3rd ed., McGraw-Hill, 1977.

6

Three-dimensional stress analysis

6.1 Introduction

It will have become obvious to the reader by this stage of the book that there is but one further step to apply the general finite element procedure to fully three-dimensional problems of stress analysis. Such problems embrace clearly all the practical cases, though for some, the various two-dimensional approximations give an adequate and more economical 'model'.

The simplest two-dimensional continuum element is a triangle. In three dimensions its equivalent is a tetrahedron, an element with four nodal corners†, and this chapter will deal with the basic formulation of such an element. Immediately, a difficulty not encountered previously is presented. It is one of ordering of the nodal numbers and, in fact, of a suitable representation of a body divided into such elements.

The first suggestions for the use of the simple tetrahedral element appear to be those of Gallagher *et al.*[1] and Melosh.[2] Argyris[3,4] elaborated further on the theme and Rashid and Rockenhauser[5] were the first to apply three-dimensional analysis to realistic problems.

It is immediately obvious, however, that the number of simple tetrahedral elements which has to be used to achieve a given degree of accuracy has to be very large. This will result in very large numbers of simultaneous equations in practical problems, which may place a severe limitation on the use of the method in practice. Further, the bandwidth of the resulting equation system becomes large, leading to increased use of iterative solution methods.

To realize the order of magnitude of the problems presented let us assume that the accuracy of a triangle in two-dimensional analysis is comparable to that of a tetrahedron in three dimensions. If an adequate stress analysis of a square, two-dimensional region requires a mesh of some $20 \times 20 = 400$ nodes, the total number of simultaneous equations is around 800 given two displacement variables at a node. (This is a fairly realistic figure.) The bandwidth of the matrix involves 20 nodes (Chapter 20), i.e., some 40 variables.

† The simplest polygonal shape which permits the approximation of the domain is known as the *simplex*. Thus a triangular and tetrahedral element constitute the simplex in two and three dimensions, respectively.

An equivalent three-dimensional region is that of a cube with $20 \times 20 \times 20 = 8000$ nodes. The total number of simultaneous equations is now some 24 000 as three displacement variables have to be specified. Further, the bandwidth now involves an interconnection of some $20 \times 20 = 400$ nodes or 1200 variables.

Given that with direct solution techniques the computation effort is roughly proportional to the number of equations and to the square of the bandwidth, the magnitude of the problems can be appreciated. It is not surprising therefore that efforts to improve accuracy by use of complex elements with many degrees of freedom have been strongest in the area of three-dimensional analysis.[6–10] The development and practical application of such elements will be described in the following chapters. However, the presentation of this chapter gives all the necessary ingredients of the formulation for three-dimensional elastic problems and so follows directly from the previous ones. Extension to more elaborate elements will be self-evident.

6.2 Tetrahedral element characteristics

6.2.1 Displacement functions

Figure 6.1 illustrates a tetrahedral element i, j, m, p in space defined by x, y, and z coordinates.

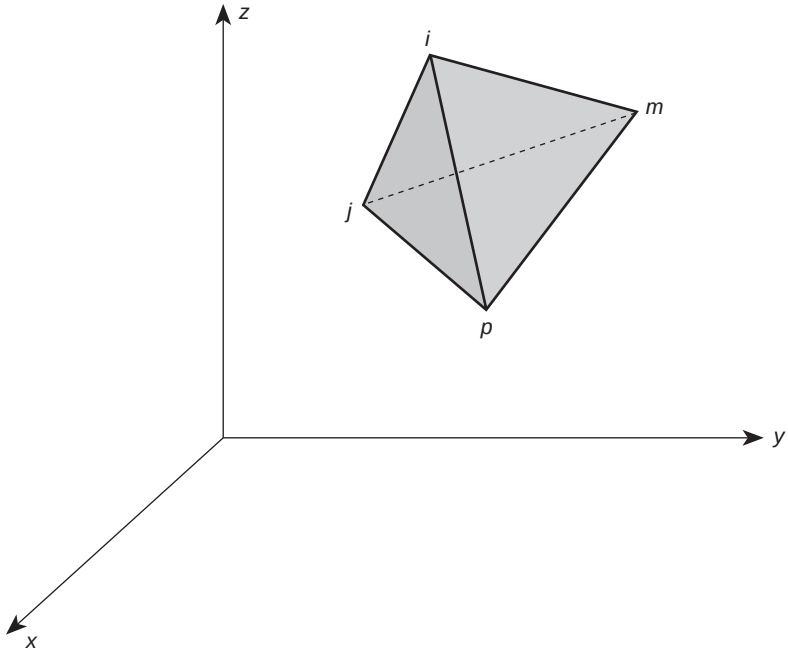


Fig. 6.1 A tetrahedral volume. (Always use a consistent order of numbering, e.g., for *p* count the other nodes in an anticlockwise order as viewed from *p*, giving the element as *ijmp*, etc.).

The state of displacement of a point is defined by three displacement components, u, v, and w, in the directions of the three coordinates x, y, and z. Thus

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{6.1}$$

Just as in a plane triangle where a linear variation of a quantity was defined by its three nodal values, here a linear variation will be defined by the four nodal values. In analogy to Eq. (4.3) we can write, for instance,

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \tag{6.2}$$

Equating the values of the displacement at the nodes we have four equations of the type

$$u_i = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 z_i, \qquad \text{etc.} \tag{6.3}$$

from which α_1 to α_4 can be evaluated.

Again, it is possible to write this solution in a form similar to that of Eq. (4.5) by using a determinant form, i.e.,

$$\begin{aligned} u = \frac{1}{6V}[&(a_i + b_i x + c_i y + d_i z)u_i + (a_j + b_j x + c_j y + d_j z)u_j \\ &+ (a_m + b_m x + c_m y + d_m z)u_m + (a_p + b_p x + c_p y + d_p z)u_p] \end{aligned} \tag{6.4}$$

with

$$6V = \det \begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_m & y_m & z_m \\ 1 & x_p & y_p & z_p \end{vmatrix} \tag{6.5a}$$

in which, incidentally, the value V represents the volume of the tetrahedron. By expanding the other relevant determinants into their cofactors we have

$$\begin{aligned} a_i &= \det \begin{vmatrix} x_j & y_j & z_j \\ x_m & y_m & z_m \\ x_p & y_p & z_p \end{vmatrix} & b_i &= -\det \begin{vmatrix} 1 & y_j & z_j \\ 1 & y_m & z_m \\ 1 & y_p & z_p \end{vmatrix} \\ c_i &= -\det \begin{vmatrix} x_j & 1 & z_j \\ x_m & 1 & z_m \\ x_p & 1 & z_p \end{vmatrix} & d_i &= -\det \begin{vmatrix} x_j & y_j & 1 \\ x_m & y_m & 1 \\ x_p & y_p & 1 \end{vmatrix} \end{aligned} \tag{6.5b}$$

with the other constants defined by cyclic interchange of the subscripts in the order i, j, m, p.

The ordering of nodal numbers i, j, m, p must follow a 'right-hand' rule obvious from Fig. 6.1. In this the first three nodes are numbered in an anticlockwise manner when viewed from the last one.

The element displacement is defined by the 12 displacement components of the nodes as

$$\mathbf{a}^e = \begin{Bmatrix} \mathbf{a}_i \\ \mathbf{a}_j \\ \mathbf{a}_m \\ \mathbf{a}_p \end{Bmatrix} \tag{6.6}$$

with

$$\mathbf{a}_i = \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \qquad \text{etc.}$$

We can write the displacements of an arbitrary point as

$$\mathbf{u} = [\mathbf{I}N_i, \mathbf{I}N_j, \mathbf{I}N_m, \mathbf{I}N_p]\mathbf{a}^e = \mathbf{N}\mathbf{a}^e \tag{6.7}$$

with shape functions defined as

$$N_i = \frac{a_i + b_i x + c_i y + d_i z}{6V}, \qquad \text{etc.} \tag{6.8}$$

and $\mathbf{I}$ being a three by three identity matrix.

Once again the displacement functions used will obviously satisfy continuity requirements on interfaces between various elements. This fact is a direct corollary of the linear nature of the variation of displacement.

6.2.2 Strain matrix

Six strain components are relevant in full three-dimensional analysis. The strain matrix can now be defined as

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial w}{\partial z} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial z} + \dfrac{\partial w}{\partial y} \\ \dfrac{\partial w}{\partial x} + \dfrac{\partial u}{\partial z} \end{Bmatrix} = \mathbf{S}\mathbf{u} \tag{6.9}$$

following the standard notation of Timoshenko's elasticity text.[11] Using Eqs (6.4)–(6.8) it is an easy matter to verify that

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{N}\mathbf{a}^e = \mathbf{B}\mathbf{a}^e = [\mathbf{B}_i, \mathbf{B}_j, \mathbf{B}_m, \mathbf{B}_p]\mathbf{a}^e \tag{6.10}$$

in which

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x}, & 0, & 0 \\ 0, & \frac{\partial N_i}{\partial y}, & 0 \\ 0, & 0, & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y}, & \frac{\partial N_i}{\partial x}, & 0 \\ 0, & \frac{\partial N_i}{\partial z}, & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z}, & 0, & \frac{\partial N_i}{\partial x} \end{bmatrix} = \frac{1}{6V} \begin{bmatrix} b_i, & 0, & 0 \\ 0, & c_i, & 0 \\ 0, & 0, & d_i \\ c_i, & b_i, & 0 \\ 0, & d_i, & c_i \\ d_i & 0, & b_i \end{bmatrix} \tag{6.11}$$

with other submatrices obtained in a similar manner simply by interchange of subscripts.

Initial strains, such as those due to thermal expansion, can be written in the usual way as a six-component vector which, for example, in an isotropic thermal expansion is simply

$$\boldsymbol{\varepsilon}_0 = \alpha\theta^e \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \alpha\theta^e \mathbf{m} \tag{6.12}$$

with α being the expansion coefficient and θ^e the average element temperature rise.

6.2.3 Elasticity matrix

With complete anisotropy the **D** matrix relating the six stress components to the strain components can contain 21 independent constants (see Sec. 4.2.3).

In general, thus,

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \tag{6.13}$$

Although no difficulty presents itself in computation when dealing with such materials, it is convenient to recapitulate here the **D** matrix for an isotropic material. This, in terms of the usual elastic constants E (modulus) and ν (Poisson's ratio),

can be written as

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu, & \nu, & \nu, & 0, & 0, & 0 \\ & 1-\nu, & \nu, & 0, & 0, & 0 \\ & & 1-\nu, & 0, & 0, & 0 \\ & & & (1-2\nu)/2, & 0, & 0 \\ & \text{Sym.} & & & (1-2\nu)/2, & 0 \\ & & & & & (1-2\nu)/2 \end{bmatrix} \tag{6.14}$$

6.2.4 Stiffness, stress, and load matrices

The stiffness matrix defined by the general relationship (2.10) can now be explicitly integrated since the strain and stress components are constant within the element.

The general ij submatrix of the stiffness matrix will be a three by three matrix defined as

$$\mathbf{K}_{ij}^e = \mathbf{B}_i^{\mathrm{T}} \mathbf{D} \mathbf{B}_j V^e \tag{6.15}$$

where V^e represents the volume of the elementary tetrahedron.

The nodal forces due to the initial strain become, similarly to Eq. (4.34),

$$\mathbf{f}_i^e = -\mathbf{B}_i^{\mathrm{T}} \mathbf{D} \boldsymbol{\varepsilon}_0 V^e \tag{6.16}$$

with a similar expression for forces due to initial stresses.

Distributed body forces can once again be expressed in terms of their b_x, b_y, and b_z components or in terms of the body force potential. Not surprisingly, it will once more be found that if the body forces are constant the nodal components of the total resultant are distributed in four equal parts [see Eq. (4.36)].

In fact, the similarity with the expressions and results of Chapter 4 is such that further explicit formulation is unnecessary. The reader will find no difficulty in repeating the various steps needed for the formulation of a computer program.

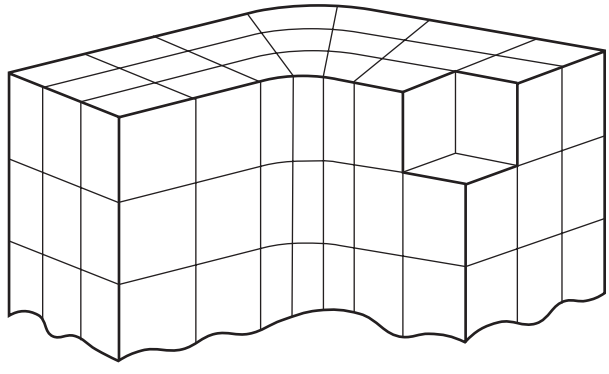


Fig. 6.2 A systematic way of dividing a three-dimensional object into 'brick'-type elements.

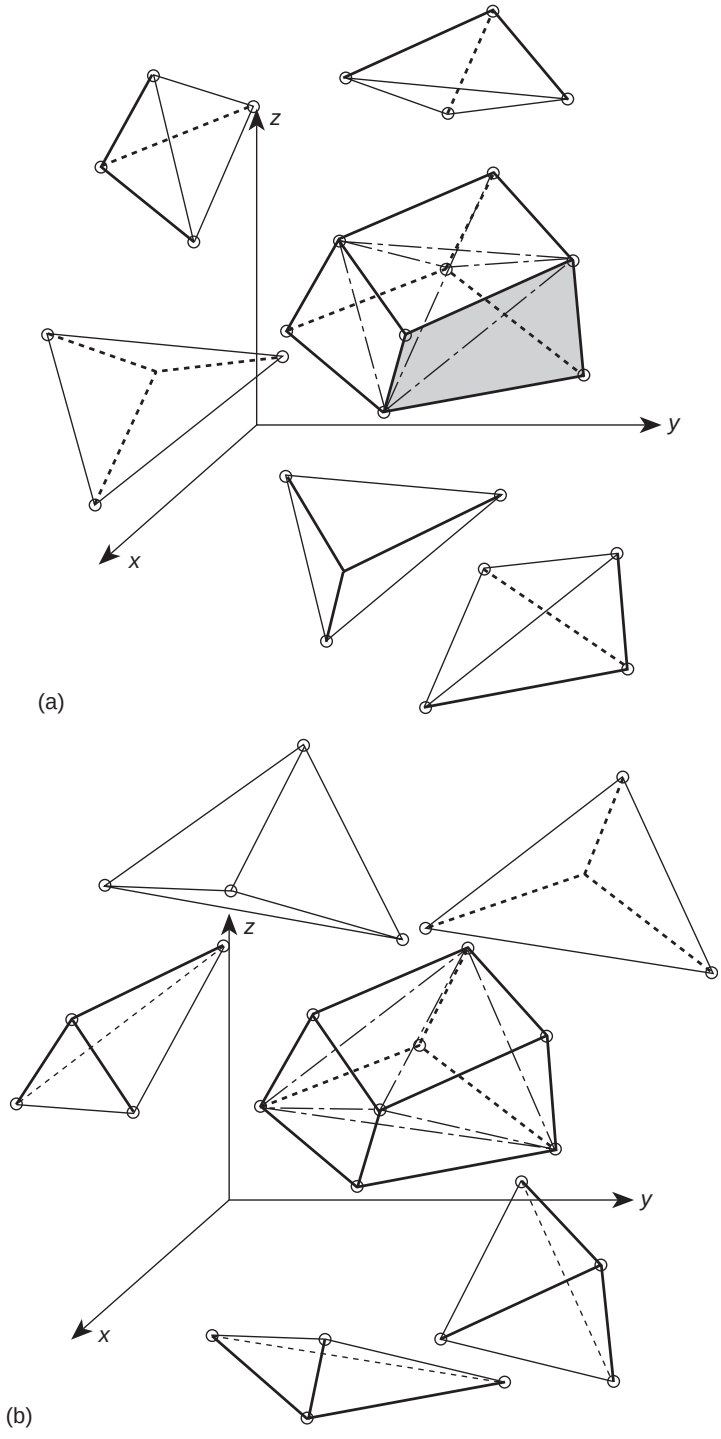


Fig. 6.3 Composite element with eight nodes and its subdivision into five tetrahedra by alternatives (a) or (b).

6.3 Composite elements with eight nodes

The division of a space volume into individual tetrahedra sometimes presents difficulties of visualization and could easily lead to errors in nodal numbering, etc., unless a fully automatic code is available. A more convenient subdivision of space is into eight-cornered brick elements (bricks being the natural way to build a universe!). By sectioning a three-dimensional body parallel sections can be drawn and, each one being subdivided into quadrilaterals, a systematic way of element definition could be devised as in Fig. 6.2.

Such elements could be assembled automatically from several tetrahedra and the process of creating these tetrahedra left to a simple logical program. For instance, Fig. 6.3 shows how a typical brick can be divided into five tetrahedra in two (and only two) distinct ways. Stresses could well be presented as averages for a whole brick-like element or as final nodal averages. We shall discuss again a rational procedure for stress recovery in Chapter 14.

In Fig. 6.4 a more convenient subdivision of a brick into six tetrahedra is shown. Here obviously the number of alternatives is very great; however (contrary to the

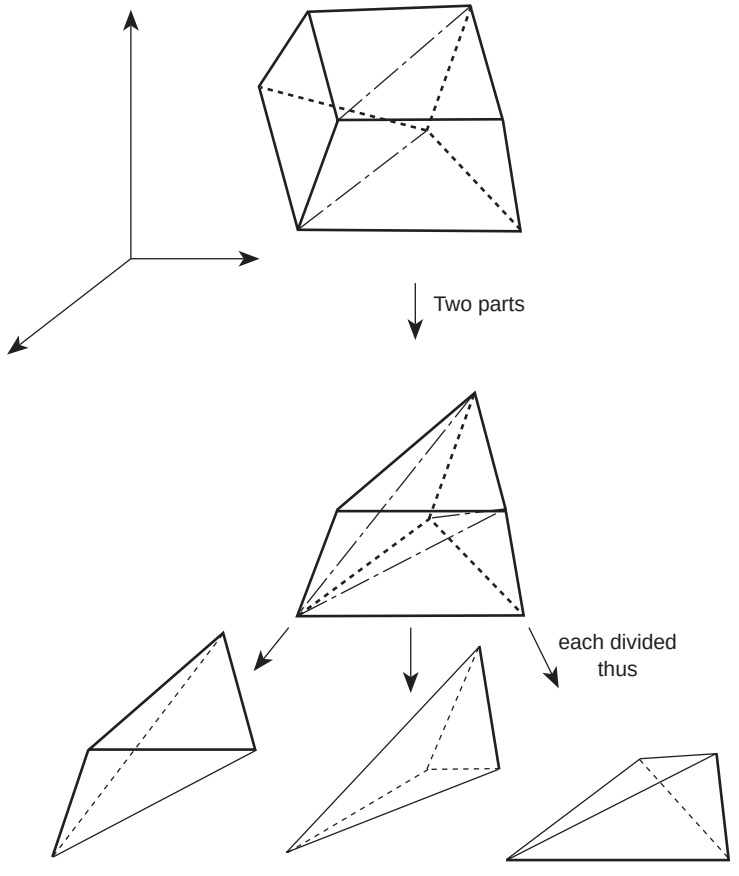


Fig. 6.4 A systematic way of splitting an eight-cornered brick into six tetrahedra.

5-element subdivision) diagonals on adjacent faces of elements for a mesh type shown in Fig. 6.2 can always be made to match. Thus the 6-element subdivision creates a conforming approximation.

In later chapters it will be seen how the basic bricks can be obtained directly with more complex types of shape function.

6.4 Examples and concluding remarks

A simple, illustrative example of the application of simple, tetrahedral, elements is shown in Figs 6.5 and 6.6. Here the well-known Boussinesq problem of an elastic half-space with a point load is approximated by analysing a cubic volume of space. Use of symmetry is made to reduce the size of the problem and the boundary displacements are prescribed in a manner shown in Fig. 6.5.[12] As zero displacements were prescribed at a finite distance below the load a correction obtained from the exact expression was applied before executing the plots shown in Fig. 6.6. Comparison of both stresses and displacement appears reasonable although it will be appreciated that the division is very coarse. However, even this trivial problem involved the solution of some 375 equations. More ambitious problems treated with simple tetrahedra are given in references 5 and 12. Figure 6.7, taken from the former, illustrates an analysis of a complex pressure vessel. Some 10 000 degrees of freedom are involved in this analysis. In Chapter 8 it will be seen how the use of complex elements permits a sufficiently accurate analysis to be performed with a much smaller total number of degrees of freedom for a very similar problem.

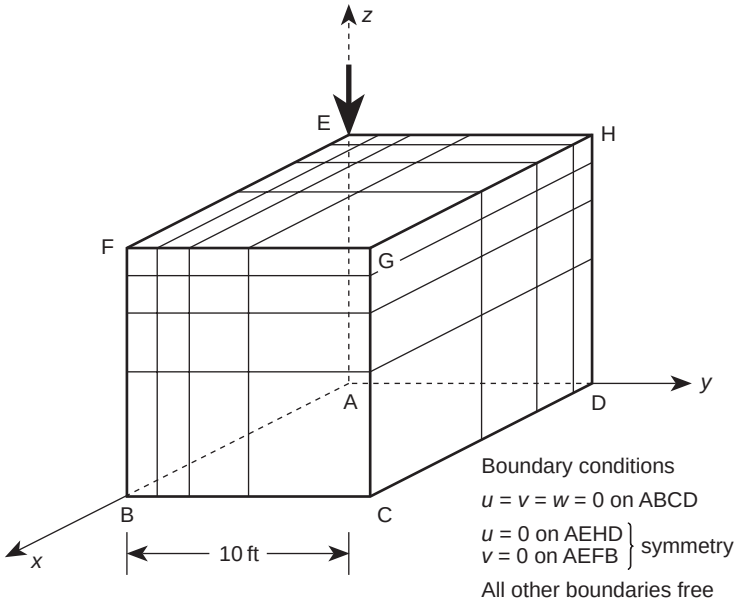


Fig. 6.5 The Boussinesq problem as one of three-dimensional stress analysis.

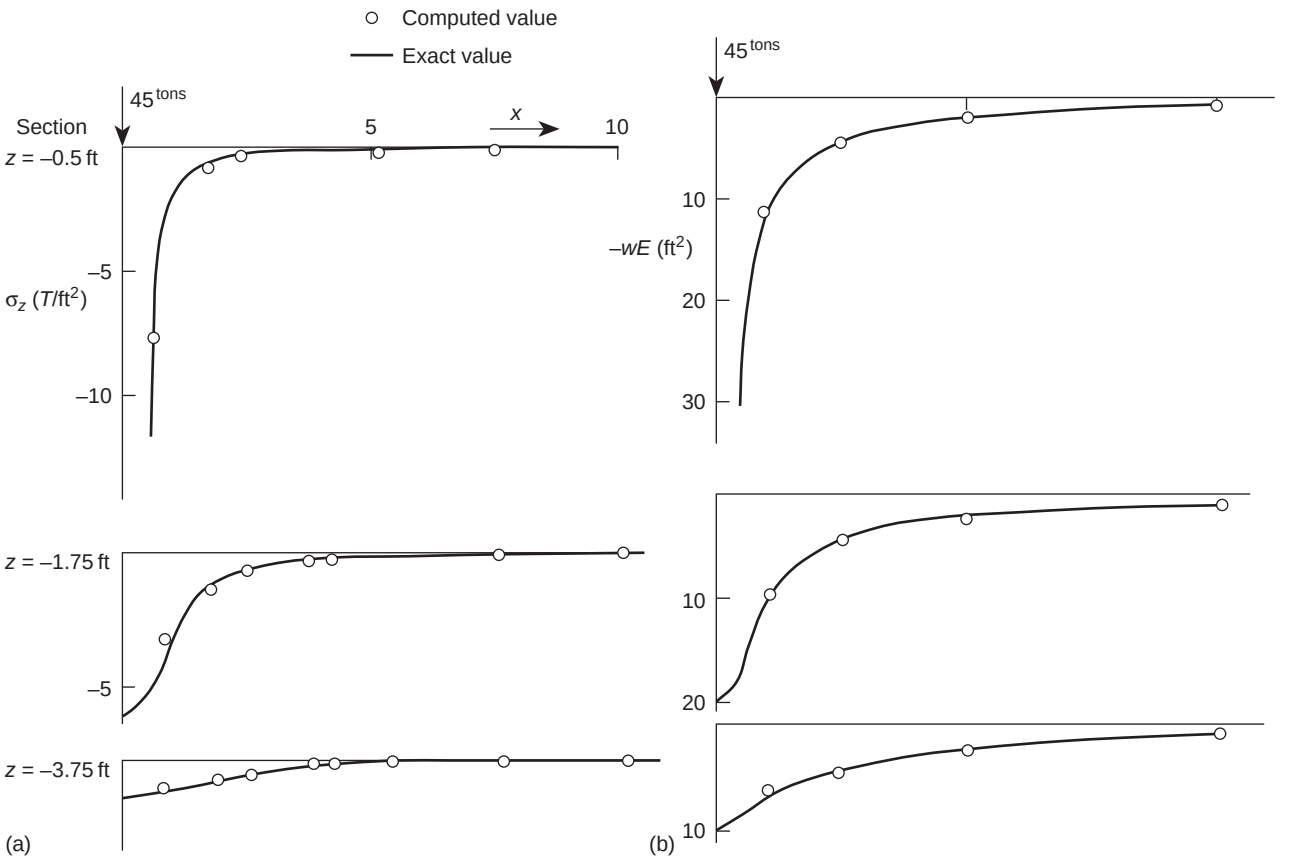


Fig. 6.6 The Boussineq problem: (*a*) vertical stresses (σ_z); (*b*) vertical displacements (*w*).

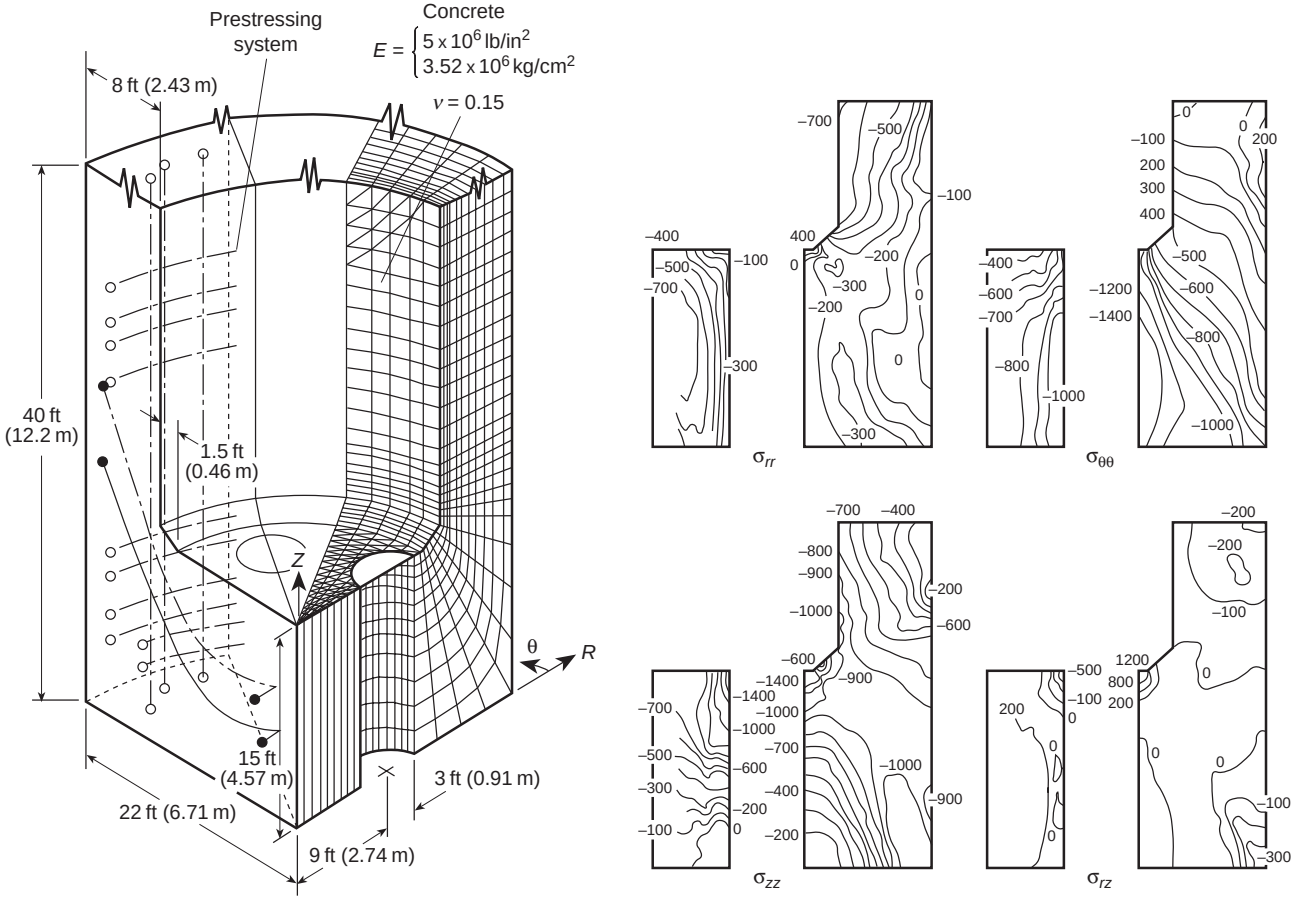


Fig. 6.7 A nuclear pressure vessel analysis using simple tetrahedral elements.[5] Geometry, subdivision, and some stress results.

Although we have in this chapter emphasized the easy visualization of a tetrahedral mesh through the use of brick-like subdivision, it is possible to generate automatically arbitrary tetrahedral meshes of great complexity with any prescribed mesh density distribution. The procedures follow the general pattern of automatic triangle generation[13] to which we shall refer in Chapter 15 when discussing efficient, adaptively constructed meshes, but, of course, the degree of complexity introduced is much greater in three dimensions. Some details of such a generator are described by Peraire *et al.*,[14] and Fig. 6.8 illustrates an intersection of such an automatically

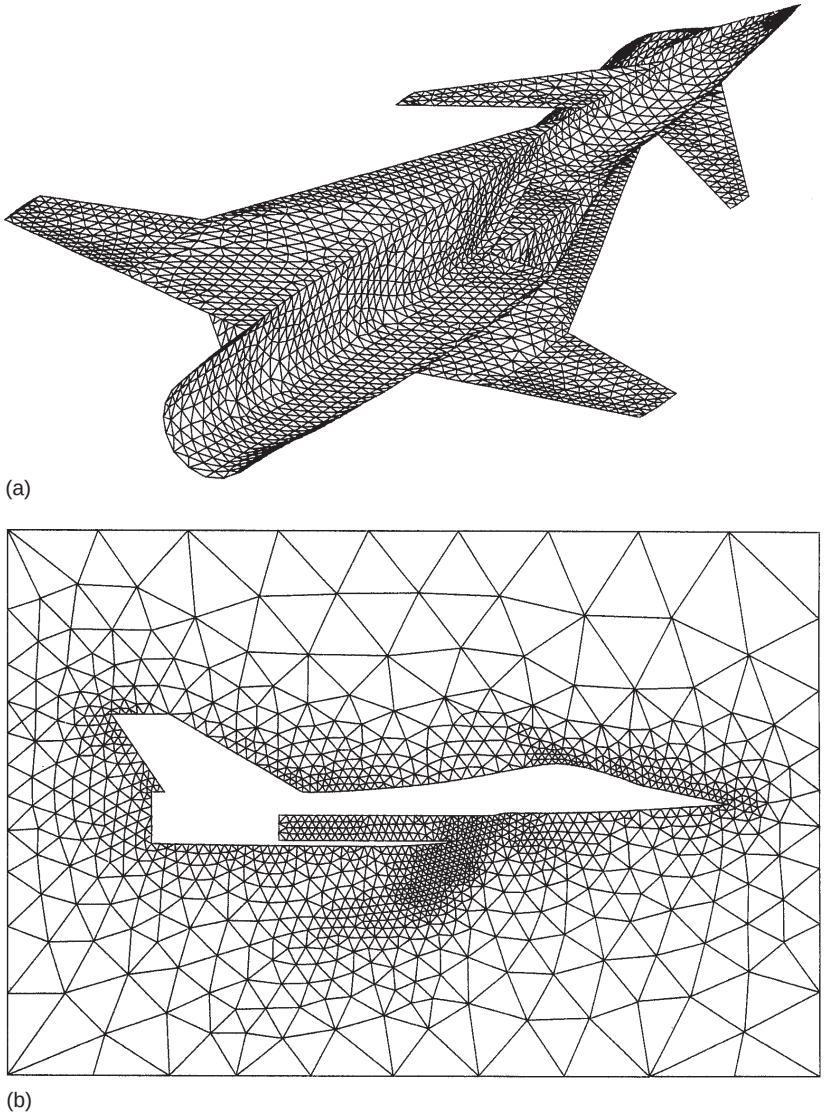


Fig. 6.8 An automatically generated mesh of tetrahedra for a specified mesh density in the exterior region on aircraft (*a*) and (*b*) an intersection of the mesh with the centreline plane.

generated mesh with an outline of an aircraft. It is impractical to show the full plot of the mesh which contains over 30 000 nodes. The important point to note is that such meshes can be generated for any configuration which can be suitably described geometrically.[15–17] Although this example concerns aerodynamics rather than elasticity, similar meshes can be generated in the latter context.

References

1. R.H. Gallagher, J. Padlog, and P.P. Bijlaard. Stress analysis of heated complex shapes. *ARS Journal*, 700–7, 1962.
2. R.J. Melosh. Structural analysis of solids. *Proc. Am. Soc. Civ. Eng.*, **ST 4**, 205–23, Aug. 1963.
3. J.H. Argyris. Matrix analysis of three-dimensional elastic media – small and large displacements. *JAIAA*, **3**, 45–51, Jan. 1965.
4. J.H. Argyris. Three-dimensional anisotropic and inhomogeneous media – matrix analysis for small and large displacements. *Ingenieur Archiv.*, **34**, 33–55, 1965.
5. Y.R. Rashid and W. Rockenhauser. Pressure vessel analysis by finite element techniques. *Proc. Conf. on Prestressed Concrete Pressure Vessels*. Inst. Civ. Eng., 1968.
6. J.H. Argyris. Continua and discontinua. *Proc. Conf. Matrix Methods in Structural Mechanics*. Wright Patterson Air Force Base, Ohio, Oct. 1965.
7. B.M. Irons. Engineering applications of numerical integration in stiffness methods. *JAIAA*, **4**, 2035–7, 1966.
8. J.G. Ergatoudis, B.M. Irons, and O.C. Zienkiewicz. Three dimensional analysis of arch dams and their foundations. *Proc. Symp. Arch Dams*. Inst. Civ. Eng., 1968.
9. J.H. Argyris and J.C. Redshaw. Three dimensional analysis of two arch dams by a finite element method. *Proc. Symp. Arch Dams*. Inst. Civ. Eng., 1968.
10. S. Fjeld. Three dimensional theory of elastics. *Finite Element Methods in Stress Analysis* (eds I. Holand and K. Bell), Tech. Univ. of Norway, Tapir Press, Trondheim, 1969.
11. S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. 2nd ed., McGraw-Hill, 1951.
12. Oliveira Pedro. Thesis, Laboratorio Nacional de Engenharia Civil, Lisbon, 1967.
13. J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comp. Physics*, **72**, 449–66, 1987.
14. J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and O.C. Zienkiewicz. Finite element Euler computations in three dimensions. *Int. J. Num. Meth. Eng.* 1988 (to be published).
15. N.P. Weatherill, P.R. Eiseman, J. Hause, and J.F. Thompson. *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*. Pineridge Press, Swansea, 1994.
16. J.F. Thompson, B.K. Soni, and N.P. Weatherill, editors. *Handbook of Grid Generation*. CRC Press, January 1999.
17. GiD – The Personal Pre/Postprocessor (CIMNE). Barcelona, Spain, 1999.

7

Steady-state field problems – heat conduction, electric and magnetic potential, fluid flow, etc.

7.1 Introduction

While, in detail, most of the previous chapters dealt with problems of an elastic continuum the general procedures can be applied to a variety of physical problems. Indeed, some such possibilities have been indicated in Chapter 3 and here more detailed attention will be given to a particular but wide class of such situations.

Primarily we shall deal with situations governed by the general 'quasi-harmonic' equation, the particular cases of which are the well-known Laplace and Poisson equations.[1–6] The range of physical problems falling into this category is large. To list but a few frequently encountered in engineering practice we have:

Heat conduction
Seepage through porous media
Irrotational flow of ideal fluids
Distribution of electrical (or magnetic) potential
Torsion of prismatic shafts
Bending of prismatic beams,
Lubrication of pad bearings, etc.

The formulation developed in this chapter is equally applicable to all, and hence little reference will be made to the actual physical quantities. Isotropic or anisotropic regions can be treated with equal ease.

Two-dimensional problems are discussed in the first part of the chapter. A generalization to three dimensions follows. It will be observed that the same, C_0, 'shape functions' as those used previously in two- or three-dimensional formulations of elasticity problems will again be encountered. The main difference will be that now only one unknown scalar quantity (the unknown function) is associated with each point in space. Previously, several unknown quantities, represented by the displacement vector, were sought.

In Chapter 3 we indicated both the 'weak form' and a variational principle applicable to the Poisson and Laplace equations (see Secs 3.2 and 3.8.1). In the following sections we shall apply these approaches to a general, quasi-harmonic equation and indicate the ranges of applicability of a *single*, *unified*, *approach* by which one computer program can solve a large variety of physical problems.

7.2 The general quasi-harmonic equation

7.2.1 The general statement

In many physical situations we are concerned with the *diffusion* or flow of some quantity such as heat, mass, or a chemical, etc. In such problems the rate of transfer per unit area, $\mathbf{q}$, can be written in terms of its cartesian components as

$$\mathbf{q}^{\mathrm{T}} = [q_x, q_y, q_z] \tag{7.1}$$

If the rate at which the relevant quantity is generated (or removed) per unit volume is Q, then for steady-state flow the balance or continuity requirement gives

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} + Q = 0 \tag{7.2}$$

Introducing the gradient operator

$$\boldsymbol{\nabla} = \begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial z} \end{Bmatrix} \tag{7.3}$$

we can write the above as

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q} + Q = 0 \tag{7.4}$$

Generally the rates of flow will be related to *gradients* of some potential quantity ϕ. This may be temperature in the case of heat flow, etc. A very general linear relationship will be of the form

$$\mathbf{q} = \begin{Bmatrix} q_x \\ q_y \\ q_z \end{Bmatrix} = -\mathbf{k} \begin{Bmatrix} \dfrac{\partial \phi}{\partial x} \\ \dfrac{\partial \phi}{\partial y} \\ \dfrac{\partial \phi}{\partial z} \end{Bmatrix} = -\mathbf{k}\,\boldsymbol{\nabla}\phi \tag{7.5}$$

where $\mathbf{k}$ is a three by three matrix. This is generally of a symmetric form due to energy arguments and is variously referred to as Fourier's, Fick's, or Darcy's law depending on the physical problem.

The final governing equation for the 'potential' ϕ is obtained by substitution of Eq. (7.5) into (7.4), leading to

$$-\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{k}\,\boldsymbol{\nabla}\phi + Q = 0 \tag{7.6}$$

which has to be solved in the domain Ω. On the boundaries of such a domain we shall usually encounter one or other of the following conditions:

1. On Γ_ϕ,

$$\phi = \bar{\phi} \tag{7.7a}$$

   i.e., the potential is specified.
2. On Γ_q the normal component of flow, q_n, is given as

$$q_n = \bar{q} + \alpha\phi \tag{7.7b}$$

   where α is a transfer or radiation coefficient.

   As

$$q_n = \mathbf{q}^{\mathrm{T}}\mathbf{n} \qquad \mathbf{n}^{\mathrm{T}} = [n_x, n_y, n_z]$$

   where $\mathbf{n}$ is a vector of direction cosines of the normal to the surface, this condition can immediately be rewritten as

$$(\mathbf{k}\,\boldsymbol{\nabla}\phi)^{\mathrm{T}}\mathbf{n} + \bar{q} + \alpha\phi = 0 \tag{7.7c}$$

   in which $\bar{q}$ and α are given.

7.2.2 Particular forms

If we consider the general statement of Eq. (7.5) as being determined for an arbitrary set of coordinate axes x, y, z we shall find that it is always possible to determine locally another set of axes x', y', z' with respect to which the matrix $\mathbf{k}'$ becomes diagonal. With respect to such axes we have

$$\mathbf{k}' = \begin{bmatrix} k_{x'} & 0 & 0 \\ 0 & k_{y'} & 0 \\ 0 & 0 & k_{z'} \end{bmatrix} \tag{7.8}$$

and the governing equation (7.6) can be written (now dropping the prime)

$$-\left[\frac{\partial}{\partial x}\left(k_x\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial\phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial\phi}{\partial z}\right)\right] + Q = 0 \tag{7.9}$$

with a suitable change of boundary conditions.

Lastly, for an isotropic material we can write

$$\mathbf{k} = k\mathbf{I} \tag{7.10}$$

where $\mathbf{I}$ is an identity matrix. This leads to the simple form of Eq. (3.10) which was discussed in Chapter 3.

7.2.3 Weak form of general quasi-harmonic equation [Eq. (7.6)]

Following the principles of Chapter 3, Sec. 3.2, we can obtain the weak form of

Eq. (7.6) by writing

$$\int_\Omega v(-\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{k}\,\boldsymbol{\nabla}\phi + Q)\,\mathrm{d}\Omega + \int_{\Gamma_q} v[(\mathbf{k}\,\boldsymbol{\nabla}\phi)^{\mathrm{T}}\mathbf{n} + \bar{q} + \alpha\phi]\,\mathrm{d}\Gamma = 0 \tag{7.11}$$

for all functions v which are zero on Γ_ϕ.

Integration by parts (see Appendix G) will result in the following weak statement which is equivalent to satisfying the governing equations and the *natural* boundary conditions (7.7b):

$$\int_\Omega (\boldsymbol{\nabla} v)^{\mathrm{T}}\mathbf{k}\,\boldsymbol{\nabla}\phi\,\mathrm{d}\Omega + \int_\Omega vQ\,\mathrm{d}\Omega + \int_{\Gamma_q} v(\alpha\phi + \bar{q})\,\mathrm{d}\Gamma = 0 \tag{7.12}$$

The *forced* boundary condition (7.7a) still needs to be imposed.

7.2.4 The variational principle

We shall leave as an exercise to the reader the verification that the functional

$$\Pi = \frac{1}{2}\int_\Omega (\boldsymbol{\nabla}\phi)^{\mathrm{T}}\mathbf{k}\,\boldsymbol{\nabla}\phi\,\mathrm{d}\Omega + \int_\Omega \phi Q\,\mathrm{d}\Omega + \frac{1}{2}\int_{\Gamma_q} \alpha\phi^2\,\mathrm{d}\Gamma + \int_{\Gamma_q} \phi\bar{q}\,\mathrm{d}\Gamma \tag{7.13}$$

gives on minimization [subject to the constraint of Eq. (7.7a)] the satisfaction of the original problem set in Eqs (7.6) and (7.7).

The algebraic manipulations required to verify the above principle follow precisely the lines of Sec. 3.8 of Chapter 3 and can be carried out as an exercise.

7.3 Finite element discretization

This can now proceed on the assumption of a trial function expansion

$$\phi = \sum N_i a_i = \mathbf{N}\mathbf{a} \tag{7.14}$$

using either the weak formulation of Eq. (7.12) or the variational statement of Eq. (7.13). If, in the first, we take

$$v = \sum W_i\,\delta a_i \qquad \text{with} \qquad W_i = N_i \tag{7.15}$$

according to the Galerkin principle, an identical form will arise with that obtained from the minimization of the variational principle.

Substituting Eq. (7.15) into (7.12) we have a typical statement giving

$$\left(\int_\Omega (\boldsymbol{\nabla} N_i)^{\mathrm{T}}\mathbf{k}\,\boldsymbol{\nabla}\mathbf{N}\,\mathrm{d}\Omega + \int_{\Gamma_q} N_i\alpha\mathbf{N}\,\mathrm{d}\Gamma\right)\mathbf{a} + \int_\Omega N_i Q\,\mathrm{d}\Omega + \int_{\Gamma_q} N_i\bar{q}\,\mathrm{d}\Gamma = 0$$
$$i = 1, \ldots, n \tag{7.16}$$

or a set of standard discrete equations of the form

$$\mathbf{H}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{7.17}$$

with

$$H_{ij} = \int_\Omega (\boldsymbol{\nabla} N_i)^{\mathrm{T}} \mathbf{k}\, \boldsymbol{\nabla} N_j \, \mathrm{d}\Omega + \int_{\Gamma_q} N_i \alpha N_j \, \mathrm{d}\Gamma \qquad f_i = \int_\Omega N_i Q \, \mathrm{d}\Omega + \int_{\Gamma_q} N_i \bar{q} \, \mathrm{d}\Gamma$$

on which prescribed values of $\bar{\phi}$ have to be imposed on boundaries Γ_ϕ.

We note now that an additional 'stiffness' is contributed on boundaries for which a radiation constant α is specified but that otherwise a complete analogy with the elastic structural problem exists.

Indeed in a computer program the same standard operations will be followed even including an evaluation of quantities analogous to the stresses. These, obviously, are the fluxes

$$\mathbf{q} \equiv -\mathbf{k}\, \boldsymbol{\nabla} \phi = -(\mathbf{k}\, \boldsymbol{\nabla} \mathbf{N})\mathbf{a} \tag{7.18}$$

and, as with stresses, the best recovery procedure is discussed in Chapter 14.

7.4 Some economic specializations

7.4.1 Anisotropic and non-homogeneous media

Clearly material properties defined by the **k** matrix can vary from element to element in a discontinuous manner. This is implied in both the weak and variational statements of the problem.

The material properties are usually known only with respect to the principal (or symmetry) axes, and if these directions are constant within the element it is convenient to use them in the formulation of local axes specified within each element, as shown in Fig. 7.1.

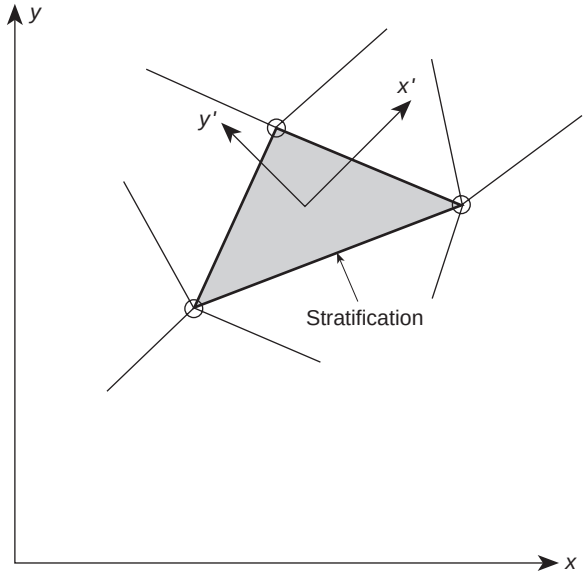


Fig. 7.1 Anisotropic material. Local coordinates coincide with the principal directions of stratification.

With respect to such axes only three coefficients k_x, k_y, and k_z need be specified, and now only a multiplication by a diagonal matrix is needed in formulating the coefficients of the matrix **H** [Eq. (7.17)].

It is important to note that as the parameters **a** *correspond to scalar values, no transformation of matrices computed in local coordinates is necessary before assembly of the global matrices.*

Thus, in many computer programs only a diagonal specification of the **k** matrix is used.

7.4.2 Two-dimensional problem

The two-dimensional plane case is obtained by taking the gradient in the form

$$\boldsymbol{\nabla} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right]^{\mathrm{T}} \tag{7.19}$$

and taking the flux as

$$\mathbf{q} = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = -\begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \phi}{\partial x} \\ \dfrac{\partial \phi}{\partial y} \end{Bmatrix} \tag{7.20}$$

On discretization by Eq. (7.16) a slightly simplified form of the matrices will now be found. Dropping the terms with α and $\bar{q}$ we can write

$$H_{ij}^e = \int_{V^e} \left(k_x \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + k_y \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) \mathrm{d}x \, \mathrm{d}y \tag{7.21}$$

No further discussion at this point appears necessary. However, it may be worthwhile to particularize here to the simplest yet still useful triangular element (Fig. 7.2).

With

$$N_i = \frac{a_i + b_i x + c_i y}{2\Delta}$$

as in Eq. (4.8) of Chapter 4, we can write down the element 'stiffness' matrix as

$$\mathbf{H}^e = \frac{k_x}{4\Delta} \begin{bmatrix} b_i b_i & b_i b_j & b_i b_m \\ & b_j b_j & b_j b_m \\ \text{symmetric} & & b_m b_m \end{bmatrix} + \frac{k_y}{4\Delta} \begin{bmatrix} c_i c_i & c_i c_j & c_i c_m \\ & c_j c_j & c_j c_m \\ \text{symmetric} & & c_m c_m \end{bmatrix} \tag{7.22}$$

The load matrices follow a similar simple pattern and thus, for instance, the reader can show that due to Q we have

$$\mathbf{f}^e = -\frac{Q\Delta}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \tag{7.23}$$

a very simple (almost 'obvious') result.

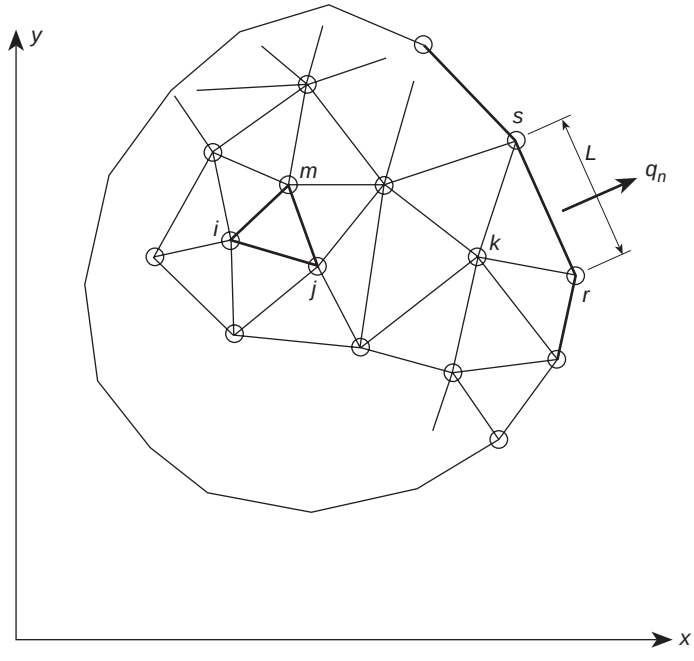


Fig. 7.2 Division of a two-dimensional region into triangular elements.

Alternatively the formulation may be specialized to cylindrical coordinates and used for the solution of axisymmetric situations by introducing the gradient

$$\boldsymbol{\nabla} = \left[\frac{\partial}{\partial r}, \frac{\partial}{\partial z}\right]^{\mathrm{T}} \tag{7.24}$$

where r, z replace x, y. With the flux now given by

$$\mathbf{q} = \begin{Bmatrix} q_r \\ q_z \end{Bmatrix} = -\begin{bmatrix} k_r & 0 \\ 0 & k_z \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \phi}{\partial r} \\ \dfrac{\partial \phi}{\partial z} \end{Bmatrix} \tag{7.25}$$

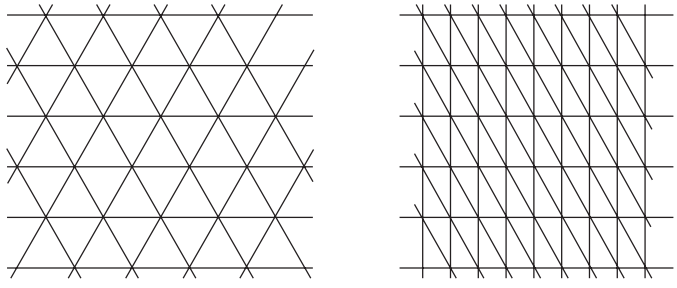
the discretization of Eq. (7.16) is now performed with the volume element expressed by

$$\mathrm{d}\Omega = 2\pi r \, \mathrm{d}r \, \mathrm{d}z$$

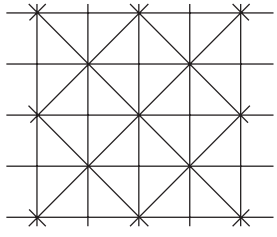
and integration carried out as described in Chapter 5, Section 5.2.5.

7.5 Examples – an assessment of accuracy

It is very easy to show that by assembling explicitly worked out 'stiffnesses' of triangular elements for 'regular' meshes shown in Fig. 7.3a, the discretized plane equations are *identical* with those that can be derived by well-known finite difference methods.[7]



(a)



(b)

Fig. 7.3 'Regular' and 'irregular' subdivision patterns.

Obviously the solutions obtained by the two methods will be identical, and so also will be the orders of approximation.†

If an 'irregular' mesh based on a square arrangement of nodes is used a difference between the two aproaches will be evident [Fig. 7.3(*b*)]. This is confined to the 'load' vector $\mathbf{f}^e$. The assembled equations will show 'loads' which differ by small amounts from node to node, but the sum of which is still the same as that due to the finite difference expressions. The solutions therefore differ only locally and will represent the same averages.

In Fig. 7.4 a test comparing the results obtained on an 'irregular' mesh with a relaxation solution of the lowest order finite difference approximation is shown. Both give results of similar accuracy, as indeed would be anticipated. However, it can be shown that in one-dimensional problems the finite element algorithm gives *exact* answers of nodes, while the finite difference method generally does not. In general, therefore, superior accuracy is available with the finite element discretization.

Further advantages of the finite element process are:

1. It can deal simply with non-homogeneous and anisotropic situations (particularly when the direction of anisotropy is variable).
2. The elements can be graded in shape and size to follow arbitrary boundaries and to allow for regions of rapid variation of the function sought, thus controlling the errors in a most efficient way (viz. Chapters 14 and 15).
3. Specified gradient or 'radiation' boundary conditions are introduced naturally and with a better accuracy than in standard finite difference procedures.

† This is only true in the case where the boundary values $\bar{\phi}$ are prescribed.

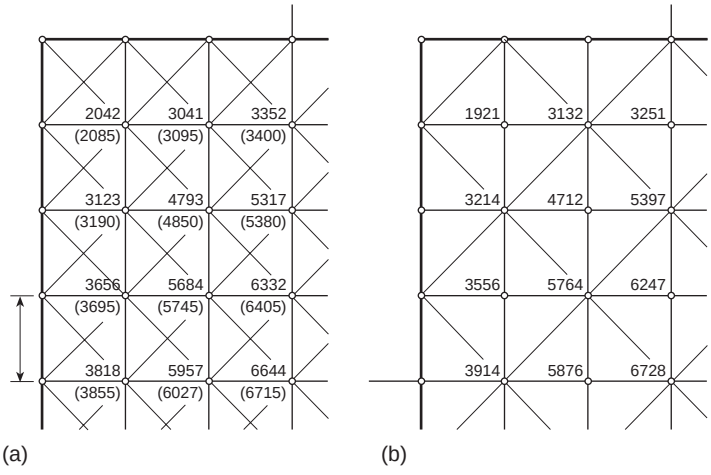


Fig. 7.4 Torsion of a rectangular shaft. Numbers in parentheses show a more accurate solution due to Southwell using a 12 × 16 mesh (values of $\phi/G\theta L^2$).

4. Higher order elements can be readily used to improve accuracy without complicating boundary conditions – a difficulty always arising with finite difference approximations of a higher order.
5. Finally, but of considerable importance in the computer age, standard programs may be used for assembly and solution.

Two more realistic examples are given at this stage to illustrate the accuracy attainable in practice. The first is the problem of pure torsion of a non-homogeneous shaft illustrated in Fig. 7.5. The basic differential equation here is

$$\frac{\partial}{\partial x}\left(\frac{1}{G}\frac{\partial \phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{G}\frac{\partial \phi}{\partial y}\right)+2\theta=0 \tag{7.26}$$

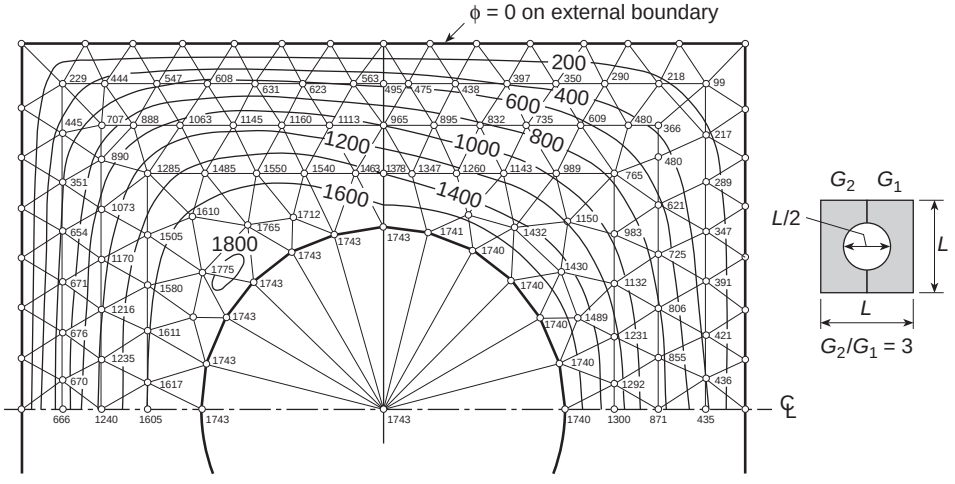


Fig. 7.5 Torsion of a hollow bimetallic shaft. $\phi/G\theta L^2 \times 10^4$.

in which ϕ is the stress function, G is the shear modulus, and θ the angle of twist per unit length of the shaft.

In the finite element solution presented, the hollow section was represented by a material for which G has a value of the order of 10^{-3} compared with the other materials.† The results compare well with the contours derived from an accurate finite difference solution.[8]

An example concerning flow through an anisotropic porous foundation is shown in Fig. 7.6.

Here the governing equation is

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial H}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial H}{\partial y}\right) = 0 \tag{7.27}$$

in which H is the hydraulic head and k_x and k_y represent the permeability coefficients in the direction of the (inclined) principal axes. The answers are here compared against contours derived by an exact solution. The possibilities of the use of a graded size of subdivision are evident in this example.

7.6 Some practical applications

7.6.1 Anisotropic seepage

The first of the problems is concerned with the flow through highly non-homogeneous, anisotropic, and contorted strata. The basic governing equation is still Eq. (7.27). However, a special feature has to be incorporated to allow for changes of x' and y' principal directions from element to element.

No difficulties are encountered in computation, and the problem together with its solution is given in Fig. 7.7.[3]

7.6.2 Axisymmetric heat flow

The axisymmetric heat flow equation results by using (7.24) and (7.25) with ϕ replaced by T. Now T is the temperature and k the conductivity.

In Fig. 7.8 the temperature distribution in a nuclear reactor pressure vessel[1] is shown for steady-state heat conduction when a uniform temperature increase is applied on the inside.

7.6.3 Hydrodynamic pressures on moving surfaces

If a submerged surface moves in a fluid with prescribed accelerations and a small amplitude of movement, then it can be shown[9] that if compressibility is ignored the

† This was done to avoid difficulties due to the 'multiple connection' of the region and to permit the use of a standard program.

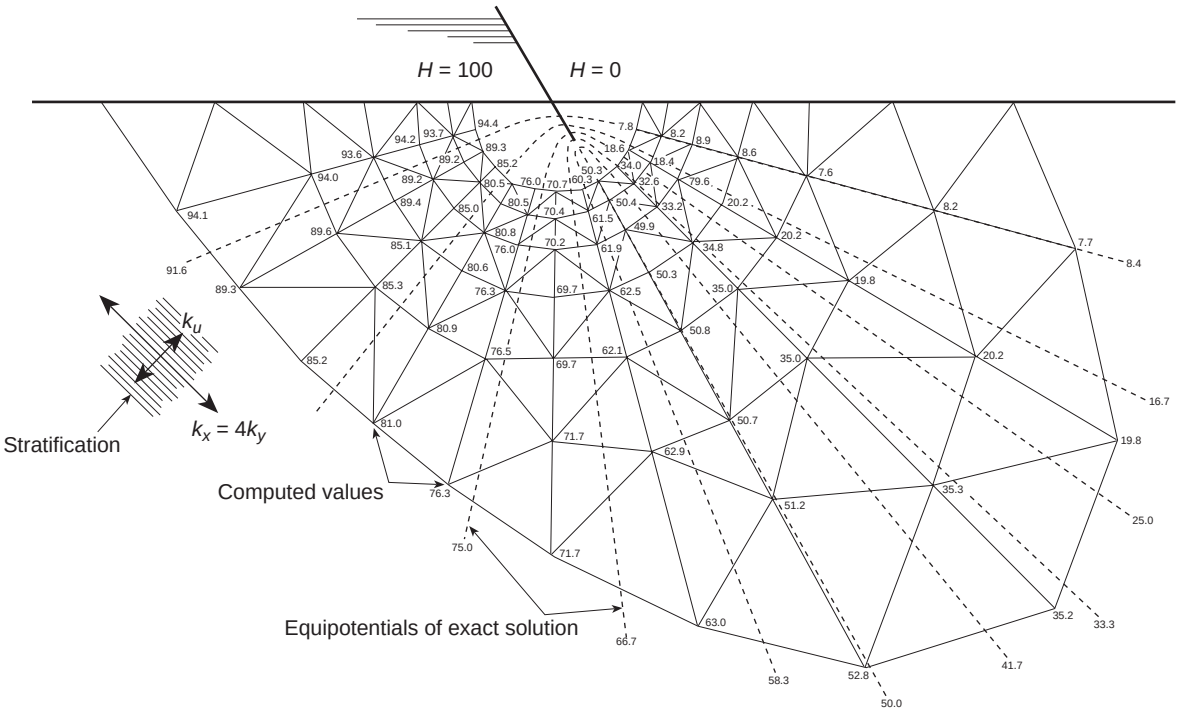


Fig. 7.6 Flow under an inclined pile wall in a stratified foundation. A fine mesh near the tip of the pile is not shown. Comparison with exact solution given by contours.

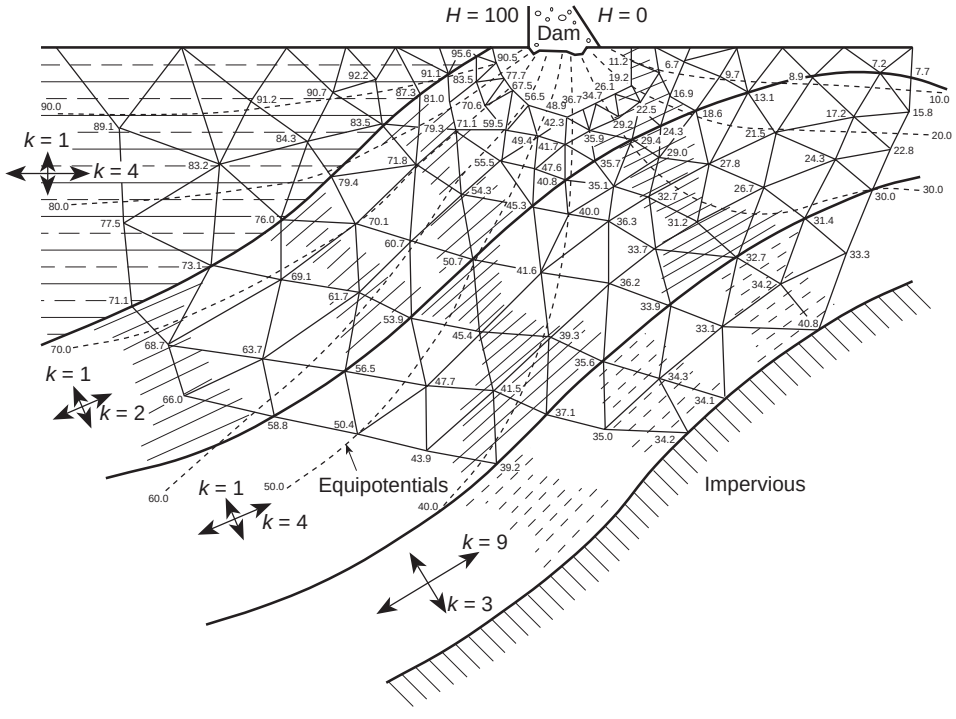


Fig. 7.7 Flow under a dam through a highly non-homogeneous and contorted foundation.

excess pressures that are developed obey the Laplace equation

$$\nabla^2 p = 0$$

On moving (or stationary) boundaries the boundary condition is of type 2 [see Eq. (7.7b)] and is given by

$$\frac{\partial p}{\partial n} = -\rho a_n \tag{7.28}$$

in which ρ is the density of the fluid and a_n is the normal component of acceleration of the boundary.

On free surfaces the boundary condition is (if surface waves are ignored) simply

$$p = 0 \tag{7.29}$$

The problem clearly therefore comes into the category of those discussed in this chapter.

As an example, let us consider the case of a vertical wall in a reservoir, shown in Fig. 7.9, and determine the pressure distribution at points along the surface of the wall and at the bottom of the reservoir for any prescribed motion of the boundary points 1 to 7.

The division of the region into elements (42 in number) is shown. Here elements of rectangular shape are used (see Sect 3.3) and combined with quadrilaterals composed

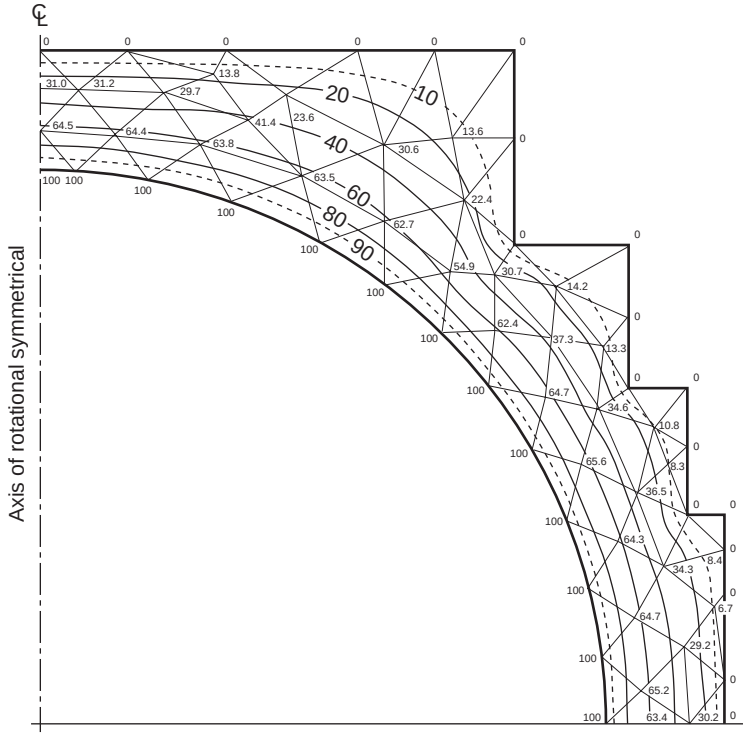


Fig. 7.8 Temperature distribution in steady-state conduction for an axisymmetrical pressure vessel.

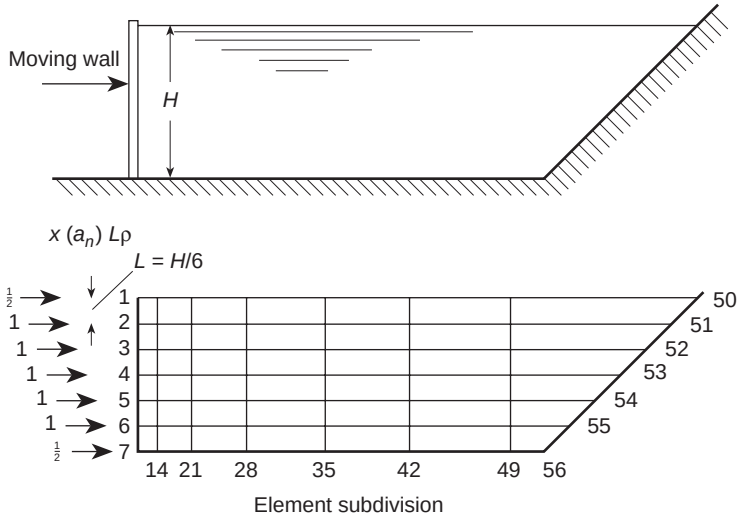


Fig. 7.9 Problem of a wall moving horizontally in a reservoir.

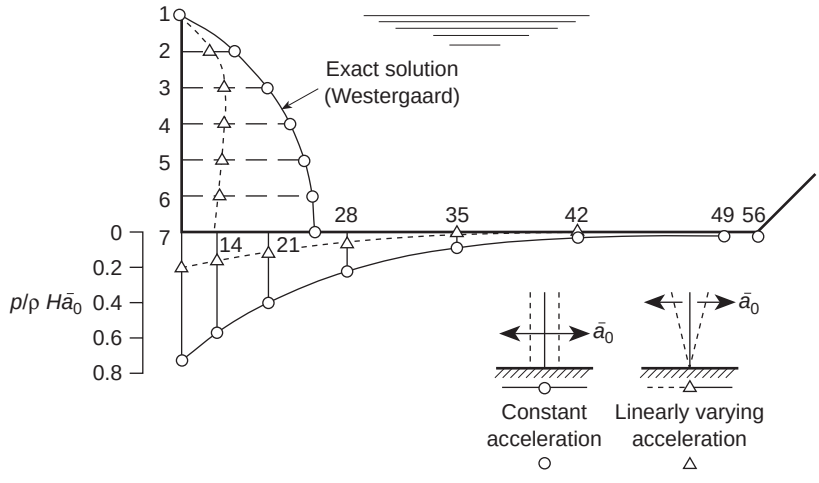


Fig. 7.10 Pressure distribution on a moving wall and reservoir bottom.

of two triangles near the sloping boundary. The pressure distribution on the wall and the bottom of the reservoir for a constant acceleration of the wall is shown in Fig. 7.10. The results for the pressures on the wall agree to within 1 per cent with the well-known, exact solution derived by Westergaard.[10]

For the wall hinged at the base and oscillating around this point with the top (point 1) accelerating by $\bar{a}_0$, the pressure distribution obtained is also plotted in Fig. 7.10.

In the study of vibration problems the interaction of the fluid pressure with structural accelerations may be determined using Eq. (7.28) and the formulation given above. This and related problems will be discussed in more detail in Chapter 19.

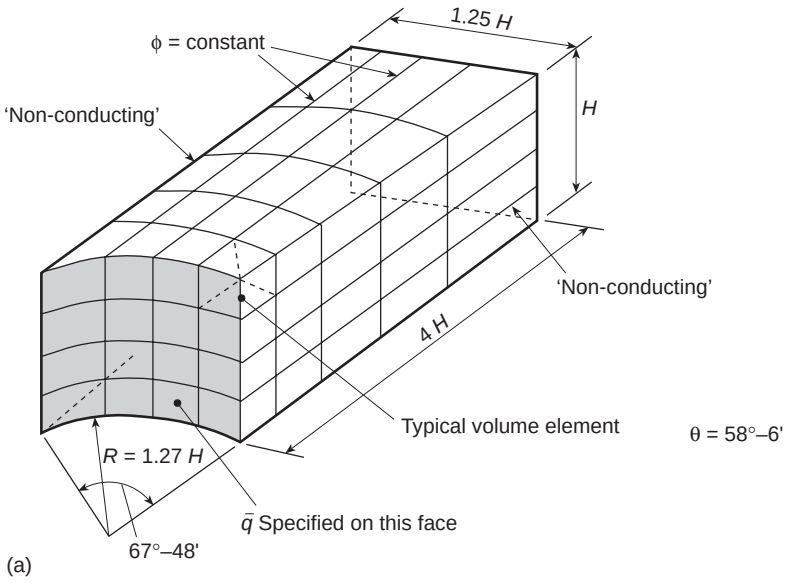
In Fig. 7.11 the solution of a similar problem in three dimensions is shown.[4] Here simple tetrahedral elements combined as bricks as described in Chapter 6 were used and very good accuracy obtained.

In many practical problems the computation of such simplified 'added' masses is sufficient, and the process described here has become widely used in this context.[11–13]

7.6.4 Electrostatic and magnetostatic problems

In this area of activity frequent need arises to determine appropriate field strengths and the governing equations are usually of the standard quasi-harmonic type discussed here. Thus the formulations are directly transferable. One of the first applications made as early as 1967[4] was to fully three-dimensional electrostatic field distributions governed by simple Laplace equations (Fig. 7.12).

In Fig. 7.13 a similar use of triangular elements was made in the context of magnetic two-dimensional fields by Winslow[6] in 1966. These early works stimulated considerable activity in this area and much work has now been published.[14–17]



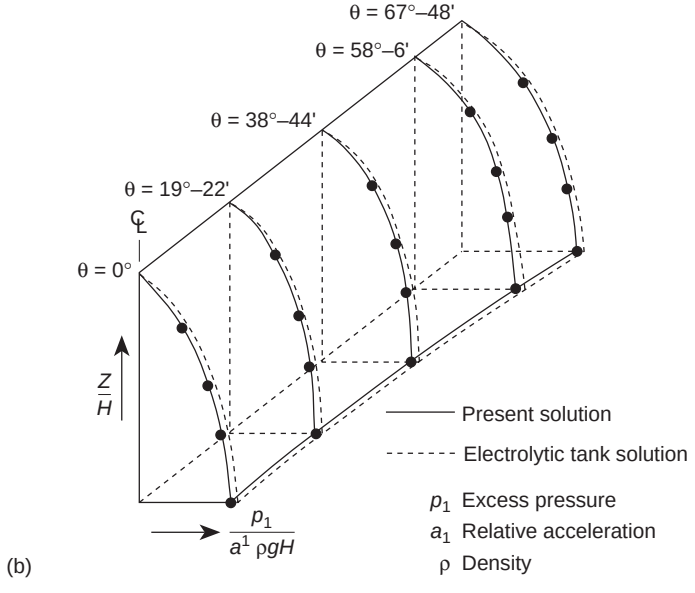


Fig. 7.11 Pressures on an accelerating surface of a dam in an incompressible fluid.

The magnetic problem is of particular interest as its formulation usually involves the introduction of a *vector potential* with three components which leads to a formulation different from those discussed in this chapter. It is, therefore, worthwhile introducing a variant which allows the standard programs of this section to be utilized for this problem.[18–20]

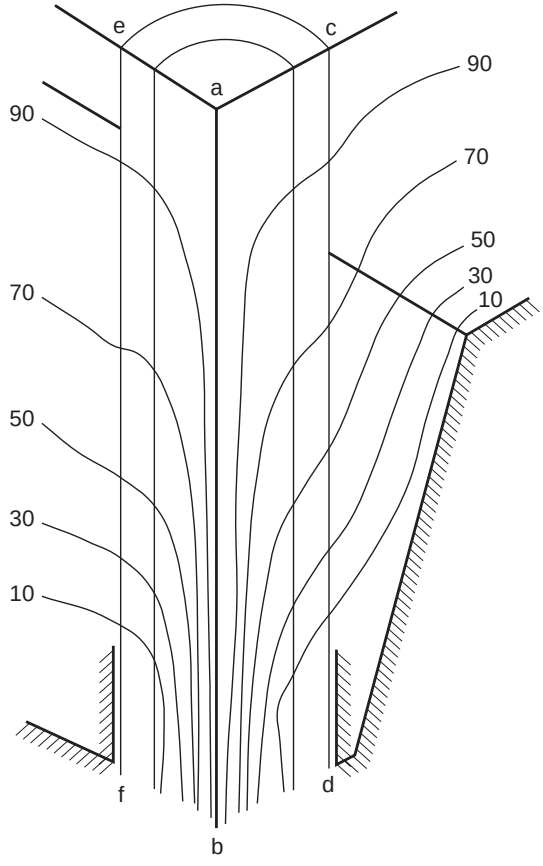


Fig. 7.12 A three-dimensional distribution of electrostatic potential around a porcelain insulator in an earthed trough[4].

In electromagnetic theory for steady-state fields the problem is governed by Maxwell's equations which are

$$\begin{aligned} \nabla \times \mathbf{H} &= -\mathbf{J} \\ \mathbf{B} &= \mu \mathbf{H} \\ \nabla^{\mathrm{T}} \mathbf{B} &= 0 \end{aligned} \tag{7.30}$$

with the boundary condition specified at an infinite distance from the disturbance, requiring **H** and **B** to tend to zero there. In the above **J** is a prescribed electric current density confined to conductors, **H** and **B** are vector quantities with three components denoting the magnetic field strength and flux density respectively, μ is the magnetic permeability which varies (in an absolute set of units) from unity *in vacuo* to several thousand in magnetizing materials and $\times$ denotes the vector product, defined in Appendix F.

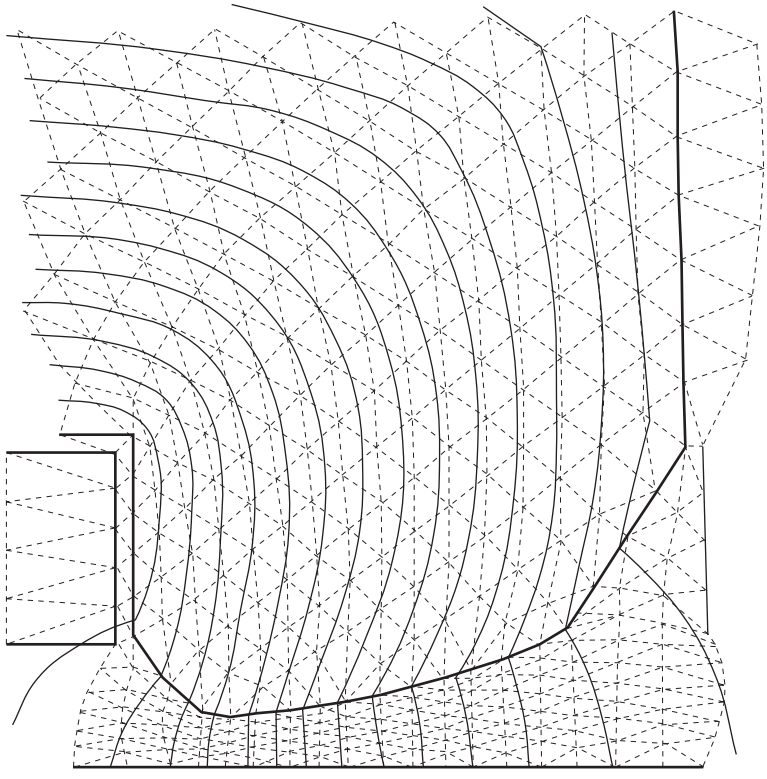


Fig. 7.13 Field near a magnet (after Winslow[6]).

The formulation presented here depends on the fact that it is a relatively simple matter to determine the field $\mathbf{H}_s$ which exactly solves Eq. (7.30) when $\mu \equiv 1$ everywhere. This is given at any point defined by a vector coordinate $\mathbf{r}$ by an integral:

$$\mathbf{H}_s = \tfrac{1}{4}\pi \int_\Omega \frac{\mathbf{J} \times (\mathbf{r} - \mathbf{r}')}{(\mathbf{r} - \mathbf{r}')^{\mathrm{T}}(\mathbf{r} - \mathbf{r}')} \, \mathrm{d}\Omega \tag{7.31}$$

In the above, $\mathbf{r}'$ refers to the coordinates of $\mathrm{d}\Omega$ and obviously the integration domain only involves the electric conductors where $\mathbf{J} \neq 0$.

With $\mathbf{H}_s$ known we can write

$$\mathbf{H} = \mathbf{H}_s + \mathbf{H}_m$$

and, on substitution into Eq. (7.30), we have a system

$$\begin{aligned} \boldsymbol{\nabla} \times \mathbf{H}_m &= \mathbf{0} \\ \mathbf{B} &= \mu(\mathbf{H}_s + \mathbf{H}_m) \\ \boldsymbol{\nabla}^{\mathrm{T}}\mathbf{B} &= 0 \end{aligned} \tag{7.32}$$

If we now introduce a *scalar* potential ϕ, defining $\mathbf{H}_m$ as

$$\mathbf{H}_m \equiv \boldsymbol{\nabla}\phi \tag{7.33}$$

we find the first of Eqs (7.36) to be automatically satisfied and, on eliminating **B** in the other two, the governing equation becomes

$$\boldsymbol{\nabla}^{\mathrm{T}}\mu\boldsymbol{\nabla}\phi + \boldsymbol{\nabla}^{\mathrm{T}}\mu\mathbf{H}_s = 0 \tag{7.34}$$

with $\phi \to 0$ at infinity. This is precisely of the standard form discussed in this chapter [Eq. (7.6)] with the second term, which is now specified, replacing Q.

An apparent difficulty exists, however, if μ varies in a discontinuous manner, as indeed we would expect it to do on the interfaces of two materials.

Here the term Q is now undefined and, in the standard discretization of Eq. (7.16) or (7.17), the term

$$\int_\Omega N_i Q \, \mathrm{d}\Omega \equiv -\int_\Omega N_i \, \boldsymbol{\nabla}^{\mathrm{T}}\mu\mathbf{H}_s \, \mathrm{d}\Omega \tag{7.35}$$

apparently has no meaning.

Integration by parts comes once again to the rescue and we note that

$$\int_\Omega N_i \, \boldsymbol{\nabla}^{\mathrm{T}}\mu\mathbf{H}_s \, \mathrm{d}\Omega \equiv -\int_\Omega \boldsymbol{\nabla}^{\mathrm{T}} N_i \mu\mathbf{H}_s + \int_\Gamma N_i \mu \mathbf{H}_s \mathbf{n} \, \mathrm{d}\Gamma \tag{7.36}$$

As in regions of constant μ, $\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{H}_s \equiv 0$, the only contribution to the forcing terms comes as a line integral of the second term at discontinuity interfaces.

Introduction of the scalar potential makes both two- and three-dimensional magnetostatic problems solvable by a standard program used for all the problems in this section. Figure 7.14 shows a typical three-dimensional solution for a transformer. Here isoparametric quadratic brick elements of the type which will be described in Chapter 8 were used.[18]

In typical magnetostatic problems a high non-linearity exists with

$$\mu = \mu(|\mathbf{H}|) \qquad \text{where} \quad |\mathbf{H}| = \sqrt{H_x^2 + H_y^2 + H_z^2} \tag{7.37}$$

The treatment of such non-linearities will be discussed in Volume 2.

Considerable economy in this and other problems of infinite extent can be achieved by the use of *infinite* elements to be discussed in Chapter 9.

7.6.5 Lubrication problems

Once again a standard Poisson type of equation is encountered in the two-dimensional domain of a bearing pad. In the simplest case of constant lubricant density and viscosity the equation to be solved is the Reynolds equation

$$\frac{\partial}{\partial x}\left(h^3 \frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(h^3 \frac{\partial p}{\partial y}\right) = 6\mu V \frac{\partial h}{\partial x} \tag{7.38}$$

where h is the film thickness, p the pressure developed, μ the viscosity and V the velocity of the pad in the x-direction.

Figure 7.15 shows the pressure distribution in the typical case of a stepped pad.[21] The boundary condition is simply that of zero pressure and it is of interest to note that

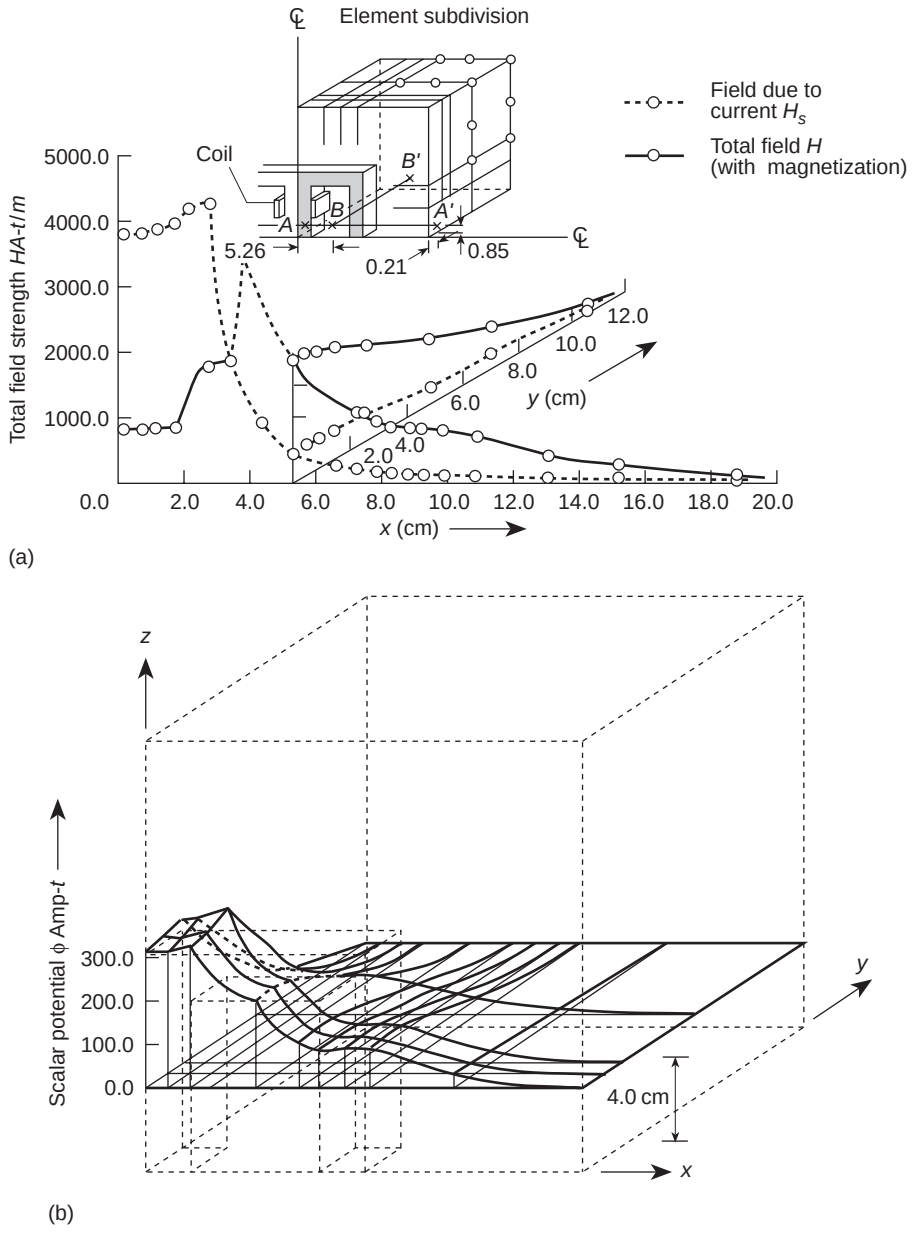


Fig. 7.14 Three-dimensional transformer. (a) Field strength H. (b) Scalar potential on plane $z = 4.0$ cm.

the step causes an equivalent of a 'line load' on integration by parts of the right-hand side of Eq. (7.38), just as in the case of magnetic discontinuity mentioned above.

More general cases of lubrication problems, including vertical pad movements (squeeze films) and compressibility, can obviously be dealt with, and much work has been done here.[22–29]

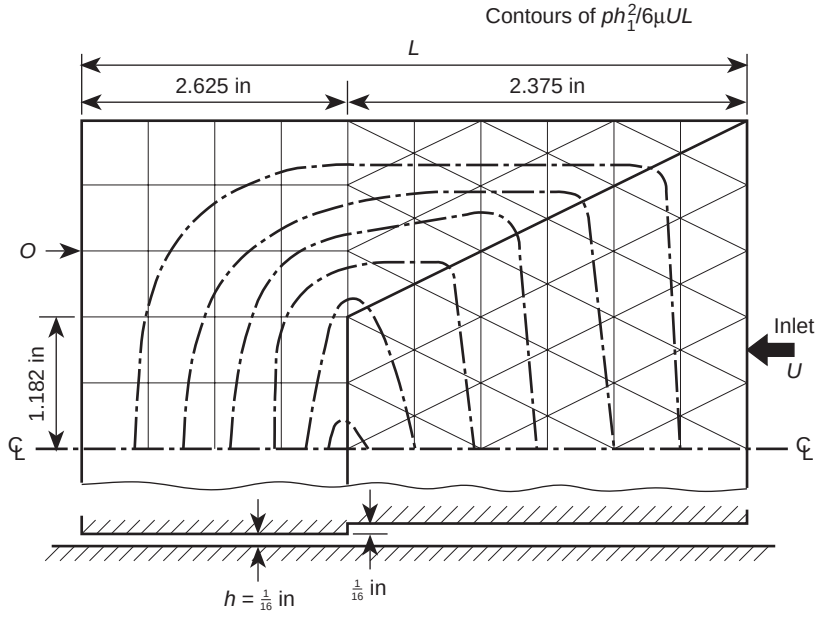


Fig. 7.15 A stepped pad bearing. Pressure distribution.

7.6.6 Irrotational and free surface flows

The basic Laplace equation which governs the flow of viscous fluid in seepage problems is also applicable in the problem of irrotational fluid flow outside the boundary layer created by viscous effects. The examples already given are adequate to illustrate the general applicability in this context. Further examples are quoted by Martin[30] and others.[31–36]

If no viscous effects exist, then it can be shown that for a fluid starting at rest the motion must be irrotational, i.e.,

$$\omega_z \equiv \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \qquad \text{etc.} \tag{7.39}$$

where u and v are appropriate velocity components.

This implies the existence of a velocity potential, giving

$$u = -\frac{\partial \phi}{\partial x} \qquad v = -\frac{\partial \phi}{\partial y} \tag{7.40}$$

$$(\text{or } \mathbf{u} = -\boldsymbol{\nabla}\phi)$$

If, further, the flow is incompressible the continuity equation [see Eq. (7.2)] has to be satisfied, i.e.,

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{u} = 0 \tag{7.41}$$

and therefore

$$\boldsymbol{\nabla}^{\mathrm{T}}\,\boldsymbol{\nabla}\phi = 0 \tag{7.42}$$

Alternatively, for two-dimensional flow a stream function may be introduced defining the velocities as

$$u = -\frac{\partial\psi}{\partial y} \qquad v = \frac{\partial\psi}{\partial x} \tag{7.43}$$

and this identically satisfies the continuity equation. The irrotationality condition must now ensure that

$$\boldsymbol{\nabla}^{\mathrm{T}}\,\boldsymbol{\nabla}\psi = 0 \tag{7.44}$$

and thus problems of ideal fluid flow can be posed in one form or the other. As the standard formulation is again applicable, there is little more that needs to be added, and for examples the reader can well consult the literature cited. We shall also discuss further such examples in Volume 3.

The similarity with problems of seepage flow, which has already been discussed, is obvious.[37,38]

A particular class of fluid flow deserves mention. This is the case when a free surface limits the extent of the flow and this surface is not known *a priori*.

The class of problem is typified by two examples – that of a freely overflowing jet [Fig. 7.16(*a*)] and that of flow through an earth dam [Fig. 7.16(*b*)]. In both, the free surface represents a streamline and in both the position of the free surface is unknown *a priori* but has to be determined so that an *additional condition* on this surface is satisfied. For instance, in the second problem, if formulated in terms of the potential H, Eq. (7.27) governs the problem.

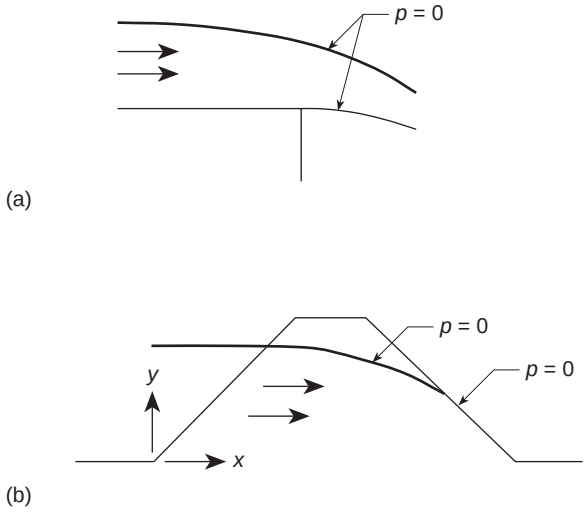


Fig. 7.16 Typical free surface problems with a streamline also satisfying an additional condition of pressure = 0. (a) Jet overflow. (b) Seepage through an earth dam.

The free surface, being a streamline, imposes the condition

$$\frac{\partial H}{\partial n} = 0 \tag{7.45}$$

to be satisfied there. In addition, however, the pressure must be zero on the surface as this is exposed to atmosphere. As

$$H = \frac{p}{\gamma} + y \tag{7.46}$$

where γ is the fluid specific weight, p is the fluid pressure, and y the elevation above some (horizontal) datum, we must have on the surface

$$H = y \tag{7.47}$$

The solution may be approached iteratively. Starting with a prescribed free surface streamline the standard problem is solved. A check is carried out to see if Eq. (7.47) is satisfied and, if not, an adjustment of the surface is carried out to make the new y equal to the H just found. A few iterations of this kind show that convergence is reasonably rapid. Taylor and Brown[39] show such a process. Alternative methods including special variational principles for dealing with this problem have been devised over the years and interested readers can consult references 40–48.

7.7 Concluding remarks

We have shown how a general formulation for the solution of a steady-state quasi-harmonic problem can be written, and how a single program of such a form can be applied to a wide variety of physical situations. Indeed, the selection of problems dealt with is by no means exhaustive and many other examples of application are of practical interest. Readers will doubtless find appropriate analogies for their own problems.

References

1. O.C. Zienkiewicz and Y.K. Cheung. Finite elements in the solution of field problems. *The Engineer*. 507–10, Sept. 1965.
2. W. Visser. A finite element method for the determination of non-stationary temperature distribution and thermal deformations. *Proc. Conf. on Matrix Methods in Structural Mechanics*. Air Force Inst. Tech., Wright-Patterson AF Base, Ohio, 1965.
3. O.C. Zienkiewicz, P. Mayer, and Y.K. Cheung. Solution of anisotropic seepage problems by finite elements. *Proc. Am. Soc. Civ. Eng*. **92**, EM1, 111–20, 1966.
4. O.C. Zienkiewicz, P.L. Arlett, and A.L. Bahrani. Solution of three-dimensional field problems by the finite element method. *The Engineer*. 27 October 1967.
5. L.R. Herrmann. Elastic torsion analysis of irregular shapes. *Proc. Am. Soc. Civ. Eng*. **91**, EM6, 11–19, 1965.
6. A.M. Winslow. Numerical solution of the quasi-linear Poisson equation in a non-uniform triangle 'mesh'. *J. Comp. Phys*. **1**, 149–72, 1966.
7. D.N. de G. Allen. *Relaxation Methods*. p. 199, McGraw-Hill, 1955.

8. J.F. Ely and O.C. Zienkiewicz. Torsion of compound bars – a relaxation solution. *Int. J. Mech. Sci.* **1**, 356–65, 1960.
9. O.C. Zienkiewicz and B. Nath. Earthquake hydrodynamic pressures on arch dams – an electric analogue solution. *Proc. Inst. Civ. Eng.* **25**, 165–76, 1963.
10. H.M. Westergaard. Water pressure on dams during earthquakes. *Trans. Am. Soc. Civ. Eng.* **98**, 418–33, 1933.
11. O.C. Zienkiewicz and R.E. Newton. Coupled vibrations of a structure submerged in a compressible fluid. *Proc. Symp. on Finite Element Techniques*. pp. 359–71, Stuttgart, 1969.
12. R.E. Newton. Finite element analysis of two-dimensional added mass and damping, in *Finite Elements in Fluids* (eds R.H. Gallagher, J.T. Oden, C. Taylor, and O.C. Zienkiewicz), Vol. I, pp. 219–32, Wiley, 1975.
13. P.A.A. Back, A.C. Cassell, R. Dungar, and R.T. Severn. The seismic study of a double curvature dam. *Prov. Inst. Civ. Eng.* **43**, 217–48, 1969.
14. P. Silvester and M.V.K. Chari. Non-linear magnetic field analysis of D.C. machines. *Trans. IEEE*. No. 7, 5–89, 1970.
15. P. Silvester and M.S. Hsieh. Finite element solution of two dimensional exterior field problems. *Proc. IEEE*. **118**, 1971.
16. B.H. McDonald and A. Wexler. Finite element solution of unbounded field problems. *Proc. IEEE*. MTT-20, No. 12, 1972.
17. E. Munro. Computer design of electron lenses by the finite element method, in *Image Processing and Computer Aided Design in Electron Optics*. p. 284, Academic Press, 1973.
18. O.C. Zienkiewicz, J.F. Lyness, and D.R.J. Owen. Three dimensional magnetic field determination using a scalar potential. A finite element solution. *IEEE, Trans. Magnetics MAG*. **13**, 1649–56, 1977.
19. J. Simkin and C.W. Trowbridge. On the use of the total scalar potential in the numerical solution of field problems in electromagnets. *Int. J. Num. Meth. Eng.* **14**, 423–40, 1979.
20. J. Simkin and C.W. Trowbridge. Three-dimensional non-linear electromagnetic field computations using scalar potentials. *Proc. Inst. Elec. Eng.* **127**, B(6), 1980.
21. D.V. Tanesa and I.C. Rao. *Student project report on lubrication*. Royal Naval College, Dartmouth, 1966.
22. M.M. Reddi. Finite element solution of the incompressible lubrication problem. *Trans. Am. Soc. Mech. Eng.* **91** (Ser. F), 524, 1969.
23. M.M. Reddi and T.Y. Chu. Finite element solution of the steady state compressible lubrication problem. *Trans. Am. Soc. Mech. Eng.* **92** (Ser. F), 495, 1970.
24. J.H. Argyris and D.W. Scharpf. The incompressible lubrication problem. *J. Roy. Aero. Soc.* **73**, 1044–6, 1969.
25. J.F. Booker and K.H. Huebner. Application of finite element methods to lubrication: an engineering approach. *J. Lubr. Techn., Trans. Am. Soc. Mech. Eng.* **14** (Ser. F), 313, 1972.
26. K.H. Huebner. Application of finite element methods to thermohydrodynamic lubrication. *Int. J. Num. Meth. Eng.* **8**, 139–68, 1974.
27. S.M. Rohde and K.P. Oh. Higher order finite element methods for the solution of compressible porous bearing problems. *Int. J. Num. Meth. Eng.* **9**, 903–12, 1975.
28. A.K. Tieu. Oil film temperature distributions in an infinitely wide glider bearing: an application of the finite element method. *J. Mech. Eng. Sci.* **15**, 311, 1973.
29. K.H. Huebner. Finite element analysis of fluid film lubrication – a survey, in *Finite Elements in Fluids* (eds R.H. Gallagher, J.T. Oden, C. Taylor, and O.C. Zienkiewicz). Vol. II, pp. 225–54, Wiley, 1975.
30. H.C. Martin. Finite element analysis of fluid flows. *Proc. 2nd Conf. on Matrix Methods in Structural Mechanics*. Air Force Inst. Tech., Wright-Patterson AF Base, Ohio, 1968.
31. G. de Vries and D.H. Norrie. *Application of the finite element technique to potential flow problems*. Reports 7 and 8, Dept. Mech. Eng., Univ. of Calgary, Alberta, Canada, 1969.

32. J.H. Argyris, G. Mareczek, and D.W. Scharpf. Two and three dimensional flow using finite elements. *J. Roy. Aero. Soc.* **73**, 961–4, 1969.
33. L.J. Doctors. An application of finite element technique to boundary value problems of potential flow. *Int. J. Num. Meth. Eng.* **2**, 243–52, 1970.
34. G. de Vries and D.H. Norrie. The application of the finite element technique to potential flow problems. *J. Appl. Mech., Am. Soc. Mech. Eng.* **38**, 978–802, 1971.
35. S.T.K. Chan, B.E. Larock, and L.R. Herrmann. Free surface ideal fluid flows by finite elements. *Proc. Am. J. Civ. Eng.* **99**, HY6, 1973.
36. B.E. Larock. Jets from two dimensional symmetric nozzles of arbitrary shape. *J. Fluid Mech.* **37**, 479–83, 1969.
37. C.S. Desai. Finite element methods for flow in porous media, in *Finite Elements in Fluids* (ed. R.H. Gallagher). Vol. 1, pp. 157–82, Wiley, 1975.
38. I. Javandel and P.A. Witherspoon. Applications of the finite element method to transient flow in porous media. *Trans. Soc. Petrol. Eng.* **243**, 241–51, 1968.
39. R.L. Taylor and C.B. Brown. Darcy flow solutions with a free surface. *Proc. Am. Soc. Civ. Eng.* **93**, HY2, 25–33, 1967.
40. J.C. Luke. A variational principle for a fluid with a free surface. *J. Fluid Mech.* **27**, 395–7, 1957.
41. K. Washizu, *Variational Methods in Elasticity and Plasticity*. 2nd ed., Pergamon Press, 1975.
42. J.C. Bruch. A survey of free-boundary value problems in the theory of fluid flow through porous media. *Advances in Water Resources.* **3**, 65–80, 1980.
43. C. Baiocchi, V. Comincioli, and V. Maione. Unconfined flow through porous media. *Meccanice. Ital. Ass. Theor. Appl. Mech.* **10**, 51–60, 1975.
44. J.M. Sloss and J.C. Bruch. Free surface seepage problem. *Proc. ASCE.* **108**, EM5, 1099–1111, 1978.
45. N. Kikuchi. Seepage flow problems by variational inequalities. *Int. J. Num. Anal. Meth. geomech.* **1**, 283–90, 1977.
46. C.S. Desai. Finite element residual schemes for unconfined flow. *Int. J. Num. Meth. Eng.* **10**, 1415–18, 1976.
47. C.S. Desai and G.C. Li. A residual flow procedure and application for free surface, and porous media. *Advances in Water Resources.* **6**, 27–40, 1983.
48. K.J. Bathe and M. Koshgoftar. Finite elements from surface seepage analysis without mesh iteration. *Int. J. Num. Anal. Meth. Geomech.* **3**, 13–22, 1979.

8

'Standard' and 'hierarchical' element shape functions: some general families of C_0 continuity

8.1 Introduction

In Chapters 4, 5, and 6 the reader was shown in some detail how linear elasticity problems could be formulated and solved using very simple finite element forms. In Chapter 7 this process was repeated for the quasi-harmonic equation. Although the detailed algebra was concerned with shape functions which arose from triangular and tetrahedral shapes only it should by now be obvious that other element forms could equally well be used. Indeed, once the element and the corresponding shape functions are determined, subsequent operations follow a standard, well-defined path which could be entrusted to an algebraist not familiar with the physical aspects of the problem. It will be seen later that in fact it is possible to program a computer to deal with wide classes of problems by specifying the shape functions only. The choice of these is, however, a matter to which intelligence has to be applied and in which the human factor remains paramount. In this chapter some rules for the generation of several families of one-, two-, and three-dimensional elements will be presented.

In the problems of elasticity illustrated in Chapters 4, 5, and 6 the displacement variable was a vector with two or three components and the shape functions were written in matrix form. They were, however, derived for each component separately and in fact the matrix expressions in these were derived by multiplying a scalar function by an identity matrix [e.g., Eqs (4.7), (5.3), and (6.7)]. This scalar form was used directly in Chapter 7 for the quasi-harmonic equation. We shall therefore concentrate in this chapter on the scalar shape function forms, calling these simply N_i.

The shape functions used in the displacement formulation of elasticity problems were such that they satisfy the convergence criteria of Chapter 2:

(*a*) the continuity of the *unknown only* had to occur between elements (i.e., slope continuity is not required), or, in mathematical language, C_0 continuity was needed;
(*b*) the function has to allow any arbitrary linear form to be taken so that the constant strain (constant first derivative) criterion could be observed.

The shape functions described in this chapter will require the satisfaction of these two criteria. They will thus be applicable to all the problems of the preceding chapters

and also to other problems which require these conditions to be obeyed. Indeed they are applicable to any situation where the functional Π or $\delta\Pi$ (see Chapter 3) is defined by derivatives of first order only.

The element families discussed will progressively have an increasing number of degrees of freedom. The question may well be asked as to whether any economic or other advantage is gained by thus increasing the complexity of an element. The answer here is not an easy one although it can be stated as a general rule that as the order of an element increases so the total number of unknowns in a problem can be reduced for a given accuracy of representation. Economic advantage requires, however, a reduction of total computation and data preparation effort, and this does not follow automatically for a reduced number of total variables because, though equation-solving times may be reduced, the time required for element formulation increases.

However, an overwhelming economic advantage in the case of three-dimensional analysis has already been hinted at in Chapters 6 and 7 for three-dimensional analyses.

The same kind of advantage arises on occasion in other problems but in general the optimum element may have to be determined from case to case.

In Sec. 2.6 of Chapter 2 we have shown that the order of error in the approximation to the unknown function is $O(h^{p+1})$, where h is the element 'size' and p is the degree of the complete polynomial present in the expansion. Clearly, as the element shape functions increase in degree so will the order of error increase, and convergence to the exact solution becomes more rapid. While this says nothing about the magnitude of error at a particular subdivision, it is clear that we should seek element shape functions with the highest complete polynomial for a given number of degrees of freedom.

8.2 Standard and hierarchical concepts

The essence of the finite element method already stated in Chapters 2 and 3 is in approximating the unknown (displacement) by an expansion given in Eqs (2.1) and (3.3). For a scalar variable u this can be written as

$$u \approx \hat{u} = \sum_{i=1}^{n} N_i a_i = \mathbf{N}\mathbf{a} \tag{8.1}$$

where n is the total number of functions used and a_i are the unknown parameters to be determined.

We have explicitly chosen to identify such variables with the values of the unknown function at element nodes, thus making

$$u_i = a_i \tag{8.2}$$

The shape functions so defined will be referred to as 'standard' ones and are the basis of most finite element programs. If polynomial expansions are used and the element satisfies Criterion 1 of Chapter 2 (which specifies that rigid body displacements cause no strain), it is clear that a constant value of a_i specified at all nodes must result in a constant value of $\hat{u}$:

$$\hat{u} = \left(\sum_{i=1}^{n} N_i \right) u_0 = u_0 \tag{8.3}$$

when $a_i = u_0$. It follows that

$$\sum_{i=1}^{n} N_i = 1 \tag{8.4}$$

at all points of the domain. This important property is known as a *partition of unity*[1] which we will make extensive use of in Chapter 16. The first part of this chapter will deal with such *standard shape functions*.

A serious drawback exists, however, with 'standard' functions, since when element refinement is made totally new shape functions have to be generated and hence all calculations repeated. It would be of advantage to avoid this difficulty by considering the expression (8.1) as a *series* in which the shape function N_i does not depend on the number of nodes in the mesh n. This indeed is achieved with *hierarchic shape functions* to which the second part of this chapter is devoted.

The hierarchic concept is well illustrated by the one-dimensional (elastic bar) problem of Fig. 8.1. Here for simplicity elastic properties are taken as constant ($D = E$) and the body force b is assumed to vary in such a manner as to produce the exact solution shown on the figure (with zero displacements at both ends).

Two meshes are shown and a linear interpolation between nodal points assumed. For both standard and hierarchic forms the coarse mesh gives

$$K_{11}^c a_1^c = f_1 \tag{8.5}$$

For a fine mesh two additional nodes are added and with the standard shape function the equations requiring solution are

$$\begin{bmatrix} K_{11}^F & K_{12}^F & 0 \\ K_{21}^F & K_{22}^F & K_{23}^F \\ 0 & K_{32}^F & K_{33}^F \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} \tag{8.6}$$

In this form the zero matrices have been automatically inserted due to element interconnection which is here obvious, and we note that as no coefficients are the same, the new equations have to be resolved. [Equation (2.13) shows how these coefficients are calculated and the reader is encouraged to work these out in detail.]

With the 'hierarchic' form using the shape functions shown, a similar form of equation arises and an identical approximation is achieved (being simply given by a series of straight segments). The *final* solution is identical but the meaning of the parameters a_i^* is now different, as shown in Fig. 8.1.

Quite generally,

$$K_{11}^F = K_{11}^c \tag{8.7}$$

as an identical shape function is used for the first variable. Further, in this particular case the off-diagonal coefficients are zero and the final equations become, for the fine mesh,

$$\begin{bmatrix} K_{11}^c & 0 & 0 \\ 0 & K_{22}^F & 0 \\ 0 & 0 & K_{33}^F \end{bmatrix} \begin{Bmatrix} a_1^* \\ a_2^* \\ a_3^* \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} \tag{8.8}$$

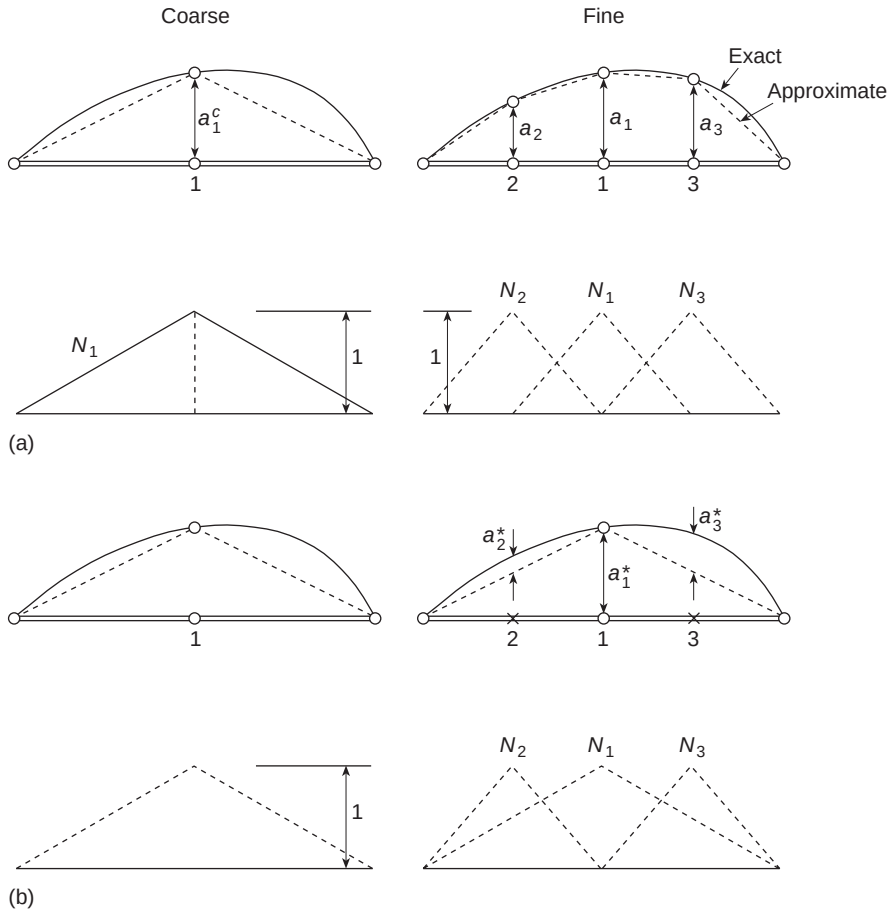


Fig. 8.1 A one-dimensional problem of stretching of a uniform elastic bar by prescribed body forces. (a) 'Standard approximation. (b) Hierarchic approximation.

The 'diagonality' feature is only true in the one-dimensional problem, but in general it will be found that the matrices obtained using hierarchic shape functions are more nearly diagonal and hence imply better conditioning than those with standard shape functions.

Although the variables are now not subject to the obvious interpretation (as local displacement values), they can be easily transformed to those if desired. Though it is not usual to use hierarchic forms in linearly interpolated elements their derivation in polynomial form is simple and very advantageous.

The reader should note that with hierarchic forms it is convenient to consider the finer mesh as still using the same, coarse, elements but now adding additional refining functions.

Hierarchic forms provide a link with other approximate (orthogonal) series solutions. Many problems solved in classical literature by trigonometric, Fourier series, expansion are indeed particular examples of this approach.

In the following sections of this chapter we shall consider the development of shape functions for high order elements with many boundary and internal degree of freedoms. This development will generally be made on simple geometric forms and the reader may well question the wisdom of using increased accuracy for such simple shaped domains, having already observed the advantage of generalized finite element methods in fitting arbitrary domain shapes. This concern is well founded, but in the next chapter we shall show a general method to map high order elements into quite complex shapes.

Part 1 'Standard' shape functions

Two-dimensional elements

8.3 Rectangular elements – some preliminary considerations

Conceptually (especially if the reader is conditioned by education to thinking in the cartesian coordinate system) the simplest element form of a two-dimensional kind is that of a rectangle with sides parallel to the x and y axes. Consider, for instance, the rectangle shown in Fig. 8.2 with nodal points numbered 1 to 8, located as shown, and at which the values of an unknown function u (here representing, for instance, one of the components of displacement) form the element parameters. How can suitable C_0 continuous shape functions for this element be determined?

Let us first assume that u is expressed in polynomial form in x and y. To ensure interelement continuity of u along the top and bottom sides the variation must be linear. Two points at which the function is common between elements lying above or below exist, and as two values uniquely determine a linear function, its identity all along these sides is ensured with that given by adjacent elements. Use of this fact was already made in specifying linear expansions for a triangle.

Similarly, if a cubic variation along the vertical sides is assumed, continuity will be preserved there as four values determine a unique cubic polynomial. Conditions for satisfying the first criterion are now obtained.

To ensure the existence of constant values of the first derivative it is necessary that all the linear polynomial terms of the expansion be retained.

Finally, as eight points are to determine uniquely the variation of the function only eight coefficients of the expansion can be retained and thus we could write

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy + \alpha_5 y^2 + \alpha_6 xy^2 + \alpha_7 y^3 + \alpha_8 xy^3 \tag{8.9}$$

The choice can in general be made unique by retaining the lowest possible expansion terms, though in this case apparently no such choice arises.† The reader will easily verify that all the requirements have now been satisfied.

† Retention of a higher order term of expansion, ignoring one of lower order, will usually lead to a poorer approximation though still retaining convergence,[2] providing the linear terms are always included.

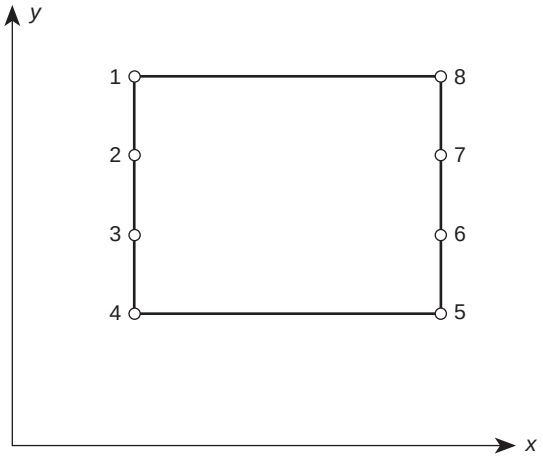


Fig. 8.2 A rectangular element.

Substituting coordinates of the various nodes a set of simultaneous equations will be obtained. This can be written in exactly the same manner as was done for a triangle in Eq. (4.4) as

$$\begin{Bmatrix} u_1 \\ \vdots \\ u_8 \end{Bmatrix} = \begin{bmatrix} 1, & x_1, & y_1, & x_1y_1, & y_1^2, & x_1y_1^2, & y_1^3, & x_1y_1^3 \\ . & . & . & . & . & . & . & . \\ 1, & x_8, & y_8, & . & . & . & . & x_8y_8^3 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \vdots \\ \alpha_8 \end{Bmatrix} \tag{8.10}$$

or simply as

$$\mathbf{u}^e = \mathbf{C}\boldsymbol{\alpha} \tag{8.11}$$

Formally,

$$\boldsymbol{\alpha} = \mathbf{C}^{-1}\mathbf{u}^e \tag{8.12}$$

and we could write Eq. (8.9) as

$$u = \mathbf{P}\boldsymbol{\alpha} = \mathbf{P}\mathbf{C}^{-1}\mathbf{u}^e \tag{8.13}$$

in which

$$\mathbf{P} = [1, x, y, xy, y^2, xy^2, y^3, xy^3] \tag{8.14}$$

Thus the shape functions for the element defined by

$$u = \mathbf{N}\mathbf{u}^e = [N_1, N_2, \ldots, N_8]\mathbf{u}^e \tag{8.15}$$

can be found as

$$\mathbf{N} = \mathbf{P}\mathbf{C}^{-1} \tag{8.16}$$

This process has, however, some considerable disadvantages. Occasionally an inverse of $\mathbf{C}$ may not exist[2,3] and *always* considerable algebraic difficulty is experienced in obtaining an expression for the inverse in general terms suitable for all element geometries. It is therefore worthwhile to consider whether shape functions $N_i(x, y)$ can be written down directly. Before doing this some general properties of these functions have to be mentioned.

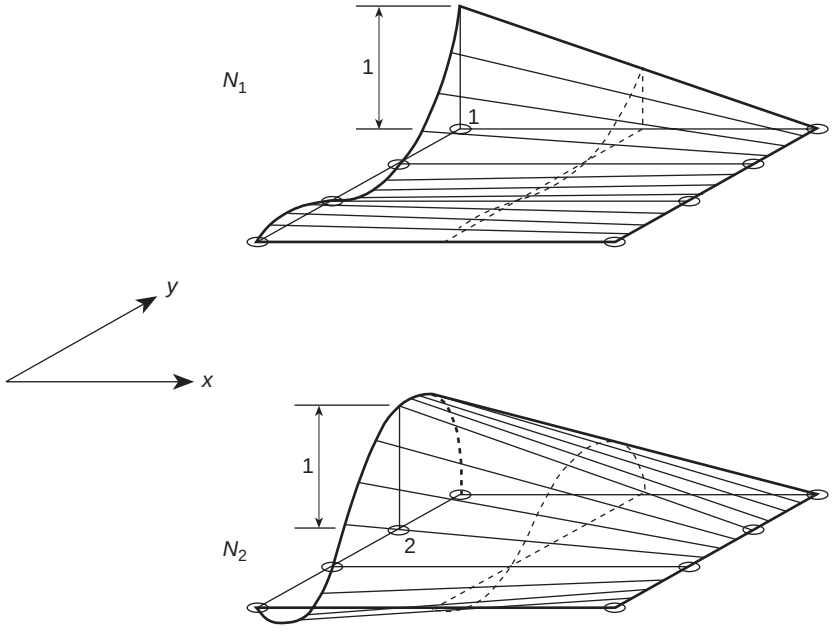


Fig. 8.3 Shape functions for elements of Fig. 8.2.

Inspection of the defining relation, Eq. (8.15), reveals immediately some important characteristics. Firstly, as this expression is valid for all components of $\mathbf{u}^e$,

$$N_i(x_j, y_j) = \delta_{ij} = \begin{cases} 1; & i = j \\ 0; & i \neq j \end{cases}$$

where δ_{ij} is known as the Kronecker delta. Further, the basic type of variation along boundaries defined for continuity purposes (e.g., linear in x and cubic in y in the above example) must be retained. The typical form of the shape functions for the elements considered is illustrated isometrically for two typical nodes in Fig. 8.3. It is clear that these could have been written down directly as a product of a suitable linear function in x with a cubic function in y. The easy solution of this example is not always as obvious but given sufficient ingenuity, a direct derivation of shape functions is always preferable.

It will be convenient to use normalized coordinates in our further investigation. Such normalized coordinates are shown in Fig. 8.4 and are chosen so that their values are ± 1 on the faces of the rectangle:

$$\begin{aligned} \xi &= \frac{x - x_c}{a} \qquad & \mathrm{d}\xi &= \frac{\mathrm{d}x}{a} \\ \eta &= \frac{y - y_c}{b} \qquad & \mathrm{d}\eta &= \frac{\mathrm{d}y}{b} \end{aligned} \tag{8.17}$$

Once the shape functions are known in the normalized coordinates, translation into actual coordinates or transformation of the various expressions occurring, for instance, in the stiffness derivation is trivial.

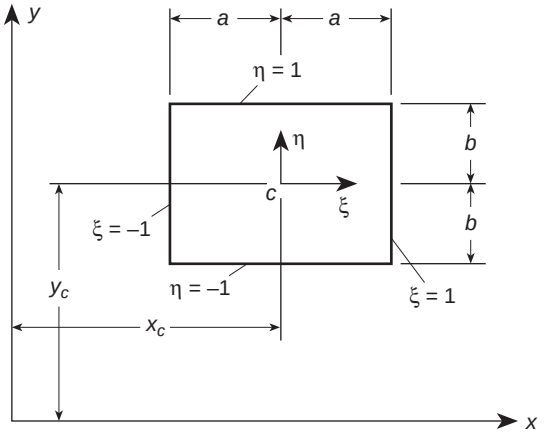


Fig. 8.4 Normalized coordinates for a rectangle.

8.4 Completeness of polynomials

The shape function derived in the previous section was of a rather special form [see Eq. (8.9)]. Only a linear variation with the coordinate x was permitted, while in y a full cubic was available. The complete polynomial contained in it was thus of order 1. In general use, a convergence order corresponding to a linear variation would occur despite an increase of the total number of variables. Only in situations where the linear variation in x corresponded closely to the exact solution would a higher order of convergence occur, and for this reason elements with such 'preferential' directions should be restricted to special use, e.g., in narrow beams or strips. In general, we shall seek element expansions which possess the highest order of a complete polynomial for a minimum of degrees of freedom. In this context it is useful to recall the Pascal triangle (Fig. 8.5) from which the number of terms

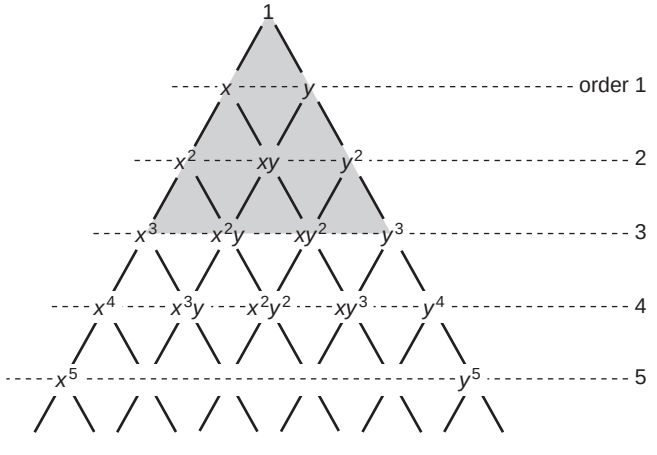


Fig. 8.5 The Pascal triangle. (Cubic expansion shaded – 10 terms).

occurring in a polynomial in two variables x, y can be readily ascertained. For instance, first-order polynomials require three terms, second-order require six terms, third-order require ten terms, etc.

8.5 Rectangular elements – Lagrange family[4–6]

An easy and systematic method of generating shape functions of any order can be achieved by simple products of appropriate polynomials in the two coordinates. Consider the element shown in Fig. 8.6 in which a series of nodes, external and internal, is placed on a regular grid. It is required to determine a shape function for the point indicated by the heavy circle. Clearly the product of a fifth-order polynomial in ξ which has a value of unity at points of the second column of nodes and zero elsewhere and that of a fourth-order polynomial in η having unity on the coordinate corresponding to the top row of nodes and zero elsewhere satisfies all the interelement continuity conditions and gives unity at the nodal point concerned.

Polynomials in one coordinate having this property are known as Lagrange polynomials and can be written down directly as

$$l_k^n(\xi) = \frac{(\xi - \xi_0)(\xi - \xi_1)\cdots(\xi - \xi_{k-1})(\xi - \xi_{k+1})\cdots(\xi - \xi_n)}{(\xi_k - \xi_0)(\xi_k - \xi_1)\cdots(\xi_k - \xi_{k-1})(\xi_k - \xi_{k+1})\cdots(\xi_k - \xi_n)} \tag{8.18}$$

giving unity at ξ_k and passing through n points.

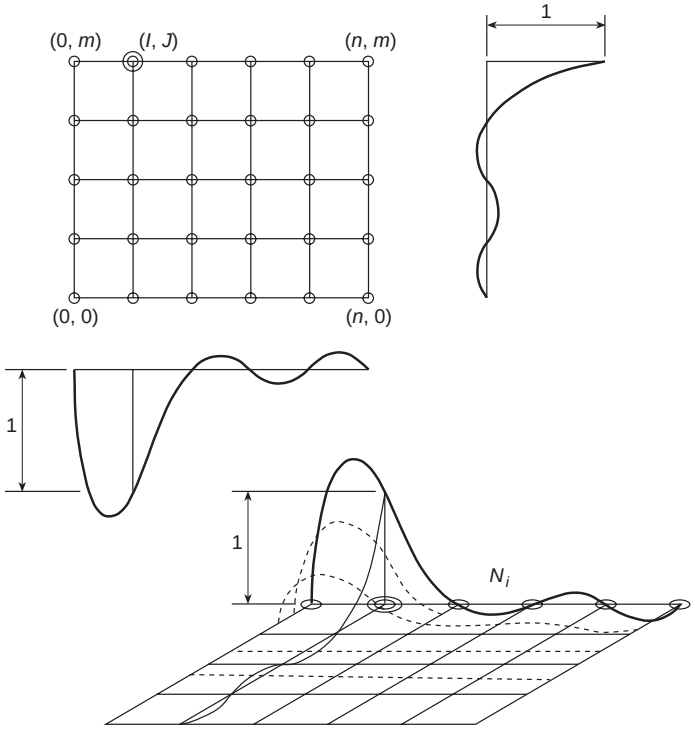


Fig. 8.6 A typical shape function for a Lagrangian element ($n = 5$, $m = 4$, $I = 1$, $J = 4$).

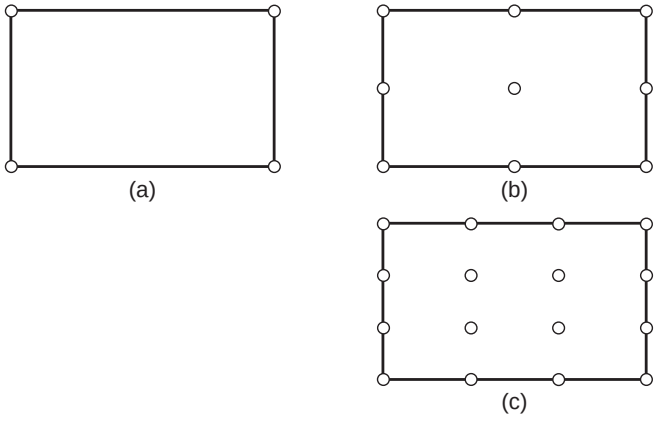


Fig. 8.7 Three elements of the Lagrange family: (*a*) linear, (*b*) quadratic, and (*c*) cubic.

Thus in two dimensions, if we label the node by its column and row number, I, J, we have

$$N_i \equiv N_{IJ} = l_I^n(\xi) l_J^m(\eta) \tag{8.19}$$

where n and m stand for the number of subdivisions in each direction.

Figure 8.7 shows a few members of this unlimited family where $m = n$.

Indeed, if we examine the polynomial terms present in a situation where $n = m$ we observe in Fig. 8.8, based on the Pascal triangle, that a large number of polynomial terms is present above those needed for a complete expansion.[7] However, when mapping of shape functions is considered (Chapter 9) some advantages occur for this family.

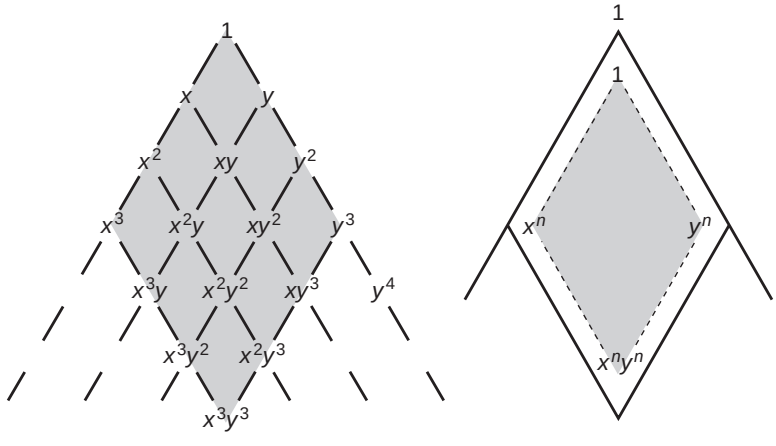


Fig. 8.8 Terms generated by a lagrangian expansion of order 3×3 (or $n \times n$). Complete polynomials of order 3 (or n).

8.6 Rectangular elements – 'serendipity' family[4,5]

It is usually more efficient to make the functions dependent on nodal values placed on the element boundary. Consider, for instance, the first three elements of Fig. 8.9. In each a progressively increasing and equal number of nodes is placed on the element boundary. The variation of the function on the edges to ensure continuity is linear, parabolic, and cubic in increasing element order.

To achieve the shape function for the first element it is obvious that a product of linear lagrangian polynomials of the form

$$\tfrac{1}{4}(\xi+1)(\eta+1) \tag{8.20}$$

gives unity at the top right corners where $\xi=\eta=1$ and zero at all the other corners. Further, a linear variation of the shape function of all sides exists and hence continuity is satisfied. Indeed this element is identical to the lagrangian one with $n=1$.

Introducing new variables

$$\xi_0=\xi\xi_i \qquad \eta_0=\eta\eta_i \tag{8.21}$$

in which ξ_i, η_i are the normalized coordinates at node i, the form

$$N_i=\tfrac{1}{4}(1+\xi_0)(1+\eta_0) \tag{8.22}$$

allows all shape functions to be written down in one expression.

As a linear combination of these shape functions yields any arbitrary linear variation of u, the second convergence criterion is satisfied.

The reader can verify that the following functions satisfy all the necessary criteria for quadratic and cubic members of the family.

'Quadratic' element

Corner nodes:

$$N_i=\tfrac{1}{4}(1+\xi_0)(1+\eta_0)(\xi_0+\eta_0-1) \tag{8.23}$$

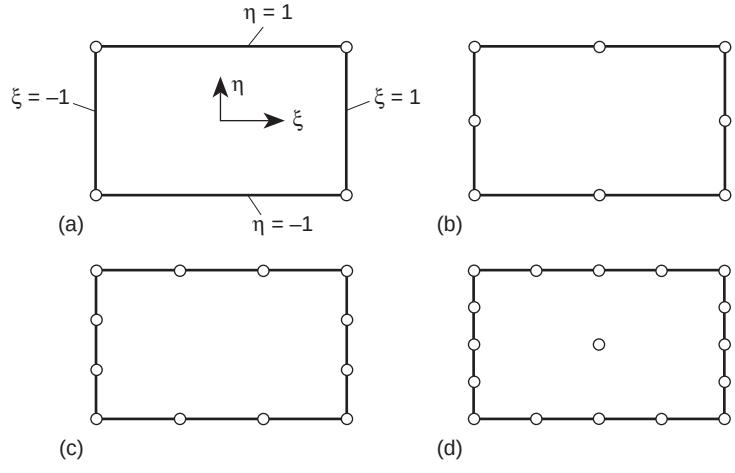


Fig. 8.9 Rectangles of boundary node (serendipity) family: (*a*) linear, (*b*) quadratic, (*c*) cubic, (*d*) quartic.

Mid-side nodes:

$$\xi_i = 0 \qquad N_i = \tfrac{1}{2}(1-\xi^2)(1+\eta_0)$$
$$\eta_i = 0 \qquad N_i = \tfrac{1}{2}(1+\xi_0)(1-\eta^2)$$

'Cubic' element

Corner nodes:

$$N_i = \tfrac{1}{32}(1+\xi_0)(1+\eta_0)[-10+9(\xi^2+\eta^2)] \tag{8.24}$$

Mid-side nodes:

$$\xi_i = \pm 1 \qquad \text{and} \qquad \eta_i = \pm\tfrac{1}{3}$$
$$N_i = \tfrac{9}{32}(1+\xi_0)(1-\eta^2)(1+9\eta_0)$$

with the remaining mid-side node expression obtained by changing variables.

In the next, quartic, member[8] of this family a central node is added so that all terms of a complete fourth-order expansion will be available. This central node adds a shape function $(1-\xi^2)(1-\eta^2)$ which is zero on all outer boundaries.

The above functions were originally derived by inspection, and progression to yet higher members is difficult and requires some ingenuity. It was therefore appropriate

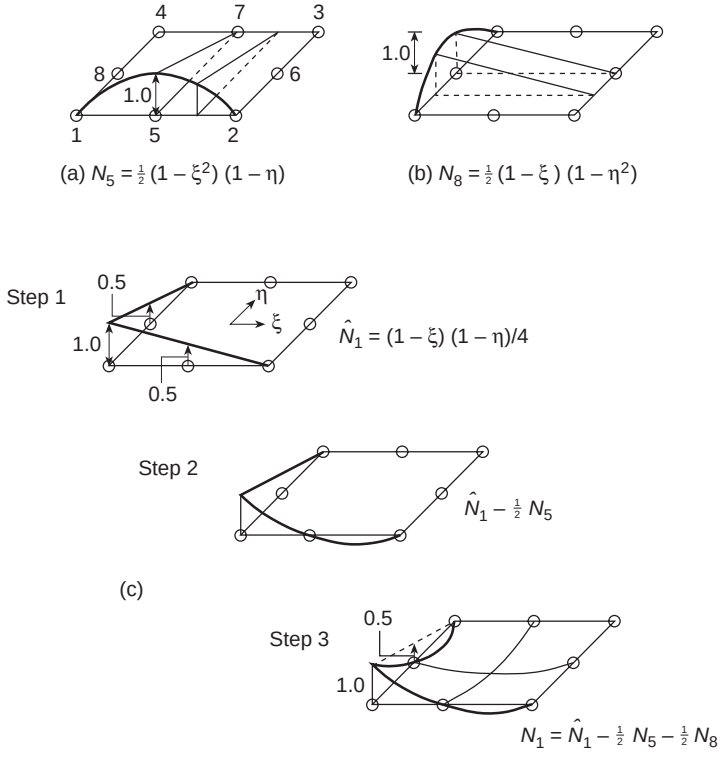


Fig. 8.10 Systematic generation of 'serendipity' shape functions.

to name this family 'serendipity' after the famous princes of Serendip noted for their chance discoveries (Horace Walpole, 1754).

However, a quite systematic way of generating the 'serendipity' shape functions can be devised, which becomes apparent from Fig. 8.10 where the generation of a quadratic shape function is presented.[7,9]

As a starting point we observe that for *mid-side* nodes a lagrangian interpolation of a quadratic $\times$ linear type suffices to determine N_i at nodes 5 to 8. N_5 and N_8 are shown at Fig. 8.10(*a*) and (*b*). For a *corner* node, such as Fig. 8.10(*c*), we start with a bilinear lagrangian family $\hat{N}_1$ and note immediately that while $\hat{N}_1 = 1$ at node 1, it is not zero at nodes 5 or 8 (step 1). Successive subtraction of $\frac{1}{2}N_5$ (step 2) and $\frac{1}{2}N_8$ (step 3) ensures that a zero value is obtained at these nodes. The reader can verify that the expressions obtained coincide with those of Eq. (8.23).

Indeed, it should now be obvious that for all higher order elements the *mid-side* and *corner shape* functions can be generated by an identical process. For the former a simple multiplication of *m*th-order and first-order lagrangian interpolations suffices. For the latter a combination of bilinear corner functions, together with appropriate fractions of mid-side shape functions to ensure zero at appropriate nodes, is necessary.

Similarly, it is quite easy to generate shape functions for elements with different numbers of nodes along each side by a systematic algorithm. This may be very

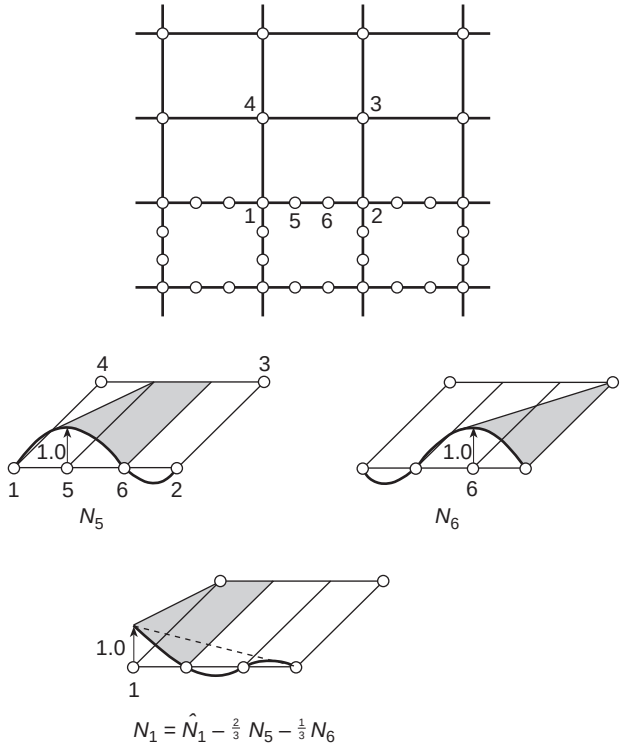


Fig. 8.11 Shape functions for a transition 'serendipity' element, cubic/linear.

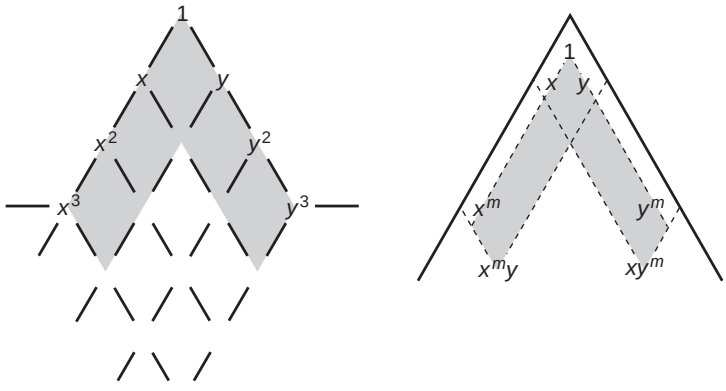


Fig. 8.12 Terms generated by edge shape functions in serendipity-type elements (3×3 and $m \times m$).

desirable if a transition between elements of different order is to be achieved, enabling a different order of accuracy in separate sections of a large problem to be studied. Figure 8.11 illustrates the necessary shape functions for a cubic/linear transition. Use of such special elements was first introduced in reference 9, but the simpler formulation used here is that of reference 7.

With the mode of generating shape functions for this class of elements available it is immediately obvious that fewer degrees of freedom are now necessary for a given complete polynomial expansion. Figure 8.12 shows this for a cubic element where only two surplus terms arise (as compared with six surplus terms in a lagrangian of the same degree).

It is immediately evident, however, that the functions generated by nodes placed only along the edges will not generate complete polynomials beyond cubic order. For higher order ones it is necessary to supplement the expansion by internal nodes (as was done in the quartic element of Fig. 8.9) or by the use of 'nodeless' variables which contain appropriate polynomial terms.

8.7 Elimination of internal variables before assembly – substructures

Internal nodes or nodeless internal parameters yield in the usual way the element properties (Chapter 2)

$$\frac{\partial \Pi^e}{\partial \mathbf{a}^e} = \mathbf{K}^e \mathbf{a}^e + \mathbf{f}^e \tag{8.25}$$

As $\mathbf{a}^e$ can be subdivided into parts which are common with other elements, $\bar{\mathbf{a}}^e$, and others which occur in the particular element only, $\bar{\bar{\mathbf{a}}}^e$, we can immediately write

$$\frac{\partial \Pi}{\partial \bar{\bar{\mathbf{a}}}^e} = \frac{\partial \Pi^e}{\partial \bar{\bar{\mathbf{a}}}^e} = \mathbf{0}$$

and eliminate $\bar{\bar{\mathbf{a}}}^e$ from further consideration. Writing Eq. (8.25) in a partitioned form we have

$$\frac{\partial \Pi^e}{\partial \mathbf{a}^e} = \begin{Bmatrix} \dfrac{\partial \Pi^e}{\partial \bar{\mathbf{a}}^e} \\ \dfrac{\partial \Pi^e}{\partial \bar{\bar{\mathbf{a}}}^e} \end{Bmatrix} = \begin{bmatrix} \bar{\mathbf{K}}^e & \hat{\mathbf{K}}^e \\ \hat{\mathbf{K}}^{e\mathrm{T}} & \bar{\bar{\mathbf{K}}}^e \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{a}}^e \\ \bar{\bar{\mathbf{a}}}^e \end{Bmatrix} + \begin{Bmatrix} \bar{\mathbf{f}}^e \\ \bar{\bar{\mathbf{f}}}^e \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial \Pi^e}{\partial \bar{\mathbf{a}}^e} \\ \mathbf{0} \end{Bmatrix} \tag{8.26}$$

From the second set of equations given above we can write

$$\bar{\bar{\mathbf{a}}}^e = -(\bar{\bar{\mathbf{K}}}^e)^{-1}(\hat{\mathbf{K}}^{e\mathrm{T}}\bar{\mathbf{a}}^e + \bar{\bar{\mathbf{f}}}^e) \tag{8.27}$$

which on substitution yields

$$\frac{\partial \Pi^e}{\partial \bar{\mathbf{a}}^e} = \mathbf{K}^{*e}\bar{\mathbf{a}}^e + \mathbf{f}^{*e} \tag{8.28}$$

in which

$$\begin{aligned} \mathbf{K}^{*e} &= \bar{\mathbf{K}}^e - \hat{\mathbf{K}}^e(\bar{\bar{\mathbf{K}}}^e)^{-1}\hat{\mathbf{K}}^{e\mathrm{T}} \\ \mathbf{f}^{*e} &= \bar{\mathbf{f}}^e - \hat{\mathbf{K}}^e(\bar{\bar{\mathbf{K}}}^e)^{-1}\bar{\bar{\mathbf{f}}}^e \end{aligned} \tag{8.29}$$

Assembly of the total region then follows, by considering only the element boundary variables, thus giving a considerable saving in the equation-solving effort at the expense of a few additional manipulations carried out at the element stage.

Perhaps a structural interpretation of this elimination is desirable. What in fact is involved is the separation of a part of the structure from its surroundings and determination of its solution separately for any prescribed displacements at the interconnecting boundaries. $\mathbf{K}^{*e}$ is now simply the overall stiffness of the separated structure and $\mathbf{f}^{*e}$ the equivalent set of nodal forces.

If the triangulation of Fig. 8.13 is interpreted as an assembly of pin-jointed bars the reader will recognize immediately the well-known device of 'substructures' used frequently in structural engineering.

Such a substructure is in fact simply a complex element from which the internal degrees of freedom have been eliminated.

Immediately a new possibility for devising more elaborate, and presumably more accurate, elements is presented.

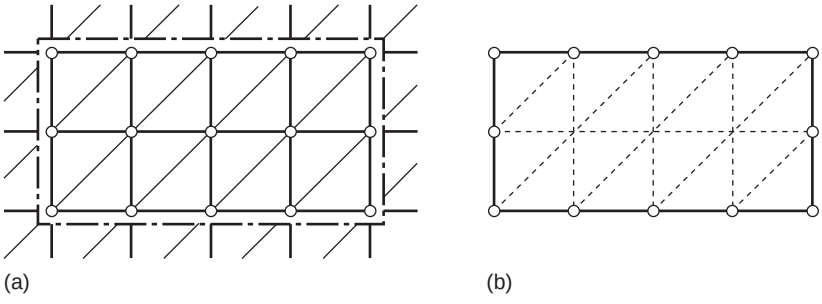


Fig. 8.13 Substructure of a complex element.

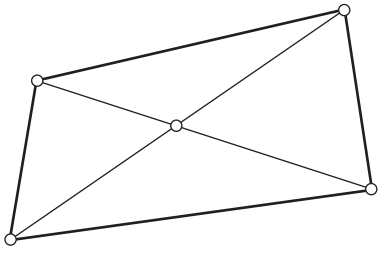


Fig. 8.14 A quadrilateral made up of four simple triangles.

Figure 8.13(*a*) can be interpreted as a continuum field subdivided into triangular elements. The substructure results in fact in one complex element shown in Fig. 8.13(*b*) with a number of boundary nodes.

The only difference from elements derived in previous sections is the fact that the unknown u is now not approximated internally by one set of smooth shape functions but by a series of piecewise approximations. This presumably results in a slightly poorer approximation but an economic advantage may arise if the total computation time for such an assembly is saved.

Substructuring is an important device in complex problems, particularly where a repetition of complicated components arises.

In simple, small-scale finite element analysis, much improved use of simple triangular elements was found by the use of simple subassemblies of the triangles (or indeed tetrahedra). For instance, a quadrilateral based on four triangles from which the central node is eliminated was found to give an economic advantage over direct use of simple triangles (Fig. 8.14). This and other subassemblies based on triangles are discussed in detail by Doherty *et al.*[10]

8.8 Triangular element family

The advantage of an arbitrary triangular shape in approximating to any boundary shape has been amply demonstrated in earlier chapters. Its apparent superiority here over rectangular shapes needs no further discussion. The question of generating more elaborate higher order elements needs to be further developed.

Consider a series of triangles generated on a pattern indicated in Fig. 8.15. The number of nodes in each member of the family is now such that a complete polynomial expansion, of the order needed for interelement compatibility, is ensured. This follows by comparison with the Pascal triangle of Fig. 8.5 in which we see the number of nodes coincides exactly with the number of polynomial terms required. This particular feature puts the triangle family in a special, privileged position, in which the inverse of the $\mathbf{C}$ matrices of Eq. (8.11) will always exist.[3] However, once again a direct generation of shape functions will be preferred – and indeed will be shown to be particularly easy.

Before proceeding further it is useful to define a special set of normalized coordinates for a triangle.

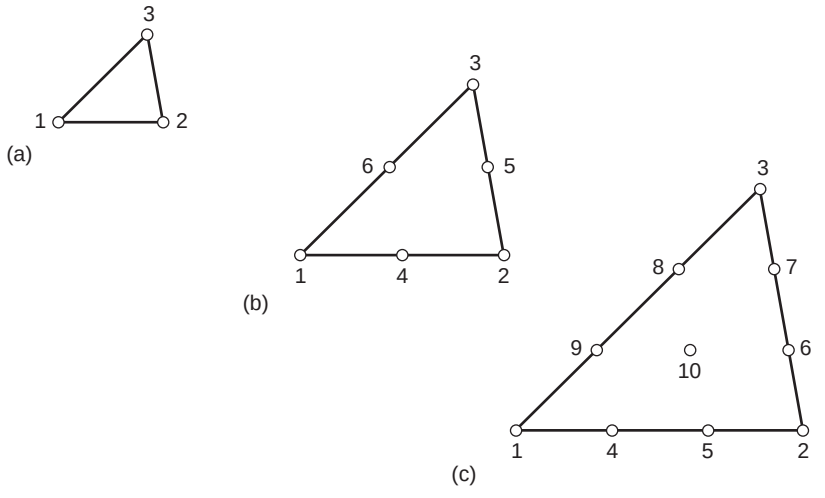


Fig. 8.15 Triangular element family: (*a*) linear, (*b*) quadratic, and (*c*) cubic.

8.8.1 Area coordinates

While cartesian directions parallel to the sides of a rectangle were a natural choice for that shape, in the triangle these are not convenient.

A new set of coordinates, L_1, L_2, and L_3 for a triangle 1, 2, 3 (Fig. 8.16), is defined by the following linear relation between these and the cartesian system:

$$\begin{aligned} x &= L_1 x_1 + L_2 x_2 + L_3 x_3 \\ y &= L_1 y_1 + L_2 y_2 + L_3 y_3 \\ 1 &= L_1 + L_2 + L_3 \end{aligned} \tag{8.30}$$

To every set, L_1, L_2, L_3 (which are not independent, but are related by the third equation), there corresponds a unique set of cartesian coordinates. At point 1, $L_1 = 1$ and $L_2 = L_3 = 0$, etc. A linear relation between the new and cartesian coordinates implies

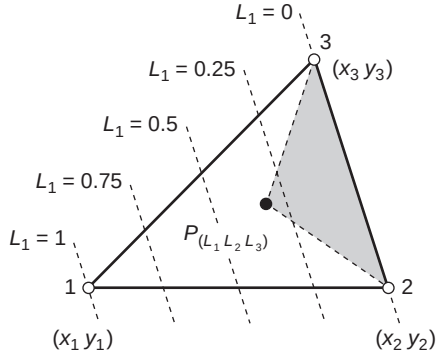


Fig. 8.16 Area coordinates.

that contours of L_1 are equally placed straight lines parallel to side 2–3 on which $L_1 = 0$, etc.

Indeed it is easy to see that an alternative definition of the coordinate L_1 of a point P is by a ratio of the area of the shaded triangle to that of the total triangle:

$$L_1 = \frac{\text{area } P23}{\text{area } 123} \tag{8.31}$$

Hence the name *area coordinates*.

Solving Eq. (8.30) gives

$$\begin{aligned} L_1 &= \frac{a_1 + b_1 x + c_1 y}{2\Delta} \\ L_2 &= \frac{a_2 + b_2 x + c_2 y}{2\Delta} \\ L_3 &= \frac{a_3 + b_3 x + c_3 y}{2\Delta} \end{aligned} \tag{8.32}$$

in which

$$\Delta = \tfrac{1}{2}\det \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \text{area } 123 \tag{8.33}$$

and

$$a_1 = x_2 y_3 - x_3 y_2 \qquad b_1 = y_2 - y_3 \qquad c_1 = x_3 - x_2$$

etc., with cyclic rotation of indices 1, 2, and 3.

The identity of expressions with those derived in Chapter 4 [Eqs (4.5b) and (4.5c)] is worth noting.

8.8.2 Shape functions

For the first element of the series [Fig. 8.15(*a*)], the shape functions are simply the area coordinates. Thus

$$N_1 = L_1 \qquad N_2 = L_2 \qquad N_3 = L_3 \tag{8.34}$$

This is obvious as each individually gives unity at one node, zero at others, and varies linearly everywhere.

To derive shape functions for other elements a simple recurrence relation can be derived.[3] However, it is very simple to write an arbitrary triangle of order M in a manner similar to that used for the lagrangian element of Sec. 8.5.

Denoting a typical node i by three numbers I, J, and K corresponding to the position of coordinates L_{1i}, L_{2i}, and L_{3i} we can write the shape function in terms of three lagrangian interpolations as [see Eq. (8.18)]

$$N_i = l_I^I(L_1) l_J^J(L_2) l_K^K(L_3) \tag{8.35}$$

In the above l_I^I, etc., are given by expression (8.18), with L_1 taking the place of ξ, etc.

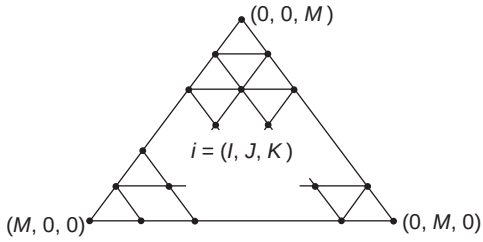


Fig. 8.17 A general triangular element.

It is easy to verify that the above expression gives

$$N_i = 1 \qquad \text{at} \quad L_1 = L_{1I}, \quad L_2 = L_{2J}, \quad L_3 = L_{3K}$$

and zero at all other nodes.

The highest term occurring in the expansion is

$$L_1^I L_2^J L_3^K$$

and as

$$I + J + K \equiv M$$

for all points the polynomial is also of order M.

Expression (8.35) is valid for quite arbitrary distributions of nodes of the pattern given in Fig. 8.17 and simplifies if the spacing of the nodal lines is equal (i.e., $1/m$). The formula was first obtained by Argyris *et al.*[11] and formalized in a different manner by others.[7,12]

The reader can verify the shape functions for the second- and third-order elements as given below and indeed derive ones of any higher order easily.

Quadratic triangle [Fig. 8.15(*b*)]

Corner nodes:

$$N_1 = (2L_1 - 1)L_1, \qquad \text{etc.}$$

Mid-side nodes:

$$N_4 = 4L_1L_2, \qquad \text{etc.}$$

Cubic triangle [Fig. 8.15(*c*)]

Corner nodes:

$$N_1 = \tfrac{1}{2}(3L_1 - 1)(3L_1 - 2)L_1, \qquad \text{etc.} \tag{8.36}$$

Mid-side nodes:

$$N_4 = \tfrac{9}{2}L_1L_2(3L_1 - 1), \qquad \text{etc.} \tag{8.37}$$

and for the internal node:

$$N_{10} = 27L_1L_2L_3$$

The last shape again is a 'bubble' function giving zero contribution along boundaries – and this will be found to be useful in many other contexts (see the mixed forms in Chapter 12).

The quadratic triangle was first derived by Veubeke[13] and used later in the context of plane stress analysis by Argyris.[14]

When element matrices have to be evaluated it will follow that we are faced with integration of quantities defined in terms of area coordinates over the triangular region. It is useful to note in this context the following exact integration expression:

$$\iint_{\Delta} L_1^a L_2^b L_3^c \, \mathrm{d}x \, \mathrm{d}y = \frac{a!\, b!\, c!}{(a+b+c+2)!} 2\Delta \tag{8.38}$$

One-dimensional elements

8.9 Line elements

So far in this book the continuum was considered generally in two or three dimensions. 'One-dimensional' members, being of a kind for which exact solutions are generally available, were treated only as trivial examples in Chapter 2 and in Sec. 8.2. In many practical two- or three-dimensional problems such elements do in fact appear in conjunction with the more usual continuum elements – and a unified treatment is desirable. In the context of elastic analysis these elements may represent lines of reinforcement (plane and three-dimensional problems) or sheets of thin lining material in axisymmetric bodies. In the context of field problems of the type discussed in Chapter 7 lines of drains in a porous medium of lesser conductivity can be envisaged.

Once the shape of such a function as displacement is chosen for an element of this kind, its properties can be determined, noting, however, that derived quantities such as strain, etc., have to be considered only in one dimension.

Figure 8.18 shows such an element sandwiched between two adjacent quadratic-type elements. Clearly for continuity of the function a quadratic variation of the

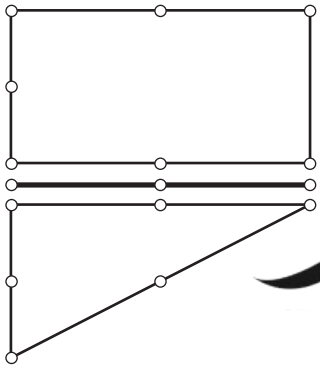


Fig. 8.18 A line element sandwiched between two-dimensional elements.

unknown with the one variable ξ is all that is required. Thus the shape functions are given directly by the Lagrange polynomial as defined in Eq. (8.18).

Three-dimensional elements

8.10 Rectangular prisms – Lagrange family

In a precisely analogous way to that given in previous sections equivalent elements of three-dimensional type can be described.

Now, for interelement continuity the simple rules given previously have to be modified. What is necessary to achieve is that along a whole face of an element the nodal values define a unique variation of the unknown function. With incomplete polynomials, this can be ensured only by inspection.

Shape function for such elements, illustrated in Fig. 8.19, will be generated by a direct product of three Lagrange polynomials. Extending the notation of Eq. (8.19) we now have

$$N_i \equiv N_{IJK} = l_I^n l_J^m l_K^p \tag{8.39}$$

for n, m, and p subdivisions along each side.

This element again is suggested by Zienkiewicz *et al.*[5] and elaborated upon by Argyris *et al.*[6] All the remarks about internal nodes and the properties of the formulation with mappings (to be described in the next chapter) are applicable here.

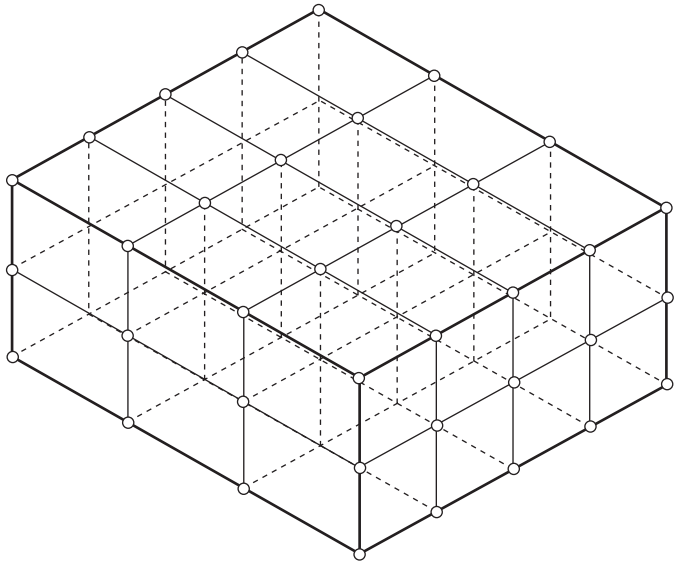


Fig. 8.19 Right prism of Lagrange family.

8.11 Rectangular prisms – 'serendipity' family[4,9,15]

The family of elements shown in Fig. 8.20 is precisely equivalent to that of Fig. 8.9. Using now three normalized coordinates and otherwise following the terminology of Sec. 8.6 we have the following shape functions:

'Linear' element (8 nodes)

$$N_i = \tfrac{1}{8}(1 + \xi_0)(1 + \eta_0)(1 + \zeta_0) \tag{8.40}$$

which is identical with the linear lagrangian element.

'Quadratic' element (20 nodes)

Corner nodes:

$$N_i = \tfrac{1}{8}(1 + \xi_0)(1 + \eta_0)(1 + \zeta_0)(\xi_0 + \eta_0 + \zeta_0 - 2) \tag{8.41}$$

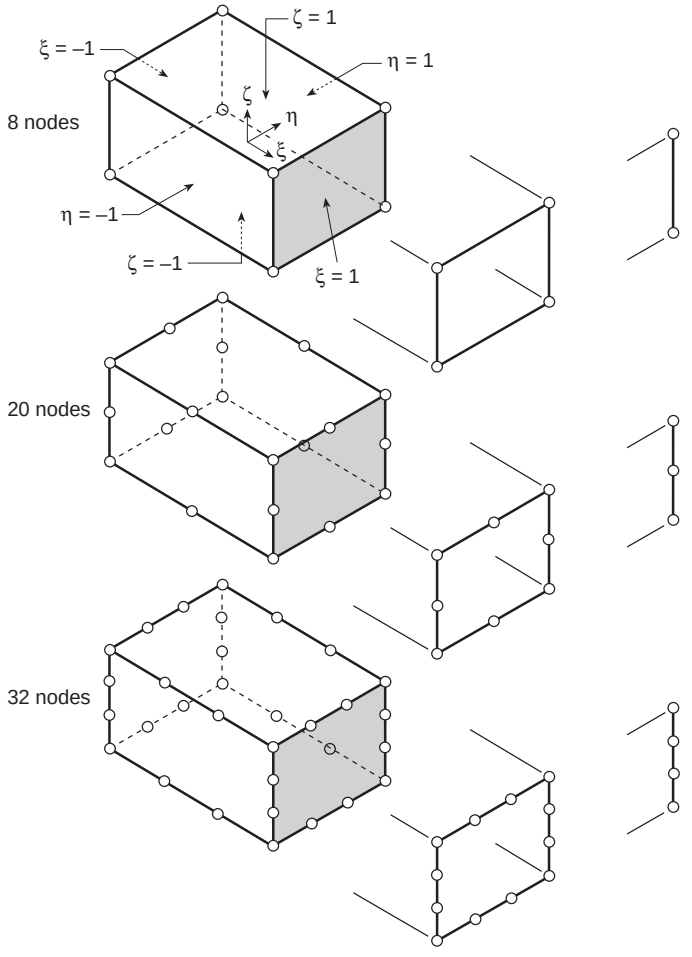


Fig. 8.20 Right prisms of boundary node (serendipity) family with corresponding sheet and line elements.

Typical mid-side node:

$$\xi_i = 0 \qquad \eta_i = \pm 1 \qquad \zeta_i = \pm 1$$
$$N_i = \tfrac{1}{4}(1-\xi^2)(1+\eta_0)(1+\zeta_0)$$

'Cubic' elements (32 nodes)

Corner node:

$$N_i = \tfrac{1}{64}(1+\xi_0)(1+\eta_0)(1+\zeta_0)[9(\xi^2+\eta^2+\zeta^2)-19] \tag{8.42}$$

Typical mid-side node:

$$\xi_i = \pm\tfrac{1}{3} \qquad \eta_i = \pm 1 \qquad \zeta_i = \pm 1$$
$$N_i = \tfrac{9}{64}(1-\xi^2)(1+9\xi_0)(1+\eta_0)(1+\zeta_0)$$

When $\zeta = 1 = \zeta_0$ the above expressions reduce to those of Eqs (8.22)–(8.24). Indeed such elements of three-dimensional type can be joined in a compatible manner to sheet or line elements of the appropriate type as shown in Fig. 8.20.

Once again the procedure for generating the shape functions follows that described in Figs 8.10 and 8.11 and once again elements with varying degrees of freedom along the edges can be derived following the same steps.

The equivalent of a Pascal triangle is now a tetrahedron and again we can observe the small number of surplus degrees of freedom – a situation of even greater magnitude than in two-dimensional analysis.

8.12 Tetrahedral elements

The tetrahedral family shown in Fig. 8.21 not surprisingly exhibits properties similar to those of the triangle family.

Firstly, once again complete polynomials in three coordinates are achieved at each stage. Secondly, as faces are divided in a manner identical with that of the previous triangles, the same order of polynomial in two coordinates in the plane of the face is achieved and element compatibility ensured. No surplus terms in the polynomial occur.

8.12.1 Volume coordinates

Once again special coordinates are introduced defined by (Fig. 8.22):

$$\begin{aligned} x &= L_1x_1 + L_2x_2 + L_3x_3 + L_4x_4 \\ y &= L_1y_1 + L_2y_2 + L_3y_3 + L_4y_4 \\ z &= L_1z_1 + L_2z_2 + L_3z_3 + L_4z_4 \\ 1 &= L_1 + L_2 + L_3 + L_4 \end{aligned} \tag{8.43}$$

Solving Eq. (8.43) gives

$$L_1 = \frac{a_1 + b_1x + c_1y + d_1z}{6V} \qquad \text{etc.}$$

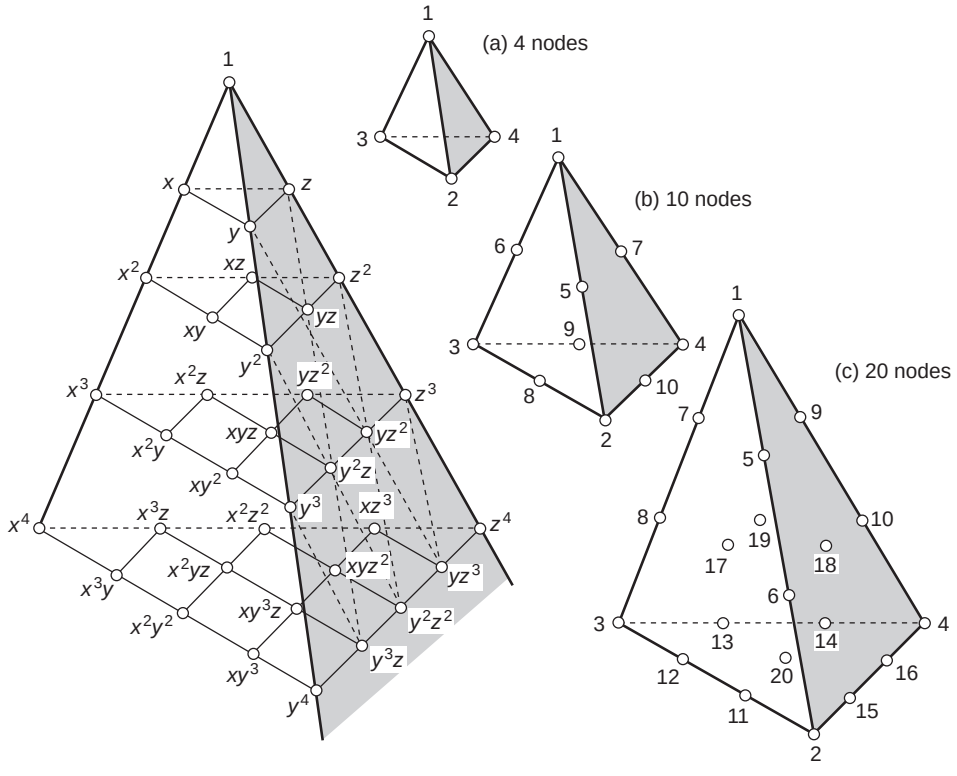


Fig. 8.21 The tetrahedron family: (*a*) linear, (*b*) quadratic, and (*c*) cubic.

where the constants can be identified from Chapter 6, Eq. (6.5). Again the physical nature of the coordinates can be identified as the ratio of volumes of tetrahedra based on an internal point P in the total volume, e.g., as shown in Fig. 8.22:

$$L_1 = \frac{\text{volume } P234}{\text{volume } 1234}, \qquad \text{etc.} \tag{8.44}$$

8.12.2 Shape function

As the volume coordinates vary linearly with the cartesian ones from unity at one node to zero at the opposite face then shape functions for the linear element [Fig. 8.21(*a*)] are simply

$$N_1 = L_1 \qquad N_2 = L_2, \qquad \text{etc.} \tag{8.45}$$

Formulae for shape functions of higher order tetrahedra are derived in precisely the same manner as for the triangles by establishing appropriate Lagrange-type formulae similar to Eq. (8.35). Leaving this to the reader as a suitable exercise we quote the following:

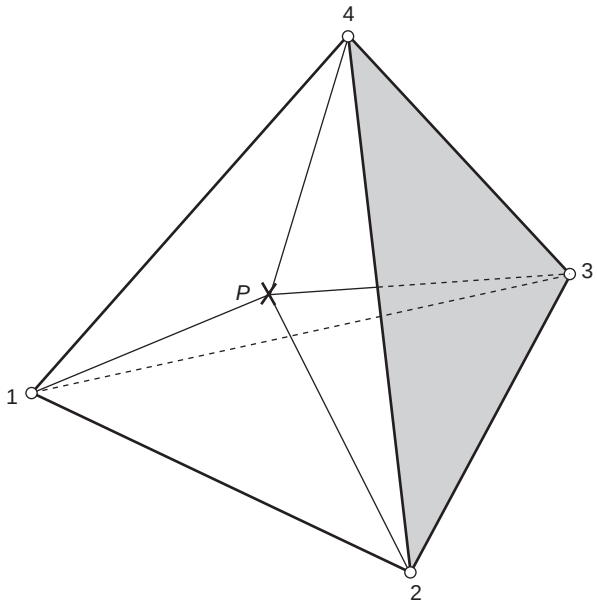


Fig. 8.22 Volume coordinates.

'Quadratic' tetrahedron [Fig. 8.21(b)]

For corner nodes:

$$N_1 = (2L_1 - 1)L_1, \qquad \text{etc.} \tag{8.46}$$

For mid-edge nodes:

$$N_5 = 4L_1L_2, \qquad \text{etc.}$$

'Cubic' tetrahedron

Corner nodes:

$$N_1 = \tfrac{1}{2}(3L_1 - 1)(3L_1 - 2)L_1, \qquad \text{etc.} \tag{8.47}$$

Mid-edge nodes:

$$N_5 = \tfrac{9}{2}L_1L_2(3L_1 - 1), \qquad \text{etc.}$$

Mid-face nodes:

$$N_{17} = 27L_1L_2L_3, \qquad \text{etc.}$$

A useful integration formula may again be here quoted:

$$\int\int\int_{\text{vol}} L_1^a L_2^b L_3^c L_4^d \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{a!\,b!\,c!\,d!}{(a+b+c+d+3)!}\,6V \tag{8.48}$$

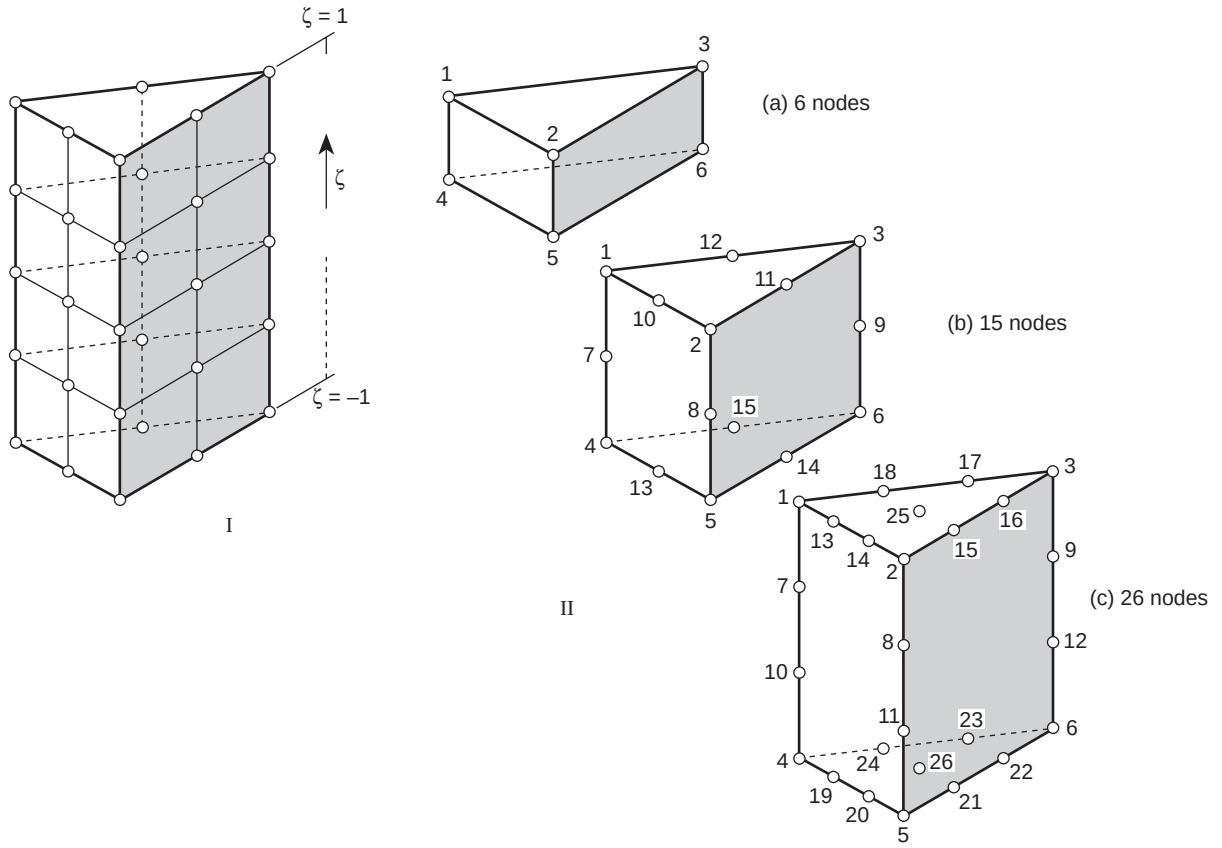


Fig. 8.23 Triangular prism elements (serendipity) family: (*a*) linear, (*b*) quadratic, and (*c*) cubic.

8.13 Other simple three-dimensional elements

The possibilities of simple shapes in three dimensions are greater, for obvious reasons, than in two dimensions. A quite useful series of elements can, for instance, be based on triangular prisms (Fig. 8.23). Here again variants of the product, Lagrange, approach or of the 'serendipity' type can be distinguished. The first element of both families is identical and indeed the shape functions for it are so obvious as not to need quoting.

For the 'quadratic' element illustrated in Fig. 8.23(*b*) the shape functions are

Corner nodes $L_1 = \zeta_1 = 1$:

$$N_1 = \tfrac{1}{2}L_1(2L_1 - 1)(1 + \zeta) - \tfrac{1}{2}L_1(1 - \zeta^2) \tag{8.49}$$

Mid-edge of triangles:

$$N_{10} = 2L_1L_2(1 + \zeta), \qquad \text{etc.} \tag{8.50}$$

Mid-edge of rectangle:

$$N_7 = L_1(1 - \zeta^2), \qquad \text{etc.}$$

Such elements are not purely esoteric but have a practical application as 'fillers' in conjunction with 20-noded serendipity elements.

Part 2 Hierarchical shape functions

8.14 Hierarchic polynomials in one dimension

The general ideas of hiearchic approximation were introduced in Sect. 8.2 in the context of simple, linear, elements. The idea of generating higher order hierarchic forms is again simple. We shall start from a one-dimensional expansion as this has been shown to provide a basis for the generation of two- and three-dimensional forms in previous sections.

To generate a polynomial of order p along an element side we do not need to introduce nodes but can instead use parameters without an obvious physical meaning. As shown in Fig. 8.24, we could use here a linear expansion specified by 'standard' functions N_0 and N_1 and add to this a series of polynomials always designed so as to have zero values at the ends of the range (i.e. points 0 and 1).

Thus for a quadratic approximation, we would write over the typical one-dimensional element, for instance,

$$\hat{u} = u_0N_0 + u_1N_1 + a_2N_2 \tag{8.51}$$

where

$$N_0 = -\frac{\xi - 1}{2} \qquad N_1 = \frac{\xi + 1}{2} \qquad N_2 = -(\xi - 1)(\xi + 1) \tag{8.52}$$

using in the above the normalized x-coordinate [viz. Eq. (8.17)].

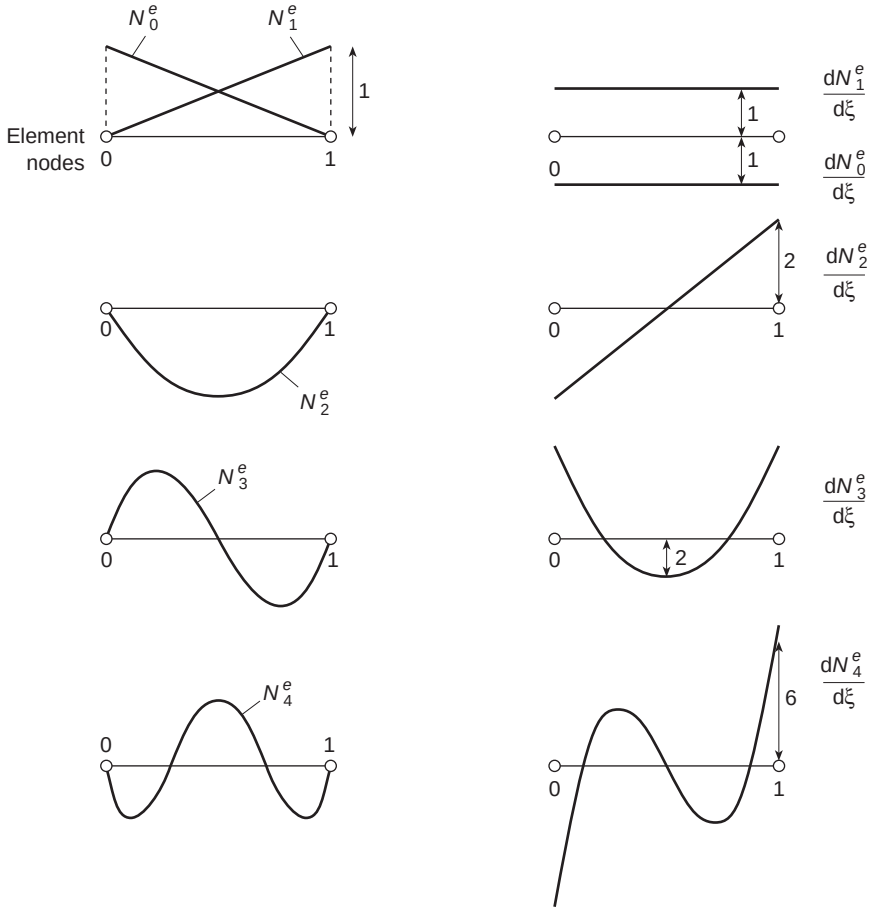


Fig. 8.24 Hierarchical element shape functions of nearly orthogonal form and their derivatives.

We note that the parameter a_2 does in fact have a meaning in this case as it is the magnitude of the departure from linearity of the approximation $\hat{u}$ at the element centre, since N_2 has been chosen here to have the value of unity at that point.

In a similar manner, for a cubic element we simply have to add $a_3 N_3$ to the quadratic expansion of Eq. (8.51), where N_3 is any cubic of the form

$$N_3^e = \alpha_0 + \alpha_1 \xi + \alpha_2 \xi^2 + \alpha_3 \xi^3 \tag{8.53}$$

and which has zero values at $\xi = \pm 1$ (i.e., at nodes 0 and 1). Again an infinity of choices exists, and we could select a cubic of a simple form which has a zero value at the centre of the element and for which $\mathrm{d}N_3/\mathrm{d}\xi = 1$ at the same point. Immediately we can write

$$N_3^e = \xi(1 - \xi^2) \tag{8.54}$$

as the cubic function with the desired properties. Now the parameter a_3 denotes the departure of the slope at the centre of the element from that of the first approximation.

We note that we could proceed in a similar manner and define the fourth-order hierarchical element shape function as

$$N_4^e = \xi^2(1-\xi^2) \tag{8.55}$$

but a physical identification of the parameter associated with this now becomes more difficult (even though it is not strictly necessary).

As we have already noted, the above set is not unique and many other possibilities exist. An alternative convenient form for the hierarchical functions is defined by

$$N_p^e(\xi) = \begin{cases} \dfrac{1}{p!}(\xi^p - 1) & p \text{ even} \\ \dfrac{1}{p!}(\xi^p - \xi) & p \text{ odd} \end{cases} \tag{8.56}$$

where p ($\geqslant 2$) is the degree of the introduced polynomial.[16] This yields the set of shape functions:

$$\begin{aligned} N_2^e &= \tfrac{1}{2}(\xi^2 - 1) \qquad & N_3^e &= \tfrac{1}{6}(\xi^3 - \xi) \\ N_4^e &= \tfrac{1}{24}(\xi^4 - 1) \qquad & N_5^e &= \tfrac{1}{120}(\xi^5 - \xi) \qquad \text{etc.} \end{aligned} \tag{8.57}$$

We observe that all derivatives of N_p^e of second or higher order have the value zero at $\xi = 0$, apart from $\mathrm{d}^p N_p^e/\mathrm{d}\xi^p$, which equals unity at that point, and hence, when shape functions of the form given by Eq. (8.57) are used, we can identify the parameters in the approximation as

$$a_p^e = \left.\frac{\mathrm{d}^p \hat{u}}{\mathrm{d}\xi^p}\right|_{\xi=0} \qquad p \geqslant 2 \tag{8.58}$$

This identification gives a general physical significance but is by no means necessary.

In two- and three-dimensional elements a simple identification of the hierarchic parameters on interfaces will automatically ensure C_0 continuity of the approximation.

As mentioned previously, an optimal form of hierarchical function is one that results in a diagonal equation system. This can on occasion be achieved, or at least approximated, quite closely.

In the elasticity problems which we have discussed in the preceding chapters the element matrix $\mathbf{K}^e$ possesses terms of the form [using Eq. (8.17)]

$$K_{lm}^e = \int_{\Omega^e} k \frac{\mathrm{d}N_l^e}{\mathrm{d}x}\frac{\mathrm{d}N_m^e}{\mathrm{d}x}\,\mathrm{d}x = \frac{1}{a}\int_{-1}^{1} k \frac{\mathrm{d}N_l^e}{\mathrm{d}\xi}\frac{\mathrm{d}N_m^e}{\mathrm{d}\xi}\,\mathrm{d}\xi \tag{8.59}$$

If shape function sets containing the appropriate polynomials can be found for which such integrals are zero for $l \neq m$, then orthogonality is achieved and the coupling between successive solutions disappears.

One set of polynomial functions which is known to possess this orthogonality property over the range $-1 \leqslant \xi \leqslant 1$ is the set of Legendre polynomials $P_p(\xi)$, and the shape functions could be defined in terms of integrals of these polynomials.[9] Here we define the Legendre polynomial of degree p by

$$P_p(\xi) = \frac{1}{(p-1)!}\frac{1}{2^{p-1}}\frac{\mathrm{d}^p}{\mathrm{d}\xi^p}[(\xi^2-1)^p] \tag{8.60}$$

and integrate these polynomials to define

$$N^e_{p+1} = \int P_p(\xi)\,\mathrm{d}\xi = \frac{1}{(p-1)!\,2^{p-1}} \frac{\mathrm{d}^{p-1}}{\mathrm{d}\xi^{p-1}}[(\xi^2-1)^p] \tag{8.61}$$

Evaluation for each p in turn gives

$$N^e_2 = \xi^2 - 1 \qquad N^e_3 = 2(\xi^3 - \xi) \qquad \text{etc.}$$

These differ from the element shape functions given by Eq. (8.57) only by a multiplying constant up to N^e_3, but for $p \geqslant 3$ the differences become significant. The reader can easily verify the orthogonality of the derivatives of these functions, which is useful in computation. A plot of these functions and their derivatives is given in Fig. 8.24.

8.15 Two- and three-dimensional, hierarchic, elements of the 'rectangle' or 'brick' type

In deriving 'standard' finite element approximations we have shown that all shape functions for the Lagrange family could be obtained by a simple multiplication of one-dimensional ones and those for serendipity elements by a combination of such multiplications. The situation is even simpler for hierarchic elements. Here *all* the shape functions can be obtained by a simple multiplication process.

Thus, for instance, in Fig. 8.25 we show the shape functions for a lagrangian nine-noded element and the corresponding hierarchical functions. The latter not only have simpler shapes but are more easily calculated, being simple products of linear and quadratic terms of Eq. (8.56), (8.57), or (8.61). Using the last of these the three functions illustrated are simply

$$\begin{aligned} N_1 &= (1-\xi)(1+\eta)/4 \\ N_2 &= (1-\xi)(1-\eta^2)/2 \\ N_3 &= (1-\xi^2)(1-\eta^2) \end{aligned} \tag{8.62}$$

The distinction between lagrangian and serendipity forms now disappears as for the latter in the present case the last shape function (N_3) is simply omitted.

Indeed, it is now easy to introduce interpolation for elements of the type illustrated in Fig. 8.11 in which a different expansion is used along different sides. This essential characteristic of hierarchical elements is exploited in adaptive refinement (viz. Chapter 15) where new degrees of freedom (or polynomial order increase) is made only when required by the magnitude of the error.

8.16 Triangle and tetrahedron family[16,17]

Once again the concepts of multiplication can be introduced in terms of area (volume) coordinates.

Returning to the triangle of Fig. 8.16 we note that along the side 1–2, L_3 is identically zero, and therefore we have

$$(L_1 + L_2)_{1-2} = 1 \tag{8.63}$$

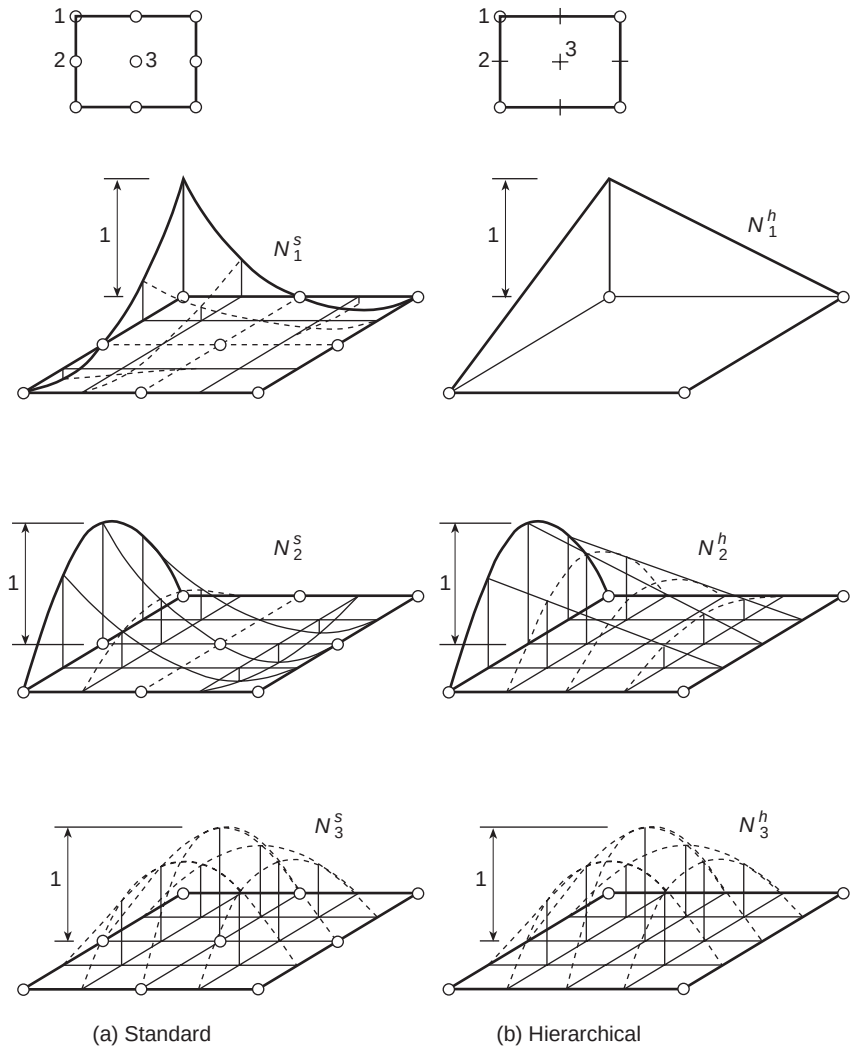


Fig. 8.25 Standard and hierarchic shape functions corresponding to a lagrangian, quadratic element.

If ξ, measured along side 1–2, is the usual non-dimensional local element coordinate of the type we have used in deriving hierarchical functions for one-dimensional elements, we can write

$$L_1|_{1-2} = \tfrac{1}{2}(1-\xi) \qquad L_2|_{1-2} = \tfrac{1}{2}(1+\xi) \tag{8.64}$$

from which it follows that we have

$$\xi = (L_2 - L_1)_{1-2} \tag{8.65}$$

This suggests that we could generate hierarchical shape functions over the triangle by generalizing the one-dimensional shape function forms produced earlier. For

example, using the expressions of Eq. (8.56), we associate with the side 1–2 the polynomial of degree p ($\geqslant 2$) defined by

$$N^e_{p(1-2)} = \begin{cases} \dfrac{1}{p!}[(L_2 - L_1)^p - (L_1 + L_2)^p] & p \text{ even} \\ \dfrac{1}{p!}[(L_2 - L_1)^p - (L_2 - L_1)(L_1 + L_2)^{p-1}] & p \text{ odd} \end{cases} \tag{8.66}$$

It follows from Eq. (8.64) that these shape functions are zero at nodes 1 and 2. In addition, it can easily be shown that $N^e_{p(1-2)}$ will be zero all along the sides 3–1 and 3–2 of the triangle, and so C_0 continuity of the approximation $\hat{u}$ is assured.

It should be noted that in this case for $p \geqslant 3$ the number of hierarchical functions arising from the element sides in this manner is insufficient to define a complete polynomial of degree p, and internal hierarchical functions, which are identically zero on the boundaries, need to be introduced; for example, for $p = 3$ the function $L_1L_2L_3$ could be used, while for $p = 4$ the three additional functions $L_1^2L_2L_3$, $L_1L_2^2L_3$, $L_1L_2L_3^2$ could be adopted.

In Fig. 8.26 typical hierarchical linear, quadratic, and cubic trial functions for a triangular element are shown. Similar hierarchical shape functions could be generated

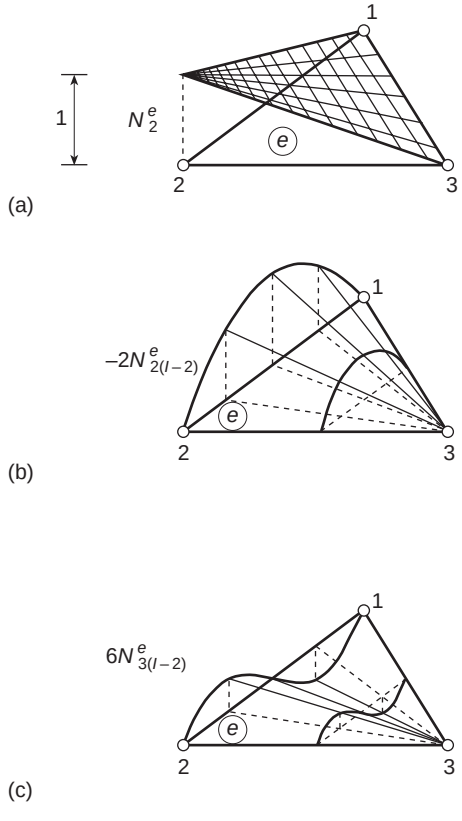


Fig. 8.26 Triangular elements and associated hierarchical shape functions of (*a*) linear, (*b*) quadratic, and (*c*) cubic form.

from the alternative set of one-dimensional shape functions defined in Eq. (8.61). Identical procedures are obvious in the context of tetrahedra.

8.17 Global and local finite element approximation

The very concept of hierarchic approximations (in which the shape functions are not affected by the refinement) means that it is possible to include in the expansion

$$u = \sum_{i=1}^{n} N_i a_i \tag{8.67}$$

functions N which are not local in nature. Such functions may, for instance, be the exact solutions of an analytical problem which in some way resembles the problem dealt with, but do not satisfy some boundary or inhomogeneity conditions. The 'finite element', local, expansions would here be a device for correcting this solution to satisfy the real conditions. This use of the global–local approximation was first suggested by Mote[18] in a problem where the coefficients of this function were fixed. The example involved here is that of a rotating disc with cutouts (Fig. 8.27). The global, known, solution is the analytical one corresponding to a disc without cutout, and finite elements are added locally to modify the solution. Other examples of such 'fixed' solutions may well be those associated with point loads, where the use of the global approximation serves to eliminate the singularity modelled badly by the discretization.

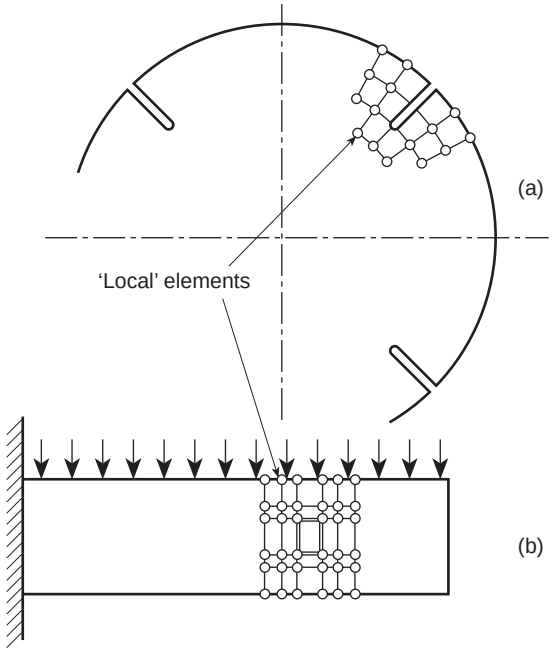


Fig. 8.27 Some possible uses of the local–global approximation: (a) rotating slotted disc, (b) perforated beam.

In some problems the singularity itself is unknown and the appropriate function can be added with an unknown coefficient.

8.18 Improvement of conditioning with hierarchic forms

We have already mentioned that hierarchic element forms give a much improved equation conditioning for steady-state (static) problems due to their form which is more nearly diagonal. In Fig. 8.28 we show the 'condition number' (which is a measure of such diagonality and is defined in standard texts on linear algebra; see Appendix A) for a single cubic element and for an assembly of four cubic elements, using standard and hierarchic forms in their formulation. The improvement of the conditioning is a distinct advantage of such forms and allows the use of iterative solution techniques to be more easily adopted.[19] Unfortunately much of this advantage disappears for transient analysis as the approximation must contain specific modes (see Chapter 17).

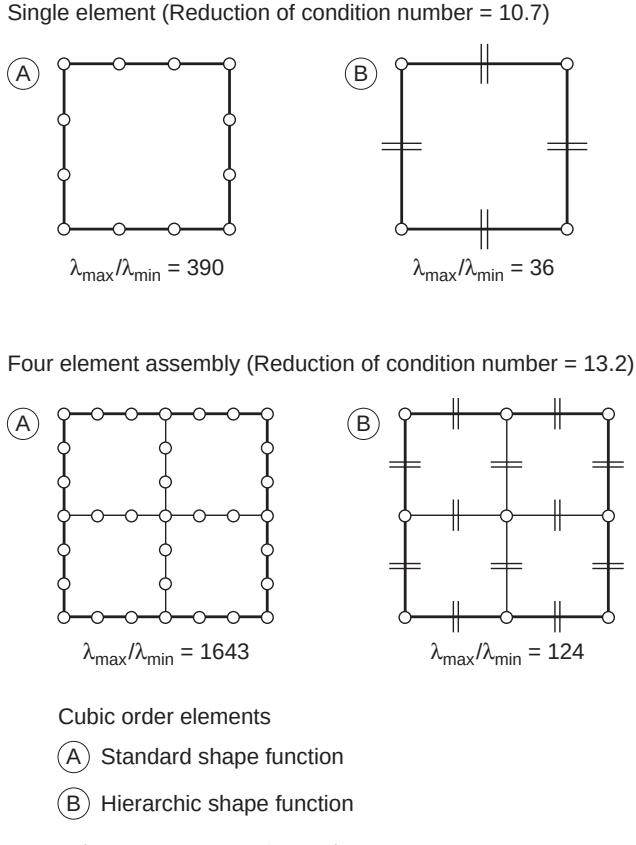


Fig. 8.28 Improvement of condition number (ratio of maximum to minimum eigenvalue of the stiffness matrix) by use of a hierarchic form (elasticity isotropic $\nu = 0.15$).

8.19 Concluding remarks

An unlimited selection of element types has been presented here to the reader – and indeed equally unlimited alternative possibilities exist.[4,9] What of the use of such complex elements in practice? The triangular and tetrahedral elements are limited to situations where the real region is of a suitable shape which can be represented as an assembly of flat facets and all other elements are limited to situations represented by an assembly of right prisms. Such a limitation would be so severe that little practical purpose would have been served by the derivation of such shape functions unless some way could be found of distorting these elements to fit realistic curved boundaries. In fact, methods for doing this are available and will be described in the next chapter.

References

1. W. Rudin. *Principles of Mathematical Analysis*. 3rd ed, McGraw-Hill, 1976.
2. P.C. Dunne. Complete polynomial displacement fields for finite element methods. *Trans. Roy. Aero. Soc.* **72**, 245, 1968.
3. B.M. Irons, J.G. Ergatoudis, and O.C. Zienkiewicz. Comment on ref. 1. *Trans. Roy. Aero. Soc.* **72**, 709–11, 1968.
4. J.G. Ergatoudis, B.M. Irons, and O.C. Zienkiewicz. Curved, isoparametric, quadrilateral elements for finite element analysis. *Int. J. Solids Struct.* **4**, 31–42, 1968.
5. O.C. Zienkiewicz *et al.* Iso-parametric and associated elements families for two and three dimensional analysis. Chapter 13 of *Finite Element Methods in Stress Analysis* (eds I. Holand and K. Bell), Tech. Univ. of Norway, Tapir Press, Norway, Trondheim, 1969.
6. J.H. Argyris, K.E. Buck, H.M. Hilber, G. Mareczek, and D.W. Scharpf. Some new elements for matrix displacement methods. *2nd Conf. on Matrix Methods in Struct. Mech.* Air Force Inst. of Techn., Wright Patterson Base, Ohio, Oct. 1968.
7. R.L. Taylor. On completeness of shape functions for finite element analysis. *Int. J. Num. Meth. Eng.* **4**, 17–22, 1972.
8. F.C. Scott. A quartic, two dimensional isoparametric element. Undergraduate Project, Univ. of Wales, Swansea, 1968.
9. O.C. Zienkiewicz, B.M. Irons, J. Campbell, and F.C. Scott. Three dimensional stress analysis. *Int. Un. Th. Appl. Mech. Symposium on High Speed Computing in Elasticity*. Liége, 1970.
10. W.P. Doherty, E.L. Wilson, and R.L. Taylor. *Stress Analysis of Axisymmetric Solids Utilizing Higher-Order Quadrilateral Finite Elements*. Report 69–3, Structural Engineering Laboratory, Univ. of California, Berkeley, Jan. 1969.
11. J.H. Argyris, I. Fried, and D.W. Scharpf. The TET 20 and the TEA 8 elements for the matrix displacement method. *Aero. J.* **72**, 618–25, 1968.
12. P. Silvester. Higher order polynomial triangular finite elements for potential problems. *Int. J. Eng. Sci.* **7**, 849–61, 1969.
13. B. Fraeijs de Veubeke. Displacement and equilibrium models in the finite element method. Chapter 9 of *Stress Analysis* (eds O.C. Zienkiewicz and G.S. Holister), Wiley, 1965.
14. J.H. Argyris. Triangular elements with linearly varying strain for the matrix displacement method. *J. Roy. Aero. Soc. Tech. Note.* **69**, 711–13, Oct. 1965.

15. J.G. Ergatoudis, B.M. Irons, and O.C. Zienkiewicz. Three dimensional analysis of arch dams and their foundations. *Symposium on Arch Dams*. Inst. Civ. Eng., London, 1968.
16. A.G. Peano. Hierarchics of conforming finite elements for elasticity and plate bending. *Comp. Math. and Applications*. **2**, 3–4, 1976.
17. J.P. de S.R. Gago. *A posteri error analysis and adaptivity for the finite element method*. Ph.D thesis, University of Wales, Swansea, 1982.
18. C.D. Mote, Global–local finite element. *Int. J. Num. Meth. Eng*. **3**, 565–74, 1971.
19. O.C. Zienkiewicz, J.P. de S.R. Gago, and D.W. Kelly. The hierarchical concept in finite element analysis. *Computers and Structures*. **16**, 53–65, 1983.

9

Mapped elements and numerical integration – 'infinite' and 'singularity' elements

9.1 Introduction

In the previous chapter we have shown how some general families of finite elements can be obtained for C_0 interpolations. A progressively increasing number of nodes and hence improved accuracy characterizes each new member of the family and presumably the number of such elements required to obtain an adequate solution decreases rapidly. To ensure that a small number of elements can represent a relatively complex form of the type that is liable to occur in real, rather than academic, problems, simple rectangles and triangles no longer suffice. This chapter is therefore concerned with the subject of distorting such simple forms into others of more arbitrary shape.

Elements of the basic one-, two-, or three-dimensional types will be 'mapped' into distorted forms in the manner indicated in Figs 9.1 and 9.2.

In these figures it is shown that the ξ, η, ζ, or $L_1L_2L_3L_4$ coordinates can be distorted to a new, curvilinear set when plotted in cartesian x, y, z space.

Not only can two-dimensional elements be distorted into others in two dimensions but the mapping of these can be taken into three dimensions as indicated by the flat sheet elements of Fig. 9.2 distorting into a three-dimensional space. This principle applies generally, providing a one-to-one correspondence between cartesian and curvilinear coordinates can be established, i.e., once the mapping relations of the type

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} f_x(\xi,\eta,\zeta) \\ f_y(\xi,\eta,\zeta) \\ f_z(\xi,\eta,\zeta) \end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix} f_x(L_1,L_2,L_3,L_4) \\ f_y(L_1,L_2,L_3,L_4) \\ f_z(L_1,L_2,L_3,L_4) \end{Bmatrix} \tag{9.1}$$

can be established.

Once such coordinate relationships are known, shape functions can be specified in local coordinates and by suitable transformations the element properties established in the global coordinate system.

In what follows we shall first discuss the so-called isoparametric form of relationship (9.1) which has found a great deal of practical application. Full details of this formulation will be given, including the establishment of element matrices by numerical integration.

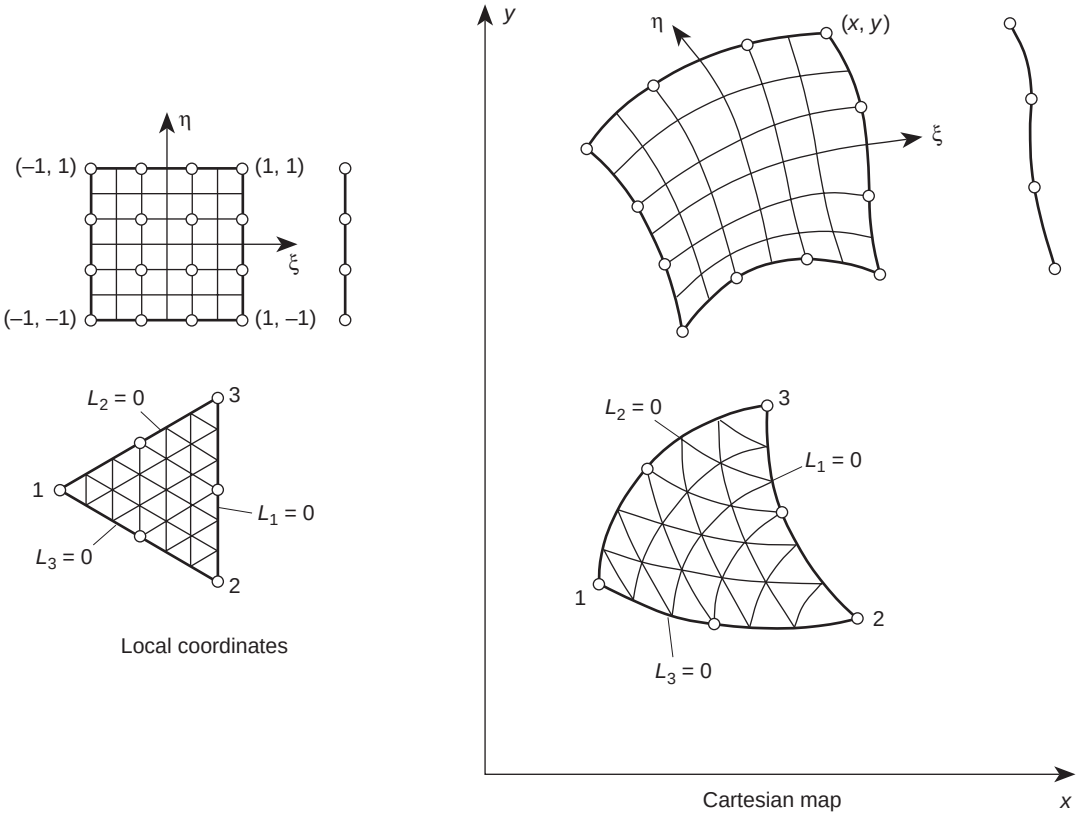


Fig. 9.1 Two-dimensional 'mapping' of some elements.

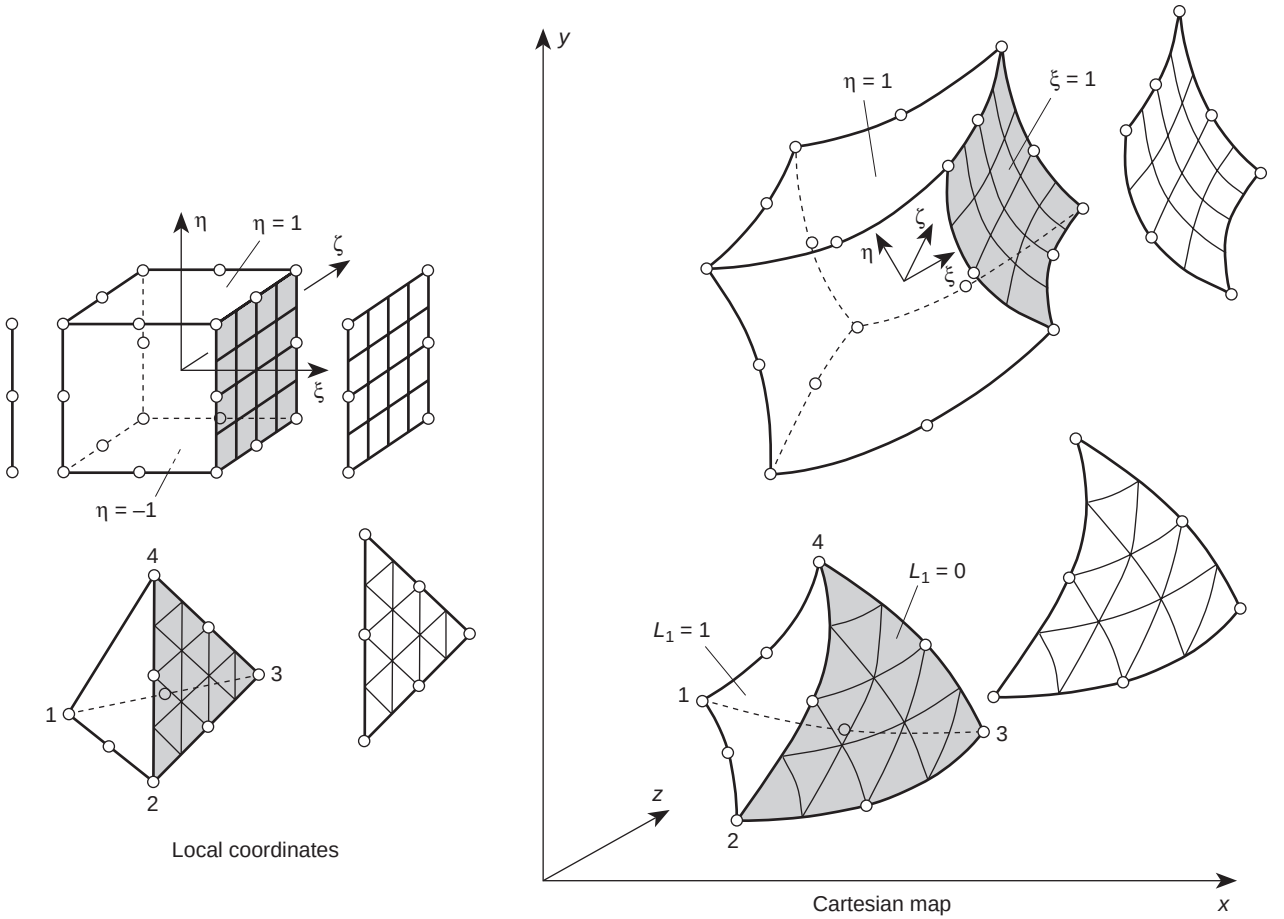


Fig. 9.2 Three-dimensional 'mapping' of some elements.

In the final section we shall show that many other coordinate transformations can be used effectively.

Parametric curvilinear coordinates

9.2 Use of 'shape functions' in the establishment of coordinate transformations

A most convenient method of establishing the coordinate transformations is to use the 'standard' type of C_0 shape functions we have already derived to represent the variation of the unknown function.

If we write, for instance, for each element

$$x = N_1' x_1 + N_2' x_2 + \cdots = \mathbf{N}' \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \end{Bmatrix} = \mathbf{N}'\mathbf{x}$$

$$y = N_1' y_1 + N_2' y_2 + \cdots = \mathbf{N}' \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \end{Bmatrix} = \mathbf{N}'\mathbf{y} \tag{9.2}$$

$$z = N_1' z_1 + N_2' z_2 = \cdots = \mathbf{N}' \begin{Bmatrix} z_1 \\ z_2 \\ \vdots \end{Bmatrix} = \mathbf{N}'\mathbf{z}$$

in which $\mathbf{N}'$ are standard shape functions given in terms of the local coordinates, then a relationship of the required form is immediately available. Further, the points with coordinates x_1, y_1, z_1, etc., will lie at appropriate points of the element boundary (as from the general definitions of the standard shape functions we know that these have a value of unity at the point in question and zero elsewhere). These points can establish nodes *a priori*.

To each set of local coordinates there will correspond a set of global cartesian coordinates and in general only one such set. We shall see, however, that a non-uniqueness may arise sometimes with violent distortion.

The concept of using such element shape functions for establishing curvilinear coordinates in the context of finite element analysis appears to have been first introduced by Taig.[1] In his first application basic linear quadrilateral relations were used. Irons[2,3] generalized the idea for other elements.

Quite independently the exercises of devising various practical methods of generating curved surfaces for purposes of engineering design led to the establishment of similar definitions by Coons[4] and Forrest,[5] and indeed today the subjects of surface definitions and analysis are drawing closer together due to this activity.

In Fig. 9.3 an actual distortion of elements based on the cubic and quadratic members of the two-dimensional 'serendipity' family is shown. It is seen here that a one-to-one relationship exists between the local (ξ, η) and global (x, y) coordinates.

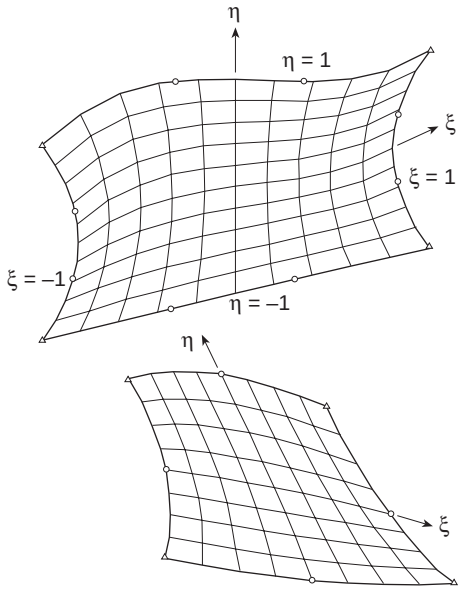


Fig. 9.3 Computer plots of curvilinear coordinates for cubic and parabolic elements (reasonable distortion).

If the fixed points are such that a violent distortion occurs then a non-uniqueness can occur in the manner indicated for two situations in Fig. 9.4. Here at internal points of the distorted element two or more local coordinates correspond to the same cartesian coordinate and in addition to some internal points being mapped outside the element. Care must be taken in practice to avoid such gross distortion.

Figure 9.5 shows two examples of a two-dimensional (ξ, η) element mapped into a three-dimensional (x, y, z) space.

We shall often refer to the basic element in undistorted, local, coordinates as a 'parent' element.

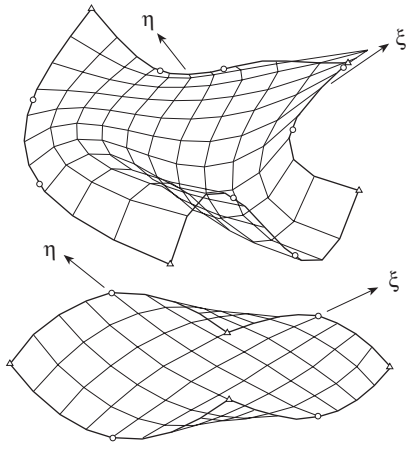


Fig. 9.4 Unreasonable element distortion leading to a non-unique mapping and 'overspill'. Cubic and parabolic elements.

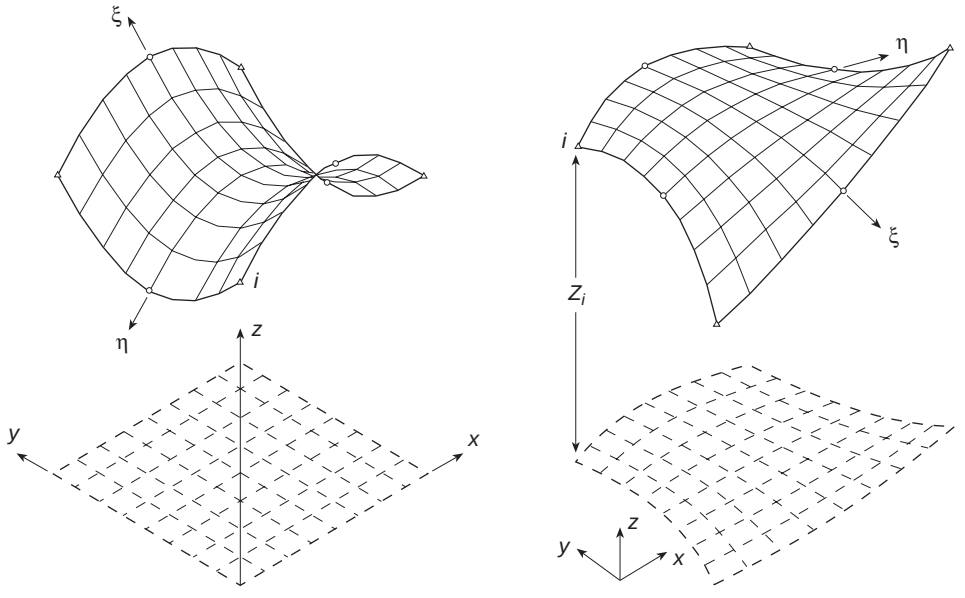


Fig. 9.5 Flat elements (of parabolic type) mapped into three-dimensions.

In Sec. 9.5 we shall define a quantity known as the jacobian determinant. The well-known condition for a *one-to-one* mapping (such as exists in Fig. 9.3 and does not in Fig. 9.4) is that the sign of this quantity should remain unchanged at all the points of the mapped element.

It can be shown that with a parametric transformation based on bilinear shape functions, the necessary condition is that no internal angle [such as α in Fig. 9.6(*a*)]

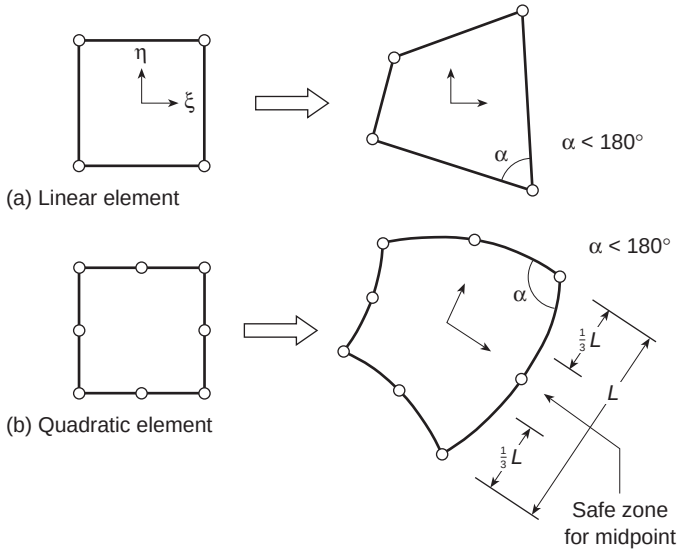


Fig. 9.6 Rules for uniqueness of mapping (*a*) and (*b*).

be greater than 180°.[6] In transformations based on parabolic-type 'serendipity' functions, it is necessary in addition to this requirement to ensure that the mid-side nodes are in the 'middle half' of the distance between adjacent corners but a 'middle third' shown in Fig. 9.6 is safer. For cubic functions such general rules are impractical and numerical checks on the sign of the jacobian determinant are necessary. In practice a parabolic distortion is usually sufficient.

9.3 Geometrical conformability of elements

While it was shown that by the use of the shape function transformation each parent element maps uniquely a part of the real object, it is important that the subdivision of this into the new, curved, elements should leave no gaps. The possibility of such gaps is indicated in Fig. 9.7.

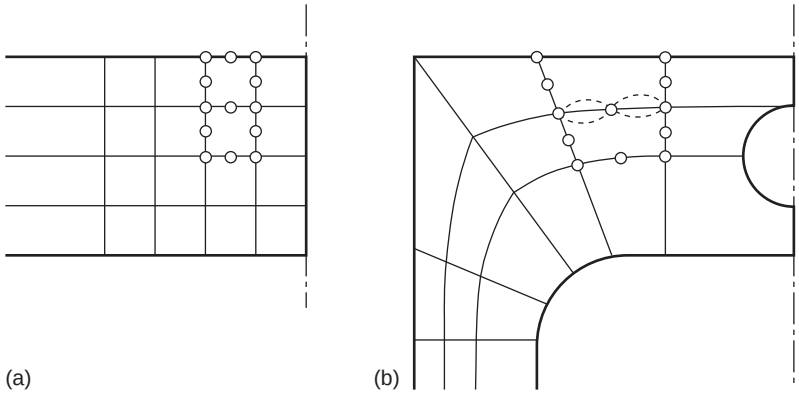


Fig. 9.7 Compatibility requirements in a real subdivision of space.

Theorem 1. *If two adjacent elements are generated from 'parents' in which the shape functions satisfy C_0 continuity requirements then the distorted elements will be contiguous* (*compatible*).

This theorem is obvious, as in such cases uniqueness of any function u required by continuity is simply replaced by that of uniqueness of the x, y, or z coordinate. As adjacent elements are given the same sets of coordinates at nodes, continuity is implied.

9.4 Variation of the unknown function within distorted, curvilinear elements. Continuity requirements

With the shape of the element now defined by the shape functions $\mathbf{N}'$ the variation of the unknown, u, has to be specified before we can establish element properties. This is

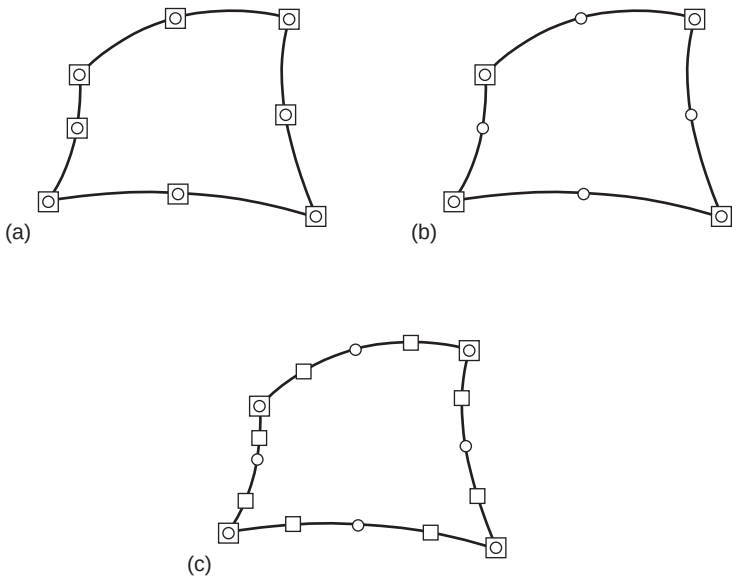


Fig. 9.8 Various element specifications: ○ point at which coordinate is specified; □ points at which the function parameter is specified. (*a*) Isoparametric, (*b*) superparametric, (*c*) subparametric.

most conveniently given in terms of local, curvilinear coordinates by the usual expression

$$u = \mathbf{N}\mathbf{a}^e \tag{9.3}$$

where $\mathbf{a}^e$ lists the nodal values.

Theorem 2. *If the shape functions* $\mathbf{N}$ *used in* (*9.3*) *are such that* C_0 *continuity of u is preserved in the parent coordinates then* C_0 *continuity requirements will be satisfied in distorted elements.*

The proof of this theorem follows the same lines as that in the previous section.

The nodal values may or may not be associated with the same nodes as used to specify the element geometry. For example, in Fig. 9.8 the points marked with a circle are used to define the element geometry. We could use the values of the function defined at nodes marked with a square to define the variation of the unknown.

In Fig. 9.8(*a*) the same points define the geometry and the finite element analysis points. If then

$$\mathbf{N} = \mathbf{N}' \tag{9.4}$$

i.e., the shape functions defining the geometry and the function are the same, the elements will be called *isoparametric*.

We could, however, use only the four corner points to define the variation of u [Fig. 9.8(*b*)]. We shall refer to such an element as *superparametric*, noting that the variation of geometry is more general than that of the actual unknown.

Similarly, if for instance we introduce more nodes to define u than are used to define the geometry, *subparametric* elements will result [Fig. 9.8(*c*)].

While for mapping it is convenient to use 'standard' forms of shape functions the interpolation of the unknown can, of course, use hierarchic forms defined in the previous chapter. Once again the definitions of sub- and superparametric variations are applicable.

Transformations

9.5 Evaluation of element matrices (transformation in ξ, η, ζ coordinates)

To perform finite element analysis the matrices defining element properties, e.g., stiffness, etc., have to be found. These will be of the form

$$\int_V \mathbf{G}\,\mathrm{d}V \tag{9.5}$$

in which the matrix $\mathbf{G}$ depends on $\mathbf{N}$ or its derivatives with respect to *global coordinates*. As an example of this we have the stiffness matrix

$$\int_V \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d}V \tag{9.6}$$

and associated body force vectors

$$\int_V \mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}V \tag{9.7}$$

For each particular class of elastic problems the matrices of $\mathbf{B}$ are given explicitly by their components [see the general form of Eqs (4.10), (5.6), and (6.11)]. Quoting the first of these, Eq. (4.10), valid for plane problems we have

$$\mathbf{B}_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x}, & 0 \\ 0, & \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial y}, & \dfrac{\partial N_i}{\partial x} \end{bmatrix} \tag{9.8}$$

In elasticity problems the matrix $\mathbf{G}$ is thus a function of the first derivatives of $\mathbf{N}$ and this situation will arise in many other classes of problem. In all, C_0 continuity is needed and, as we have already noted, this is readily satisfied by the functions of Chapter 8, written now in terms of curvilinear coordinates.

To evaluate such matrices we note that two transformations are necessary. In the first place, as N_i is defined in terms of local (curvilinear) coordinates, it is necessary to devise some means of expressing the global derivatives of the type occurring in Eq. (9.8) in terms of local derivatives.

In the second place the element of volume (or surface) over which the integration has to be carried out needs to be expressed in terms of the local coordinates with an appropriate change of limits of integration.

Consider, for instance, the set of local coordinates ξ, η, ζ and a corresponding set of global coordinates x, y, z. By the usual rules of partial differentiation we can write, for instance, the ξ derivative as

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi} \tag{9.9}$$

Performing the same differentiation with respect to the other two coordinates and writing in matrix form we have

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi}, & \frac{\partial y}{\partial \xi}, & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta}, & \frac{\partial y}{\partial \eta}, & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta}, & \frac{\partial y}{\partial \zeta}, & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} \tag{9.10}$$

In the above, the left-hand side can be evaluated as the functions N_i are specified in local coordinates. Further, as x, y, z are explicitly given by the relation defining the curvilinear coordinates [Eq. (9.2)], the matrix $\mathbf{J}$ can be found explicitly in terms of the local coordinates. This matrix is known as the *jacobian matrix*.

To find now the global derivatives we invert $\mathbf{J}$ and write

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \tag{9.11}$$

In terms of the shape function defining the coordinate transformation $\mathbf{N}'$ (which as we have seen are only identical with the shape functions $\mathbf{N}$ when the isoparametric formulation is used) we have

$$\mathbf{J} = \begin{bmatrix} \sum \frac{\partial N_i'}{\partial \xi} x_i, & \sum \frac{\partial N_i'}{\partial \xi} y_i, & \sum \frac{\partial N_i'}{\partial \xi} z_i \\ \sum \frac{\partial N_i'}{\partial \eta} x_i, & \sum \frac{\partial N_i'}{\partial \eta} y_i, & \sum \frac{\partial N_i'}{\partial \eta} z_i \\ \sum \frac{\partial N_i'}{\partial \zeta} x_i, & \sum \frac{\partial N_i'}{\partial \zeta} y_i, & \sum \frac{\partial N_i'}{\partial \zeta} z_i \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1'}{\partial \xi}, & \frac{\partial N_2'}{\partial \xi} \cdots \\ \frac{\partial N_1'}{\partial \eta}, & \frac{\partial N_2'}{\partial \eta} \cdots \\ \frac{\partial N_1'}{\partial \zeta}, & \frac{\partial N_2'}{\partial \zeta} \cdots \end{bmatrix} \begin{bmatrix} x_1, & y_1, & z_1 \\ x_2, & y_2, & z_2 \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{9.12}$$

9.5.1 Volume integrals

To transform the variables and the region with respect to which the integration is made, a standard process will be used which involves the determinant of $\mathbf{J}$. Thus, for instance, a volume element becomes

$$\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \det \mathbf{J}\,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta \tag{9.13}$$

This type of transformation is valid irrespective of the number of coordinates used. For its justification the reader is referred to standard mathematical texts.† (See also Appendix F.)

Assuming that the inverse of $\mathbf{J}$ can be found we now have reduced the evaluation of the element properties to that of finding integrals of the form of Eq. (9.5).

More explicitly we can write this as

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \bar{\mathbf{G}}(\xi,\eta,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta \tag{9.14}$$

if the curvilinear coordinates are of the normalized type based on the right prism. Indeed the integration *is carried out within such a prism* and not in the complicated distorted shape, thus accounting for the simple integration limits. One- and two-dimensional problems will similarly result in integrals with respect to one or two coordinates within simple limits.

While the limits of integration are simple in the above case, unfortunately the explicit form of $\bar{\mathbf{G}}$ is not. Apart from the simplest elements, algebraic integration usually defies our mathematical skill, and numerical integration has to be used. This, as will be seen from later sections, is not a severe penalty and has the advantage that algebraic errors are more easily avoided and that general programs, not tied to a particular element, can be written for various classes of problems. Indeed in such numerical calculations the analytical inverses of $\mathbf{J}$ are never explicitly found.

9.5.2 Surface integrals

In elasticity and other applications, surface integrals frequently occur. Typical here are the expressions for evaluating the contributions of surface tractions [see Chapter 2, Eq. (2.24b)]:

$$\mathbf{f} = -\int_A \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}A$$

The element $\mathrm{d}A$ will generally lie on a surface where one of the coordinates (say ζ) is constant.

The most convenient process of dealing with the above is to consider $\mathrm{d}A$ as a vector oriented in the direction normal to the surface (see Appendix F). For three-dimensional problems we form the vector product

$$\mathbf{n}\,\mathrm{d}A = \mathrm{d}\mathbf{A} = \begin{Bmatrix} \dfrac{\partial x}{\partial \xi} \\ \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial z}{\partial \xi} \end{Bmatrix} \times \begin{Bmatrix} \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \eta} \\ \dfrac{\partial z}{\partial \eta} \end{Bmatrix} \mathrm{d}\xi\,\mathrm{d}\eta$$

and on substitution integrate within the domain $-1 \leqslant \xi, \eta \leqslant 1$.

† The determinant of the jacobian matrix is known in the literature simply as 'the jacobian' and is often written as

$$\det \mathbf{J} \equiv \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}$$

For two dimensions a line length $\mathrm{d}S$ arises and here the magnitude is simply

$$\mathbf{n}\,\mathrm{d}S = \mathrm{d}\mathbf{S} = \begin{Bmatrix} \dfrac{\partial x}{\partial \xi} \\ \dfrac{\partial y}{\partial \xi} \\ 0 \end{Bmatrix} \times \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \mathrm{d}\xi = \begin{Bmatrix} \dfrac{\partial y}{\partial \xi} \\ -\dfrac{\partial x}{\partial \xi} \\ 0 \end{Bmatrix} \mathrm{d}\xi$$

on constant η surfaces. This may now be reduced to two components for the two-dimensional problem.

9.6 Element matrices. Area and volume coordinates

The general relationship (9.2) for coordinate mapping and indeed all the following theorems are equally valid for any set of local coordinates and could relate the local $L_1, L_2, \ldots$ coordinates used for triangles and tetrahedra in the previous chapter, to the global cartesian ones.

Indeed most of the discussion of the previous chapter is valid if we simply rename the local coordinates suitably. However, two important differences arise.

The first concerns the fact that the local coordinates are not independent and in fact number one more than the cartesian system. The matrix $\mathbf{J}$ would apparently therefore become rectangular and would not possess an inverse. The second is simply the difference of integration limits which have to correspond with a triangular or tetrahedral 'parent'.

The simplest, though perhaps not the most elegant, way out of the first difficulty is to consider the last variable as a dependent one. Thus, for example, we can introduce formally, in the case of the tetrahedra,

$$\begin{aligned} \xi &= L_1 \\ \eta &= L_2 \\ \zeta &= L_3 \\ 1 - \xi - \eta - \zeta &= L_4 \end{aligned} \tag{9.15}$$

(by definition in the previous chapter) and thus preserve without change Eq. (9.9) and all the equations up to Eq. (9.14).

As the functions N_i are given in fact in terms of L_1, L_2, etc., we must observe that

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial L_1}\frac{\partial L_1}{\partial \xi} + \frac{\partial N_i}{\partial L_2}\frac{\partial L_2}{\partial \xi} + \frac{\partial N_i}{\partial L_3}\frac{\partial L_3}{\partial \xi} + \frac{\partial N_i}{\partial L_4}\frac{\partial L_4}{\partial \xi} \tag{9.16}$$

On using Eq. (9.15) this becomes simply

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial L_1} - \frac{\partial N_i}{\partial L_4}$$

with the other derivatives obtainable by similar expressions.

The integration limits of Eq. (9.14) now change, however, to correspond with the tetrahedron limits, typically

$$\int_0^1 \int_0^{1-\zeta} \int_0^{1-\eta-\zeta} \bar{\mathbf{G}}(\xi,\eta,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta \tag{9.17}$$

The same procedure will clearly apply in the case of triangular coordinates.

It must be noted that once again the expression $\bar{\mathbf{G}}$ will necessitate numerical integration which, however, is carried out over the simple, undistorted, parent region whether this be triangular or tetrahedral.

An alternative to the above is to express the coordinates and constraint as

$$\begin{aligned} r_x &= x - x_1 N_1' - x_2 N_2' - x_3 N_3' - \cdots = 0 \\ r_y &= y - y_1 N_1' - y_2 N_2' - y_3 N_3' - \cdots = 0 \\ r_z &= z - z_1 N_1' - z_2 N_2' - z_3 N_3' - \cdots = 0 \\ r_1 &= 1 - L_1 - L_2 - L_3 - L_4 = 0 \end{aligned} \tag{9.18}$$

where $N_i' = N'(L_1, L_2, L_3, L_4)$, etc. Now derivatives of the above with respect to x and y may be written directly as

$$\begin{bmatrix} \frac{\partial r_x}{\partial x} & \frac{\partial r_x}{\partial y} & \frac{\partial r_x}{\partial z} \\ \frac{\partial r_y}{\partial x} & \frac{\partial r_y}{\partial y} & \frac{\partial r_y}{\partial z} \\ \frac{\partial r_z}{\partial x} & \frac{\partial r_z}{\partial y} & \frac{\partial r_z}{\partial z} \\ \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} \sum x_k \frac{\partial N_k}{\partial L_1} & \sum x_k \frac{\partial N_k}{\partial L_2} & \sum x_k \frac{\partial N_k}{\partial L_3} & \sum x_k \frac{\partial N_k}{\partial L_4} \\ \sum y_k \frac{\partial N_k}{\partial L_1} & \sum y_k \frac{\partial N_k}{\partial L_2} & \sum y_k \frac{\partial N_k}{\partial L_3} & \sum y_k \frac{\partial N_k}{\partial L_4} \\ \sum z_k \frac{\partial N_k}{\partial L_1} & \sum z_k \frac{\partial N_k}{\partial L_2} & \sum z_k \frac{\partial N_k}{\partial L_3} & \sum z_k \frac{\partial N_k}{\partial L_4} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} \frac{\partial L_1}{\partial x} & \frac{\partial L_1}{\partial y} & \frac{\partial L_1}{\partial z} \\ \frac{\partial L_2}{\partial x} & \frac{\partial L_2}{\partial y} & \frac{\partial L_2}{\partial z} \\ \frac{\partial L_3}{\partial x} & \frac{\partial L_3}{\partial y} & \frac{\partial L_3}{\partial z} \\ \frac{\partial L_4}{\partial x} & \frac{\partial L_4}{\partial y} & \frac{\partial L_4}{\partial z} \end{bmatrix} = \mathbf{0} \tag{9.19}$$

The above may be solved for the partial derivatives of L_i with respect to the x, y, z coordinates and used directly with the chain rule written as

$$\frac{\partial N_i}{\partial x} = \frac{\partial N_i}{\partial L_1}\frac{\partial L_1}{\partial x} + \frac{\partial N_i}{\partial L_2}\frac{\partial L_2}{\partial x} + \frac{\partial N_i}{\partial L_3}\frac{\partial L_3}{\partial x} + \frac{\partial N_i}{\partial L_4}\frac{\partial L_4}{\partial x} \tag{19.20}$$

The above has advantages when the coordinates are written using mapping functions as the computation can still be more easily carried out. Also, the calculation of integrals will normally be performed numerically (as described in Sec. 9.10) where the points for integration are defined directly in terms of the volume coordinates.

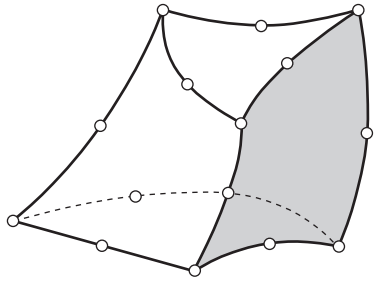


Fig. 9.9 A distorted triangular prism.

Finally it should be remarked that any of the elements given in the previous chapter are capable of being mapped. In some, such as the triangular prism, both area and rectangular coordinates are used (Fig. 9.9). The remarks regarding the dependence of coordinates apply once again with regard to the former but the processes of the present section should make procedures clear.

9.7 Convergence of elements in curvilinear coordinates

To consider the convergence aspects of the problem posed in curvilinear coordinates it is convenient to return to the starting point of the approximation where an energy functional Π, or an equivalent integral form (Galerkin problem statement), was defined by volume integrals essentially similar to those of Eq. (9.5), in which the integrand was a function of u and its first derivatives.

Thus, for instance, the variational principles of the energy kind discussed in Chapter 2 (or others of Chapter 3) could be stated for a scalar function u as

$$\Pi = \int_{\Omega} F\left(u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, x, y\right) \mathrm{d}\Omega + \int_{\Gamma} E(u, \ldots)\, \mathrm{d}\Gamma \tag{9.21}$$

The coordinate transformation changes the derivatives of any function by the jacobian relation (9.11). Thus

$$\begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} = \mathbf{J}^{-1}(\xi, \eta) \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} \tag{9.22}$$

and the functional can be stated simply by a relationship of the form (9.21) with x, y, etc., replaced by ξ, η, etc., with the maximum order of differentiation unchanged.

It follows immediately that if the shape functions are chosen in curvilinear coordinate space so as to observe the usual rules of convergence (continuity and presence of complete first-order polynomials in these coordinates), then convergence will occur. Further, all the arguments concerning the order of convergence with the element size h still hold, providing the solution *is related to the curvilinear coordinate system*.

Indeed, all that has been said above is applicable to problems involving higher derivatives and to most unique coordinate transformations. It should be noted that

the patch test as conceived in the $x, y, \ldots$ coordinate system (see Chapters 2 and 10) is no longer simply applicable and in principle should be applied with polynomial fields imposed in the curvilinear coordinates. In the case of isoparametric (or subparametric) elements the situation is more advantageous. Here a linear (constant derivative x, y) field is always reproduced by the curvilinear coordinate expansion, and thus the lowest order patch test will be passed in the standard manner on such elements.

The proof of this is simple. Consider a standard isoparametric expansion

$$u = \sum_{i=1}^{n} N_i a_i \equiv \mathbf{Na} \qquad \mathbf{N} = \mathbf{N}(\xi, \eta, \zeta) \tag{9.23}$$

with coordinates of nodes defining the transformation as

$$x = \sum N_i x_i \qquad y = \sum N_i y_i \qquad z = \sum N_i z_i \tag{9.24}$$

The question is under what circumstances is it possible for expression (9.23) to define a linear expansion in cartesian coordinates:

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \\ &\equiv \alpha_1 + \alpha_2 \sum N_i x_i + \alpha_3 \sum N_i y_i + \alpha_4 \sum N_i z_i \end{aligned} \tag{9.25}$$

If we take

$$a_i = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 z_i$$

and compare expression (9.23) with (9.25) we note that identity is obtained between these providing

$$\sum N_i = 1$$

As this is the usual requirement of standard element shape functions [see Eq. (8.4)] we can conclude that the following theorem is valid.

Theorem 3. *The constant derivative condition will be satisfied for all isoparametric elements.*

As subparametric elements can always be expressed as specific cases of an isoparametric transformation this theorem is obviously valid here also.

It is of interest to pursue the argument and to see under what circumstances higher polynomial expansions in cartesian coordinates can be achieved under various transformations. The simple linear case in which we 'guessed' the solution has now to be replaced by considering in detail the polynomial terms occurring in expressions such as (9.23) and (9.25) and establishing conditions for equating appropriate coefficients.

Consider a specific problem: the circumstances under which the bilinearly mapped quadrilateral of Fig. 9.10 can fully represent any quadratic cartesian expansion. We now have

$$x = \sum_{1}^{4} N_i' x_i \qquad y = \sum_{1}^{4} N_i' y_i \tag{9.26}$$

and we wish to be able to reproduce

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 \tag{9.27}$$

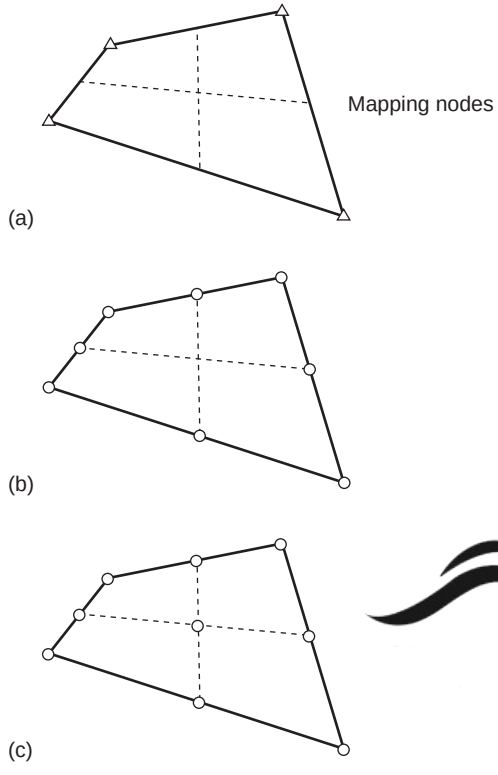


Fig. 9.10 Bilinear mapping of subparametric quadratic eight- and nine-noded element.

Noting that the bilinear form of N_i' contains terms such as $1, \xi, \eta$ and $\xi\eta$, the above can be written as

$$u = \beta_1 + \beta_2\xi + \beta_3\eta + \beta_4\xi^2 + \beta_5\xi\eta + \beta_6\eta^2 + \beta_7\xi\eta^2 + \beta_8\xi^2\eta + \beta_9\xi^2\eta^2 \tag{9.28}$$

where β_1 to β_9 depend on the values of α_1 to α_6.

We shall now try to match the terms arising from the quadratic expansions of the serendipity and lagrangian kinds shown in Fig. 9.10(b) and (c):

$$u = \sum_{1}^{8} N_i a_i \tag{9.29a}$$

$$u = \sum_{1}^{9} N_i a_i \tag{9.29b}$$

where the appropriate terms are of the kind defined in the previous chapter.

For the eight-noded element (serendipity) [Fig. 9.10(b)] we can write (9.29(a)) directly using polynomial coefficients b_i, $i = 1, \ldots, 8$, in place of the nodal variables a_i (noting the terms occurring in the Pascal triangle) as

$$u = b_1 + b_2\xi + b_3\eta + b_4\xi^2 + b_5\xi\eta + b_6\eta^2 + b_7\xi\eta^2 + b_8\xi^2\eta \tag{9.30}$$

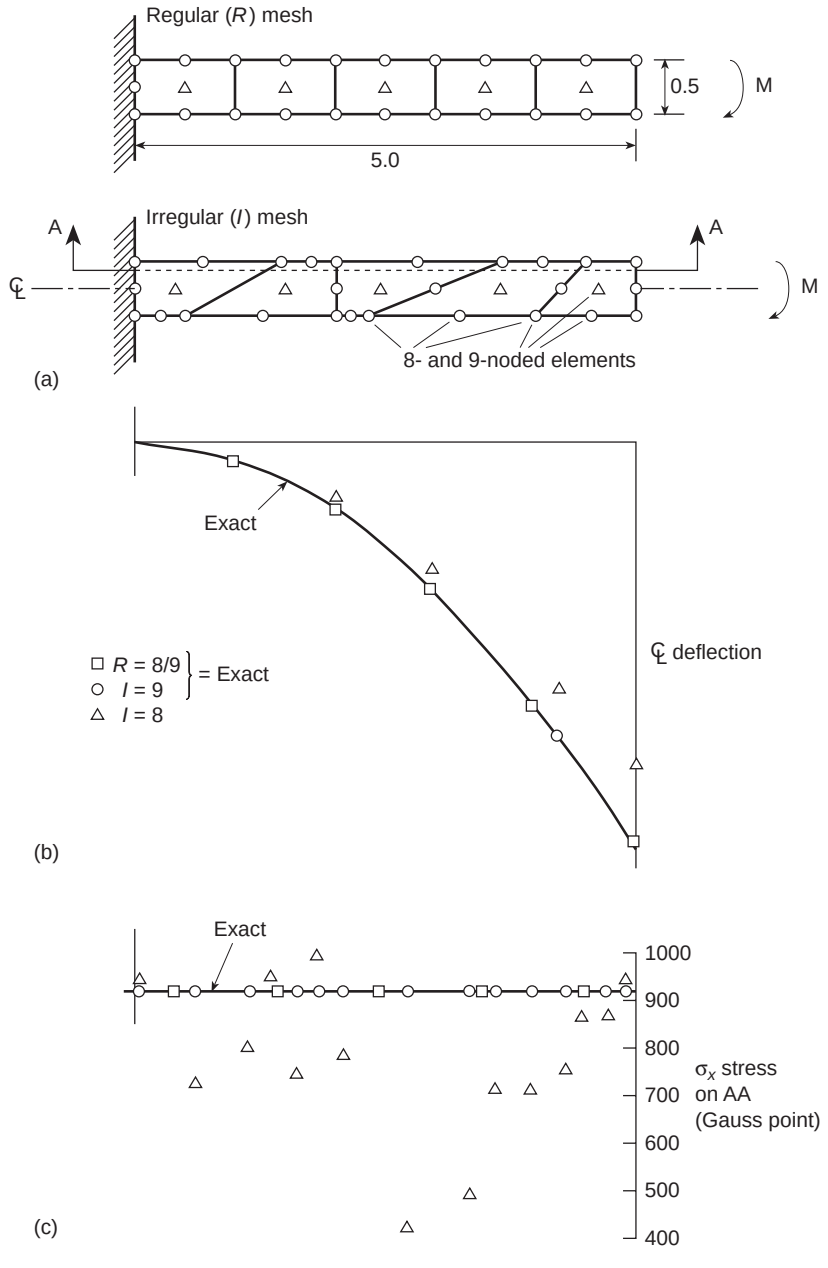


Fig. 9.11 Quadratic serendipity and Lagrange eight- and nine-noded elements in regular and distorted form. Elastic deflection of a beam under constant moment. Note the poor results of the eight-noded element.

It is immediately evident that for arbitrary values of β_1 to β_9 it is impossible to match the coefficients b_1 to b_8 due to the absence of the term $\xi^2\eta^2$ in Eq. (9.30). [However if higher order (quartic, etc.) expansions of the serendipity kind were used such matching would evidently be possible and we could conclude that for

linearly distorted elements the serendipity family of order four or greater will always represent quadratics.]

For the nine-noded, lagrangian, element [Fig. 9.10(*c*)] the expansion similar to (9.30) gives

$$u = b_1 + b_2\xi + b_3\eta + b_4\xi^2 + \cdots + b_8\xi^2\eta + b_9\xi^2\eta^2 \tag{9.31}$$

and the matching of the coefficients of Eqs (9.31) and (9.28) can be made directly.

We can conclude therefore that nine-noded elements represent better cartesian polynomials (when distorted linearly) and therefore are generally preferable in modelling smooth solutions. This matter was first presented by Wachspress but the simple proof presented above is due to Crochet.[8] An example of this is given in Fig. 9.11 where we consider the results of a finite element calculation with eight- and nine-noded elements respectively used to reproduce a simple beam solution in which we know that the exact answers are quadratic. With no distortion both elements give exact results but when distorted only the nine-noded element does so, with the eight-noded element giving quite wild stress fluctuation.

Similar arguments will lead to the conclusion that in three dimensions again only the lagrangian 27-noded element is capable of reproducing fully the quadratic in cartesian coordinates when trilinearly distorted.

Lee and Bathe[9] investigate the problem for cubic and quartic serendipity and lagrangian quadrilateral elements and show that under bilinear distortions the full order cartesian polynomial terms remain in Lagrange elements but not in serendipity ones. They also consider edge distortion and show that this polynomial order is always lost. Additional discussion of such problems is also given by Wachspress.[7]

9.8 Numerical integration – one-dimensional

In Chapter 5, dealing with a relatively simple problem of axisymmetric stress distribution and simple triangular elements, it was noted that exact integration of expressions for element matrices could be troublesome. Now for the more complex distorted elements numerical integration is essential.

Some principles of numerical integration will be summarized here together with tables of convenient numerical coefficients.

To find numerically the integral of a function of one variable we can proceed in one of several ways.[10]

9.8.1 Newton–Cotes quadrature†

In the most obvious procedure, points at which the function is to be found are determined *a priori* – usually at equal intervals – and a polynomial passed through the values of the function at these points and exactly integrated [Fig. 9.12(*a*)].

† 'Quadrature' is an alternative term to 'numerical integration'.

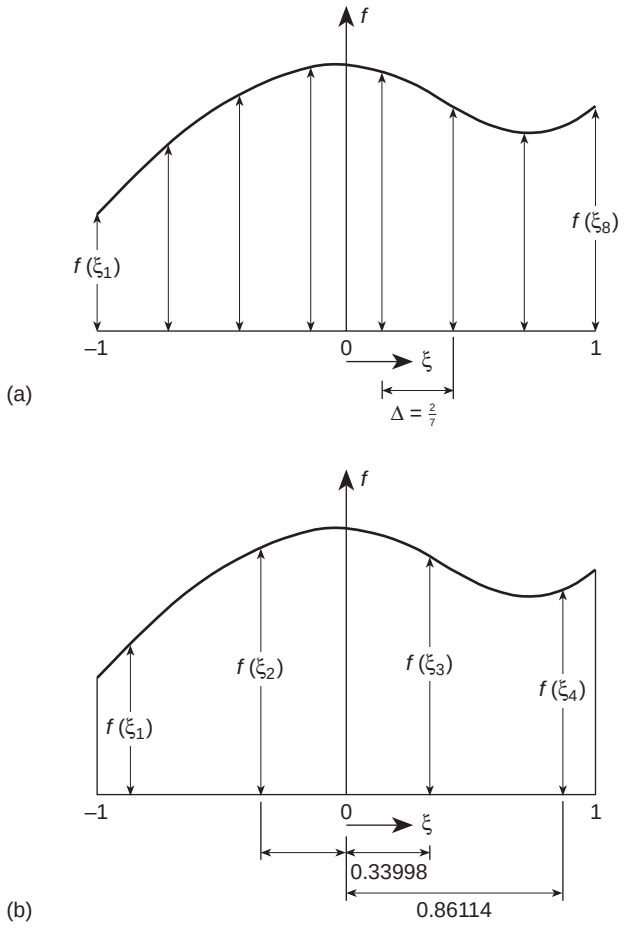


Fig. 9.12 (a) Newton–Cotes and (b) Gauss integrations. Each integrates exactly a seventh-order polynomial [i.e., error $O(h^8)$].

As n values of the function define a polynomial of degree $n-1$, the errors will be of the order $O(h^n)$ where h is the element size. This leads to the well-known Newton–Cotes 'quadrature' formulae. The intregrals can be written as

$$I = \int_{-1}^{1} f(\xi)\,\mathrm{d}\xi = \sum_{1}^{n} H_i f(\xi_i) \tag{9.32}$$

for the range of integration between -1 and $+1$ [Fig. 9.12(a)]. For example, if $n = 2$, we have the well-known trapezoidal rule:

$$I = f(-1) + f(1) \tag{9.33}$$

for $n = 3$, the Simpson 'one-third' rule:

$$I = \tfrac{1}{3}[f(-1) + 4f(0) + f(1)] \tag{9.34}$$

and for $n = 4$:

$$I = \tfrac{1}{4}[f(-1) + 3f(-\tfrac{1}{3}) + 3f(\tfrac{1}{3}) + f(1)] \tag{9.35}$$

Formulae for higher values of n are given in reference 10.

9.8.2 Gauss quadrature

If in place of specifying the position of sampling points *a priori* we allow these to be located at points to be determined so as to aim for best accuracy, then for a given number of sampling points increased accuracy can be obtained. Indeed, if we again consider

$$I = \int_{-1}^{1} f(\xi)\,\mathrm{d}\xi = \sum_{1}^{n} H_i f(\xi_i) \tag{9.36}$$

and again assume a polynomial expression, it is easy to see that for n sampling points we have $2n$ unknowns (H_i and ξ_i) and hence a polynomial of degree $2n - 1$ could be constructed and exactly integrated [Fig. 9.12(*b*)]. The error is thus of order $O(h^{2n})$.

The simultaneous equations involved are difficult to solve, but some mathematical manipulation will show that the solution can be obtained explicitly in terms of Legendre polynomials. Thus this particular process is frequently known as Gauss–Legendre quadrature.[10]

Table 9.1 shows the positions and weighting coefficients for gaussian integration.

For purposes of finite element analysis complex calculations are involved in determining the values of f, the function to be integrated. Thus the Gauss-type processes, requiring the least number of such evaluations, are ideally suited and from now on will be used exclusively.

Other expressions for integration of functions of the type

$$I = \int_{-1}^{1} w(\xi) f(\xi)\,\mathrm{d}\xi = \sum_{1}^{n} H_i f(\xi_i) \tag{9.37}$$

can be derived for prescribed forms of $w(\xi)$, again integrating up to a certain order of accuracy a polynomial expansion of $f(\xi)$.[10]

9.9 Numerical integration – rectangular (2D) or right prism (3D) regions

The most obvious way of obtaining the integral

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\,\mathrm{d}\xi\,\mathrm{d}\eta \tag{9.38}$$

is to first evaluate the inner integral keeping η constant, i.e.,

$$\int_{-1}^{1} f(\xi,\eta)\,\mathrm{d}\xi = \sum_{j=1}^{n} H_j f(\xi_j,\eta) = \psi(\eta) \tag{9.39}$$

Table 9.1 Abscissae and weight coefficients of the gaussian quadrature formula $\int_{-1}^{1} f(x)\,\mathrm{d}x = \sum_{j=1}^{n} H_i f(a_j)$

$\pm a$		H
	$n = 1$	
0		2.000 000 000 000 000
	$n = 2$	
$1/\sqrt{3}$		1.000 000 000 000 000
	$n = 3$	
$\sqrt{0.6}$		5/9
0.000 000 000 000 000		8/9
	$n = 4$	
0.861 136 311 594 953		0.347 854 845 137 454
0.339 981 043 584 856		0.652 145 154 862 546
	$n = 5$	
0.906 179 845 938 664		0.236 926 885 056 189
0.538 469 310 105 683		0.478 628 670 499 366
0.000 000 000 000 000		0.568 888 888 888 889
	$n = 6$	
0.932 469 514 203 152		0.171 324 492 379 170
0.661 209 386 466 265		0.360 761 573 048 139
0.238 619 186 083 197		0.467 913 934 572 691
	$n = 7$	
0.949 107 912 342 759		0.129 484 966 168 870
0.741 531 185 599 394		0.279 705 391 489 277
0.405 845 151 377 397		0.381 830 050 505 119
0.000 000 000 000 000		0.417 959 183 673 469
	$n = 8$	
0.960 289 856 497 536		0.101 228 536 290 376
0.796 666 477 413 627		0.222 381 034 453 374
0.525 532 409 916 329		0.313 706 645 877 887
0.183 434 642 495 650		0.362 683 783 378 362
	$n = 9$	
0.968 160 239 507 626		0.081 274 388 361 574
0.836 031 107 326 636		0.180 648 160 694 857
0.613 371 432 700 590		0.260 610 696 402 935
0.324 253 423 403 809		0.312 347 077 040 003
0.000 000 000 000 000		0.330 239 355 001 260
	$n = 10$	
0.973 906 528 517 172		0.066 671 344 308 688
0.865 063 366 688 985		0.149 451 349 150 581
0.679 409 568 299 024		0.219 086 362 515 982
0.433 395 394 129 247		0.269 266 719 309 996
0.148 874 338 981 631		0.295 524 224 714 753

Evaluating the outer integral in a similar manner, we have

$$\begin{aligned} I = \int_{-1}^{1} \psi(\eta)\,\mathrm{d}\eta &= \sum_{i=1}^{n} H_i \psi(\eta_i) \\ &= \sum_{i=1}^{n} H_i \sum_{j=1}^{n} H_j f(\xi_j, \eta_i) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} H_i H_j f(\xi_j, \eta_i) \end{aligned} \tag{9.40}$$

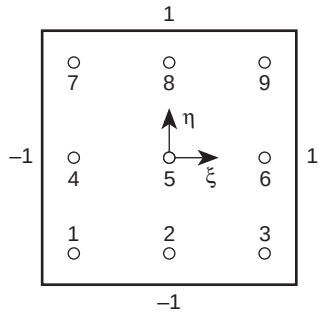


Fig. 9.13 Integrating points for $n = 3$ in a square region. (Exact for polynomial of fifth order in each direction).

For a right prism we have similarly

$$I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta$$
$$= \sum_{m=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} H_i H_j H_m f(\xi_i,\eta_j,\xi_m) \tag{9.41}$$

In the above, the number of integrating points in each direction was assumed to be the same. Clearly this is not necessary and on occasion it may be an advantage to use different numbers in each direction of integration.

It is of interest to note that in fact the double summation can be readily interpreted as a single one over $(n \times n)$ points for a rectangle (or n^3 points for a cube). Thus in Fig. 9.13 we show the nine sampling points that result in exact integrals of order 5 in each direction.

However, we could approach the problem directly and require an exact integration of a fifth-order polynomial in two dimensions. At any sampling point two coordinates and a value of f have to be determined in a weighting formula of type

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\,\mathrm{d}\xi\,\mathrm{d}\eta = \sum_{1}^{m} w_i f(\xi_i,\eta_i) \tag{9.42}$$

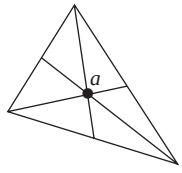
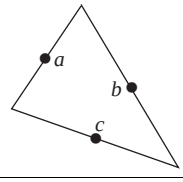
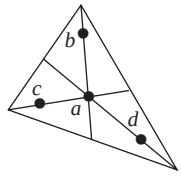
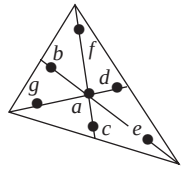
There it would appear that only seven points would suffice to obtain the same order of accuracy. Some formulae for three-dimensional bricks have been derived by Irons[11] and used successfully.[12]

9.10 Numerical integration – triangular or tetrahedral regions

For a triangle, in terms of the area coordinates the integrals are of the form

$$I = \int_{0}^{1}\int_{0}^{1-L_1} f(L_1L_2L_3)\,\mathrm{d}L_2\,\mathrm{d}L_1 \qquad L_3 = 1 - L_1 - L_2 \tag{9.43}$$

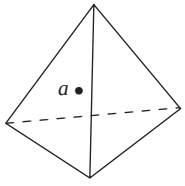
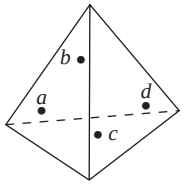
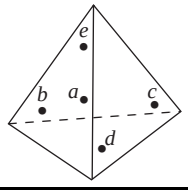
Table 9.2 Numerical integration formulae for triangles

Order	Figure	Error	Points	Triangular coordinates	Weights
Linear		$R = O(h^2)$	a	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$	1
Quadratic		$R = O(h^3)$	a b c	$\frac{1}{2}, \frac{1}{2}, 0$ $0, \frac{1}{2}, \frac{1}{2}$ $\frac{1}{2}, 0, \frac{1}{2}$	$\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$
Cubic		$R = O(h^4)$	a b c d	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$ $0.6, 0.2, 0.2$ $0.2, 0.6, 0.2$ $0.2, 0.2, 0.6$	$-\frac{27}{48}$ $\frac{25}{48}$
Quintic		$R = O(h^6)$	a b c d e f g with $\alpha_1 = 0.059\,715\,871\,7$ $\beta_1 = 0.470\,142\,064\,1$ $\alpha_2 = 0.797\,426\,985\,3$ $\beta_2 = 0.101\,286\,507\,3$	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$ $\alpha_1, \beta_1, \beta_1$ $\beta_1, \alpha_1, \beta_1$ $\beta_1, \beta_1, \alpha_1$ $\alpha_2, \beta_2, \beta_2$ $\beta_2, \alpha_2, \beta_2$ $\beta_2, \beta_2, \alpha_2$	0.225 000 000 0 0.132 394 152 7 0.125 939 180 5

Once again we could use n Gauss points and arrive at a summation expression of the type used in the previous section. However, the limits of integration now involve the variable itself and it is convenient to use alternative sampling points for the second integration by use of a special Gauss expression for integrals of the type given by Eq. (9.37) in which w is a linear function. These have been devised by Radau[13] and used successfully in the finite element context.[14] It is, however, much more desirable (and aesthetically pleasing) to use special formulae in which no bias is given to any of the natural coordinates L_i. Such formulae were first derived by Hammer *et al.*[15] and Felippa[16] and a series of necessary sampling points and weights is given in Table 9.2.[17] (A more comprehensive list of higher formulae derived by Cowper is given on p. 184 of reference 17.)

A similar extension for tetrahedra can obviously be made. Table 9.3 presents some formulae based on reference 15.

Table 9.3 Numerical integration formulae for tetrahedra

No.	Order	Figure	Error	Points	Tetrahedral coordinates	Weights
1	Linear		$R = O(h^2)$	a	$\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$	1
2	Quadratic		$R = O(h^3)$	a b c d	$\alpha, \beta, \beta, \beta$ $\beta, \alpha, \beta, \beta$ $\beta, \beta, \alpha, \beta$ $\beta, \beta, \beta, \alpha$ $\alpha = 0.585\,410\,20$ $\beta = 0.138\,196\,60$	$\frac{1}{4}$
3	Cubic		$R = O(h^4)$	a b c d e	$\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$ $\frac{1}{2}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}$ $\frac{1}{6}, \frac{1}{2}, \frac{1}{6}, \frac{1}{6}$ $\frac{1}{6}, \frac{1}{6}, \frac{1}{2}, \frac{1}{6}$ $\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2}$	$-\frac{4}{5}$ $\frac{9}{20}$

9.11 Required order of numerical integration

With numerical integration used in place of exact integration, an additional error is introduced into the calculation and the first impression is that this should be reduced as much as possible. Clearly the cost of numerical integration can be quite significant, and indeed in some early programs numerical formulation of element characteristics used a comparable amount of computer time as in the subsequent solution of the equations. It is of interest, therefore, to determine (*a*) the minimum integration requirement permitting convergence and (*b*) the integration requirements necessary to preserve the rate of convergence which would result if exact integation were used.

It will be found later (Chapters 10 and 12) that it is in fact often a positive disadvantage to use higher orders of integration than those actually needed under (*b*) as, for very good reasons, a 'cancellation of errors' due to discretization and due to inexact integration can occur.

9.11.1 Minimum order of integration for convergence

In problems where the energy functional (or equivalent Galerkin integral statements) defines the approximation we have already stated that convergence can occur providing any arbitrary constant value of the mth derivatives can be reproduced.

In the present case $m = 1$ and we thus require that in integrals of the form (9.5) a constant value of $\mathbf{G}$ be correctly integrated. *Thus the volume of the element* $\int_V \mathrm{d}V$ *needs to be evaluated correctly for convergence to occur*. In curvilinear coordinates we can thus argue that $\int_V \det|J|\,\mathrm{d}\zeta\,\mathrm{d}\eta\,\mathrm{d}\xi$ has to be evaluated exactly.[3,6]

9.11.2 Order of integration for no loss of convergence

In a general problem we have already found that the finite element approximate evaluation of energy (and indeed all the other integrals in a Galerkin-type approximation, see Chapter 3) was exact to the order $2(p-m)$, where p was the degree of the complete polynomial present and m the order of differentials occurring in the appropriate expressions.

Providing the integration is exact to order $2(p-m)$, or shows an error of $O(h^{2(p-m)+1})$, or less, then no loss of convergence order will occur.† If in curvilinear coordinates we take a curvilinear dimension h of an element, the same rule applies. For C_0 problems (i.e., $m = 1$) the integration formulae should be as follows:

$$\begin{aligned} p = 1, &\quad \text{linear elements} &\quad O(h) \\ p = 2, &\quad \text{quadratic elements} &\quad O(h^3) \\ p = 3, &\quad \text{cubic elements} &\quad O(h^5) \end{aligned}$$

We shall make use of these results in practice, as will be seen later, but it should be noted that for a linear quadrilateral or triangle a single-point integration is adequate. For parabolic quadrilaterals (or bricks) 2×2 (or $2 \times 2 \times 2$), Gauss point integration is adequate and for parabolic triangles (or tetrahedra) three-point (and four-point) formulae of Tables 9.2 and 9.3 are needed.

The basic theorems of this section have been introduced and proved numerically in published work.[18–20]

9.11.3 Matrix singularity due to numerical integration

The final outcome of a finite element approximation in linear problems is an equation system

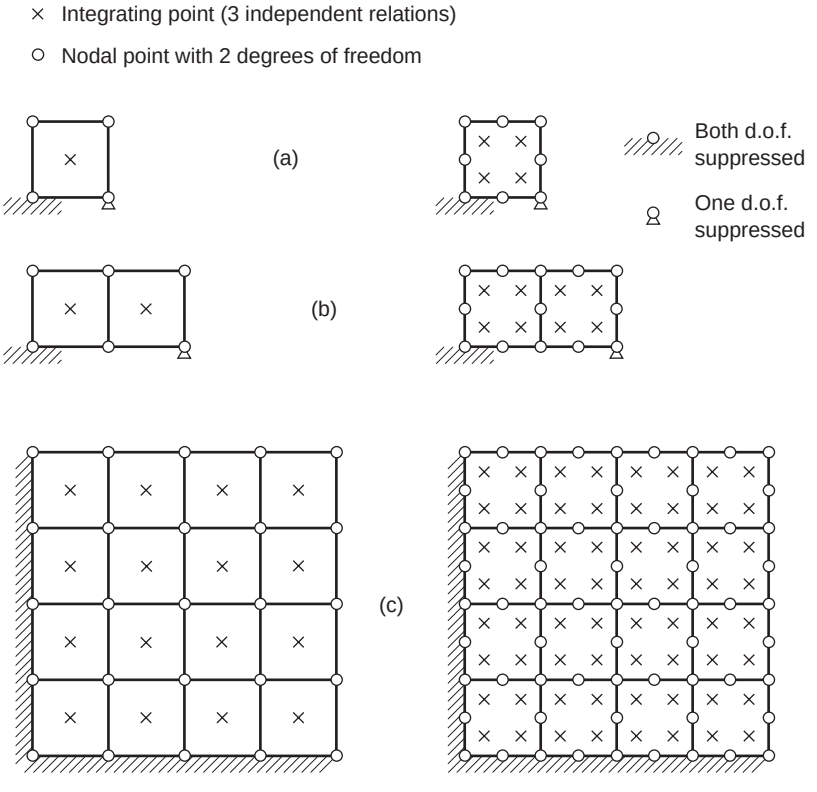
$$\mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{9.44}$$

in which the boundary conditions have been inserted and which should, on solution for the parameter $\mathbf{a}$, give an approximate solution for the physical situation. If a solution is unique, as is the case with well-posed physical problems, the equation matrix $\mathbf{K}$ should be non-singular. We have *a priori* assumed that this was the case with exact integration and in general have not been disappointed. With numerical integration singularities may arise for low integration orders, and this may make such orders impractical. It is easy to show how, in some circumstances, a singularity of $\mathbf{K}$ must

† For an energy principle use of quadrature may result in loss of a bound for $\Pi(\mathbf{a})$.

arise, but it is more difficult to prove that it will not. We shall, therefore, concentrate on the former case.

With numerical integration we replace the integrals by a weighted sum of independent linear relations between the nodal parameters **a**. These linear relations supply the only information from which the matrix **K** is constructed. *If the number of unknowns* **a** *exceeds the number of independent relations supplied at all the integrating points, then the matrix* **K** *must be singular.*



	Linear		Quadratic	
	Degree of freedom	Independent relation	Degree of freedom	Independent relation
(a)	4 x 2 – 3 = 5 > 1 x 3 = 3 singular		2 x 8 – 3 = 13 > 4 x 3 = 12 singular	
(b)	6 x 2 – 3 = 9 > 2 x 3 = 6 singular		13 x 2 – 3 = 23 < 8 x 3 = 24	
(c)	25 x 2 – 18 = 32 < 16 x 3 = 48		48 x 2 = 96< 64 x 3 = 192	

Fig. 9.14 Check on matrix singularity in two-dimensional elasticity problems (a), (b), and (c).

To illustrate this point we shall consider two-dimensional elasticity problems using linear and parabolic serendipity quadrilateral elements with one- and four-point quadratures respectively.

Here at each integrating point *three* independent 'strain relations' are used and the total number of independent relations equals 3 × (number of integration points). The number of unknowns **a** is simply 2 × (number of nodes) less restrained degrees of freedom.

In Fig. 9.14(*a*) and (*b*) we show a single element and an assembly of two elements supported by a minimum number of specified displacements eliminating rigid body motion. The simple calculation shows that only in the assembly of the quadratic elements is elimination of singularities possible, all the other cases remaining strictly singular.

In Fig. 9.14(*c*) a well-supported block of both kinds of elements is considered and here for both element types non-singular matrices may arise although local, near singularity may still lead to unsatisfactory results (see Chapter 10).

The reader may well consider the same assembly but supported again by the minimum restraint of three degrees of freedom. The assembly of linear elements with a single integrating point *will* be singular while the quadratic ones will, in fact, usually be well behaved.

For the reason just indicated, linear single-point integrated elements are used infrequently in static solutions, though they do find wide use in 'explicit' dynamics codes – but needing certain remedial additions (e.g., hourglass control[21,22]) – while four-point quadrature is often used for quadratic serendipity elements.†

In Chapter 10 we shall return to the problem of convergence and will indicate dangers arising from local element singularities.

However, it is of interest to mention that in Chapter 12 we shall in fact *seek* matrix singularities for special purposes (e.g., incompressibility) using similar arguments.

9.12 Generation of finite element meshes by mapping. Blending functions

It would have been observed that it is an easy matter to obtain a coarse subdivision of the analysis domain with a small number of isoparametric elements. If second- or third-degree elements are used, the fit of these to quite complex boundaries is reasonable, as shown in Fig. 9.15(*a*) where four parabolic elements specify a sectorial region. This number of elements would be too small for analysis purposes *but a simple subdivision into finer elements* can be done automatically by, say, assigning new positions of nodes of the central points of the curvilinear coordinates and thus deriving a larger number of similar elements, as shown in Fig. 9.15(*b*). Indeed, automatic subdivision could be carried out further to generate a field of triangular elements. The process thus allows us, with a small amount of original *input data*, to derive a finite element mesh of any refinement desirable. In reference 23 this type of mesh generation is developed for two- and three-dimensional solids and surfaces and is reasonably

† Repeating the test for quadratic lagrangian elements indicates a singularity for 2 × 2 quadrature (see Chapter 10 for dangers).

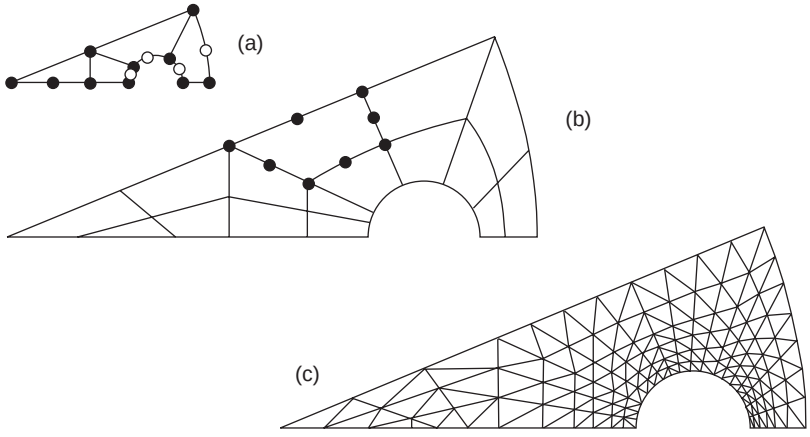


Fig. 9.15 Automatic mesh generation by parabolic isoparametric elements. (a) Specified mesh points. (b) Automatic subdivision into a small number of isoparametric elements. (c) Automatic subdivision into linear triangles.

efficient. However, elements of predetermined size and/or gradation cannot be easily generated.

The main drawback of the mapping and generation suggested is the fact that the originally circular boundaries in Fig. 9.15(*a*) are approximated by simple parabolae and a geometric error can be developed there. To overcome this difficulty another form of mapping, originally developed for the representation of complex motor-car body shapes, can be adopted for this purpose.[24] In this mapping blending functions interpolate the unknown u in such a way as to satisfy *exactly* its variations along the edges of a square ξ, η domain. If the coordinates x and y are used in a parametric expression of the type given in Eq. (9.1), then any complex shape can be mapped by a single element. In reference 24 the region of Fig. 9.15 is in fact so mapped and a mesh subdivision obtained directly without any geometric error on the boundary.

The blending processes are of considerable importance and have been used to construct some interesting element families[25] (which in fact include the standard serendipity elements as a subclass). To explain the process we shall show how a function with prescribed variations along the boundaries can be interpolated.

Consider a region $-1 \leqslant \xi, \eta \leqslant 1$, shown in Fig. 9.16, on the edges of which an arbitrary function ϕ is specified [i.e., $\phi(-1,\eta), \phi(1,\eta), \phi(\xi,-1), \phi(\xi,1)$ are given]. The problem presented is that of interpolating a function $\phi(\xi,\eta)$ so that a smooth surface reproducing precisely the boundary values is obtained. Writing

$$\begin{aligned} N^1(\xi) &= \frac{1-\xi}{2} \qquad N^2(\xi) = \frac{1+\xi}{2} \\ N^1(\eta) &= \frac{1-\eta}{2} \qquad N^2(\eta) = \frac{1+\eta}{2} \end{aligned} \tag{9.45}$$

for our usual one-dimensional linear interpolating functions, we note that

$$P_\eta \phi \equiv N^1(\eta)\phi(\xi,-1) + N^2(\eta)\phi(\xi,1) \tag{9.46}$$

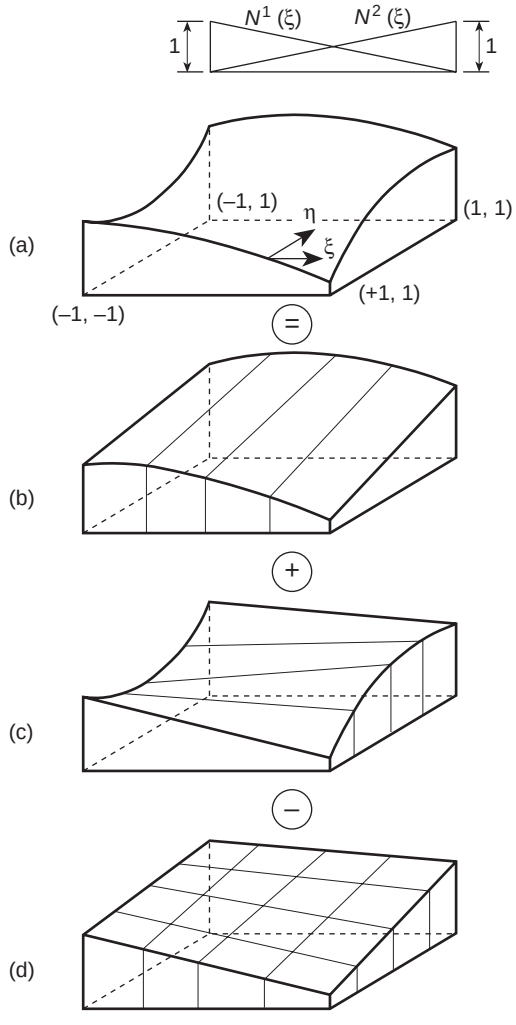


Fig. 9.16 Stages of construction of a blending interpolation (a), (b), (c), and (d).

interpolates linearly between the specified functions in the η direction, as shown in Fig. 9.16(*b*). Similarly,

$$P_\xi \phi \equiv N^1(\xi)\phi(\eta, -1) + N^2(\xi)\phi(\eta, 1) \tag{9.47}$$

interpolates linearly in the ξ direction [Fig. 9.16(*c*)]. Constructing a third function which is a standard linear, bilinear interpolation of the kind we have already encountered [Fig. 9.16(*d*)], i.e.,

$$\begin{aligned} P_\xi P_\eta \phi = {} & N^2(\xi)N^2(\eta)\phi(1,1) + N^2(\xi)N^1(\eta)\phi(1,-1) \\ & + N^1(\xi)N^2(\eta)\phi(-1,1) + N^1(\xi)N^1(\eta)\phi(-1,-1) \end{aligned} \tag{9.48}$$

we note by inspection that

$$\phi = P_\eta \phi + P_\xi \phi - P_\xi P_\eta \phi \tag{9.49}$$

is a smooth surface interpolating exactly the boundary functions.

Extension to functions with higher order blending is almost evident, and immediately the method of mapping the quadrilateral region $-1 \leqslant \xi, \eta \leqslant 1$ to any arbitrary shape is obvious.

Though the above mesh generation method derives from mapping and indeed has been widely applied in two and three dimensions, we shall see in the chapter devoted to adaptivity (Chapter 15) that the optimal solution or specification of *mesh density* or *size* should guide the mesh generation. We shall discuss this problem in that chapter to some extent, but the interested reader is directed to references 26, 27 or books that have appeared on the subject.[28–31] The subject has now grown to such an extent that discussion in any detail is beyond the scope of this book. In the programs mentioned at the end of each volume of this book we shall refer to the GiD system which is available to readers.[32]

9.13 Infinite domains and infinite elements

9.13.1 Introduction

In many problems of engineering and physics infinite or semi-infinite domains exist. A typical example from structural mechanics may, for instance, be that of three-dimensional (or axisymmetric) excavation, illustrated in Fig. 9.17. Here the problem is one of determining the deformations in a semi-infinite half-space due to the removal of loads with the specification of zero displacements at infinity. Similar problems abound in electromagnetics and fluid mechanics but the situation illustrated is typical. The question arises as to how such problems can be dealt with by a method of approximation in which elements of decreasing size are used in the modelling process. The first intuitive answer is the one illustrated in Fig. 9.17(*a*) where the infinite boundary condition is specified at a finite boundary placed at a *large distance* from the object. This, however, begs the question of what is a 'large distance' and obviously substantial errors may arise if this boundary is not placed far enough away. On the other hand, pushing this out excessively far necessitates the introduction of a large number of elements to model regions of relatively little interest to the analyst.

To overcome such 'infinite' difficulties many methods have been proposed. In some a sequence of nesting grids is used and a recurrence relation derived.[33,34] In others a boundary-type exact solution is used and coupled to the finite element domain.[35,36] However, without doubt, the most effective and efficient treatment is the use of 'infinite elements'[37–40] pioneered originally by Bettess.[41] In this process the conventional, finite elements are coupled to elements of the type shown in Fig. 9.17(*b*) which model in a reasonable manner the material stretching to infinity.

The shape of such two-dimensional elements and their treatment is best accomplished by mapping[39–41] these onto a bi-unit square (or a finite line in one dimension or cube in three dimensions). However, it is essential that the sequence of trial

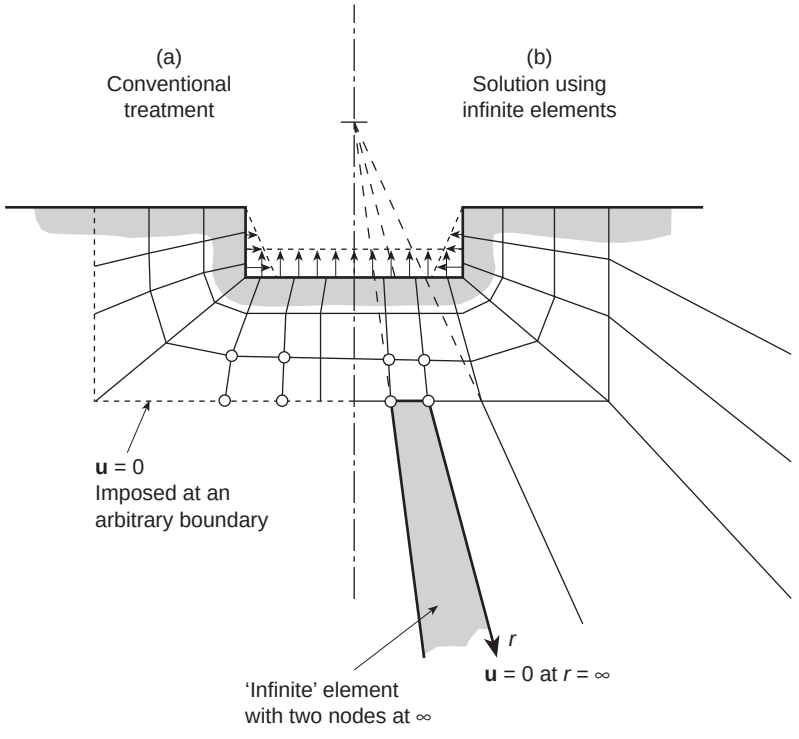


Fig. 9.17 A semi-infinite domain. Deformations of a foundation due to removal of load following an excavation. (a) Conventional treatment and (b) use of infinite elements.

functions introduced in the mapped domain be such that it is complete and capable of modelling the true behaviour as the radial distance r increases. Here it would be advantageous if the mapped shape functions could approximate a sequence of the decaying form

$$\frac{C_1}{r} + \frac{C_2}{r^2} + \frac{C_3}{r^3} + \cdots \tag{9.50}$$

where C_i are arbitrary constants and r is the radial distance from the 'focus' of the problem.

In the next subsection we introduce a mapping function capable of doing just this.

9.13.2 The mapping function

Figure 9.18 illustrates the principles of generation of the derived mapping function.

We shall start with a one-dimensional mapping along a line CPQ coinciding with the x-direction. Consider the following function:

$$x = -\frac{\xi}{1-\xi} x_{\mathrm{C}} + \left(1 + \frac{\xi}{1-\xi}\right) x_{\mathrm{Q}} = \bar{N}_{\mathrm{C}} x_{\mathrm{C}} + \bar{N}_{\mathrm{Q}} x_{\mathrm{Q}} \tag{9.51a}$$

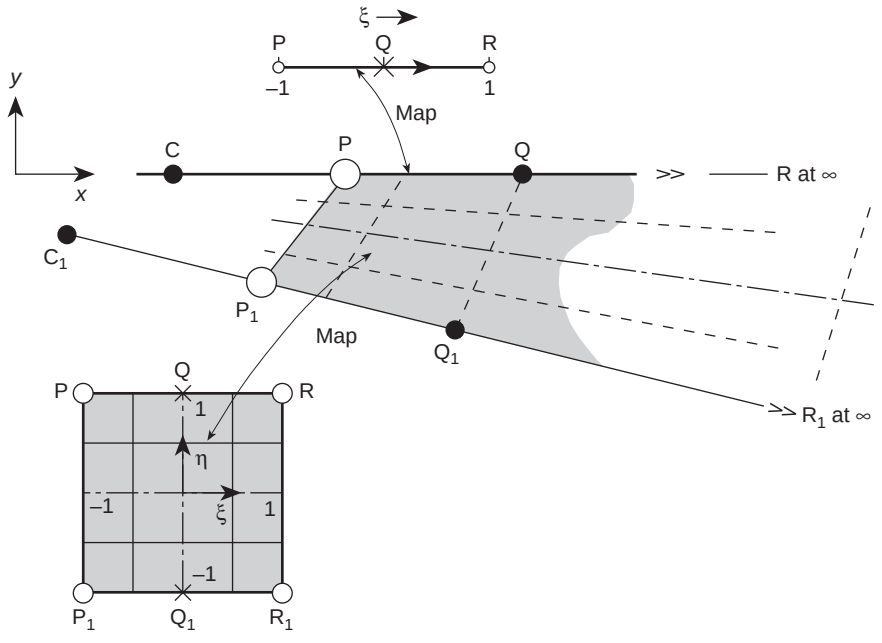


Fig. 9.18 Infinite line and element map. Linear η interpolation.

and we immediately observe that

$$\begin{aligned} \xi = -1 \quad & \text{corresponds to } x = \frac{x_Q + x_C}{2} \equiv x_P \\ \xi = 0 \quad & \text{corresponds to } x = x_Q \\ \xi = 1 \quad & \text{corresponds to } x = \infty \end{aligned}$$

where x_P is a point midway between Q and C.

Alternatively the above mapping could be written directly in terms of the Q and P coordinates by simple elimination of x_C. This gives, using our previous notation:

$$\begin{aligned} x &= N_Q x_Q + N_P x_P \\ &= \left(1 + \frac{2\xi}{1-\xi}\right) x_Q - \frac{2\xi}{1-\xi} x_P \end{aligned} \tag{9.51b}$$

Both forms give a mapping that is independent of the origin of the x-coordinate as

$$N_Q + N_P = 1 = \bar{N}_C + \bar{N}_Q \tag{9.52}$$

The significance of the point C is, however, of great importance. It represents the centre from which the 'disturbance' originates and, as we shall now show, allows the expansion of the form of Eq. (9.50) to be achieved on the assumption that r is measured from C. Thus

$$r = x - x_C \tag{9.53}$$

If, for instance, the unknown function u is approximated by a polynomial function using, say, hierarchical shape functions and giving

$$u = \alpha_0 + \alpha_1\xi + \alpha_2\xi^2 + \alpha_3\xi^3 + \cdots \tag{9.54}$$

we can easily solve Eqs (9.51a) for ξ, obtaining

$$\xi = 1 - \frac{x_Q - x_C}{x - x_C} = 1 - \frac{x_Q - x_C}{r} \tag{9.55}$$

Substitution into Eq. (9.54) shows that a series of the form given by Eq. (9.50) is obtained with the linear shape function in ξ corresponding to $1/r$ terms, quadratic to $1/r^2$, etc.

In one dimenson the objectives specified have thus been achieved and the element will yield convergence as the degree of the polynomial expansion, p, increases. Now a generalization to two or three dimensions is necessary. It is easy to see that this can be achieved by simple products of the one-dimensional infinite mapping with a 'standard' type of shape function in η (and ζ) directions in the manner indicated in Fig. 9.18.

Firstly we generalize the interpolation of Eqs (9.51) for any straight line in x, y, z space and write (for such a line as $C_1P_1Q_1$ in Fig. 9.18)

$$\begin{aligned} x &= -\frac{\xi}{1-\xi}x_{C_1} + \left(1 + \frac{\xi}{1-\xi}\right)x_{Q_1} \\ y &= -\frac{\xi}{1-\xi}y_{C_1} + \left(1 + \frac{\xi}{1-\xi}\right)y_{Q_1} \\ z &= -\frac{\xi}{1-\xi}z_{C_1} + \left(1 + \frac{\xi}{1-\xi}\right)z_{Q_1} \qquad \text{(in three dimensions)} \end{aligned} \tag{9.56}$$

Secondly we complete the interpolation and map the whole $\xi\eta(\zeta)$ domain by adding a 'standard' interpolation in the $\eta(\zeta)$ directions. Thus for the linear interpolation shown we can write for elements $PP_1QQ_1RR_1$ of Fig. 9.18, as

$$\begin{aligned} x = N_1(\eta)\left[-\frac{\xi}{1-\xi}x_C\left(1 + \frac{\xi}{1-\xi}\right)x_Q\right] \\ + N_0(\eta)\left(-\frac{\xi}{1-\xi}x_{C_1} + \frac{\xi}{1-\xi}x_{Q_1}\right), \qquad \text{etc.} \end{aligned} \tag{9.57}$$

with

$$N_1(\eta) = \frac{1+\eta}{2} \qquad N_0(\eta) = \frac{1-\eta}{2}$$

and map the points as shown.

In a similar manner we could use quadratic interpolations and map an element as shown in Fig. 9.19 by using quadratic functions in η.

Thus it is an easy matter to create infinite elements and join these to a standard element mesh as shown in Fig. 9.17(b). The reader will observe that in the generation of such element properties only the transformation jacobian matrix differs from standard forms, hence only this has to be altered in conventional programs.

The 'origin' or 'pole' of the coordinates C can be fixed arbitrarily for each radial line, as shown in Fig. 9.18. This will be done by taking account of the knowledge of the physical solution expected.

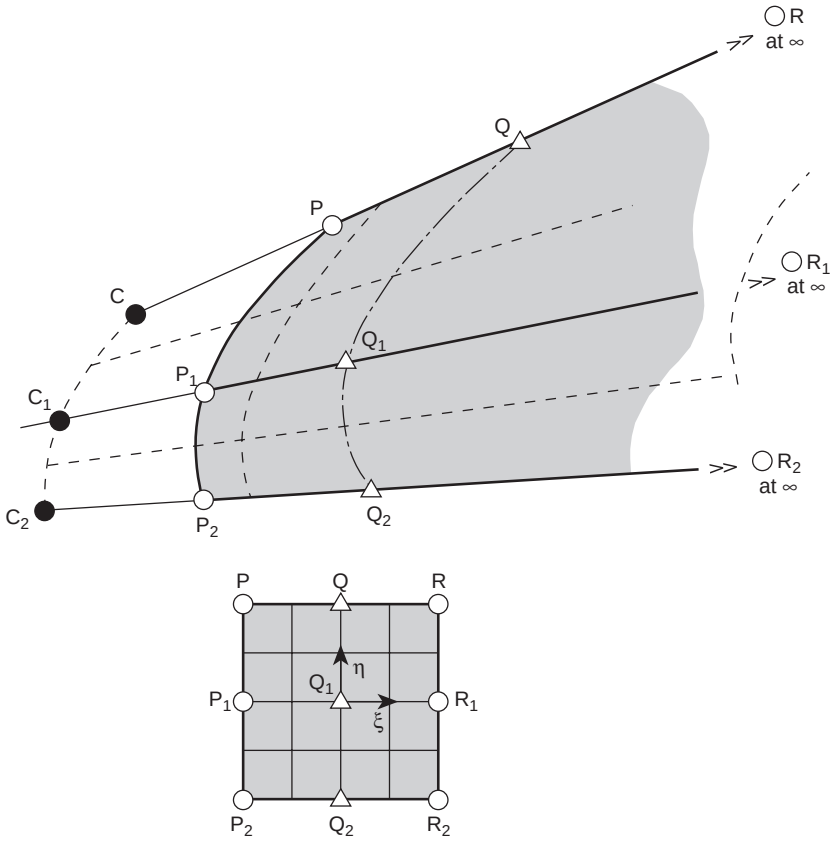


Fig. 9.19 Infinite element map. Quadratic η interpolation.

In Fig. 9.20 we show a solution of the Boussinesq problem (a point load on an elastic half-space). Here results of using a fixed displacement or infinite elements are compared and the big changes in the solution noted. In this example the pole of each element was taken at the load point for obvious reasons.[40]

Figure 9.21 shows how similar infinite elements (of the linear kind) can give excellent results, even when combined with very few standard elements. In this example where a solution of the Laplace equation is used (see Chapter 7) for an irrotational fluid flow, the poles of the infinite elements are chosen at arbitrary points of the aerofoil centre-line.

In concluding this section it should be remarked that the use of infinite elements (as indeed of any other finite elements) must be tempered by background analytical knowledge and 'miracles' should not be expected. Thus the user should not expect, for instance, such excellent results as those shown in Fig. 9.20 for a plane elasticity problem for the displacements. It is 'well known' that in this case the displacements under any load which is not self-equilibrated will be infinite everywhere and the numbers obtained from the computation will not be, whereas for the three-dimensional case it is infinite only at a point load.

Extensive use of infinite elements is made in Volume 3 in the context of the solution of wave problems.

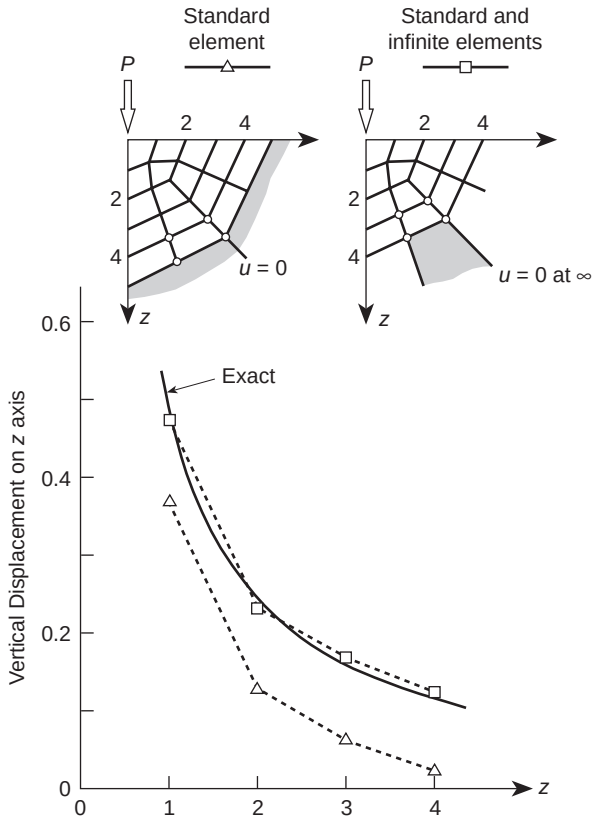


Fig. 9.20 A point load on an elastic half-space (Boussinesq problem). Standard linear elements and infinite line elements ($E = 1$, $\nu = 0.1$, $P = 1$).

9.14 Singular elements by mapping for fracture mechanics, etc.

In the study of fracture mechanics interest is often focused on the singularity point where quantities such as stress become (mathematically, but not physically) infinite. Near such singularities normal, polynomial-based, finite element approximations perform badly and attempts have frequently been made here to include special functions within an element which can model the analytically known singular function. References 42–69 give an extensive literature survey of the problem and finite element solution techniques. An alternative to the introduction of special functions within an element – which frequently poses problems of enforcing continuity requirements with adjacent, standard, elements – lies in the use of special mapping techniques.

An element of this kind, shown in Fig. 9.22(a), was introduced almost simultaneously by Henshell and Shaw[65] and Barsoum[66,67] for quadrilaterals by a simple shift of the mid-side node in quadratic, isoparametric elements to the quarter point.

It can now be shown (and we leave this exercise to the curious reader) that along the element edges the derivatives $\partial u/\partial x$ (or strains) vary as $1/\sqrt{r}$ where r is the distance

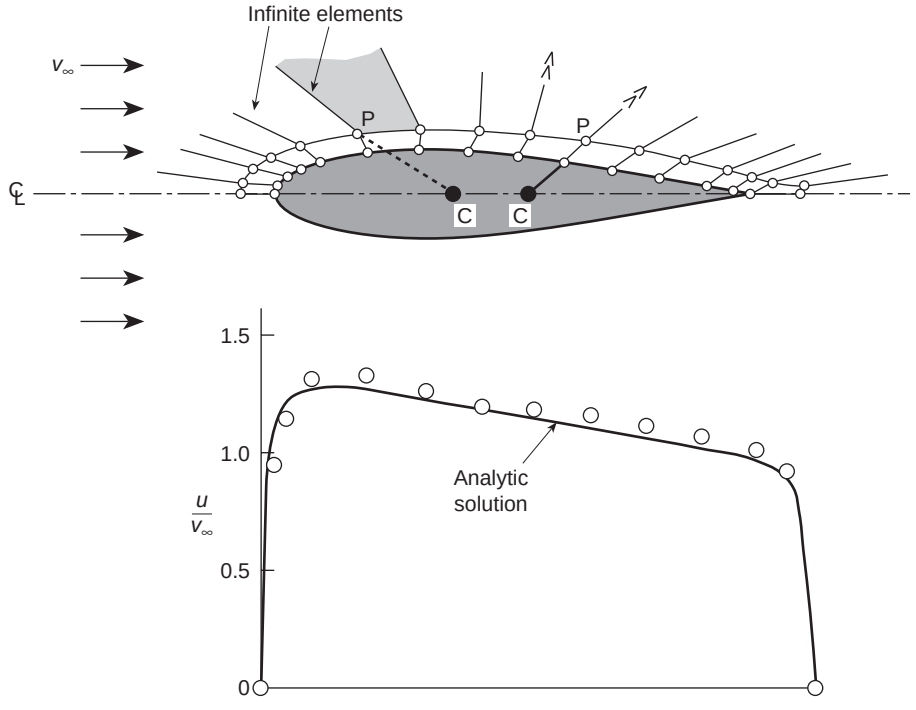


Fig. 9.21 Irrotational flow around a NACA 0018 wing section.[36] (a) Mesh of bilinear isoparametric and infinite elements. (b) Computed ○ and analytical — results for velocity parallel to surface.

from the corner node at which the singularity develops. Although good results are achievable with such elements the singularity is, in fact, not well modelled on lines other than element edges. A development suggested by Hibbitt[68] achieves a better result by using triangular second-order elements for this purpose [Fig. 9.22(*b*)].

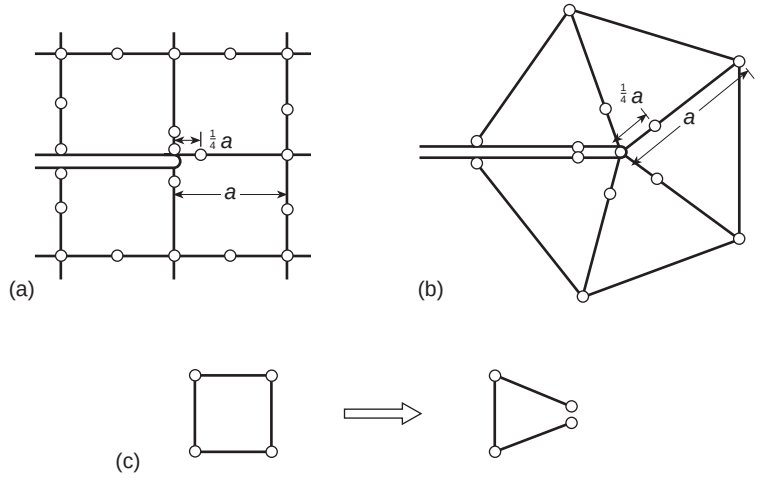


Fig. 9.22 Singular elements from degenerate isoparameters (a), (b), and (c).

Indeed, the use of distorted or degenerate isoparametrics is not confined to elastic singularities. Rice[56] shows that in the case of plasticity a shear strain singularity of $1/r$ type develops and Levy *et al.*[49] use an isoparametric, linear quadrilateral to generate such a singularity by the simple device of coalescing two nodes but treating these displacements independently. A variant of this is developed by Rice and Tracey.[45]

The elements just described are evidently simple to implement without any changes in a standard finite element program. However, in Chapter 16 we introduce a method whereby any singularity (or other function) can be modelled directly. We believe the methods to be described there supercede the above described techniques.

9.15 A computational advantage of numerically integrated finite elements[70]

One considerable gain that is possible in numerically integrated finite elements is the versatility that can be achieved in a single computer program.

It will be observed that for a *given class of problems* the general matrices are always of the same form [see the example of Eq. (9.8)] in terms of the shape function and its derivatives.

To proceed to evaluation of the element properties it is necessary first to *specify the shape function* and its derivatives and, second, to *specify the order of integration*.

The computation of element properties is thus composed of three distinct parts as shown in Fig. 9.23. For a *given class of problems* it is only necessary to change the prescription of the shape functions to achieve a variety of possible elements.

Conversely, the *same shape function* routines can be used in many different classes of problem, as is shown in Chapter 20.

Use of different elements, testing the efficiency of a new element in a given context, or extension of programs to deal with new situations can thus be readily achieved, and considerable algebra (with its inherent possibilities of mistakes) avoided.

The computer is thus placed in the position it deserves, i.e., of being the obedient slave capable of saving routine work.

The greatest practical advantage of the use of universal shape function routines is that they can be checked decisively for errors by a simple program with the patch test (viz. Chapter 10) playing the crucial role.

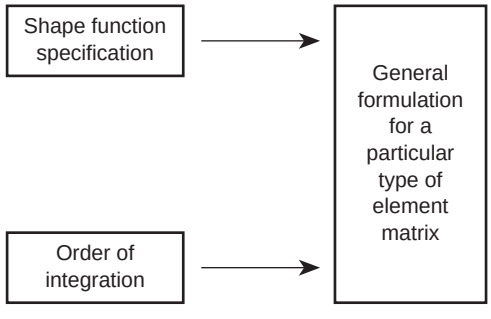


Fig. 9.23 Computation scheme for numerically integrated elements.

The incorporation of simple, exactly integrable, elements in such a system is, incidentally, not penalized as the time of exact and numerical integration in many cases is almost identical.

9.16 Some practical examples of two-dimensional stress analysis[71–77]

Some possibilities of two-dimensional analysis offered by curvilinear elements are illustrated in the following axisymmetric examples.

9.16.1 Rotating disc (Fig. 9.24)

Here only 18 elements are needed to obtain an adequate solution. It is of interest to observe that all mid-side nodes of the cubic elements are generated within a program and need not be specified.

9.16.2 Conical water tank (Fig. 9.25)

In this problem cubic elements are again used. It is worth noting that single-element thickness throughout is adequate to represent the bending effects in both the thick and thin parts of the container. With simple triangular elements, several layers of elements would have been needed to give an adequate solution.

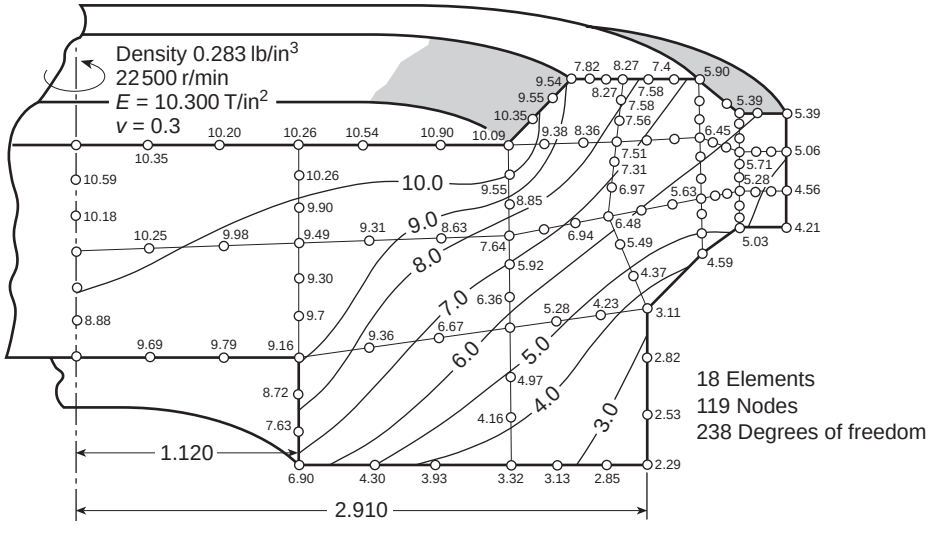


Fig. 9.24 A rotating disc analysed with cubic elements.

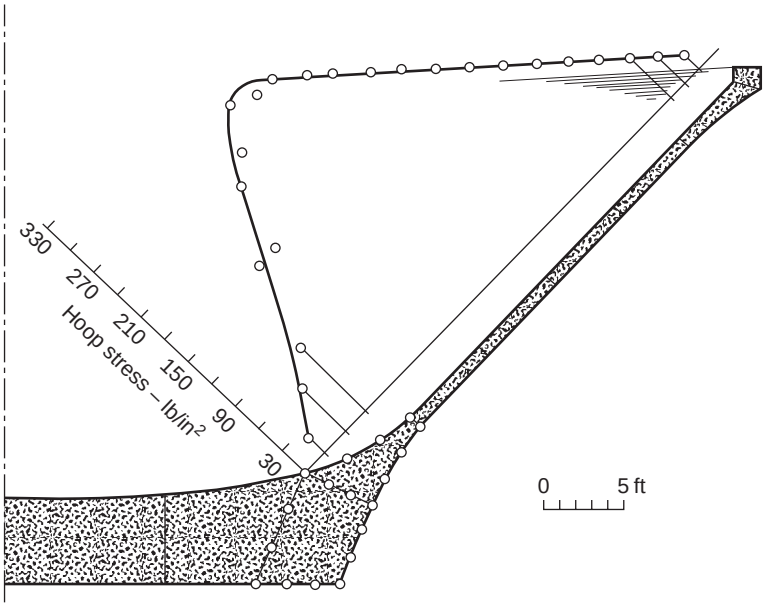


Fig. 9.25 Conical water tank.

9.16.3 A hemispherical dome (Fig. 9.26)

The possibilities of dealing with shells approached in the previous example are here further exploited to show how a limited number of elements can adequately solve a thin shell problem, with precisely the same program. This type of solution can be further improved upon from the economy viewpoint by making use of the well-known shell assumptions involving a linear variation of displacements across the thickness. Thus the number of degrees of freedom can be reduced. Methods of this kind will be dealt with in detail in the second volume of this text.

9.17 Three-dimensional stress analysis

In three-dimensional analysis, as was already hinted at in Chapter 6, the complex element presents a considerable economic advantage. Some typical examples are shown here in which the quadratic, serendipity-type formulation is used almost exclusively. In all problems integration using *three* Gauss points in each direction was used.

9.17.1 Rotating sphere (Fig. 9.27)

This example, in which the stresses due to centrifugal action are compared with exact

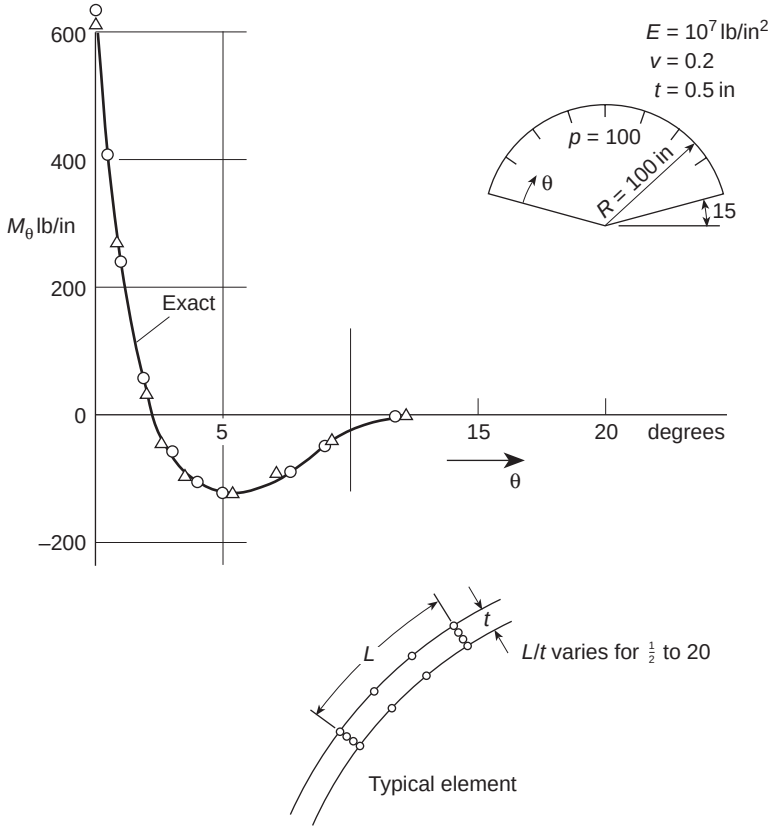


Fig. 9.26 *Encastré*, thin hemispherical shell. Solution with 15 and 24 cubic elements.

values, is perhaps a test on the efficiency of highly distorted elements. Seven elements are used here and results show good agreement with exact stresses.

9.17.2 Arch dam in rigid valley

This problem, perhaps a little unrealistic from the engineer's viewpoint, was the subject of a study carried out by a committee of the Institution of Civil Engineers and provided an excellent test for a convergence evaluation of three-dimensional analysis.[75] In Fig. 9.28 two subdivisions into quadratic and two into cubic elements are shown. In Fig. 9.29 the convergence of displacements in the centre-line section is shown, indicating that quite remarkable accuracy can be achieved with even one element.

The comparison of stresses in Fig. 9.30 is again quite remarkable, though showing a greater 'oscillation' with coarse subdivision. The finest subdivision results can be taken as 'exact' from checks by models and alternative methods of analysis.

The above test problems illustrate the general applicability and accuracy. Two further illustrations typical of real situations are included.

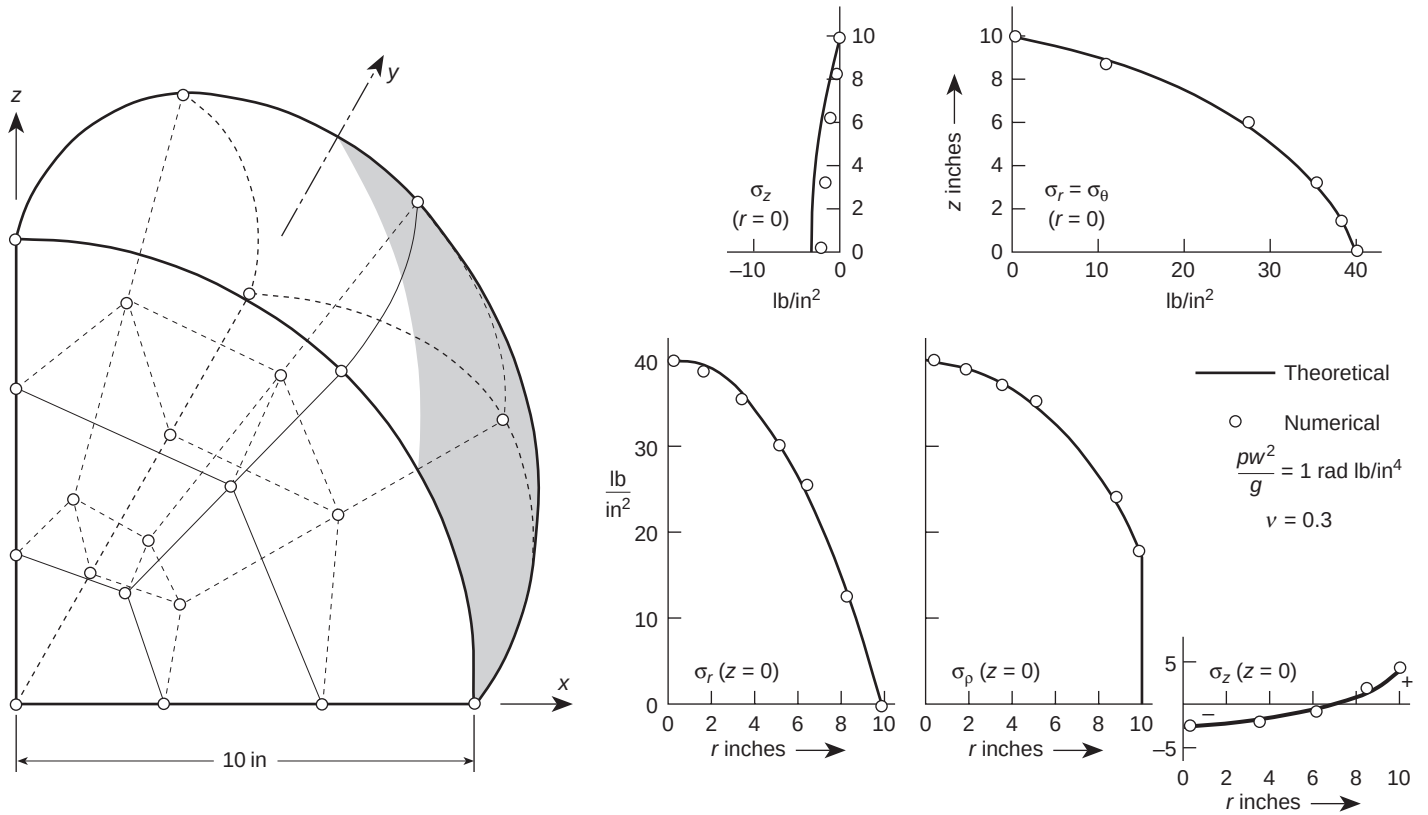


Fig. 9.27 A rotating sphere as a three-dimensional problem. Seven parabolic elements. Stresses along $z = 0$ and $r = 0$.

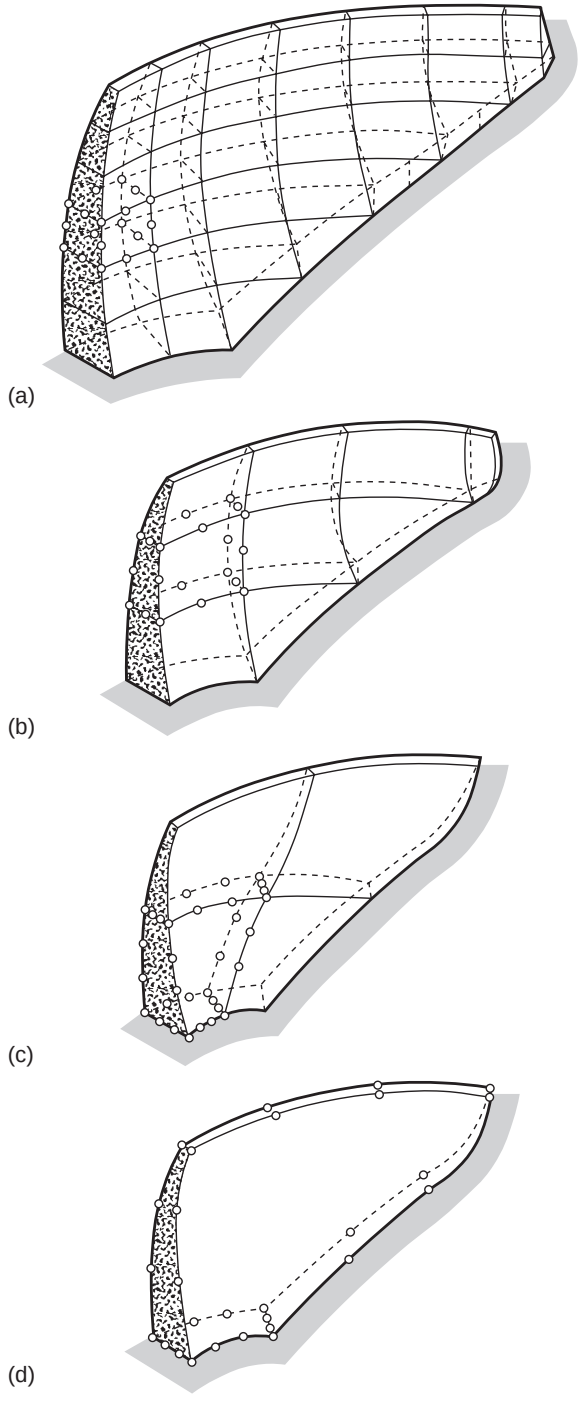


Fig. 9.28 Arch dam in a rigid valley – various element subdivisions.

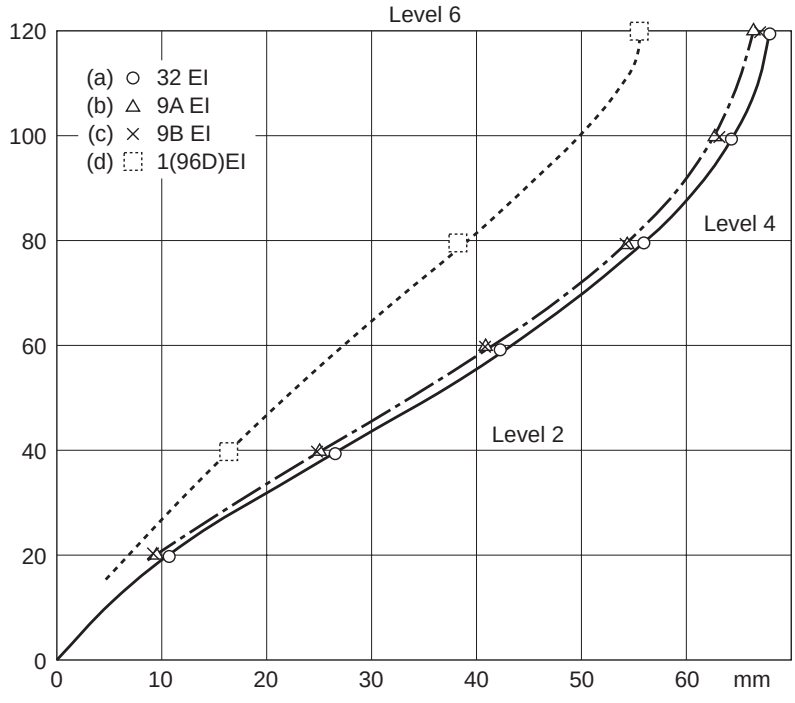


Fig. 9.29 Arch dam in a rigid valley – centre-line displacements.

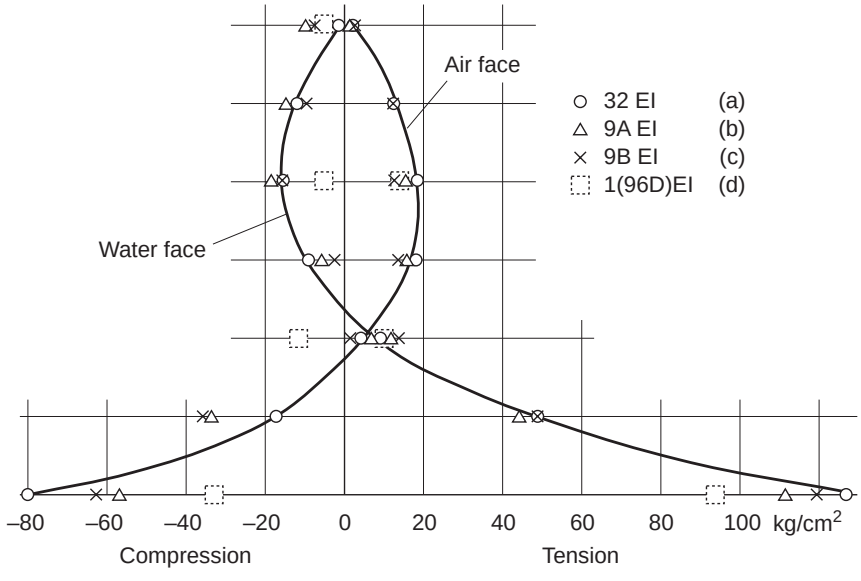


Fig. 9.30 Arch dam in a rigid valley – vertical stresses on centre-line.

9.17.3 Pressure vessel (Fig. 9.31): an analysis of a biomechanic problem (Fig. 9.32)

Both show subdivisions sufficient to obtain reasonable engineering accuracy. The pressure vessel, somewhat similar to the one indicated in Chapter 6, Fig. 6.7, shows the very considerable reduction of degrees of freedom possible with the use of more complex elements to obtain similar accuracy.

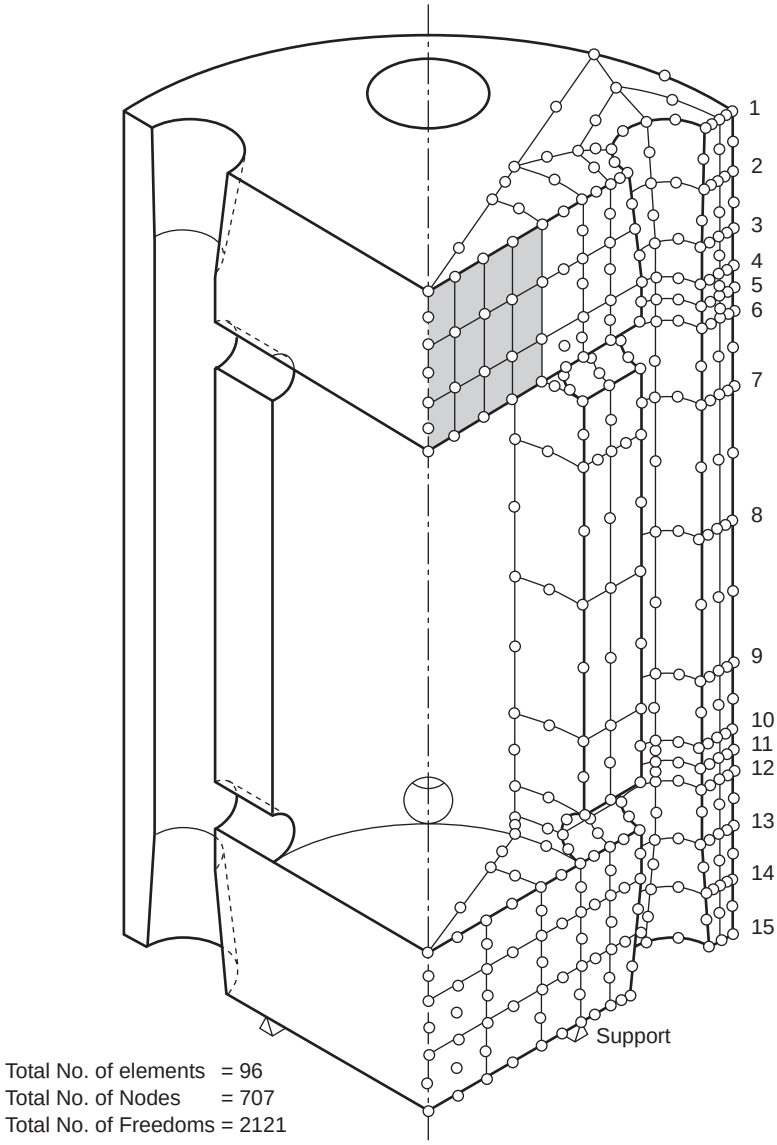


Fig. 9.31 Three-dimensional analysis of a pressure vessel.

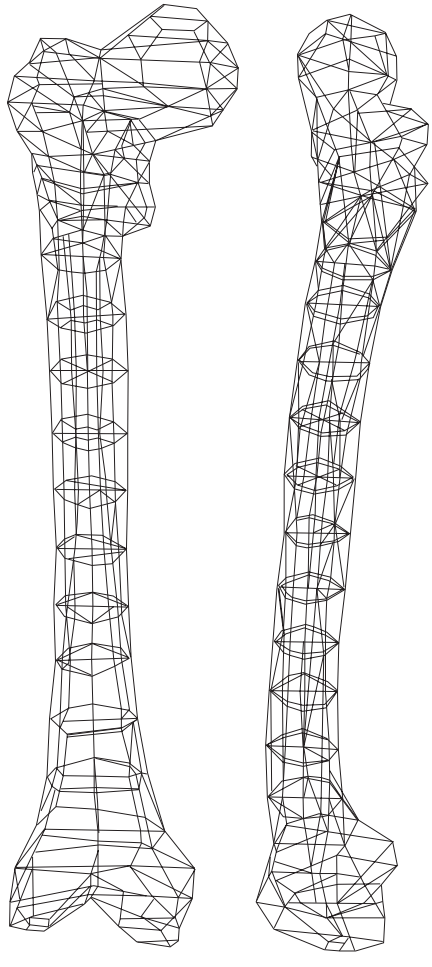


Fig. 9.32 A problem of biomechanics. Plot of linear element form only; curvature of elements omitted. Note degenerate element shapes.

The example of Fig. 9.32 shows a perspective view of the elements used. Such plots are not only helpful in visualization of the problem but also form an essential part of *data correctness checks* as any gross geometric error can be easily discovered.

The importance of avoiding data errors in complex three-dimensional problems should be obvious in view of their large usage of computer time. These, and indeed other,[76] checking methods must form an essential part of any computation system.

9.18 Symmetry and repeatability

In most of the problems shown, the advantage of symmetry in loading and geometry was taken when imposing the boundary conditions, thus reducing the whole problem to manageable proportions. The use of symmetry conditions is so well known to the

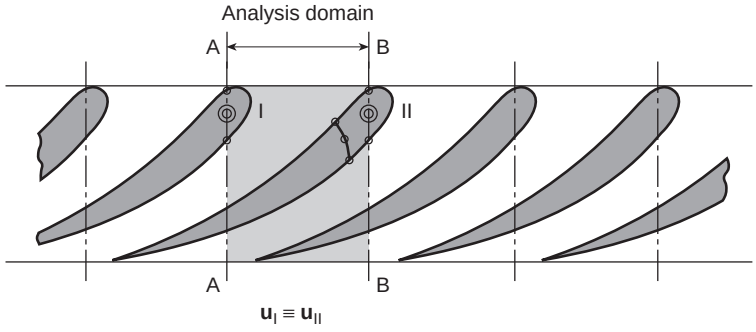


Fig. 9.33 Repeatability segments and analysis domain (shaded).

engineer and physicist that no statement needs to be made about it explicitly. Less known, however, appears to be the use of *repeatability*[78] when an identical structure (and) loading is continuously repeated, as shown in Fig. 9.33 for an infinite blade cascade. Here it is evident that a typical segment shown shaded behaves identically to the next one, and thus functions such as velocities and displacements at corresponding points of AA and BB are simply identified, i.e.,

$$\mathbf{u}_I = \mathbf{u}_{II}$$

This identification is made directly in a computer program.

Similar repeatability, in radial coordinates, occurs very frequently in problems involving turbine or pump impellers. Figure 9.34 shows a typical three-dimensional analysis of such a repeatable segment.

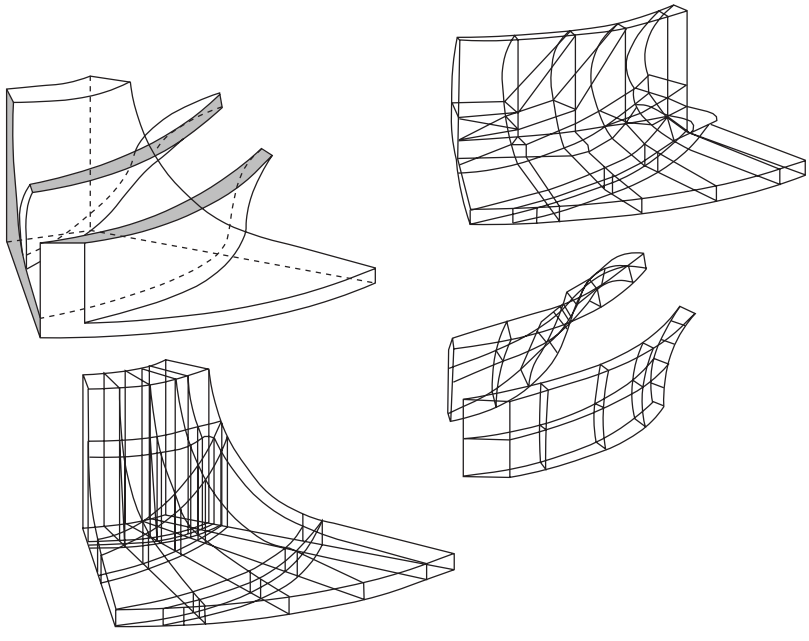


Fig. 9.34 Repeatable sector in analysis of an impeller.

References

1. I.C. Taig. *Structural analysis by the matrix displacement method*. Engl. Electric Aviation Report No. S017, 1961.
2. B.M. Irons. Numerical integration applied to finite element methods. *Conf. Use of Digital Computers in Struct. Eng*. Univ. of Newcastle, 1966.
3. B.M. Irons. Engineering application of numerical integration in stiffness method. *JAIAA*. **14**, 2035–7, 1966.
4. S.A. Coons. *Surfaces for computer aided design of space form*. MIT Project MAC, MAC-TR-41, 1967.
5. A.R. Forrest. *Curves and Surfaces for Computer Aided Design*. Computer Aided Design Group, Cambridge, England, 1968.
6. G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. pp. 156–63, Prentice-Hall, 1973.
7. E.L. Wachspress. High order curved finite elements. *Int. J. Num. Meth. Eng*. **17**, 735–45, 1981.
8. M. Crochet. Personal communication. 1988.
9. Nam-Sua Lee and K.-J. Bathe. Effects of element distortion on the performance of isoparametric elements. *Internat. J. Num. Meth. Eng*. **36**, 3553–76, 1993.
10. N. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1965.
11. B.M. Irons. Quadrature rules for brick based finite elements. *Int. J. Num. Meth. Eng*. **3**, 293–4, 1971.
12. T.K. Hellen. Effective quadrature rules for quadratic solid isoparametric finite elements. *Int. J. Num. Meth. Eng*. **4**, 597–600, 1972.
13. Radau. *Journ. de Math*. **3**, 283, 1880.
14. R.G. Anderson, B.M. Irons, and O.C. Zienkiewicz. Vibration and stability of plates using finite elements. *Int. J. Solids Struct*. **4**, 1031–55, 1968.
15. P.C. Hammer, O.P. Marlowe, and A.H. Stroud. Numerical integration over simplexes and cones. *Math. Tables Aids Comp*. **10**, 130–7, 1956.
16. C.A. Felippa. *Refined finite element analysis of linear and non-linear two-dimensional structures*. Structures Materials Research Report No. 66–22, Oct. 1966, Univ. of California, Berkeley.
17. G.R. Cowper. Gaussian quadrature formulas for triangles. *Int. J. Num. Mech. Eng*. **7**, 405–8, 1973.
18. I. Fried. Accuracy and condition of curved (isoparametric) finite elements. *J. Sound Vibration*. **31**, 345–55, 1973.
19. I. Fried. Numerical integration in the finite element method. *Comp. Struc*. **4**, 921–32, 1974.
20. M. Zlamal. Curved elements in the finite element method. *SIAM J. Num. Anal*. **11**, 347–62, 1974.
21. D. Kosloff and G.A. Frasier. Treatment of hour glass patterns in low order finite element codes. *Int. J. Num. Anal. Meth. Geomechanics*. **2**, 57–72, 1978.
22. T. Belytschko and W.E. Bachrach. The efficient implementation of quadrilaterals with high coarse mesh accuracy. *Comp. Methods Appl. Mech and Engineering*. **54**, 276–301, 1986.
23. O.C. Zienkiewicz and D.V. Phillips. An automatic mesh generation scheme for plane and curved element domains. *Int. J. Num. Meth. Eng*. **3**, 519–28, 1971.
24. W.J. Gordon and C.A. Hall. Construction of curvilinear co-ordinate systems and application to mesh generation. *Int. J. Num. Meth. Eng*. **7**, 461–77, 1973.
25. W.J. Gordon and C.A. Hall. Transfinite element methods – blending-function interpolation over arbitrary curved element domains. *Numer. Math*. **21**, 109–29, 1973.

26. J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comp. Phys.* **72**, 449–66, 1987.
27. J. Peraire, K. Morgan, M. Vahdati, and O.C. Zienkiewicz. Finite element Euler computations in 3-d. *Internat. J. Num. Meth. Eng.* **26**, 2135–59, 1988.
28. P. Kaupp and S. Steinberg. *Fundamentals of Grid Generation*. CRC Press, 1993.
29. N.P. Weatherill, P.R. Eiseman, J. Hause, and J.P. Thompson. *Numerical Grid Generation in Fluid Dynamics and Related Fields*. Pineridge Press, Swansea, 1994.
30. P.L. George and H. Borouchaki. *Delaunay Triangulation and Meshing*. Hermes, Paris, 1998.
31. J.F. Thompson, B.K. Soni, and N.P. Weatherill, eds. *Handbook of Grid Generation*. CRC Press, January 1999.
32. GiD–The Personal Pre/Postprocessor, CIMNE, Barcelona, Spain, 1999.
33. R.W. Thatcher. On the finite element method for unbounded regions. *SIAM J. Numerical Analysis*. **15**, 3, pp. 466–76, June 1978.
34. P. Silvester, D.A. Lowther, C.J. Carpenter, and E.A. Wyatt. Exterior finite elements for 2-dimensional field problems with open boundaries. *Proc. IEE*. **123**, No. 12, December 1977.
35. S.F. Shen. An aerodynamicist looks at the finite element method, in *Finite Elements in Fluids* (eds R.H. Gallagher *et al.*). Vol. 2, pp. 179–204, Wiley, 1975.
36. O.C. Zienkiewicz, D.W. Kelly, and P. Bettess. The coupling of the finite element and boundary solution procedures. *Int. J. Num. Meth. Eng.* **11**, 355–75, 1977.
37. P. Betess. Infinite elements. *Int. J. Num. Meth. Eng.* **11**, 53–64, 1977.
38. P. Bettess and O.C. Zienkiewicz. Diffraction and refraction of surface waves using finite and infinite elements. *Int. J. Num. Meth. Eng.* **11**, 1271–90, 1977.
39. G. Beer and J.L. Meek. Infinite domain elements. *Int. J. Num. Meth. Eng.* **17**, 43–52, 1981.
40. O.C. Zienkiewicz, C. Emson, and P. Bettess. A novel boundary infinite element. *Int. J. Num. Meth. Eng.* **19**, 393–404, 1983.
41. P. Bettess. *Infinite Elements*. Penshaw Press, 1992.
42. R.H. Gallagher. Survey and evaluation of the finite element method in fracture mechanics analysis, in *Proc. 1st Int. Conf. on Structural Mechanics in Reactor Technology*. Vol. 6, Part L, pp. 637–53, Berlin, 1971.
43. N. Levy, P.V. Marçal, and J.R. Rice. Progress in three-dimensional elastic–plastic stress analysis for fracture mechanics. *Nucl. Eng. Des.* **17**, 64–75, 1971.
44. J.J. Oglesby and O. Lomacky. An evaluation of finite element methods for the computation of elastic stress intensity factors. *J. Eng. Ind.* **95**, 177–83, 1973.
45. J.R. Rice and D.M. Tracey. Computational fracture mechanics, in *Numerical and Computer Methods in Structural Mechanics* (eds S.J. Fenves *et al.*). pp. 555–624, Academic Press, 1973.
46. A.A. Griffiths. The phenomena of flow and rupture in solids. *Phil. Trans. Roy. Soc. (London)*. **A221**, 163–98, Oct. 1920.
47. J.L. Swedlow. Elasto-plastic cracked plates in plane strain. *Int. J. Fract. Mech.* **4**, 33–44, March 1969.
48. T. Yokobori and A. Kamei. The size of the plastic zone at the tip of a crack in plane strain state by the finite element method. *Int. J. Fract. Mech.* **9**, 98–100, 1973.
49. N. Levy, P.V. Marçal, W.J. Ostergren, and J.R. Rice. Small scale yielding near a crack in plane strain: a finite element analysis. *Int. J. Fract. Mech.* **7**, 143–57, 1967.
50. J.R. Dixon and L.P. Pook. Stress intensity factors calculated generally by the finite element technique. *Nature*. **224**, 166, 1969.
51. J.R. Dixon and J.S. Strannigan. Determination of energy release rates and stress-intensity factors by the finite element method. *J. Strain Analysis*. **7**, 125–31, 1972.
52. V.B. Watwood. Finite element method for prediction of crack behavior. *Nucl. Eng. Des.* **II** (No. 2), 323–32, March 1970.

53. D.F. Mowbray. A note on the finite element method in linear fracture mechanics. *Eng. Fract. Mech.* **2**, 173–6, 1970.
54. D.M. Parks. A stiffness derivative finite element technique for determination of elastic crack tip stress intensity factors. *Int. J. Fract.* **10**, 487–502, 1974.
55. T.K. Hellen. On the method of virtual crack extensions. *Int. J. Num. Meth. Eng.* **9** (No. 1), 187–208, 1975.
56. J.R. Rice. A path-independent integral and the approximate analysis of strain concentration by notches and cracks. *J. Appl. Mech.. Trans. Am. Soc. Mech. Eng.* **35**, 379–86, 1968.
57. P. Tong and T.H.H. Pian. On the convergence of the finite element method for problems with singularity. *Int. J. Solids Struct.* **9**, 313–21, 1972.
58. T.A. Cruse and W. Vanburen. Three dimensional elastic stress analysis of a fracture specimen with edge crack. *Int. J. Fract. Mech.* **7**, 1–15, 1971.
59. E. Byskov. The calculation of stress intensity factors using the finite element method with cracked elements. *Int. J. Fract. Mech.* **6**, 159–67, 1970.
60. P.F. Walsh. Numerical analysis in orthotropic linear fracture mechanics. *Inst. Eng. Australia. Civ. Eng.. Trans.* **15**, 115–19, 1973.
61. P.F. Walsh. The computation of stress intensity factors by a special finite element technique. *Int. J. Solids Struct.* **7**, 1333–42, Oct. 1971.
62. A.K. Rao, I.S. Raju, and A. Murthy Krishna. A powerful hybrid method in finite element analysis. *Int. J. Num. Meth. Eng.* **3**, 389–403, 1971.
63. W.S. Blackburn. Calculation of stress intensity factors at crack tips using special finite elements, in *The Mathematics of Finite Elements* (ed. J.R. Whiteman), pp. 327–36, Academic Press, 1973.
64. D.M. Tracey. Finite elements for determination of crack tip elastic stress intensity factors. *Eng. Fract. Mech.* **3**, 255–65, 1971.
65. R.D. Henshell and K.G. Shaw. Crack tip elements are unnecessary. *Int. J. Num. Meth. Eng.* **9**, 495–509, 1975.
66. R.S. Barsoum. On the use of isoparametric finite elements in linear fracture mechanics. *Int. J. Num. Meth. Eng.* **10**, 25–38, 1976.
67. R.S. Barsoum. Triangular quarter point elements as elastic and perfectly elastic crack tip elements. *Int. J. Num. Meth. Eng.* **11**, 85–98, 1977.
68. H.D. Hibbitt. Some properties of singular isoparametric elements. *Int. J. Num. Meth. Eng.* **11**, 180–4, 1977.
69. S.E. Benzley. Representation of singularities with isoparametric finite elements. *Int. J. Num. Meth. Eng.* **8** (No. 3), 537–45, 1974.
70. B.M. Irons. Economical computer techniques for numerically integrated finite elements. *Int. J. Num. Meth. Eng.* **1**, 201–3, 1969.
71. O.C. Zienkiewicz, B.M. Irons, J.G. Ergatoudis, S. Ahmad, and F.C. Scott. Isoparametric and associated element families for two and three dimensional analysis, in *Proc. Course on Finite Element Methods in Stress Analysis* (eds I. Holland and K. Bell). Trondheim Tech. University, 1969.
72. B.M. Irons and O.C. Zienkiewicz. The isoparametric finite element system – a new concept in finite element analysis. *Proc. Conf. Recent Advances in Stress Analysis*. Royal Aero Soc., 1968.
73. J.G. Ergatoudis, B.M. Irons, and O.C. Zienkiewicz. Curved, isoparametric. 'quadrilateral' elements for finite element anlysis. *Int. J. Solids Struct.* **4**, 31–42, 1968.
74. J.G. Ergatoudis. *Isoparametric elements in two and three dimensional analysis*. Ph.D. thesis, University of Wales, Swansea, 1968.
75. J.G. Ergatoudis, B.M. Irons, and O.C. Zienkiewicz. Three dimensional analysis of arch dams and their foundations. *Symposium on Arch Dams*. Inst. Civ. Eng., London, 1968.
76. O.C. Zienkiewicz, B.M. Irons, J. Campbell, and F.C. Scott. Three dimensional stress analysis. *Int. Un. Th. Appl. Mech. Symp. on High Speed Computing in Elasticity*. Liège, 1970.

77. O.C. Zienkiewicz. Isoparametric and other numerically integrated elements, in *Numerical and Computer Methods in Structural Mechanics* (eds S.J. Fenves, N. Perrone, A.R. Robinson, and W.C. Schnobrich). pp. 13–41, Academic Press, 1973.
78. O.C. Zienkiewicz and F.C. Scott. On the principle of repeatability and its application in analysis of turbine and pump impellers. *Int. J. Num. Meth. Eng.* **9**, 445–52, 1972.

10

The patch test, reduced integration, and non-conforming elements

10.1 Introduction

We have briefly referred in Chapter 2 to the patch test as a means of assessing convergence of displacement-type elements for elasticity problems in which the shape functions violate continuity requirements. In this chapter we shall deal in more detail with this test which is *applicable to all finite element forms* and will show that

(a) it is a *necessary* condition for assessing the convergence of any finite element approximation and further that, if properly extended and interpreted, it can provide
(b) a *sufficient* requirement for convergence,
(c) an assessment of the (asymptotic) convergence rate of the element tested,
(d) a check on the robustness of the algorithm, and
(e) a means of developing new and accurate finite element forms which violate compatibility (continuity) requirements.

While for elements which *a priori* satisfy all the continuity requirements, have correct polynomial expansions, and are exactly integrated such a test is superfluous in principle, it is nevertheless useful as it gives

(f) a check that correct programming was achieved.

For all the reasons cited above the patch test has been, since its inception, and continues to be the most important check for practical finite element codes.

The original test was introduced by Irons *et al.*[1–3] in a physical way and could be interpreted as a check which ascertained whether a patch of elements (Fig. 10.1) subject to a constant strain reproduced exactly the constitutive behaviour of the material and resulted in correct stresses when it became infinitesimally small. If it did, it could then be argued that the finite element model represented the real material behaviour and, in the limit, as the size of the elements decreased would therefore reproduce exactly the behaviour of the real structure.

Clearly, although this test would only have to be passed when the size of the element patch became infinitesimal, for most elements in which polynomials are used the patch size did not in fact enter the consideration and the requirement that the patch test be passed for any element size became standard.

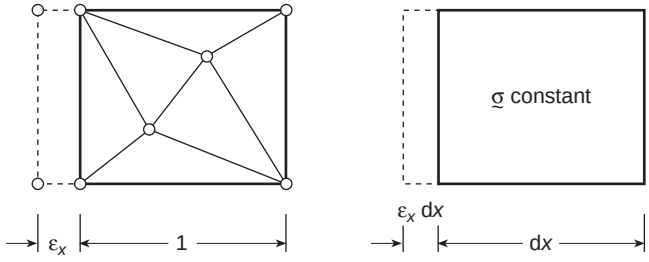


Fig. 10.1 A patch of element and a volume of continuum subject to constant strain ε_x. A physical interpretation of the constant strain or linear displacement field patch test.

Quite obviously a rigid body displacement of the patch would cause no strain, and if the proper constitutive laws were reproduced no stress changes would result. The patch test thus guarantees that no rigid body motion straining will occur.

When curvilinear coordinates are used the patch test is still required to be passed in the limit but generally will not do so for a finite size of the patch. (An exception here is the isoparametric coordinate system in problems discussed in Chapter 9 since it is guaranteed to contain linear polynomials in the global coordinates.) Thus for many problems such as shells, where local curvilinear coordinates are used, this test has to be restricted to infinitesimal patch sizes and, on physical grounds alone, appears to be a *necessary and sufficient condition* for convergence.

Numerous publications on the theory and practice of the test have followed the original publications cited[4–6] and mathematical respectability was added to those by Strang.[7,8] Although some authors have cast doubts on its validity[9,10] these have been fully refuted[11–13] and if the test is used as described here it fulfils the requirements (a)–(f) stated above.

In the present chapter we consider the patch test applied to irreducible forms (see Chapter 3) but an extension to mixed forms is more important. This has been studied in references 13, 14 and 15 and made use of in many subsequent publications. The matter of mixed form patch tests will be fully discussed in the next chapter; however, the consistency and stability tests developed in the present chapter are always required.

One additional use of the patch test was suggested by Babuška *et al.*[16] with a shorter description given by Boroomand and Zienkiewicz.[17] This test can establish the efficiency of gradient (stress) recovery processes which are so important in error estimation as will be discussed in Chapter 14.

10.2 Convergence requirements

We shall consider in the following the patch test as applied to a finite element solution of a set of differential equations

$$\mathbf{A}(\mathbf{u}) \equiv \mathbf{L}(\mathbf{u}) + \mathbf{g} = \mathbf{0} \tag{10.1}$$

in the domain Ω together with the conditions

$$\mathbf{B}(\mathbf{u}) = \mathbf{0} \tag{10.2}$$

on the boundary of the domain, Γ.

The finite element approximation is given in the form

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}\mathbf{a} \tag{10.3}$$

where $\mathbf{N}$ are shape functions defined in each element, Ω_e, and $\mathbf{a}$ are unknown parameters.

By applying standard procedures of finite element approximation (see Chapters 2 and 3) the problem reduces in a linear case to a set of algebraic equations

$$\mathbf{K}\mathbf{a} = \mathbf{f} \tag{10.4}$$

which when solved give an approximation to the differential equation and its boundary conditions.

What is meant by 'convergence' in the approximation sense is that the approximate solution, $\hat{\mathbf{u}}$, should tend to the exact solution $\mathbf{u}$ when the size of the elements h approaches zero (with some specified subdivision pattern). Stated mathematically we must find that the error at any point becomes (when h is sufficiently small)

$$|\mathbf{u} - \hat{\mathbf{u}}| = O(h^q) \leqslant Ch^q \tag{10.5}$$

where $q > 0$ and C is a positive constant, depending on the position.

This must also be true for all the derivatives of $\mathbf{u}$ defined in the approximation.

By the order of convergence in the variable $\mathbf{u}$ we mean the value of the index q in the above definition.

To ensure convergence it is necessary that the approximation fulfil both consistency and stability conditions.[18]

The *consistency requirement* ensures that as the size of the elements h tends to zero, the approximation equation (10.4) will represent the exact differential equation (10.1) and the boundary conditions (10.2) (at least in the weak sense).

The *stability condition* is simply translated as a requirement that the solution of the discrete equation system (10.4) be unique and avoid spurious mechanisms which may pollute the solution for all sizes of elements. For linear problems in which we solve the system of algebraic equations (10.4) as

$$\mathbf{a} = \mathbf{K}^{-1}\mathbf{f} \tag{10.6}$$

this means simply that the matrix $\mathbf{K}$ must be non-singular for all possible element assemblies (subject to imposing minimum stable boundary conditions).

The patch test traditionally has been used as a procedure for verifying the consistency requirement; the stability was checked independently by ensuring non-singularity of matrices.[19] Further, it generally tested only the consistency in satisfaction of the differential equation (10.1) but not of its natural boundary conditions. In what follows we shall show how all the necessary requirements of convergence can be tested by a properly conceived patch test.

A 'weak' singularity of a single element may on occasion be permissible and some elements exhibiting it have been, and still are, successfully used in practice. One such case is given by the eight-node isoparametric element with a 2×2 Gauss quadrature, to which we shall refer later here. This element is on occasion observed to show peculiar behaviour (though its use has advantages as discussed in Chapter 11). An element that occasionally fails is termed *non-robust* and the patch test provides a means of assessing the *degree of robustness*.

10.3 The simple patch test (tests A and B) – a necessary condition for convergence

We shall first consider the consistency condition which requires that in the limit (as h tends to zero) the finite element approximation of Eq. (10.4) should model exactly the differential equation (10.1) and the boundary conditions (10.2). If we consider a 'small' region of the domain (of size $2h$) we can expand the unknown function u and the essential derivatives entering the weak approximation in a Taylor series. From this we conclude that for convergence of the function and its first derivative in typical problems of a second-order equation and two dimensions, we require that around a point i assumed to be at the coordinate origin,

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_i + \left(\frac{\partial \mathbf{u}}{\partial x}\right)_i x + \left(\frac{\partial \mathbf{u}}{\partial y}\right)_i y + \cdots + O(h^p) \\
\frac{\partial \mathbf{u}}{\partial x} &= \left(\frac{\partial \mathbf{u}}{\partial x}\right)_i + \cdots + O(h^{p-1}) \\
\frac{\partial \mathbf{u}}{\partial y} &= \left(\frac{\partial \mathbf{u}}{\partial y}\right)_i + \cdots + O(h^{p-1})
\end{aligned} \tag{10.7}$$

with $p \geqslant 2$. The finite element approximation should therefore reproduce exactly the problem posed for *any linear forms* of u as h tends to zero. Similar conditions can obviously be written for higher order problems. This requirement is tested by the current interpretation of the patch test illustrated in Fig. 10.2. We refer to this as the *base solution*.

In this we compute first an arbitrary linear solution of the differential equation and the corresponding set of parameters $\mathbf{a}$ [see Eq. (10.3)] at all 'nodes' of a *patch* which assembles completely the nodal variable $\mathbf{a}_i$ (i.e., provides all the equation terms corresponding to it).

In *test A* we simply insert the exact value of the parameters $\mathbf{a}$ into the ith equation and verify that

$$\mathbf{K}_{ij}\mathbf{a}_j - \mathbf{f}_i \equiv \mathbf{0} \tag{10.8}$$

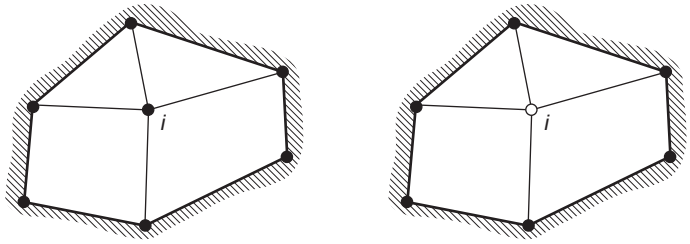


Fig. 10.2 Patch test of forms A and B.

where $\mathbf{f}_i$ is a force which results from any 'body force' required to satisfy the base solution differential equation (10.1). Generally in problems given in cartesian coordinates the required body force is zero; however, in curvilinear coordinates (e.g., axisymmetric elasticity problems) it can be non-zero.

In *test B* only the values of $\mathbf{a}$ corresponding to the boundaries of the 'patch' are inserted and $\mathbf{a}_i$ is found as

$$\mathbf{a}_i = \mathbf{K}_{ii}^{-1}(\mathbf{f}_i - \mathbf{K}_{ij}\mathbf{a}_j) \qquad j \neq i \tag{10.9}$$

and compared against the exact value.

Both patch tests verify only the satisfaction of the basic differential equation and not of the boundary approximations, as these have been explicitly excluded here.

We mentioned earlier that the test is, in principle, required only for an infinitesimally small patch of elements; however, for differential equations with constant coefficients and with a mapping involving constant jacobian the size of the patch is immaterial and the test can be carried out on a patch of arbitrary dimensions.

Indeed, if the coefficients are not constant the same size independence exists providing that a constant set of such coefficients is used in the formulation of the test. (This applies, for instance, in axisymmetric problems where coefficients of the type 1/radius enter the equations and when the patch test is here applied, it is simply necessary to enter the computation with such quantities assumed constant.)

If mapped curvilinear elements are used it is not obvious that the patch test posed in global coordinates needs to be satisfied. Here, in general, convergence in the mapping coordinates may exist but a finite patch test may not be satisfied. However, once again if we specify the nature of the subdivision without changing the mapping function, in the limit the jacobian becomes locally constant and the previous remarks apply. To illustrate this point consider, for instance, a set of elements in which local coordinates are simply the polar coordinates as shown in Fig. 10.3. With shape functions using polynomial expansions in the r, θ terms the patch test of the kind we have described above will not be satisfied with elements of finite size – nevertheless in the limit as the element size tends to zero it will become true. Thus it is evident that patch test satisfaction is a *necessary condition* which has always to be achieved *providing the size of the patch is infinitesimal*.

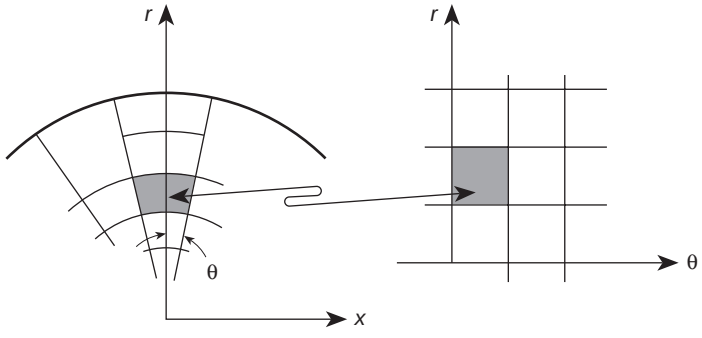


Fig. 10.3 Polar coordinate mapping.

This proviso which we shall call *weak patch test satisfaction* is not always simple to verify, particularly if the element coding does not easily permit the insertion of constant coefficients or a jacobian. In Sec. 10.10 we shall discuss in some detail its implementation, which, however, is only necessary in very special element forms. It is indeed fortunate that the standard isoparametric element form reproduces exactly the linear polynomial global coordinates (see Chapter 9) and for this reason does not require special treatment unless some other *crime* (such as selective or reduced integration) is introduced.

10.4 Generalized patch test (test C) and the single-element test

The patch test described in the preceding section was shown to be a necessary condition for convergence of the formulation but did not establish sufficient conditions for it. In particular, it omitted the testing of the boundary 'load' approximation for the case when the 'natural' (e.g. 'traction of elasticity') conditions are specified. Further it did not verify the stability of the approximation. A test including a check on the above conditions is easily constructed. We show this in Fig. 10.4 for a two-dimensional plane problem as *test C*. In this the patch of elements is assembled as before but subject to prescribed natural boundary conditions (or tractions around its perimeter) corresponding to the base function. The assembled matrix of the whole patch is written as

$$\mathbf{K}\mathbf{a} = \mathbf{f}$$

Fixing only the minimum number of parameters $\mathbf{a}$ necessary to obtain a physically valid solution (e.g., eliminating the rigid body motion in an elasticity example or a single value of temperature in a heat conduction problem) a solution is sought for remaining $\mathbf{a}$ values and compared with the exact base solution assumed.

Now any singularity of the $\mathbf{K}$ matrix will be immediately observed and, as the vector $\mathbf{f}$ includes all necessary source and boundary traction terms, the formulation will be completely tested (providing of course a sufficient number of test states is used). The test described is now not only *necessary* but *sufficient* for convergence.

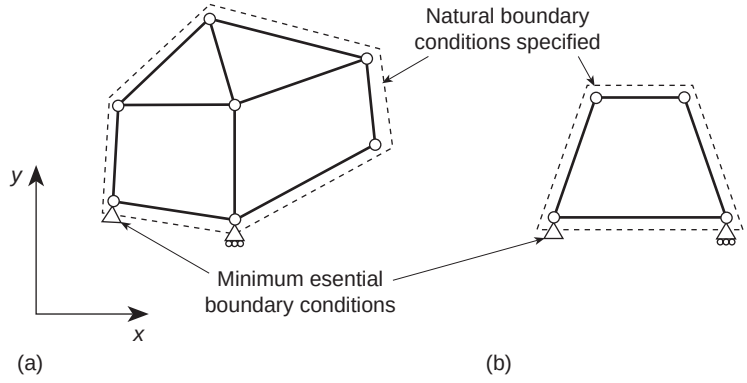


Fig. 10.4 (a) Patch test of form C. (b) The single-element test.

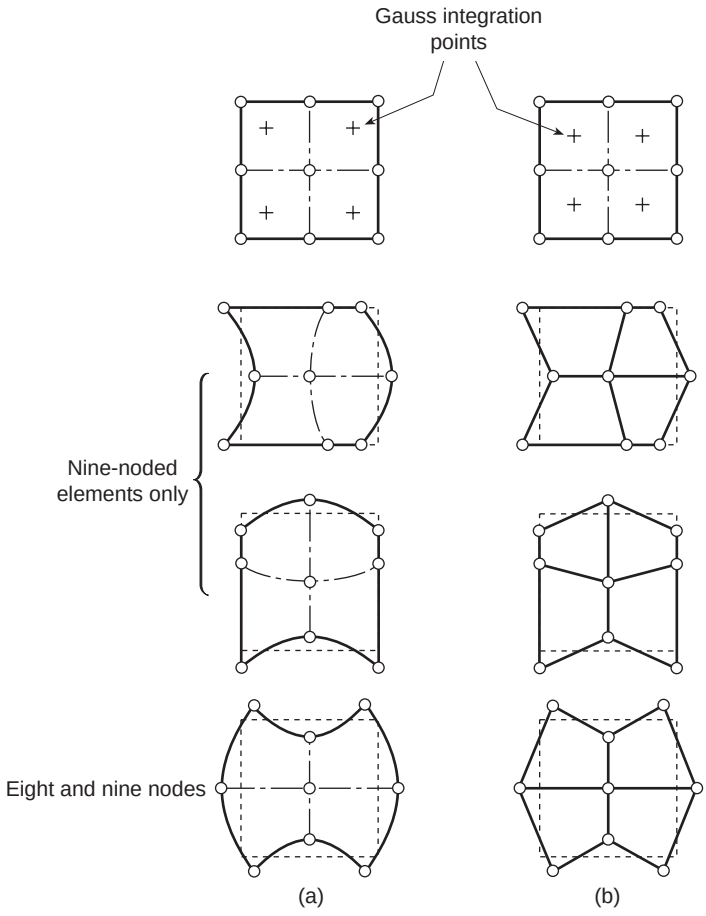


Fig. 10.5 (a) Zero energy (singular) modes for eight- and nine-noded quadratic elements and (b) for a patch of bilinear elements with single integration points.

With boundary traction included it is of course possible to reduce the size of the patch to a single element and an alternative form of test C is illustrated in Fig. 10.4(*b*), which is termed the single-element test.[11] This test is indeed one requirement of a good finite element formulation as, on occasion, a larger patch may not reveal the inherent instabilities of a single element. This happens in the well-documented case of the plane strain–stress eight-noded isoparametric element with (reduced) four-point Gauss quadrature i.e., where the singular deformation mode of a single element (see Fig. 10.5) disappears when several elements are assembled.† *It should be noted, however, that satisfaction of a single element test is not a sufficient condition for convergence. For sufficiency we require at least one internal element boundary to test that consistency of a patch solution is maintained between elements.*

† This figure also shows a similar singularity for a patch of four bilinear elements with single-point quadrature, and we note the similar shape of zero energy modes (see Chapter 9, Sec. 9.11.3).

10.5 The generality of a numerical patch test

In the previous section we have defined in some detail the procedures for conducting a patch test. We have also asserted the fact that such tests if passed guarantee that convergence will occur. However all the tests are numerical and it is impractical to test all possible combinations.

In particular let us consider the base solutions used. These will invariably be a set of polynomials given in two dimensions as

$$\mathbf{u} = \sum \boldsymbol{\alpha}_i P_i(x, y) \tag{10.10}$$

where P_i are a suitable set of low order polynomials (e.g., 1, x, y for Galerkin forms possessing only first-order derivatives) and $\boldsymbol{\alpha}_i$ are parameters. It is fairly obvious that if patch tests are conducted on each of these polynomials individually any base function of the form given in Eq. (10.10) can be reproduced and the generality preserved for the particular combination of elements tested. This must always be done and is almost a standard procedure in engineering tests, necessitating only a limited number of combinations.

However, as various possible patterns of elements can occur and it is possible to increase the size without limit the reader may well ask whether the test is complete from the geometrical point of view. We believe it is necessary in a numerical test to consider the possibility of several pathological arrangements of elements but that if the test is purely limited to a single element and a complete patch around a node we can be confident about the performance on more general geometric patterns.

Indeed even mathematical assessments of convergence are subject to limits often imposed *a posteriori*. Such limits may arise if for instance a singular mapping is used.

The procedures referred to in this section satisfy most readers as to the validity and generality of the test.

On some limited occasions it is possible to perform the test purely algebraically and then its validity cannot be doubted. Some such algebraic tests will be referred to later in connection with incompatible elements.

In this chapter we have only considered linear differential equations and linear material behaviour. In Volume 2 non-linear problems will be fully discussed and on some occasions the patch test can well be used and extended to cover such areas.

10.6 Higher order patch tests[6,8]

While the patch tests discussed in the last three sections ensure (when satisfied) that convergence will occur, they did not test the order of this convergence, beyond assuring us that in the case of Eq. (10.7) the errors were, at least, of order $O(h^2)$ in u. It is an easy matter to determine the actual highest asymptotic rate of convergence of a given element by simply imposing, instead of a linear solution, exact higher order polynomial solutions. The highest value of such polynomials for which complete satisfaction of the patch test is achieved automatically evaluates the corresponding convergence rate. It goes without saying that for such exact solutions generally non-zero source (e.g., body force) terms in the original equation (10.1) will need to be involved.

In addition, test C in conjunction with a higher order patch test may be used to illustrate any tendency for 'locking' to occur (see Chapter 11). Accordingly, element robustness with regard to various parameters (e.g., Poisson's ratios near one-half for elasticity problems in plane strain) may be established.

In such higher order patch tests it will of course first be assumed that the patch is subject to the base expansion solution as described. Thus, for higher order terms it will be necessary to start and investigate solutions of the type

$$\boldsymbol{\alpha}_3 x^2 + \boldsymbol{\alpha}_4 xy + \boldsymbol{\alpha}_5 y^2 + \cdots$$

each of which should be applied individually or as linearly independent combinations and for each the solution should be appropriately tested.

In particular, we shall expect higher order elements to exactly satisfy certain order solutions. However in Chapter 14 we shall use this idea to find the error between the exact solution and the recovery using precisely the same type of formulation.

10.7 Application of the patch test to plane elasticity elements with 'standard' and 'reduced' quadrature

In the next few sections we consider several applications of the patch test in the evaluation of finite element models. In each case we consider only one of the necessary tests which need to be implemented. For a complete evaluation of a formulation it is necessary to consider all possible independent base polynomial solutions as well as a variety of patch configurations which test the effects of element distortion or alternative meshing interconnections which will be commonly used in analysis. As we shall emphasize, it is important that both consistency and stability be evaluated in a properly conducted test.

In Chapter 9 (Sec. 9.11) we have discussed the minimum required order of numerical integration for various finite element problems which results in no loss of convergence rate. However, it was also shown that for some elements such a minimum integration order results in singular matrices. If we define the *standard* integration as one which evaluates the stiffness of an element exactly (at least in the undistorted form) then any lower order of integration is called *reduced*.

Such *reduced* integration has some merits in certain problems for reasons which we shall discuss in Chapter 12 (Sec. 12.5), but it can cause singularities which should be discovered by a patch test (which supplements and verifies the arguments of Sec. 9.11.3). Application of the patch test to some typical problems will now be shown.

10.7.1 Example 1: Patch test for base solution

We consider first a plane stress problem on the patch shown in Fig. 10.6(*a*). The material is linear, isotropic elastic with properties $E = 1000$ and $\nu = 0.3$. The finite element procedure used is based on the displacement form using four-noded isoparametric shape functions and numerical integration. Analyses are conducted using the plane element and program described in Chapter 20. Since the stiffness computation

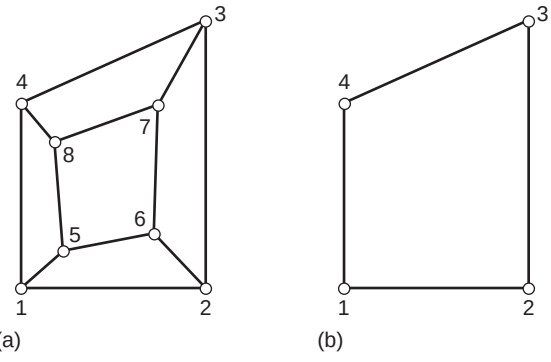


Fig. 10.6 Patch for evaluation of numerically integrated plane stress problems. (a) Five-element patch. (b) One-element patch.

includes only first derivatives of displacements, the formulation converges provided that the patch test is satisfied for all linear polynomial solutions of displacements in the base solution. Here we consider only one of the six independent linear polynomial solutions necessary to verify satisfaction of the patch test. The solution considered is

$$\begin{aligned} u &= 0.002x \\ v &= -0.0006y \end{aligned} \tag{10.11}$$

which produces zero body forces and zero stresses except for

$$\sigma_x = 2 \tag{10.12}$$

The solution given in Table 10.1 is obtained for the nodal displacements and satisfies Eq. (10.10) exactly.

The patch test is performed first using 2×2 gaussian 'standard' quadrature to compute each element stiffness and resulting reaction forces at nodes. For patch test A all nodes are restrained and nodal displacement values are specified according to Table 10.1. Stresses are computed at specified Gauss points (1×1, 2×2, and 3×3 Gauss points were sampled) and all are exact to within round-off error (double precision was used which produced round-off errors less than 10^{-15} in the quantities computed). Reactions were also computed at all nodes and again produced the

Table 10.1 Patch solution for Fig. 10.6

	Coordinates		Computed displacements		Forces	
Node i	x_i	y_i	u_i	v_i	F_{x_i}	F_{y_i}
1	0.0	0.0	0.0	0.0	−2	0
2	2.0	0.0	0.0040	0.0	3	0
3	2.0	3.0	0.0040	−0.00186	2	0
4	0.0	2.0	0.0	−0.00120	−3	0
5	0.4	0.4	0.0008	−0.00024	0	0
6	1.4	0.6	0.0028	−0.00036	0	0
7	1.5	2.0	0.0030	−0.00120	0	0
8	0.3	1.6	0.0006	−0.00096	0	0

force values shown in Table 10.1 to within round-off limits. This approximation satisfies all conditions required for a finite element procedure (i.e., conforming shape functions and normal-order quadrature). Accordingly, the patch test merely verifies that the programming steps used contain no errors. Patch test A does not require explicit use of the stiffness matrix to compute results; consequently the above patch test was repeated using patch test B where only nodes 1 to 4 are restrained with their displacements specified according to Table 10.1. This tests the accuracy of the stiffness matrix and, as expected, exact results are once again recovered to within round-off errors. Finally, patch test C was performed with node 1 fully restrained and node 4 restrained only in the x-direction. Nodal forces were applied to nodes 2 and 3 in accordance with the values generated through the boundary tractions by σ_x (i.e., nodal forces shown in Table 10.1). This test also produced exact solutions for all other nodal quantities in Table 10.1 and recovered σ_x of 2 at all Gauss points in each element.

The above test was repeated for patch tests A, B, and C but using a 1×1 'reduced' Gauss quadrature to compute the element stiffness and nodal force quantities. Patch test C indicated that the global stiffness matrix contained two global 'zero energy modes' (i.e., the global stiffness matrix was rank deficient by 2), thus producing incorrect nodal displacements whose results depend solely on the round-off errors in the calculations. These in turn produced incorrect stresses except at the 1×1 Gauss point used in each element to compute the stiffness and forces. Thus, based upon stability considerations, the use of 1×1 quadrature on four-noded elements produces a failure in the patch test. The element does satisfy consistency requirements, however, and provided a proper stabilization scheme is employed (e.g., stiffness or viscous methods are used in practice) this element may be used for practical calculations.[20,21]

It should be noted that a one-element patch test may be performed using the mesh shown in Fig. 10.6(*b*). The results are given by nodes 1 to 4 in Table 10.1. For the one-element patch, patch tests A and B coincide and neither evaluates the accuracy or stability of the stiffness matrix. On the other hand, patch test C leads to the conclusions reached using the five-element patch: namely, 2×2 gaussian quadrature passes a patch test whereas 1×1 quadrature fails the stability part of the test (as indeed we would expect by the arguments of Chapter 9, Sec. 9.11).

A simple test on cancellation of a diagonal during the triangular decomposition step is sufficient to warn of rank deficiencies in the stiffness matrix. In the profile method, described in Chapter 20, this is easily monitored as compact elimination converts the initial value of a diagonal element to the final value in one step. Thus only one extra scalar variable is needed to test the initial and final values.

10.7.2 Example 2: Patch test for quadratic elements: quadrature effects

In Fig. 10.7 we show a two-element patch of quadratic isoparametric quadrilaterals. Both eight-noded serendipity and nine-noded lagrangian types are considered and a basic patch test type C is performed for load case 1. For the eight-noded element both 2×2 ('reduced') and 3×3 ('standard') gaussian quadrature satisfy the patch test,

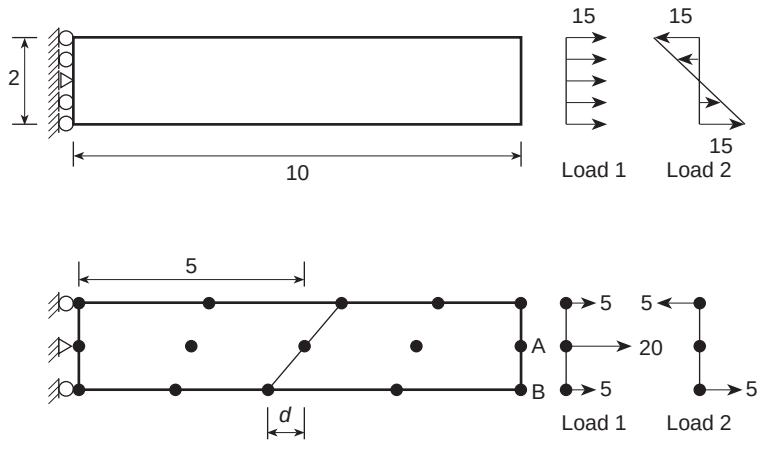


Fig. 10.7 Patch test for eight- and nine-noded isoparametric quadrilaterals.

whereas for the nine-noded element only 3×3 quadrature is satisfactory, with 2×2 reduced quadrature leading to failure in rank of the stiffness matrix. However, if we perform a one-element test for the eight-noded and 2×2 quadrature element, we discover the spurious zero-energy mode shown in Fig. 10.5 and thus the one-element test has failed. We consider such elements suspect and to be used only with the greatest of care. To illustrate what can happen in practice we consider the simple problem shown in Fig. 10.8(*a*). In this example the 'structure' modelled by a single element is considered rigid and interest is centred on the 'foundation' response. Accordingly only one element is used to model the structure. Use of 2×2 quadrature throughout leads to answers shown in Fig. 10.8(*b*) while results for 3×3 quadrature are shown in Fig. 10.8(*c*). It should be noted that no zero-energy mode exists since more than one element is used. There is, however, here a spurious response due to the large modulus variation between structure and foundation. This suggests that problems in which non-linear response may lead to a large variation in material parameters could also induce such performance, and thus use of the eight-noded 2×2 integrated element should always be closely monitored to detect such anomalous behaviour.

Indeed, support or loading conditions may themselves induce very suspect responses for elements in which near singularity occurs. Figure 10.9 shows some amusing peculiarities which can occur for reduced integration elements and which disappear entirely if full integration is used.[22] In all cases the *assembly* of elements is non-singular even though individual elements are rank deficient.

10.7.3 Example 3: Higher order patch test – assessment of order

In order to demonstrate a higher order patch test we consider the two-element plane stress problem shown in Fig. 10.7 and subjected to bending loading shown as Load 2. As above, two different types of element are considered: (*a*) an eight-noded serendipity quadrilateral elenent and (*b*) a nine-noded lagrangian quadrilateral element. In our

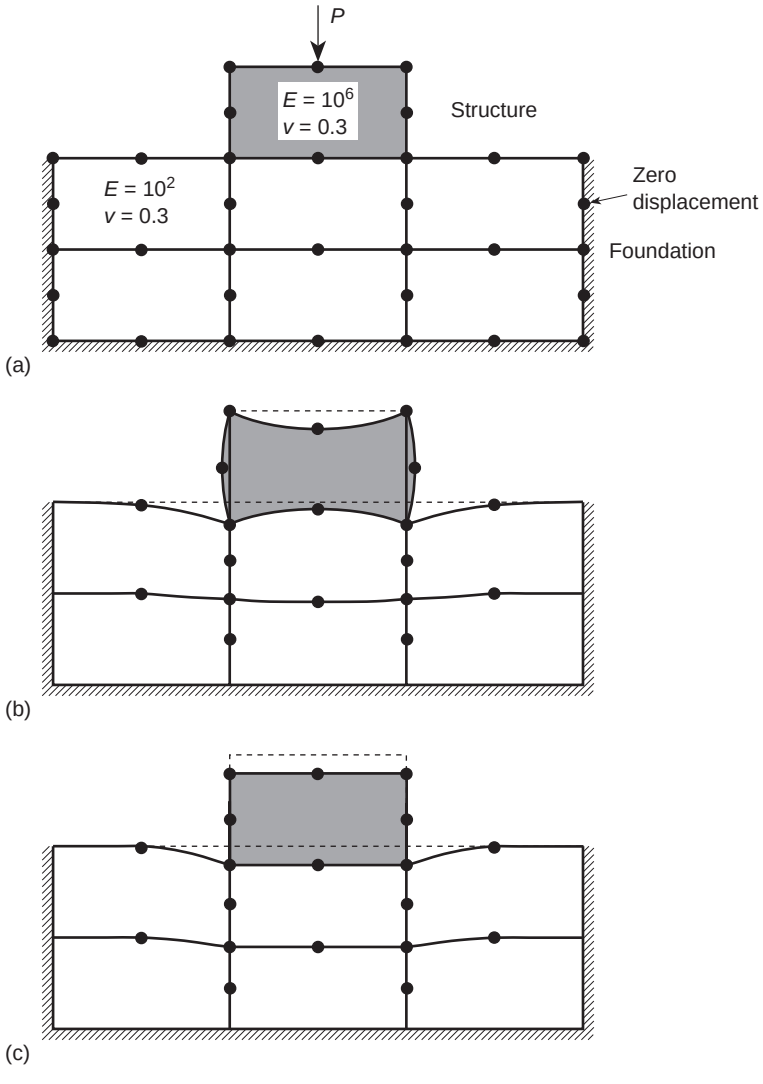


Fig. 10.8 A propagating spurious mode from a single unsatisfactory element. (a) Problem and mesh. (b) 2×2 integration. (c) 3×3 integration.

test we wish to demonstrate a feature for nine-noded element mapping discussed in Chapter 9 (see Sec. 9.7) and first shown by Wachspress.[23] In particular we restrict the mapping into the xy plane to be that produced by the four-noded isoparametric bilinear element, but permit the dependent variable to assume the full range of variations consistent with the eight- or nine-noded shape functions. In Chapter 9 we showed that the nine-noded element can approximate a complete quadratic displacement function in x, y whereas the eight-noded element cannot. Thus we expect that the nine-noded element when restricted to the isoparametric mappings of the four-noded element will pass a higher order patch test for all arbitrary quadratic displacement fields. The pure bending solution in elasticity is composed of polynomial terms

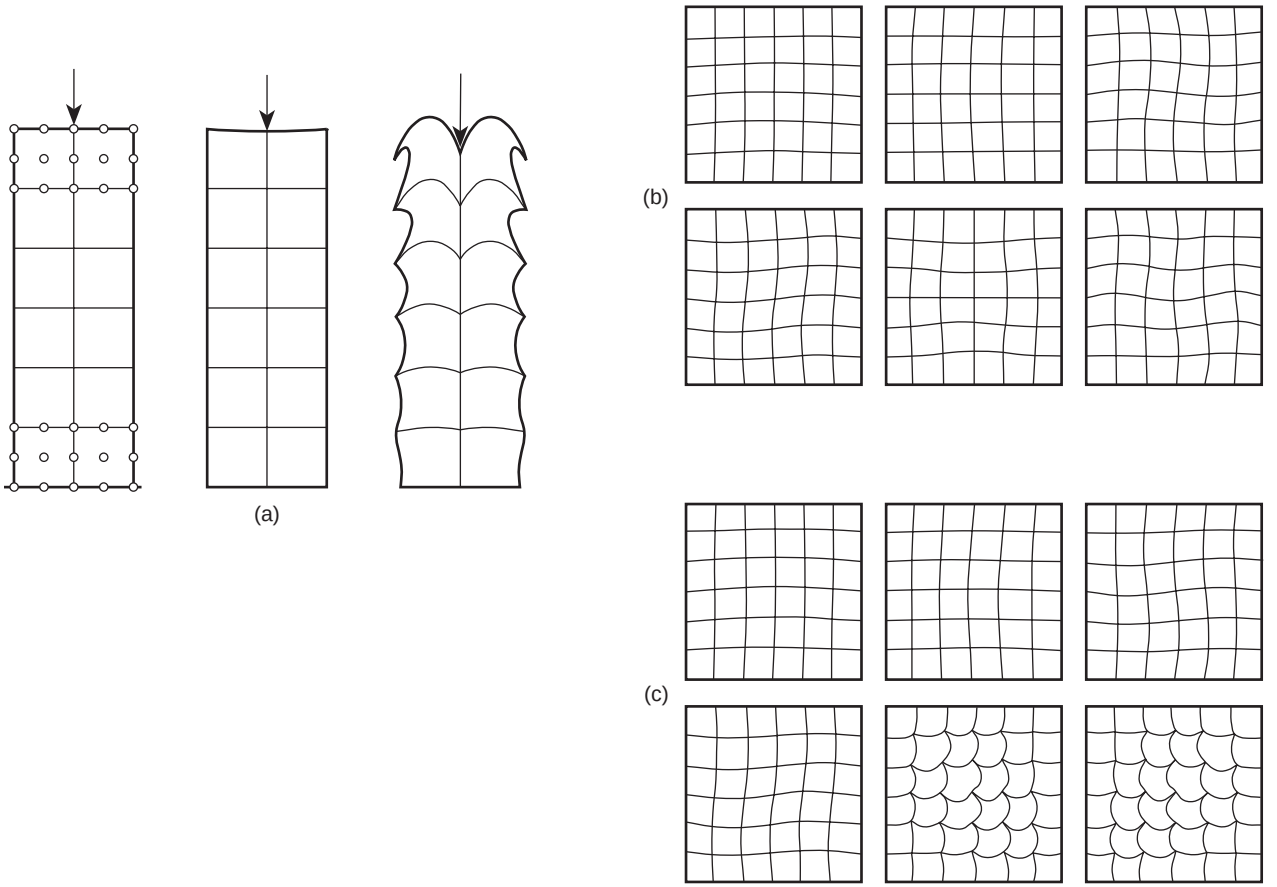


Fig. 10.9 Peculiar response of near singular assemblies of elements.[22] (a) A column of nine-noded elements with point load response of full 3×3 and 2×2 integration. The whole assembly is non-singular but singular element modes are apparent. (b) A fully constrained assembly of nine-noded elements with no singularity – first six eigenmodes with full (3×3) integration. (c) As (b) but with 2×2 integration. Note the appearance of 'wild' modes called 'Escher' modes named so in reference 22 after this graphic artist.

Table 10.2 Bending load case ($E = 100, \nu = 0.3$)

Element	Quadrature	d	v_A	u_B	v_B
Eight-node	3×3	0	0.750	0.150	0.75225
Eight-node	2×2		0.750	0.150	0.75225
Nine-node	3×3		0.750	0.150	0.75225
Eight-node	3×3	1	0.7448	0.1490	0.74572
Eight-node	2×2		0.750	0.150	0.75100
Nine-node	3×3		0.750	0.150	0.75225
Eight-node	3×3	2	0.6684	0.1333	0.66364
Eight-node	2×2		0.750	0.150	0.75225
Nine-node	3×3		0.750	0.150	0.75225
Exact	—	—	0.750	0.150	0.75225

up to quadratic order. Furthermore, no body force loadings are necessary to satisfy the equilibrium equations. For the mesh considered the nodal loadings are equal and opposite on the top and bottom nodes as shown in Fig. 10.7. The results for the two elements are shown in Table 10.2 for the indicated quadratures with $E = 100$ and $\nu = 0.3$.

From this test we observe that the nine-noded element does pass the higher order test performed. Indeed, provided the mapping is restricted to the four-noded shape it will always pass a patch test for displacements with terms no higher than quadratic. On the other hand, the eight-noded element passes the higher order patch test performed only for rectangular element (or constant jacobian) mappings. Moreover, the accuracy of the eight-noded element deteriorates very rapidly with increased distortions defined by the parameter d in Fig. 10.7.

The use of 2×2 reduced quadrature improves results for the higher order patch test performed. Indeed, two of the points sampled give exact results and the third is only slightly in error. As noted previously, however, a single element test for the 2×2 integrated eight-noded element will fail the stability part of the patch test and it should thus be used with great care.

10.8 Application of the patch test to an incompatible element

In order to demonstrate the use of the patch test for a finite element formulation which violates the usually stated requirements for shape function continuity, we consider the plane strain incompatible modes first introduced by Wilson *et al.*[24] and discussed by Taylor *et al.*[25] The specific incompatible formulation considered uses the element displacement approximations:

$$\hat{\mathbf{u}} = \mathbf{N}_i \mathbf{a}_i + N_1^n \boldsymbol{\alpha}_1 + N_2^n \boldsymbol{\alpha}_2 \tag{10.13}$$

where $\mathbf{N}_i$ ($i = 1, \ldots, 4$) are the usual conforming bilinear shape functions and the last two terms are *incompatible modes of deformation* defined by the hierarchical functions

$$N_1^n = 1 - \xi^2 \qquad \text{and} \qquad N_2^n = 1 - \eta^2 \tag{10.14}$$

defined independently for each element.

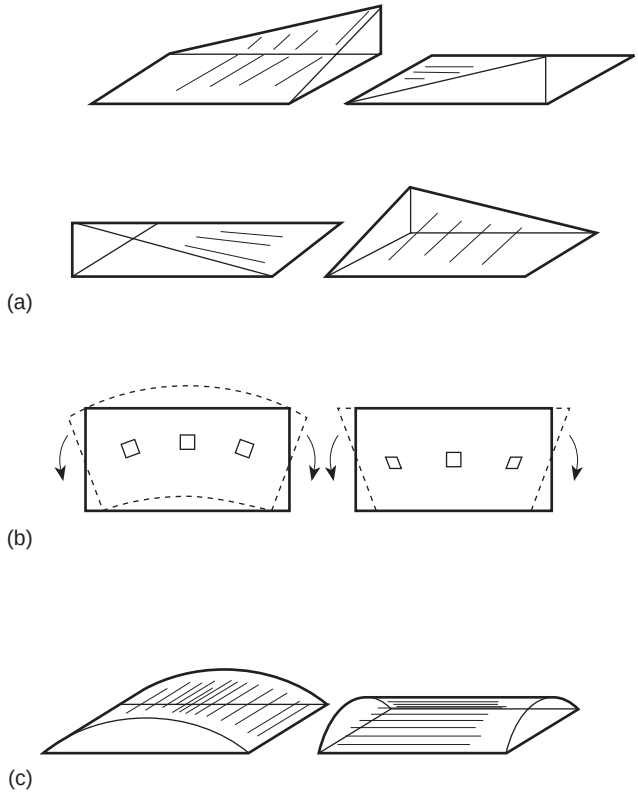


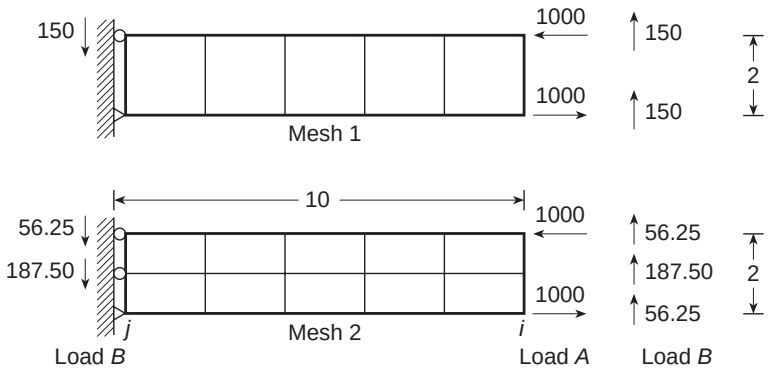
Fig. 10.10 (a) Linear quadrilateral with auxiliary incompatible shape functions; (b) pure bending and linear displacements causing shear; (c) auxiliary 'bending' shape functions with internal variables.

The shape functions used are illustrated in Fig. 10.10. The first, a set of standard bilinear type, gives a displacement pattern which, as shown in Fig. 10.10(*b*), introduces spurious shear strains in pure bending. The second, in which the parameters $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are strictly associated with a specific element, therefore introduces incompatibility but assures correct bending behaviour in an individual element. The excellent performance of this element in the bending situation is illustrated in Fig. 10.11.

In reference 25 the finite element approximation is computed by summing the potential energies of each element and computing the nodal loads due to boundary tractions from the conforming part of the displacement field only. Thus for the purposes of conducting patch tests we compute the strains using all parts of the displacement field leading to a generalization of (10.4) which may be written as

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \tag{10.15}$$

Here $\mathbf{K}_{11}$ and $\mathbf{f}_1$ are the stiffness and loads of the four-noded (conforming) bilinear element, $\mathbf{K}_{12}$ and $\mathbf{K}_{21}$ $(= \mathbf{K}_{12}^{\mathrm{T}})$ are coupling stiffnesses between the conforming and



		Displacement at *i*		Displacement at *j*	
		Load *A*	Load *B*	Load *A*	Load *B*
Beam theory		10.00	103.0	300.0	4050
(a)	Mesh 1	6.81	70.1	218.2	2945
	Mesh 2	7.06	72.3	218.8	2954
(b)	Mesh 1	10.00	101.5	300.0	4050
	Mesh 2	10.00	101.3	300.0	4050

Fig. 10.11 Performance of the non-conforming quadrilateral in beam bending treated as plane stress: (a) conforming linear quadrilateral, (b) non-conforming quadrilateral.

non-conforming displacements, and $\mathbf{K}_{22}$ and $\mathbf{f}_2$ are the stiffness and loads of the non-conforming displacements. We note that, according to the algorithm of reference 24, $\mathbf{f}_2$ must vanish from the patch test solutions.

For a patch test in plane strain or plane stress, only linear polynomials need be considered for which all non-conforming displacements must vanish. Thus for a successful patch test we must have

$$\mathbf{K}_{11}\mathbf{a} = \mathbf{f}_1 \tag{10.16a}$$

and

$$\mathbf{K}_{21}\mathbf{a} = \mathbf{f}_2 \tag{10.16b}$$

If we carry out a patch test for the mesh shown in Fig. 10.12(*a*) we find that all three forms (i.e., patch tests A, B, and C) satisfy these conditions and thus pass the patch test. If we consider the patch shown in Fig. 10.12(*b*), however, the patch test is not satisfied. The lack of satisfaction shows up in different ways for each form of the patch test. Patch test A produces non-zero $\mathbf{f}_2$ values when $\boldsymbol{\alpha}$ is set to zero and $\mathbf{a}$ according to the displacements considered. In form B the values of the nodal displacements

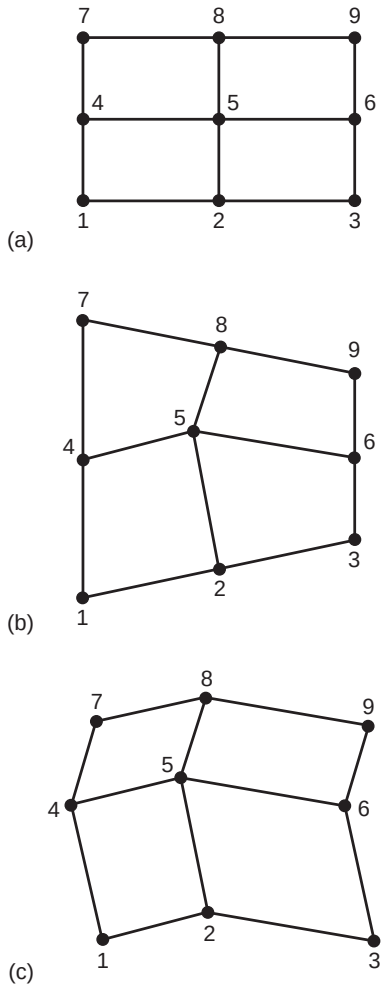


Fig. 10.12 Patch test for an incompatible element form. (a) Regular discretization. (b) Irregular discretization about node 5. (c) Constant jacobian discretization about node 5.

$\mathbf{a}_5$ are in error and $\boldsymbol{\alpha}$ are non-zero, also leading to erroneous stresses in each element. In form C all unspecified displacements are in error as well as the stresses.

It is interesting to note that when a patch is constructed according to Fig. 10.12(c) in which all elements are parallelograms all three forms of the patch test are once again satisfied. Accordingly we can note that if any mesh is systematically refined by subdivision of each element into four elements whose sides are all along ξ, η lines in the original element with values of -1, 0, or 1 (i.e., by bisections) the mesh converges to constant jacobian approximations of the type shown in Fig. 10.12(c). Thus, in this special case the incompatible mode element satisfies a weak patch test and will thus converge. In general, however, it may be necessary to use a very fine discretization to achieve sufficient accuracy, and hence the element probably has no practical (nor efficient) engineering use.

A simple artifice to ensure that an element passes the patch test is to replace the derivatives of the incompatible modes by

$$\begin{Bmatrix} \dfrac{\partial N_j^n}{\partial x} \\ \dfrac{\partial N_j^n}{\partial y} \end{Bmatrix} = \frac{j_0}{j(\xi,\eta)} \mathbf{J}_0^{-1} \begin{Bmatrix} \dfrac{\partial N_j^n}{\partial \xi} \\ \dfrac{\partial N_j^n}{\partial \eta} \end{Bmatrix} \tag{10.17}$$

where $j(\xi,\eta)$ is the determinant of the jacobian $J(\xi,\eta)$ and J_0 and j_0 are the values of the inverse jacobian and jacobian determinant evaluated at the element centre ($\xi = \eta = 0$). This ensures satisfaction of the patch test for all element shapes, and with this alteration of the algorithm the incompatible element proves convergent and quite accurate.[25]

10.9 Generation of incompatible shape functions which satisfy the patch test

In the previous section we have shown how an incompatible element can, on occasion, produce superior results despite its violation of the rules generally postulated. In solving plates and shells we deal with problems requiring C_1 continuity; the use of such incompatible functions is widespread not only because these produce superior results but also due to the difficulty of developing functions which satisfy not only the continuity of the functions but also their slope (viz. Volume 2, Chapter 4). In this section we address the problem of how to generate incompatible shape functions in a manner that will automatically ensure the satisfaction of the patch test and hence convergence. The rules for doing this have been developed[26,27] and applied to the derivation of plate bending elements. We derive these rules here in a simple example of a second-order partial differential equation problem but the results are easy to generalize to other situations.

Consider the finite element solution to the following equation:

$$A(u) \equiv -T\nabla^2 u + ku - q = 0 \qquad \text{in the domain } \Omega \tag{10.18}$$

with boundary conditions

$$u = \bar{u} \qquad \text{on } \Gamma_u$$

and (10.19)

$$T\frac{\partial u}{\partial n} = \bar{t} \qquad \text{on } \Gamma_t$$

This may represent the displacement u of an elastic membrane with an initial tension T on an elastic foundation with spring constant k. Let the unknown u be approximated by two sets of (hierarchic) expansions

$$u = u^{\mathrm{c}} + u^{\mathrm{n}} \tag{10.20a}$$

$$u^{\mathrm{c}} = \mathbf{N}^{\mathrm{c}}\mathbf{a}^{\mathrm{c}} \qquad \text{and} \qquad u^{\mathrm{n}} = \mathbf{N}^{\mathrm{n}}\boldsymbol{\alpha} \tag{10.20b}$$

in which $\mathbf{N}^{\mathrm{c}}$ and $\mathbf{N}^{\mathrm{n}}$ are, respectively, compatible and non-conforming shape functions. It must be stressed that these are linearly independent as otherwise stability conditions (i.e., the non-singularity of matrices) would be violated as was the case in the counterexample of Stummel.[9]

When a patch of elements is subject to a linear variation of u such that Eq. (10.18) is satisfied, the approximation u^{c} is capable of yielding this solution and satisfying all the patch test requirements. (Now, of course, $q = -ku$ has to be assumed.)

It follows therefore that in the patch test u^{n} will be zero. However, it is important to consider here a single element test in which the constant traction $\bar{t}$ (deduced from $u = u^{c}$) is applied. The Galerkin equation corresponding to the incompatible mode now yields

$$\int_{\Gamma_e} N_i^n T \frac{\partial u}{\partial n}\,\mathrm{d}\Gamma \equiv \int_{\Gamma_e} N_i^n T \left(n_x \frac{\partial u}{\partial x} + n_y \frac{\partial u}{\partial y} \right) \mathrm{d}\Gamma = \int_{\Gamma_{te}} N_i^n \bar{t}\,\mathrm{d}\Gamma \tag{10.21}$$

and this equation has to be satisfied identically with $\bar{t}$, T, and $\partial u/\partial n$ being constants. In the above Γ_e represents the total boundary of the element and n_x and n_y are components of the boundary normal vector (see Appendix G).

The above condition can be easily achieved by ensuring that

$$\int_{\Gamma_e} N_i^n n_x \frac{\partial u}{\partial x}\,\mathrm{d}\Gamma = \int_{\Gamma_e} N_i^n n_y \frac{\partial u}{\partial y}\,\mathrm{d}\Gamma = 0 \tag{10.22}$$

for each element, thus imposing the constraint

$$\int_{\Gamma_{te}} N_i^n \bar{t}\,\mathrm{d}\Gamma = 0 \tag{10.23}$$

which implies (as originally suggested by Wilson *et al.*[24]) that the effects of boundary loads (and loads q) from the incompatible displacements must vanish or be ignored.

In order to illustrate the use of the above procedure in developing incompatible mode shape functions, we consider the case of a non-conforming four-noded quadrilateral element which in the special case of a rectangle reproduces the non-conforming element of reference 24. The convergence of this non-conforming element for the rectangular or constant jacobian case has been illustrated in the previous section.

We take the conforming part of the shape function for each displacement component as the four-noded isoparametric functions

$$u^{c} = N_I a_I^{c} \tag{10.24}$$

where

$$N_I = \tfrac{1}{4}(1 + \xi_I \xi)(1 + \eta_I \eta) \tag{10.25}$$

and ξ, η are natural coordinates on the interval $(-1, 1)$ with values at each corner node I given by ξ_I, η_I. The non-conforming functions will be constructed from the remaining four shape functions for the eight-noded isoparametric serendipity element (Chapter 8). Accordingly we take for the non-conforming field

$$\begin{aligned} u^{n} = {} & \tfrac{1}{2}(1 - \xi^2)(1 - \eta)\alpha_1 + \tfrac{1}{2}(1 + \xi)(1 - \eta^2)\alpha_2 + \tfrac{1}{2}(1 - \xi^2)(1 + \eta)\alpha_3 \\ & + \tfrac{1}{2}(1 - \xi)(1 + \eta^2)\alpha_4 \end{aligned} \tag{10.26}$$

Substitution into the constraint conditions (10.23) yields the two scalar conditions

$$\sum_{i=1}^{4} b_i \alpha_i = 0 \tag{10.27}$$

and

$$\sum_{i=1}^{4} c_i \alpha_i = 0 \tag{10.28}$$

where b_i, c_i depend on the geometry of the element through

$$b_i = x_i - x_j$$

and

$$c_i = y_j - y_i \tag{10.29}$$

with

$$j = \text{mod}\,(i, 4) + 1$$

The two constraint conditions may be used to express two of the α_i in terms of the other two. The result gives two incompatible displacement modes which may be added to the conforming field with the satisfaction of a strong patch test still ensured. For elements which are rectangular the two resulting modes are identical to those proposed and used in Eq. (10.14).

Other possibilities exist for constructing non-conforming or incompatible functions.[5]

10.10 The weak patch test – example

The problems described above yield exact solutions for the patch tests performed and accordingly satisfy strong conditions. In order to illustrate the performance of an element which only satisfies a weak patch test we consider an axisymmetric linear elastic problem modelled by four-noded isoparametric elements. The material is assumed isotropic and the finite element stiffness and reaction force matrices are computed using a selective integration method where terms associated with the bulk modulus are evaluated by a single-point Gauss quadrature, whereas all other terms are computed using a 2×2 (normal) gaussian quadrature (such as will be discussed in Chapter 12). It may be readily verified that the stiffness matrix is of proper rank and thus stability of solutions is not an issue. On the other hand, consistency must still be evaluated.

In order to assess the performance of a selective reduced quadrature formulation we consider the patch of elements shown in Fig. 10.13. The patch is not as generally shaped as desirable and is only used to illustrate performance of an element that satisfies a weak patch test. The polynomial solution considered is

$$\begin{aligned} u &= 2r \\ w &= 0 \end{aligned} \tag{10.30}$$

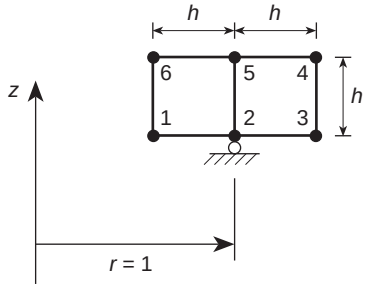


Fig. 10.13 Patch for selective, reduced quadrature on axisymmetric four-noded elements.

Table 10.3 Exact solution for patch

Node	Radius	Displacement		Force	
I	r_I	U_I	W_I	F_{rI}	F_{zI}
1, 4	$1-h$	$2(1-h)$	0	$-(1-h)h$	0
2, 5	1	2	0	0	0
3, 6	$1+h$	$2(1+h)$	0	$(1+h)h$	0

and material constants $E = 1$ and $\nu = 0$ are used in the analysis. The resulting stress field is given by

$$\sigma_r = \sigma_\theta = 2 \tag{10.31}$$

with other components identically zero. The exact solution for the nodal quantities of the mesh shown in Fig. 10.13 are summarized in Table 10.3. Patch tests have been performed for this problem using the selective reduced integration scheme described above and values of h of 0.8, 0.4, 0.2, 0.1, and 0.05. The result for the radial displacement at nodes 2 and 5 (reported to six digits) is given in Table 10.4. All other quantities (displacements, strains, and stresses) have a similar performance with convergence rates of at least $O(h)$ or more. Based on this assessment we conclude the element passes a weak patch test.

Table 10.4 Radial displacement at nodes 2 and 5

h	u
0.8	2.01114
0.4	2.00049
0.2	2.00003
0.1	2.00000
0.05	2.00000

10.11 Higher order patch test – assessment of robustness

A higher order patch test may also be used to assess element 'robustness'. An element is termed robust if its performance is not sensitive to physical parameters of the differential equation. For example, the performance of many elements for solution of plane strain linear elasticity problems is sensitive to Poisson's ratio values near 0.5 (called 'near incompressibility'). Indeed, for Poisson ratios near 0.5 the energy stored by a unit volumetric strain is many orders larger than the energy stored by a unit deviatoric strain. Accordingly finite elements which exhibit a strong coupling between volumetric and deviatoric strains often produce poor results in the nearly incompressible range, a problem discussed further in Chapter 12.

This may be observed using a four-noded element to solve a problem with a quadratic displacement field (i.e., a higher order patch test). If we again consider a

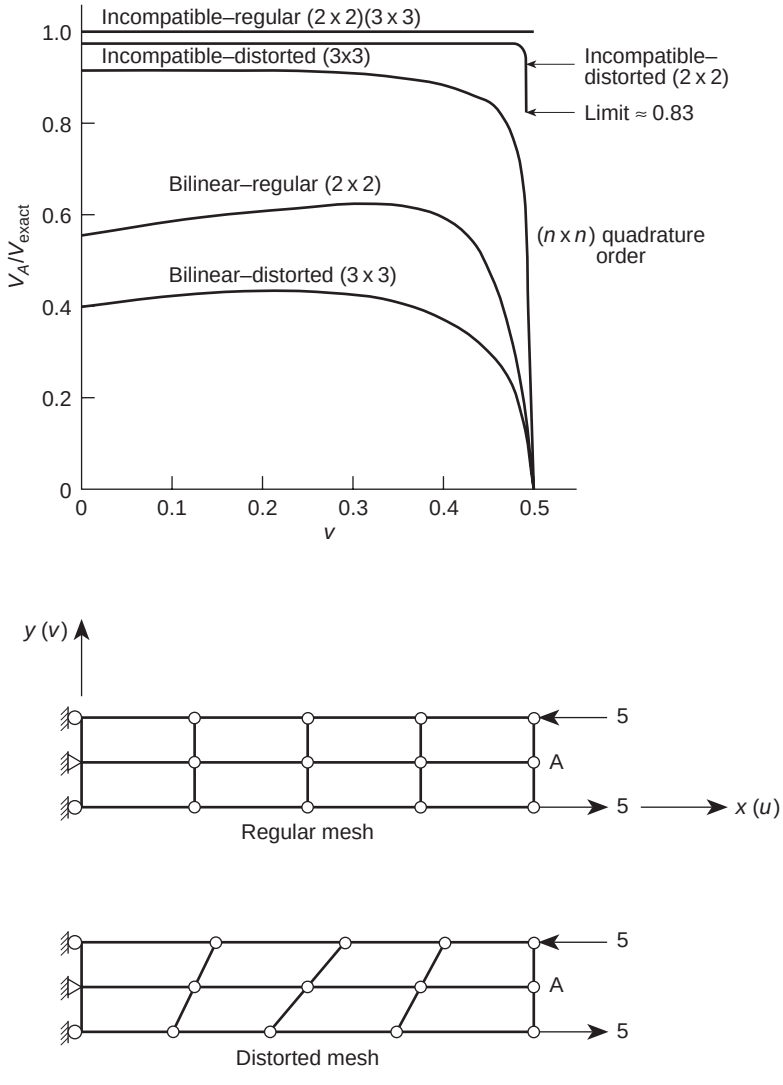


Fig. 10.14 Plane strain four-noded quadrilaterals with and without incompatible modes (higher order patch test for performance evaluation).

pure bending example and an eight-element mesh shown in Fig. 10.14 we can clearly observe the deterioration of results as Poisson's ratio approaches a value of one-half. Also shown in Fig. 10.14 are results for the incompatible modes derived in Sec. 10.9. It is evident that the response is considerably improved by adding these modes, especially if 2×2 quadrature is used.

If we consider the regular mesh and four-noded elements and further keep the domain constant and successively refine the problem using meshes of 8, 32, 128, and 512 elements, we observe that the answers do converge as guaranteed by the patch test. However, as shown in Fig. 10.15, the rate of convergence in energy for

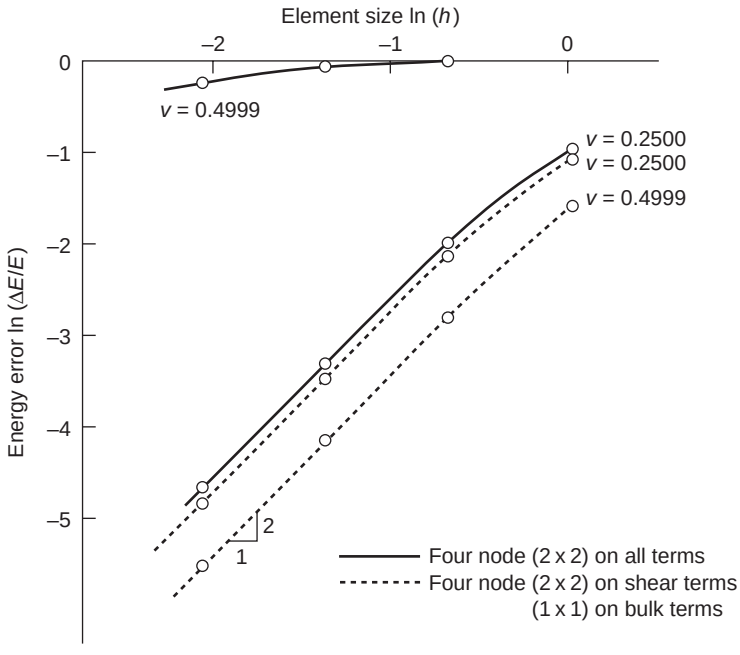


Fig. 10.15 Higher order patch test on element robustness (see Fig. 10.14) (convergence test under subdivision of elements).

Poisson ratio values of 0.25 and 0.4999 is quite different. For 0.25 the rate of convergence is nearly a straight line for all meshes, whereas for 0.4999 the rate starts out quite low and approaches an asymptotic value of 2 as h tends towards zero. For ν near 0.25 the element is called robust, whereas for ν near 0.5 it is not. If we use selective reduced integration (which for the plane strain case passes strong patch tests) and repeat the experiment, both values of ν produce a similar response and thus the element becomes robust for all values of Poisson's ratio less than 0.5.

The use of higher order patch tests can thus be very important to separate robust elements from non-robust elements. For methods which seek to automatically refine a mesh adaptively in regions with high errors, as discussed in Chapter 15, it is extremely important to use robust elements.

10.12 Conclusion

In the preceding sections we have described the patch test and its use in practice by considering several example problems. The patch test described has two essential parts: (*a*) *a consistency evaluation* and (*b*) *a stability check*. In the consistency test a set of linearly independent essential polynomials (i.e., all independent terms up to the order needed to describe the finite element model) is used as a solution to the differential equations and boundary conditions, and in the limit as the size of a patch tends to zero the finite element model must exactly satisfy each solution. We presented three forms to perform this portion of the test which we call forms A, B, and C.

The use of form C, where all boundary conditions are the natural ones (e.g., tractions for elasticity) except for the minimum number of essential conditions needed to ensure a unique solution to the problem (e.g., rigid body modes for elasticity), is recommended to test consistency and stability simultaneously. Both one-element and more-than-one-element tests are necessary to ensure that the patch test is satisfied. With these conditions and assuming that the solution procedure used can detect any possible rank deficiencies the stability of solution is also tested. If no such condition is included in the program a stability test must be conducted independently. This can be performed by computing the number of zero eigenvalues in the coefficient matrix for methods that use a solution of linear equations to compute the finite element parameters, **a**. Alternatively, the loading used for the patch solution may be perturbed at one point by a small value (say square root of the round-off limit, e.g., by 10^{-8} for round-offs of order 10^{-15}) and the solution tested to ensure that it does not change by a large amount.

Once an element has been shown to pass all of the essential patch tests for both consistency and stability, convergence is assured as the size of elements tends to zero. However, in some situations (e.g., the nearly incompressible elastic problem) convergence may be very slow until a very large number of elements is used. Accordingly, we recommend that higher order patch tests be used to establish element robustness. Higher order patch tests involve the use of polynomial solutions of the differential equation and boundary conditions with the order of terms larger than the basic polynomials used in a patch test. Indeed, the order of polynomials used should be increased until the patch test is satisfied only in a weak sense (i.e., as h tends to zero). The advantage of using a higher order patch test, as opposed to other boundary value problems, is that the exact solution may be easily computed everywhere in the model.

In some of the examples we have tested the use of incompatible function and inexact numerical integration procedures (reduced and selective integration). Some of these violations of the rules previously stipulated have proved justified not only by yielding improved performance but by providing methods for which convergence is guaranteed. We shall discuss in Chapter 12 some of the reasons for such improved performance.

References

1. B.M. Irons. Numerical integration applied to finite element methods. *Conf. on Use of Digital Computers in Structural Engineering*. Univ. of Newcastle, 1966.
2. G.P. Bazeley, Y.K. Cheung, B.M. Irons, and O.C. Zienkiewicz. Triangular elements in plate bending. Conforming and nonconforming solutions. *Proc. 1st Conf. on Matrix Methods in Structural Mechanics*. pp. 547–76, AFFDLTR-CC-80, Wright-Patterson AF Base, Ohio, 1966.
3. B.M. Irons and A. Razzaque. Experience with the patch test for convergence of finite element method, in *Mathematical Foundations of the Finite Element Method* (ed. A.K. Aziz). pp. 557–87, Academic Press, 1972.
4. B. Fraeijs de Veubeke. Variational principles and the patch test. *Int. J. Num. Meth. Eng.* **8**, 783–801, 1974.
5. G. Sander and P. Beckers. The influence of the choice of connectors in the finite element method. *Int. J. Num. Meth. Eng.* **11**, 1491–505, 1977.

6. E.R. de Arantes Oliveira. The patch test and the general convergence criteria of the finite element method. *Int. J. Solids Struct.* **13**, 159–78, 1977.
7. G. Strang. Variational crimes and the finite element method, in *Proc. Foundations of the Finite Element Method* (ed. A.K. Aziz). pp. 689–710, Academic Press, 1972.
8. G. Strang and G.J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.
9. F. Stummel. The limitations of the patch test. *Int. J. Num. Meth. Eng.* **15**, 177–88, 1980.
10. J. Robinson *et al.* Correspondence on patch test. *Finite Element News.* **1**, 30–4, 1982.
11. R.L. Taylor, O.C. Zienkiewicz, J.C. Simo, and A.H.C. Chan. The patch test – a condition for assessing f.e.m. convergence. *Int. J. Num. Meth. Eng.* **22**, 39–62, 1986.
12. R.E. Griffiths and A.R. Mitchell. Non-conforming elements, in *Mathematical Basis of Finite Element Methods.* Inst. Math. and Appl. Conference series, pp. 41–69, Clarendon Press, Oxford, 1984.
13. O.C. Zienkiewicz and R.L. Taylor. The finite element patch test revisited. A computer test for convergence, validation and error estimates. *Comp. Meth. Appl. Mech. Eng.* **149**, 223–54, 1997.
14. O.C. Zienkiewicz, S. Qu, R.L. Taylor, and S. Nakazawa. The patch test for mixed formulations. *Internat. J. Num. Meth. Eng.* **23**, 1873–83, 1986.
15. W.X. Zhong. Convergence of fem and the conditions of the patch test. Technical Report 97-3002, Research Institute Engineering Mechanics, Dalian University of Technology, 1997 (in Chinese).
16. I. Babuška, T. Strouboulis, and C.S. Upadhyay. A model study of the quality of *a posteriori* error estimators for linear elliptic problems. Error estimation in the interior of patchwise uniform grids of triangles. *Comp. Meth. Appl. Mech. Eng.* **114**, 307–78, 1994.
17. B. Boroomand and O.C. Zienkiewicz. An improved REP recovery and the effectivity robustness test. *Internat. J. Num. Meth. Eng.* **40**, 3247–77, 1997.
18. A. Ralston, *A First Course in Numerical Analysis.* McGraw-Hill, New York, 1965.
19. B.M. Irons and S. Ahmad. *Techniques of Finite Elements.* Horwood, Chichester, 1980.
20. D. Kosloff and G.A. Fraser. Treatment of hour glass patterns in low order finite element codes. *Int. J. Num. Anal. Meth. Geomechanics.* **2**, 57–72, 1978.
21. T. Belytchko and W.E. Bachrach. The efficient implementation of quadrilaterals with high coarse mesh accuracy. *Comp. Meth. Appl. Mech. Eng.* **54**, 276–301, 1986.
22. N. Bićanić and E. Hinton. Spurious modes in two dimensional isoparametric elements. *Int. J. Num. Meth. Eng.* **14**, 1545–57, 1979.
23. E.L. Wachspress. High-order curved finite elements. *Int. J. Num. Meth. Eng.* **17**, 735–45, 1981.
24. E.L. Wilson, R.L. Taylor, W.P. Doherty, and J. Ghaboussi. Incompatible displacement models, in *Num. and Comp. Meth. in Struct. Mech.* (eds S.T. Fenves *et al*,). pp. 43–57, Academic Press, 1973.
25. R.L. Taylor, P.J. Beresford, and E.L. Wilson. A non-conforming element for stress analysis. *Int. J. Num. Meth. Eng.* **10**, 1211–20, 1976.
26. A. Samuelsson. The global constant strain condition and the patch test. Chapter 3 of *Energy Methods in Finite Element Methods* (eds R. Glowinski, E.Y. Rodin, and O.C. Zienkiewicz). pp. 47–58, Wiley, 1979.
27. B. Specht. Modified shape functions for the three-node plate bending element passing the patch test. *Int. J. Num. Mech. Eng.* **26**, 705–15, 1988.

11

Mixed formulation and constraints – complete field methods

11.1 Introduction

The set of differential equations from which we start the discretization process will determine whether we refer to the formulation as *mixed* or *irreducible*. Thus if we consider an equation system with several dependent variables $\mathbf{u}$ written as [see Eqs (3.1) and (3.2)]

$$\mathbf{A}(\mathbf{u}) = \mathbf{0} \qquad \text{in domain } \Omega$$

and (11.1)

$$\mathbf{B}(\mathbf{u}) = \mathbf{0} \qquad \text{on boundary } \Gamma$$

in which none of the components of $\mathbf{u}$ can be eliminated still leaving a well-defined problem, then the formulation will be termed *irreducible*. If this is not the case the formulation will be called *mixed*. These definitions were given in Chapter 3 (p. 421).

This definition is not the only one possible[1] but appears to the authors to be widely applicable[2,3] if in the elimination process referred to we are allowed to introduce penalty functions. Further, for any given physical situation we shall find that more than one irreducible form is usually possible.

As an example we shall consider the simple problem of heat conduction (or the quasi-harmonic equation) to which we have referred in Chapters 3 and 7. In this we start with a physical constitutive relation defining the fluxes [see Eq. (7.5)] in terms of the potential (temperature) gradients, i.e.,

$$\mathbf{q} = -\mathbf{k}\,\boldsymbol{\nabla}\phi \qquad \mathbf{q} = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} \tag{11.2}$$

The continuity equation can be written as [see Eq. (7.7)]

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q} \equiv \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = -Q \tag{11.3}$$

If the above equations are satisfied in Ω and the boundary conditions

$$\phi = \bar{\phi} \text{ on } \Gamma_\phi \qquad \text{or} \qquad q_n = \bar{q}_n \text{ on } \Gamma_q \tag{11.4}$$

are obeyed then the problem is solved.

Clearly elimination of the vector $\mathbf{q}$ is possible and simple substitution of Eq. (11.2) into Eq. (11.3) leads to

$$-\boldsymbol{\nabla}^{\mathrm{T}}(\mathbf{k}\,\boldsymbol{\nabla}\phi) + Q = 0 \qquad \text{in } \Omega \tag{11.5}$$

with appropriate boundary conditions expressed in terms of ϕ or its gradient.

In Chapter 7 we showed discretized solutions starting from this point and clearly, as no further elimination of variables is possible, the formulation was *irreducible*.

On the other hand, if we start the discretization from Eqs (11.2)–(11.4) the formulation would be *mixed*.

An alternative irreducible form is also possible in terms of the variables $\mathbf{q}$. Here we have to introduce a penalty form and write in place of Eq. (11.3)

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q} + Q = \frac{\phi}{\alpha} \tag{11.6}$$

where α is a penalty number which tends to infinity. Clearly in the limit both equations are the same and in general if α is very large but finite the solutions should be approximately the same.

Now substitution into Eq. (11.2) gives the single governing equation

$$\boldsymbol{\nabla}\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q} + \frac{1}{\alpha}\,\mathbf{k}^{-1}\mathbf{q} + \boldsymbol{\nabla}Q = \mathbf{0} \tag{11.7}$$

which again could be used for the start of a discretization process as a possible irreducible form.[4]

The reader should observe that, by the definition given, the formulations so far used in this book were *irreducible*. In subsequent sections we will show how elasticity problems can be dealt with in *mixed* form and indeed will show how such formulations are essential in certain problems typified by the incompressible elasticity example to which we have referred in Chapter 4. In Chapter 3 (Sec. 3.8.2) we have shown how discretization of a mixed problem can be accomplished.

Before proceeding to a discussion of such discretization (which will reveal the advantages and disadvantages of mixed methods) it is important to observe that if the operator specifying the mixed form is *symmetric* or *self-adjoint* (see Sec. 3.9.1) the formulation can proceed from the basis of a *variational principle* which can be directly obtained for linear problems. We invite the reader to prove by using the methods of Chapter 3 that stationarity of the *variational principle* given below is equivalent to the differential equations (11.2) and (11.3) together with the boundary conditions (11.4):

$$\Pi = \frac{1}{2}\int_{\Omega} \mathbf{q}^{\mathrm{T}}\mathbf{k}^{-1}\mathbf{q}\,\mathrm{d}\Omega + \int_{\Omega} \mathbf{q}^{\mathrm{T}}\boldsymbol{\nabla}\phi\,\mathrm{d}\Omega + \int_{\Omega} \phi Q\,\mathrm{d}\Omega - \int_{\Gamma_q} \phi\bar{q}_n\,\mathrm{d}\Gamma \tag{11.8}$$

for

$$\phi = \bar{\phi} \qquad \text{on} \qquad \Gamma_{\phi}$$

The establishment of such variational principles is a worthy academic pursuit and had led to many famous forms given in the classical work of Washizu.[5] However, we also know (see Sec. 3.7) that if symmetry of weighted residual matrices is obtained in a linear problem then a variational principle exists and can be determined. As such symmetry can be established by inspection we shall, in what follows, proceed with such weighting directly and thus avoid some unwarranted complexity.

11.2 Discretization of mixed forms – some general remarks

We shall demonstrate the discretization process on the basis of the mixed form of the heat conduction equations (11.2) and (11.3). Here we start by assuming that each of the unknowns is approximated in the usual manner by appropriate shape functions and corresponding unknown parameters. Thus†

$$\mathbf{q} \approx \hat{\mathbf{q}} = \mathbf{N}_q \tilde{\mathbf{q}} \qquad \text{and} \qquad \phi \approx \hat{\phi} = \mathbf{N}_\phi \tilde{\boldsymbol{\phi}} \tag{11.9}$$

where $\tilde{\mathbf{q}}$ and $\tilde{\boldsymbol{\phi}}$ stand for nodal or element parameters that have to be determined. Similarly the weighting functions are given by

$$\mathbf{v}_q \approx \hat{\mathbf{v}}_q = \mathbf{W}_q \delta\tilde{\mathbf{q}} \qquad \text{and} \qquad v_\phi \approx \hat{v}_\phi = \mathbf{W}_\phi \delta\tilde{\boldsymbol{\phi}} \tag{11.10}$$

where $\delta\tilde{\mathbf{q}}$ and $\delta\tilde{\boldsymbol{\phi}}$ are arbitrary parameters.

Assuming that the boundary conditions for $\phi = \bar{\phi}$ are satisfied by the choice of the expansion, the weighted statement of the problem is, for Eq. (11.2) after elimination of the arbitrary parameters,

$$\int_\Omega \mathbf{W}_q^{\mathrm{T}} (\mathbf{k}^{-1}\hat{\mathbf{q}} + \boldsymbol{\nabla}\hat{\phi})\, \mathrm{d}\Omega = \mathbf{0} \tag{11.11}$$

and, for Eq. (11.3) and the 'natural' boundary conditions,

$$-\int_\Omega \mathbf{W}_\phi^{\mathrm{T}} (\nabla^{\mathrm{T}}\hat{\mathbf{q}} + Q)\, \mathrm{d}\Omega + \int_{\Gamma_q} \mathbf{W}_\phi^{\mathrm{T}} (\hat{q}_n - \bar{q}_n)\, \mathrm{d}\Gamma = \mathbf{0} \tag{11.12}$$

The reason we have premultiplied Eq. (11.2) by $\mathbf{k}^{-1}$ is now evident as the choice

$$\mathbf{W}_q = \mathbf{N}_q \qquad \mathbf{W}_\phi = \mathbf{N}_\phi \tag{11.13}$$

will yield symmetric equations [using Green's theorem to perform integration by parts on the gradient term in Eq. (11.12)] of the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{q}} \\ \tilde{\boldsymbol{\phi}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \tag{11.14}$$

with

$$\begin{aligned} \mathbf{A} &= \int_\Omega \mathbf{N}_q^{\mathrm{T}} \mathbf{k}^{-1} \mathbf{N}_q \, \mathrm{d}\Omega \\ \mathbf{C} &= \int_\Omega \mathbf{N}_q^{\mathrm{T}} \boldsymbol{\nabla} \mathbf{N}_\phi \, \mathrm{d}\Omega \\ \mathbf{f}_1 &= \mathbf{0} \\ \mathbf{f}_2 &= -\int_\Omega \mathbf{N}_\phi^{\mathrm{T}} Q \, \mathrm{d}\Omega - \int_{\Gamma_q} \mathbf{N}_\phi^{\mathrm{T}} \bar{q} \, \mathrm{d}\Gamma \end{aligned} \tag{11.15}$$

† The reader will note that we have now changed the notation slightly, having previously used a different symbol such as $\mathbf{a}_i$ for nodal quantities. We do this because now more than one variable occurs and it is convenient to denote this variable with a similarly denoted nodal parameter.

This problem, which we shall consider as typifying a large number of mixed approximations, illustrates the main features of the mixed formulation, including its advantages and disadvantages. We note that

1. The continuity requirements on the shape functions chosen are different. It is easily seen that those given for $\mathbf{N}_\phi$ can be C_0 continuous while those for $\mathbf{N}_q$ can be discontinuous in or between elements (C_{-1} continuity) as no derivatives of this are present. Alternatively, this discontinuity can be transferred to $\mathbf{N}_\phi$ (using Green's theorem on the integral in $\mathbf{C}$) while maintaining C_0 continuity for $\mathbf{N}_q$.

   This relaxation of continuity is of particular importance in plate and shell bending problems (see Volume 2) and indeed many important early uses of mixed forms have been made in that context.[6–9]
2. If interest is focused on the variable $\mathbf{q}$ rather than ϕ, use of an improved approximation for this may result in higher accuracy than possible with the irreducible form previously discussed. *However, we must note that if the approximation function for* $\mathbf{q}$ *is capable of reproducing precisely the same type of variation as that determinable from the irreducible form then no additional accuracy will result and, indeed, the two approximations will yield identical answers.*

   Thus, for instance, if we consider the mixed approximation to the field problems discussed using a linear triangle to determine $\mathbf{N}_\phi$ and piecewise constant $\mathbf{N}_q$, as shown in Fig. 11.1, we will obtain precisely the same results as those obtained by the irreducible formulation with the same $\mathbf{N}_\phi$ applied directly to Eq. (11.5), *providing* $\mathbf{k}$ *is constant within each element*. This is evident as the second of Eqs (11.14) is precisely the weighted continuity statement used in deriving the irreducible formulation in which the first of the equations is identically satisfied.

   Indeed, should we choose to use a linear but discontinuous approximation form of $\mathbf{N}_q$ in the interior of such a triangle, we would still obtain precisely the same answers, with the additional coefficients becoming zero. This discovery was made

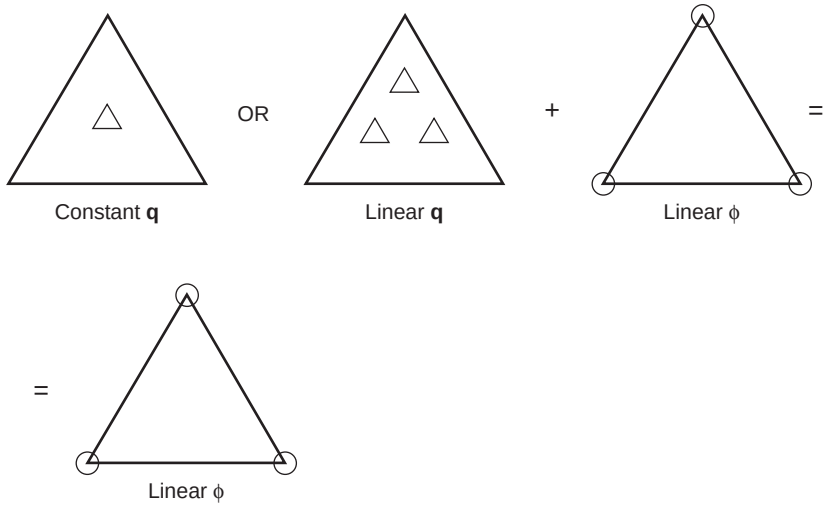


Fig. 11.1 A mixed approximation to the heat conduction problem yielding identical results as the corresponding irreducible form (the constant $\mathbf{k}$ is assumed in each element).

by Fraeijs de Veubeke[10] and is called the *principle of limitation*, showing that under some circumstances no additional accuracy is to be expected from a mixed formulation. In a more general case where $\mathbf{k}$ is, for instance, discontinuous and variable within an element, the results of the mixed approximation will be different and on occasion superior.[2] Note that a C_0-continuous approximation for $\mathbf{q}$ does not fall into this category as it is not capable of reproducing the discontinuous ones.

3. The equations resulting from mixed formulations frequently have zero diagonal terms as indeed in the case of Eq. (11.14).

   We noted in Chapter 3 that this is a characteristic of problems constrained by a Lagrange multiplier variable. Indeed, this is the origin of the problem, which adds some difficulty to a standard gaussian elimination process used in equation solving (see Chapter 20). As the form of Eq. (11.14) is typical of many two-field problems we shall refer to the first variable (here $\tilde{\mathbf{q}}$) as the *primary variable* and the second (here $\tilde{\boldsymbol{\phi}}$) as the *constraint variable*.
4. The added number of variables means that generally larger size algebraic problems have to be dealt with. However, in Sec. 11.6 we shall show how such difficulties can often be avoided by a suitable iterative solution.

The characteristics so far discussed did not mention one vital point which we elaborate in the next section.

11.3 Stability of mixed approximation. The patch test

11.3.1 Solvability requirement

Despite the relaxation of shape function continuity requirements in the mixed approximation, for certain choices of the individual shape functions the mixed approximation will not yield meaningful results. This limitation is indeed much more severe than in an *irreducible* formulation where a very simple 'constant gradient' (or constant strain) condition sufficed to ensure a convergent form once continuity requirements were satisfied.

The mathematical reasons for this difficulty are discussed by Babuška[11] and Brezzi,[12] who formulated a mathematical criterion associated with their names. However, some sources of the difficulties (and hence ways of avoiding them) follow from quite simple reasoning.

If we consider the equation system (11.14) to be typical of many mixed systems in which $\tilde{\mathbf{q}}$ is the *primary variable* and $\tilde{\boldsymbol{\phi}}$ is the *constraint variable* (equivalent to a lagrangian multiplier), we note that the solution can proceed by eliminating $\tilde{\mathbf{q}}$ from the first equation and by substituting into the second to obtain

$$(\mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{C})\tilde{\boldsymbol{\phi}} = -\mathbf{f}_2 + \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{f}_1 \tag{11.16}$$

which requires the matrix $\mathbf{A}$ to be non-singular (or $\mathbf{A}\tilde{\mathbf{q}} \neq \mathbf{0}$ for all $\tilde{\mathbf{q}} \neq \mathbf{0}$). To calculate $\tilde{\boldsymbol{\phi}}$ it is necessary to ensure that the bracketed matrix, i.e.

$$\mathbf{H} = \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{C} \tag{11.17}$$

is non-singular.

Singularity of the $\mathbf{H}$ matrix will always occur if the number of unknowns in the vector $\tilde{\mathbf{q}}$, which we call n_q, is less than the number of unknowns n_ϕ in the vector $\tilde{\boldsymbol{\phi}}$. Thus for avoidance of singularity

$$n_q \geqslant n_\phi \tag{11.18}$$

is *necessary* though not *sufficient* as we shall find later.

The reason for this is evident as the rank of the matrix (11.17), which needs to be n_ϕ, cannot be greater than n_q, i.e., the rank of $\mathbf{A}^{-1}$.

In some problems the matrix $\mathbf{A}$ may well be singular. It can normally be made non-singular by addition of a multiple of the second equation, thus changing the first equation to

$$\bar{\mathbf{A}} = \mathbf{A} + \gamma \mathbf{C}\mathbf{C}^{\mathrm{T}}$$

$$\bar{\mathbf{f}}_1 = \mathbf{f}_1 + \gamma \mathbf{C}\mathbf{f}_2$$

where γ is an arbitrary number.

Although both the matrices $\mathbf{A}$ and $\mathbf{C}\mathbf{C}^{\mathrm{T}}$ are singular their combination $\bar{\mathbf{A}}$ should not be, providing we ensure that for all vectors $\tilde{\mathbf{q}} \neq \mathbf{0}$ either

$$\mathbf{A}\tilde{\mathbf{q}} \neq \mathbf{0} \qquad \text{or} \qquad \mathbf{C}^{\mathrm{T}}\tilde{\mathbf{q}} \neq \mathbf{0}$$

In mathematical terminology this means that $\mathbf{A}$ is non-singular in the null space of $\mathbf{C}\mathbf{C}^{\mathrm{T}}$.

The requirement of Eq. (11.18) is a necessary but not sufficient condition for non-singularity of the matrix $\mathbf{H}$. An additional requirement evident from Eq. (11.16) is

$$\mathbf{C}\tilde{\boldsymbol{\phi}} \neq \mathbf{0} \qquad \text{for all} \qquad \tilde{\boldsymbol{\phi}} \neq \mathbf{0}$$

If this is not the case the solution would not be unique.

The above requirements are inherent in the Babuška–Brezzi condition previously mentioned, but can always be verified algebraically.

11.3.2 Locking

The condition (11.18) ensures that non-zero answers for the variables $\tilde{\mathbf{q}}$ are possible. If it is violated *locking* or non-convergent results will occur in the formulation, giving near-zero answers for $\tilde{\mathbf{q}}$ [see Chapter 3, Eq. (3.159) ff.].

To show this, we shall replace Eq. (11.14) by its penalized form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & -\dfrac{1}{\alpha}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{q}} \\ \tilde{\boldsymbol{\phi}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \qquad \begin{matrix} \text{with } \alpha \to \infty \\ \text{and } \mathbf{I} = \text{identity matrix} \end{matrix} \tag{11.19}$$

Elimination of $\tilde{\boldsymbol{\phi}}$ leads to

$$(\mathbf{A} + \alpha \mathbf{C}\mathbf{C}^{\mathrm{T}})\tilde{\mathbf{q}} = \mathbf{f}_1 + \alpha \mathbf{C}\mathbf{f}_2 \tag{11.20}$$

As $\alpha \to \infty$ the above becomes simply

$$(\mathbf{C}\mathbf{C}^{\mathrm{T}})\tilde{\mathbf{q}} = \mathbf{C}\mathbf{f}_2 \tag{11.21}$$

Non-zero answers for $\tilde{\mathbf{q}}$ should exist even when $\mathbf{f}_2$ is zero and hence the matrix $\mathbf{C}\mathbf{C}^{\mathrm{T}}$ *must be singular*. This singularity will always exist if $n_q > n_\phi$.

The stability conditions derived on the particular example of Eq. (11.14) are generally valid for any problem exhibiting the standard Lagrange multiplier form. In particular the necessary count condition will in many cases suffice to determine element acceptability; however, final conclusions for successful elements which pass all count conditions must be evaluated by rank tests on the full matrix.

In the example just quoted $\tilde{\mathbf{q}}$ denote fluxes and $\tilde{\boldsymbol{\phi}}$ temperatures and perhaps the concept of locking was not clearly demonstrated. It is much more definite where the first primary variable is a displacement and the second constraining one is a stress or a pressure. There locking is more evident physically and simply means an occurrence of zero displacements throughout as the solution approaches a limit. This unfortunately will happen on occasion.

11.3.3 The patch test

The patch test for mixed elements can be carried out in exactly the way we have described in the previous chapter for irreducible elements. As *consistency* is easily assured by taking a polynomial approximation for each of the variables, only *stability* needs generally to be investigated. *Most answers* to this can be obtained by simply ensuring that *count condition* (11.18) is satisfied for any isolated patch on the boundaries of which we constrain the *maximum* number of primary variables and the *minimum* number of constraint variables.[13]

In Fig. 11.2 we illustrate a single element test for two possible formulations with C_0 continuous N_ϕ (quadratic) and discontinuous N_q, assumed to be either constant or linear within an element of triangular form. As no values of $\tilde{\mathbf{q}}$ can here be specified on the boundaries, we shall fix a single value of $\tilde{\boldsymbol{\phi}}$ only, as is necessary to ensure

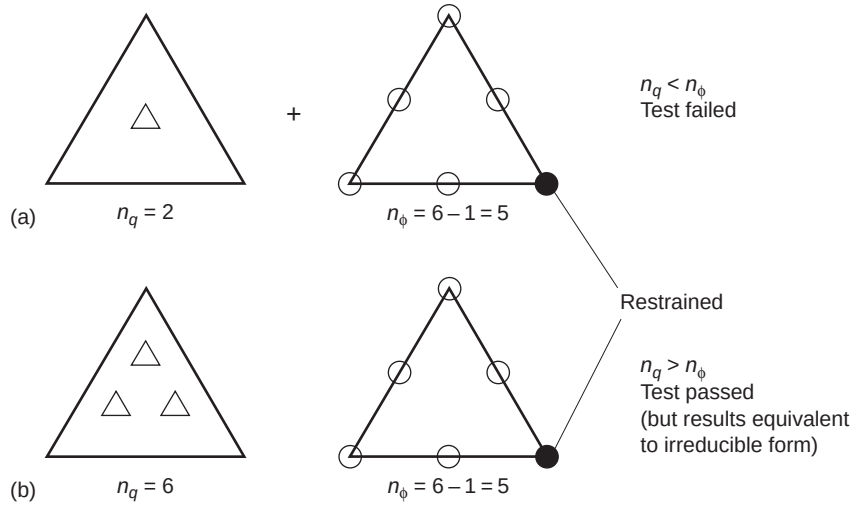


Fig. 11.2 Single element patch test for mixed approximations to the heat conduction problem with discontinuous flux **q** assumed: (a) quadratic C_0, ϕ; constant **q**; (b) quadratic C_0, ϕ; linear **q**.

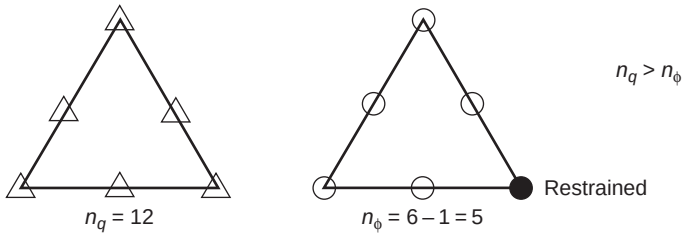


Fig. 11.3 As Fig. 11.2 but with C_0 continuous **q**.

uniqueness, on the patch boundary, which is here simply that of a single element. A count shows that only one of the formulations, i.e., that with linear flux variation, satisfies condition (11.18) and therefore may be acceptable.

In Fig. 11.3 we illustrate a similar patch test on the same element but with identical C_0 continuous shape functions specified for both $\tilde{\mathbf{q}}$ and $\tilde{\boldsymbol{\phi}}$ variables. This example shows satisfaction of the basic condition of Eq. (11.18) and therefore is apparently a permissible formulation. The permissible formulation must always be subjected to a numerical rank test. Clearly condition (11.18) will need to be satisfied and many useful conclusions can be drawn from such counts. These eliminate elements which will not function and on many occasions will give guidance to elements which will.

Even if the patch test is satisfied occasional difficulties can arise, and these are indicated mathematically by the Babuška–Brezzi condition already referred to.[14] These difficulties can be due to *excessive continuity* imposed on the problem by requiring, for instance, the flux condition to be of C_0 continuity class. In Fig. 11.4 we illustrate some cases in which the imposition of such continuity is *physically incorrect* and therefore can be expected to produce erroneous (and usually highly oscillating) results. In all such problems we recommend that the continuity be relaxed at least locally.

We shall discuss this problem further in Sec. 11.4.3.

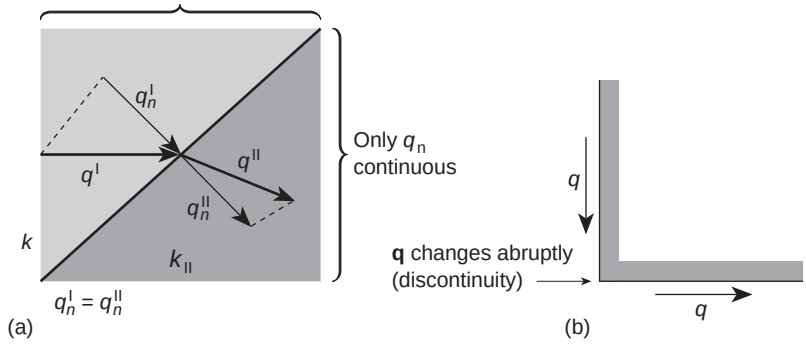


Fig. 11.4 Some situations for which C_0 continuity of flux **q** is inappropriate: (a) discontinuous change of material properties; (b) singularity.

11.4 Two-field mixed formulation in elasticity

11.4.1 General

In all the previous formulations of elasticity problems in this book we have used an irreducible formulation, using the displacement $\mathbf{u}$ as the primary variable. The virtual work principle was used to establish the equilibrium conditions which were written as (see Chapter 2)

$$\int_{\Omega} \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_{\Omega} \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \tag{11.22}$$

where $\bar{\mathbf{t}}$ are the tractions prescribed on Γ_t and with

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{11.23}$$

as the constitutive relation (omitting here initial strains and stresses for clarity).

We recall that statements such as Eq. (11.22) are equivalent to weighted residual forms (see Chapter 3) and in what follows we shall use these frequently. In the above the strains are related to displacement by the matrix operator $\mathbf{S}$ introduced in Chapter 2, giving

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u} \tag{11.24}$$

$$\delta\boldsymbol{\varepsilon} = \mathbf{S}\,\delta\mathbf{u} \tag{11.25}$$

with the displacement expansions constrained to satisfy the prescribed displacements on Γ_u. This is, of course, equivalent to Galerkin-type weighting.

With the displacement $\mathbf{u}$ approximated as

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_u\tilde{\mathbf{u}} \tag{11.26}$$

the required stiffness equations were obtained in terms of the unknown displacement vector $\tilde{\mathbf{u}}$ and the solution obtained.

It is possible to use mixed forms in which either $\boldsymbol{\sigma}$ or $\boldsymbol{\varepsilon}$, or, indeed, both these variables, are approximated independently. We shall discuss such formulations below.

11.4.2 The u–σ mixed form

In this we shall assume that Eq. (11.22) is valid but that we approximate $\boldsymbol{\sigma}$ independently as

$$\boldsymbol{\sigma} \approx \hat{\boldsymbol{\sigma}} = \mathbf{N}_\sigma\tilde{\boldsymbol{\sigma}} \tag{11.27}$$

and approximately satisfy the constitutive relation

$$\boldsymbol{\sigma} = \mathbf{D}\mathbf{S}\mathbf{u} \tag{11.28}$$

which replaces (11.23) and (11.24). The approximate integral form is written as

$$\int_{\Omega} \delta\boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{S}\mathbf{u} - \mathbf{D}^{-1}\boldsymbol{\sigma})\,\mathrm{d}\Omega = 0 \tag{11.29}$$

where the expression in the brackets is simply Eq. (11.28) premultiplied by $\mathbf{D}^{-1}$ to establish symmetry and $\delta\boldsymbol{\sigma}$ is introduced as a weighting variable.

Indeed, Eqs (11.22) and (11.29) which now define the problem are equivalent to the stationarity of the functional

$$\Pi_{\mathrm{HR}} = \int_\Omega \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{S}\mathbf{u}\,\mathrm{d}\Omega - \frac{1}{2}\int_\Omega \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{D}^{-1}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{11.30}$$

where the boundary displacement

$$\mathbf{u} = \bar{\mathbf{u}}$$

is enforced on Γ_u, as the reader can readily verify. This is the well-known Hellinger–Reissner[15,16] variational principle, but, as we have remarked earlier, it is unnecessary in deriving approximate equations. Using

$$\begin{aligned} \mathbf{N}_u\delta\tilde{\mathbf{u}} \quad &\text{in place of} \quad \delta\mathbf{u} \\ \mathbf{B}\delta\tilde{\mathbf{u}} \equiv \mathbf{S}\mathbf{N}_u\delta\tilde{\mathbf{u}} \quad &\text{in place of} \quad \delta\boldsymbol{\varepsilon} \\ \mathbf{N}_\sigma\delta\tilde{\boldsymbol{\sigma}} \quad &\text{in place of} \quad \delta\boldsymbol{\sigma} \end{aligned}$$

we write the approximate equations (11.29) and (11.22) in the standard form [see Eq. (11.14)]

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}} \\ \tilde{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \tag{11.31}$$

with

$$\begin{aligned} \mathbf{A} &= -\int_\Omega \mathbf{N}_\sigma^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{N}_\sigma\,\mathrm{d}\Omega \\ \mathbf{C} &= +\int_\Omega \mathbf{N}_\sigma^{\mathrm{T}}\mathbf{B}\,\mathrm{d}\Omega \\ \mathbf{f}_1 &= \mathbf{0} \\ \mathbf{f}_2 &= +\int_\Omega \mathbf{N}_u^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_t} \mathbf{N}_u^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \end{aligned} \tag{11.32}$$

In the form given above the $\mathbf{N}_u$ shape functions have still to be of C_0 continuity, though $\mathbf{N}_\sigma$ can be discontinuous. However, integration by parts of the expression for $\mathbf{C}$ allows a reduction of such continuity and indeed this form has been used by Herrmann[6,17,18] for problems of plates and shells.

11.4.3 Stability of two-field approximation in elasticity (u–σ)

Before attempting to formulate practical mixed approach approximations in detail, identical stability problems to those discussed in Sec. 11.3 have to be considered.

For the **u**–**σ** forms it is clear that **σ** is the *primary variable* and **u** the *constraint variable* (see Sec. 11.2), and for the total problem as well as for element patches we must have as a necessary, though not sufficient condition

$$n_\sigma \geqslant n_u \tag{11.33}$$

where n_σ and n_u stand for numbers of degrees of freedom in appropriate variables.

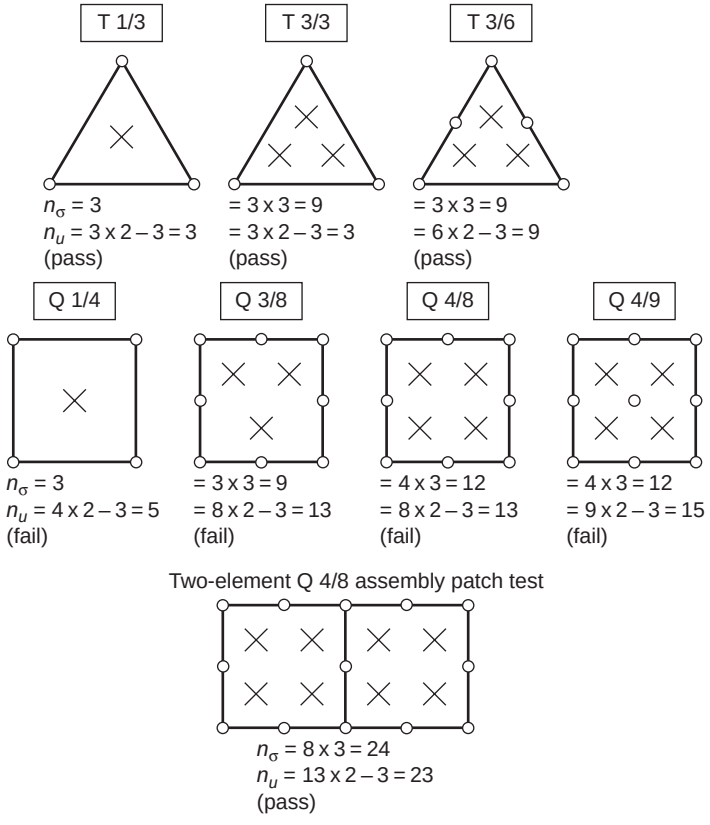


Fig. 11.5 Elasticity by the mixed $\boldsymbol{\sigma}$–$\mathbf{u}$ formulation. Discontinuous stress approximation. Single element patch test. No restraint on $\tilde{\boldsymbol{\sigma}}$ variables but three $\tilde{\mathbf{u}}$ degrees of freedom restrained on patch. Test condition $n_\sigma \geqslant n_u$ (X denotes $\tilde{\boldsymbol{\sigma}}$ (3 DOF) and ○ the $\tilde{\mathbf{u}}$ (2 DOF) variables).

In Fig. 11.5 we consider a two-dimensional plane problem and show a series of elements in which $\mathbf{N}_\sigma$ is discontinuous while $\mathbf{N}_u$ has C_0 continuity. We note again, by invoking the Veubeke 'principle of limitation', that all the elements that pass the single-element test here will in fact yield identical results to those obtained by using the equivalent irreducible form, providing the $\mathbf{D}$ matrix is constant within each element. They are therefore of little interest. However, we note in passing that the Q 4/8, which fails in a single-element test, passes that patch test for assemblies of two or more elements, and performs well in many circumstances. We shall see later that this is equivalent to using four-point Gauss, *reduced* integration (see Sec. 12.5), and as we have mentioned in Chapter 10 such elements will not always be robust.

It is of interest to note that if a higher order of interpolation is used for $\boldsymbol{\sigma}$ than for $\mathbf{u}$ the patch test is still satisfied, but in general the results will not be improved because of the principle of limitation.

We do not show the similar patch test for the C_0 continuous $\mathbf{N}_\sigma$ assumption but state simply that, similarly to the example of Fig. 11.3, identical interpolation of

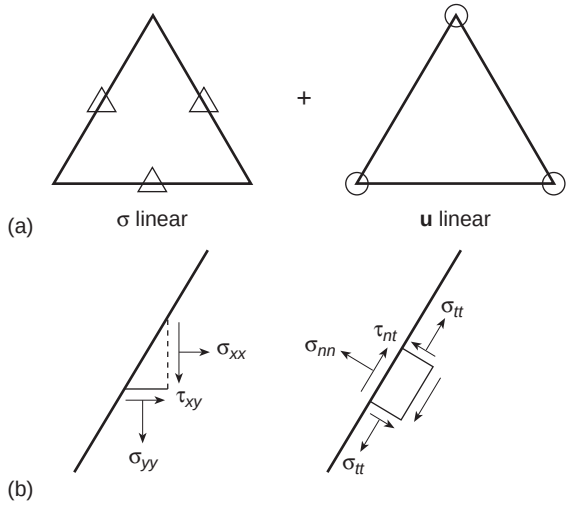


Fig. 11.6 Elasticity by the mixed **σ**–**u** formulation. Partially continuous **σ** (continuity at nodes only). (*a*) **σ** linear, **u** linear; (*b*) possible transformation of interface stresses with σ_{tt} disconnected.

$\mathbf{N}_\sigma$ and $\mathbf{N}_u$ is acceptable from the point of view of stability. However, as in Fig. 11.4, restriction of *excessive continuity* for stresses has to be avoided at singularities and at abrupt material property change interfaces, where only the normal and tangential tractions are continuous.

The disconnection of stress variables at corner nodes can only be accomplished for all the stress variables. For this reason an alternative set of elements with continuous stress nodes at element interfaces can be introduced (see Fig. 11.6).[19]

In suchs elements excessive continuity can easily be avoided by disconnecting only the direct stress components parallel to an interface at which material changes occur. It should be noted that even in the case when all stress components are connected at a mid-side node such elements do not ensure stress continuity along the whole interface. Indeed, the amount of such discontinuity can be useful as an error measure. However, we observe that for the linear element [Fig. 11.6(*a*)] the interelement stresses are continuous *in the mean*.

It is, of course, possible to derive elements that exhibit complete continuity of the appropriate components along interfaces and indeed this was achieved by Raviart and Thomas[20] in the case of the heat conduction problem discussed previously. Extension to the full stress problem is difficult[21] and as yet such elements have not been successfully noted.

11.4.4 Pian–Sumihara quadrilateral

Today very few two-field elements based on interpolation of the full stress and displacement fields are used. One, however, deserves to be mentioned. We begin by first considering a rectangular element where interpolations may be given directly in terms of cartesian coordinates. A four-node plane rectangular element with side

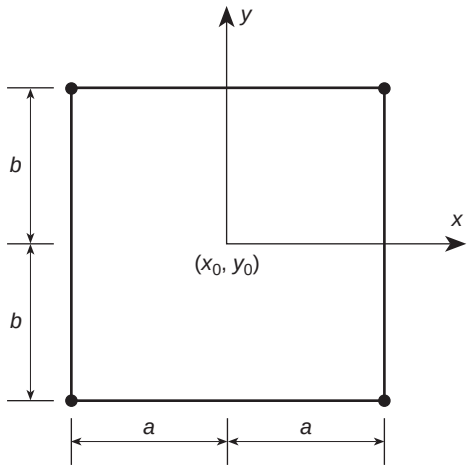


Fig. 11.7 Geometry of rectangular **σ**–**u** element.

lengths $2a$ in the x-direction and $2b$ in the y-direction, shown in Fig. 11.7, has displacement interpolation given by

$$\mathbf{u} = \sum_{i=1}^{4} N_i(x,y)\tilde{\mathbf{u}}_i$$

The shape functions are given by

$$N_1(x,y) = \frac{1}{4}\left(1 - \frac{x-x_0}{a}\right)\left(1 - \frac{y-y_0}{b}\right)$$

$$N_2(x,y) = \frac{1}{4}\left(1 + \frac{x-x_0}{a}\right)\left(1 - \frac{y-y_0}{b}\right)$$

$$N_3(x,y) = \frac{1}{4}\left(1 + \frac{x-x_0}{a}\right)\left(1 + \frac{y-y_0}{b}\right)$$

$$N_4(x,y) = \frac{1}{4}\left(1 - \frac{x-x_0}{a}\right)\left(1 + \frac{y-y_0}{b}\right)$$

where x_0 and y_0 are the cartesian coordinates of the element centre. The strains generated from this interpolation will be such that

$$\varepsilon_x = \beta_1 + \beta_2 y$$

$$\varepsilon_y = \beta_3 + \beta_4 x$$

$$\gamma_{xy} = \beta_5 + \beta_6 x + \beta_7 y$$

where β_j are expressed in terms of $\tilde{\mathbf{u}}$. For isotropic linear elasticity problems these strains will lead to stresses which have a complete linear polynomial variation in each element (except for the special case when $\nu = 0$).

Here the stress interpolation is restricted to each element individually and, thus, can be discontinuous between adjacent elements. The limitation principle restricts the possible choices which lead to different results from the standard displacement solution. Namely, the approximation must be less than a complete linear polynomial. To satisfy the stability condition given by Eq. (11.18) we need at least five stress parameters in each element. A viable choice for a five-term approximation is one which has the same variation in each element as the normal strains given above but only a constant shear stress. Accordingly,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & y-y_0 & 0 \\ 0 & 1 & 0 & 0 & x-x_0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{Bmatrix}$$

Indeed, this approximation satisfies Eq. (11.18) and leads to excellent results for a rectangular element. We now rewrite the formulation to permit a general quadrilateral shape to be used.

The element coordinate and displacement field are given by a standard bilinear isoparametric expansion

$$\mathbf{x} = \sum_{i=1}^{4} N_i(\xi,\eta)\tilde{\mathbf{x}}_i \qquad \hat{\mathbf{u}} = \sum_{i=1}^{4} N_i(\xi,\eta)\,\tilde{\mathbf{u}}_i$$

where now

$$N_i(\xi,\eta) = \tfrac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)$$

in which ξ_i and η_i are the values of the parent coordinates at the nodes.

The problem remains to deduce an approximation for stresses for the general quadrilateral element. Here this is accomplished by first assuming stresses on the parent element (for convenience in performing the coordinate transformation the tensor form is used, see Appendix B) in an analogous manner as the rectangle above:

$$\boldsymbol{\Sigma}(\xi,\eta) = \begin{bmatrix} \Sigma_{\xi\xi} & \Sigma_{\xi\eta} \\ \Sigma_{\eta\xi} & \Sigma_{\eta\eta} \end{bmatrix} = \begin{bmatrix} \alpha_1+\alpha_4\eta & \alpha_3 \\ \alpha_3 & \alpha_2+\alpha_5\xi \end{bmatrix}$$

In the above the normal stresses again produce constant and bending terms while shear stress is only constant. These stresses are then mapped (transformed) to cartesian space using

$$\boldsymbol{\sigma} = \mathbf{T}^{\mathrm{T}}\boldsymbol{\Sigma}(\xi,\eta)\mathbf{T}$$

It remains now only to select an appropriate transformation. The transformation must

1. produce stresses in cartesian space which satisfy the patch test (i.e., can produce constant stresses and be stable);

2. be independent of the orientation of the initially chosen element coordinate system and numbering of element nodes (frame invariance requirement).

Pian and Sumihara[22] use a constant array (to preserve constant stresses) deduced from the jacobian matrix at the centre of the element. Accordingly, with

$$\mathbf{J}_0 = \begin{bmatrix} J_{0,11} & J_{0,12} \\ J_{0,21} & J_{0,22} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}_{\xi,\eta=0}$$

the elements of the jacobian matrix at the centre are given by [see Eq. (8.10)]

$$J_{0,11} = \tfrac{1}{4} x_i \xi_i \qquad J_{0,12} = \tfrac{1}{4} x_i \eta_i$$
$$J_{0,21} = \tfrac{1}{4} y_i \xi_i \qquad J_{0,22} = \tfrac{1}{4} y_i \eta_i$$

Using $\mathbf{T} = \mathbf{J}_0$ gives the stresses (in matrix form)

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \bar{\alpha}_1 \\ \bar{\alpha}_2 \\ \bar{\alpha}_3 \end{Bmatrix} + \begin{bmatrix} J_{0,11}^2 \eta & J_{0,12}^2 \xi \\ J_{0,21}^2 \eta & J_{0,22}^2 \xi \\ J_{0,12} J_{0,21} \eta & J_{0,12} J_{0,22} \xi \end{bmatrix} \begin{Bmatrix} \alpha_4 \\ \alpha_5 \end{Bmatrix}$$

where the parameters $\bar{\alpha}_i$, $i = 1, 2, 3$, replace the transformed quantities for the constant part of the stresses. This approximation clearly satisfies the constant stress condition (Condition 1) and can also be shown to satisfy the frame invariance condition (Condition 2). The development is now complete and the arrays indicated

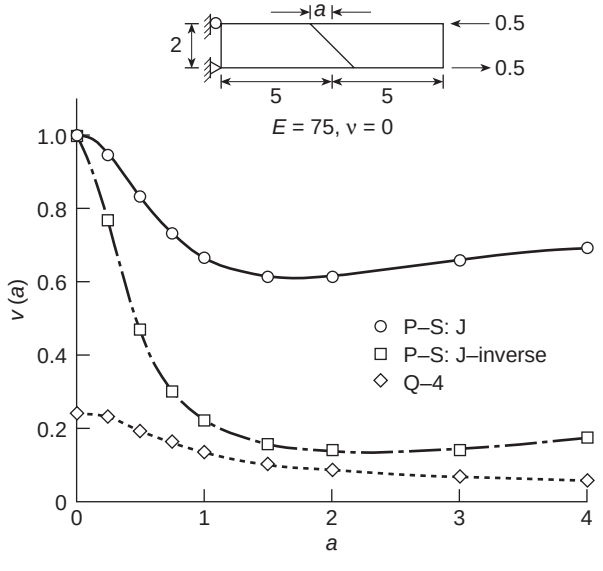


Fig. 11.8 Pian–Sumihara quadrilateral (P-S) compared with displacement quadrilateral (Q-4). Effect of element distortion (Exact = 1.0).

in Eq. (11.32) may be computed. We note that the integrals are computed exactly for all quadrilateral elements (with constant $\mathbf{D}$) using 2×2 gaussian quadrature.

An alternative to the above definition for $\mathbf{T}_0$ is to use the transpose of the jacobian inverse at the centre of the element (i.e., $\mathbf{T}_0 = \mathbf{J}_0^{-\mathrm{T}}$). This has also been suggested recently by several authors as a frame invariant transformation. However, as shown in Fig. 11.8, the sensitivity to element distortion is much greater for this form than the original one given by Pian and Sumihara for the above two-field approximation. The other two options (e.g., $\mathbf{T} = \mathbf{J}_0^{\mathrm{T}}$ and $\mathbf{T} = \mathbf{J}_0^{-1}$) do not satisfy the frame invariance requirement, thus giving elements which depend on the orientation of the element with respect to the global coordinates.

11.5 Three-field mixed formulations in elasticity

11.5.1 The u–σ–ε form

It is, of course, possible to use an independent approximation to all the essential variables entering the elasticity problem. We can then write the three equations (11.24), (11.23), and (11.22) in their weak form as

$$\begin{aligned} \int_\Omega \delta\boldsymbol{\varepsilon}^{\mathrm{T}}(\mathbf{D}\boldsymbol{\varepsilon} - \boldsymbol{\sigma})\,\mathrm{d}\Omega &= 0 \\ \int_\Omega \delta\boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{S}\mathbf{u} - \boldsymbol{\varepsilon})\,\mathrm{d}\Omega &= 0 \\ \int_\Omega \delta(\mathbf{S}\mathbf{u})^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma &= 0 \end{aligned} \tag{11.34}$$

with a corresponding variational principle requiring the stationarity of

$$\Pi_{\mathrm{HW}} = \int_\Omega \tfrac{1}{2}\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\varepsilon}\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{\sigma}^{\mathrm{T}}(\boldsymbol{\varepsilon} - \mathbf{S}\mathbf{u})\,\mathrm{d}\Omega - \int_\Omega \mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{11.35}$$

where $\mathbf{u} \equiv \bar{\mathbf{u}}$ on Γ_u is enforced.† This principle is known by the name of Hu–Washizu.[5] However, again we can proceed directly, using Eq. (11.34), taking the following approximations

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_u\tilde{\mathbf{u}} \qquad \boldsymbol{\sigma} \approx \hat{\boldsymbol{\sigma}} = \mathbf{N}_\sigma\tilde{\boldsymbol{\sigma}} \qquad \text{and} \qquad \boldsymbol{\varepsilon} \approx \hat{\boldsymbol{\varepsilon}} = \mathbf{N}_\varepsilon\tilde{\boldsymbol{\varepsilon}}$$

with corresponding 'variations' (i.e., the Galerkin form $\mathbf{W}_u = \mathbf{N}_u$, etc.) and writing the approximating equations in a similar fashion as we have in the previous section. This yields an equation system of the following form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} & \mathbf{0} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} & \mathbf{E} \\ \mathbf{0} & \mathbf{E}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{\sigma}} \\ \tilde{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{Bmatrix} \tag{11.36}$$

† It is possible to include the displacement boundary conditions in Eq. (11.35) as a natural rather than imposed constraint; however, most finite element applications of the principle are in the form shown.

where

$$\begin{aligned}
\mathbf{A} &= \int_\Omega \mathbf{N}_\varepsilon^{\mathrm{T}} \mathbf{D} \mathbf{N}_\varepsilon \, \mathrm{d}\Omega \\
\mathbf{E} &= \int_\Omega \mathbf{N}_\sigma^{\mathrm{T}} \mathbf{B} \, \mathrm{d}\Omega \\
\mathbf{C} &= -\int_\Omega \mathbf{N}_\varepsilon^{\mathrm{T}} \mathbf{N}_\sigma \, \mathrm{d}\Omega \\
\mathbf{f}_1 &= \mathbf{f}_2 = 0 \\
\mathbf{f}_3 &= \int_\Omega \mathbf{N}_u^{\mathrm{T}} \mathbf{b} \, \mathrm{d}\Omega + \int_{\Gamma_t} \mathbf{N}_u^{\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d}\Gamma
\end{aligned} \tag{11.37}$$

The reader will have observed again that in this section we have quoted the variational principle purely as a matter of interest and that all the approximations have been made directly.

11.5.2 Stability condition of three-field approximation (u–σ–ε)

The stability condition derived in Sec. 11.3 [Eq. (11.18)] for two-field problems, which we later used in Eq. (11.33) for the simple mixed elasticity form, needs to be modified when three-field approximations of the form given in Eq. (11.36) are considered.

Many other problems fall into a similar category (for instance, plate bending) and hence the conditions of stability are generally useful. The requirement now is that

$$\begin{aligned}
n_\varepsilon + n_u &\geqslant n_\sigma \\
n_\sigma &\geqslant n_u
\end{aligned} \tag{11.38}$$

This was first stated in reference 23 and follows directly from the two-field criterion as shown below.

The system of Eq. (11.36) can be 'regularized' by adding $\gamma\mathbf{E}$ times the third equation to the second, with γ being an arbitrary constant. We now have

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} & \mathbf{0} \\ \mathbf{C}^{\mathrm{T}} & \gamma\mathbf{E}\mathbf{E}^{\mathrm{T}} & \mathbf{E} \\ \mathbf{0} & \mathbf{E}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{\sigma}} \\ \tilde{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 + \gamma\mathbf{E}\mathbf{f}_3 \\ \mathbf{f}_3 \end{Bmatrix}$$

On elimination of $\boldsymbol{\varepsilon}$ using the first of the above we have

$$\begin{bmatrix} \gamma\mathbf{E}\mathbf{E}^{\mathrm{T}} - \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{C}, & \mathbf{E} \\ \mathbf{E}^{\mathrm{T}}, & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}} \\ \tilde{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_2 + \gamma\mathbf{E}\mathbf{f}_3 - \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{f}_1 \\ \mathbf{f}_3 \end{Bmatrix}$$

From the two-field requirement [Eq. (11.18)] it follows that we require for no singularity

$$n_\sigma \geqslant n_u \tag{11.39}$$

Rearranging Eq. (11.36) we can write

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{E}^{\mathrm{T}} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{E} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\mathbf{u}} \\ \tilde{\boldsymbol{\sigma}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_3 \\ \mathbf{f}_2 \end{Bmatrix}$$

This again can be regularized by adding multiples $\gamma\mathbf{C}$ and $\gamma\mathbf{E}^{\mathrm{T}}$ of the third of the above equations to the first and second respectively obtaining

$$\left[\begin{array}{ccc} \mathbf{A}+\gamma\mathbf{C}\mathbf{C}^{\mathrm{T}}, & \gamma\mathbf{C}\mathbf{E} & \mathbf{C} \\ \gamma\mathbf{E}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}, & \gamma\mathbf{E}^{\mathrm{T}}\mathbf{E} & \mathbf{E}^{\mathrm{T}} \\ \hline \mathbf{C}^{\mathrm{T}}, & \mathbf{E} & \mathbf{0} \end{array}\right] \begin{Bmatrix} \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\mathbf{u}} \\ \tilde{\boldsymbol{\sigma}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 + \gamma\mathbf{C}\mathbf{f}_2 \\ \mathbf{f}_3 + \gamma\mathbf{E}^{\mathrm{T}}\mathbf{f}_2 \\ \mathbf{f}_2 \end{Bmatrix}$$

By partitioning as above it is evident that we require

$$n_\varepsilon + n_u \geqslant n_\sigma \tag{11.40}$$

We shall not discuss in detail any of the possible approximations to the $\boldsymbol{\varepsilon}$–$\boldsymbol{\sigma}$–$\mathbf{u}$ formulation or their corresponding patch tests as the arguments are similar to those of two-field problems.

In some practical applications of the three-field form the approximation of the second and third equations in (11.34) is used directly to eliminate all but the displacement terms. This leads to a special form of the displacement method which has been called a $\bar{\mathbf{B}}$ ($\mathbf{B}$-bar) form.[24,25] In the $\bar{\mathbf{B}}$ form the shape function derivatives are replaced by approximations resulting from the mixed form. We shall illustrate this concept with an example of a *nearly incompressible* material in Sec. 12.4.

11.5.3 The u–σ–ε_{en} form. Enhanced strain formulation

In the previous two sections the general form and stability conditions of the three-field formulation for elasticity problems is given in Eqs (11.34) and (11.38). Here we consider a special case of this form from which several useful elements may be deduced.

In the special form considered the strain approximation is split into two parts: one the usual displacement-gradient term; and, second, an added or *enhanced strain* part. Accordingly, we write

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u} + \boldsymbol{\varepsilon}_{\mathrm{en}} \qquad \delta\boldsymbol{\varepsilon} = \delta(\mathbf{S}\mathbf{u}) + \delta\boldsymbol{\varepsilon}_{\mathrm{en}} \tag{11.41}$$

Substitution into Eq. (11.34) yields the weak forms as

$$\begin{aligned} \int_\Omega \delta(\mathbf{S}\mathbf{u})^{\mathrm{T}}(\mathbf{D}(\mathbf{S}\mathbf{u}+\boldsymbol{\varepsilon}_{\mathrm{en}}) - \boldsymbol{\sigma})\,\mathrm{d}\Omega = 0 \\ \int_\Omega \delta\boldsymbol{\varepsilon}_{\mathrm{en}}^{\mathrm{T}}(\mathbf{D}(\mathbf{S}\mathbf{u}+\boldsymbol{\varepsilon}_{\mathrm{en}}) - \boldsymbol{\sigma})\,\mathrm{d}\Omega = 0 \\ \int_\Omega \delta\boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon}_{\mathrm{en}}\,\mathrm{d}\Omega = 0 \\ \int_\Omega \delta(\mathbf{S}\mathbf{u})^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \end{aligned} \tag{11.42}$$

with, for completeness, a corresponding variational principle requiring the stationarity of

$$\Pi_{\text{en}} = \int_\Omega \frac{1}{2}(\mathbf{Su} + \boldsymbol{\varepsilon}_{\text{en}})^{\mathrm{T}}\mathbf{D}\,(\mathbf{Su} + \boldsymbol{\varepsilon}_{\text{en}})\,\mathrm{d}\Omega + \int \boldsymbol{\sigma}\boldsymbol{\varepsilon}_{\text{en}}\,\mathrm{d}\Omega - \int_\Omega \mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{11.43}$$

where, as before, $\mathbf{u} = \bar{\mathbf{u}}$ is enforced on Γ_u.

We can directly discretize Eq. (11.42) by taking the following approximations

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_u\tilde{\mathbf{u}} \qquad \boldsymbol{\sigma} \approx \hat{\boldsymbol{\sigma}} = \mathbf{N}_\sigma\tilde{\boldsymbol{\sigma}} \qquad \boldsymbol{\varepsilon}_{\text{en}} \approx \hat{\boldsymbol{\varepsilon}}_{\text{en}} = \mathbf{N}_{\text{en}}\tilde{\boldsymbol{\varepsilon}}_{\text{en}} \tag{11.44}$$

with corresponding expressions for variations. Substituting the approximations into Eq. (11.42) yields the discrete equation system

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} & \mathbf{G} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}^{\mathrm{T}} & \mathbf{0} & \mathbf{K} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\varepsilon}}_{\text{en}} \\ \tilde{\boldsymbol{\sigma}} \\ \tilde{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{Bmatrix} \tag{11.45}$$

where

$$\begin{aligned} \mathbf{A} &= \int_\Omega \mathbf{N}_{\text{en}}^{\mathrm{T}}\mathbf{D}\mathbf{N}_{\text{en}}\,\mathrm{d}\Omega \\ \mathbf{C} &= -\int_\Omega \mathbf{N}_{\text{en}}^{\mathrm{T}}\mathbf{N}_\sigma\,\mathrm{d}\Omega \\ \mathbf{G} &= \int_\Omega \mathbf{N}_{\text{en}}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d}\Omega \\ \mathbf{K} &= \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d}\Omega \\ \mathbf{f}_1 &= \mathbf{f}_2 = \mathbf{0} \\ \mathbf{f}_3 &= \int_\Omega \mathbf{N}_u^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_t} \mathbf{N}_u^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \end{aligned} \tag{11.46}$$

In this form there is only one zero diagonal term and the stability condition reduces to the single condition

$$n_u + n_{\text{en}} \geqslant n_\sigma \tag{11.47}$$

Further, the use of the strains deduced from the displacement interpolation leads to a matrix which is identical to that from the irreducible form and we have thus included this in Eq. (11.46) as $\mathbf{K}$.

11.5.4 Simo–Rifai quadrilateral

An enhanced strain formulation for application to problems in plain elasticity was introduced by Simo and Rifai.[26] The element has four nodes and employs iso-parametric interpolation for the displacement field. The derivatives of the shape

functions yield a form

$$\begin{Bmatrix} \dfrac{\partial N_j}{\partial x} \\ \dfrac{\partial N_j}{\partial y} \end{Bmatrix} = \begin{Bmatrix} \dfrac{a_{x,j}(y_i) + b_{x,j}(y_i)\xi + c_{x,j}(y_i)\eta}{j(\xi,\eta)} \\ \dfrac{a_{y,j}(x_i) + b_{y,j}(x_i)\xi + c_{y,j}(x_i)\eta}{j(\xi,\eta)} \end{Bmatrix}$$

where a_j, b_j and c_j depend on the nodal coordinates, and the jacobian determinant for the 4-node quadrilateral is given by†

$$\det \mathbf{J} = j(\xi,\eta) = j_0 + j_1\xi + j_2\eta$$

The enhanced strains are first assumed in the parent coordinate frame and transformed to the cartesian frame using a transformation similar to that used in developing the Pian–Sumihara quadrilateral in Sec. 11.4.2. Due to the presence of the jacobian determinant in the strains computed from the displacements (as well as the requirement to later pass the patch test for constant stress states) the enhanced strains are computed from

$$\boldsymbol{\varepsilon}_{\text{en}} = \frac{1}{j(\xi,\eta)}\,\mathbf{T}^{\mathrm{T}}\mathbf{E}(\xi,\eta)\mathbf{T}$$

In matrix form this may be written as

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \frac{1}{j(\xi,\eta)} \begin{bmatrix} T_{11}^2 & T_{21}^2 & T_{11}T_{21} \\ T_{12}^2 & T_{22}^2 & T_{12}T_{22} \\ 2T_{11}T_{12} & 2T_{21}T_{22}^2 & T_{11}T_{22} + T_{12}T_{21} \end{bmatrix} \begin{Bmatrix} E_{\xi\xi} \\ E_{\eta\eta} \\ 2E_{\xi\eta} \end{Bmatrix}$$

The parent strains (strains with components in the parent element frame) are assumed as

$$\begin{Bmatrix} E_{\xi\xi} \\ E_{\eta\eta} \\ 2E_{\xi\eta} \end{Bmatrix} = \begin{bmatrix} \xi & 0 & 0 & 0 \\ 0 & \eta & 0 & 0 \\ 0 & 0 & \xi & \eta \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{Bmatrix}$$

The above is motivated by the fact that the derivatives of the shape functions with respect to parent coordinates yields

$$\frac{\partial N_j}{\partial \xi} = a_\xi + b_\xi\eta \qquad \frac{\partial N_j}{\partial \eta} = a_\eta + b_\eta\xi$$

and these may be combined to form strains in the usual manner, but in the parent frame. Thus, by design, the above enhanced strains are specified to generate complete polynomials in the parent coordinates for each strain component. References 27 and 28 discuss the relationship between the design of assumed stress elements using the two-field form and the selection of enhanced strain modes so as to produce the same result.

† In general, the determinant of the jacobian for the two-dimensional Lagrange family of elements will not contain the term with the product of the highest order polynomial, e.g., $\xi\eta$ for the 4-node element, $\xi^2\eta^2$ for the 9-node element, etc.

Remarks

1. The above enhanced strains are defined so that the **C** array is identically zero for constant assumed stresses in each element.
2. Parent normal strains have linearly independent terms added. However, the assumed parent shear strains are linearly dependent. Due to this linear dependence the final shearing strain will usually be nearly constant in each element. Accordingly, to be more explicit, normal strains are *enhanced* while shearing strain is *de-enhanced*.

Since the **C** array vanishes, the equation set to be solved becomes

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^{\mathrm{T}} & \mathbf{K} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\varepsilon}}_{\mathrm{en}} \\ \tilde{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_3 \end{Bmatrix}$$

and in this form no additional count conditions are apparently needed. The solution may be accomplished partly at the element level by eliminating the equation associated with the enhanced strain parameters. Accordingly,

$$\mathbf{K}^{*}\tilde{\mathbf{u}} = \mathbf{f}_3^{*}$$

where

$$\mathbf{K}^{*} = \mathbf{K} - \mathbf{G}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{G}$$

and

$$\mathbf{f}_3^{*} = \mathbf{f}_3 - \mathbf{G}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{f}_1$$

The sensitivity of the enhanced strain element to geometric distortion is evaluated using the problem shown in Fig. 11.8. The transformation from the parent to the global frame is assessed using $\mathbf{T} = \mathbf{J}_0$ and $\mathbf{T} = \mathbf{J}_0^{-\mathrm{T}}$. These are the only options which maintain frame invariance for the element. As observed in Fig. 11.9 the results

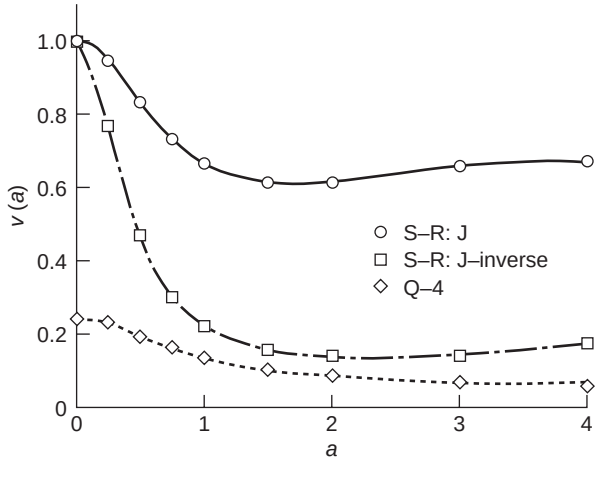


Fig. 11.9 Simo–Rifai enhanced strain quadrilateral (S-R) compared with displacement quadrilateral (Q-4). Effect of element distortion (Exact = 1.0).

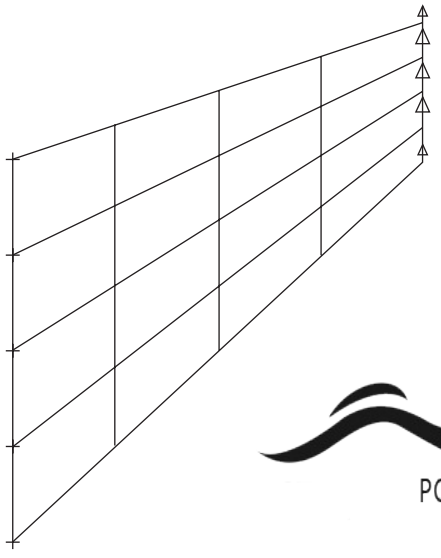


Fig. 11.10 Mesh with 4 × 4 elements for shear load.

are now better using the inverse transpose. Since the stress and strain are conjugates in an energy sense, this result could be anticipated from the equivalence relationship

$$E = \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon}\,\mathrm{d}\Omega \equiv \frac{1}{2}\int_{\square} \boldsymbol{\Sigma}^{\mathrm{T}}\mathbf{E}\,\mathrm{d}\square$$

where E is energy and $\square$ denotes the domain of the element in the parent coordinate system (i.e., the bi-unit square for a quadrilateral element).

The performance of the enhanced element is compared to the Pian–Sumihara element for a shear loading on the mesh shown in Fig. 11.10. In Fig. 11.11 the convergence results for various order meshes are shown for linear elastic, plane strain conditions with: (*a*) $E = 70$ and $\nu = 1/3$ and (*b*) for $E = 70$ and $\nu = 0.499995$. The results shown in Fig. 11.11 clearly show the strong dependence of the displacement

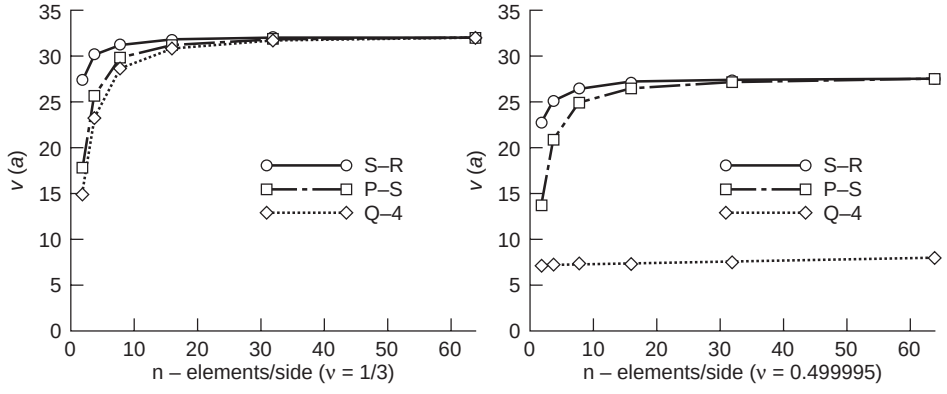


Fig. 11.11 Convergence behaviour for: (a) $\nu = 1/3$; (b) $\nu = 0.499995$.

formulation on Poisson's ratio – namely the tendency for the element to *lock* for values which approach the incompressibility limit of $\nu = 1/2$. On the other hand, the performance of both the enhanced strain and the Pian–Sumihara element are nearly insensitive to the value of Poisson's ratio selected, with somewhat better performance of the enhanced element on coarse meshing.

11.6 An iterative method solution of mixed approximations

It is of interest to consider here the procedure first suggested by Cantin *et al.*[29,30] in which the authors aimed at an iterative improvement of the displacement type solution. This iterative process in fact solves two equations. In this the first equation replaces the discontinuous stresses computed from a displacement type solution by continuous stresses calculated by a *least square* smoothing. The continuous stress is expressed using

$$\boldsymbol{\sigma}^* = \mathbf{N}\tilde{\boldsymbol{\sigma}} \tag{11.48}$$

where $\mathbf{N}$ are the same shape functions used in the displacement solution and $\tilde{\boldsymbol{\sigma}}$ are nodal values of stresses. The least square problem is then expressed as

$$\int_\Omega (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}})^{\mathrm{T}} (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}})\, \mathrm{d}\Omega = \min \tag{11.49}$$

whose solution for a typical iteration i may be written as

$$\mathbf{A}\tilde{\boldsymbol{\sigma}}^{(k+1)} - \mathbf{C}^* \tilde{\mathbf{u}}^{(k)} = \mathbf{0} \tag{11.50}$$

with

$$\mathbf{A} = \int_\Omega \mathbf{N}^{\mathrm{T}}\mathbf{N}\, \mathrm{d}\Omega$$

$$\mathbf{C}^* = \int_\Omega \mathbf{N}^{\mathrm{T}}\mathbf{D}\mathbf{B}\, \mathrm{d}\Omega$$

This type of stress smoothing was suggested by Brauchli and Oden in 1973.[31] Though we shall discuss its achievements later in Chapter 14 on recovery methods it has been quite successfully used in the iterative improvement discussed here.

The second stage of the calculation takes the stresses computed above $\boldsymbol{\sigma}^*$ and calculates the out-of-balance residual

$$\mathbf{r}^{(k+1)} = \int_\Omega \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma}^{*,(k+1)}\, \mathrm{d}\Omega + \mathbf{f} \tag{11.51}$$

The correction to the displacements using this residual is then expressed by

$$\mathbf{K}\tilde{\mathbf{u}}^{(k+1)} = \mathbf{K}\tilde{\mathbf{u}}^{(k)} - \mathbf{r}^{(k+1)} \tag{11.52}$$

The iteration may now proceed by incrementing k and computing new smoothed stresses followed by new displacements.

The two steps may be written in a matrix setting as

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C}^{\mathrm{T}} & -\mathbf{K} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}}^{(k+1)} \\ \mathbf{u}^{(k+1)} \end{Bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{C}^* \\ \mathbf{0} & -\mathbf{K} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}}^{(k)} \\ \mathbf{u}^{(k)} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ -\mathbf{f} \end{Bmatrix} \tag{11.53}$$

where

$$\mathbf{C} = \int_{\Omega} \mathbf{N}^{\mathrm{T}} \mathbf{B} \, \mathrm{d}\Omega \tag{11.54}$$

At convergence the solutions become

$$\tilde{\mathbf{u}}^{(k)} = \tilde{\mathbf{u}}^{(k+1)} = \tilde{\mathbf{u}} \qquad \tilde{\boldsymbol{\sigma}}^{(k)} = \tilde{\boldsymbol{\sigma}}^{(k+1)} = \tilde{\boldsymbol{\sigma}}$$

Combining the two sides of the above equation yields

$$\begin{bmatrix} \mathbf{A} & \mathbf{C}^{*} \\ -\mathbf{C}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}} \\ \mathbf{u} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ -\mathbf{f} \end{Bmatrix} \tag{11.55}$$

The reader will notice that the equations which result at the end of this process are in fact a mixed problem in stress and displacement form.

The convergence of the process is quite rapid and very often considerable improvement in the answers is obtained. In Fig. 11.12 we show some results by Nakazawa *et al.*[32–34] using the bilinear displacement element and it is seen how much the results are improved. In Fig. 11.13 a similar iteration is carried out using now triangular

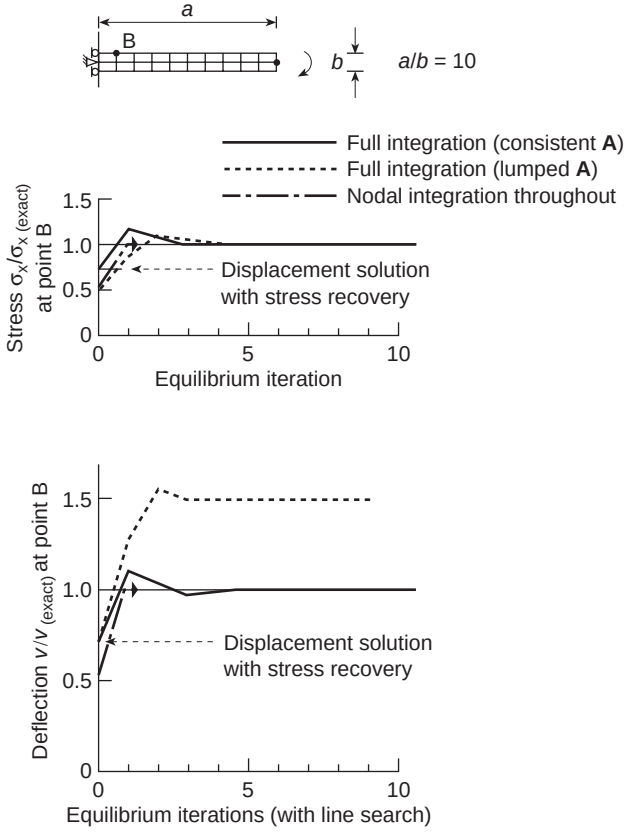


Fig. 11.12 Iterative solution of the mixed **σ/u** formulation for a beam. Bilinear **u** and **σ**.

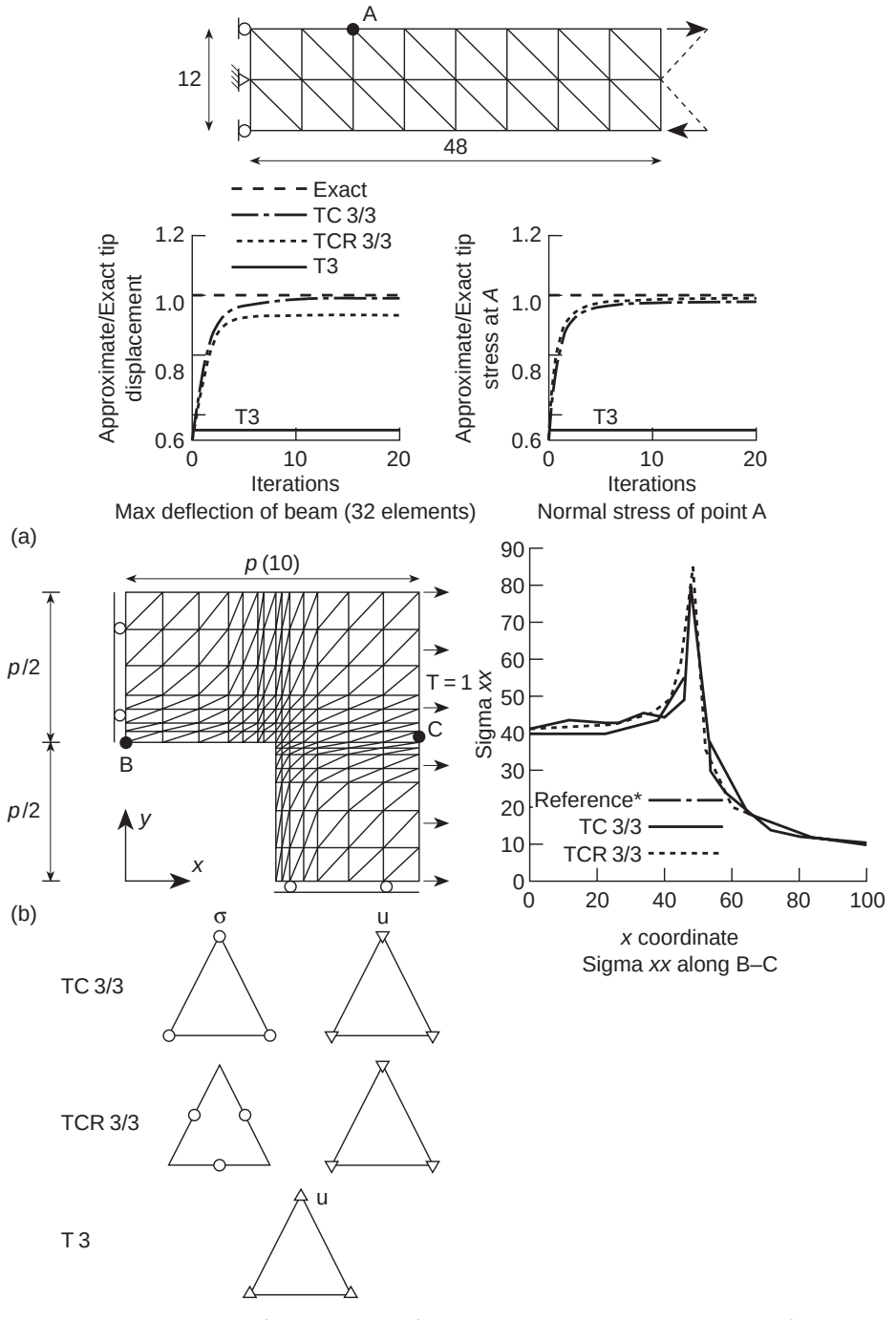


Fig. 11.13 Iterative solution of the mixed **σ/u** formulation using two triangular element forms TC 3/3 and TCR 3/3. (a) A beam showing convergence with iterations. (b) An L-shaped domain showing the improved results of stress distribution when no continuity of stress is imposed at singularity (element TCR 3/3).

elements. Here various combinations of displacement and stress variation have been used and, in particular, the reader should note that at the singularity point some means of stress disconnection is used as difficulties in C_0 stress continuity exist. The very simplest procedure of disconnecting all components of stress at such points has proven to be optimal. Details of such calculations are given in reference 19.

In a subsequent chapter, where we shall deal with problems of incompressibility, we shall deal with an iteration due to Uzawa.[35] The particular iteration used in the above iteration process is in fact a form of the Uzawa algorithm to which we will refer in more detail later.

11.7 Complementary forms with direct constraint

11.7.1 General forms

In the introduction to this chapter we defined the irreducible and mixed forms and indicated that on occasion it is possible to obtain more than one 'irreducible' form. To illustrate this in the problem of heat transfer given by Eqs (11.2) and (11.3) we introduced a penalty function α in Eq. (11.6) and derived a corresponding single governing equation (11.7) given in terms of $\mathbf{q}$. This penalty function here has no obvious physical meaning and served simply as a device to obtain a *close enough* approximation to the satisfaction of the continuity of flow equations.

On occasion it is possible to solve the problem as an irreducible one assuming *a priori* that the choice of the variable satisfies one of the equations. We call such forms *directly constrained* and obviously the choice of the shape function becomes difficult.

We shall consider two examples.

The complementary heat transfer problem

In this we assume *a priori* that the choice of $\mathbf{q}$ is such that it satisfies Eq. (11.3) and the natural boundary conditions

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q} = -Q \text{ in } \Omega \qquad \text{and} \qquad q_n = \bar{q}_n \text{ on } \Gamma_q \tag{11.56}$$

Thus we only have to satisfy the constitutive relation (11.2), i.e.,

$$\mathbf{k}^{-1}\mathbf{q} + \boldsymbol{\nabla}\phi = \mathbf{0} \text{ in } \Omega \qquad \text{with} \qquad \phi = \bar{\phi} \text{ on } \Gamma_\phi \tag{11.57}$$

A weak statement of the above is

$$\int_\Omega \delta\mathbf{q}^{\mathrm{T}}(\mathbf{k}^{-1}\mathbf{q} + \boldsymbol{\nabla}\phi)\,\mathrm{d}\Omega - \int_{\Gamma_\phi} \delta q_n(\phi - \bar{\phi})\,\mathrm{d}\Gamma = 0 \tag{11.58}$$

in which δq_n represents the variation of normal flux on the boundary.

Use of Green's theorem transforms the above into

$$\int_\Omega \delta\mathbf{q}^{\mathrm{T}}\mathbf{k}^{-1}\mathbf{q}\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{\nabla}^{\mathrm{T}}\delta\mathbf{q}\phi\,\mathrm{d}\Omega + \int_{\Gamma_\phi} \delta q_n\bar{\phi}\,\mathrm{d}\Gamma + \int_{\Gamma_q} \delta q_n\phi\,\mathrm{d}\Gamma = 0 \tag{11.59}$$

If we further assume that $\nabla^{\mathrm{T}}\delta\mathbf{q} \equiv 0$ in Ω and $\delta q_n = 0$ on Γ_q, i.e., that the weighting functions are simply the variations of $\mathbf{q}$, the equation reduces to

$$\int_{\Omega} \delta\mathbf{q}^{\mathrm{T}}\mathbf{k}^{-1}\mathbf{q}\,\mathrm{d}\Omega + \int_{\Gamma_\phi} \delta q_n \bar{\phi}\,\mathrm{d}\Gamma = 0 \tag{11.60}$$

This is in fact the variation of a complementary flux principle

$$\Pi = \int_{\Omega} \tfrac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^{-1}\mathbf{q}\,\mathrm{d}\Omega + \int_{\Gamma_\phi} q_n \bar{\phi}\,\mathrm{d}\Gamma \tag{11.61}$$

Numerical solutions can obviously be started from either of the above equations but the difficulty is the choice of the trial function satisfying the constraints. We shall return to this problem in Sec. 11.7.2.

The complementary elastic energy principle

In the elasticity problem specified in Sec. 11.4 we can proceed similarly, assuming stress fields which satisfy the equilibrium conditions both on the boundary Γ_t and in the domain Ω.

Thus in an analogous manner to that of the previous example we impose on the permissible stress field the constraints which we assume to be satisfied by the approximation identically, i.e.,

$$\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \text{ in } \Omega \qquad \text{and} \qquad \mathbf{t} = \bar{\mathbf{t}} \text{ on } \Gamma_t \tag{11.62}$$

Thus only the constitutive relations and displacement boundary conditions remain to be satisfied, i.e.,

$$\mathbf{D}^{-1}\boldsymbol{\sigma} - \mathbf{S}\mathbf{u} = \mathbf{0} \text{ in } \Omega \qquad \text{and} \qquad \mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u \tag{11.63}$$

The weak statement of the above can be written as

$$\int_{\Omega} \delta\boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{D}^{-1}\boldsymbol{\sigma} - \mathbf{S}\mathbf{u})\,\mathrm{d}\Omega + \int_{\Gamma_u} \delta\mathbf{t}^{\mathrm{T}}(\mathbf{u} - \bar{\mathbf{u}})\,\mathrm{d}\Gamma = 0 \tag{11.64}$$

which on integration by Green's theorem gives

$$\int_{\Omega} \delta\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{D}^{-1}\boldsymbol{\sigma}\,\mathrm{d}\Omega + \int_{\Omega} (\mathbf{S}^{\mathrm{T}}\delta\boldsymbol{\sigma})^{\mathrm{T}}\mathbf{u}\,\mathrm{d}\Omega - \int_{\Gamma_u} \delta\mathbf{t}^{\mathrm{T}}\bar{\mathbf{u}}\,\mathrm{d}\Gamma - \int_{\Gamma_t} \delta\mathbf{t}^{\mathrm{T}}\mathbf{u}\,\mathrm{d}\Gamma = 0 \tag{11.65}$$

Again assuming that the test functions are complete variations satisfying the homogeneous equilibrium equation, i.e.,

$$\mathbf{S}^{\mathrm{T}}\delta\boldsymbol{\sigma} = \mathbf{0} \text{ in } \Omega \qquad \text{and} \qquad \delta\mathbf{t} = \mathbf{0} \text{ on } \Gamma_t \tag{11.66}$$

we have as the weak statement

$$\int_{\Omega} \delta\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{D}^{-1}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_{\Gamma_u} \delta\mathbf{t}^{\mathrm{T}}\bar{\mathbf{u}}\,\mathrm{d}\Gamma = 0 \tag{11.67}$$

The corresponding complementary energy variational principle is

$$\Pi = \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{D}^{-1}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_{\Gamma_u} \mathbf{t}^{\mathrm{T}}\bar{\mathbf{u}}\,\mathrm{d}\Gamma \tag{11.68}$$

Once again in practical use the difficulties connected with the choice of the approximating function arise but on occasion a direct choice is possible.[30]

11.7.2 Solution using auxiliary functions

Both the complementary forms can be solved using auxiliary functions to ensure the satisfaction of the constraints.

In the *heat transfer problem* it is easy to verify that the homogeneous equation

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q} \equiv \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = 0 \tag{11.69}$$

is automatically satisfied by defining a function ψ such that

$$q_x = \frac{\partial \psi}{\partial y} \qquad q_y = -\frac{\partial \psi}{\partial x} \tag{11.70}$$

Thus we define

$$\mathbf{q} = \mathbf{L}\psi + \mathbf{q}_0 \qquad \text{and} \qquad \delta\mathbf{q} = \mathbf{L}\delta\psi \tag{11.71}$$

where $\mathbf{q}_0$ is any flux chosen so that

$$\boldsymbol{\nabla}^{\mathrm{T}}\mathbf{q}_0 = -Q \tag{11.72}$$

and

$$\mathbf{L} = \left[\frac{\partial}{\partial y}, -\frac{\partial}{\partial x}\right]^{\mathrm{T}} \tag{11.73}$$

the formulations of Eqs (11.60) and (11.61) can be used without any constraints and, for instance, the stationarity

$$\Pi = \int_{\Omega} \frac{1}{2}(\mathbf{L}\psi + \mathbf{q}_0)^{\mathrm{T}}\mathbf{k}^{-1}(\mathbf{L}\psi + \mathbf{q}_0)\,\mathrm{d}\Omega - \int_{\Gamma_\phi} \left(\frac{\partial \psi}{\partial s}\right)\bar{\phi}\,\mathrm{d}\Gamma \tag{11.74}$$

will suffice to so formulate the problem (here s is the tangential direction to the boundary).

The above form will require shape functions for ψ satisfying C_0 continuity.

In the corresponding elasticity problem a similar two-dimensional form can be obtained by the use of the so-called Airy stress function ψ.[36]

Now the equilibrium equations

$$\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma} + \mathbf{b} \equiv \begin{Bmatrix} \dfrac{\partial \sigma_x}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + b_x \\ \dfrac{\partial \tau_{xy}}{\partial x} + \dfrac{\partial \sigma_y}{\partial y} + b_y \end{Bmatrix} = \mathbf{0} \tag{11.75}$$

are identically solved by choosing

$$\boldsymbol{\sigma} = \mathbf{L}\psi + \boldsymbol{\sigma}_0 \tag{11.76}$$

where

$$\mathbf{L} = \left[\frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial x^2}, -\frac{\partial^2}{\partial x\,\partial y}\right]^{\mathrm{T}} \tag{11.77}$$

and $\boldsymbol{\sigma}_0$ is an arbitrary stress chosen so that

$$\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma}_0 + \mathbf{b} = \mathbf{0} \tag{11.78}$$

Again the substitution of (11.76) into the weak statement (11.67) or the complementary variational problem (11.68) will yield a direct formulation to which no additional constraints need be applied. However, use of the above forms does lead to further complexity in multiply connected regions where further conditions are needed. The reader will note that in Chapter 7 we encountered this in a similar problem in torsion and suggested a very simple procedure of avoidance (see Sec. 7.5).

The use of this stress function formulation in the two-dimensional context was first made by de Veubeke and Zienkiewicz[37] and Elias,[38] but the reader should note that now with second-order operators present, C_1 continuity of shape functions is needed in a similar manner to the problems which we have to consider in plate bending (see Volume 2).

Incidentally, analogies with plate bending go further here and indeed it can be shown that some of these can be usefully employed for other problems.[39]

11.8 Concluding remarks – mixed formulation or a test of element 'robustness'

The mixed form of finite element formulation outlined in this chapter opens a new range of possibilities, many with potentially higher accuracy and robustness than those offered by irreducible forms. However, an additional advantage arises even in situations where, by the *principle of limitation*, the irreducible and mixed forms yield identical results. Here the study of the behaviour of the mixed form can frequently reveal weaknesses or lack of 'robustness' in the irreducible form which otherwise would be difficult to determine.

The mixed approximation, if properly understood, expands the potential of the finite element method and presents almost limitless possibilities of detailed improvement. Some of these will be discussed further in the next two chapters, and others in Volumes 2 and 3.

References

1. S.N. Atluri, R.H. Gallagher, and O.C. Zienkiewicz (eds). *Hybrid and Mixed Finite Element Methods*. Wiley, 1983.
2. O.C. Zienkiewicz, R.L. Taylor, and J.A.W. Baynham. Mixed and irreducible formulations in finite element analysis. Chapter 21 of *Hybrid and Mixed Finite Element Methods* (eds S.N. Atluri, R.H. Gallagher, and O.C. Zienkiewicz), pp. 405–31, Wiley, 1983.
3. I. Babuška and J.E. Osborn. Generalized finite element methods and their relations to mixed problems. *SIAM J. Num. Anal.* **20**, 510–36, 1983.
4. R.L. Taylor and O.C. Zienkiewicz. Complementary energy with penalty function in finite element analysis. Chapter 8 of *Energy Methods in Finite Element Analysis* (eds R. Glowinski, E.Y. Rodin, and O.C. Zienkiewicz). Wiley, 1979.
5. K. Washizu, *Variational Methods in Elasticity and Plasticity*. 2nd ed., Pergamon Press, 1975.
6. L.R. Herrmann. Finite element bending analysis of plates. *Proc. 1st Conf. Matrix Methods in Structural Mechanics*. AFFDL-TR-80-66, Wright-Patterson AF Base, Ohio, 1965.
7. K. Hellan. Analysis of plates in flexure by a simplified finite element method. *Acta Polytechnica Scandinavia*. Civ. Eng. Series 46, Trondheim, 1967.

8. R.S. Dunham and K.S. Pister. A finite element application of the Hellinger Reissner variational theorem. *Proc. 1st Conf. Matrix Methods in Structural Mechanics*. Wright-Patterson AF Base, Ohio, 1965.
9. R.L. Taylor and O.C. Zienkiewicz. Mixed finite element solution of fluid flow problems. Chapter 1 of *Finite Elements in Fluid*. Vol. 4 (eds R.H. Gallagher, D.N. Norrie, J.T. Oden, and O.C. Zienkiewicz), pp. 1–20, Wiley, 1982.
10. B. Fraeijs de Veubeke. Displacement and equilibrium models in finite element method. Chapter 9 of *Stress Analysis* (eds O.C. Zienkiewciz and C.S. Holister), pp. 145–97, Wiley, 1965.
11. I. Babuška. The finite element method with Lagrange multipliers. *Num. Math.* **20**, 179–92, 1973; also 'Error bounds for finite element methods. *Num. Math.* **16**, 322–33, 1971.
12. F. Brezzi. On the existence, uniqueness and approximation of saddle point problems arising from lagrangian multipliers. *RAIRO*. 8-R2, 129–51, 1974.
13. O.C. Zienkiewicz, S. Qu, R.L. Taylor, and S. Nakazawa. The patch test for mixed formulation. *Int. J. Num. Meth. Eng.* **23**, 1873–83, 1986.
14. J.T. Oden and N. Kikuchi. Finite element methods for constrained problems of elasticity. *Int. J. Num. Mech. Eng.* **18**, 701–25, 1982.
15. E. Hellinger. Die allgemeine Aussetze der Mechanik der Kontinua, in *Encyclopedia der Matematischen Wissenschaften*. Vol. 4 (eds F. Klein and C. Muller). Tebner, Leipzig, 1914.
16. E. Reissner. On a variational theorem in elasticity. *J. Math. Phys.* **29**, 90–5, 1950.
17. L.R. Herrmann. Finite element bending analysis of plates. *Proc. Am. Soc. Civ. Eng.* **94**, EM5, 13–25, 1968.
18. L.R. Herrmann and D.M. Campbell. A finite element analysis for thin shells. *JAIAA*. **6**, 1842–7, 1968.
19. O.C. Zienkiewicz and D. Lefebvre. Mixed methods for FEM and the patch test. Some recent developments, in *Analyse Mathematique of Application* (eds F. Murat and O. Pirenneau). Gauthier Villars, Paris, 1988.
20. P.A. Raviart and J.M. Thomas. A mixed finite element method for second order elliptic problems. *Lect. Notes in Math.* no. 606, pp. 292–315, Springer-Verlag, 1977.
21. D. Arnold, F. Brezzi, and J. Douglas. PEERS, a new mixed finite element for plane elasticity. *Japan J. Appl. Math.* **1**, 347–67, 1984.
22. T.H.H. Pian and K. Sumihara. Rational approach for assumed stress finite elements. *Int. J. Num. Meth. Eng.*, **20**, 1685–95, 1985.
23. O.C. Zienkiewicz and D. Lefebvre. Three field mixed approximation and the plate bending problem. *Comm. Appl. Num. Math.* **3**, 301–9, 1987.
24. T.J.R. Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *Int. J. Num. Meth. Eng.* **15**, 1413–18, 1980.
25. J.C. Simo, R.L. Taylor, and K.S. Pister. Variational and projection methods for the volume constraint in finite deformation plasticity. *Comp. Meth. App. Mech. Eng.* **51**, 177–208, 1985.
26. J.C. Simo and M.S. Rifai. A class of mixed assumed strain methods and the method of incompatible modes. *Int. J. Num. Meth. Eng.*, **29**, 1595–638, 1990.
27. U. Andelfinger and E. Ramm. EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements. *Int. J. Num. Meth. Eng.*, **36**, 1311–37, 1993.
28. M. Bischoff, E. Ramm, and D. Braess. A class of equivalent enhanced assumed strain and hybrid stress finite elements. *Comp. Mech.*, **22**, 443–49, 1999.
29. G. Cantin, C. Loubignac, and C. Touzot. An iterative scheme to build continuous stress and displacement solutions. *Int. J. Num. Meth. Eng.*, **12**, 1493–506, 1978.
30. C. Loubibnac, G. Cantin, and C. Touzot. Continuous stress fields in finite element analysis. *J. AIAA*, **15**, 1645–47, 1978.

31. H.J. Brauchli and J.T. Oden. On the calculation of consistent stress distributions in finite element applications. *Int. J. Num. Meth. Eng.*, **3**, 317–25, 1971.
32. S. Nakazawa. Mixed finite elements and iterative solution procedures. In W.K. Liu *et al.*, editor, *Innovative Methods in Non-linear Problems*. Pineridge Press, Swansea, 1984.
33. O.C. Zienkiewicz, J.P. Vilotte, S. Toyoshima, and S. Nakazawa. Iterative method for constrained and mixed approximation. An inexpensive improvement of FEM performance. *Comp. Meth. Appl. Mech. Eng.*, **51**, 3–29, 1985.
34. O.C. Zienkiewicz, Xi Kui Li, and S. Nakazawa. Iterative solution of mixed problems and stress recovery procedures. *Comm. Appl. Num. Meth.*, **1**, 3–9, 1985.
35. K.J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Non-Linear Programming*. Stanford University Press, Stanford, CA, 1958.
36. S.P. Timoshenko and J.N. Goodier. *Theory of Elasticity*. 3rd edn, McGraw-Hill, New York, 1969.
37. B. Fraeijs de Veubeke and O.C. Zienkiewicz. Strain energy bounds in finite element analysis by slab analogy. *J. Strain Anal.*, **2**, 265–71, 1967.
38. Z.M. Elias. Duality in finite element methods. *Proc. Am. Soc. Civ. Eng.*, **94**(EM4), 931–46, 1968.
39. R.V. Southwell. On the analogues relating flexure and displacement of flat plates. *Quart. J. Mech. Appl. Math.*, **3**, 257–70, 1950.

12

Incompressible materials, mixed methods and other procedures of solution

12.1 Introduction

We have noted earlier that the standard displacement formulation of elastic problems fails when Poisson's ratio ν becomes 0.5 or when the material becomes incompressible. Indeed, problems arise even when the material is nearly incompressible with $\nu > 0.4$ and the simple linear approximation with triangular elements gives highly oscillatory results in such cases.

The application of a mixed formulation for such problems can avoid the difficulties and is of great practical interest as *nearly* incompressible behaviour is encountered in a variety of real engineering problems ranging from soil mechanics to aerospace engineering. Identical problems also arise when the flow of incompressible fluids is encountered.

In this chapter we shall discuss fully the mixed approaches to incompressible problems, generally using a two-field manner where displacement (or fluid velocity) $\mathbf{u}$ and the pressure p are the variables. Such formulation will allow us to deal with full incompressibility as well as near incompressibility as it occurs. However, what we will find is that the interpolations used will be very much limited by the stability conditions of the mixed patch test. For this reason much interest has been focused on the development of so-called *stabilized* procedures in which the violation of the mixed patch test (or Babuška–Brezzi conditions) is artificially compensated. A part of this chapter will be devoted to such stabilized methods.

12.2 Deviatoric stress and strain, pressure and volume change

The main problem in the application of a 'standard' displacement formulation to incompressible or nearly incompressible problems lies in the determination of the mean stress or pressure which is related to the volumetric part of the strain (for isotropic materials). For this reason it is convenient to separate this from the total stress field and treat it as an independent variable. Using the 'vector' notation of stress, the mean stress or pressure is given by

$$p = \tfrac{1}{3}\left(\sigma_x + \sigma_y + \sigma_z\right) = \tfrac{1}{3}\mathbf{m}^{\mathrm{T}}\boldsymbol{\sigma} \tag{12.1}$$

where $\mathbf{m}$ for the general three-dimensional state of stress is given by

$$\mathbf{m} = [\,1, \quad 1, \quad 1, \quad 0, \quad 0, \quad 0\,]^{\mathrm{T}}$$

For isotropic behaviour the 'pressure' is related to the volumetric strain, ε_v, by the bulk modulus of the material, K. Thus,

$$\varepsilon_v = \varepsilon_x + \varepsilon_y + \varepsilon_z = \mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon} \tag{12.2}$$

$$\varepsilon_v = \frac{p}{K} \tag{12.3}$$

For an incompressible material $K = \infty$ ($\nu \equiv 0.5$) and the volumetric strain is simply zero.

The deviatoric strain $\boldsymbol{\varepsilon}^d$ is defined by

$$\boldsymbol{\varepsilon}^d = \boldsymbol{\varepsilon} - \tfrac{1}{3}\mathbf{m}\varepsilon_v \equiv \left(\mathbf{I} - \tfrac{1}{3}\mathbf{m}\mathbf{m}^{\mathrm{T}}\right)\boldsymbol{\varepsilon} = \mathbf{I}_d\boldsymbol{\varepsilon} \tag{12.4}$$

where $\mathbf{I}_d$ is a deviatoric projection matrix which proves useful later and in Volume 2. In isotropic elasticity the deviatoric strain is related to the deviatoric stress by the shear modulus G as

$$\boldsymbol{\sigma}^d = \mathbf{I}_d\boldsymbol{\sigma} = 2G\mathbf{I}_0\boldsymbol{\varepsilon}^d = 2G\left(\mathbf{I}_0 - \tfrac{1}{3}\mathbf{m}\mathbf{m}^{\mathrm{T}}\right)\boldsymbol{\varepsilon} \tag{12.5}$$

where the diagonal matrix

$$\mathbf{I}_0 = \tfrac{1}{2}\begin{bmatrix} 2 & & & & & \\ & 2 & & & & \\ & & 2 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

is introduced because of the vector notation. A deviatoric form for the elastic moduli of an isotropic material is written as

$$\mathbf{D}_d = 2G\left(\mathbf{I}_0 - \tfrac{1}{3}\mathbf{m}\mathbf{m}^{\mathrm{T}}\right) \tag{12.6}$$

for convenience in writing subsequent equations.

The above relationships are but an alternate way of determining the stress strain relations shown in Chapters 2 and 4–6, with the material parameters related through

$$\begin{aligned} G &= \frac{E}{2(1+\nu)} \\ K &= \frac{E}{3(1-2\nu)} \end{aligned} \tag{12.7}$$

and indeed Eqs (12.5) and (12.3) can be used to define the standard $\mathbf{D}$ matrix in an alternative manner.

12.3 Two-field incompressible elasticity (u–*p* form)

In the mixed form considered next we shall use as variables the displacement $\mathbf{u}$ and the pressure p.

Now the equilibrium equation (11.22) is rewritten using (12.5), treating p as an independent variable, as

$$\int_\Omega \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{D}_d\boldsymbol{\varepsilon}\,\mathrm{d}\Omega + \int_\Omega \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{m}\,p\,\mathrm{d}\Omega - \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \tag{12.8}$$

and in addition we shall impose a weak form of Eq. (12.3), i.e.,

$$\int_\Omega \delta p\left[\mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon} - \frac{p}{K}\right]\mathrm{d}\Omega = 0 \tag{12.9}$$

with $\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u}$. Independent approximation of $\mathbf{u}$ and p as

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_u\tilde{\mathbf{u}} \qquad \text{and} \qquad p \approx \hat{p} = \mathbf{N}_p\tilde{\mathbf{p}} \tag{12.10}$$

immediately gives the mixed approximation in the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & -\mathbf{V} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \tag{12.11}$$

where

$$\begin{gathered} \mathbf{A} = \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}_d\,\mathbf{B}\,\mathrm{d}\Omega \qquad \mathbf{C} = \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{m}\mathbf{N}_p\,\mathrm{d}\Omega \\ \mathbf{V} = \int_\Omega \mathbf{N}_p^{\mathrm{T}}\frac{1}{K}\mathbf{N}_p\,\mathrm{d}\Omega \qquad \mathbf{f}_1 = \int_\Omega \mathbf{N}_u^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_t}\mathbf{N}_u^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \qquad \mathbf{f}_2 = \mathbf{0} \end{gathered} \tag{12.12}$$

We note that for incompressible situations the equations are of the 'standard' form, see Eq. (11.14) with $\mathbf{V} = \mathbf{0}$ (as $K = \infty$), but the formulation is useful in practice when K has a high value (or $\nu \to 0.5$).

A formulation similar to that above and using the corresponding variational theorem was first proposed by Herrmann[1] and later generalized by Key[2] for anisotropic

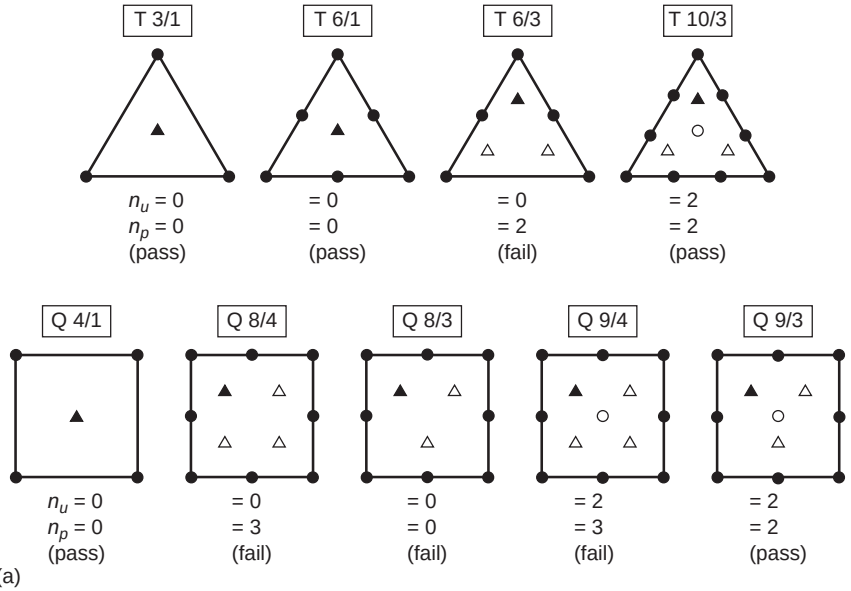


Fig. 12.1 Incompressible elasticity **u**–*p* formulation. Discontinuous pressure approximation. (a) Single-element patch tests.

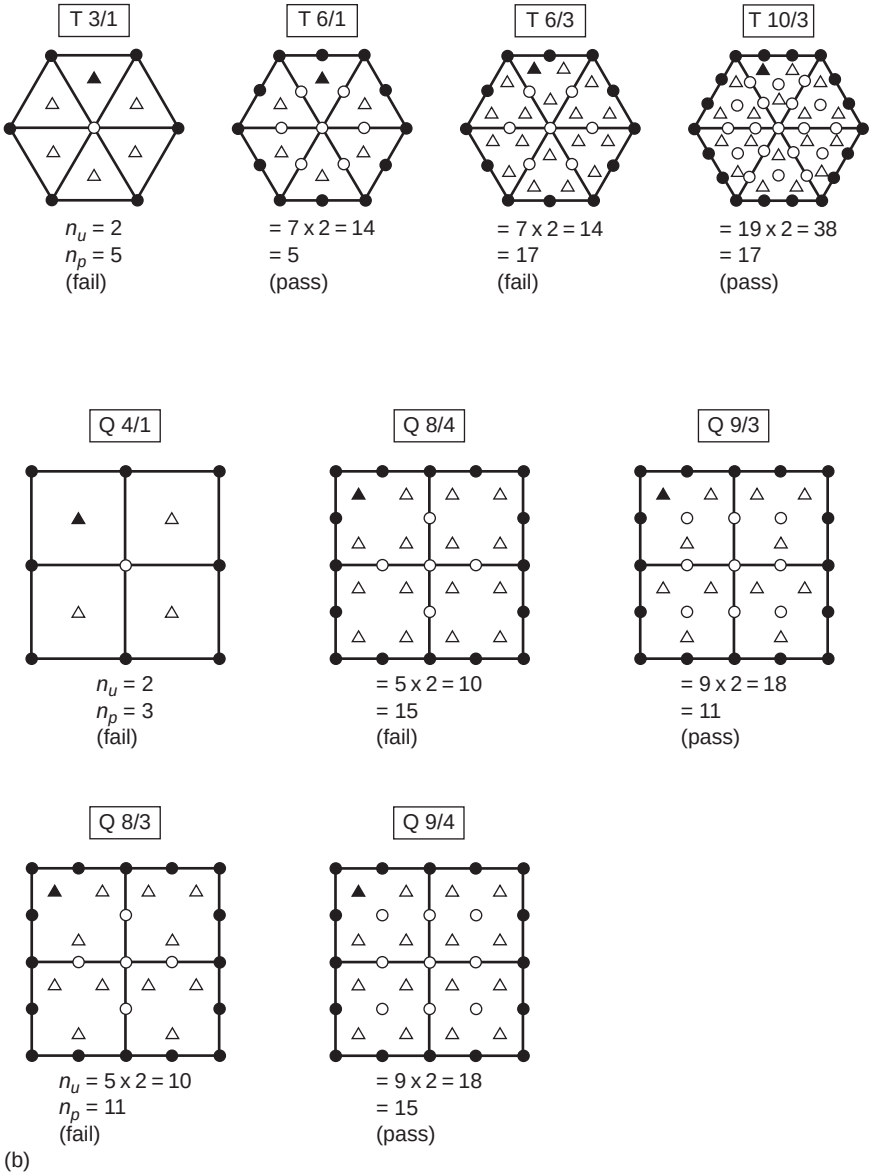


Fig. 12.1 (continued) Incompressible elasticity $\mathbf{u}$–p formulation. Discontinuous pressure approximation. (b) Multiple-element patch tests.

elasticity. The arguments concerning stability (or singularity) of the matrices which we presented in Sec. 11.3 are again of great importance in this problem.

Clearly the mixed patch condition about the number of degress of freedom now yields [see Eq. (11.18)]

$$n_u \geqslant n_p \tag{12.13}$$

and *is necessary* for prevention of locking (or instability) with the pressure acting now as the constraint variable of the lagrangian multiplier enforcing incompressibility.

In the form of a patch test this condition is most critical and we show in Figs 12.1 and 12.2 a series of such patch tests on elements with C_0 continuous interpolation of $\mathbf{u}$ and either discontinuous or continuous interpolation of p. For each we have included all combinations of constant, linear and quadratic functions.

In the test we prescribe *all* the displacements on the boundaries of the patch and one pressure variable (as it is well known that in fully incompressible situations pressure will be indeterminate by a constant for the problem with all boundary displacements prescribed).

The single-element test is very stringent and eliminates most continuous pressure approximations whose performance is known to be acceptable in many situations. For this reason we attach more importance to the assembly test and it would appear that the following elements could be permissible according to the criteria of Eq. (12.13) (indeed all pass the B-B condition fully):

Triangles: T6/1; T10/3; T6/C3

Quadrilaterals: Q9/3; Q8/C4; Q9/C4

We note, however, that in practical applications quite adequate answers have been reported with Q4/1, Q8/3 and Q9/4 quadrilaterals, although severe oscillations of p may occur. If full robustness is sought the choice of the elements is limited.[3]

It is unfortunate that in the present 'acceptable' list, the linear triangle and quadrilateral are missing. This appreciably restricts the use of these simplest elements. A possible and indeed effective procedure here is to not apply the pressure constraint at the level of a single element but on an assembly. This was done by Herrmann in his original presentation[1] where four elements were chosen for such a constraint as shown in Fig. 12.3(a). This composite 'element' passes the single-element (and multiple-element) patch tests but apparently so do several others fitting into this category. In Fig. 12.3(b) we show how a single triangle can be internally subdivided into three parts by the introduction of a central node. This coupled with constant pressure on the assembly allows the necessary count condition to be satisfied and a standard element procedure applies to the original triangle treating the central node as an internal variable. Indeed, the same effect could be achieved by the introduction of any other internal element function which gives zero value on the main triangle perimeter. Such a *bubble function* can simply be written in terms of the area coordinates (see Chapter 8) as

$$L_1 L_2 L_3$$

However, as we have stated before, the degree of freedom count is a necessary but not sufficient condition for stability and a direct rank test is always required. In particular it can be verified by algebra that the conditions stated in Sec. 11.3 are not fulfilled for this triple subdivision of a linear triangle (or the case with the bubble function) and thus

$$\mathbf{Cp} = \mathbf{0} \text{ for some non-zero values of } \mathbf{p}$$

indicating instability.

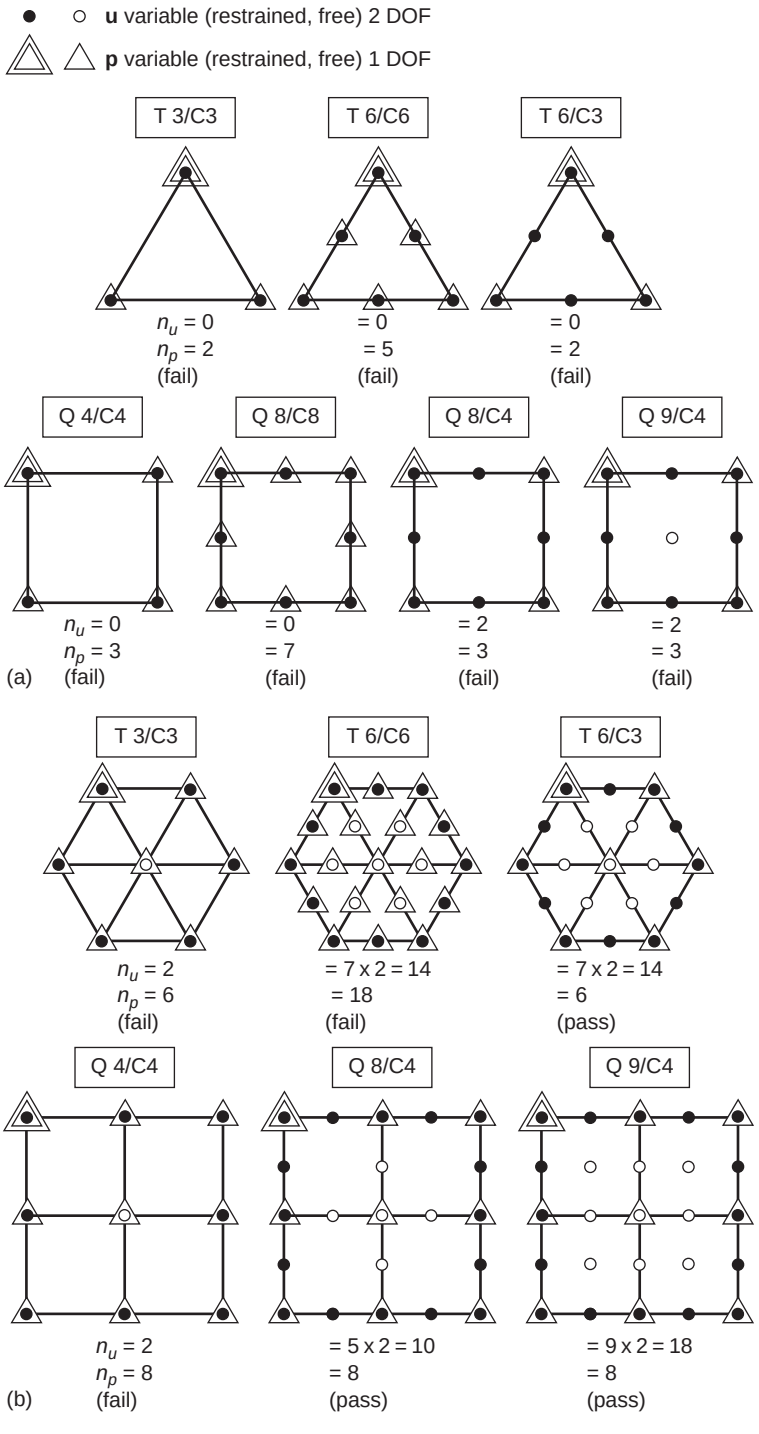


Fig. 12.2 Incompressible elasticity $\mathbf{u}$–p formulation. Continuous (C_0) pressure approximation. (a) Single-element patch tests. (b) Multiple-element patch tests.

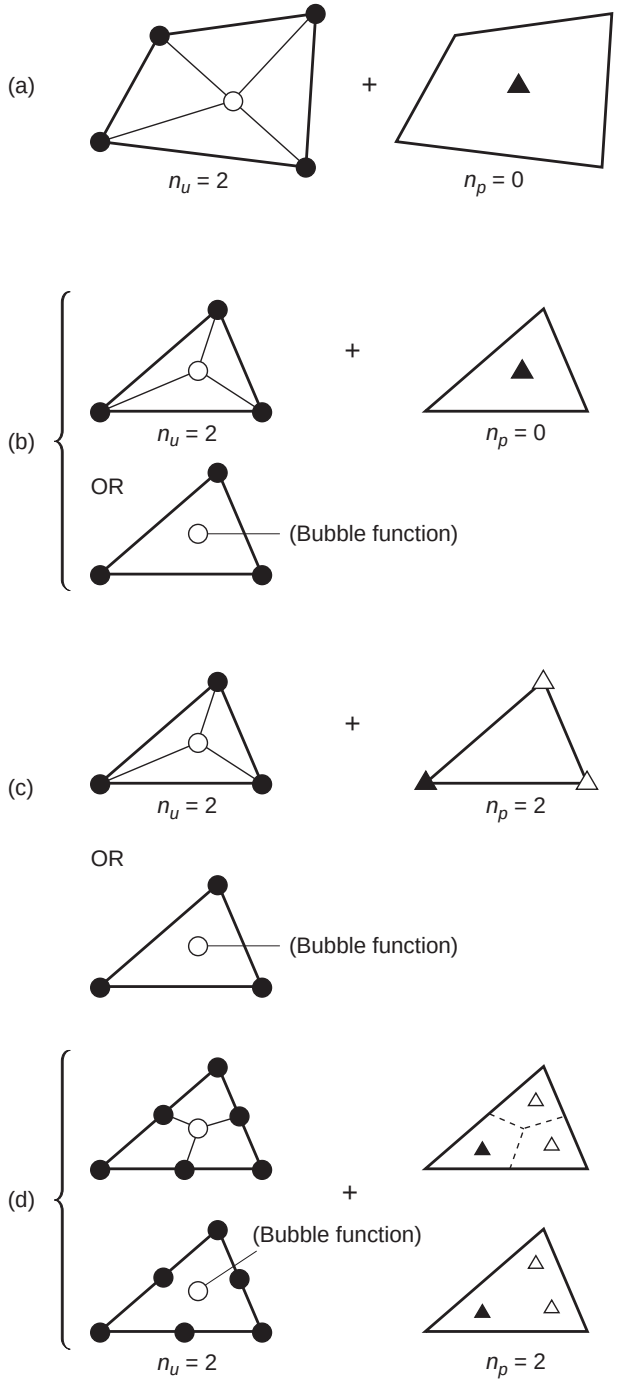


Fig. 12.3 Some simple combinations of linear triangles and quadrilaterals that pass the necessary patch test counts. Combinations (a), (c), and (d) are successful but (b) is still singular and not usable.

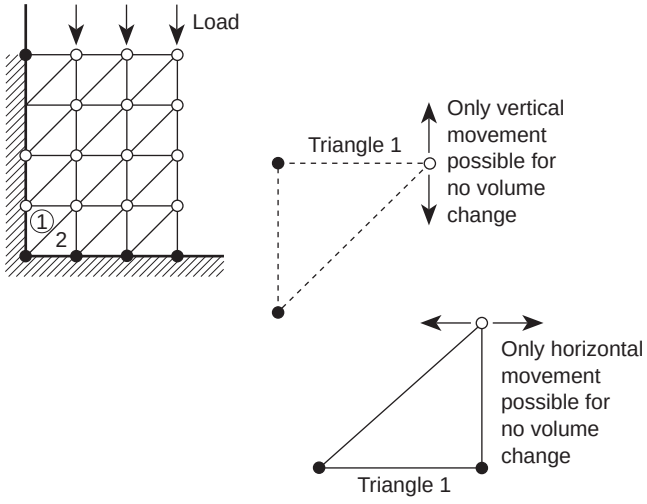


Fig. 12.4 Locking (zero displacements) of a simple assembly of linear triangles for which incompressibility is fully required ($n_p = n_u = 24$).

In Fig. 12.3(c) we show, however, that the same concept can be used with good effect for C_0 continuous p.[4] Similar internal subdivision into quadrilaterals or the introduction of bubble functions in quadratic triangles can be used, as shown in Fig. 12.3(d), with success.

The performance of all the elements mentioned above has been extensively discussed[5–10] but detailed comparative assessment of merit is difficult. As we have observed, it is essential to have $n_u \geqslant n_p$ but if near equality is only obtained in a large problem no meaningful answers will result for $\mathbf{u}$ as we observe, for example, in Fig. 12.4 in which linear triangles for $\mathbf{u}$ are used with the element constant p. Here the only permissible answer is of course $\mathbf{u} = \mathbf{0}$ as the triangles have to preserve constant volumes.

The ratio n_u/n_p which occurs as the field of elements is enlarged gives some indication of the relative performance, and we show this in Fig. 12.5. This approximates to the behaviour of a very large element assembly, but of course for any practical problem such a ratio will depend on the boundary conditions imposed.

We see that for the discontinuous pressure approximation this ratio for 'good' elements is 2–3 while for C_0 continuous pressure it is 6–8. All the elements shown in Fig. 12.5 perform very well, though two (Q4/1 and Q9/4) can on occasion lock when most boundary conditions are on $\mathbf{u}$.

12.4 Three-field nearly incompressible elasticity (u–p–ε_v form)

A direct approximation of the three-field form leads to an important method in finite element solution procedures for nearly incompressible materials which has sometimes been called the **B**-bar method. The methodology can be illustrated for the nearly

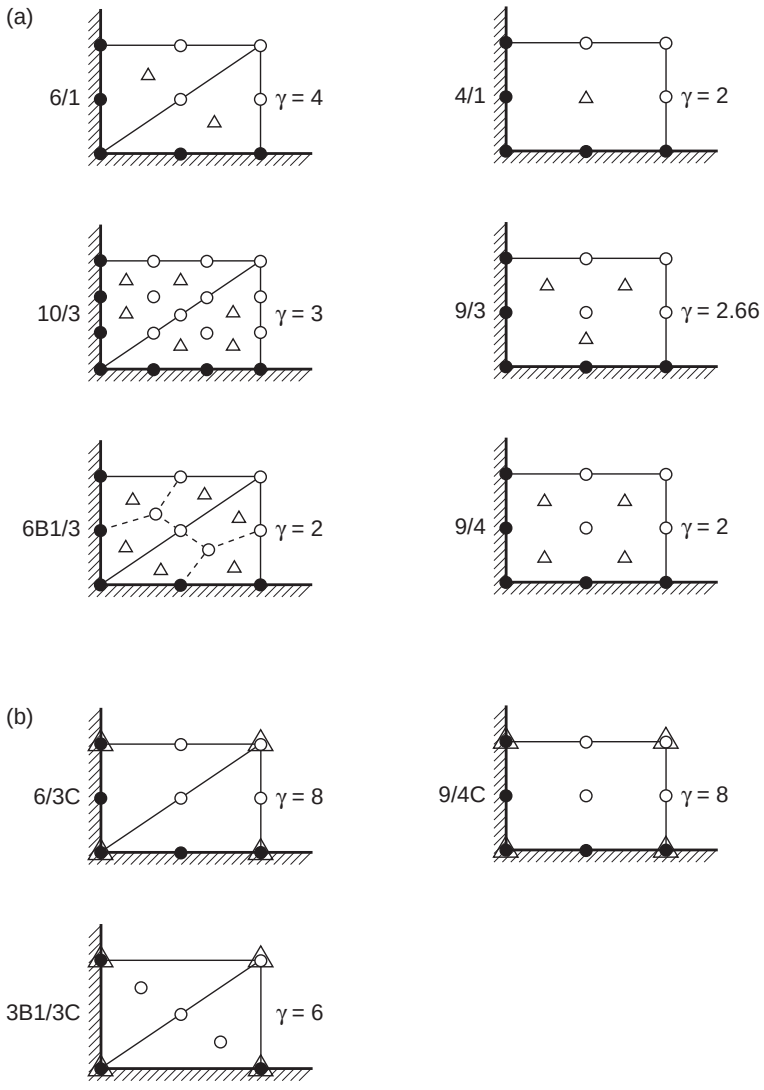


Fig. 12.5 The *freedom index* or *infinite patch ratio* for various **u**–p elements for incompressible elasticity ($\gamma = n_u/n_p$). (a) Discontinuous pressure. (b) Continuous pressure.

incompressible isotropic problem. For this problem the method often reduces to the same two-field form previously discussed. However, for more general anisotropic or inelastic materials and in finite deformation problems the method has distinct advantages as will be discussed further in Volume 2. The usual irreducible form (displacement method) has been shown to 'lock' for the nearly incompressible problem. As shown in Sec. 12.3, the use of a two-field mixed method can avoid this locking phenomenon when properly implemented (e.g., using the Q9/3 two-field form). Below we present an alternative which leads to an efficient and accurate implementation in many situations. For the development shown we shall assume

that the material is isotropic linear elastic but it may be extended easily to include anisotropic materials.

Assuming an independent approximation to ε_v and p we can formulate the problem by use of Eq. (12.8) and the weak statement of relations (12.2) and (12.3) written as

$$\int_\Omega \delta p \left[\mathbf{m}^{\mathrm{T}}\mathbf{S}\mathbf{u} - \varepsilon_v\right] \mathrm{d}\Omega = 0 \tag{12.14}$$

$$\int_\Omega \delta\varepsilon_v \left[K\varepsilon_v - p\right] \mathrm{d}\Omega = 0 \tag{12.15}$$

If we approximate the $\mathbf{u}$ and p fields by Eq. (12.10) and

$$\varepsilon_v \approx \hat{\varepsilon}_v = \mathbf{N}_v\tilde{\boldsymbol{\varepsilon}}_v \tag{12.16}$$

we obtain a mixed approximation in the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} & \mathbf{0} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} & -\mathbf{E} \\ \mathbf{0} & -\mathbf{E}^{\mathrm{T}} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \\ \tilde{\boldsymbol{\varepsilon}}_v \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{Bmatrix} \tag{12.17}$$

where $\mathbf{A}$, $\mathbf{C}$, $\mathbf{f}_1$, $\mathbf{f}_2$ are given by Eq. (12.12) and

$$\mathbf{E} = \int_\Omega \mathbf{N}_v^{\mathrm{T}}\mathbf{N}_p \,\mathrm{d}\Omega \qquad \mathbf{f}_3 = \mathbf{0} \tag{12.18}$$

with

$$\mathbf{H} = \int_\Omega \mathbf{N}_v^{\mathrm{T}} K \mathbf{N}_v \,\mathrm{d}\Omega \tag{12.19}$$

For completeness we give the variational theorem whose first variation gives Eqs (12.8), (12.14) and (12.15). First we define the strain deduced from the standard displacement approximation as

$$\boldsymbol{\varepsilon}_u = \mathbf{S}\mathbf{u} \approx \mathbf{B}\tilde{\mathbf{u}} \tag{12.20}$$

The variational theorem is then given as

$$\begin{aligned} \Pi = \frac{1}{2}\int_\Omega \left(\boldsymbol{\varepsilon}_u^{\mathrm{T}}\mathbf{D}_d\boldsymbol{\varepsilon}_u + \varepsilon_v K \varepsilon_v\right) \mathrm{d}\Omega + \int_\Omega p\left(\mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon}_u - \varepsilon_v\right) \mathrm{d}\Omega \\ - \int_\Omega \mathbf{u}^{\mathrm{T}}\mathbf{b} \,\mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}} \,\mathrm{d}\Gamma \end{aligned} \tag{12.21}$$

12.4.1 The B-bar method for nearly incompressible problems

The second of (12.17) has the solution

$$\tilde{\boldsymbol{\varepsilon}}_v = \mathbf{E}^{-1}\mathbf{C}^{\mathrm{T}}\tilde{\mathbf{u}} = \mathbf{W}\tilde{\mathbf{u}} \tag{12.22}$$

In the above we assume that $\mathbf{E}$ may be inverted, which implies that $\mathbf{N}_v$ and $\mathbf{N}_p$ have the same number of terms. Furthermore, the approximations for the volumetric strain and pressure are constructed for each element individually and are not continuous

across element boundaries. Thus, the solution of Eq. (12.22) may be performed for each individual element. In practice $\mathbf{N}_v$ is normally assumed identical to $\mathbf{N}_p$ so that $\mathbf{E}$ is symmetric positive definite. The solution of the third of (12.17) yields the pressure parameters in terms of the volumetric strain parameters and is given by

$$\tilde{\mathbf{p}} = \mathbf{E}^{-\mathrm{T}}\mathbf{H}^{\mathrm{T}}\tilde{\boldsymbol{\varepsilon}}_v \tag{12.23}$$

Substitution of (12.22) and (12.23) into the first of (12.17) gives a solution that is in terms of displacements only. Accordingly,

$$\bar{\mathbf{A}}\tilde{\mathbf{u}} = \mathbf{f}_1 \tag{12.24}$$

where for isotropy

$$\begin{aligned}\bar{\mathbf{A}} &= \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}_d\mathbf{B}\,\mathrm{d}\Omega + \mathbf{W}^{\mathrm{T}}\mathbf{H}\mathbf{W} \\ &= \mathbf{A} + \mathbf{W}^{\mathrm{T}}\mathbf{H}\mathbf{W}\end{aligned} \tag{12.25}$$

The solution of (12.24) yields the nodal parameters for the displacements. Use of (12.22) and (12.23) then gives the approximations for the volumetric strain and pressure.

The result given by (12.25) may be further modified to obtain a form that is similar to the standard displacement method. Accordingly, we write

$$\bar{\mathbf{A}} = \int_\Omega \bar{\mathbf{B}}^{\mathrm{T}}\mathbf{D}\bar{\mathbf{B}}\,\mathrm{d}\Omega \tag{12.26}$$

where the strain–displacement matrix is now

$$\bar{\mathbf{B}} = \mathbf{I}_d\mathbf{B} + \tfrac{1}{3}\mathbf{m}\mathbf{N}_v\mathbf{W} \tag{12.27}$$

For isotropy the modulus matrix is

$$\mathbf{D} = \mathbf{D}_d + K\mathbf{m}\mathbf{m}^{\mathrm{T}} \tag{12.28}$$

We note that the above form is identical to a standard displacement model except that $\mathbf{B}$ is replaced by $\bar{\mathbf{B}}$. The method has been discussed more extensively in references 11, 12 and 13.

The equivalence of (12.25) and (12.26) can be verified by simple matrix multiplication. Extension to treat general small strain formulations can be simply performed by replacing the isotropic $\mathbf{D}$ matrix by an appropriate form for the general material model. The formulation shown above has been implemented into an element included as part of the program referred to in Chapter 20. The elegance of the method is more fully utilized when considering non-linear problems, such as plasticity and finite deformation elasticity (see Volume 2).

We note that elimination starting with the third equation could be accomplished leading to a $\mathbf{u}$–p two-field form using K as a penalty number. This is convenient for the case where p is continuous but ε_v remains discontinuous – this is discussed further in Sec. 12.7.3. Such an elimination, however, points out that precisely the same stability criteria operate here as in the two-field approximation discussed earlier.

12.5 Reduced and selective integration and its equivalence to penalized mixed problems

In Chapter 9 we mentioned the lowest order numerical integration rules that still preserve the required convergence order for various elements, but at the same time pointed out the possibility of a singularity in the resulting element matrices. In Chapter 10 we again referred to such low order integration rules, introducing the name 'reduced integration' for those that did not evaluate the stiffness exactly for simple elements and pointed out some dangers of its indiscriminate use due to resulting instability. Nevertheless, such reduced integration and selective integration (where the low order approximation is only applied to certain parts of the matrix) has proved its worth in practice, often yielding much more accurate results than the use of more precise integration rules. This was particularly noticeable in nearly incompressible elasticity (or Stokes fluid flow which is similar)[14–16] and in problems of plate and shell flexure dealt with as a case of a degenerate solid[17,18] (see Volume 2).

The success of these procedures derived initially by heuristic arguments proved quite spectacular – though some consider it somewhat verging on immorality to obtain improved results while doing less work! Obviously fuller justification of such processes is required.[19] The main reason for success is associated with the fact that it provides the necessary singularity of the constraint part of the matrix [viz. Eqs (11.19)–(11.21)] which avoids locking. Such singularity can be deduced from a count of integration points,[19,20] but it is simpler to show that there is a complete equivalence between reduced (of selective) integration procedures and the mixed formulation already discussed. This equivalence was first shown by Malkus and Hughes[21] and later in a general context by Zienkiewicz and Nakazawa.[22]

We shall demonstrate this equivalence on the basis of the nearly incompressible elasticity problem for which the mixed weak integral statement is given by Eqs (12.8) and (12.9). It should be noted, however, that equivalence holds only for the discontinuous pressure approximation.

The corresponding irreducible form can be written by satisfying the second of these equations exactly by implying

$$p = K\mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon} \tag{12.29}$$

and substituting above into (12.8) as

$$\int_{\Omega} \delta\boldsymbol{\varepsilon}^{\mathrm{T}} 2G\left(\mathbf{I}_0 - \frac{1}{3}\mathbf{m}^{\mathrm{T}}\mathbf{m}\right)\boldsymbol{\varepsilon}\,\mathrm{d}\Omega + \int_{\Omega} \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{m}K\mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon}\,\mathrm{d}\Omega - \int_{\Omega} \mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \tag{12.30}$$

On substituting

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_u\tilde{\mathbf{u}} \qquad \text{and} \qquad \boldsymbol{\varepsilon} \approx \hat{\boldsymbol{\varepsilon}} = \mathbf{S}\mathbf{N}_u\tilde{\mathbf{u}} = \mathbf{B}\tilde{\mathbf{u}} \tag{12.31}$$

we have

$$\left(\mathbf{A} + \bar{\mathbf{A}}\right)\tilde{\mathbf{u}} = \mathbf{f}_1 \tag{12.32}$$

where $\mathbf{A}$ and $\mathbf{f}_1$ are exactly as given in Eq. (12.12) and

$$\bar{\mathbf{A}} = \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{m}K\mathbf{m}^{\mathrm{T}}\mathbf{B}\,\mathrm{d}\Omega \tag{12.33}$$

The solution of Eq. (12.32) for $\tilde{\mathbf{u}}$ allows the pressures to be determined at all points by Eq. (12.29). In particular, if we have used an integration scheme for evaluating (12.33) which samples at points (ξ_k) we can write

$$p(\xi_k) = K\mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon}(\xi_k) = K\mathbf{m}^{\mathrm{T}}\mathbf{B}(\xi_k)\tilde{\mathbf{u}} = \sum_j N_{pj}(\xi_k)\tilde{p}_j \tag{12.34}$$

Now if we turn our attention to the penalized mixed form of Eqs (12.8)–(12.12) we note that the second of Eqn. (12.11) is explicitly

$$\int_\Omega \mathbf{N}_p^{\mathrm{T}}\left(\mathbf{m}^{\mathrm{T}}\mathbf{B}\tilde{\mathbf{u}} - \frac{1}{K}\mathbf{N}_p\tilde{\mathbf{p}}\right)\mathrm{d}\Omega = \mathbf{0} \tag{12.35}$$

If a numerical integration is applied to the above sampling at the pressure nodes located at coordinate (ξ_l), previously defined in Eq. (12.34), we can write for each scalar component of $\mathbf{N}_p$

$$\sum_l N_{pj}(\xi_l)\left(\mathbf{m}^{\mathrm{T}}\mathbf{B}(\xi_l)\tilde{\mathbf{u}} - \frac{1}{K}\mathbf{N}_p(\xi_l)\tilde{\mathbf{p}}\right)W_l = 0 \tag{12.36}$$

in which the summation is over all integration points (ξ_l) and W_l are the appropriate weights and jacobian determinants.

Now as

$$N_{pj}(\xi_k) = \delta_{jk}$$

if ξ_l is at the pressure node j and zero at other pressure nodes, Eq. (12.36) reduces simply to the requirement that at all pressure nodes

$$\mathbf{m}^{\mathrm{T}}\mathbf{B}(\xi_l)\,\tilde{\mathbf{u}} = \frac{1}{K}\mathbf{N}_p(\xi_l)\tilde{\mathbf{p}} \tag{12.37}$$

This is precisely the same condition as that given by Eq. (12.34) and the equivalence of the procedures is proved, *providing the integrating scheme used for evaluating* $\bar{\mathbf{A}}$ *gives an identical integral of the mixed form of Eq. (12.35)*.

This is true in many cases and for these the reduced integration-mixed equivalence is exact. In all other cases this equivalence exists for a mixed problem in which an inexact rule of integration has been used in evaluating equations such as (12.35).

For curved isoparametric elements the equivalence is in fact inexact, and slightly different results can be obtained using reduced integration and mixed forms. This is illustrated in examples given in reference 23.

We can conclude without detailed proof that this type of equivalence is quite general and that with any problem of a similar type the application of numerical quadrature at n_p points in evaluating the matrix $\bar{\mathbf{A}}$ within each element is equivalent to a mixed problem in which the variable p is interpolated element-by-element using as p-nodal values the same integrating points.

The equivalence is only complete for the selective integration process, i.e., application of reduced numerical quadrature only to the matrix $\bar{\mathbf{A}}$, and ensures that this

matrix is singular, i.e., no locking occurs if we have satisfied the previously stated conditions ($n_u > n_p$).

The full use of reduced integration on the remainder of the matrix determining $\tilde{\mathbf{u}}$, i.e., $\mathbf{A}$, is only permissible if that remains non-singular – the case which we have discussed previously for the Q8/4 element.

It can therefore be concluded that all the elements with discontinuous interpolation of p which we have verified as applicable to the mixed problem (viz. Fig. 12.1, for instance) can be implemented for nearly incompressible situations by a penalized irreducible form using corresponding selective integration.†

In Fig. 12.6 we show an example which clearly indicates the improvement of displacements achieved by such reduced integration as the compressibility modulus K increases (or the Poisson ratio tends to 0.5). We note also in this example the dramatically improved performance of such points for stress sampling.

For problems in which the p (constraint) variable is continuously interpolated (C_0) the arguments given above fail as quantities such as $\mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon}$ are not interelement continuous in the irreducible form.

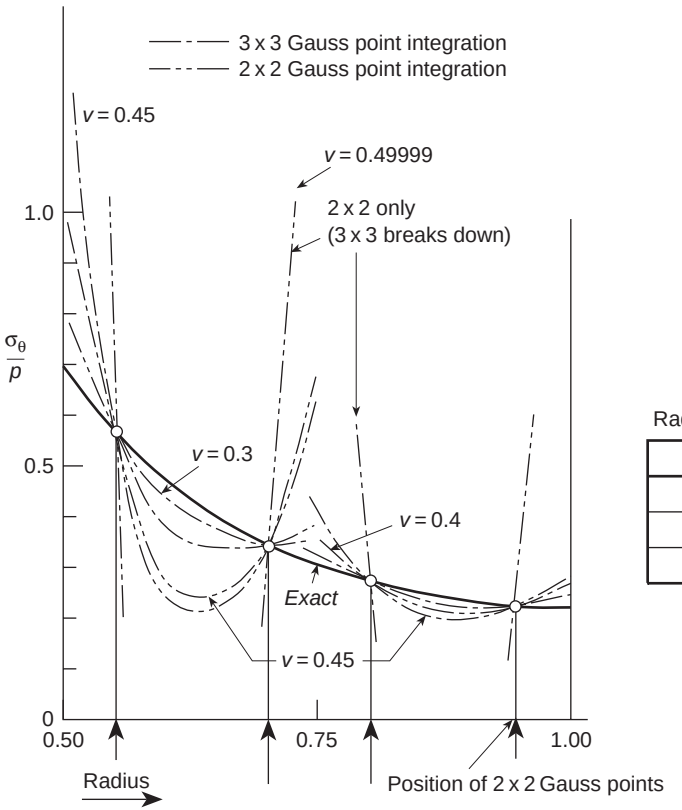
A very interesting corollary of the equivalence just proved for (nearly) incompressible behaviour is observed if we note the rapid increase of order of integrating formulae with the number of quadrature points (viz. Chapter 9). *For high order elements the number of quadrature points equivalent to the p constraint permissible for stability rapidly reaches that required for exact integration and hence their performance in nearly incompressible situations is excellent, even if exact integration is used*. This was observed on many occasions[24–26] and Sloan and Randolf[27] have shown good performance with the quintic triangle. Unfortunately such high order elements pose other difficulties and are seldom used in practice.

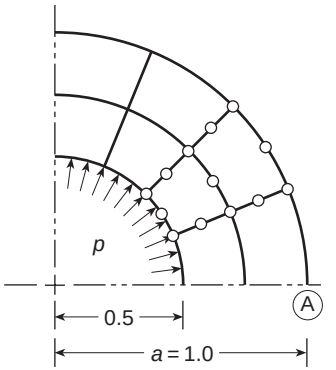
A final remark concerns the use of 'reduced' integration in particular and of penalized, mixed, methods in general. As we have pointed out in Sec. 11.3.1 it is possible in such forms to obtain sensible results for the *primary variable* (**u** in the present example) even though the general stability conditions are violated, providing some of the *constraint equations* are linearly dependent. Now of course the *constraint variable* (p in the present example) is not determinate in the limit.

This situation occurs with some elements that are occasionally used for the solution of incompressible problems but which do not pass our mixed patch test, such as Q8/4 and Q9/4 of Fig. 12.1. If we take the latter number to correspond to the integrating points these will yield acceptable **u** fields, though not p.

Figure 12.7 illustrates the point on an application involving slow viscous flow through an orifice – a problem that obeys identical equations to those of incompressible elasticity. Here elements Q8/4, Q8/3, Q9/4 and Q9/3 are compared although only the last completely satisfies the stability requirements of the mixed patch test. All elements are found to give a reasonable velocity (**u**) field but pressures are acceptable only for the last one, with element Q8/4 failing to give results which can be plotted.[3]

† The Q9/3 element would involve three-point quadrature which is somewhat unnatural for quadrilaterals. It is therefore better to simply use the mixed form here – and, indeed, in any problem which has non-linear behaviour between p and **u** (see Volume 2).





Radial displacements at (A) pa/E

Integ.	$v = 0.3$	$v = 0.4$	$v = 0.45$	$v = 0.49999$
3 x 3	0.3779	0.3904	0.3950	No result
2 x 2	0.3776	0.3910	0.3977	0.4041
Exact	0.3809	0.3945	0.4013	0.4081

Fig. 12.6 Sphere under internal pressure. Effect of numerical integration rules on results with different Poisson ratios.

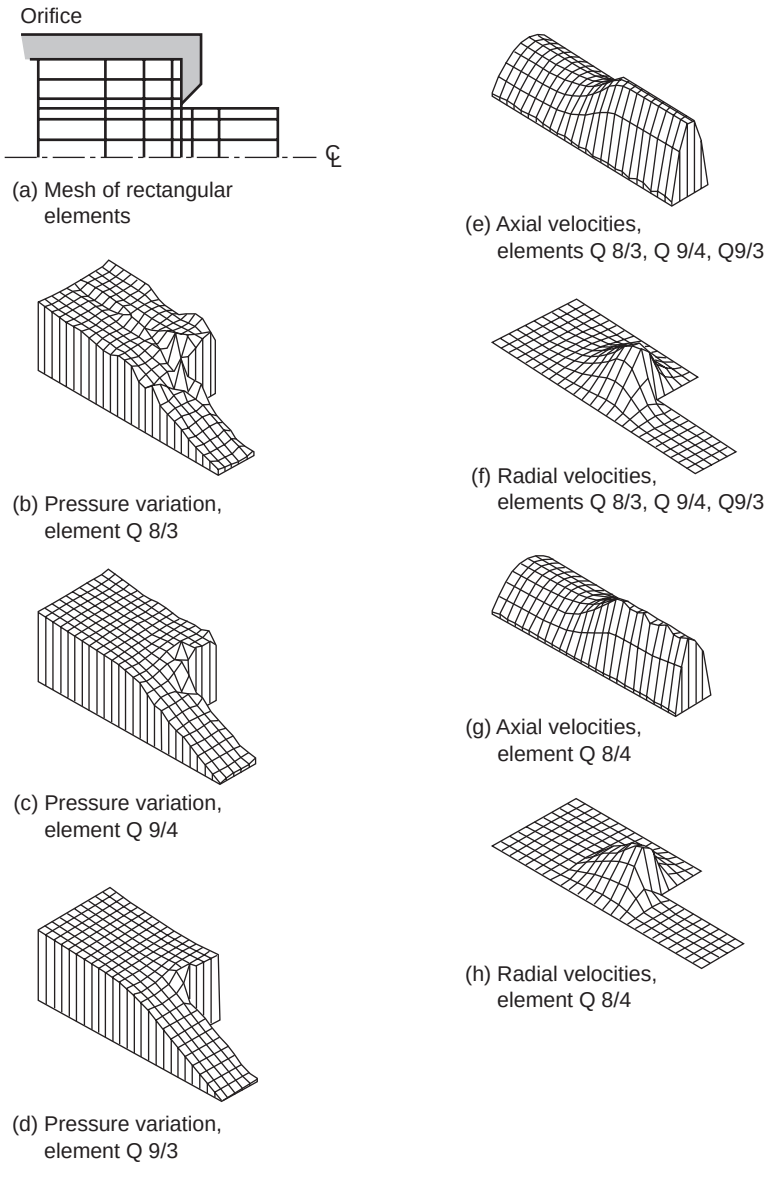


Fig. 12.7 Steady-state, low Reynolds number flow through an orifice. Note that pressure variation for element Q8/4 is so large it cannot be plotted. Solution with $\mathbf{u}/p$ elements Q8/3, Q8/4, Q9/3, Q9/4.

It is of passing interest to note that a similar situation develops if four triangles of the T3/1 type are assembled to form a quadrilateral in the manner of Fig. 12.8. Although the original element locks, as we have previously demonstrated, a linear dependence of the constraint equation allows the assembly to be used quite effectively in many incompressible situations, as shown in reference 25.

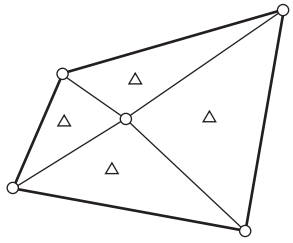


Fig. 12.8 A quadrilateral with intersecting diagonals forming an assembly of four T3/1 elements. This allows displacements to be determined for nearly incompressible behaviour but does not yield pressure results.

12.6 A simple iterative solution process for mixed problems: Uzawa method

12.6.1 General

In the general remarks on the algebraic solution of mixed problems characterized by equations of the type [viz. Eq. (11.14)]

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{y} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \tag{12.38}$$

we have remarked on the difficulties posed by the zero diagonal and the increased number of unknowns ($n_x + n_y$) as compared with the irreducible form (n_x or n_y).

A general iterative form of solution is possible, however, which substantially reduces the cost.[28] In this we solve successively

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \boldsymbol{\rho}\mathbf{r}^{(k)} \tag{12.39}$$

where $\mathbf{r}^{(k)}$ is the residual of the second equation computed as

$$\mathbf{r}^{(k)} = \mathbf{C}^{\mathrm{T}}\mathbf{x}^{(k)} - \mathbf{f}_2 \tag{12.40}$$

and follow with solution of the first equation, i.e.,

$$\mathbf{x}^{(k+1)} = \mathbf{A}^{-1}(\mathbf{f}_1 - \mathbf{C}\mathbf{y}^{(k+1)}) \tag{12.41}$$

In the above $\boldsymbol{\rho}$ is a 'convergence accelerator matrix' and is chosen to be efficient and simple to use.

The algorithm is similar to that described initially by Uzawa[29] and has been widely applied in an optimization context.[30–35]

Its relative simplicity can best be grasped when a particular example is considered.

12.6.2 Iterative solution for incompressible elasticity

In this case we start from Eq. (12.11) now written with $\mathbf{V} = \mathbf{0}$, i.e., complete incompressibility is assumed. The various matrices are defined in (12.12), resulting

in the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{0} \end{Bmatrix} \tag{12.42}$$

Now, however, for three-dimensional problems the matrix $\mathbf{A}$ is singular (as volumetric changes are not restrained) and it is necessary to *augment* it to make it non-singular. We can do this in the manner described in Sec. 11.3.1, or equivalently by the addition of a fictitious compressibility matrix, thus replacing $\mathbf{A}$ by

$$\bar{\mathbf{A}} = \mathbf{A} + \int_{\Omega} \mathbf{B}^{\mathrm{T}}(\lambda G \mathbf{m}\mathbf{m}^{\mathrm{T}})\mathbf{B}\,\mathrm{d}\Omega \tag{12.43}$$

If the second matrix uses an integration consistent with the number of discontinuous pressure parameters assumed, then this is precisely equivalent to writing

$$\bar{\mathbf{A}} = \mathbf{A} + \lambda G \mathbf{C}\mathbf{C}^{\mathrm{T}} \tag{12.44}$$

and is simpler to evaluate. Clearly this addition does not change the equation system.

The iteration of the algorithm (12.39)–(12.41) is now conveniently taken with the 'convergence accelerator' being simply defined as

$$\boldsymbol{\rho} = \lambda G \mathbf{I} \tag{12.45}$$

We now have the iterative system given as

$$\tilde{\mathbf{p}}^{(k+1)} = \tilde{\mathbf{p}}^{(k)} + \lambda G \mathbf{r}^{(k)} \tag{12.46}$$

where

$$\mathbf{r}^{(k)} = \mathbf{C}^{\mathrm{T}} \tilde{\mathbf{u}}^{(k)} \tag{12.47}$$

the residual of the incompressible constraint, and

$$\tilde{\mathbf{u}}^{(k+1)} = \bar{\mathbf{A}}^{-1}(\mathbf{f}_1 - \mathbf{C}\tilde{\mathbf{p}}^{(k+1)}) \tag{12.48}$$

In this $\bar{\mathbf{A}}$ can be interpreted as the stiffness matrix of a compressible material with bulk modulus $K = \lambda G$ and the process may be interpreted as the successive addition of volumetric 'initial' strains designed to reduce the volumetric strain to zero. Indeed this simple approach led to the first realization of this algorithm.[36–38] Alternatively the process can be visualized as an amendment of the original equation (12.42) by subtracting the term $\mathbf{p}/(\lambda G)$ from each side of the second to give (this is often called an *augmented lagrangian form*)[28,34]

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & -\dfrac{1}{\lambda G}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ -\dfrac{1}{\lambda, G}\tilde{\mathbf{p}} \end{Bmatrix} \tag{12.49}$$

and adopting the iteration

$$\begin{bmatrix} \bar{\mathbf{A}} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & -\dfrac{1}{\lambda G}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix}^{(k+1)} = \begin{Bmatrix} \mathbf{f}_1 \\ -\dfrac{1}{\lambda, G}\tilde{\mathbf{p}}^{(k)} \end{Bmatrix} \tag{12.50}$$

With this, on elimination, a sequence similar to Eqs (12.46)–(12.48) will be obtained provided $\bar{\mathbf{A}}$ is defined by Eq. (12.44).

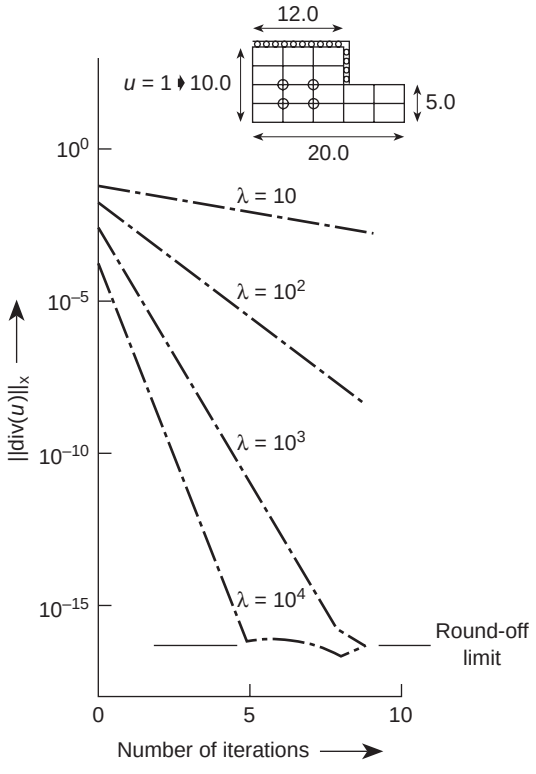


Fig. 12.9 Convergence of iterations in an extrusion problem for different values of the penalty ratio $\lambda = \gamma/\mu$.

Starting the iteration from

$$\tilde{\mathbf{u}}^{(0)} = \mathbf{0} \qquad \text{and} \qquad \tilde{\mathbf{p}}^{(0)} = \mathbf{0}$$

in Fig. 12.9 we show the convergence of the maximum $\operatorname{div}\mathbf{u}$ computed at any of the integrating points used. We note that this convergence becomes quite rapid for large values of $\lambda = (10^3 – 10^4)$.

For smaller λ values the process can be accelerated by using different $\boldsymbol{\rho}$[28] but for practical purposes the simple algorithm suffices for many problems, including applications in large strain.[39] Clearly much better satisfaction of the incompressibility constraint can now be obtained than by the simple use of a 'large enough' bulk modulus or penalty parameter. With $\lambda = 10^4$, for instance, in five iterations the initial $\operatorname{div}\mathbf{u}$ is reduced from the value $\sim 10^{-4}$ to 10^{-16}, which is at the round-off limit of the particular computer used.

The reader will note that the solution improvement strategy discussed in Sec. 11.6 is indeed a similar example of the above iteration process.

Finally, we remind the reader that the above iterative process solves the equations of a mixed problem. Accordingly, it is fully effective only when the element used satisfies the stability and consistency conditions of the mixed patch test.

12.7 Stabilized methods for some mixed elements failing the incompressibility patch test

12.7.1 Introduction

It has been observed earlier in this chapter that many of the two field $\mathbf{u}$–p elements do not pass the stability conditions imposed by the mixed patch test at the incompressible limit(or the Babuška–Brezzi conditions). Here in particular we have such methods in which the displacement and pressure are interpolated in an identical manner (for instance, linear triangles, linear quadrilaterals, quadratic triangles, etc.) and many attempts for *stabilization* of such elements have been introduced.

The most obvious stabilized element can be directly achieved from the formulation suggested in Fig. 12.3(b) of the triangle with a displacement *bubble* introduced. If this internal displacement is eliminated, then we have a stable element which has a triangular shape with linear displacement and pressure interpolations from nodal values. However, alternatives to this exist and these form several categories. The first category is the introduction of non-zero diagonal terms by adding a least-square form to the Galerkin formulation. This was first suggested by Courant[40] and it appears that Brezzi and Pitkaranta in 1984[41] have produced an element of this kind. Numerous further suggestions have been proposed by Hughes *et al.* between 1986 and 1989.[42–44] More recently, an alternative proposal of achieving similar answers has been proposed by Oñate[45] which gains the addition of diagonal terms by the introduction of so-called *finite increment calculus* to the formulation.

There is, however, an alternative possibility introduced by time integration of the full incompressible formulation. Here many of the algorithms will yield, when steady-state conditions are recovered, a stabilized form. A number of such algorithms have been discussed by Zienkiewicz and Wu in 1991[46] and more recently a very efficient method has appeared as a by-product of a fluid mechanics algorithm named the *characteristic based split* (CBS) procedure[47–50] which will be discussed at length in Volume 3.

In the latter algorithm there exists a free parameter. This parameter depends on the size of the time increment. In the other methods (with the exception of the bubble formulation) there is a weighting parameter applied to the additional terms introduced. We shall discuss each of these algorithms in the following subsections and compare the numerical results obtainable.

One may question, perhaps, that resort to stabilization procedures is not worthwhile in view of the relative simplicity of the full mixed form. But this is a matter practice will decide and is clearly in the hands of the analyst applying the necessary solutions.

12.7.2 Simple triangle with bubble eliminated

In Fig. 12.3(c) we indicated that the simple triangle with C_0 linear interpolation and an added bubble for the displacements $\mathbf{u}$ together with continuous C_0 linear

interpolation for the pressure p satisfied the count test part of the mixed patch test and can be used with success. Here we consider this element further to develop some understanding about its performance at the incompressible limit.

The displacement field with the bubble is written as

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_i N_i \tilde{\mathbf{u}}_i + N_b \tilde{\mathbf{u}}_b \tag{12.51}$$

where here

$$N_b = L_1 L_2 L_3 \tag{12.52}$$

$\tilde{\mathbf{u}}_i$ are nodal parameters of displacement and $\tilde{\mathbf{u}}_b$ are parameters of the hierarchical bubble function. The pressures are similarly given by

$$p \approx \hat{p} = \sum_i N_i \tilde{p}_i \tag{12.53}$$

where $\tilde{p}_i$ are nodal parameters of the pressure. In the above the shape functions are given by (e.g., see Eq. (8.34) and (8.32))

$$N_i = L_i = \frac{1}{2\Delta}(a_i + b_i x + c_i y) \tag{12.54}$$

where

$$a_i = x_j y_k - x_k y_j; \qquad b_i = y_j - y_k; \qquad c_i = x_k - y_j$$

j, k are cyclic permutations of i and

$$2\Delta = \det \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = a_1 + a_2 + a_3$$

The derivatives of the shape functions are thus given by

$$\frac{\partial N_i}{\partial x} = \frac{b_i}{2\Delta} \qquad \text{and} \qquad \frac{\partial N_i}{\partial y} = \frac{c_i}{2\Delta}$$

Similarly the derivatives of the bubble are given by

$$\frac{\partial N_b}{\partial x} = \frac{1}{2\Delta}(b_1 L_2 L_3 + b_2 L_3 L_1 + b_3 L_1 L_2)$$
$$\frac{\partial N_b}{\partial y} = \frac{1}{2\Delta}(c_1 L_2 L_3 + c_2 L_3 L_1 + c_3 L_1 L_2)$$

The strains may be expressed in terms of the above and the nodal parameters as†

$$\boldsymbol{\varepsilon} = \sum_i \frac{1}{2\Delta} \begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix} \tilde{\mathbf{u}}_i + \sum_i \frac{L_j L_k}{2\Delta} \begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix} \tilde{\mathbf{u}}_b \tag{12.55}$$

where again j, k are cyclic permutations of i.

† At this point it is also possible to consider the term added to the derivatives to be *enhanced modes* and delete the bubble mode from displacement terms.

Substituting the above strains into Eq. (12.12) and evaluating the integrals give

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{0} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{bb} \end{bmatrix} \tag{12.56}$$

where

$$\mathbf{A}_{ij} = \frac{G}{6\Delta} \begin{bmatrix} (4b_i b_j + 3c_i c_j) & (3c_i b_j - 2b_i c_j) \\ (3b_i c_j - 2c_i b_j) & (3b_i b_j + 4c_i c_j) \end{bmatrix}$$

$$\mathbf{A}_{bb} = \frac{G}{2160\Delta} \begin{bmatrix} (4\mathbf{b}^{\mathrm{T}}\mathbf{b} + 3\mathbf{c}^{\mathrm{T}}\mathbf{c}) & \mathbf{b}^{\mathrm{T}}\mathbf{c} \\ \mathbf{b}^{\mathrm{T}}\mathbf{c} & (3\mathbf{b}^{\mathrm{T}}\mathbf{b} + 4\mathbf{c}^{\mathrm{T}}\mathbf{c}) \end{bmatrix}$$

and

$$\mathbf{b} = [b_1, \quad b_2, \quad b_3] \qquad \text{and} \qquad \mathbf{c} = [c_1, \quad c_2, \quad c_3]$$

Note in the above that all terms except $\mathbf{A}_{bb}$ are standard displacement stiffnesses for the deviatoric part. Similarly,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23} \\ \mathbf{C}_{31} & \mathbf{C}_{32} & \mathbf{C}_{33} \\ \mathbf{C}_{b1} & \mathbf{C}_{b2} & \mathbf{C}_{b3} \end{bmatrix} \tag{12.57}$$

where

$$\mathbf{C}_{ij} = \frac{1}{6} \begin{bmatrix} b_j \\ c_j \end{bmatrix} \qquad \text{and} \qquad \mathbf{C}_{bj} = -\frac{1}{120} \begin{bmatrix} b_j \\ c_j \end{bmatrix}$$

In all the above arrays i and j have values from 1 to 3 and b denotes the bubble mode.

We note that the bubble mode is decoupled from the other entries in the $\mathbf{A}$ array – it is precisely for this reason that the discontinuous constant pressure case shown in Fig. 12.3(b) cannot be improved by the addition of the internal parameters associated with $\tilde{\mathbf{u}}_b$. Also, the parameters $\tilde{\mathbf{u}}_b$ are defined separately for each element. Consequently, we may perform a partial solution at the element level to obtain the set of equations in the form Eq. (12.11) where now

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix}; \qquad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23} \\ \mathbf{C}_{31} & \mathbf{C}_{32} & \mathbf{C}_{33} \end{bmatrix}; \qquad \mathbf{V} = \begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \end{bmatrix}$$

with

$$V_{ij} = \begin{bmatrix} \dfrac{b_i}{2\Delta} & \dfrac{c_i}{2\Delta} \end{bmatrix} \begin{bmatrix} \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \end{bmatrix} \begin{Bmatrix} \dfrac{b_i}{2\Delta} \\ \dfrac{c_i}{2\Delta} \end{Bmatrix} \Delta \tag{12.58}$$

and

$$\boldsymbol{\tau} = \frac{3\Delta^2}{10Gc} \begin{bmatrix} (3\mathbf{b}^{\mathrm{T}}\mathbf{b} + 4\mathbf{c}^{\mathrm{T}}\mathbf{c}) & -\mathbf{b}^{\mathrm{T}}\mathbf{c} \\ -\mathbf{b}^{\mathrm{T}}\mathbf{c} & (4\mathbf{b}^{\mathrm{T}}\mathbf{b} + 3\mathbf{c}^{\mathrm{T}}\mathbf{c}) \end{bmatrix} \tag{12.59}$$

in which

$$c = 12\left(\mathbf{b}^{\mathrm{T}}\mathbf{b}\right)^2 + 25\left(\mathbf{b}^{\mathrm{T}}\mathbf{b}\right)\left(\mathbf{c}^{\mathrm{T}}\mathbf{c}\right) + 12\left(\mathbf{c}^{\mathrm{T}}\mathbf{c}\right)^2 - \left(\mathbf{b}^{\mathrm{T}}\mathbf{c}\right)^2$$

The reader may recognize the $\mathbf{V}$ array given above as that for the two-dimensional, steady heat equation with conductivity $\mathbf{k} = \boldsymbol{\tau}$ and discretized by linear triangular elements. The direct reduction of the bubble matrix $\mathbf{A}_{bb}$ as given above leads to an anisotropic stabilization matrix $\boldsymbol{\tau}$. A diagonal form of the stabilization results if the weak form for the bubble terms is given by expressing the equilibrium equation in terms of the laplacian of each displacement component and the gradient of the pressure. This is permitted only for bubble terms which vanish identically on the boundary of each element. In Sec. 12.7.4 we indicate how such a reduction could be performed and leave as an exercise to the reader the construction of the weak form terms and the resulting diagonal matrix $\mathbf{A}_{bb}$. Numerical experiments indicate that very little difference is achieved between the two approaches. Since the construction of the diagonal form requires substitution of the constitutive equations into the equilibrium equation it is very limited in the type of applications which can be pursued (e.g., consideration of non-linear problems will preclude such simple substitution).

12.7.3 An enhanced strain stabilization

In the previous section we presented a simple two-field formulation using continuous $\mathbf{u}$ and p approximations together with added bubble modes to the displacements. For more general applications this form is not the most convenient. For example, if transient problems are considered the accelerations will also involve the bubble mode and affect the inertial terms. We will also find in the comparisons section that use of the above bubble is not fully effective in eliminating pressure oscillations in solutions. An alternative form is discussed in this section. In the alternative form we use a three-field approximation involving $\mathbf{u}$, p and ε_v discussed in Sec. 12.4 together with an enhanced strain formulation as discussed in Sec. 11.5.3.

The enhanced strains are added to those computed from displacements as

$$\breve{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}_u + \boldsymbol{\varepsilon}_e \tag{12.60}$$

in which $\boldsymbol{\varepsilon}_e$ represents a set of enhanced strain terms. The internal strain energy is represented by

$$W(\breve{\boldsymbol{\varepsilon}}, \varepsilon_v) = \tfrac{1}{2}\left(\breve{\boldsymbol{\varepsilon}}^{\mathrm{T}}\mathbf{D}_d\breve{\boldsymbol{\varepsilon}} + \varepsilon_v K \varepsilon_v\right) \tag{12.61}$$

Using the above notation a Hu–Washizu type variational theorem for the deviatoric-spherical split may be written as

$$\Pi_{me} = \int_\Omega \left[W(\breve{\boldsymbol{\varepsilon}}, \varepsilon_v) + p\left(\mathbf{m}^{\mathrm{T}}\breve{\boldsymbol{\varepsilon}} - \varepsilon_v\right) + \boldsymbol{\sigma}^{\mathrm{T}}(\boldsymbol{\varepsilon}_u - \breve{\boldsymbol{\varepsilon}})\right] \mathrm{d}\Omega + \Pi_{ext} \tag{12.62}$$

where Π_{ext} represents the terms associated with body and traction forces.

After substitution for the mixed enhanced strain the last term in the integral simplifies as:

$$\int_\Omega \boldsymbol{\sigma}^{\mathrm{T}}(\boldsymbol{\varepsilon}_u - \breve{\boldsymbol{\varepsilon}})\,\mathrm{d}\Omega = -\int_\Omega \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon}_e\,\mathrm{d}\Omega \tag{12.63}$$

Taking variations with respect to $\mathbf{u}$, p, ε_v, $\boldsymbol{\varepsilon}_e$ and $\boldsymbol{\sigma}$ yields

$$\begin{aligned}\delta\Pi_{me} = &\int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}[\mathbf{D}_d\breve{\boldsymbol{\varepsilon}} + \mathbf{m}p]\,\mathrm{d}\Omega + \delta\Pi_{ext}\\ &+ \int_\Omega \delta\varepsilon_v[K\varepsilon_v - p]\,\mathrm{d}\Omega + \int_\Omega \delta p\left[\mathbf{m}^{\mathrm{T}}\breve{\boldsymbol{\varepsilon}} - \varepsilon_v\right]\mathrm{d}\Omega\\ &+ \int_\Omega \delta\boldsymbol{\varepsilon}_e^{\mathrm{T}}[\mathbf{D}_d\breve{\boldsymbol{\varepsilon}} + \mathbf{m}p - \boldsymbol{\sigma}]\,\mathrm{d}\Omega + \int_\Omega \delta\boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\varepsilon}_e\,\mathrm{d}\Omega = 0\end{aligned} \tag{12.64}$$

Equal order interpolation with shape functions $\mathbf{N}$ are used to approximate $\mathbf{u}$, p and ε_v as

$$\begin{aligned}\mathbf{u} &\approx \hat{\mathbf{u}} = \mathbf{N}\tilde{\mathbf{u}}\\ p &\approx \hat{p} = \mathbf{N}\tilde{\mathbf{p}}\\ \varepsilon_v &\approx \hat{\varepsilon}_v = \mathbf{N}\tilde{\boldsymbol{\varepsilon}}_v\end{aligned} \tag{12.65}$$

However, only approximations for $\mathbf{u}$ and p are C_0 continuous between elements. The approximation for ε_v may be discontinuous between elements. The stress $\boldsymbol{\sigma}$ in each element is assumed constant. Thus, only the approximation for $\boldsymbol{\varepsilon}_e$ remains to be constructed in such a way that Eq. (11.49) is satisfied. For the present we shall assume that this approximation may be represented by

$$\boldsymbol{\varepsilon}_e \approx \hat{\boldsymbol{\varepsilon}}_e = \mathbf{B}_e\tilde{\boldsymbol{\alpha}}_e \tag{12.66}$$

and will satisfy Eq. (11.49) so that the terms involving $\boldsymbol{\sigma}$ and its variation in Eq. (12.64) are zero and thus do not appear in the final discrete equations.

With the above approximations, Eq. (12.63) may be evaluated as

$$\begin{bmatrix} \mathbf{A}_{uu} & \mathbf{A}_{ue} & \mathbf{C}_u & \mathbf{0} \\ \mathbf{A}_{eu} & \mathbf{A}_{ee} & \mathbf{C}_e & \mathbf{0} \\ \mathbf{C}_u^{\mathrm{T}} & \mathbf{C}_e^{\mathrm{T}} & \mathbf{0} & -\mathbf{E} \\ \mathbf{0} & \mathbf{0} & -\mathbf{E}^{\mathrm{T}} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\boldsymbol{\alpha}}_e \\ \tilde{\mathbf{p}} \\ \tilde{\boldsymbol{\varepsilon}}_v \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{0} \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{Bmatrix} \tag{12.67}$$

where $\mathbf{A}_{uu} = \mathbf{A}$, $\mathbf{C}_u = \mathbf{C}$, $\mathbf{f}_i$, $\mathbf{E}$ and $\mathbf{H}$ are as defined in Eqs (12.12), (12.18) and (12.19) and

$$\begin{aligned}\mathbf{A}_{ue} &= \int_\Omega \mathbf{B}\mathbf{D}_d\mathbf{B}_e\,\mathrm{d}\Omega = \mathbf{A}_{eu}^{\mathrm{T}}\\ \mathbf{A}_{ee} &= \int_\Omega \mathbf{B}_e\mathbf{D}_d\mathbf{B}_e\,\mathrm{d}\Omega\\ \mathbf{C}_e &= \int_\Omega \mathbf{B}_e\mathbf{m}\mathbf{N}\,\mathrm{d}\Omega\end{aligned} \tag{12.68}$$

Since the approximations for ε_v and $\boldsymbol{\varepsilon}_e$ are discontinuous between elements we can again perform a partial solution for $\tilde{\boldsymbol{\varepsilon}}_v$ and $\tilde{\boldsymbol{\alpha}}_e$ using the second and fourth row of (12.67). After eliminating these variables from the first and third equation we again, as in the simple triangle with bubble eliminated, obtain a form identical to Eq. (12.11).

As an example we consider again the three-noded triangular element with linear approximations for $\mathbf{N}$ in terms of area coordinates L_i. We will construct enhanced strain terms from the derivatives of a function. The simplest such approximation is the *bubble mode* used in Sec. 12.7.2 where the function is given as

$$N_e(\boldsymbol{\xi}) = L_1 L_2 L_3 \tag{12.69}$$

and the enhanced strain part is given by

$$\boldsymbol{\varepsilon}_e(L_i) = \mathbf{B}_e(L_i)\tilde{\boldsymbol{\alpha}}_e \tag{12.70}$$

where $\tilde{\boldsymbol{\alpha}}_e$ are two enhanced strain parameters and $\mathbf{B}_e$ is computed using Eq. (12.69) in the usual strain–displacement matrix

$$\mathbf{B}_e = \begin{bmatrix} \dfrac{\partial \mathbf{N}_e}{\partial x} & 0 \\ 0 & \dfrac{\partial \mathbf{N}_e}{\partial y} \\ \dfrac{\partial \mathbf{N}_e}{\partial y} & \dfrac{\partial \mathbf{N}_e}{\partial x} \end{bmatrix} \tag{12.71}$$

The result using Eq. (12.69) is identical to the bubble mode since here we are only considering static problems in the absence of body loads. If we considered the transient case or added body loads there would be a difference since the displacement in the enhanced form contains only the linear interpolations in $\mathbf{N}$.

While this is an admissible form we have noted above that it does not eliminate all oscillations for problems where strong pressure gradients occur. Accordingly, we also consider here an alternative form resulting from three enhanced functions

$$N_e^i = aL_i + L_j L_k \tag{12.72}$$

in which i, j, k is a cyclic permutation and a is a parameter to be determined. Note that this form only involves quadratic terms and thus gives linear strains which are fully consistent with the linear interpolations for p and θ. The derivatives of the enhanced function are given by

$$\begin{aligned} \frac{\partial N_e^i}{\partial x} &= \frac{1}{2\Delta}\left[ab_i + L_j b_k + L_k b_j\right] \\ \frac{\partial N_e^i}{\partial y} &= \frac{1}{2\Delta}\left[ac_i + L_j c_k + L_k c_j\right] \end{aligned} \tag{12.73}$$

where

$$b_i = y_j - y_k \qquad \text{and} \qquad c_i = x_k - x_j$$

and Δ is the area of a triangular element. The requirement imposed by Eq. 11.49 gives $a = 1/3$.

While the use of added enhanced modes leads to increased cost in eliminating the $\tilde{\boldsymbol{\varepsilon}}_v$ and $\boldsymbol{\alpha}_e$ parameters in Eq. (12.67) the results obtained are free of pressure oscillations in the problems considered in Sec. 12.7.7. Furthermore, this form leads to improved consistency between the pressure and strain.

12.7.4 A pressure stabilization

In the first part of this chapter we separated the stress into the deviatoric and pressure components as

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^d + \mathbf{m}p$$

Using the tensor form described in Appendix B this may be written in index form as

$$\sigma_{ij} = \sigma_{ij}^d + \delta_{ij}p$$

The deviatoric stresses are related to the deviatoric strains through the relation

$$\sigma_{ij}^d = 2G\varepsilon_{ij}^d = G\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{12.74}$$

The equilibrium equations (in the absence of inertial forces) are:

$$\frac{\partial \sigma_{ij}^d}{\partial x_i} + \frac{\partial p}{\partial x_j} + b_j = 0$$

Substituting the constitutive equations for the deviatoric part yields the equilibrium form (assuming G is constant)

$$G\left[\frac{\partial^2 u_j}{\partial x_i \partial x_i} + \frac{1}{3}\frac{\partial^2 u_i}{\partial x_i \partial x_j}\right] + \frac{\partial p}{\partial x_j} + b_j = 0 \tag{12.75}$$

In intrinsic form this is given as

$$G[\nabla^2 \mathbf{u} + \tfrac{1}{3}\boldsymbol{\nabla}(\operatorname{div}\mathbf{u})] + \boldsymbol{\nabla}p + \mathbf{b} = \mathbf{0}$$

where ∇^2 is the laplacian operator and $\boldsymbol{\nabla}$ the gradient operator. The constitutive equation (12.2) is expressed in terms of the displacement as

$$\varepsilon_v = \frac{\partial u_i}{\partial x_i} = \operatorname{div}\mathbf{u} = \frac{1}{K}p \tag{12.76}$$

where $\operatorname{div}(\cdot)$ is the divergence of the quantity. A single equation for pressure may be deduced from the divergence of the equilibrium equation. Accordingly, from Eq. (12.75) we obtain

$$\frac{4G}{3}\nabla^2(\operatorname{div}\mathbf{u}) + \nabla^2 p + \operatorname{div}\mathbf{b} = 0 \tag{12.77}$$

Upon noting (12.76) we obtain

$$\left(1 + \frac{4G}{3K}\right)\nabla^2 p + \operatorname{div}\mathbf{b} = 0 \tag{12.78}$$

Thus, in general, the pressure must satisfy a Poisson equation, or in the absence of body forces, a Laplace equation. We have noted the dangers of artificially raising the order of the differential equation in introducing spurious solutions, however, in the context of constructing approximate solutions to the incompressible problem the above is useful in providing additional terms to the weak form which otherwise would be zero. Brezzi and Pitkaranta[41] suggested adding a weighted Eq. (12.78) to Eq. (12.8) and (on setting the body force to zero for simplicity) obtain

$$\int_\Omega \delta p \left(\mathbf{m}^{\mathrm{T}} \boldsymbol{\varepsilon} - \frac{1}{K} p \right) \mathrm{d}\Omega + \beta \int_{\Omega_e} \delta p \boldsymbol{\nabla}^2 p \, \mathrm{d}\Omega = 0 \tag{12.79}$$

The last term may be integrated by parts to yield a form which is more amenable to computation as

$$\int_\Omega \delta p \left(\mathbf{m}^{\mathrm{T}} \boldsymbol{\varepsilon} - \frac{1}{K} p \right) \mathrm{d}\Omega + \beta \int_{\Omega_e} \frac{\partial \delta p}{\partial x_i} \frac{\partial p}{\partial x_i} \, \mathrm{d}\Omega = 0 \tag{12.80}$$

in which the resulting boundary terms are ignored. Upon discretization using equal order linear interpolation on triangles for $\mathbf{u}$ and p we obtain a form identical to that for the bubble with the exception that $\boldsymbol{\tau}$ is now given by

$$\boldsymbol{\tau} = \beta \mathbf{I} \tag{12.81}$$

On dimensional considerations with the first term in Eq. (12.80) the parameter β should have a value proportional to L^4/F, where L is length and F is force.

12.7.5 Galerkin least square method

In Chapter 3, Sec. 3.12.3 we introduced the Galerkin least square (GLS) approach as a modification to constructing a weak form. As a general scheme for solving the differential equations (3.1) by a finite element method we may write the GLS form as

$$\int_\Omega \delta \mathbf{u}^{\mathrm{T}} \mathbf{A}(\mathbf{u}) \, \mathrm{d}\Omega + \int_{\Omega_e} \delta \mathbf{A}(\mathbf{u})^{\mathrm{T}} \boldsymbol{\tau} \mathbf{A}(\mathbf{u}) \, \mathrm{d}\Omega = 0 \tag{12.82}$$

where the first term represents the normal Galerkin form and the added terms are computed for each element individually including a weight $\boldsymbol{\tau}$ to provide dimensional balance and scaling. Generally, the $\boldsymbol{\tau}$ will involve parameters which have to be selected for good performance. Discontinuous terms on boundaries between elements that arise from higher order terms in $\mathbf{A}(\mathbf{u})$ are commonly omitted.

The form given above has been used by Hughes[44] as a means of stabilizing the fluid flow equations, which for the case of the incompressible Stokes problem coincide with those for incompressible linear elasticity. For this problem only the momentum equation is used in the least square terms. After substituting Eq. (12.75) into Eq. (12.76) the momentum equation may be written as (assuming that G and K are constant in each element)

$$G \frac{\partial^2 u_j}{\partial x_i^2} + \left(1 + \frac{G}{3K} \right) \frac{\partial p}{\partial x_j} = 0 \tag{12.83}$$

A more convenient form results by using a single parameter defined as

$$\bar{G} = \frac{G}{1 + G/3K} \tag{12.84}$$

With this form the least square term to be appended to each element may be written as

$$\int_{\Omega_e} \left(\bar{G} \frac{\partial^2 \delta u_i}{\partial x_k^2} + \frac{\partial \delta p}{\partial x_i} \right) \tau_{ij} \left(\bar{G} \frac{\partial^2 u_j}{\partial x_m^2} + \frac{\partial p}{\partial x_j} \right) \mathrm{d}\Omega \tag{12.85}$$

This leads to terms to be added to the standard Galerkin equations and is expressed as

$$\begin{bmatrix} \mathbf{A}^s & \mathbf{C}^s \\ \mathbf{C}^{s,T} & \mathbf{V}^s \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix}$$

where

$$\mathbf{A}_{ij}^s = \int_{\Omega_e} \bar{G}^2 \nabla^2 N_i \boldsymbol{\tau} \nabla^2 N_j \, \mathrm{d}\Omega$$

$$\mathbf{C}_{ij}^s = \int_{\Omega_e} \bar{G} \nabla^2 N_i \boldsymbol{\tau} \boldsymbol{\nabla} N_j \, \mathrm{d}\Omega$$

$$V_{ij}^s = \int_{\Omega_e} (\boldsymbol{\nabla} N_i)^{\mathrm{T}} \boldsymbol{\tau} \boldsymbol{\nabla} N_j \, \mathrm{d}\Omega$$

and the operators on the shape functions are given in two dimensions by

$$\nabla^2 N_i = \frac{\partial^2 N_i}{\partial x_1^2} + \frac{\partial^2 N_i}{\partial x_2^2}$$

$$\boldsymbol{\nabla} N_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x_1} & \dfrac{\partial N_i}{\partial x_2} \end{bmatrix}^{\mathrm{T}}$$

Note again that all infinite terms between elements are ignored.

For linear triangular elements the second derivatives of the shape functions are identically zero within the element and only the $\mathbf{V}$ term remains and is now nearly identical to the form obtained by eliminating the bubble mode. In the work of Hughes *et al*, $\boldsymbol{\tau}$ is given by

$$\boldsymbol{\tau} = -\frac{\alpha h^2}{2G} \mathbf{I} \tag{12.86}$$

where α is a parameter which is recommended to be of $O(1)$ for linear triangles and quadrilaterals.

12.7.6 Incompressibility by time stepping

The fully incompressible case (i.e., $K = \infty$) has been studied by Zienkiewicz and Wu[46] using various time stepping procedures. Their applications concern the solution of fluid problems in which the rate effects for the Stokes problem appear as first derivatives of time. We can consider such a method here as a procedure to obtain the static solutions of elasticity problems in the limit as the rate terms become zero. Thus, this approach is considered here as a method for either the Stokes problem or the case of static incompressible elasticity.

The governing equations for slightly compressible Stokes flow may be written as

$$\rho_0 \frac{\partial u_i}{\partial t} - \frac{\partial \sigma_{ij}^d}{\partial x_j} - \frac{\partial p}{\partial x_i} = 0 \tag{12.87}$$

$$\frac{1}{\rho_0 c^2} \frac{\partial p}{\partial t} - \frac{\partial u_i}{\partial x_i} = 0 \tag{12.88}$$

where ρ_0 is density (taken as unity in subsequent developments), $c = (K/\rho_0)^{1/2}$ is the speed of compressible waves, p is the pressure (here taken as positive in tension), and u_i is a velocity (or for elasticity interpretations a displacement) in the i-coordinate direction. Note that the above form assumes some compressibility in order to introduce the pressure rate term. At the steady limit this term is not involved, consequently, the solution will correspond to the incompressible case. Deviatoric stresses σ_{ij}^d are related to deviatoric strains (or strain rates for fluids) as described by Eq. (12.74).

Zienkiewicz and Wu consider many schemes for integrating the above equations in time. Here we introduce only one of the forms, which will also be used in the solution of the fluid equations which include transport effects (see Volume 3). For the full fluid equations the algorithm is part of the *characteristic based split* (CBS) method.[47–50]

The equations are discretized in time using the approximations $u(t_n) \approx u^n$ and time derivatives

$$\frac{\partial u_i}{\partial t} \approx \frac{u^{n+1} - u^n}{\Delta t} \tag{12.89}$$

where $\Delta t = t_{n+1} - t_n$. The time discretized equations are given by

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\partial \sigma_{ij}^{d,n}}{\partial x_j} + \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \tag{12.90}$$

$$\frac{1}{c^2} \frac{p^{n+1} - p^n}{\Delta t} = \frac{\partial u_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta u_i}{\partial x_i} \tag{12.91}$$

where $\Delta p = p^{n+1} - p^n$; $\Delta u_i = u_i^{n+1} - u_i^n$; θ_1 can vary between 1/2 and 1; and θ_2 can vary between 0 and 1. In all that follows we shall use $\theta_1 = 1$.

The form to be considered uses a split of the equations by defining an intermediate approximate velocity u_i^* at time t_{n+1} when integrating the equilibrium equation (12.90). Accordingly, we consider

$$\frac{u_i^* - u_i^n}{\Delta t} = \frac{\partial \sigma_{ij}^{d,n}}{\partial x_j} \tag{12.92}$$

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \tag{12.93}$$

Differentiating the second of these with respect to x_i to get the divergence of u_i^{n+1} and combining with the discrete pressure equation (12.91) results in

$$\frac{1}{c^2} \frac{\Delta p}{\Delta t} - \theta_2 \Delta t \frac{\partial^2 \Delta p}{\partial x_i \partial x_i} = \Delta t \frac{\partial^2 p^n}{\partial x_i \partial x_i} + \frac{\partial u_i^*}{\partial x_i} \tag{12.94}$$

Thus, the original problem has been replaced by a set of three equations which need to be solved successively.

Equations (12.92), (12.93) and (12.94) may be written in a weak form using as weighting functions $\delta\mathbf{u}^*$, $\delta\mathbf{u}$ and δp, respectively (viz. Chapter 3). They are then discretized in space using the approximations

$$\begin{aligned}
\mathbf{u}^n &\approx \hat{\mathbf{u}}^n = \mathbf{N}_u\tilde{\mathbf{u}}^n \qquad \text{and} \qquad \delta\mathbf{u}^n \approx \delta\hat{\mathbf{u}} = \mathbf{N}_u\delta\tilde{\mathbf{u}}^n \\
\mathbf{u}^* &\approx \hat{\mathbf{u}}^* = \mathbf{N}_u\tilde{\mathbf{u}}^* \qquad \text{and} \qquad \delta\mathbf{u}^* \approx \delta\hat{\mathbf{u}}^* = \mathbf{N}_u\delta\tilde{\mathbf{u}}^* \\
p^n &\approx \hat{p}^n = \mathbf{N}_p\tilde{\mathbf{p}}^n \qquad \text{and} \qquad \delta p \approx \delta\hat{p} = \mathbf{N}_p\delta\tilde{\mathbf{p}}
\end{aligned}$$

with similar expressions for $\mathbf{u}^{n+1}$ and p^{n+1}. The final discrete form is given by the three equation sets

$$\frac{1}{\Delta t}\mathbf{M}_u(\tilde{\mathbf{u}}^* - \tilde{\mathbf{u}}^n) = -\mathbf{A}\tilde{\mathbf{u}}^n + \mathbf{f}_1 \tag{12.95}$$

$$\frac{1}{\Delta t}\mathbf{M}_u\big(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^*\big) = -\mathbf{C}^{\mathrm{T}}(\tilde{\mathbf{p}}^n + \theta_2\Delta\tilde{\mathbf{p}}) \tag{12.96}$$

$$\left[\frac{1}{\Delta t}\mathbf{M}_p + \theta_2\Delta t\mathbf{H}\right]\Delta\tilde{\mathbf{p}} = -\mathbf{C}\tilde{\mathbf{u}}^* - \Delta t\mathbf{H}\tilde{\mathbf{p}}^n + \mathbf{f}_3 \tag{12.97}$$

In the above we have integrated by parts all the terms which involve derivatives on deviator stress (σ_{ij}^d), pressure (p) and displacements (velocities). In addition we consider only the case where $u_i^{n+1} = u_i^* = \bar{u}_i$ on the boundary Γ_u (thus requiring $\delta u_i = \delta u_i^* = 0$ on Γ_u). Accordingly, the matrices are defined as

$$\begin{aligned}
\mathbf{M}_u &= \int_\Omega \mathbf{N}_u^{\mathrm{T}}\mathbf{N}_u\,\mathrm{d}\Omega & \mathbf{M}_p &= \int_\Omega \frac{1}{c^2}\mathbf{N}_p^{\mathrm{T}}\mathbf{N}_p\,\mathrm{d}\Omega \\
\mathbf{A} &= \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}_d\mathbf{B}\,\mathrm{d}\Omega & \mathbf{C} &= \int_\Omega \frac{\partial\mathbf{N}_p}{\partial x_i}\mathbf{N}_u\,\mathrm{d}\Omega \\
\mathbf{H} &= \int_\Omega \frac{\partial\mathbf{N}_p^{\mathrm{T}}}{\partial x_i}\frac{\partial\mathbf{N}_p}{\partial x_i}\,\mathrm{d}\Omega & \mathbf{f}_1 &= \int_{\Gamma_t} \mathbf{N}_u^{\mathrm{T}}(\bar{\mathbf{t}} - k\mathbf{n}p^n)\,\mathrm{d}\Gamma \\
\mathbf{f}_3 &= \int_{\Gamma_u} \mathbf{N}_p^{\mathrm{T}}\mathbf{n}^{\mathrm{T}}\bar{\mathbf{u}}\,\mathrm{d}\Gamma
\end{aligned} \tag{12.98}$$

in which $\mathbf{D}_d$ are the deviatoric moduli defined previously. The parameter k denotes an option on alternative methods to split the boundary traction term and is taken as either zero or unity. We note that a choice of zero simplifies the computation of boundary contributions, however, some would argue that unity is more consistent with the integration by parts.

The boundary pressure acting on Γ_t is computed from the specified surface tractions ($\bar{t}_i$) and the 'best' estimate for the deviator stress at step-$n+1$ which is given by $\sigma_{ij}^{d,*}$. Accordingly,

$$\bar{p}^{n+1} \approx n_i\bar{t}_i - n_i\sigma_{ij}^{d,*}n_j$$

is imposed at each node on the boundary Γ_t.

In general we require that $\Delta t < \Delta t_{\text{crit}}$ where the critical time step is $h^2/2G$ (in which h is the element size). Such a quantity is obviously calculated independently for each element and the lowest value occurring in any element governs the overall stability. It is possible and useful to use here the value of Δt calculated for each element separately when calculating incompressible stabilizing terms in the pressure calculation and the overall time step elsewhere (we shall label the time increments multiplying $\mathbf{H}$ in Eq. (12.97) as Δt_{int}). A ratio of $\gamma = \Delta t_{\text{int}}/\Delta t$ greater than unity improves considerably the stabilizing properties. As Eq. (12.97) has greater stability than Eqs (12.95) and (12.96), and for $\theta_2 \geqslant 1/2$ is unconditionally stable, we recommend that the time step used in this equation be $\gamma \Delta t_{\text{cr}}$ for each node. Generally a value of 2 is good as we shall show in the examples (for details see reference 50).

Equation (12.95) defines a value of $\tilde{\mathbf{u}}^*$ entirely in terms of known quantities at the n-step. If the mass matrix $\mathbf{M}_u$ is made diagonal by lumping (see Chapter 17 and Appendix I) the solution is thus trivial. Such an equation is called *explicit*. The equation for $\Delta\tilde{\mathbf{p}}$, on the other hand depends on both $\mathbf{M}_p$ and $\mathbf{H}$ and it is not possible to make the latter diagonal easily.† It is possible to make $\mathbf{M}_p$ diagonal using a similar method as that employed for $\mathbf{M}_u$. Thus, if θ_2 is zero this equation will also be explicit, otherwise it is necessary to solve a set of algebraic equations and the method for this equation is called *implicit*. Once the value of $\Delta\tilde{\mathbf{p}}$ is known the solution for $\tilde{\mathbf{u}}^{n+1}$ is again explicit. In practice the above process is quite simple to implement, however, it is necessary to satisfy *stability* requirements by limiting the size of the time increment. This is discussed further in Chapter 18 and in reference 47. Here we only wish to show the limit result as the changes in time go to zero (i.e., for a constant in time load value) and when full incompressibility is imposed.

At the steady limit the solutions become

$$\tilde{\mathbf{u}}^n = \tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}} \qquad \text{and} \qquad \tilde{\mathbf{p}}^n = \tilde{\mathbf{p}}^{n+1} = \tilde{\mathbf{p}} \tag{12.99}$$

Eliminating $\mathbf{u}^*$ the discrete equations reduce to the mixed problem

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \Delta t\left(\mathbf{C}^{\mathrm{T}}\mathbf{M}_u^{-1}\mathbf{C} - \theta_1\mathbf{H}\right) \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{f} \\ \mathbf{0} \end{Bmatrix} = \mathbf{0} \tag{12.100}$$

At the steady limit we again recover a term on the diagonal which stabilizes the solution. This term is again of a Laplace equation type – indeed, it is now the difference between two discrete forms for the Laplace equation. The term $\mathbf{C}^{\mathrm{T}}\mathbf{M}_u^{-1}\mathbf{C}$ makes the bandwidth of the resulting equations larger – thus this form is different from all the previous methods discussed above.

12.7.7 Comparisons

To provide some insight into the behaviour of the above methods we consider two example problems. The first is a problem often used to assess the performance of

† It is possible to diagonalize the matrix by solving an eigenproblem as shown in Chapter 17 – for large problems this requires more effort than is practical.

codes to solve steady-state Stokes flow problems – which is identical to the case for incompressible linear elasticity. The second example is a problem in nearly incompressible linear elasticity.

Example: Driven cavity A two-dimensional plane (strain) case is considered for a square domain with unit side lengths. The material properties are assumed to be fully incompressible ($\nu = 0.5$) with unit viscosity (elastic shear modulus, G, of unity). All boundaries of the domain are restrained in the x and y directions with the top boundary having a unit tangential velocity (displacement) at all nodes except the corner ones. Since the problem is incompressible it is necessary to prescribe the pressure at one point in the mesh – this is selected as the centre node along the bottom edge. The 10×10 element mesh of triangular elements (200 elements total) used for the comparison is shown in Fig. 12.10(a). The elements used for the analysis use linear velocity (displacement) and pressure on three-noded triangles. Results are presented for the horizontal velocity along the vertical centre line AA and for vertical velocity and pressure along the horizontal centre line BB. Three forms of stabilization are considered:

1. Galerkin least square (GLS) Brezzi–Pitkaranta (BP) where the effect of α on τ is assessed. The results for the horizontal velocity are given in Fig. 12.10(b) and for the vertical velocity and pressure in Figs 12.10(c) and (d), respectively. From the analysis it is assessed that the stabilization parameter τ should be about 0.5 to 1 (as also indicated by Hughes *et al.*[44]). Use of lower values leads to excessive oscillation in pressure and use of higher values to strong dissipation of pressure results.
2. Cubic bubble (MINI) element stabilization. Results for vertical velocity are nearly indistinguishable from the GLS results as indicated in Fig. 12.11; however, those for pressure show oscillation. Such oscillation has also been observed by others along with some suggested boundary modifications.[51] No free parameters exist for this element (except possible modification of the bubble mode used), thus, no artificial 'tuning' is possible. Use of more refined meshes leads to a strong decrease in the oscillation.
3. Enhanced strain stabilization with quadratic modes. In Fig. 12.11 we show results obtained using the enhanced formulation presented in Eq. (12.73). These results are free of oscillation in pressures and require no tuning parameters. For use in solving linear elasticity and Stokes problems they prove to be the most robust; however, when used with other material models there are limitations in their use.
4. The CBS algorithm. Finally in Fig. 12.11 we present results using the CBS solution which may be compared with GLS, $\alpha = 0.5$. Once again the reader will observe that with $\gamma = 2$, the results of CBS reproduce very closely those of GLS, $\alpha = 0.5$. However, in results for $\gamma = 1$ no oscillations are observed and they are quite reasonable. This ratio for γ is where the algorithm gives excellent results in incompressible flow modelling as will be demonstrated further in results presented in Volume 3.

Example: Tension strip with slot As a second example we consider a plane strain linear problem on a square domain with a central slot. The domain is two units square and the central slot has a total width of 0.4 units and a height of 0.1 units. The ends of the slot are semicircular. Lateral boundaries have specified normal

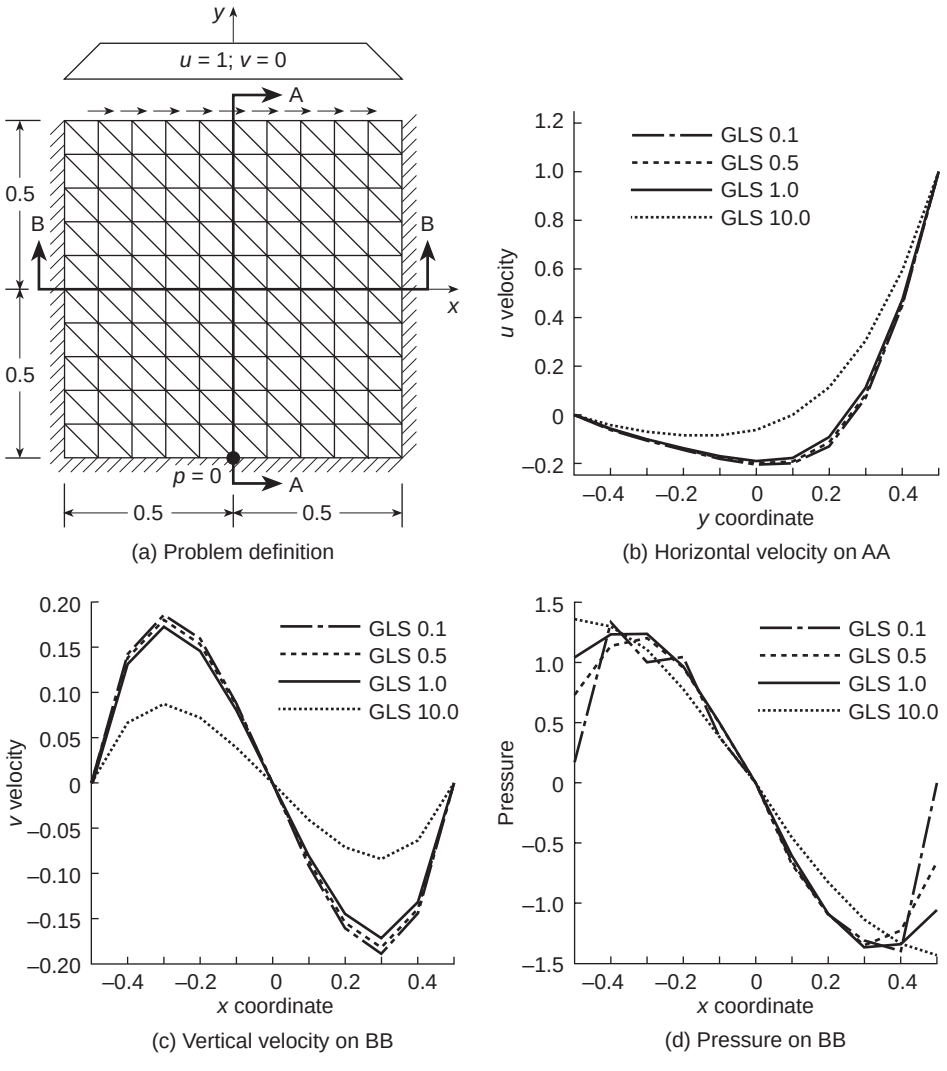
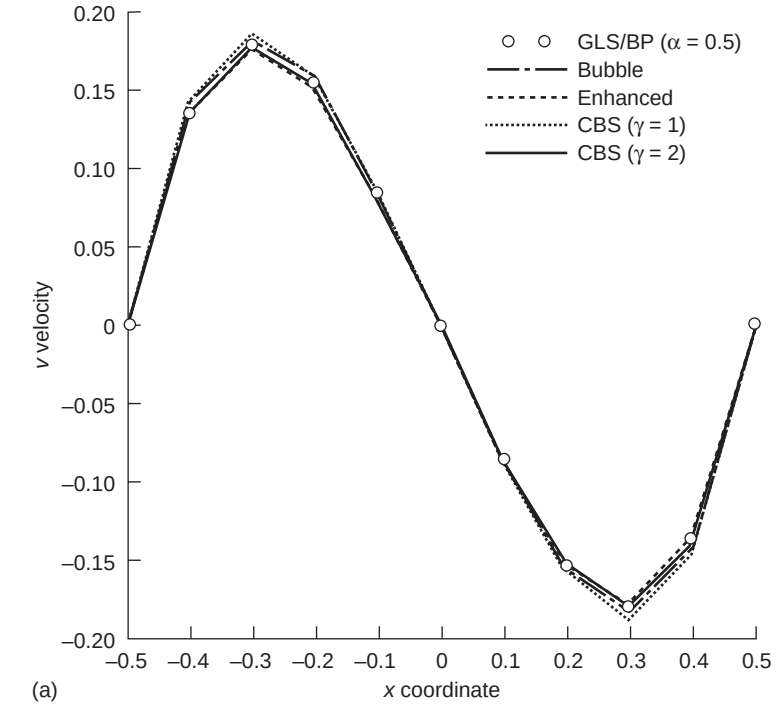


Fig. 12.10 Mesh and GLS/Brezzi–Pitkaranta results.

displacement and zero tangential traction. The top and bottom boundaries are uniformly stretched by a uniform axial loading and lateral boundaries are maintained at zero displacement. We consider the linear elastic problem with elastic properties $E = 24$ and $\nu = 0.499995$; thus, giving a nearly incompressible situation. An unstructured mesh of triangles is constructed as shown in Fig. 12.12(b). Results for the pressure along the horizontal and vertical centre lines (i.e., the x and y axes) are presented in Figs 12.13(a) and 12.13(b) and the distribution of the vertical displacement is shown in Fig. 12.13(c). We note that the results for this problem cause very strong gradients in stress near the ends of the slot. The mesh used for the analysis is not highly refined in this region and hence results from different analyses can be



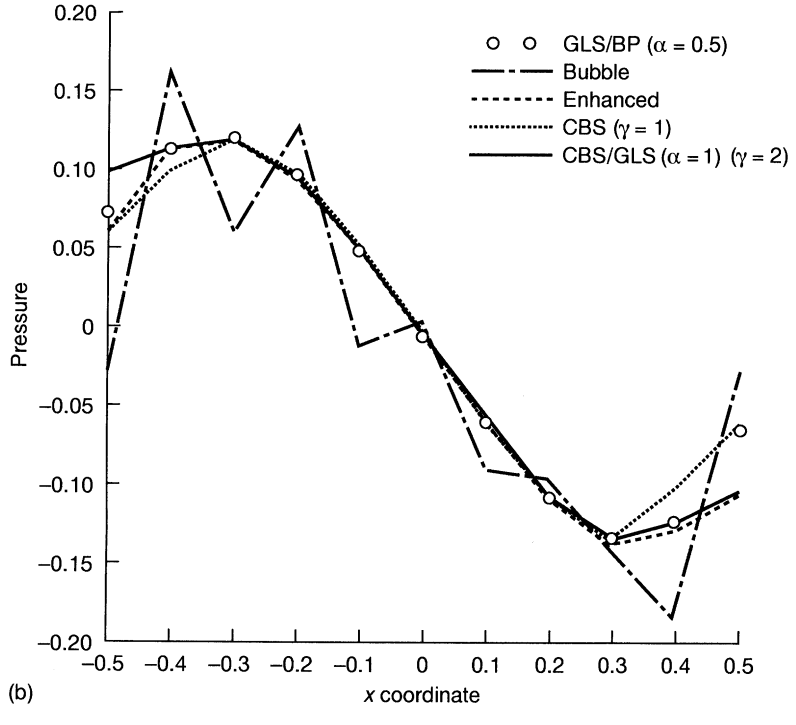
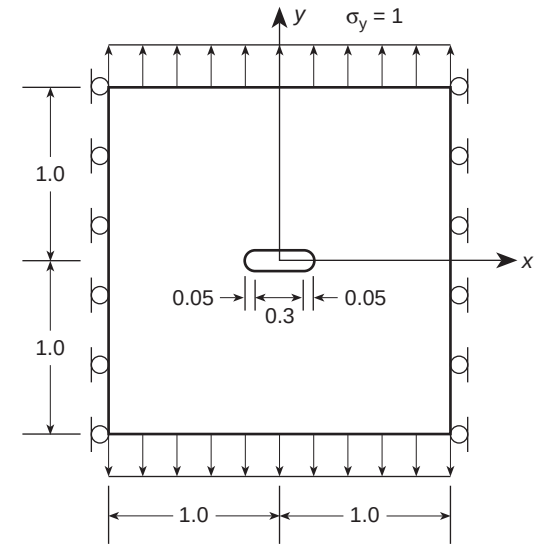
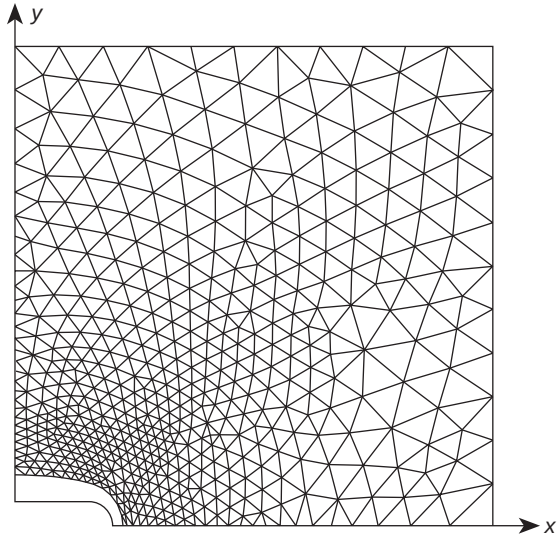


Fig. 12.11 Vertical velocity and pressure for driven cavity problem.



(a) Problem description



(b) Mesh for quadrant

Fig. 12.12 Region and mesh used for slotted tension strip.

expected to differ in this region. The results obtained using all formulations are similar in distribution. However, the bubble form does show some oscillations in pressure indicating that the stabilization achieved is not completely adequate. Results for the CBS algorithm show an oscillation in the pressure along the x-axis at the boundary of the slot. This is caused, we believe, by an inadequate resolution of the pressure condition at this point of the curved boundary. In general, however,

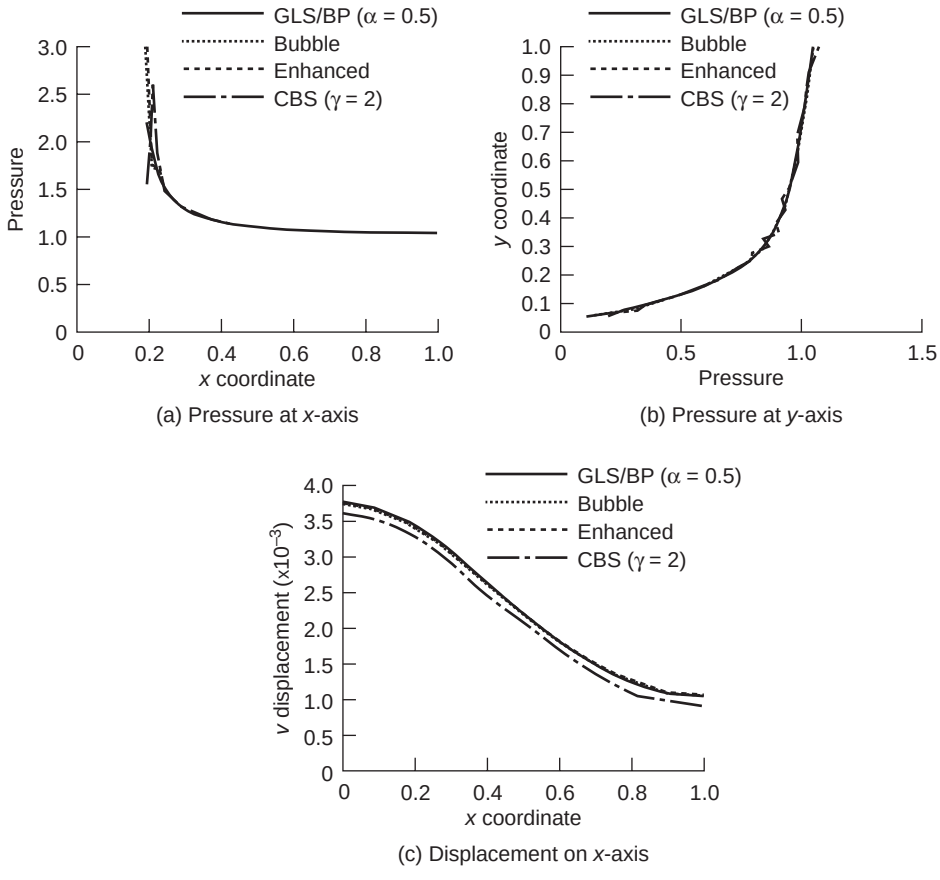


Fig. 12.13 Pressures and displacements for slot problems.

the results achieved with all forms are satisfactory and indicate that stabilized methods may be considered for use in problems where constraints, such as incompressibility, are encountered.

12.8 Concluding remarks

In this chapter we have considered in some detail the application of mixed methods to incompressible problems and also we have indicated some alternative procedures. The extension to non-isotropic problems and non-linear problems will be presented in Volume 2, but will follow similar lines. In Volume 3 we shall note how important the problem is in the context of fluid mechanics and it is there that much of the attention to it has been given.

In concluding this chapter we would like to point out two matters:

1. The mixed formulation discovers immediately the non-robustness of certain irreducible (displacement) elements and, indeed, helps us to isolate those which

perform well from those that do not. Thus, it has merit which as a test is applicable to many irreducible forms at all times.

2. In elasticity, certain mixed forms work quite well at the near incompressible limit without resort to splits into deviatoric and mean parts. These include the two-field quadrilateral element of Pian–Sumihara and the enhanced strain quadrilateral element of Simo–Rifai which were presented in the previous chapter. There we noted how well such elements work for Poisson's ratio approaching one-half as compared to the standard irreducible element of a similar type.

References

1. L.R. Herrmann. Finite element bending analysis of plates. In *Proc. 1st Conf. Matrix Methods in Structural Mechanics*. AFFDL-TR-66-80, pages 577–602, Wright-Patterson Air Force Base, Ohio, 1965.
2. S.W. Key. Variational principle for incompressible and nearly incompressibly anisotropic elasticity. *Intern. J. Solids Struct.*, **5**, 951–964, 1969.
3. O.C. Zienkiewicz, R.L. Taylor, and J.A.W. Baynham. Mixed and irreducible formulations in finite element analysis. In S.N. Atluri, R.H. Gallagher, and O.C. Zienkiewicz, editors, *Hybrid and Mixed Finite Element Methods*, pages 405–431. John Wiley & Sons, 1983.
4. D.N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, **21**, 337–344, 1984.
5. M. Fortin and N. Fortin. Newer and newer elements for incompressible flow. In R.H. Gallagher, G.F. Carey, J.T. Oden, and O.C. Zienkiewicz, editors, *Finite Elements in Fluids*, volume 6, chapter 7, pages 171–188. John Wiley & Sons, 1985.
6. J.T. Oden. R.I.P. methods for Stokesian flow. In R.H. Gallagher, D.N. Norrie, J.T. Oden, and O.C. Zienkiewicz, editors, *Finite Elements in Fluids*, volume 4, chapter 15, pages 305–318. John Wiley & Sons, 1982.
7. M. Crouzcix and P.A. Raviart. Conforming and non-conforming finite element methods for solving stationary Stokes equations. *RAIRO*, **7-R3**, 33–76, 1973.
8. D.S. Malkus. Eigenproblems associated with the discrete LBB condition for incompressible finite elements. *Int. J. Eng. Sci.*, **19**, 1299–1370, 1981.
9. M. Fortin. Old and new finite elements for incompressible flow. *International Journal for Numerical Methods in Fluids*, **1**, 347–364, 1981.
10. C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Comput. Fluids*, **1**, 73–100, 1973.
11. T.J.R. Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *Intern. J. Num. Meth. Engng*, **15**, 1413–1418, 1980.
12. J.C. Simo, R.L. Taylor, and K.S. Pister. Variational and projection methods for the volume constraint in finite deformation plasticity. *Comput. Meth. Appl. Mech. Engng*, **51**, 177–208, 1985.
13. J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*, volume 7 of *Interdisciplinary Applied Mathematics*. Springer-Verlag, Berlin, 1998.
14. D.J. Naylor. Stresses in nearly incompressible materials for finite elements with application to the calculation of excess pore pressures. *Intern. J. Num. Meth. Engng*, **8**, 443–460, 1974.
15. O.C. Zienkiewicz and P.N. Godbole. Viscous incompressible flow with special reference to non-Newtonian (plastic) flows. In R.H. Gallagher *et al.*, editor, *Finite Elements in Fluids*, volume 1, chapter 2, pages 25–55. John Wiley & Sons, 1975.

16. T.J.R. Hughes, R.L. Taylor, and J.F. Levy. High Reynolds number, steady, incompressible flows by a finite element method. In R.H. Gallagher *et al*., editor, *Finite Elements in Fluids*, volume 3. John Wiley & Sons, 1978.
17. O.C. Zienkiewicz, J. Too, and R.L. Taylor. Reduced integration technique in general analysis of plates and shells. *Intern. J. Num. Meth. Engng*, **3**, 275–290, 1971.
18. S.F. Pawsey and R.W. Clough. Improved numerical integration of thick slab finite elements. *Intern. J. Num. Meth. Engng*, **3**, 575–586, 1971.
19. O.C. Zienkiewicz and E. Hinton. Reduced integration, function smoothing and nonconformity in finite element analysis. *J. Franklin Inst*., **302**, 443–461, 1976.
20. O.C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, London, 3rd edition, 1977.
21. D.S. Malkus and T.J.R. Hughes. Mixed finite element methods in reduced and selective integration techniques: A unification of concepts. *Comput. Meth. Appl. Mech. Engng*, **15**, 63–81, 1978.
22. O.C. Zienkiewicz and S. Nakazawa. On variational formulations and its modification for numerical solution. *Comput. Struct*., **19**, 303–313, 1984.
23. M.S. Engleman, R.L. Sani, P.M. Gresho, and H. Bercovier. Consistent vs. reduced integration penalty methods for incompressible media using several old and new elements. *Internat. J. Num. Meth. Fluids*, **2**, 25–42, 1982.
24. D.N. Arnold. Discretization by finite elements of a model parameter dependent problem. *Num. Meth*., **37**, 405–421, 1981.
25. J.C. Nagtegaal, D.M. Parks, and J.R. Rice. On numerical accurate finite element solutions in the fully plastic range. *Comput. Meth. Appl. Mech. Engng*, **4**, 153–177, 1974.
26. M. Vogelius. An analysis of the *p*-version of the finite element method for nearly incompressible materials; uniformly optimal error estimates. *Num. Math*., **41**, 39–53, 1983.
27. S.W. Sloan and M.F. Randolf. Numerical prediction of collapse loads using finite element methods. *Internat. J. Num. Anal. Meth. Geomech*., **6**, 47–76, 1982.
28. O.C. Zienkiewicz, J.P. Vilotte, S. Toyoshima, and S. Nakazawa. Iterative method for constrained and mixed approximation. An inexpensive improvement of FEM performance. *Comput. Meth. Appl. Mech. Engng*, **51**, 3–29, 1985.
29. K.J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Non-Linear Programming*. Stanford University Press, Stanford, CA, 1958.
30. M.R. Hestenes. Multiplier and gradient methods. *J. Opt. Theory Appl*., **4**, 303–320, 1969.
31. M.J.D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*. Academic Press, London, 1969.
32. C.A. Felippa. Iterative procedure for improving penalty function solutions of algebraic systems. *Intern. J. Num. Meth. Engng*, **12**, 165–185, 1978.
33. M. Fortin and F. Thomasset. Mixed finite element methods for incompressible flow problems. *J. Comp. Physics*, **31**, 113–145, 1973.
34. M. Fortin and R. Glowinski. *Augmented Lagrangian Methods: Applications to Numerical Solution of Boundary- Value Problems*. North-Holland, Amsterdam, 1983.
35. D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Mass. 1984.
36. J. H. Argyris. Three-dimensional anisotropic and inhomogeneous media – matrix analysis for small and large displacements. *Ingenieur Archiv*, **34**, 33–55, 1965.
37. O.C. Zienkiewicz and S. Valliappan. Analysis of real structures for creep plasticity and other complex constitutive laws. In M. Te'eni, editor, *Structure of Solid Mechanics and Engineering Design*, volume Part 1, pages 27–48. 1971.
38. O.C. Zienkiewicz. *The Finite Element Method in Engineering Science*. McGraw-Hill, London, 2nd edition, 1971.

39. J.C. Simo and R.L. Taylor. Quasi-incompressible finite elasticity in principal stretches: Continuum basis and numerical algorithms. *Comput. Meth. Appl. Mech. Engng*, **85**, 273–310, 1991.
40. R. Courant. Variational methods for the solution of problems of equilibrium and vibration. *Bull. Amer. Math. Soc.*, **49**, 1–61, 1943.
41. F. Brezzi and J. Pitkäranta. On the stabilization of finite element approximations of the Stokes problem. In W. Hackbusch, editor, *Efficient solution of Elliptic Problems, Notes on Numerical Fluid Mechanics*, volume 10. Vieweg, Wiesbaden, 1984.
42. T.J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Comput. Meth. Appl. Mech. Engng*, **59**, 85–99, 1986.
43. T.J.R. Hughes and L.P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulation that converge for all velocity/pressure spaces. *Comput. Meth. Appl. Mech. Engng*, **65**, 85–96, 1987.
44. T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Comput. Meth. Appl. Mech. Engng*, **73**, 173–189, 1989.
45. E. Onate. Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems. *Comput. Meth. Appl. Mech. Engng*, **151**, 233–265, 1998.
46. O.C. Zienkiewicz and J. Wu. Incompressibility without tears! How to avoid restrictions of mixed formulations. *Intern. J. Num. Meth. Engng*, **32**, 1184–1203, 1991.
47. O.C. Zienkiewicz and R. Codina. A general algorithm for compressible and incompressible flow – Part 1: The split, characteristic-based scheme. *Internat. J. Num. Meth. Fluids*, **20**, 869–885, 1995.
48. O.C. Zienkiewicz, P. Nithiarasu, R. Codina, M. Vazquez, and P. Ortiz. The characteristic-based-split procedure: An efficient and accurate algorithm for fluid problems. *Internat. J. Num. Meth. Fluids*, **31**, 359 – 392, 1999.
49. O.C. Zienkiewicz, K. Morgan, B.V.K. Satya Sai, R. Codina, and M. Vasquez. A general algorithm for compressible and incompressible flow Part II: Tests on the explicit form. *Internat. J. Num. Meth. Fluids*, **20**, 887–913, 1995.
50. P. Nithiarasu and O.C. Zienkiewicz. On stabilization of the CBS algorithm. Internal and external time steps. *Intern. J. Num. Meth. Engng*, **48**, 875–880.
51. R. Pierre. Simple c_0 approximations for the computation of incompressible flows. *Comput. Meth. Appl. Mech. Engng*, **68**, 205–227, 1988.

13

Mixed formulation and constraints – incomplete (hybrid) field methods, boundary/Trefftz methods

13.1 General

In the previous two chapters we have assumed in the mixed approximation that all the variables were defined and approximated in the same manner throughout the domain of the analysis. This process can, however, be conveniently abandoned on occasion with different formulations adopted in different subdomains and with some variables being only approximated on surfaces joining such subdomains. In this part we shall discuss such *incomplete* or *partial field* approximations which include various so-called *hybrid* formulations.

In all the examples given here we shall consider elastic solid body approximations only, but extension to the heat transfer or other field problems, etc., can be readily made as a simple exercise following the procedures outlined.

13.2 Interface traction link of two (or more) irreducible form subdomains

One of the most obvious and frequently encountered examples of an 'incomplete field' approximation is the subdivision of a problem into two (or more) subdomains in each of which an irreducible (displacement) formulation is used. Independently approximated Lagrange multipliers (tractions) are used on the interface to join the subdomains, as in Fig. 13.1(a).

In this problem we formulate the approximation in domain Ω^1 in terms of displacements $\mathbf{u}^1$ and the interface tractions $\mathbf{t}^1 = \boldsymbol{\lambda}$. With the weak form using the standard virtual work expression [see Eqs (11.22)–(11.24)] we have

$$\int_{\Omega^1} \delta(\mathbf{S}\mathbf{u}^1)^{\mathrm{T}} \mathbf{D}^1 \mathbf{S}\mathbf{u}^1 \, \mathrm{d}\Omega - \int_{\Gamma_I} \delta\mathbf{u}^{1\mathrm{T}} \boldsymbol{\lambda} \, \mathrm{d}\Gamma - \int_{\Omega^1} \delta\mathbf{u}^{1\mathrm{T}} \mathbf{b} \, \mathrm{d}\Omega - \int_{\Gamma_t^1} \delta\mathbf{u}^{1\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d}\Gamma = 0 \qquad (13.1)$$

in which as usual we assume that the satisfaction of the prescribed displacement on Γ_{u^1} is implied by the approximation for $\mathbf{u}^1$. Similarly in domain Ω^2 we can write, now putting the interface traction as $\mathbf{t}^2 = -\boldsymbol{\lambda}$ to ensure equilibrium between the

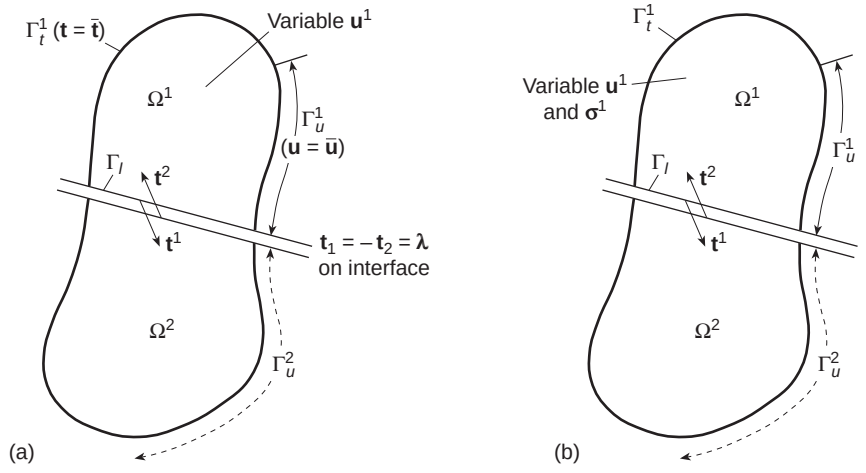


Fig. 13.1 Linking of two (or more) domains by traction variables defined only on the interfaces. (a) Variables in each domain are displacements **u** (internal irreducible form). (b) Variables in each domain are displacements and stresses **σ**–**u** (mixed form).

two domains,

$$\int_{\Omega^2} \delta(\mathbf{S}\mathbf{u}^2)^{\mathrm{T}}\mathbf{D}^2\mathbf{S}\mathbf{u}^2\,\mathrm{d}\Omega + \int_{\Gamma_I} \delta\mathbf{u}^{2\mathrm{T}}\boldsymbol{\lambda}\,\mathrm{d}\Gamma - \int_{\Omega^2} \delta\mathbf{u}^{2\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t^2} \delta\mathbf{u}^{2\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \tag{13.2}$$

The two subdomain equations are completed by a weak statement of displacement continuity on the interface between the two domains, i.e.,

$$\int_{\Gamma_I} \delta\boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{u}^2 - \mathbf{u}^1)\,\mathrm{d}\Gamma = 0 \tag{13.3}$$

Discretization of displacements in each domain and of the tractions $\boldsymbol{\lambda}$ on the interface yields the final system of equations. Thus putting the independent approximations as

$$\mathbf{u}^1 = \mathbf{N}_{u^1}\bar{u}^1 \tag{13.4}$$

$$\mathbf{u}^2 = \mathbf{N}_{u^2}\bar{u}^2 \tag{13.5}$$

$$\boldsymbol{\lambda} = \mathbf{N}_{\lambda}\bar{\boldsymbol{\lambda}} \tag{13.6}$$

we have

$$\begin{bmatrix} \mathbf{K}^1 & \mathbf{0} & \mathbf{Q}^1 \\ \mathbf{0} & \mathbf{K}^2 & \mathbf{Q}^2 \\ \mathbf{Q}^{1\mathrm{T}} & \mathbf{Q}^{2\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}}^1 \\ \tilde{\mathbf{u}}^2 \\ \tilde{\boldsymbol{\lambda}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \\ \mathbf{0} \end{Bmatrix} \tag{13.7a}$$

where

$$\begin{aligned}
\mathbf{K}^1 &= \int_{\Omega^1} \mathbf{B}^{1\mathrm{T}}\mathbf{D}^1\mathbf{B}^1 \,\mathrm{d}\Omega \\
\mathbf{K}^2 &= \int_{\Omega^2} \mathbf{B}^{2\mathrm{T}}\mathbf{D}^2\mathbf{B}^2 \,\mathrm{d}\Omega \\
\mathbf{Q}^1 &= -\int_{\Gamma_1} \mathbf{N}_{u^1}^{\mathrm{T}}\mathbf{N}_\lambda \,\mathrm{d}\Gamma \\
\mathbf{Q}^2 &= \int_{\Gamma_1} \mathbf{N}_{u^2}^{\mathrm{T}}\mathbf{N}_\lambda \,\mathrm{d}\Gamma \\
\mathbf{f}^1 &= \int_{\Omega^1} \mathbf{N}_{u^1}^{\mathrm{T}}\mathbf{b}^1 \,\mathrm{d}\Gamma + \int_{\Gamma_t^1} \mathbf{N}_{u^1}^{\mathrm{T}}\bar{\mathbf{t}}^1 \,\mathrm{d}\Gamma \\
\mathbf{f}^2 &= \int_{\Omega^2} \mathbf{N}_{u^2}^{\mathrm{T}}\mathbf{b}^2 \,\mathrm{d}\Gamma + \int_{\Gamma_t^2} \mathbf{N}_{u^2}^{\mathrm{T}}\bar{\mathbf{t}}^2 \,\mathrm{d}\Gamma
\end{aligned} \tag{13.7b}$$

We note that in the derivation of the above matrices the shape function $\mathbf{N}_\lambda$ and hence $\boldsymbol{\lambda}$ itself are only specified along the interface line – hence complying with our definition of partial field approximation.

The formulation just outlined can obviously be extended to many subdomains and in many cases of practical analysis is useful in ensuring a better matrix conditioning and allowing the solution to be obtained with reduced computational effort.[1]

The variables $\mathbf{u}^1$ and $\mathbf{u}^2$, etc., appear as internal variables within each subdomain (or superelement) and can be eliminated locally providing the matrices $\mathbf{K}^1$ and $\mathbf{K}^2$ are non-singular. Such non-singularity presupposes, however, that each of the subdomains has enough prescribed displacements to eliminate rigid body modes. If this is not the case partial elimination is always possible, retaining the rigid body modes until the complete solution is achieved.

The process described here is very similar to that introduced by Kron[2] at a very early date and, more recently, used by Farhat *et al.*[3] in the FETI method which uses the process on many individual element partitions as a means of iteratively solving large problems.

The formulation just used can, of course, be applied to a single field displacement formulation in which we are required to specify the displacement on the boundaries in a weak sense (rather than imposing these directly on the displacement shape functions).

This problem can be approached directly or can be derived simply via the first equation of (13.7a) in which we put $u^2 = \bar{u}$, the specified displacement on Γ_I.

Now the equation system is simply

$$\begin{bmatrix} \mathbf{K}^1 & \mathbf{Q}^1 \\ \mathbf{Q}^{1\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}}^1 \\ \tilde{\boldsymbol{\lambda}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\lambda \end{Bmatrix} \tag{13.8}$$

where

$$\mathbf{f}_\lambda = -\int_{\Gamma_I} \mathbf{N}_\lambda^{\mathrm{T}}\bar{\mathbf{u}} \,\mathrm{d}\Gamma \tag{13.9}$$

This formulation is often convenient for imposing a prescribed displacement on a displacement element field when the boundary values cannot fit the shape function field.

We have approached the above formulation directly via weak forms or weighted residuals. Of course, a variational principle could be given here simply as the minimization of total potential energy (see Chapter 2) subject to a Lagrange multiplier $\boldsymbol{\lambda}$ imposing subdomain continuity. The stationarity of

$$\Pi = \tfrac{1}{2}\int_{\Omega} (\mathbf{S}\mathbf{u})^{\mathrm{T}}\mathbf{D}(\mathbf{S}\mathbf{u})\,\mathrm{d}\Omega - \int_{\Omega}\mathbf{u}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t}\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma + \int_{\Gamma_I}\boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{u}^1 - \mathbf{u}^2)\,\mathrm{d}\Gamma \qquad (13.10)$$

would result in the equation set (13.1)–(13.3). The formulation is, of course, subject to limitations imposed by the stability and consistency conditions of the mixed patch test for selection of the appropriate number of $\boldsymbol{\lambda}$ variables.

13.3 Interface traction link of two or more mixed form subdomains

The problem discussed in the previous section could of course be tackled by assuming a mixed type of two-field approximation ($\boldsymbol{\sigma}/\mathbf{u}$) in each subdomain, as illustrated in Fig. 13.1(b).

Now in each subdomain variables $\mathbf{u}$ and $\boldsymbol{\sigma}$ will appear, but the linking will be carried out again with the interface traction $\boldsymbol{\lambda}$.

We now have, using the formulation of Sec. 11.4.2 for domain Ω^1 [see Eqs (11.29) and (11.22)],

$$\int_{\Omega^1}\delta\boldsymbol{\sigma}^{1\mathrm{T}}[(\mathbf{D}^1)^{-1}\boldsymbol{\sigma}^1 - \mathbf{S}\mathbf{u}^1]\,\mathrm{d}\Omega = 0 \qquad (13.11a)$$

$$\int_{\Omega^1}\delta(\mathbf{S}\mathbf{u}^1)^{\mathrm{T}}\boldsymbol{\sigma}^1\,\mathrm{d}\Omega - \int_{\Gamma_I}\delta\mathbf{u}^{1\mathrm{T}}\boldsymbol{\lambda}\,\mathrm{d}\Gamma - \int_{\Omega^1}\delta\mathbf{u}^{1\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t^1}\delta\mathbf{u}^{1\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \qquad (13.11b)$$

and for domain Ω^2 similarly

$$\int_{\Omega^2}\delta\boldsymbol{\sigma}^{2\mathrm{T}}[(\mathbf{D}^2)^{-1}\boldsymbol{\sigma}^2 - \mathbf{S}\mathbf{u}^2]\,\mathrm{d}\Omega = 0 \qquad (13.12a)$$

$$\int_{\Omega^2}\delta(\mathbf{S}\mathbf{u}^2)^{\mathrm{T}}\boldsymbol{\sigma}^2\,\mathrm{d}\Omega + \int_{\Gamma_I}\delta\mathbf{u}^{2\mathrm{T}}\boldsymbol{\lambda}\,\mathrm{d}\Gamma - \int_{\Omega^2}\delta\mathbf{u}^{2\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_t^2}\delta\mathbf{u}^{2\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma = 0 \qquad (13.12b)$$

With interface tractions in equilibrium the restoration of continuity demands that

$$\int_{\Gamma_I}\delta\boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{u}^2 - \mathbf{u}^1)\,\mathrm{d}\Gamma = 0 \qquad (13.13)$$

On discretization we now have

$$\mathbf{u}^1 = \mathbf{N}_{u^1}\tilde{\mathbf{u}}^1 \qquad \mathbf{u}^2 = \mathbf{N}_{u^2}\tilde{\mathbf{u}}^2$$

$$\boldsymbol{\sigma}^1 = \mathbf{N}_{\sigma^1}\tilde{\boldsymbol{\sigma}}^1 \qquad \boldsymbol{\sigma}^2 = \mathbf{N}_{\sigma^2}\tilde{\boldsymbol{\sigma}}^2$$

$$\boldsymbol{\lambda} = \mathbf{N}_{\lambda}\tilde{\boldsymbol{\lambda}}$$

and

$$\begin{bmatrix} \mathbf{A}^1 & \mathbf{C}^1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}^{1\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^1 \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^2 & \mathbf{C}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}^{2\mathrm{T}} & \mathbf{0} & \mathbf{Q}^2 \\ \mathbf{0} & \mathbf{Q}^{1\mathrm{T}} & \mathbf{0} & \mathbf{Q}^{2\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}}^1 \\ \tilde{\mathbf{u}}^1 \\ \tilde{\boldsymbol{\sigma}}^2 \\ \tilde{\mathbf{u}}^2 \\ \tilde{\boldsymbol{\lambda}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1^1 \\ \mathbf{f}_1^2 \\ \mathbf{f}_2^1 \\ \mathbf{f}_2^2 \\ \mathbf{0} \end{Bmatrix} \tag{13.14}$$

with $\mathbf{A}$, $\mathbf{C}$, $\mathbf{f}_1$, and $\mathbf{f}_2$ defined similarly to Eq. (11.32) with appropriate subdomain subscripts and $\mathbf{Q}^1$ and $\mathbf{Q}^2$ given as in (13.7b).

All the remarks made in the previous section apply here once again – though use of the above form does not appear frequently.

13.4 Interface displacement 'frame'

13.4.1 General

In the preceding examples we have used traction as the interface variable linking two or more subdomains. Due to lack of rigid body constraints the elimination of local subdomain displacements has generally been impossible. For this and other reasons it is convenient to accomplish the linking of subdomains via a displacement field *defined only on the interface* [Fig. 13.2(a)] and to eliminate all the interior variables so that this linking can be accomplished via a standard stiffness matrix procedure using only the interface variables.

The *displacement frame* can be made to surround the subdomain completely and if all internal variables are eliminated will yield a stiffness matrix of a new 'element'

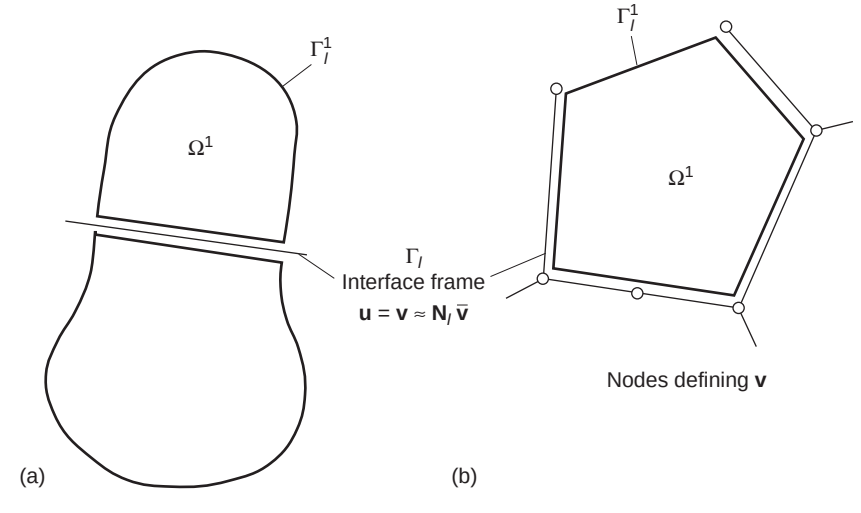


Fig. 13.2 Interface displacement field specified on a 'frame' linking subdomains: (a) two-domain link; (b) a 'superelement' (hybrid) which can be linked to many other similar elements.

which can be used directly in coupling with any other element with similar displacement assumptions on the interface, irrespective of the procedure used for deriving such an element [Fig. 13.2(b)].

In all the examples of this section we shall approximate the frame displacements as

$$\mathbf{v} = \mathbf{N}_v \tilde{\mathbf{v}} \qquad \text{on} \qquad \Gamma_I \tag{13.15}$$

and consider the 'nodal forces' contributed by a single subdomain Ω^1 to the 'nodes' on this frame. Using virtual work (or weak) statements we have with discretization

$$\int_{\Gamma_t^1} \mathbf{N}_v^{\mathrm{T}} \mathbf{t} \, \mathrm{d}\Gamma = \mathbf{q}^1 \tag{13.16}$$

where $\mathbf{t}$ are the tractions the interior exerts on the imaginary frame and $\mathbf{q}^1$ are the nodal forces developed. The balance of the nodal forces contributed by each subdomain now provides the weak condition for traction continuity.

As finally the tractions $\mathbf{t}$ can be expressed in terms of the frame parameters $\tilde{\mathbf{v}}$ only, we shall arrive at

$$\mathbf{q}^1 = \mathbf{K}^1 \tilde{\mathbf{v}} + \mathbf{f}_0^1 \tag{13.17}$$

where $\mathbf{K}^1$ is the stiffness matrix of the subdomain Ω^1 and $\mathbf{f}_0^1$ its internally contributed 'forces'.

From this point onwards the standard assembly procedures are valid and the subdomain can be treated as a standard element which can be assembled with others by ensuring that

$$\sum_j \mathbf{q}^j = \mathbf{0} \tag{13.18}$$

where the sum includes all subdomains (elements!). We thus have only to consider a single subdomain in what follows.

13.4.2 Linking two or more mixed form subdomains

We shall assume as in Sec. 13.3 that in each subdomain, now labelled e for generality, the stresses $\boldsymbol{\sigma}^e$ and displacements $\mathbf{u}^e$ are independently approximated. The equations (13.11) are rewritten adding to the first the weak statement of displacement continuity.

We now have in place of (13.11a) and (13.13) (dropping superscripts)

$$\int_{\Omega^e} \delta\boldsymbol{\sigma}^{\mathrm{T}} (\mathbf{D}^{-1}\boldsymbol{\sigma} - \mathbf{S}\mathbf{u}) \, \mathrm{d}\Omega - \int_{\Gamma_{I^e}} \delta\mathbf{t}^{\mathrm{T}} (\mathbf{u} - \mathbf{v}) \, \mathrm{d}\Gamma = 0 \tag{13.19}$$

Equation (13.11b) will be rewritten as the weighted statement of the equilibrium relation, i.e.,

$$-\int_{\Omega^e} \delta\mathbf{u}^{\mathrm{T}} (\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma} + \mathbf{b}) \, \mathrm{d}\Omega + \int_{\Gamma_{t^e}} \delta\mathbf{u}^{\mathrm{T}} (\mathbf{t} - \bar{\mathbf{t}}) \, \mathrm{d}\Gamma = 0$$

or, after integration by parts

$$\int_{\Omega^e} \delta(\mathbf{S}\mathbf{u})^{\mathrm{T}} \boldsymbol{\sigma} \, \mathrm{d}\Omega - \int_{\Omega^e} \delta\mathbf{u}^{\mathrm{T}} \mathbf{b} \, \mathrm{d}\Omega - \int_{\Gamma_{I^e}} \delta\mathbf{u}^{\mathrm{T}} \mathbf{t} \, \mathrm{d}\Gamma - \int_{\Gamma_{t^e}} \delta\mathbf{u}^{\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d}\Gamma = 0 \tag{13.20}$$

In the above, $\mathbf{t}$ are the tractions corresponding to the stress field $\boldsymbol{\sigma}$ [see Eq. (11.30)]:

$$\mathbf{t} = \mathbf{G}\boldsymbol{\sigma} \tag{13.21}$$

In what follows Γ_{t^e}, i.e, the boundary with prescribed tractions, will generally be taken as zero.

On approximating Eqs (13.19), (13.20) and (13.16) with

$$\mathbf{u} = \mathbf{N}_u\tilde{\mathbf{u}} \qquad \boldsymbol{\sigma} = \mathbf{N}_\sigma\tilde{\boldsymbol{\sigma}} \qquad \text{and} \qquad \mathbf{v} = \mathbf{N}_\mathbf{v}\tilde{\mathbf{v}}$$

we can write, using Galerkin weighting and limiting the variables to the 'element' e,

$$\begin{bmatrix} \mathbf{A}^e & \mathbf{C}^e & \mathbf{Q}^e \\ \mathbf{C}^{e\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{Q}^{e\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}}^e \\ \tilde{\mathbf{u}}^e \\ \tilde{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \mathbf{f}^e \\ \mathbf{q}^e \end{Bmatrix} \tag{13.22a}$$

where

$$\begin{aligned} \mathbf{A}^e &= \int_{\Omega^e} \mathbf{N}_\sigma^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{N}_\sigma \,\mathrm{d}\Omega \\ \mathbf{C}^e &= \int_{\Omega^e} \mathbf{N}_\sigma^{\mathrm{T}}\mathbf{B} \,\mathrm{d}\Omega - \int_{\Gamma_{I^e}} (\mathbf{G}\mathbf{N}_\sigma)^{\mathrm{T}}\mathbf{N}_u \,\mathrm{d}\Gamma \\ \mathbf{Q}^e &= \int_{\Gamma_{I^e}} (\mathbf{G}\mathbf{N}_\sigma)^{\mathrm{T}}\mathbf{N}_v \,\mathrm{d}\Gamma \\ \mathbf{f}^e &= \int_{\Omega^e} \mathbf{N}_u^{\mathrm{T}}\mathbf{b} \,\mathrm{d}\Omega \end{aligned} \tag{13.22b}$$

Elimination of $\tilde{\boldsymbol{\sigma}}^e$ and $\tilde{\mathbf{u}}^e$ from the above yields the stiffness matrix of the element and the internally contributed force [see Eq. (13.17)].

Once again we can note that the simple stability criteria discussed in Chapter 11 will help in choosing the number of $\boldsymbol{\sigma}$, $\mathbf{u}$, and $\mathbf{v}$ parameters. As the final stiffness matrix of an element should be singular for three rigid body displacements we must have [by Eq. (11.18)]

$$n_\sigma \geqslant n_u + n_v - 3 \tag{13.23}$$

in two-dimensional applications.

Various alternative variational forms of the above formulation exist. A particularly useful one is developed by Pian *et al.*[4,5] In this the full mixed representation can be written completely in terms of a single variational principle (for zero body forces) and no boundary of type Γ_t present:

$$\Pi_\Omega = -\int_\Omega \tfrac{1}{2}\boldsymbol{\sigma}\mathbf{D}^{-1}\boldsymbol{\sigma} \,\mathrm{d}\Omega - \int_\Omega (\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma})^{\mathrm{T}}\mathbf{u}_I \,\mathrm{d}\Omega + \int_\Omega \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{S}\mathbf{v} \,\mathrm{d}\Omega \tag{13.24}$$

In the above it is assumed that the compatible field of $\mathbf{v}$ is *specified throughout the element* domain and not only on its interfaces and $\mathbf{u}_I$ stands for an incompatible field defined only inside the element domain.†

† In this form, of course, the element could well fit into Chapter 11 and the subdivision of hybrid and mixed forms is not unique here.

We note that in the present definition

$$\mathbf{u} = \mathbf{u}_I + \mathbf{v} \tag{13.25}$$

To show the validity of this variational principle, which is convenient as no interface integrals need to be evaluated, we shall derive the weak statement corresponding to Eqs (13.19) and (13.20) using the condition (13.25).

We can now write in place of (13.19) (noting that for interelement compatibility we have to ensure that $\mathbf{u}_I = \mathbf{0}$ on the interfaces)

$$\int_{\Omega^e} \delta\boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{D}^{-1}\boldsymbol{\sigma} - \mathbf{S}\mathbf{v})\,\mathrm{d}\Omega - \int_{\Omega^e} \delta\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{S}\mathbf{u}_I\,\mathrm{d}\Omega + \int_{\Gamma_{I^e}} \delta\mathbf{t}^{\mathrm{T}}\mathbf{u}_I\,\mathrm{d}\Gamma = 0 \tag{13.26}$$

After use of Green's theorem the above becomes simply

$$\int_{\Omega^e} \delta\boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{D}^{-1}\boldsymbol{\sigma} - \mathbf{S}\mathbf{v})\,\mathrm{d}\Omega + \int_{\Omega^e} (\mathbf{S}^{\mathrm{T}}\delta\boldsymbol{\sigma})^{\mathrm{T}}\mathbf{u}_I\,\mathrm{d}\Gamma = 0 \tag{13.27}$$

In place of (13.20) we write (in the absence of body forces $\mathbf{b}$ and boundary Γ_t)

$$\int_{\Omega^e} \delta\mathbf{u}_I^{\mathrm{T}}(\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma})\,\mathrm{d}\Omega + \int_{\Omega^e} \delta\mathbf{v}^{\mathrm{T}}(\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma})\,\mathrm{d}\Omega = 0 \tag{13.28}$$

and again after use of Green's theorem

$$\int_{\Omega^e} \delta\mathbf{u}_I^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_{\Omega^e} \delta(\mathbf{S}\mathbf{v})^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega = 0 \qquad (\text{if } \delta\mathbf{v} = 0 \quad \text{on } \Gamma_I) \tag{13.29}$$

These equations are precisely the variations of the functional (13.24).

Of course, the procedure developed in this section can be applied to other mixed or irreducible representations with 'frame' links. Tong and Pian[6,7] developed several alternative element forms by using this procedure.

13.4.3 Linking of equilibrating form subdomains

In this form we shall assume *a priori* that the stress field expansion is such that

$$\boldsymbol{\sigma}_T = \boldsymbol{\sigma} + \boldsymbol{\sigma}_0 \tag{13.30}$$

and that the equilibrium equations are identically satisfied. Thus

$$\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma} \equiv \mathbf{0};\ \mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma}_0 \equiv \mathbf{b} \text{ in } \Omega \qquad \text{and} \qquad \mathbf{G}\boldsymbol{\sigma} = \mathbf{0};\ \mathbf{G}\boldsymbol{\sigma}_0 = \bar{\mathbf{t}} \text{ on } \Gamma_{t^e}$$

In the absence of Γ_{t^e}, Eq. (13.20) is identically satisfied and we write (13.19) as (see Chapter 11, Sec. 11.7)

$$\begin{aligned}\int_{\Omega^e} \delta\boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{D}^{-1}\boldsymbol{\sigma}_T - \mathbf{S}\mathbf{u})\,\mathrm{d}\Omega + \int_{\Gamma_{I^e}} \delta\mathbf{t}^{\mathrm{T}}(\mathbf{u} - \mathbf{v})\,\mathrm{d}\Gamma \\ \equiv \int_{\Omega^e} \delta\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{D}^{-1}(\boldsymbol{\sigma} + \boldsymbol{\sigma}_0)\,\mathrm{d}\Omega - \int_{\Gamma_{I^e}} (\mathbf{G}\delta\boldsymbol{\sigma})^{\mathrm{T}}\mathbf{v}\,\mathrm{d}\Gamma = 0\end{aligned} \tag{13.31}$$

On discretization, noting that the field $\mathbf{u}$ does not enter the problem

$$\boldsymbol{\sigma} = \mathbf{N}_\sigma\tilde{\boldsymbol{\sigma}} \qquad \mathbf{v} = \mathbf{N}_v\tilde{\mathbf{v}}$$

we have, on including Eq. (13.16)

$$\begin{bmatrix} \mathbf{A}^e & \mathbf{Q}^e \\ \mathbf{Q}^{e\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\boldsymbol{\sigma}} \\ \tilde{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1^e \\ \mathbf{q}^e - \mathbf{f}_2^e \end{Bmatrix} \tag{13.32}$$

where

$$\mathbf{A}^e = \int_{\Omega^e} \mathbf{N}_\sigma \mathbf{D}^{-1} \mathbf{N}_\sigma \, \mathrm{d}\Omega \qquad \mathbf{f}_1 = \int_{\Omega^e} \mathbf{N}_\sigma \mathbf{D}^{-1} \boldsymbol{\sigma}_0 \, \mathrm{d}\Omega$$

$$\mathbf{Q}^e = \int_{\Gamma_{I^e}} (\mathbf{G}\mathbf{N}_\sigma)^{\mathrm{T}} \mathbf{N}_v \, \mathrm{d}\Gamma$$

and

$$\mathbf{f}_2^e = \int_{\Gamma_{I^e}} \mathbf{N}_v \mathbf{G} \boldsymbol{\sigma}_0 \, \mathrm{d}\Gamma$$

Here elimination of $\tilde{\boldsymbol{\sigma}}$ is simple and we can write directly

$$\mathbf{K}^e \tilde{\mathbf{v}} = \mathbf{q}^e - \mathbf{f}_2^e - \mathbf{Q}^{e\mathrm{T}} (\mathbf{A}^e)^{-1} \mathbf{f}_1^e \qquad \text{and} \qquad \mathbf{K}^e = \mathbf{Q}^{e\mathrm{T}} (\mathbf{A}^e)^{-1} \mathbf{Q}^e \tag{13.33}$$

In Sec. 11.7 we have discussed the possible equilibration fields and have indicated the difficulties in choosing such fields for a finite element, subdivided, field. In the present case, on the other hand, the situation is quite simple as the parameters describing the equilibrating stresses inside the element can be chosen arbitrarily in a polynomial expression.

For instance, if we use a simple polynomial expression in two dimensions:

$$\begin{aligned} \sigma_x &= \alpha_0 + \alpha_1 x + \alpha_2 y \\ \sigma_y &= \beta_0 + \beta_1 x + \beta_2 y \\ \tau_{xy} &= \gamma_0 + \gamma_1 x + \gamma_2 y \end{aligned} \tag{13.34}$$

we note that to satisfy the equilibrium we require

$$\mathbf{S}^{\mathrm{T}} \boldsymbol{\sigma} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & \dfrac{\partial}{\partial y} \\ 0 & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \boldsymbol{\sigma} = \begin{Bmatrix} \alpha_1 + \gamma_2 \\ \beta_2 + \gamma_1 \end{Bmatrix} = \mathbf{0} \tag{13.35}$$

and this simply means

$$\begin{aligned} \gamma_2 &= -\alpha_1 \\ \gamma_1 &= -\beta_2 \end{aligned}$$

Thus a linear expansion in terms of $9 - 2 = 7$ independent parameters is easily achieved. Similar expansions can of course be used with higher order terms.

It is interesting to observe that:

1. $n_\sigma \geqslant n_v - 3$ is needed to preserve stability.
2. By the principle of limitation, the accuracy of this approximation cannot be better than that achieved by a simple displacement formulation with compatible expansion of $\mathbf{v}$ throughout the element, providing similar polynomial expressions arise in stress component variations.

However, in practice two advantages of such elements, known as hybrid-stress elements, are obtained. In the first place it is not necessary to construct compatible displacement fields throughout the element (a point useful in their application to, say, a plate bending problem). In the second for distorted (isoparametric) elements it is easy to use stress fields varying with the global coordinates and thus achieve higher order accuracy.

The first use of such elements was made by Pian[8] and many successful variants are in use today.[9–22]

13.5 Linking of boundary (or Trefftz)-type solution by the 'frame' of specified displacements

We have already referred to boundary (Trefftz)-type solutions[23] earlier (Chapter 3). Here the chosen displacement/stress fields are such that *a priori* the homogeneous equations of equilibrium and constitutive relation are satisfied indentically in the domain under consideration (and indeed on occasion some prescribed boundary traction or displacement conditions).

Thus in Eqs (13.19) and (13.20) the subdomain (element e) Ω_e integral terms disappear and, as the internal $\delta\mathbf{t}$ and $\delta\mathbf{u}$ variations are linked, we combine all into a single statement (in the absence of body force terms) as

$$-\int_{\Gamma_{I^e}} \delta\mathbf{t}^{\mathrm{T}}(\mathbf{u}-\mathbf{v})\,\mathrm{d}\Gamma - \int_{\Gamma_{t^e}} \delta\mathbf{u}^{\mathrm{T}}(\mathbf{t}-\bar{\mathbf{t}})\,\mathrm{d}\Gamma = 0 \tag{13.36}$$

This coupled with the boundary statement (13.16) provides the means of devising stiffness matrix statements of such subdomains.

For instance, if we express the approximate fields as

$$\mathbf{u} = \mathbf{N}\tilde{\mathbf{a}} \tag{13.37}$$

implying

$$\boldsymbol{\sigma} = \mathbf{D}(\mathbf{SN})\tilde{\mathbf{a}} \qquad \text{and} \qquad \mathbf{t} = \mathbf{G}\boldsymbol{\sigma} = \mathbf{GD}(\mathbf{SN})\tilde{\mathbf{a}}$$

we can write in place of (13.22)

$$\begin{bmatrix} -\mathbf{H}^e & \mathbf{Q}^e \\ \mathbf{Q}^{e\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1^e \\ \mathbf{q} \end{Bmatrix} \tag{13.38}$$

where

$$\begin{aligned} \mathbf{H}^e &= \int_{\Gamma_{I^e}} [\mathbf{GD}(\mathbf{SN})]^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Gamma + \int_{\Gamma_{t^e}} \mathbf{N}^{\mathrm{T}}\mathbf{GD}(\mathbf{SN})\,\mathrm{d}\Gamma \\ \mathbf{Q}^e &= \int_{\Gamma_{I^e}} [\mathbf{GD}(\mathbf{SN})]^{\mathrm{T}}\mathbf{N}_v\,\mathrm{d}\Gamma \\ \mathbf{f}_1^e &= -\int_{\Gamma_{t^e}} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \end{aligned} \tag{13.39}$$

In Eqs (13.38) and (13.39) we have omitted the domain integral of the particular solution $\boldsymbol{\sigma}_0$ corresponding to the body forces $\mathbf{b}$ but have allowed a portion of the

boundary Γ_{t^e} to be subject to prescribed tractions. Full expressions including the particular solution can easily be derived.

Equation (13.38) is immediately available for solution of a single boundary problem in which $\mathbf{v}$ and $\bar{\mathbf{t}}$ are described on portions of the boundary. More importantly, however, it results in a very simple stiffness matrix for a full element enclosed by the frame. We now have

$$\mathbf{K}^e\tilde{\mathbf{v}} = \mathbf{q} - \mathbf{f}^e \tag{13.40}$$

in which

$$\begin{aligned} \mathbf{K}^e &= \mathbf{Q}^{e\mathrm{T}}(\mathbf{H}^e)^{-1}\mathbf{Q}^e \\ \mathbf{f}^e &= \mathbf{Q}^{e\mathrm{T}}(\mathbf{H}^e)^{-1}\mathbf{f}_1^e \end{aligned} \tag{13.41}$$

This form is very similar to that of Eq. (13.33) except that now only integrals on the boundaries of the subdomain element need to be evaluated.

Much has been written about so-called 'boundary elements' and their merits and disadvantages.[24–36] Very frequently singular Green's functions are used to satisfy the governing field equations in the domain.[31–35] The singular function distributions used do not lend themselves readily to the derivation of symmetric coupling forms of the type given in Eq. (13.38). Zienkiewicz *et al.*[36–39] show that it is possible to obtain symmetry at a cost of two successive integrations. Further it should be noted that the singular distributions always involve difficult integration over a point of singularity and special procedures need to be used for numerical implementation. For this reason the use of generally non-singular Trefftz functions is preferable and it is possible to derive complete sets of functions satisfying the governing equations without introducing singularities,[36–39] and simple integration then suffices.

While boundary solutions are confined to linear homogeneous domains these give very accurate solutions for a limited range of parameters, and their combination with 'standard' finite elements has been occasionally described. Several coupling procedures have been developed in the past,[36–39] but the form given here coincides with the work of Zielinski and Zienkiewicz,[40] Jirousek[41–44] and Piltner.[45] Jirousek *et al.* have developed very general two-dimensional elasticity and plate bending elements which can be enclosed by a many-sided polygonal domain (element) that can be directly coupled to standard elements providing that same-displacement interpolation along the edges is involved, as shown in Fig. 13.3. Here both *interior* elements with a frame enclosing an element volume and *exterior* elements satisfying tractions at free surface and infinity are illustrated.

Rather than combining in a finite element mesh the standard and the Trefftz-type elements ('T-elements'[28]) it is often preferable to use the T-elements alone. This results in the whole domain being discretized by elements of the same nature and offering each about the same degree of accuracy. The subprogram of such elements can include an arsenal of homogeneous 'shape functions' $\mathbf{N}^e$ [see Eq. (13.37)] which are exact solutions to different types of singularities as well as those which automatically satisfy traction boundary conditions on internal boundaries, e.g., circles or ellipses inscribed within large elements as shown in Fig. 13.4. Moreover, by com-

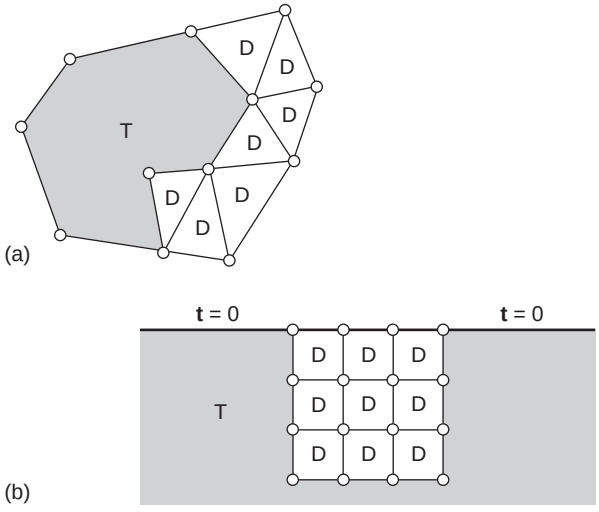


Fig. 13.3 Boundary–Trefftz-type elements (T) with complex-shaped 'frames' allowing combination with standard, displacement elements (D): (a) an *interior* element; (b) an *exterior* element.

pleting the set of homogeneous shape functions by suitable 'load terms' representing the non-homogeneous differential equation solution, $\mathbf{u}_0$, one may account accurately for various discontinuous or concentrated loads without laborious adjustment of the finite element mesh.

Clearly such elements can perform very well when compared with standard ones, as the nature of the analytical solution has been essentially included. Figure 13.5 shows

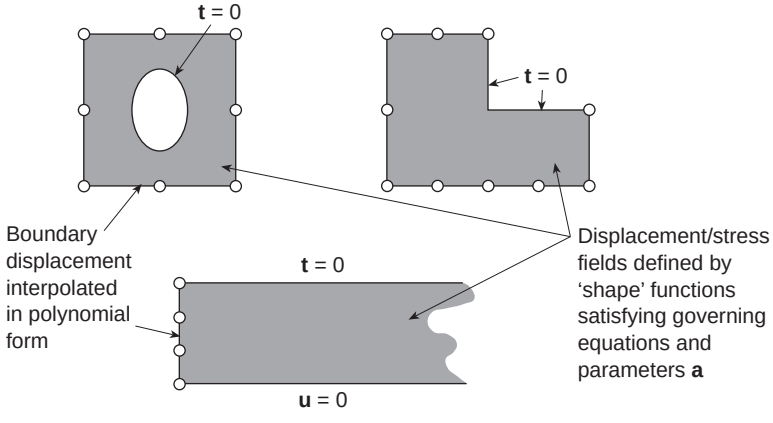
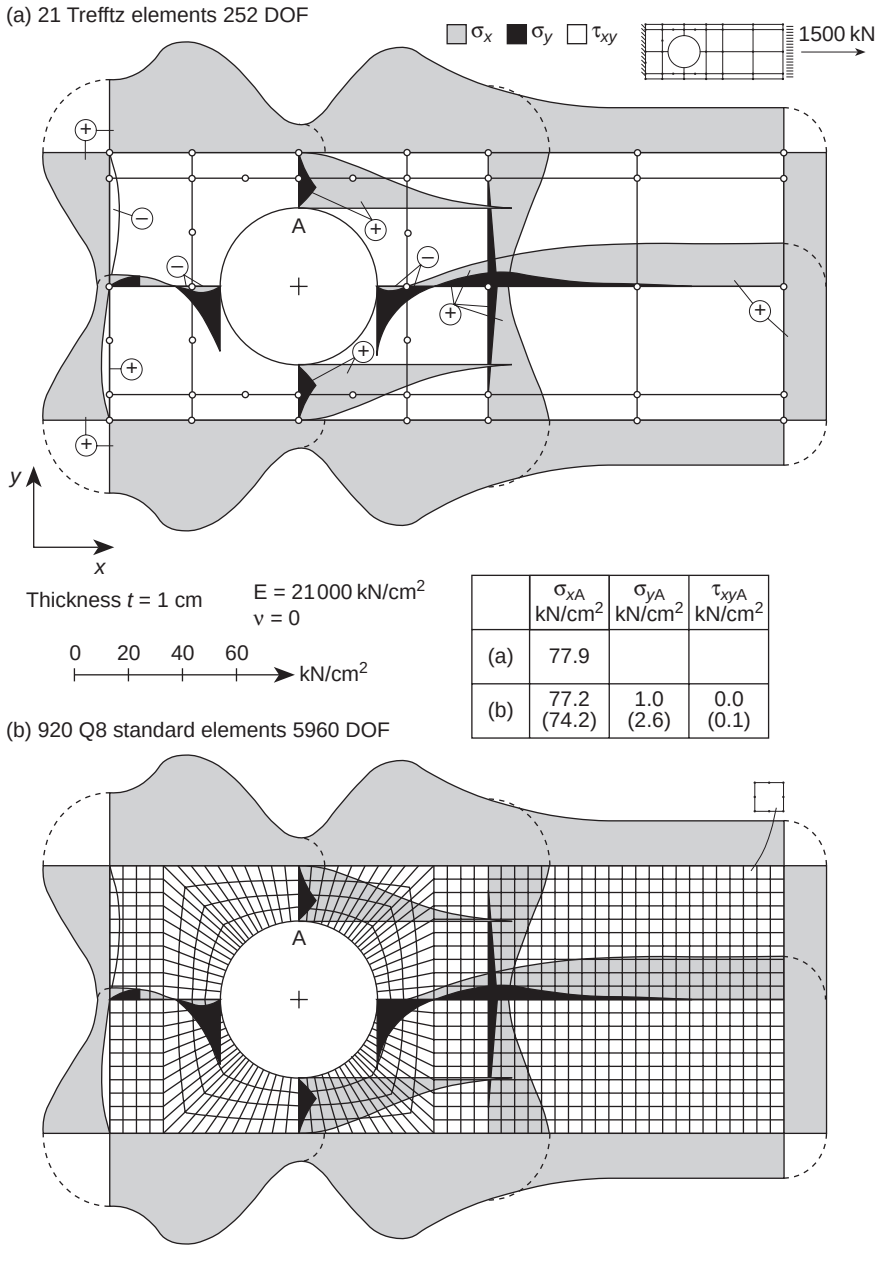


Fig. 13.4 Boundary–Trefftz-type elements. Some useful general forms.[43]



	σ_{xA} kN/cm^2	σ_{yA} kN/cm^2	τ_{xyA} kN/cm^2
(a)	77.9		
(b)	77.2 (74.2)	1.0 (2.6)	0.0 (0.1)

Fig. 13.5 Application of Trefftz-type elements to a problem of a plane-stress tension bar with a circular hole. (a) Trefftz element solution. (b) Standard displacement element solution. (Numbers in parentheses indicate standard solution with 230 elements, 1600 DOF).

excellent results which can be obtained using such complex elements. The number of degrees of freedom is here much smaller than with a standard displacement solution but, of course, the bandwidth is much larger.[43]

Two points come out clearly in the general formulation of Eqs (13.36)–(13.39).

First, the displacement field, $\mathbf{u}$ given by parameters $\tilde{\mathbf{a}}$, can only be determined by excluding any rigid body modes. These can only give strains $\mathbf{SN}$ identically equal to zero and hence make no contribution to the $\mathbf{H}$ matrix.

Second, stability conditions require that (in two dimensions)

$$n_a \geqslant n_v - 3$$

and thus the minimum n_a can be readily found (viz. Chapter 11). Once again there is little point in increasing the number of internal parameters substantially above the minimum number as additional accuracy may not be gained.

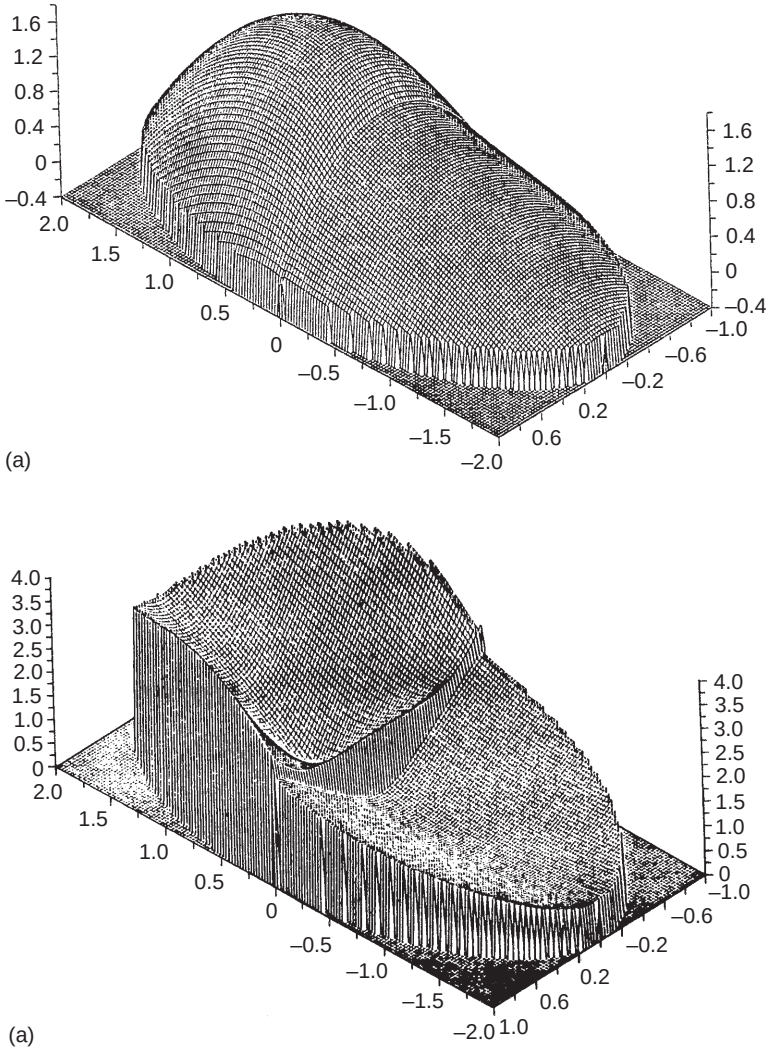


Fig. 13.6 Boundary–Trefftz-type 'elements' linking two domains of different materials in an elliptic bar subject to torsion (Poisson equations).[40] (a) Stress function given by internal variables showing almost complete continuity. (b) x component of shear stress (gradient of stress function showing abrupt discontinuity of material junction).

We have said earlier that the 'translation' of the formulation discussed to problems governed by the quasi-harmonic equations is almost evident. Now identical relations will hold if we replace

$$\begin{aligned} \mathbf{u} &\rightarrow \phi \\ \boldsymbol{\sigma} &\rightarrow \mathbf{q} \\ \mathbf{t} &\rightarrow q_n \\ \mathbf{S} &\rightarrow \nabla \end{aligned} \tag{13.42}$$

For the Poisson equation

$$\nabla^2 \phi = Q \tag{13.43}$$

a complete series of analytical solutions in two dimensions can be written as

$$\begin{aligned} &\mathrm{Re}\,(z^n) = 1, x, x^2 - y^2, x^3 - 3xy^3, \ldots \\ &\mathrm{Im}\,(z^n) = y, 2xy, \ldots \end{aligned} \qquad \text{for } z = x + \mathrm{i}y \tag{13.44}$$

With the above we get

$$\mathbf{N}^e = \left[1, \quad x, \quad y, \quad x^2 - y^2, \quad 2xy, \quad x^3 - 3xy^2, \quad 3x^2 y, \quad \ldots\right] \tag{13.45}$$

A simple solution involving two subdomains with constant but different values of Q and a linking on the boundary is shown in Fig. 13.6, indicating the accuracy of the linking procedures.

13.6 Subdomains with 'standard' elements and global functions

The procedure just described can be conveniently used with approximations made internally with standard (displacement) elements and global functions helping to deal with singularities or other internal problems. Now simply an additional term will arise inside nodes placed internally in the subdomain but the effect of global functions can be contained inside the subdomain. The formulation is somewhat simpler as complicated Trefftz-type functions need not be used.

We leave details to the reader and in Fig. 13.7 show some possible, useful subdomain assemblies. We shall return to this again in Chapter 16.

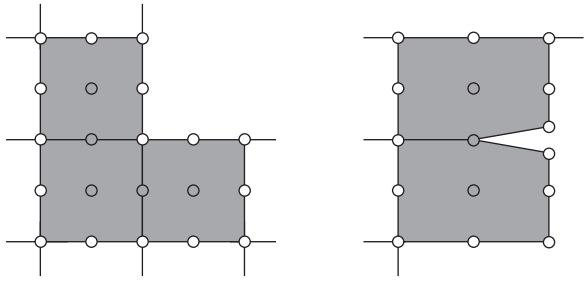


Fig. 13.7 'Superelements' built from assembly of standard displacement elements with global functions eliminating singularities confined to the assembly.

13.7 Lagrange variables or discontinuous Galerkin methods?

In all of the preceding examples we have linked the various element subdomains by a line on which the additional Lagrange multipliers have been specified. These multipliers could well be displacements or tractions which in fact were the same variables as those inside the element domain.

The lagrangian variables which are so identified can be directly substituted in terms of the variables given inside each subdomain. For instance the interface displacement can be reproduced as the average displacement of those given in each subdomain

$$\mathbf{u} = \tfrac{1}{2}(\mathbf{u}_1 + \mathbf{u}_2)$$

The total number of variables occurring in the problem is thus reduced (though now element variables have to be carried in the solution and the solution cost may well be increased). The idea was first used by Kikuchi and Ando[46] who used it to improve the performance of non-conforming plate bending elements.

Recently a revival of such methods has taken place. The basic idea appear to be presented by Makridakis and Babuška *et al.*[47] and in the context of a 'discontinuous Galerkin method' is demonstrated by Oden and co-workers.[48–50] We shall refer to the discontinuous Galerkin method in Volume 3 when dealing with convection dominated problems and in a different context in Sec. 18.6 of Chapter 18 for discrete time approximation problems. The process has practical advantages such as:

1. different local interpolations can be used;
2. the stress (flux) continuity is preserved on each individual element.

We shall discuss these properties further when we address the method in Volume 3.

13.8 Concluding remarks

The possibilities of elements of 'superelements' constructed by the mixed-incomplete field methods of this chapter are very numerous. Many have found practical use in existing computer codes as 'hybrid elements'; others are only now being made widely available. The use of a frame of specified displacements is only one of the possible methods for linking Trefftz-type solutions. As an alternative, a frame of specified boundary tractions $\mathbf{t}$ has also been successfully investigated.[26,30] In addition, the so-called 'frameless formulation'[27,29] has been found to be another efficient solution (for a review see reference 28) in the Trefftz-type element approach. All of the above mentioned alternative approaches may be implemented into standard finite element computer codes. Much further research will elucidate the advantages of some of the forms discovered and we expect the use of such developments to continue to increase in the future.

References

1. N.E. Wiberg. Matrix structural analysis with mixed variables. *Int. J. Num. Meth. Eng.*, **8**, 167–94, 1974.
2. G. Kron. *Tensor Analysis of Networks*. John Wiley & Sons, New York, 1939.
3. Ch. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Intern. J. Num. Meth. Engng*, **32**, 1205–27, 1991.
4. T.H.H. Pian and D.P. Chen. Alternative ways for formulation of hybrid elements. *Int. J. Num. Meth. Eng.*, **18**, 1679–84, 1982.
5. T.H.H. Pian, D.P. Chen, and D. Kong. A new formulation of hybrid/mixed finite elements. *Comp. Struct.*, **16**, 81–7, 1983.
6. P. Tong. A family of hybrid elements. *Int. J. Num. Meth. Eng.*, **18**, 1455–68, 1982.
7. T.H.H. Pian and P. Tong. Relations between incompatible displacement model and hybrid strain model. *Int. J. Num. Meth. Eng.*, **22**, 173–181, 1986.
8. T.H.H. Pian. Derivation of element stiffness matrices by assumed stress distributions. *JAIAA*, **2**, 1333–5, 1964.
9. S.N. Atluri, R.H. Gallagher, and O.C. Zienkiewicz (Eds). *Hybrid and Mixed Finite Element Methods*. Wiley, 1983.
10. T.H.H. Pian. Element stiffness matrices for boundary compatibility and for prescribed boundary stresses. *Proc. Conf. Matrix Methods in Structural Mechanics*. AFFDL-TR-66-80, pp. 457–78, 1966.
11. R.D. Cook and J. At-Abdulla. Some plane quadrilateral 'hybrid' finite elements. *JAIAA*, **7**, 1969.
12. T.H. Pian and P. Tong. Basis of finite element methods for solid continua. *Int. J. Num. Meth. Eng.*, **1**, 3–28, 1969.
13. S.N. Atluri. A new assumed stress hydrid finite element model for solid continua. *JAIAA*, **9**, 1647–9, 1971.
14. R.D. Henshell. On hybrid finite elements, in *The Mathematics of Finite Elements and Applications* (ed. J.R. Whiteman), pp. 299–312, Academic Press, 1973.
15. R. Dungar and R.T. Severn. Triangular finite elements of variable thickness. *J. Strain Analysis*, **4**, 10–21, 1969.
16. R.J. Allwood and G.M.M. Cornes. A polygonal finite element for plate bending problems using the assumed stress approach. *Int. J. Num. Meth. Eng.*, **1**, 135–49, 1969.
17. T.H.H. Pian. Hybrid models, in *Numerical and Computer Methods in Applied Mechanics* (eds S.J. Fenves *et al.*). Academic Press, 1971.
18. Y. Yoshida. A hybrid stress element for thin shell analysis, in *Finite Element Methods in Engineering* (eds V. Pulmano and A. Kabaila), pp. 271–86, University of New South Wales, Australia, 1974.
19. R.D. Cook and S.G. Ladkany. Observations regarding assumed-stress hybrid plate elements. *Int. J. Num. Meth. Eng.*, **8**(3), 513–20, 1974.
20. J.P. Wolf. Generated hybrid stress finite element models. *JAIAA*, **11**, 1973.
21. P.L. Gould and S.K. Sen. Refined mixed method finite elements for shells of revolution. *Proc. 3rd Air Force Conf. Matrix Methods in Structural Mechanics*. Wright-Patterson AF Base, Ohio, 1971.
22. P. Tong. New displacement hybrid finite element models for solid continua. *Int. J. Num. Meth. Eng.*, **2**, 73–83, 1970.
23. E. Trefftz. Ein Gegenstruck zum Ritz'schem Verfohren. *Proc. 2nd Int. Cong. Appl. Mech.* Zurich, 1926.
24. P.K. Banerjee and R. Butterfield. *Boundary Element Methods in Engineering Science*. McGraw-Hill, London and New York, 1981.

25. J.A. Ligget and P.L-F. Liu. *The Boundary Integral Equation Method for Porous Media Flow*. Allen and Unwin, London, 1983.
26. C.A. Brebbia and S. Walker. *Boundary Element Technique in Engineering*. Newnes-Butterworth, London, 1980.
27. J. Jirousek and A. Wróblewski. Least-squares T-elements: Equivalent FE and BE forms of a substructure-oriented boundary solution approach. *Comm. Num. Meth. Eng.*, **10**, 21–32, 1994.
28. J. Jirousek and A. Wróblewski. T-elements: State of the art and future trends. *Arch. Comp. Meth. Eng.*, **3**(4), 1996.
29. J. Jirousek and A.P. Zieliński. Study of two complementary hybrid-Trefftz p-element formulations, in *Numerical Methods in Engineering 92*, 583–90. Elsevier, 1992.
30. J. Jirousek and A.P. Zieliński. Dual hybrid-Trefftz element formulation based on independent boundary traction frame. *Internat. J. Num. Meth. Eng.*, **36**, 2955–80, 1993.
31. I. Herrera. Boundary methods: a criteria for completeness. *Proc. Nat. Acad. Sci. USA*, **77**(8), 4395–8, August 1980.
32. I. Herrera. Boundary methods for fluids, Chapter 19 of *Finite Elements in Fluids*. Vol. 4 (eds R.H. Gallagher, H.D. Norrie, J.T. Oden, and O.C. Zienkiewicz). Wiley, New York, 1982.
33. I. Herrera. Trefftz method, in *Progress in Boundary Element Methods*. Vol. 3 (ed. C.A. Brebbia). Wiley, New York, 1983.
34. I. Herrera and H. Gourgeon. Boundary methods, *C*-complete system for Stokes problems. *Comp. Meth. Appl. Mech. Eng.*, **30**, 225–44, 1982.
35. I. Herrera and F.J. Sabina. Connectivity as an alternative to boundary integral equations: construction of bases. *Proc. Nat. Acad. Sci. USA*, **75**(5), 2059–63, May 1978.
36. O.C. Zienkiewicz, D.W. Kelly, and P. Bettess. The coupling of the finite element method and boundary solution procedures. *Int. J. Num. Meth. Eng.*, **11**, 355–75, 1977.
37. O.C. Zienkiewicz, D.W. Kelly, and P. Bettess. Marriage a la mode – the best of both worlds (finite elements and boundary integrals). Chapter 5 of *Energy Methods in Finite Element Analysis* (eds R. Glowinski, E.Y. Rodin, and O.C. Zienkiewicz), pp. 81–107, Wiley, London and New York, 1979.
38. O.C. Zienkiewicz and K. Morgan. *Finite Elements and Approximation*. Wiley, London and New York, 1983.
39. O.C. Zienkiewicz. The generalized finite element method – state of the art and future directions. *J. Appl. Mech.* 50th anniversary issue, 1983.
40. A.P. Zielinski and O.C. Zienkiewicz. Generalized finite element analysis with *T* complete boundary solution functions. *Int. J. Num. Mech. Eng.*, **21**, 509–28, 1985.
41. J. Jirousek. A powerful finite element for plate bending. *Comp. Meth. Appl. Mech. Eng.*, **12**, 77–96, 1977.
42. J. Jirousek. Basis for development of large finite elements locally satisfying all field equations. *Comp. Meth. Appl. Mech. Eng.*, **14**, 65–92, 1978.
43. J. Jirousek and P. Teodorescu. Large finite elements for the solution of problems in the theory of elasticity. *Comp. Struct.*, **15**, 575–87, 1982.
44. J. Jirousek and Lan Guex. The hybrid Trefftz finite element model and its application to plate bending. *Int. J. Num. Mech. Eng.*, **23**, 651–93, 1986.
45. R. Piltner. Special elements with holes and internal cracks. **21**, 1471–85, 1985.
46. F. Kikichi and Y. Ando. A new variational functional for the finite-element method and its application to plate and shell problems. *Nucl. Engng and Design*, **21**(1), 95–113, 1972.
47. C.G. Makridakis and I. Babuška. On the stability of the discontinuous Galerkin method for the heat equation. *SIAM J. Num. Anal.*, **34**, 389–401, 1997.
48. J.T. Oden, I. Babuška, and C.E. Baumann. A discontinuous *hp* finite element method for diffusion problems. *J. Comp. Physics*, **146**(2), 491–519, 1998.

49. J.T. Oden and C.E. Baumann. A discontinuous *hp* finite element method for convection-diffusion problems. *Comput. Meth. Appl. Mech. Engng*, **175**(3-4), 311–41, 1999.
50. C.E. Baumann and J.T. Oden. A discontinuous *hp* finite element method for convection-diffusion problems. *Comput. Meth. Appl. Mech. Engng*, **175**, 311–41, 1999.

14

Errors, recovery processes and error estimates

14.1 Definition of errors

We have stressed from the beginning of this book the approximate nature of the finite element method and on many occasions to show its capabilities we have compared it with exact solutions when these were known. Also on many occasions we have spoken about the 'accuracy' of the procedures we suggested and discussed the manner by which this accuracy could be improved. Indeed one of the objectives of this chapter is concerned with the question of accuracy and a possible improvement on it by an *a posteriori* treatment of the finite element data. We refer to such processes as *recovery*. We shall also consider the discretization error of the finite element approximation and *a posteriori* estimates of such error. In particular, we describe two distinct types of error estimators, *recovery based error estimators* and *residual based error estimators*. The critical role that the recovery processes play in the computation of these error estimators will be discussed.

Before proceeding further it is necessary to define what we mean by error. This we consider to be the difference between the exact solution and the approximate one. This can apply to the basic function, such as displacement which we have called $\mathbf{u}$ and can be given as

$$\mathbf{e} = \mathbf{u} - \hat{\mathbf{u}} \tag{14.1}$$

In a similar way, however, we could focus on the error in the strains (i.e., gradients in the solution), such as $\boldsymbol{\varepsilon}$ or stresses $\boldsymbol{\sigma}$ and describe an error in those quantities as

$$\mathbf{e}_{\varepsilon} = \boldsymbol{\varepsilon} - \hat{\boldsymbol{\varepsilon}} \tag{14.2}$$

$$\mathbf{e}_{\sigma} = \boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}} \tag{14.3}$$

The specification of local error in the manner given in Eqs (14.1)–(14.3) is generally not convenient and occasionally misleading. For instance, under a point load both errors in displacements and stresses will be locally infinite but the overall solution may well be acceptable. Similar situations will exist near re-entrant corners where, as is well known, stress singularities exist in elastic analysis and gradient singularities develop in field problems. For this reason various 'norms' representing some integral scalar quantity are often introduced to measure the error.

If, for instance, we are concerned with a general linear equation of the form of Eq. (3.6) (cf. Chapter 3), i.e.,

$$\mathbf{L}\mathbf{u} + \mathbf{p} = \mathbf{0} \tag{14.4}$$

we can define an *energy norm* written for the error as

$$||\mathbf{e}|| = \left[\int_\Omega \mathbf{e}^{\mathrm{T}}\mathbf{L}\mathbf{e}\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \equiv \left[\int_\Omega (\mathbf{u}-\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{L}(\mathbf{u}-\hat{\mathbf{u}})\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \tag{14.5}$$

This scalar measure corresponds in fact to the square root of the quadratic functional such as we have discussed in Sec. 3.8 of Chapter 3 and where we sought its minimum in the case of a self-adjoint operator $\mathbf{L}$.

For elasticity problems the energy norm is identically defined and yields,

$$||\mathbf{e}|| = \left[\int_\Omega (\mathbf{S}\mathbf{e})^{\mathrm{T}}\mathbf{D}\mathbf{S}\mathbf{e}\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \tag{14.6}$$

(with symbols as used in Chapter 2).

Here $\mathbf{e}$ is given by Eq. (14.1) and the operator $\mathbf{S}$ defines the strains as

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u} \qquad \text{and} \qquad \hat{\boldsymbol{\varepsilon}} = \mathbf{S}\hat{\mathbf{u}} \tag{14.7}$$

and $\mathbf{D}$ is the elasticity matrix (see Chapter 2), giving the stress as

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \qquad \text{and} \qquad \hat{\boldsymbol{\sigma}} = \mathbf{D}\hat{\boldsymbol{\varepsilon}} \tag{14.8}$$

in which for simplicity we ignore initial stresses and strains.

The energy norm of Eq. (14.6) can thus be written alternatively as

$$\begin{aligned} ||\mathbf{e}|| &= \left[\int_\Omega (\boldsymbol{\varepsilon}-\hat{\boldsymbol{\varepsilon}})^{\mathrm{T}}\mathbf{D}(\boldsymbol{\varepsilon}-\hat{\boldsymbol{\varepsilon}})\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \\ &= \left[\int_\Omega (\boldsymbol{\varepsilon}-\hat{\boldsymbol{\varepsilon}})^{\mathrm{T}}(\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}})\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \\ &= \left[\int_\Omega (\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}})^{\mathrm{T}}\mathbf{D}^{-1}(\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}})\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \end{aligned} \tag{14.9}$$

and its relation to strain energy is evident.

Other scalar norms can easily be devised. For instance, the L_2 norm of displacement and stress error can be written as

$$||\mathbf{e}||_{L_2} = \left[\int_\Omega (\mathbf{u}-\hat{\mathbf{u}})^{\mathrm{T}}(\mathbf{u}-\hat{\mathbf{u}})\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \tag{14.10}$$

$$||\mathbf{e}_\sigma||_{L_2} = \left[\int_\Omega (\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}})^{\mathrm{T}}(\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}})\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \tag{14.11}$$

Such norms allow us to focus on the particular quantity of interest and indeed it is possible to evaluate 'root mean square' (RMS) values of its error. For instance, the RMS error in displacement, $\Delta\mathbf{u}$, becomes for the domain Ω

$$|\Delta\mathbf{u}| = \left(\frac{||\mathbf{e}||^2_{L_2}}{\Omega}\right)^{\frac{1}{2}} \tag{14.12}$$

Similarly, the RMS error in stress, $\Delta\sigma$, becomes for the domain Ω

$$|\Delta\sigma| = \left(\frac{||\mathbf{e}_\sigma||^2_{L_2}}{\Omega} \right)^{\frac{1}{2}} \tag{14.13}$$

Any of the above norms can be evaluated over the whole domain or over subdomains or even individual elements.

We note that

$$||\mathbf{e}||^2 = \sum_{i=1}^{m} ||\mathbf{e}||_i^2 \tag{14.14}$$

where i refers to individual elements Ω_i such that their sum (union) is Ω.

We note further that the energy norm given in terms of the stresses, the L_2 stress norm and the RMS stress error have a very similar structure and that these are similarly approximated.

At this stage it is of interest to invoke the discussion of Chapter 2 (Sec. 2.6) concerning the rates of convergence. We noted there that with trial functions in the displacement formulation of degree p, the errors in the stresses were of the order $O(h^p)$. This order of error should therefore apply to the energy norm error $||\mathbf{e}||$. While the arguments are correct for well-behaved problems with no singularity, it is of interest to see how the above rule is violated when singularities exist.

To describe the behaviour of stress analysis problems we define the variation of the *relative energy norm error* (percentage) as

$$\eta = \frac{||\mathbf{e}||}{||\mathbf{u}||} \times 100\% \tag{14.15}$$

where

$$||\mathbf{u}|| = \left(\int_\Omega \boldsymbol{\varepsilon}^{\mathrm{T}} \mathbf{D} \boldsymbol{\varepsilon} \, \mathrm{d}\Omega \right)^{\frac{1}{2}} \tag{14.16}$$

is the energy norm of the solution. In Figs 14.1 and 14.2 we consider two similar stress analysis problems, in the first of which a strong singularity is, however, present. In both figures we show the relative energy norm error for an h refinement constructed by uniform subdivision of the initial mesh and of a p refinement in which polynomial order is increased throughout the original mesh.

We note two interesting facts. First, the h convergence rates for various polynomial orders of the shape functions are nearly the same in the example with singularity (Fig. 14.1) and are well below the theoretically predicted optimal order $O(h^p)$, [or $O(\mathrm{NDF})^{-p/2}$ as the NDF (number of degrees of freedom) is approximately inversely proportional to h^2 for a two-dimensional problem].

Secondly, in the case shown in Fig. 14.2, where the singularity is avoided by rounding the corner, the convergence rates improve for elements of higher order, though again the theoretical (asymptotic) rates are not achieved.

The reason for this behaviour is clearly the singularity, and in general it can be shown that the rate of convergence for problems with singularity is

$$O(\mathrm{NDF})^{-[\min(\lambda, p)]/2} \tag{14.17}$$

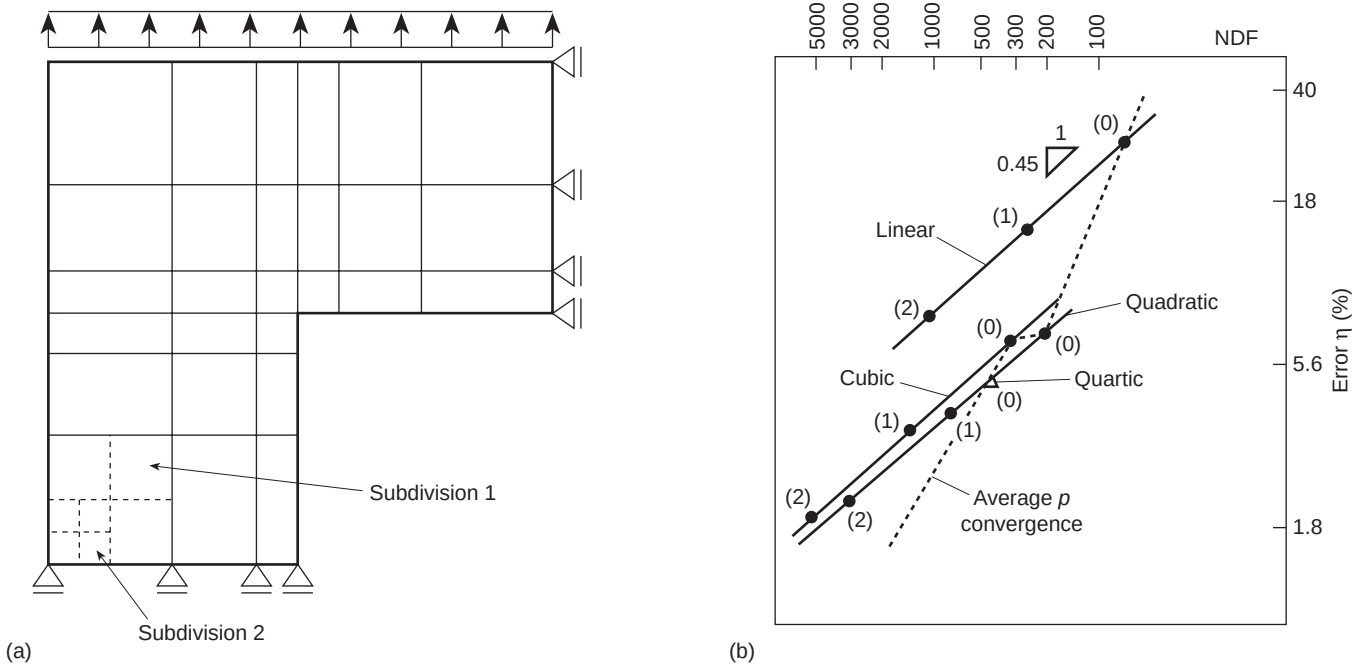


Fig. 14.1 Analysis of L-shaped domain with singularity.

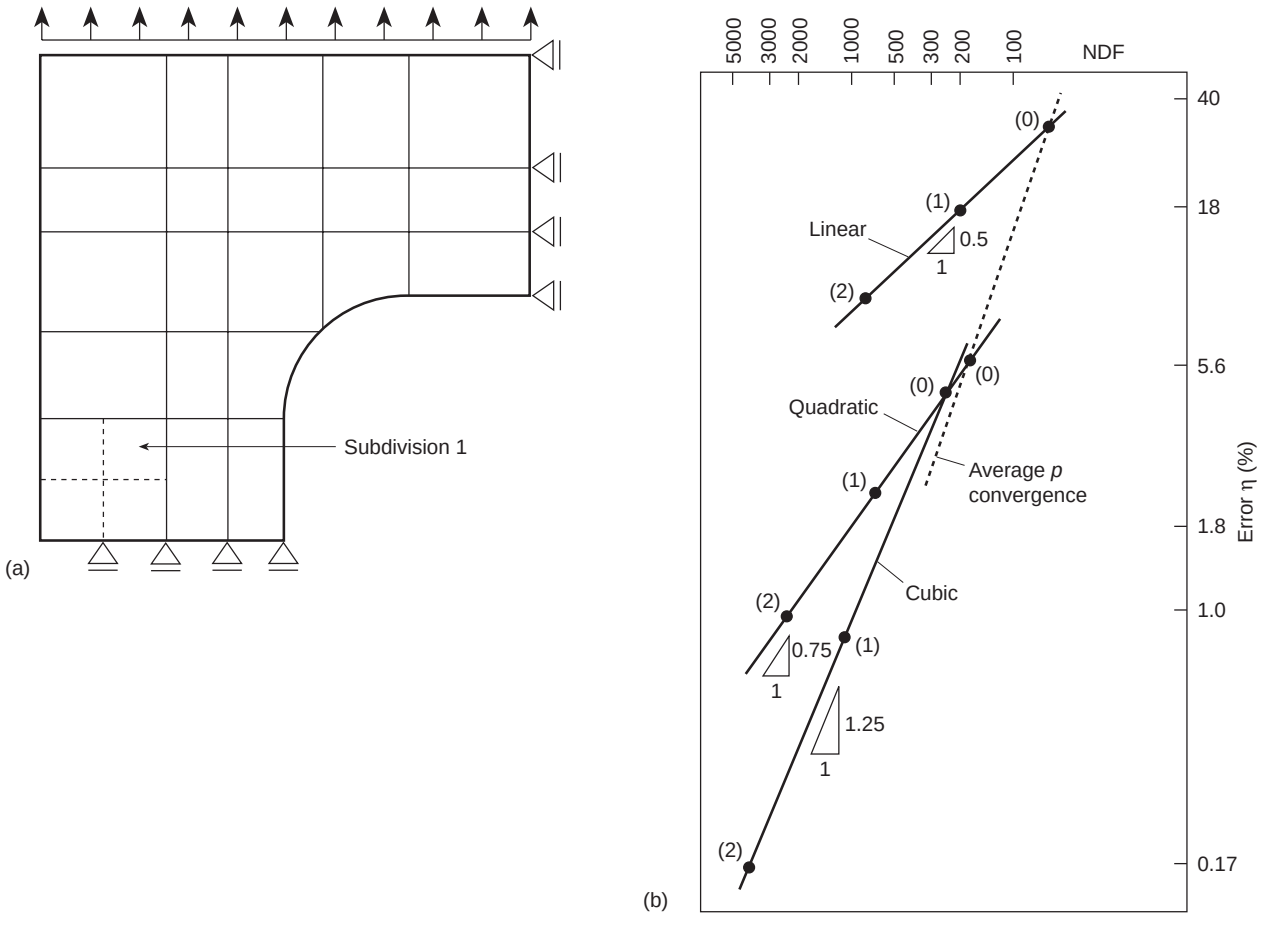


Fig. 14.2 Analysis of L-shaped domain without singularity.

where λ is a number associated with the intensity of the singularity. For elasticity problems λ ranges from 0.5 for a nearly closed crack to 0.71 for a 90° corner. The rate of convergence illustrated in Fig. 14.2 approaches the value controlled by the singularity for all values of p used in the elements.

14.2 Superconvergence and optimal sampling points

In this section we shall consider the matter of points at which the stresses, or displacements, give their most accurate values in typical problems of a self-adjoint kind. We shall note that on many occasions the displacements, or the function itself, are most accurately sampled at the nodes defining an element and that the gradients or stresses are best sampled at some interior points. Indeed in one dimension at least we shall find that such points often exhibit the quality known as *superconvergence* (i.e., the values sampled at these points show an error which decreases more rapidly than elsewhere). Obviously, the user of finite element analysis should be encouraged to employ such points but at the same time note that the errors overall may be much larger. To clarify ideas we shall start with a typical problem of second order in one dimension.

14.2.1 A one-dimensional example

Here we consider a problem of a second-order equation such as we have frequently discussed in Chapter 3 and which may be typical of either one-dimensional heat conduction or the displacements of an elastic bar with varying cross-section. This equation can readily be written as

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(k\frac{\mathrm{d}u}{\mathrm{d}x}\right)+\beta u+Q=0 \tag{14.18}$$

with the boundary conditions either defining the values of the function u or of its gradients at the ends of the domain.

Let us consider a typical problem shown in Fig. 14.3. Here we show an exact solution for u and $\mathrm{d}u/\mathrm{d}x$ for a span of several elements and indicate the type of solution which will result from a finite element calculation using linear elements. We have already noted that on occasions we shall obtain exact solutions for u at nodes (see Fig. 3.4). This will happen when the shape functions contain the exact solution of the homogeneous differential equation (Appendix H) – a situation which happens for Eq. (14.18) when $\beta = 0$ and polynomial shape functions are used. In all cases, even when β is non-zero and linear shape functions are used, the nodal values generally will be much more accurate than those elsewhere, Fig. 14.3(a). For the gradients shown in Fig. 14.3(b) we observe large discrepancies of the finite element solution from the exact solution but we note that somewhere within each element the results are nearly exact.

It would be useful to locate such points and indeed we have already remarked in the context of two-dimensional analysis that values obtained within the elements tend to be more accurate for gradients (strains and stresses) than those values calculated at

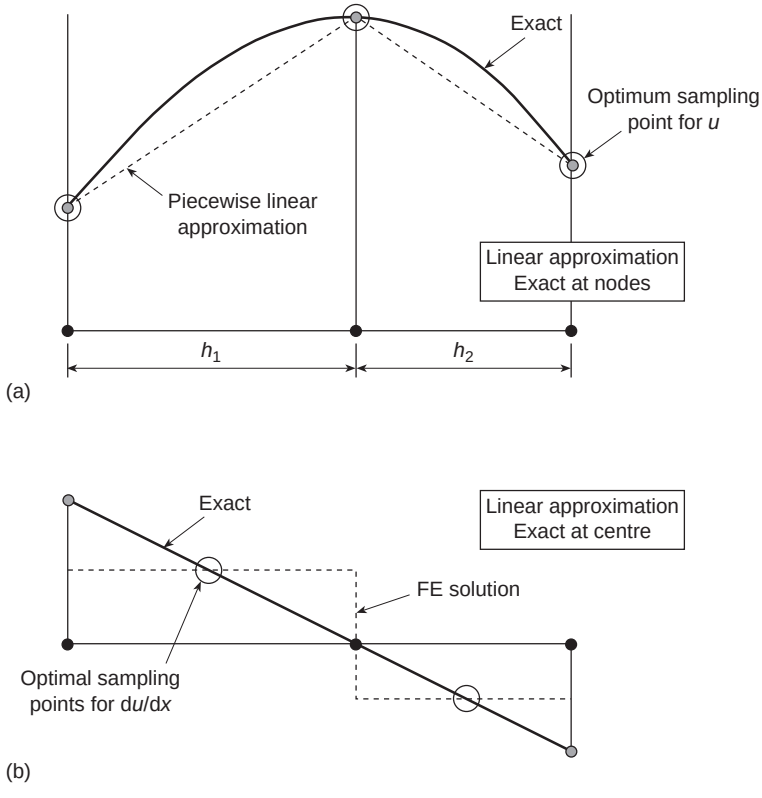


Fig. 14.3 Optimal sampling points for the function (a) and its gradient (b) in one dimension (linear elements).

nodes. Clearly, for the problem illustrated in Fig 14.3(b) we should sample somewhere near the centre of each element.

Pursuing this problem further in a heuristic manner we shall note that if higher order elements (e.g., quadratic elements) are used the solution still remains exact or nearly exact at the end nodes of an element but may depart from exactness at the interior nodes, as shown in Fig. 14.4(a). The stresses, or gradients, in this case will be optimal at points which correspond to the two Gauss quadrature points for each element as indicated in Fig. 14.4(b). This fact was observed experimentally by Barlow[1], and such points are frequently referred to as *Barlow points*.

We shall now state in an axiomatic manner that:

(*a*) the displacements are best sampled at the nodes of the element, whatever the order of the element is, and
(*b*) the best accuracy is obtainable for gradients or stresses at the Gauss points corresponding, in order, to the polynomial used in the solution.

At such points the order of the convergence of the function or its gradients is one order higher than that which would be anticipated from the appropriate polynomial and thus such points are known as *superconvergent*. The reason for such superconvergence will be shown in the next section where we introduce the reader to a theorem developed by Herrmann.[2]

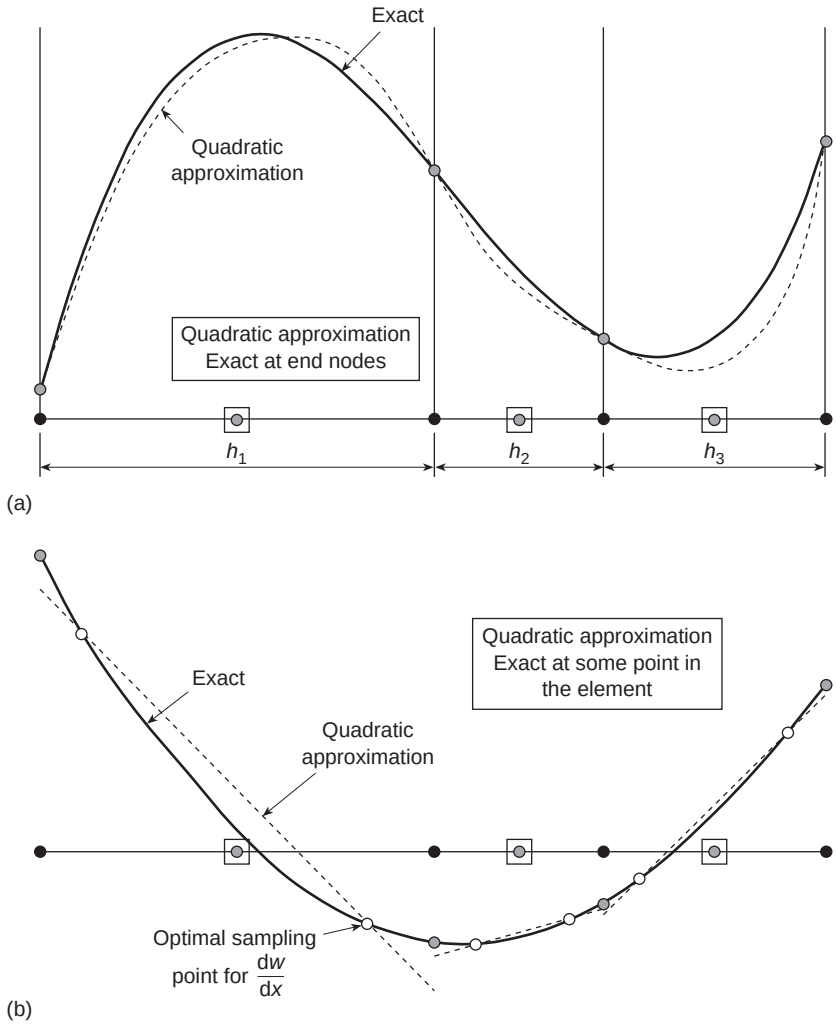


Fig. 14.4 Optimal sampling points for the function (a) and its gradient (b) in one dimension (quadratic

14.2.2 The Herrmann theorem and optimal sampling points

The concept of least square fitting has additional justification in self-adjoint problems in which an energy functional is minimized. In such cases, typical of a displacement formulation of elasticity, it can be readily shown that the minimization is equivalent to a least square fit of approximation stresses to the exact ones. Thus quite generally we can start from a theory which states that *minimization of an energy functional* Π *defined as*

$$\Pi = \frac{1}{2}\int_{\Omega} (\mathbf{Su})^{\mathrm{T}} \mathbf{ASu}\, d\Omega + \int_{\Omega} \mathbf{u}^{\mathrm{T}}\mathbf{p}\, d\Omega \tag{14.19}$$

which at an absolute minimum gives the exact solution $\hat{\mathbf{u}} = \tilde{\mathbf{u}}$ *this is equivalent to minimization of another functional* Π^* *defined as*

$$\Pi^* = \tfrac{1}{2}\int_\Omega [\mathbf{S}(\mathbf{u} - \bar{\mathbf{u}})]^{\mathrm{T}}\mathbf{A}\mathbf{S}(\mathbf{u} - \bar{\mathbf{u}})\,\mathrm{d}\Omega \tag{14.20}$$

In the above, $\mathbf{S}$ is a self-adjoint operator and $\mathbf{A}$ and $\mathbf{p}$ are prescribed matrices of position. The above quadratic form [Eq. (14.19)] arises in the majority of linear self-adjoint problems.

For elasticity problems this theorem is given by Herrmann[2] and shows that the approximate solution for $\mathbf{Su}$ approaches the exact one $\mathbf{S}\bar{\mathbf{u}}$ as a *weighted least square approximation*.

The proof of the Herrmann theorem is as follows. The variation of Π defined in Eq. (14.19) gives, at $\hat{\mathbf{u}} = \tilde{\mathbf{u}}$ (the exact solution),

$$\delta\Pi = \tfrac{1}{2}\int_\Omega (\mathbf{S}\delta\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{A}\mathbf{S}\bar{\mathbf{u}}\,\mathrm{d}\Omega + \tfrac{1}{2}\int_\Omega (\mathbf{S}\mathbf{u})^{\mathrm{T}}\mathbf{A}\mathbf{S}\delta\hat{\mathbf{u}}\,\mathrm{d}\Omega + \int_\Omega \delta\hat{\mathbf{u}}^{\mathrm{T}}\mathbf{p}\,\mathrm{d}\Omega = 0 \tag{14.21}$$

or as $\mathbf{A}$ is symmetric

$$\delta\Pi = \int_\Omega (\mathbf{S}\delta\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{A}\mathbf{S}\bar{\mathbf{u}}\,\mathrm{d}\Omega + \int_\Omega \delta\hat{\mathbf{u}}^{\mathrm{T}}\mathbf{p}\,\mathrm{d}\Omega = 0 \tag{14.22}$$

in which $\delta\mathbf{u}$ is any arbitrary variation. Thus we can write

$$\delta\mathbf{u} = \mathbf{u} \tag{14.23}$$

and

$$\int_\Omega (\mathbf{S}\hat{\mathbf{u}})^{\mathrm{T}}\mathbf{A}\mathbf{S}\bar{\mathbf{u}}\,\mathrm{d}\Omega + \int_\Omega \hat{\mathbf{u}}^{\mathrm{T}}\mathbf{p}\,\mathrm{d}\Omega = 0 \tag{14.24}$$

Subtracting the above from Eq. (14.19) and noting the symmetry of the $\mathbf{A}$ matrix, we can write

$$\Pi = \tfrac{1}{2}\int_\Omega [\mathbf{S}(\hat{\mathbf{u}} - \mathbf{u})]^{\mathrm{T}}\mathbf{A}\mathbf{S}(\hat{\mathbf{u}} - \mathbf{u})\,\mathrm{d}\Omega - \tfrac{1}{2}\int_\Omega [\mathbf{S}(\mathbf{u})]^{\mathrm{T}}\mathbf{A}\mathbf{S}\mathbf{u}\,\mathrm{d}\Omega \tag{14.25}$$

where the last term is not subject to variation. Thus

$$\Pi^* = \Pi + \text{constant} \tag{14.26}$$

and its stationarity is equivalent to the stationarity of Π.

It follows directly from the Herrmann theorem that, for one dimension and by a well-known property of the Gauss–Legendre quadrature points, if the approximate gradients are defined by a polynomial of degree $p - 1$, where p is the degree of the polynomial used for the unknown function u, then stresses taken at these quadrature points must be superconvergent. The single point at the centre of an element integrates precisely all linear functions passing through that point and, hence, if the stresses are exact to the linear form they will be exact at that point of integration. For any higher order polynomial of order p, the Gauss–Legendre points numbering p will provide points of superconvergent sampling. We see this from Fig. 14.5 directly. Here we indicate one, two, and three point Gauss–Legendre quadrature showing why exact results are recovered there for gradients and stresses.

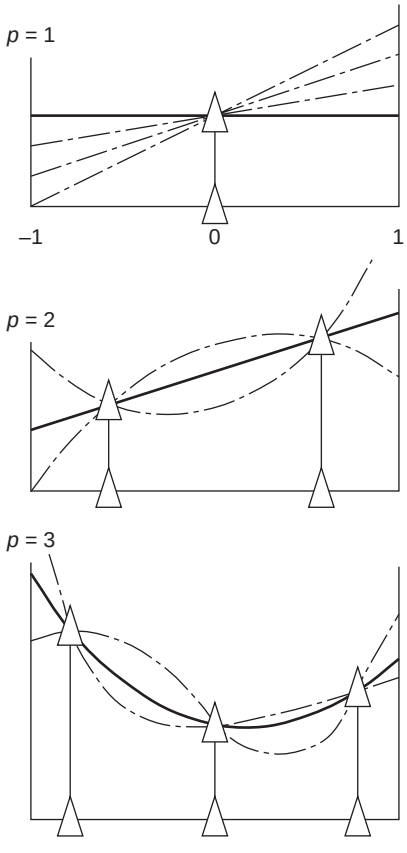


Fig. 14.5 The integration property of Gauss points: $p = 1$, $p = 2$, and $p = 3$ which guarantees superconvergence.

For points based on rectangles and products of polynomial functions it is clear that the exact integration points will exist at the product points as shown in Fig. 14.6 for various rectangular elements assuming that the weighting matrix $\mathbf{A}$ is diagonal. In the same figure we show, however, some triangles and what appear to be 'good' but not necessarily superconvergent sampling points. These are suggested by Moan.[3] Though we find that superconvergent points do not exist in triangles, the points shown in Fig. 14.6 are optimal. In Fig. 14.6 we contrast these points with the minimum number of quadrature points necessary for obtaining an accurate (though not always stable) stiffness representation and find these to be almost coincident at all times.

In Fig. 14.7 representing an analysis of a cantilever by four rectangular quadratic serendipity elements we see how well the stresses sampled at superconvergent points behave compared to the overall stress pattern computed in each element. It is from results like this that many suggestions have been made to obtain improved nodal values and one method proposed by Hinton and Campbell has proved to be quite widely used.[4] However, we shall discuss better recovery procedures later.

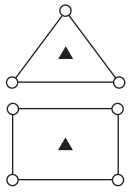
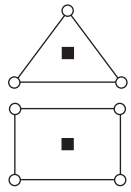
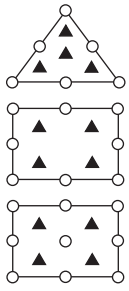
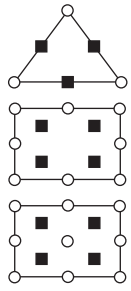
p	Optimal error $O(h^{2(p-m)+2})$	Minimal quadrature $O(h^{2(p-m)+1})$
1	$O(h^2)$	$\geq O(h^2)$
	 $O(h^2)$ $O(h^2)$	 $O(h^2)$ $O(h^2)$
2	$O(h^4)$	$\geq O(h^3)$
	 $O(h^4)$ $O(h^4)$ $O(h^4)$	 $O(h^3)$ $O(h^4)$ $O(h^4)$

Fig. 14.6 Optimal superconvergent sampling and minimum integration points for some C_0 elements.

The extension of the idea of superconvergent points from one-dimensional elements to two-dimensional rectangles was fairly obvious. However, the full superconvergence is lost when isoparametric distortion occurs. We have shown, however, that results at the pth-order Gauss–Legendre points still remain excellent and we suggest that superconvergent properties of the integration points continue to be used for sampling.

In all of the above discussion we have assumed that the weighting matrix $\mathbf{A}$ is diagonal. But if such diagonality does not exist then the existence of superconvergent points is questionable. However excellent results are still available through the sampling points defined as above.

Finally, we refer readers to references 5–9 for surveys on the superconvergence phenomenon and its detailed analyses.

14.3 Recovery of gradients and stresses

In the previous section we have shown that sampling of the gradients and stresses at some particular points is generally optimal and possesses a higher order accuracy when such points are superconvergent. However, we would also like to have similarly accurate quantities elsewhere within each element for general analysis purposes, and in particular we need such highly accurate gradients and stresses when the energy norm or other similar norms have to be evaluated in error estimates. We have already

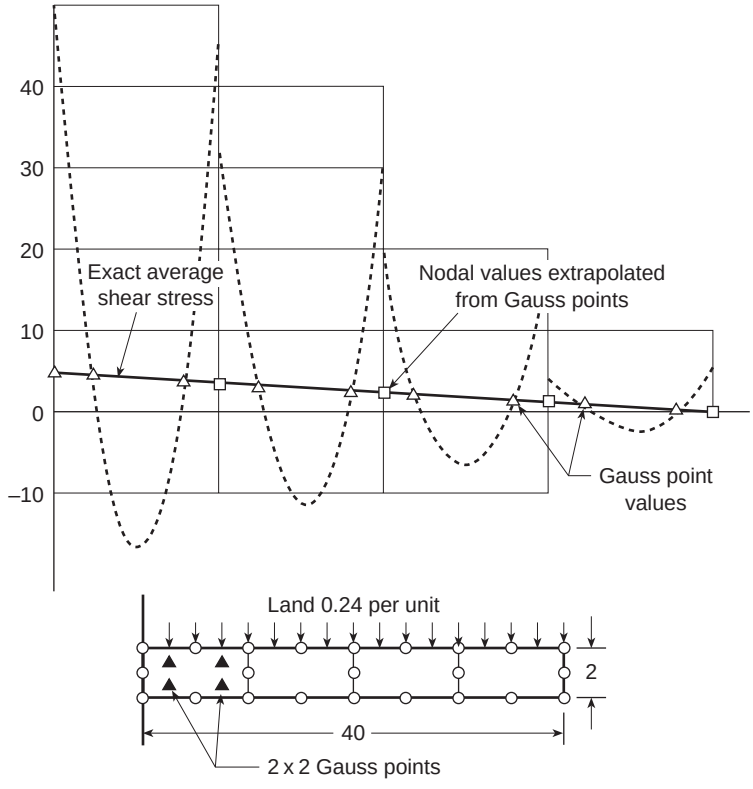


Fig. 14.7 Cantilever beam with four quadratic (Q8) elements. Stress sampling at cubic order (2×2) Gauss points with extrapolation to nodes.

shown how with some elements very large errors exist beyond the superconvergent point and attempts have been made from the earliest days to obtain a complete picture of stresses which is more accurate overall. Here attempts are generally made to recover the nodal values of stresses and gradients from those sampled internally and then to assume that throughout the element the recovered stresses $\boldsymbol{\sigma}^*$ are obtained by interpolation in the same manner as the displacements

$$\boldsymbol{\sigma}^* = \mathbf{N}_u \tilde{\boldsymbol{\sigma}}^* \tag{14.27}$$

We have already suggested a process used almost from the beginning of finite element calculations for triangular elements, where elements are sampled at the centroid (assuming linear shape functions have been used) and then the stresses are averaged at nodes. We have referred to such recovery in Chapter 4. However this is not the best for triangles and for higher order elements such averaging is inadequate. Here other procedures were necessary, for instance Hinton and Campbell[4] suggested a procedure in which stresses at all nodes were calculated by extrapolating the Gauss point values. A further improvement of a similar kind was suggested by Brauchli and Oden[10] who used the stresses in the manner given by Eq. (14.27) and assumed that these stresses should represent in a least square sense the actual finite element stresses, therefore an L_2

projection. Though this has a similarity with the ideas contained in the Herrmann theorem it reverses the order of least square application and has not proved to be always stable and accurate, especially for even order elements. We have already described this procedure in the chapter on mixed elements (see Sec. 11.6) and noted that to obtain results it is necessary to invert a 'mass' type matrix. This can only be achieved without high cost if the mass matrix is diagonal. However, in the following presentation we will show that highly improved results can be obtained by direct polynomial 'smoothing' of the superconvergent values. Here the first method of importance is called *superconvergent patch recovery*.[11–13]

14.4 Superconvergent patch recovery – SPR

14.4.1 Recovery for gradients and stresses

We have already noted that the stresses sampled at certain points in an element possess the superconvergent property (i.e., converge at the same rate as displacement) and have errors of order $O(h^{p+1})$. A fairly obvious procedure for utilizing such sampled values seems to the authors to be that of involving a smoothing of such values by a polynomial of order p within a *patch of elements* for which the number of sampling points can be taken as greater than the number of parameters in the polynomial. In Fig. 14.8 we show several such patches each assembled around a central corner node. The first four represent rectangular elements where the superconvergent points are well defined. The last two give patches of triangles where the best sampling points are used which are not superconvergent.

If we accept the superconvergence of $\hat{\boldsymbol{\sigma}}$ at certain points s in each element then it is a simple matter (which also turns out computationally much less expensive than the L_2 projection) *to compute* $\boldsymbol{\sigma}^*$ *which is superconvergent at all points within the element*. The procedure is illustrated for two dimensions in Fig. 14.8, where we shall consider interior patches (assembling all elements at interior nodes) as shown.

At the superconvergent point the values of $\hat{\boldsymbol{\sigma}}$ are accurate to order $p+1$ (not p as is true elsewhere). However, we can easily obtain an approximation given by a polynomial of degree p, with identical order to these occurring in the shape function for displacement, which has superconvergent accuracy everywhere if this polynomial is made to fit the superconvergent points in a least square manner.

Thus we proceed for each component $\hat{\sigma}_i$ of $\hat{\boldsymbol{\sigma}}$ as follows: Writing the recovered solution as

$$\begin{aligned} \sigma_i^* &= \mathbf{p}\mathbf{a} = [\,1, \quad x, \quad y, \quad \cdots, \quad y^p\,]\mathbf{a} \\ \mathbf{a} &= [\,a_1, \quad a_2, \quad \cdots, \quad a_m\,]^{\mathrm{T}} \end{aligned} \tag{14.28}$$

we minimize, for an element patch with total n sampling points,

$$\begin{aligned} \Pi &= \sum_{k=1}^{n} [\hat{\sigma}_i(x_k, y_k) - \mathbf{p}_k\mathbf{a}]^2 \\ \mathbf{p}_k &= \mathbf{p}(x_k, y_k) \end{aligned} \tag{14.29}$$

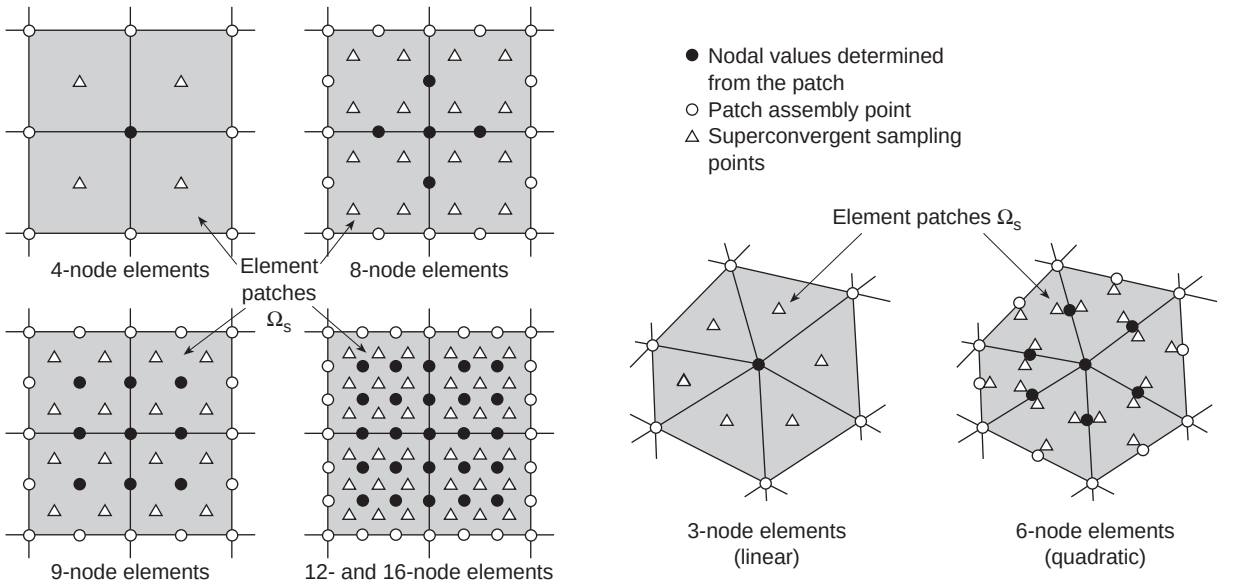


Fig. 14.8 Interior superconvergent patches for quadrilateral elements (linear, quadratic, and cubic) and triangles (linear and quadratic).

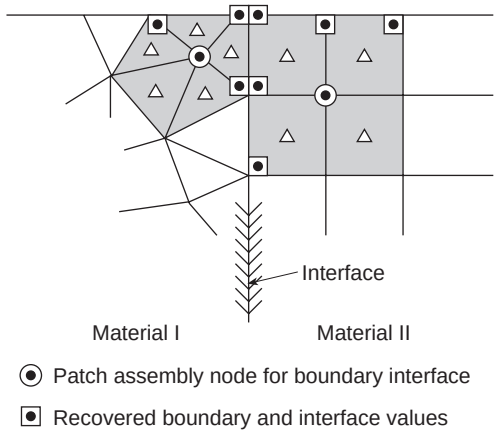


Fig. 14.9 Recovery of boundary or interface gradients.

[(x_k, y_k) corresponding to coordinates of superconvergent points] obtaining immediately the coefficient $\mathbf{a}$ as

$$\mathbf{a} = \mathbf{A}^{-1}\mathbf{b} \tag{14.30}$$

where

$$\mathbf{A} = \sum_{k=1}^{n} \mathbf{p}_k^{\mathrm{T}}\mathbf{p}_k \qquad \text{and} \qquad \mathbf{b} = \sum_{k=1}^{n} \mathbf{p}_k^{\mathrm{T}}\hat{\sigma}_i(x_k, y_k) \tag{14.31}$$

The availability of $\boldsymbol{\sigma}^*$ allows the superconvergent values of $\tilde{\boldsymbol{\sigma}}^*$ to be determined at all nodes. As some nodes belong to more than one patch, average values of $\tilde{\boldsymbol{\sigma}}^*$ are best obtained. The superconvergence of $\boldsymbol{\sigma}^*$ throughout each element is achieved with Eq. (14.27).

It should be noted that on external boundaries and indeed on interfaces where stresses are discontinuous the nodal values should be calculated from interior patches in the manner shown in Fig. 14.9.

In Fig. 14.10 we show in a one-dimensional example how the superconvergent patch recovery reproduces *exactly* the stress (gradient) solutions of order $p+1$ for linear or quadratic elements. Following the arguments of Chapter 10 on the patch test it is evident that superconvergent recovery is now achieved at all points. Indeed, the same figure shows why averaging (or L_2 projection) is inferior (particularly on boundaries).

Figure 14.11 shows experimentally determined convergence rates for a one-dimensional problem (stress distribution in a bar of length $L = 1$; $0 \leqslant x \leqslant 1$ and prescribed body forces). A uniform subdivision is used here to form the elements, and the convergence rates for the stress error at $x = 0.5$ are shown using the direct stress approximation $\hat{\sigma}$, the L_2 recovery σ_L and σ^* obtained by the SPR procedure using linear, quadratic and cubic elements. It is immediately evident that σ^* is superconvergent with a rate of convergence being at least one order higher than that of $\hat{\sigma}$. However, as anticipated, the L_2 recovery gives much inferior answers, showing superconvergence only for odd values of p and almost no improvement for even

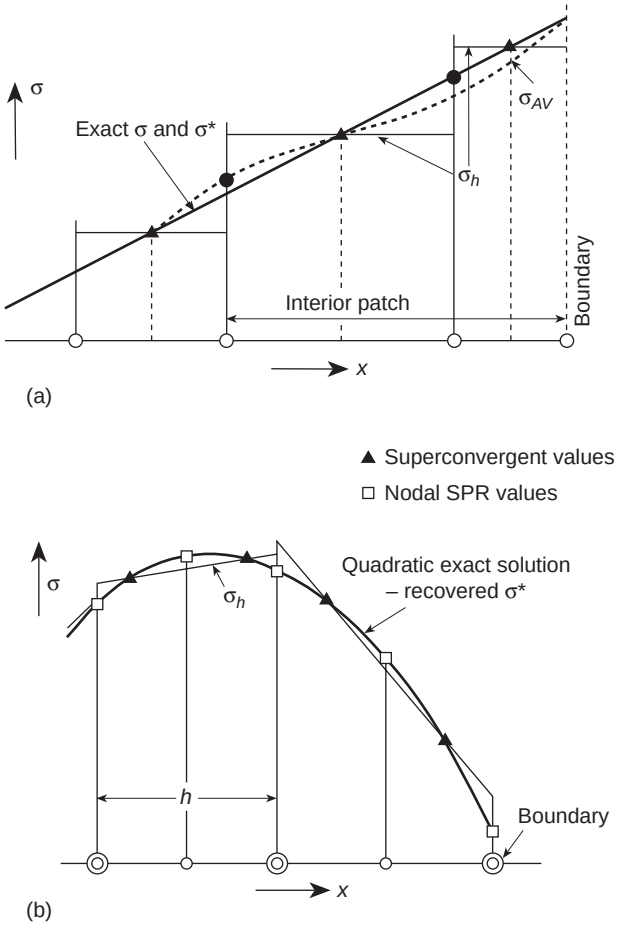


Fig. 14.10 Recovery of exact σ of degree p by linear elements ($p = 1$) and quadratic elements ($p = 2$).

values of p, while σ^* shows a two-order increase of convergence rate for even order elements (tests on higher order polynomials are reported in reference 14). This *ultra convergence* has been verified mathematically.[15] Although it is not observed when elements of varying size are used, the important tests shown in Figs 14.12 and 14.13 indicate how well the recovery process works.

In the first of these, Fig. 14.12, a field problem is solved in two dimensions using a very irregular mesh for which the existence of superconvergent points is only inferred heuristically. The very small error in σ_x^* is compared with the error of $\hat{\sigma}_x$ and the improvement is obvious. Here $\sigma_x = \partial u/\partial x$ where u is the fluid variable.

In the second, i.e., Fig. 14.13, a problem of stress analysis, for which an exact solution is known, is solved using three different recovery methods. Once again the recovered solution σ^* (SPR) shows the much improved values compared with σ_L and it is clear that the SPR process *should be included in all codes if simply to present improved stress values.*

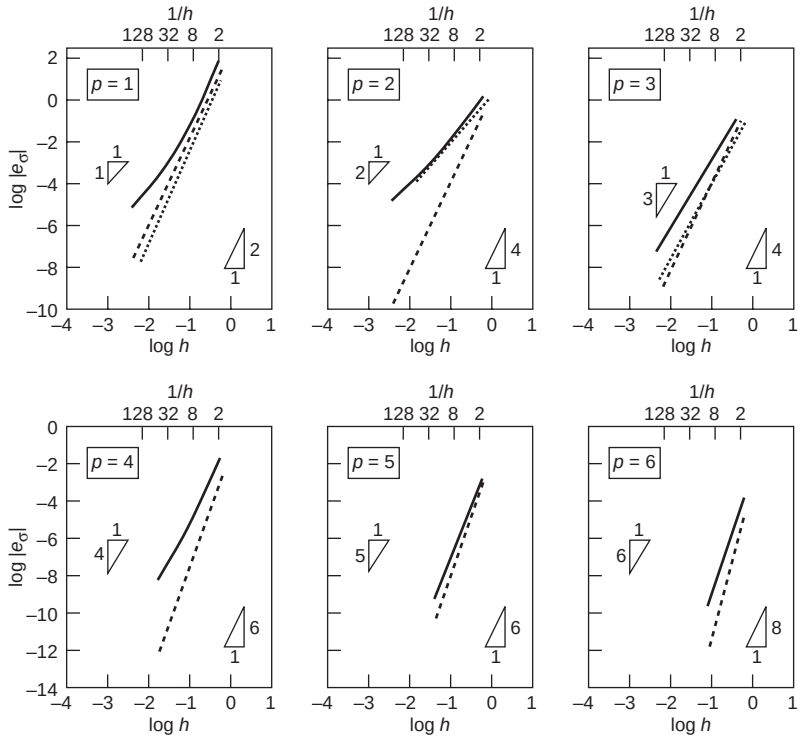


Fig. 14.11 Problem of a stressed bar. Rates of convergence (error) of stress, where $x = 0.5$ ($0 \leqslant x \leqslant 1$). ($\hat{\sigma}$ ——; σ_L · · · ·; σ^* - - - -)

The SPR procedure which we have just outlined has proved to be a very powerful tool leading to superconvergent results on regular meshes and much improved results (nearly superconvergent) on irregular meshes. It has been shown numerically that it produces superconvergent recovery even for triangular elements which do not have superconvergent points within the element. A recent mathematical proof confirms this capability of SPR.[6] The procedure was introduced by Zienkiewicz and Zhu in 1992[11–13] and we still recommend it as the best procedure which is simple to use. However, many investigators have modified the procedure by increasing the functional where

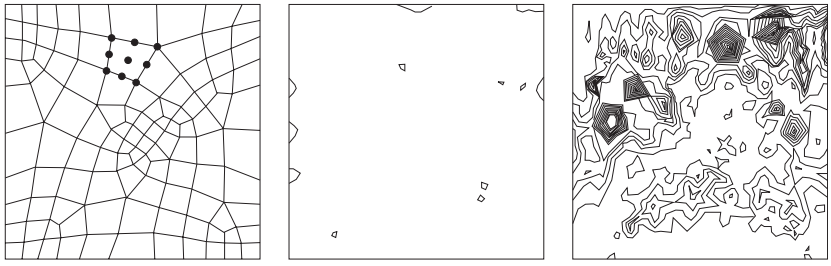


Fig. 14.12 Poisson equation in two dimensions solved using arbitrary shaped quadratic quadrilaterals.

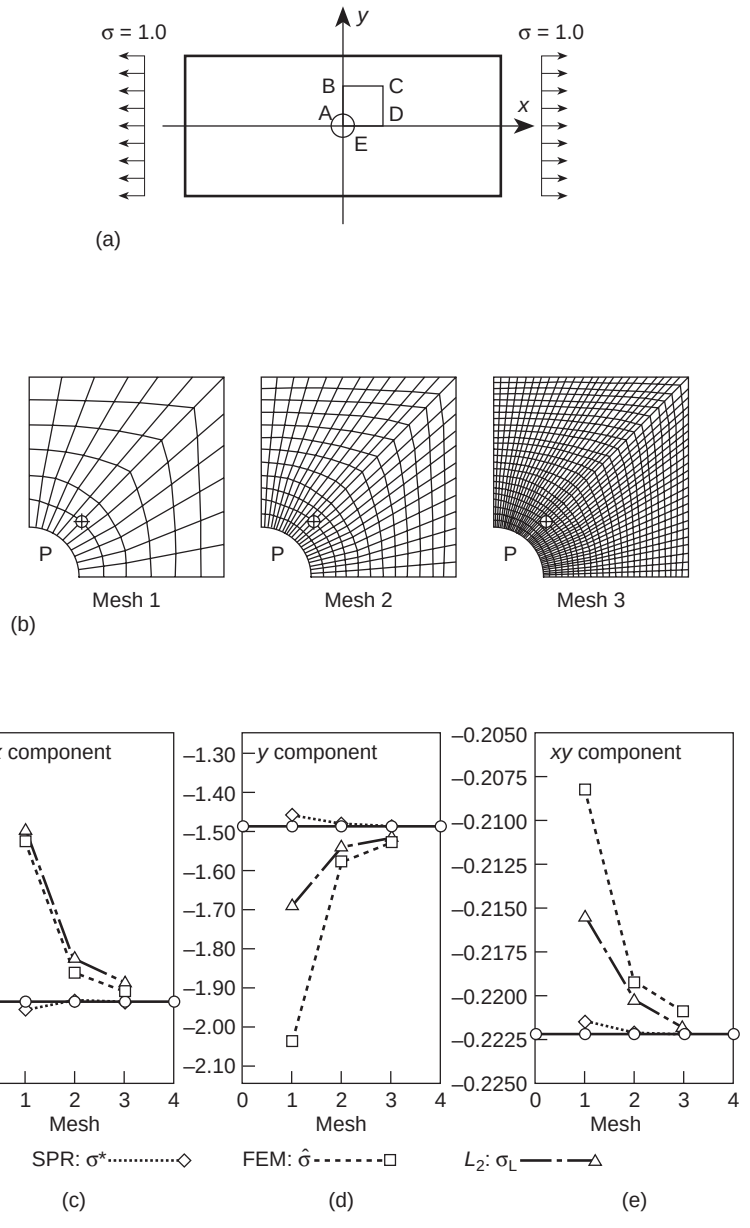


Fig. 14.13 Plane stress analysis of stresses around a circular hole in a uniaxial field.

the least square fit is performed to include satisfaction of discrete equilibrium equations or boundary conditions, etc. While the satisfaction of known boundary tractions can on occasion be useful most of these additional constraints introduced have affected the superconvergent properties adversely and in general the modified versions of SPR by Wiberg *et al.*[17] and by Blacker and Belytschko[18] have not proved to be fully effective.

14.4.2 SPR for displacements

The superconvergent patch recovery can be extended to produce superconvergent displacements. The procedure for the displacements is quite simple if we assume the superconvergent points to be at nodes of the patch. However, as we have already observed it is always necessary to have more data than the number of coefficients in the particular polynomial to be able to execute a least square minimization. Here of course we occasionally need a patch which extends further than before, particularly since the displacements will be given by a polynomial one order higher than that used for the shape functions. In Fig. 14.8 however we show for most assemblies that a similar patch as given before can be again applied producing a good approximation for $\mathbf{u}$ within its interior. Larger element patches have also been suggested in reference 19.

The recovered solution $\mathbf{u}^*$ has on occasion been used in dynamic problems (e.g., Wiberg[19,20]), because in dynamic problems the displacements themselves are often important. We shall find such recovery useful in some problems of fluid dynamics in Volume 3.

The SPR recovery technique described in this section takes advantage of the superconvergence property of the finite element solutions and the availability of the optimal sampling points. Very recently a new method of recovery which does not need such information has been devised and will be discussed in the next section.

14.5 Recovery by equilibration of patches – REP

Although SPR has proved to work well generally, the reason behind its capability of producing an accurate recovered solution even when superconvergent points do not in fact exist remains an open question. We have therefore sought to determine viable recovery alternatives. One of these, known by the acronym REP (recovery by equilibrium of patches), will be described next. This procedure was first presented in reference 21 and later improved in reference 22.

To some extent the motivation is similar to that of Ladevèze *et al.*[23,24] who sought to establish (for somewhat different reasons) a fully equilibrating stress field which can replace that of the finite element approximation. However we believe that the process derived in reference 21 is simpler though equilibration is only approximate.

The starting point is the governing equilibrium equation

$$\mathbf{S}^{\mathrm{T}}\boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \tag{14.32}$$

In the finite element approximation this becomes

$$\int_{\Omega_p} \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}\,\mathrm{d}\Omega - \int_{\Omega_p} \mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_p} \mathbf{N}^{\mathrm{T}}\mathbf{t}\,\mathrm{d}\Gamma = \mathbf{0} \tag{14.33}$$

where $\hat{\boldsymbol{\sigma}}$ are the stresses from the finite element solution. In the above Ω_p is the domain of the patch and the last term comes from the tractions on the boundary of the patch domain Γ_p. These can, of course, represent the whole of the problem, an element patch or only a single element.

As is well known the stresses $\hat{\boldsymbol{\sigma}}$ which result from the finite element analysis will in general be discontinuous and we shall seek to replace them in *every element patch* by a recovered system which is smooth and continuous.

To achieve the recovery we proceed in an exactly analogous way to that used in the SPR procedure, *first* approximating the stress in each patch by a polynomial of appropriate order $\boldsymbol{\sigma}^*$, *second* using this approximation to obtain nodal values of $\tilde{\boldsymbol{\sigma}}^*$ and *finally* interpolating these values by standard shape functions.

The stress $\boldsymbol{\sigma}$ is taken as a vector of appropriate components, which for convenience we write as:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{Bmatrix} \tag{14.34}$$

The above notation is general with, for instance, $\sigma_1 = \sigma_x$, $\sigma_2 = \sigma_y$ and $\sigma_3 = \tau_{xy}$ in two-dimensional plane elastic analysis.

We shall write each component of the above as a polynomial expansion of the form:

$$\sigma_i^* = [\,1, \quad x, \quad y, \quad \cdots]\mathbf{a}_i = \mathbf{p}(x,y)\mathbf{a}_i \tag{14.35}$$

where $\mathbf{p}$ is a vector of polynomials and $\mathbf{a}_i$ is a set of unknown coefficients for the ith component of stress.

For equilibrium we shall always attempt to ensure that the total smoothed stress $\boldsymbol{\sigma}^*$ satisfies in the least square sense the same patch equilibrium conditions as the finite element solution. Accordingly,

$$\int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \hat{\boldsymbol{\sigma}}\, \mathrm{d}\Omega \approx \int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma}^*\, \mathrm{d}\Omega \tag{14.36}$$

where

$$\boldsymbol{\sigma}^* = \mathbf{P}\mathbf{a} = \begin{bmatrix} \mathbf{p} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{p} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{Bmatrix} \tag{14.37}$$

written here again for the case of three stress components. Obvious modifications are made for more or less components.

It has been found in practice that the constraints provided by Eq. (14.36) are not sufficient to always produce non-singular least square minimization. Accordingly, the equilibrium constraints are split into an alternative form in which each component of stress is subjected to equilibrium requirements. This may be achieved by expressing the stress as

$$\boldsymbol{\sigma}^* = \sum_i \mathbf{1}_i \sigma_i^* = \sum_i \boldsymbol{\sigma}_i^* \tag{14.38}$$

$$\hat{\boldsymbol{\sigma}} = \sum_i \mathbf{1}_i \hat{\sigma}_i = \sum_i \hat{\boldsymbol{\sigma}}_i{}^* \tag{14.39}$$

$$\mathbf{1}_1 = [\,1, \quad 0, \quad 0\,]^{\mathrm{T}} \tag{14.40}$$

where

$$\mathbf{1}_2 = [\,0, \quad 1, \quad 0\,]^{\mathrm{T}} \quad \text{etc.} \tag{14.41}$$

and imposing the set of constraints

$$\int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \hat{\boldsymbol{\sigma}}_i \, \mathrm{d}\Omega \approx \int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma}_i^* \, \mathrm{d}\Omega = \int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \mathbf{1}_i \mathbf{p} \, \mathrm{d}\Omega \mathbf{a}_i \tag{14.42}$$

The imposition of the approximate equation (14.42) allows each set of coefficients $\mathbf{a}_i$ to be solved independently reducing considerably the solution cost and here repeating a procedure used with success in SPR.

A least square minimization of Eq. (14.42) is expressed as

$$\Pi = (\mathbf{H}_i \mathbf{a}_i - \mathbf{f}_i^p)^{\mathrm{T}} (\mathbf{H}_i \mathbf{a}_i - \mathbf{f}_i^p) \tag{14.43}$$

where

$$\mathbf{H}_i = \int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \mathbf{1}_i \mathbf{p} \, \mathrm{d}\Omega \tag{14.44}$$

and

$$\mathbf{f}_i^p = \int_{\Omega_p} \mathbf{B}^{\mathrm{T}} \hat{\boldsymbol{\sigma}}_i \, \mathrm{d}\Omega \tag{14.45}$$

The minimization condition results in

$$\mathbf{a}_i = \left[\mathbf{H}_i^{\mathrm{T}} \mathbf{H}_i\right]^{-1} \mathbf{H}_i^{\mathrm{T}} \mathbf{f}_i^p \tag{14.46}$$

For patches in some problems Eq. (14.43) may be unstable. Generally, this may be eliminated by modifying the patch requirement to the minimization of

$$\Pi^* = (\mathbf{H}_i \mathbf{a}_i - \mathbf{f}_i^p)^{\mathrm{T}} (\mathbf{H}_i \mathbf{a}_i - \mathbf{f}_i^p) + \sum_e \alpha (\mathbf{H}_i^e \mathbf{a}_i - \mathbf{f}_i^e)^{\mathrm{T}} (\mathbf{H}_i^e \mathbf{a}_i - \mathbf{f}_i^e) \tag{14.47}$$

where the added terms represent modification on individual elements and α is a parameter. Minimization now gives

$$\mathbf{a}_i = \left[\mathbf{H}_i^{\mathrm{T}} \mathbf{H}_i + \alpha \sum_e \mathbf{H}_i^{e,\mathrm{T}} \mathbf{H}_i^e\right]^{-1} \left[\mathbf{H}_i^{\mathrm{T}} \mathbf{f}_i^p + \alpha \sum_e \mathbf{H}_i^{e,\mathrm{T}} \mathbf{f}_i^e\right] \tag{14.48}$$

The REP procedure follows precisely the details of SPR near boundaries and gives overall an approximation which does not require knowledge of any superconvergent points. The accuracy of both processes is comparable and we are of the opinion that many other alternative recovery procedures are still possible.

14.6 Error estimates by recovery

One of the most important applications of the recovery methods is its use in the computation of the *a posteriori* error estimators. With the recovered solutions available, we can now evaluate errors simply by replacing the exact values of quantities such as $\mathbf{u}$, $\boldsymbol{\sigma}$, etc., which are in general unknown, in Eqs (14.1)–(14.3), by the recovered values which are much more accurate than the direct finite element solution. We write the error estimators in various norms such as

$$||\mathbf{e}|| \approx ||\bar{\mathbf{e}}|| = ||\mathbf{u}^* - \hat{\mathbf{u}}|| \tag{14.49}$$

$$||\mathbf{e}||_{L_2} \approx ||\bar{\mathbf{e}}||_{L_2} = ||\mathbf{u}^* - \hat{\mathbf{u}}||_{L_2} \tag{14.50}$$

$$||\mathbf{e}_\sigma||_{L_2} \approx ||\bar{\mathbf{e}}_\sigma||_{L_2} = ||\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}||_{L_2} \tag{14.51}$$

For example, the energy norm error estimator for elasticity problems has the form of

$$||\bar{\mathbf{e}}|| = \left[\int_\Omega (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}})^{\mathrm{T}} \mathbf{D}^{-1} (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}})\, \mathrm{d}\Omega\right]^{\frac{1}{2}} \tag{14.52}$$

Similarly, estimates of the RMS error in displacement and stress can be obtained through Eqs (14.12) and (14.13). Error estimators formulated by replacing the exact solution with the recovered solution are sometimes called *recovery based error estimators*. This type of error estimator was first introduced by Zienkiewicz and Zhu.[25]

The accuracy or the quality of the error estimator is measured by the effectivity index θ, which is defined as

$$\theta = \frac{||\bar{\mathbf{e}}||}{||\mathbf{e}||} \tag{14.53}$$

A theorem proposed by Zienkiewicz and Zhu[12] shows that for all estimators based on recovery we can establish the following bounds for the effectivity index:

$$1 - \frac{||\mathbf{e}^*||}{||\mathbf{e}||} \leqslant \theta \leqslant 1 + \frac{||\mathbf{e}^*||}{||\mathbf{e}||} \tag{14.54}$$

where $\mathbf{e}$ is the actual error and $\mathbf{e}^*$ is the error of the recovered solution, e.g.

$$||\mathbf{e}^*|| = ||\mathbf{u} - \mathbf{u}^*||$$

The proof of the above theorem is straightforward if we write Eq. (14.52) as

$$||\bar{\mathbf{e}}|| = ||\mathbf{u}^* - \hat{\mathbf{u}}|| = ||(\mathbf{u} - \hat{\mathbf{u}}) - (\mathbf{u} - \mathbf{u}^*)|| = ||\mathbf{e} - \mathbf{e}^*|| \tag{14.55}$$

Using now the triangle inequality we have

$$||\mathbf{e}|| - ||\mathbf{e}^*|| \leqslant ||\bar{\mathbf{e}}|| \leqslant ||\mathbf{e}|| + ||\mathbf{e}^*|| \tag{14.56}$$

from which the inequality (14.54) follows after division by $||\mathbf{e}||$. Obviously, the theorem is also true for error estimators of other norms. Two important conclusions follow:

1. *any recovery process* which results in reduced error will give a reasonable error estimator and, more importantly,
2. if the recovered solution converges at a higher rate than the finite element solution we shall always have asymptotically exact estimation.

To prove the second point we consider a typical finite element solution with shape functions of order p where we know that the error (in the energy norm) is:

$$||\mathbf{e}|| = O(h^p) \tag{14.57}$$

If the recovered solution gives an error of a higher order, e.g.,

$$||\mathbf{e}^*|| = O(h^{p+\alpha}) \qquad \alpha > 0 \tag{14.58}$$

then the bounds of the effectivity index are:

$$1 - O(h^{\alpha}) \leqslant \theta \leqslant 1 + O(h^{\alpha}) \tag{14.59}$$

and the error estimator is asymptotically exact, that is

$$\theta \to 1 \qquad \text{as} \qquad h \to 0 \tag{14.60}$$

This means that the error estimator converges to the true error. This is a very important property of error estimators based on recovery not generally shared by residual based estimators which we shall discuss in the next section.

14.7 Other error estimators – residual based methods

Other methods to obtain error estimators have been proposed by many investigators working in the field.[26–34] Most of these make use of the residuals of the finite element approximation, either explicitly or implicitly. Error estimators based on these methods are often called *residual error estimators*. Those using residuals explicitly are termed explicit residual error estimators; the others are called implicit residual error estimators.

In this section we are mainly concerned with implicit residual error estimators, in particular, the *equilibrated element residual estimator* which has been shown to be the most robust among all the residual error estimators.[35–37]

Here we consider the heat conduction problem in a two-dimensional domain as an example. The differential equation is given by

$$-\boldsymbol{\nabla}^{\mathrm{T}}(k\,\boldsymbol{\nabla}\phi) = Q \quad \text{in } \Omega \tag{14.61}$$

with boundary conditions

$$\phi = \bar{\phi} \quad \text{on } \Gamma_{\phi}$$

$$\mathbf{q}^{\mathrm{T}}\mathbf{n} = q_n = \bar{q} \quad \text{on } \Gamma_{q}$$

In the above

$$\mathbf{q} = -k\,\boldsymbol{\nabla}\phi$$

is the heat flux, $\mathbf{n}$ is the outward normal to the boundary Γ and q_n is the flux normal to the boundary (see Chapters 3 and 7).

The error of the finite element solution is

$$e = \phi - \hat{\phi}$$

and for element i the energy norm error is written as

$$||e|| = \left[\int_{\Omega_i} (\boldsymbol{\nabla}\phi)^{\mathrm{T}} k\,\boldsymbol{\nabla}\phi\,\mathrm{d}\Omega\right]^{\frac{1}{2}} \tag{14.62}$$

In what follows we shall construct the equilibrated residual error estimator for this problem. The procedure of constructing an estimator for other problems, such as elasticity problems, is analogous.

We start by considering an interior element i. Substitute the finite element solution $\hat{\phi}$ into Eq. (14.61). Subtracting the resulting equation from Eq. (14.61) gives an

element boundary value problem for error e given by

$$-\nabla^{\mathrm{T}}(k\nabla e) = r_i \quad \text{in } \Omega_i \tag{14.63}$$

with boundary condition

$$-(k\nabla e)^{\mathrm{T}}\mathbf{n} = q_n - \hat{q}_n \quad \text{on } \Gamma_i$$

Here

$$r_i = \nabla^{\mathrm{T}}(k\nabla\phi) + Q$$

is the residual in the finite element and

$$\hat{q}_n = \hat{\mathbf{q}}^{\mathrm{T}}\mathbf{n}$$

is the finite element normal flux.

We notice immediately that Eq. (14.63) is not solvable because the exact normal flux on the element boundary is in general unknown. A natural strategy to overcome this difficulty is to replace the exact normal flux by a recovered solution q_n^* which can be computed from the finite element flux in element i and its surrounding elements.

We can now write the boundary value problem of the element error as

$$-\nabla^{\mathrm{T}}(k\nabla e) = r_i \quad \text{in } \Omega_i \tag{14.64}$$

with boundary condition

$$-(k\nabla e)^{\mathrm{T}}\mathbf{n} = q_n^* - \hat{q}_n \quad \text{on } \Gamma_i$$

The approximate solution of the above equations $\bar{e}$ in the energy norm, $||\bar{e}||$, is defined as the *element residual error estimator*.

Various recovery techniques can be used to recover the normal flux q_n^*.[30,31] However, the Neumann problem of Eq. (14.64) will guarantee to have a solution if q_n^* is computed such that the residuals satisfy

$$\int_{\Omega_i} N_j r_i \, \mathrm{d}\Omega + \int_{\Gamma_i} N_j (q_n^* - \hat{q}_n) \, \mathrm{d}\Gamma = 0 \tag{14.65}$$

where N_j is the shape function for node j of element i. Although N_j can be a shape function of any order, a linear shape function seems to be the most practical in the following computation.

The residuals which satisfy Eq. (14.65) are said to be equilibrated, thus the recovered solution q_n^* satisfying Eq. (14.65) is called the equilibrated flux. An error estimator which uses the solution of the element error problem of Eq. (14.64) with the equilibrated flux q_n^* is termed an equilibrated residual error estimator. This type of residual error estimator was first introduced by Bank and Weiser[30] and later pursued by Ainsworth and Oden.[34]

It is apparent that the most important step in the computation of the equilibrated residual error estimator is to achieve the recovered normal flux q_n^* which satisfies Eq. (14.65). Once q_n^* is determined, the error problem Eq. (14.64) can be readily solved, over an element, following the standard finite element procedure. Therefore we shall focus on the recovery process.

The technique of recovering normal flux by equilibrated residuals was first proposed by Ladevéze *et al.*[23] A different version of this technique was later used by Ainsworth and Oden.[34]

Integrating by parts, we can write Eq. (14.65) in a computationally more convenient form:

$$\int_{\Omega_i} N_j Q \, \mathrm{d}\Omega - \int_{\Omega_i} \boldsymbol{\nabla}^{\mathrm{T}} \left(k \, \boldsymbol{\nabla} \hat{\phi} \right) \mathrm{d}\Omega + \int_{\Gamma_i} N_j q_n^* \, \mathrm{d}\Gamma = 0 \tag{14.66}$$

Let the recovered element boundary normal flux, for each edge of the element, have the form

$$q_n^* = \tfrac{1}{2} (\hat{\mathbf{q}}_i + \hat{\mathbf{q}}_k)^{\mathrm{T}} \mathbf{n}_s + Z_s \tag{14.67}$$

where the first term on the right-hand side is the average of the normal flux of the finite element solution from element i and its neighbour element k; $\mathbf{n}_s$ is the outward normal on the edge s of element i; and Z_s is a linear function defined on the edge s, shared by elements i and k, with end nodes l and r and

$$Z_s = L_l a_l^s + L_r a_r^s \tag{14.68}$$

with

$$L_l = \frac{2}{|h_s|} (2N_l^s - N_r^s) \qquad L_r = \frac{2}{|h_s|} (2N_r^s - N_l^s) \tag{14.69}$$

where N_l^s and N_r^s are linear shape functions defined over edge s and h_s is the length of edge s. The unknown parameters a_l^s and a_r^s are to be determined from the residual equilibrium equation (14.66).

It is easy to verify that

$$\int_s N_m^s L_n \, \mathrm{d}\Gamma = \delta_{mn} \tag{14.70}$$

where δ_{mn} is the Kronecker delta, is given by:

$$\delta_{ij} = 1, \quad i = j; \qquad \delta_{ij} = 0, \quad i \neq j \tag{14.71}$$

Let X_n denote a typical interior vertex node. Choose $N_j = N_n$ in Eq. (14.66) and consider the element patch associated with the linear shape function N_n as shown in Fig. 14.14. A local numbering for the elements and edges connected to node X_n in the patch is given. The edge normals shown here are the results of a global edge orientation.

Assume X_n be the end node l of all the edges connected with X_n. For element e_1 in the patch, substituting Eq. (14.67) into Eq. (14.66) for each edge and observing that N_n is non-zero only on s_1 and s_2 and at the directions of the edge normals, we have

$$\begin{aligned} &\int_{\Omega_i} N_n Q \, \mathrm{d}\Omega - \int_{\Omega_i} (\boldsymbol{\nabla} N_n)^{\mathrm{T}} (k \, \boldsymbol{\nabla} \hat{\phi}) \, \mathrm{d}\Omega - \int_{s_1} \tfrac{1}{2} N_n (\hat{\mathbf{q}}_{e_1} + \hat{\mathbf{q}}_{e_5})^{\mathrm{T}} \mathbf{n}_{s_1} \, \mathrm{d}\Gamma \\ &\quad + \int_{s_2} \tfrac{1}{2} N_n (\hat{\mathbf{q}}_{e_1} + \hat{\mathbf{q}}_{e_2})^{\mathrm{T}} \mathbf{n}_{s_2} \, \mathrm{d}\Gamma - \int_{s_1} N_n Z_{s_1} \, \mathrm{d}\Gamma \, \mathrm{d}\Gamma - \int_{s_2} N_n Z_{s_2} \, \mathrm{d}\Gamma = 0 \end{aligned} \tag{14.72}$$

where the boundary integral takes a negative sign if the edge normal shown in Fig. 14.15 is inward for the element.

Let f_{e_1} denote the first four, computable, terms of the above equation and notice that [using Eq. (14.70)]

$$\int_{s_1} N_n Z_{s_1} \, \mathrm{d}\Gamma = \int_{s_1} N_n (L_{x_n} a_{x_n}^{s_1} + L_r a_r^{s_1}) \, \mathrm{d}\Gamma = a_{x_n}^{s_1} \tag{14.73}$$

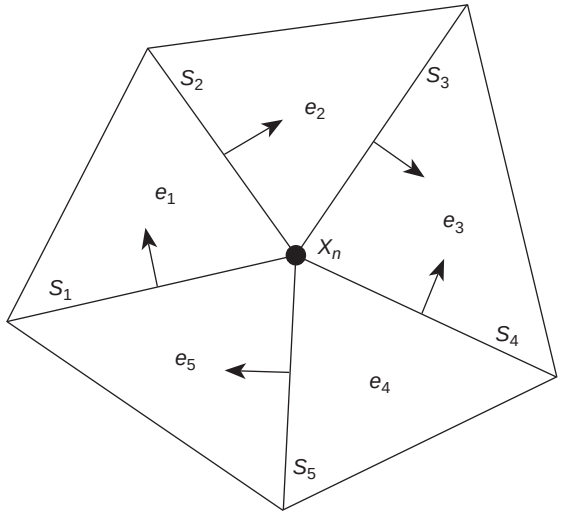


Fig. 14.14 Typical patch with interior vertex node X_n showing a local numbering of elements e_i and edges S_i.

and

$$\int_{s_2} N_n Z_{s_2}\,\mathrm{d}\Gamma = \int_{s_2} N_n\left(L_{x_n} a_{x_n}^{s_2} + L_r a_r^{s_2}\right)\mathrm{d}\Gamma = a_{x_n}^{s_2} \tag{14.74}$$

Equation (14.71) now becomes

$$-a_{x_n}^{s_1} + a_{x_n}^{s_2} = -f_{e_1} \tag{14.75}$$

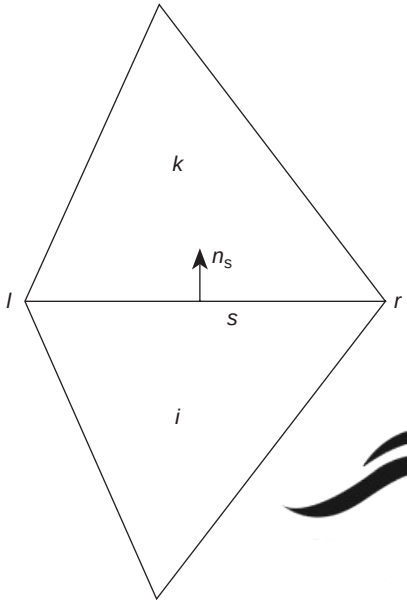


Fig. 14.15 Element interface for equilibrated flux recovery

Similarly for element e_2 to e_5 we have

$$\begin{aligned} -a_{x_n}^{s_2} + a_{x_n}^{s_3} &= -f_{e_2} \\ -a_{x_n}^{s_3} - a_{x_n}^{s_4} &= -f_{e_3} \\ +a_{x_n}^{s_4} + a_{x_n}^{s_5} &= -f_{e_4} \\ -a_{x_n}^{s_5} + a_{x_n}^{s_1} &= -f_{e_5} \end{aligned} \tag{14.76}$$

or in matrix form

$$\mathbf{A}\mathbf{a} = \mathbf{b} \tag{14.77}$$

where

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{14.78}$$

$$\mathbf{a} = \left[a_{x_n}^{s_1}, \quad a_{x_n}^{s_2}, \quad a_{x_n}^{s_3}, \quad a_{x_n}^{s_4}, \quad a_{x_n}^{s_5} \right]^{\mathrm{T}} \tag{14.79}$$

and

$$\mathbf{b} = \left[-f_{e_1}, \quad -f_{e_2}, \quad -f_{e_3}, \quad -f_{e_4}, \quad -f_{e_5} \right]^{\mathrm{T}} \tag{14.80}$$

It is easy to verify that these equations are linearly dependent but have solutions determined up to an arbitrary constant. A procedure to obtain an *optimal* particular solution is described as follows.[23,30,38] First, a particular solution $\mathbf{a}_0$ is found by choosing, for example, $a_{x_n}^{s_5} = 0$. Secondly, the corresponding homogeneous equation

$$\mathbf{A}\mathbf{b} = \mathbf{0} \tag{14.81}$$

with $\mathbf{b} = [b_1, b_2, b_3, b_4, b_5]^{\mathrm{T}}$ is solved for a non-zero particular solution with the choice of, corresponding $a_{x_n}^{s_5}$, $b_5 = 1$. It is easy to verify that b_i is either 1 or -1 due to the structure of $\mathbf{A}$. In the element patch considered here $\mathbf{b} = [1, 1, -1, -1, 1]^{\mathrm{T}}$.

The final particular solution of Eq. (14.77) takes the form

$$\mathbf{a} = \mathbf{a}_0 + \gamma \mathbf{b} \tag{14.82}$$

where the constant γ is determined by the minimization of

$$\Pi = \mathbf{a}^{\mathrm{T}}\mathbf{a} \tag{14.83}$$

The minimization condition gives

$$\gamma = -\frac{\mathbf{b}^{\mathrm{T}}\mathbf{a}_0}{\mathbf{b}^{\mathrm{T}}\mathbf{b}} \tag{14.84}$$

The solution gives the nodal value $a_{x_n}^{s_i}$ for each edge connected to node X_n in the element patch.

Boundary nodes and their related element patches can be considered in the same fashion except that we can take $q_n^* = \bar{q}_n$, the known flux, for the element edge being part of Γ_q. For edges coincident with Γ_ϕ, we let the first term on the right-hand side of Eq. (14.67) be zero. By considering each vertex node of the mesh and its associated element patch, we will be able to determine a_l^s and a_r^s for every edge, thus the recovered normal flux q_n^* on the element boundary is achieved. The procedure described above for recovering the normal flux is a *recovery by element residual*.

We note that the non-uniqueness of the solution of Eq. (14.77) represents the non-uniqueness of the equilibrium status of the element residuals. The choice of the arbitrary constant in solving Eq. (14.77) will certainly affect the accuracy of the recovered solution q_n^*, and therefore the accuracy of the error estimator.

The local error problem Eq. (14.64) is usually solved by a higher order (e.g., $p+1$ or even $p+2$) approximation. The solution of the problem is then employed in the element equilibrated error estimator $||\bar{e}||_i$. The global error estimator $||\bar{e}||$ is obtained through Eq. (14.15). The global error estimator has been shown to be an upper bound of the exact error,[34] although it is not a trivial task to prove its convergence.

We have shown here that the recovery method is the key to the computation of implicit residual error estimators. It can be shown that using a properly designed recovery method some of the explicit residual error estimators or their equivalent can, in fact, be directly derived from recovery based error estimators.[39,40] Numerical performance of residual based error estimators was tested by Babuška *et al.*[35–37] and compared with that of recovery based error estimators.

14.8 Asymptotic behaviour and robustness of error estimators – the Babuška patch test

It is well known that elements in which polynomials of order p are used to represent the unknown $\mathbf{u}$ will reproduce exactly any problem for which the exact solution is also defined by such a polynomial. Indeed the verification of this behaviour is an essential part of the 'patch test' which has to be satisfied by all elements to ensure convergence, as we have discussed in Chapter 10.

Thus if we are attempting to determine the error in a general smooth solution we will find that this error is dominated by terms of order $p+1$. The response of any patch to an exact solution of order $p+1$ will therefore determine the asymptotic behaviour when both the size of the patch and of all the elements tends to zero. If the patch is assumed to be one of a repeatable kind, its behaviour when subjected to an exact solution of order $p+1$ will give the exact asymptotic error of the finite element solution. Thus, any estimator can be compared with this exact value and the asymptotic effectivity index can be established. Figure 14.16 shows such a repeatable patch of quadrilateral elements which evaluate the performance for quite irregular meshes.

We have indeed shown how true superconvergent behaviour reproduces exactly such higher order solutions and thus leads to an effectivity index of unity in the

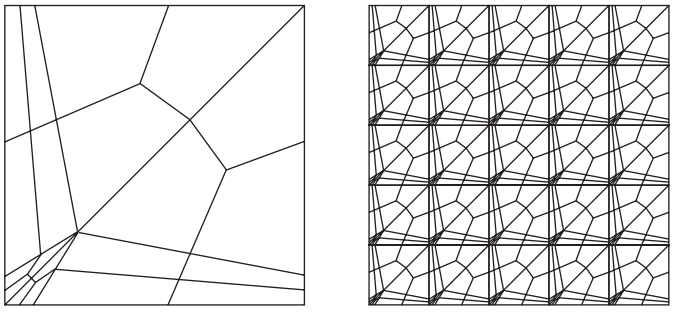


Fig. 14.16 Repeating patch of irregular and quadrilateral elements.

asymptotic limit. In the papers presented by Babuška *et al.*[35–37,41] the procedure of dealing with such repeatable patches for various patterns of two-dimensional elements is developed. Thus, if we are interested in solving the differential equation

$$L(u) + f = 0 \tag{14.85}$$

where L is a linear differential operator of order $2p$, we consider *exact solutions* (harmonic solutions) to the homogeneous equation ($f = 0$) of the form

$$u_{\text{ex}} = \sum_m a_m X^m Y^n = \mathbf{P}(x, y)\mathbf{a}; \qquad n = p + 1 - m \tag{14.86}$$

The boundary conditions are taken as

$$u_{\text{ex}}|_{x+L_x} = u_{\text{ex}}|_x \qquad \text{and} \qquad u_{\text{ex}}|_{y+L_y} = u_{\text{ex}}|_y \tag{14.87}$$

where L_x and L_y are periods in the x and y directions, respectively (viz. repeatability Section 9.18). In general, the individual terms of Eq. (14.86) do not satisfy the differential equation and it is necessary to consider linear combinations in terms of the parameters in L as

$$\mathbf{a}' = \mathbf{T}\mathbf{a} \tag{14.88}$$

This solution serves as the basis for conducting a patch test in which the boundary conditions are assigned to be periodic and to prevent constant changes to u.† The correct constant value may be computed from

$$\int_{\text{patch}} (\mathbf{N}\tilde{\mathbf{u}}_h + C)\, \mathrm{d}\Omega = \int_{\text{patch}} u_{\text{ex}}\, \mathrm{d}\Omega \tag{14.89}$$

To compute upper and lower bounds (θ_U and θ_L) on the possible effectivity indices, all possible combinations of the harmonic solution must be considered. This may be achieved by constructing an error norm of the solutions, for example the L_2 norm of the flux (or stress)

$$||\mathbf{e}_q||^2_{L_2} = \int_{\text{patch}} (\mathbf{q}_{\text{ex}} - \mathbf{q}_h)^{\mathrm{T}} (\mathbf{q}_{\text{ex}} - \mathbf{q}_h)\, \mathrm{d}\Omega = \left(\mathbf{a}'\right)^{\mathrm{T}} \mathbf{T}^{\mathrm{T}} \mathbf{E}_{\text{ex}} \mathbf{T}\mathbf{a}' \tag{14.90}$$

† For elasticity type problems the periodic boundary conditions prevent rigid rotations.

Table 14.1 Robustness index for the equilibrated residuals (ERpB) and SPR (ZZ-discrete) estimators for a variety of anisotropic situations and element patterns, $p = 2$

Estimator	Robustness index
ERpB	10.21
SPR (ZZ-discrete)	0.02

and

$$||\bar{\mathbf{e}}_q||^2_{L_2} = \int_{\text{patch}} (\mathbf{q}_{\text{re}} - \mathbf{q}_h)^{\text{T}} (\mathbf{q}_{\text{re}} - \mathbf{q}_h)\, \mathrm{d}\Omega = \left(\mathbf{a}'\right)^{\text{T}} \mathbf{T}^{\text{T}} \mathbf{E}_{\text{re}} \mathbf{T} \mathbf{a}' \tag{14.91}$$

and solving the eigenproblem

$$\mathbf{T}^{\text{T}} \mathbf{E}_{\text{re}} \mathbf{T} \mathbf{a}' = \theta^2 \mathbf{T}^{\text{T}} \mathbf{E}_{\text{ex}} \mathbf{T} \mathbf{a}' \tag{14.92}$$

to determine the minimum (lower bound) and maximum (upper bound) effectivity indices. Further details of the process summarized here are given in Boroomand and Zienkiewicz[21,22] and by Zienkiewicz *et al.*[42]

These bounds on the effectivity index are very useful for comparing various error estimators and their behaviour for different mesh and element patterns. However, a single parameter called the *robustness index* has also been devised[35] and is useful as a guide to the robustness of any particular estimator

$$R = \max\left(|1 - \theta_L| + |1 - \theta_U|, \quad |1 - \frac{1}{\theta_L}| + |1 - \frac{1}{\theta_U}| \right) \tag{14.93}$$

A large value of this index obviously indicates a poor performance. Conversely the best behaviour is that in which

$$\theta_L = \theta_U = 1 \tag{14.94}$$

and this gives

$$R = 0 \tag{14.95}$$

In the series of tests reported in references 35–41 various estimators have been compared. Table 14.1 shows the highest robustness index value of an equilibrating residual based error estimator and the SPR recovery error estimator for a set of particular patches of triangular elements.[37]

This performance comparison is quite remarkable and it seems that in all the tests quoted by Babuška *et al.*[35–41] the SPR recovery estimator performs best. Indeed we shall observe that in many cases of regular subdivision, when full superconvergence occurs the ideal, asymptotically exact solution characterized by $R = 0$ will be obtained.

In Table 14.2 we show some results obtained for regular meshes of triangles and rectangles with linear and quadratic elements. In the rectangular elements used for problems of heat conduction type, superconvergent points are exact and the ideal result is obtained for both linear and quadratic elements. It is surprising that this

Table 14.2 Effectivity bounds and robustness of SPR and REP recovery estimator for regular meshes of triangles and rectangles with linear and quadratic shape function (applied to heat conduction and elasticity problems). Aspect ratio = length(L)/height(H) of elements in patch tested

Linear triangles and rectangles (heat conduction/elasticity)						
	SPR			REP		
Aspect ratio L/H	θ_L	θ_U	R	θ_L	θ_U	R
1/1	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/2	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/4	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/8	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/16	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/32	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/64	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
Quadratic rectangles (heat conduction)						
	θ_L	θ_U	R	θ_L	θ_U	R
1/1	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/2	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/4	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/8	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/16	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/32	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
1/64	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
Quadratic rectangles (elasticity)						
	θ_L	θ_U	R	θ_L	θ_U	R
1/1	1.0000	1.0000	0.0000	0.9991	1.0102	0.0111
1/2	1.0000	1.0000	0.0000	0.9991	1.0181	0.0189
1/4	1.0000	1.0000	0.0000	0.9991	1.0136	0.0145
1/8	1.0000	1.0000	0.0000	0.9991	1.0030	0.0039
1/16	1.0000	1.0000	0.0000	0.9968	1.0001	0.0033
1/32	1.0000	1.0000	0.0000	0.9950	1.0000	0.0050
1/64	1.0000	1.0000	0.0000	0.9945	1.0000	0.0055
Quadratic triangles (elasticity)						
	θ_L	θ_U	R	θ_L	θ_U	R
1/1	0.9966	1.0929	0.0963	0.9562	1.0503	0.0940
1/2	0.9966	1.0931	0.0965	0.9559	1.0481	0.0923
1/4	0.9967	1.0937	0.0970	0.9535	1.0455	0.0924
1/8	0.9967	1.0943	0.0976	0.9522	1.0603	0.1081
1/16	0.9966	1.0946	0.0980	0.9518	1.0666	0.1148
1/32	0.9966	1.0947	0.0981	0.9517	1.0684	0.1167
1/64	0.9965	1.0947	0.0982	0.9516	1.0688	0.1172

also occurs in elasticity where the proof of superconvergent points is lacking (for $\nu > 0$). Further, the REP procedure also seems to yield superconvergence except for elasticity with quadratic elements.

For regular meshes of quadratic triangles generally superconvergence is not expected and it does not occur for either heat conduction or elasticity problems. However, the robustness index has very small values ($R < 0.10$ for SPR and $R < 0.12$ for REP) and these estimators are therefore very good.

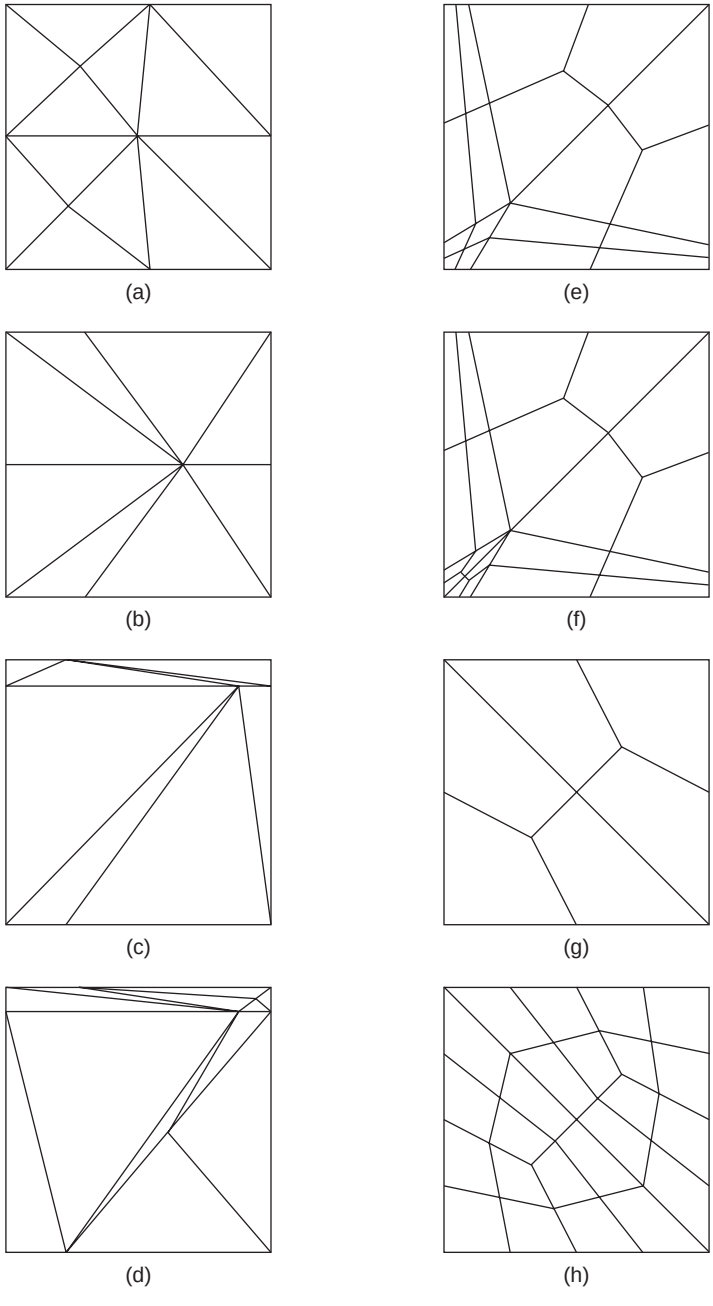


Fig. 14.17.

In Fig. 14.17 and Table 14.3 very irregular meshes of triangular and quadrilateral elements are analysed in repeatable patterns. It is of course not possible to present here all tests conducted by the effectivity patch test. The results shown are, however, typical – others are given in reference 21. It is interesting to observe that the

Table 14.3 Effectivity bounds and robustness of SPR and REP recovery estimator for irregular meshes of triangles (a, b, c, d) and quadrilaterals (e, f, g, h)

Linear element (heat conduction)						
	SPR			REP		
Mesh pattern	θ_L	θ_U	R	θ_L	θ_U	R
a	0.9626	1.0054	0.0442	0.9709	1.0145	0.0443
b	0.9715	1.0156	0.0447	0.9838	1.0167	0.0329
c	0.9228	1.4417	0.5189	0.8938	1.8235	0.9297
d	0.8341	1.2027	0.3685	0.9463	1.9272	0.9810
e	0.9943	1.0175	0.0232	0.9800	1.0589	0.0789
f	0.9969	1.0152	0.0183	0.9849	1.0582	0.0733
g	0.9987	1.0175	0.0188	0.9987	1.0175	0.0188
h	0.9991	1.0068	0.0077	0.9979	1.0062	0.0083
Linear elements (elasticity)						
	SPR			REP		
	θ_L	θ_U	R	θ_L	θ_U	R
a	0.9404	1.0109	0.0741	0.9468	1.0148	0.0707
b	0.8869	1.0250	0.1520	0.9392	1.0275	0.0915
c	0.8550	1.6966	0.8415	0.8037	2.0522	1.2486
d	0.7945	1.2734	0.4788	0.7576	1.9416	1.1840
e	0.9946	1.0247	0.0301	0.9579	1.0508	0.0928
f	1.0038	1.0281	0.0318	0.9612	1.0467	0.0855
g	0.9959	1.0300	0.0341	0.9960	1.0298	0.0338
h	0.9972	1.0139	0.0168	0.9965	1.0122	0.0157
Quadratic elements (heat conduction)						
	θ_L	θ_U	R	θ_L	θ_U	R
a	0.9443	1.0295	0.0877	0.9339	1.0098	0.0805
b	0.8146	1.0037	0.2313	0.9256	1.0028	0.0832
c	0.7640	1.0486	0.3000	0.9559	1.2229	0.2670
d	0.8140	1.0141	0.2423	0.9091	1.2808	0.3717
e	0.9762	1.0053	0.0296	0.9901	1.0177	0.0276
f	0.9691	1.0045	0.0363	0.9901	1.0322	0.0421
g	0.9692	1.0004	0.0322	0.9833	1.0024	0.0195
h	0.9906	1.0113	0.0207	1.0045	1.0261	0.0307
Quadratic elements (elasticity)						
	θ_L	θ_U	R	θ_L	θ_U	R
a	0.9144	1.0353	0.1277	0.9197	1.0244	0.1111
b	0.7302	1.0355	0.4038	0.8643	1.0346	0.1905
c	0.7556	1.1024	0.4163	0.8387	1.2422	0.4035
d	0.7624	1.0323	0.3430	0.8244	1.2632	0.4388
e	0.9702	1.0102	0.0408	0.9682	1.0058	0.0386
f	0.9651	1.0085	0.0446	0.9749	1.0286	0.0537
g	0.9457	1.0115	0.0688	0.9807	1.0125	0.0321
h	0.9852	1.0141	0.0290	0.9996	1.0522	0.0526

performance measured by the robustness index on quadrilateral elements is always superior to that measured on triangles.

The results in a recent paper of Babuška *et al.*[41] show that alternative versions of SPR (such as references 17, 18, 43) generally give much worse robustness index performance than the original version, especially on irregular elements assembled near boundaries.

14.9 Which errors should concern us?

In this chapter we have shown how various recovery procedures can accurately estimate the overall error of the finite element approximation and thus provide a very accurate error estimating method. We have also shown how superior are estimators based on SPR recovery to those based on residual computation. The error estimation discussed here concerns however only the original solution and if the user takes advantage of the recovered values a much better solution is already available. In the next chapter we shall be concerned with adaptivity processes aiming at reduction of the original finite element error for which a vast body of literature already exists. Here again we shall show the excellent values of the effectivity index which can be obtained with SPR type methods on examples for which an 'exact' solution is available from very fine mesh computations. What perhaps we should also be concerned with are the errors remaining in the recovered solutions, if indeed these are to be made use of. This problem is still unsolved and at the moment all the adaptive methods simply aim at the reduction of various norms of error in the finite element solution directly provided.

References

1. J. Barlow. Optimal stress locations in finite element models. *Internat. J. Num. Meth. Eng.*, **10**, 243–51, 1976.
2. L.R. Herrmann. Interpretation of finite element procedures in stress error minimization. *Proc. Am. Soc. Civ. Eng.*, **98**(EM5), 1331–36, 1972.
3. T. Moan. Orthogonal polynomials and 'best' numerical integration formulas on a triangle. *ZAMM*, **54**, 501–8, 1974.
4. E. Hinton and J. Campbell. Local and global smoothing of discontinuous finite element function using a least squares method. *Internat. J. Num. Meth. Eng.*, **8**, 461–80, 1974.
5. M. Krizek and P. Neitaanmaki. On superconvergence techniques. *Acta. Appl. Math.*, **9**, 75–198, 1987.
6. Q.D. Zhu and Q. Lin. *Superconvergence Theory of the Finite Element Methods*. Hunan Science and Technology Press, Hunan, China, 1989.
7. L.B. Wahlbin. *Superconvergence in Galerkin Finite Element Methods. Lectures Notes in Mathematics*, Vol. 1605. Springer, Berlin, 1995.
8. C.M. Chen and Y. Huang. *High Accuracy Theory of Finite Element Methods*. Hunan Science and Technology Press, Hunan, China, 1995.
9. Q. Lin and N. Yan. *Construction and Analyses of Highly Effective Finite Elements*. Hebei University Press, Hebei, China, 1996.
10. H.J. Brauchli and J.T. Oden. On the calculation of consistent stress distributions in finite element applications. *Internat. J. Num. Meth. Eng.*, **3**, 317–25, 1971.

11. O.C. Zienkiewicz and J.Z. Zhu. Superconvergent patch recovery and *a posteriori* error estimation in the finite element method, Part I: A general superconvergent recovery technique. *Internat. J. Num. Meth. Eng.*, **33**, 1331–64, 1992.
12. O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery (SPR) and *a posteriori* error estimates. Part 2: Error estimates and adaptivity. *Internat. J. Num. Meth. Eng.*, **33**, 1365–82, 1992.
13. O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery (SPR) and adaptive finite element refinement. *Comp. Meth. Appl. Mech. Eng.*, **101**, 207–24, 1992.
14. O.C. Zienkiewicz, J.Z. Zhu, and J. Wu. Superconvergent recovery techniques – some further tests. *Commun. Num. Meth. Eng.*, **9**, 251–58, 1993.
15. Z. Zhang. Ultraconvergence of the patch recovery technique. *Math. Comput.*, **65**, 1431–37, 1996.
16. B. Li and Z. Zhang. Analysis of a class of superconvergence patch recovery techniques for linear and bilinear finite elements. *Num. Meth. Partial Diff. Eq.*, **15**, 151–67, 1999.
17. N.-E. Wiberg, F. Abdulwahab, and S. Ziukas. Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions. *Internat. J. Num. Meth. Eng.*, **37**, 3417–40, 1994.
18. T.D. Blacker and T. Belytschko. Superconvergent patch recovery with equilibrium and conjoint interpolation enhancements. *Internat. J. Num. Meth. Eng.*, **37**, 517–36, 1994.
19. N.-E. Wiberg and X.D. Li. Superconvergent patch recovery of finite element solutions and *a posterior* l_2 norm error estimate. *Commun. Num. Meth. Eng.*, **10**, 313–20, 1994.
20. X.D. Li and N.-E. Wiberg. *A posteriori* error estimate by element patch postprocessing, adaptive analysis in energy and L_2 norm. *Comp. Struct.*, **53**, 907–19, 1994.
21. B. Boroomand and O.C. Zienkiewicz. Recovery by equilibrium patches (REP). *Internat. J. Num. Meth. Eng.*, **40**, 137–54, 1997.
22. B. Boroomand and O.C. Zienkiewicz. An improved REP recovery and the effectivity robustness test. *Internat. J. Num. Meth. Eng.*, **40**, 3247–77, 1997.
23. P. Ladevèze and D. Leguillon. Error estimate procedure in the finite element method and applications. *SIAM J. Num. Anal.*, **20**(3), 485–509, 1983.
24. P. Ladevèze, G. Coffignal, and J.P. Pelle. Accuracy of elastoplastic and dynamic analysis. In I. Babuška, O.C. Zienkiewicz, J. Gago, and E.R. de A. Oliviera, editors. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, chapter 11, 1986.
25. O.C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Num. Meth. Eng.*, **24**, 337–57, 1987.
26. I. Babuška and C. Rheinboldt. A-posteriori error estimates for the finite element method. *Internat. J. Num. Meth. Eng.*, **12**, 1597–1615, 1978.
27. I. Babuška and W.C. Rheinboldt. Analysis of optimal finite element meshes in r^1. *Math. Comp.*, **33**, 435–63, 1979.
28. O.C. Zienkiewicz, J.P. De S.R. Gago, and D.W. Kelly. The hierarchical concept in finite element analysis. *Comp. Struct.*, **16**(53-65), 53–65, 1983.
29. D.W. Kelly, J.P. De S.R. Gago, O.C. Zienkiewicz, and I. Babuška. *A posteriori* error analysis and adaptive processes in the finite element method: Part 1 – Error analysis. *Internat. J. Num. Meth. Eng.*, **19**, 1593–1619, 1983.
30. R.E. Bank and A. Weiser. Some *a posteriori* error estimators for elliptic partial differential equations. *Math. Comput.*, **44**, 283–301, 1985.
31. J.T. Oden, L. Demkowicz, W. Rachowicz, and Westermann T, A. Toward a universal h-p adaptive finite element strategy. Part 2: *A posteriori* error estimation. *Comp. Meth. Appl. Mech. Eng.*, **77**, 113–80, 1989.
32. R. Verfurth. *A posteriori* error estimators for the stokes equations. *Numer. Math.*, **55**, 309–25, 1989.

33. C. Johnson and P. Hansbo. Adaptive finite element methods in computational mechanics. *Comp. Meth. Appl. Mech. Eng.*, **101**, 143–81, 1992.
34. M. Ainsworth and J.T. Oden. A unified approach to a posteriors error estimation using element residual methods. *Numerische Mathematik*, **65**, 23–50, 1993.
35. I. Babuška, T. Strouboulis, and C.S. Upadhyay. A model study of the quality of *a posteriori* error estimators for linear elliptic problems. Error estimation in the interior of patchwise uniform grids of triangles. *Comp. Meth. Appl. Mech. Eng.*, **114**, 307–78, 1994.
36. I. Babuška, T. Strouboulis, C.S. Upadhyay, S.K. Gangaraj, and K. Copps. Validation of *a posteriori* error estimators by numerical approach. *Internat. J. Num. Meth. Eng.*, **37**, 1073–1123, 1994.
37. I. Babuška, T. Strouboulis, C.S. Upadhyay, S.K. Gangaraj, and K. Copps. An objective criterion for assessing the reliability of *a posteriori* error estimators in finite element computations. *U.S.A.C.M. Bulletin*, No. **7**, 4–16, 1994.
38. P. Ladevéze, J.P. Pelle, and P. Rougeot. Error estimation and mesh optimization for classical finite elements. *Engng. Comput.*, **8**, 69–80, 1991.
39. J.Z. Zhu and O.C. Zienkiewicz. Superconvergence recovery technique and *a posteriori* error estimators. *Internat. J. Num. Meth. Eng.*, **30**, 1321–39, 1990.
40. J.Z. Zhu. *A posteriori* error estimation – the relationship between different procedures. *Comp. Meth. Appl. Mech. Eng.*, **150**, 411–22, 1997.
41. I. Babuška, T. Strouboulis, and C.S. Upadhyay. A model study of the quality of *a posteriori* error estimators for finite element solutions of linear elliptic problems, with particular reference to the behavior near the boundary. *Internat. J. Num. Meth. Eng.*, **40**, 2521–77, 1997.
42. O.C. Zienkiewicz, B. Boroomand, and J.Z. Zhu. Recovery procedures in error estimation and adaptivity: Adaptivity in linear problems. In P. Ladevèze and J.T. Oden, editors, *Advances in Adaptive Computational Mechanics in Mechanics*, pages 3–23. Elsevier Science Ltd., 1998.
43. N.-E. Wiberg and F. Abdulwahab. Patch recovery based on superconvergent derivatives and equilibrium. *Internat. J. Num. Meth. Eng.*, **36**, 2703–24, 1993.

15

Adaptive finite element refinement

15.1 Introduction

In the previous chapter we have discussed at some length various methods of recovery by which the finite element solution results could be made more accurate and this led us to devise various procedures for error estimation. In this chapter we shall be concerned with methods which can be used to reduce the errors generally once a finite element solution has been obtained. As the process depends on previous results at all stages it is called adaptive. Such adaptive methods were first introduced to finite element calculations by Babuška and Rheinbolt in the late 1970s.[1,2] Before proceeding further it is necessary to clarify the objectives of refinement and specify 'permissible error magnitudes' and here the engineer or user must have very clear aims. For instance the naive requirement that all displacements or all stresses should be given within a specified tolerance is not always acceptable. The reasons for this are obvious as at singularities, for example, stresses will always be infinite and therefore no finite tolerance could be specified. The same difficulty is true for displacements if point or knife edge loads are considered. The most common criterion in general engineering use is that of prescribing a total limit of the error computed in the energy norm. Often this error is required not to exceed a specified percentage of the total energy norm of the solution and in the many examples presented later we shall use this criterion. However, using a recovery type of error estimator it is possible to adaptively refine the mesh so that the accuracy of a certain quantity of interest, such as the RMS error in displacement and/or RMS error in stress (see Chapter 14, Eqs. (14.10a) and (14.10b)), satisfy some user-specified criterion. We should recognize that mesh refinement based on reducing the RMS error in displacement is in effect reducing the average displacement error in each element; similarly mesh refinement based on reducing the RMS error in stress is the same as reducing the average stress error in each element. Here we could, for instance, specify directly the permissible error in stresses or displacements at any location. Some investigators (e.g., Zienkiewicz and Zhu[3]) have used RMS error in stress in the adaptive mesh refinement to obtain more accurate stress solutions. Others (e.g., Oñate and Bugeda[4]) have used the requirement of constant energy norm density in the adaptive analysis, which is in fact equivalent to specifying a uniform distribution of RMS error in stress in each element. We note that the recovery type of error estimators are particularly useful

and convenient in designing adaptive analysis procedures for the quantities of interest.

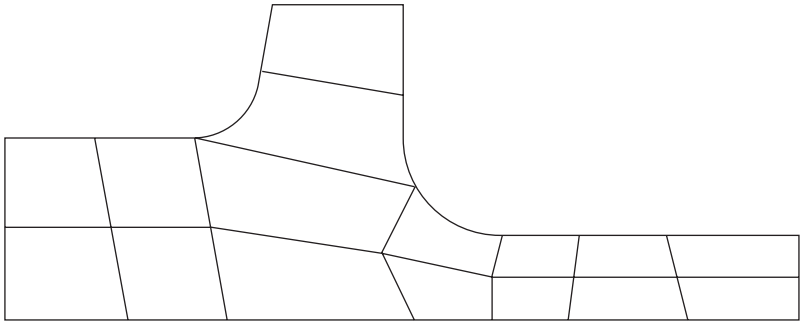
As we have already remarked in the previous chapter we will at all times consider the error in the actual finite element solution rather than the error in the recovered solution. It may indeed be possible in special problems for the error in the recovered solution to be zero, even if the error in the finite element solution itself is quite substantial. (Consider here for instance a problem with a linear stress distribution being solved by linear elements which result in constant element stresses. Obviously the element error will be quite large. But if recovered stresses are used, exact results can be obtained and no errors will exist.) The problem of which of the errors to consider still needs to be answered. At the present time we shall consider the question of recovery as that of providing a very substantial margin of safety in the definition of errors.

Various procedures exist for the refinement of finite element solutions. Broadly these fall into two categories:

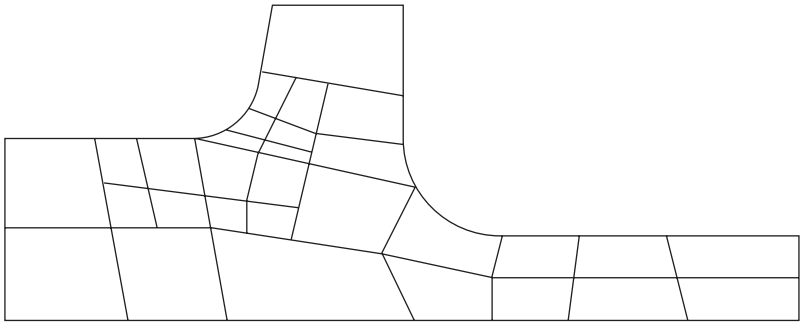
1. The *h*-refinement in which the same class of elements continue to be used but are changed in size, in some locations made larger and in others made smaller, to provide maximum economy in reaching the desired solution.
2. The *p*-refinement in which we continue to use the same element size and simply increase, generally hierarchically, the order of the polynomial used in their definition.

It is occasionally useful to divide the above categories into subclasses, as the *h*-refinement can be applied and thought of in different ways. In Fig. 15.1 we illustrate three typical methods of *h*-refinement.

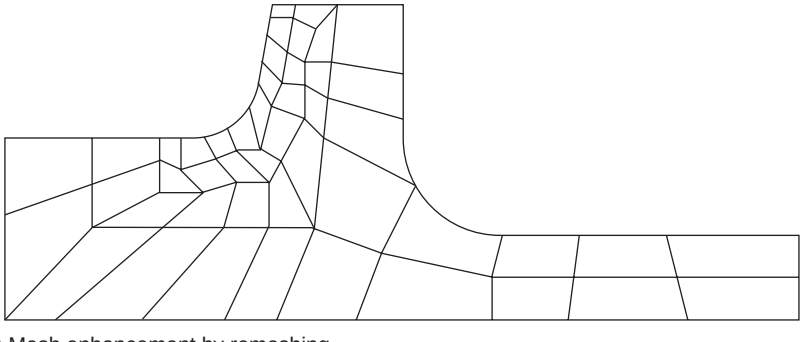
1. The first of these *h*-refinement methods is *element subdivision* (enrichment) [Fig. 15.1(b)]. Here refinement can be conveniently implemented and existing elements, if they show too much error, are simply divided into smaller ones keeping the original element boundaries intact. Such a process is cumbersome as many *hanging points* are created where an element with mid-side nodes is joined to a linear element with no such nodes. On such occasions it is necessary to provide local constraints at the hanging points and the calculations become more involved. In addition, the implementation of de-refinement requires rather complex data management which may reduce the efficiency of the method. Nevertheless, the method of element subdivision is quite widely used.
2. The second method is that of a complete *mesh regeneration or remeshing* [Fig. 15.1(c)]. Here, on the basis of a given solution, a new element size is predicted in all the domain and a totally new mesh is generated. Thus a refinement and de-refinement are simultaneously allowed. This of course can be expensive, especially in three dimensions where mesh generation is difficult for certain types of elements, and it also presents a problem of transferring data from one mesh to another. However, the results are generally much superior and this method will be used in most of the examples shown in this chapter. For many practical engineering problems, particularly of those for which the element shape will be severely distorted during the analysis, adaptive mesh regeneration is a natural choice.
3. The final method, sometimes known as *r-refinement* [Fig. 15.1(d)], keeps the total number of nodes constant and adjusts their position to obtain an optimal



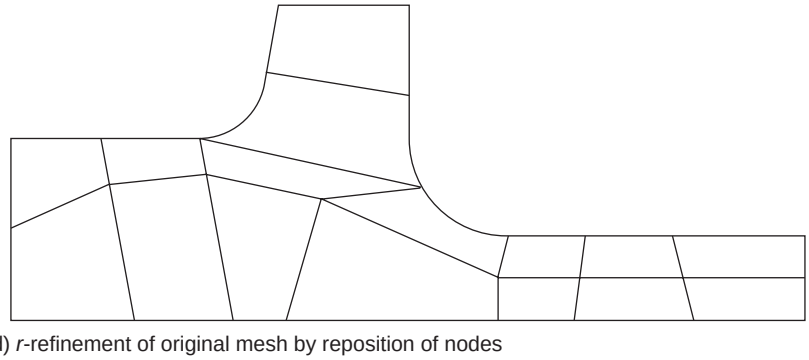
(a) Original mesh



(b) Mesh enhancement by subdivision (enrichment)



(c) Mesh enhancement by remeshing



(d) *r*-refinement of original mesh by reposition of nodes

Fig. 15.1 Various procedures by *h*-refinement.

approximation.[5–7] While this procedure is theoretically of interest it is difficult to use in practice and there is little to recommend it. Further it is not a true refinement procedure as a prespecified accuracy cannot generally be reached.

We shall see that with energy norms specified as the criterion, it is a fairly simple matter to predict the element size required for a given degree of approximation. Thus very few re-solutions are generally necessary to reach the objective.

With *p*-refinement the situation is different. Here two subclasses exist:

1. One in which the polynomial order is increased uniformly throughout the whole domain;
2. One in which the polynomial order is increased locally using hierarchical refinement.

In neither of these has a direct procedure been developed which allows the prediction of the best refinement to be used to obtain a given error. Here the procedures generally require more resolutions and tend to be more costly. However, the convergence for a given number of variables is more rapid with *p*-refinement and it has much to recommend it.

On occasion it is possible to combine efficiently the *h*- and *p*-refinements and call it the *hp*-refinement. In this procedure both the size of elements *h* and their degree of polynomial *p* are altered. Much work has been reported in the literature by Babuška, Oden and others and the interested reader is referred to the references.[8–18] In the next two sections, Sec. 15.2 and 15.3, we shall discuss both the *h*- and the *p*-refinements. In Sec. 15.3 we also include some details of the very simple and yet efficient *hp*-refinement process introduced by Zienkiewicz, Zhu and Gong.[19]

15.2 Some examples of adaptive *h*-refinement

15.2.1 Mesh regeneration procedures

In the introduction to this chapter we have mentioned several alternative processes of *h*-adaptivity and we suggested that the process in which the complete mesh is regenerated is in general the most efficient. Such a procedure allows elements to be de-refined (or enlarged) as well as refined (made smaller) and invariably starts at each stage of the analysis from a specification of the mesh size defined at each nodal point of the previous mesh. Standard interpolation is used to find the size of elements required at any point in the domain. This interpolation helps in the refinement subsequently. Indeed at the starting point an initial mesh need not include the boundaries of the problem as it will be used only to interpolate the sizes required in the domain during the process of mesh generation. However, after this first stage of analysis as the refinement proceeds the mesh sizes will be specified at the nodes of the last mesh.

In Chapter 9 of this book, where we discussed mapping, we also discussed various possible mesh generators. These did not allow a mesh size variation of the refined kind to be specified. In adaptivity it is very important to be able to define quite precisely the element size or density of mesh so that a minimum number of elements can be used.

The generators which can do this have been developed since the mid-1980s. The first of these by Peraire *et al.*[20] was applied to aerospace engineering and fluid

mechanics calculations. Its basis is the *frontal method* of mesh generation developed originally by Cavendish[21] and Lo[22] and the original generator was made available only for triangular elements. Later such generators were generalized to include tetrahedral elements in three-dimensional space.[23] Today both triangular and tetrahedral generators form the basis of most adaptive codes.

Extension to quadrilateral and hexahedral elements is by no means easy. First, procedures for generating quadrilateral elements in two dimensions have been devised. The work of Zhu and Zienkiewicz[24,25] and Rank *et al.*[26,27] has to be noted. The procedures are based on the joining of two triangles into a quadrilateral at different stages of the mesh generation process.

However, so far no extension of such methodologies to hexahedral elements in space have been made. To the knowledge of the authors no efficient hexahedral mesh generators exist for adaptivity, though very many attempts have been reported in the literature.[28–33]

In the more recent mesh generators used for both triangles and tetrahedra the frontal procedure has been largely replaced by *Delauney triangulation* and the reader is well advised to consult the following references and texts.[34,35]

15.2.2 Predicting the required element size in *h* adaptivity

The error estimators discussed in the previous chapter allow the global energy (or similar) norm of the error to be determined and the errors occurring locally (at the element level) are usually also well represented. If these errors are within the limits prescribed by the analyst then clearly the work is completed. More frequently these limits are exceeded and refinement is necessary. The question which this section addresses is how best to effect this refinement. Here obviously many strategies are possible and much depends on the *objectives* to be achieved.

In the simplest case we shall seek, for instance, to make the relative energy norm percentage error η less than some specified value $\bar{\eta}$ (say 5% in many engineering applications). Thus

$$\eta \leqslant \bar{\eta} \tag{15.1}$$

is to be achieved.

In an 'optimal mesh' it is desirable that the distribution of energy norm error (i.e., $||\mathbf{e}||_k$) should be equal for all elements. Thus if the total permissible error is determined (assuming that it is given by the result of the approximate analysis) as

$$\text{Permissible error} \equiv \bar{\eta}||\mathbf{u}|| \approx \bar{\eta}\left(||\hat{\mathbf{u}}||^2 + ||\mathbf{e}||^2\right)^{1/2} \tag{15.2}$$

here we have used[36]

$$||\mathbf{e}||^2 = ||\mathbf{u}||^2 - ||\hat{\mathbf{u}}||^2 \tag{15.3}$$

We could pose a requirement that the error in any element k should be

$$||\mathbf{e}||_k < \bar{\eta}\left(\frac{||\hat{\mathbf{u}}||^2 + ||\mathbf{e}||^2}{m}\right)^{1/2} \equiv \bar{e}_m \tag{15.4}$$

where m is the number of elements involved.

Elements in which the above is not satisfied are obvious candidates for refinement. Thus if we define the ratio

$$\frac{||\mathbf{e}||_k}{\bar{e}_m} = \xi_k \tag{15.5}$$

we shall refine whenever†

$$\xi_k > 1 \tag{15.6}$$

ξ_k can be approximated, of course, by replacing the true error in Eqs (15.4) and (15.5) with the error estimators.

The refinement could be carried out progressively by refining only a certain number of elements in which ξ is higher than a certain limit and at each time of refining halve the size of such elements. This type of element subdivision process is also known as *mesh enrichment*. This process of refinement though ultimately leading to a satisfactory solution being obtained with a relatively small number of total degrees of freedom, is in general not economical as the total number of trial solutions may be excessive.

It is more efficient to try to design a completely new mesh which satisfies the requirement that

$$\xi_k \leqslant 1 \tag{15.7}$$

One possibility here is to invoke the asymptotic convergence rate criteria at the element level (although we have seen that these are not realistic in the presence of singularities) and to predict the element size distribution. For instance, if we assume

$$||\mathbf{e}||_k \propto h_k^p \tag{15.8}$$

where h_k is the current element size and p the polynomial order of approximation, then to satisfy the requirement of Eq. (15.4) the new generated element size should be no larger than

$$h_{\text{new}} = \xi_k^{-1/p} h_k \tag{15.9}$$

Mesh generation programs in which the local element size can be specified are available now as we have already stated and these can be used to design a new mesh for which the re-analysis is carried out.[20,24] In the figures we show how starting from a relatively coarse solution a single mesh prediction often allows a solution (almost) satisfying the specified accuracy requirement to be achieved.

The reason for the success of the mesh regeneration based on the simple assumption of asymptotic convergence rate implied in Eq. (15.8) is the fact that with refinement the mesh tends to be 'optimal' and the localized singularity influence no longer affects the overall convergence.

Of course the effects of singularity will still remain present in the elements adjacent to it and improved mesh subdivision can be obtained if in such elements we use the appropriate convergence and write, if in Eqs (15.8) and (15.9) p is replaced by λ, see Chapter 14, Eq. (14.17)

$$h_{\text{new}} = \xi_k^{-1/\lambda} h_k \tag{15.10}$$

† We can indeed 'de-refine' or use a larger element spacing where $\xi_k < 1$ if computational economy is desired.

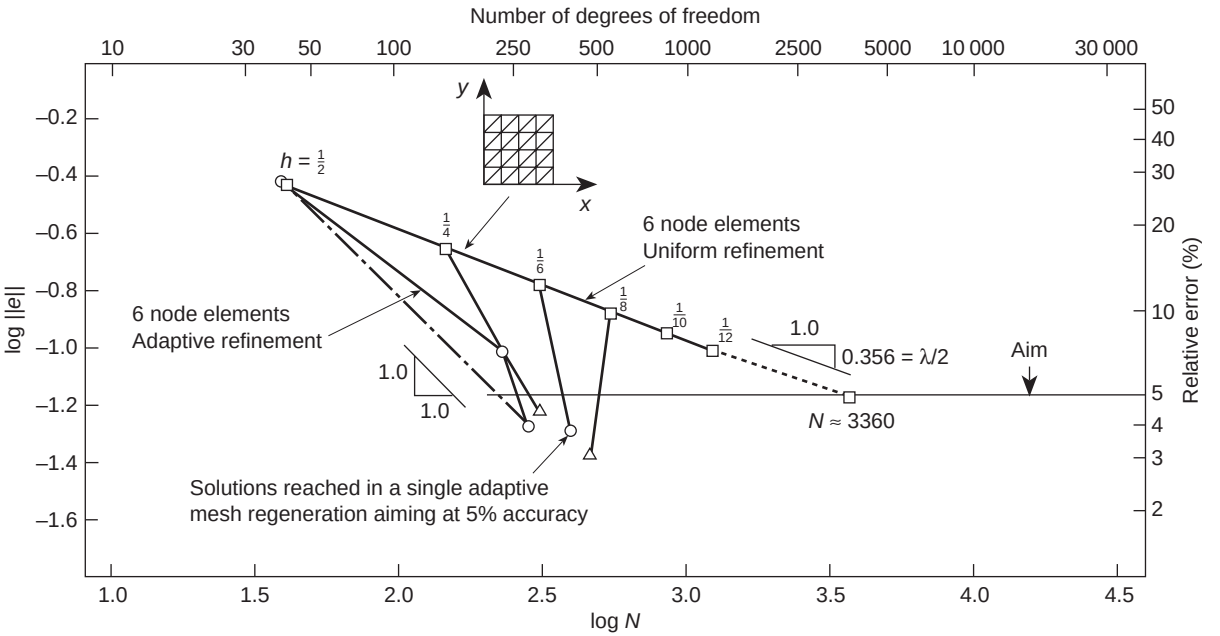


Fig. 15.2 The influence of initial mesh to convergence rates in *h* version. Adaptive refinement using quadratic triangular elements. Problem of Fig. 15.3. Note that if initial mesh is finer than $h = 1/8$ adaptive refinement reduces the number of equations.

in which λ is the singularity strength. A convenient number to use here is $\lambda = 0.5$ as most singularity parameters lie in the range 0.5–1.0. With this procedure, added to the refinement strategy, we frequently achieve accuracies better than the prescribed limit in one remeshing.

In the examples which follow we will show in general a process of refinement in which the total number of degrees of freedom increases with each stage, even though the mesh is redesigned. This need not necessarily be the case as a fine but badly structured mesh can show much greater error than a near-optimal one. To illustrate this point we show in Fig. 15.2 the one stage refinement designed to reach

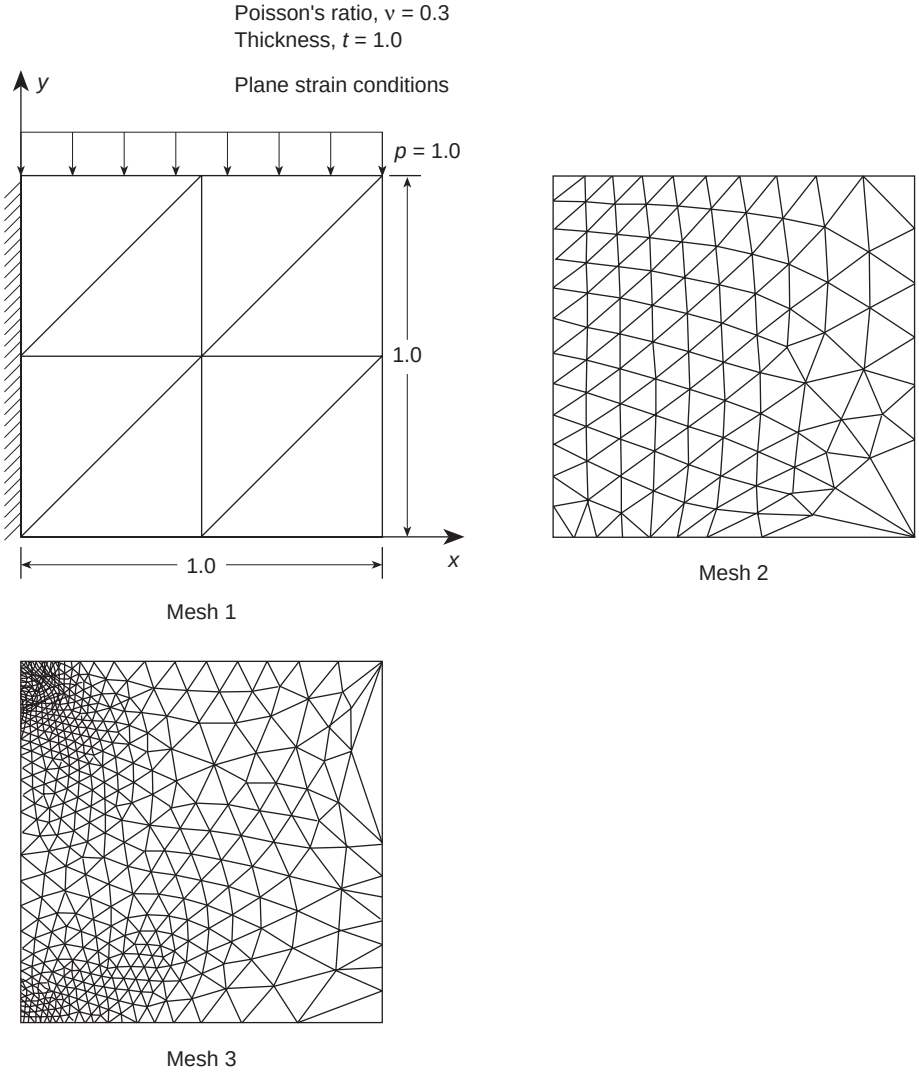


Fig. 15.3 Short cantilever beam and adaptive meshes of linear triangular elements.

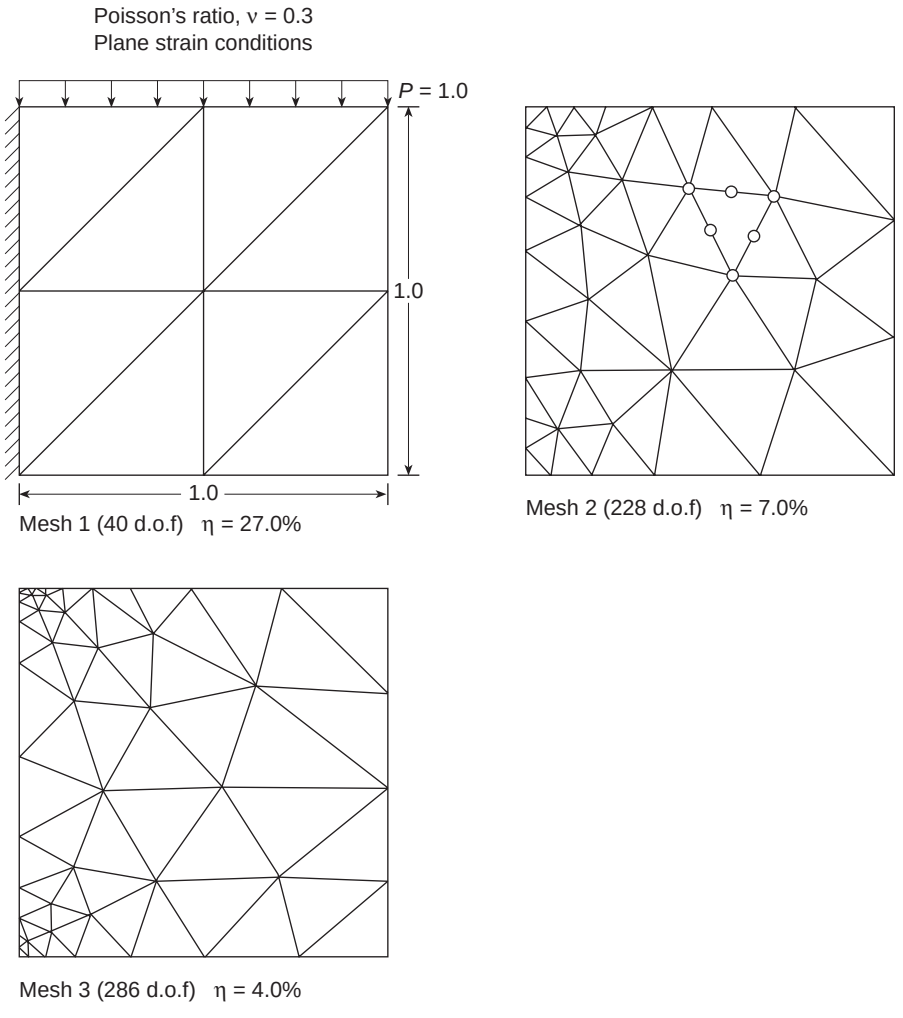
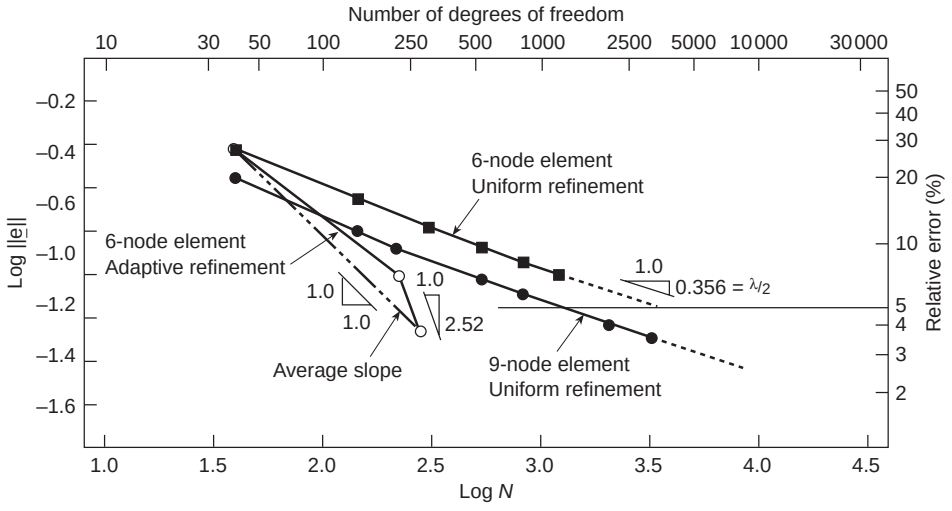


Fig. 15.4 Adaptive mesh of quadratic triangular elements for short cantilever beam.

5% accuracy in one step starting from uniform mesh subdivisions. The problem here is the same as illustrated in Figs 15.3, 15.4, and 15.5 and in the refinement process we use both the mesh criteria of Eqs (15.9) and (15.10).[37] This problem refers to a short stubby cantilever beam in which two very high singularities exist at the corners attached to a rigid wall. In Fig. 15.4 we show three stages of an adaptive solution and in Fig. 15.5 we indicate how rapidly this converges although all uniform refinements converge at a very slow rate (due to the singularities).

We note that now, in at least one refinement, a decrease of total error occurs with a reduction of total degrees of freedom (starting from a uniform 8×8 subdivision with $NDF = 544$ and $\eta = 9.8\%$ to $\eta = 3.1\%$ with $NDF = 460$).

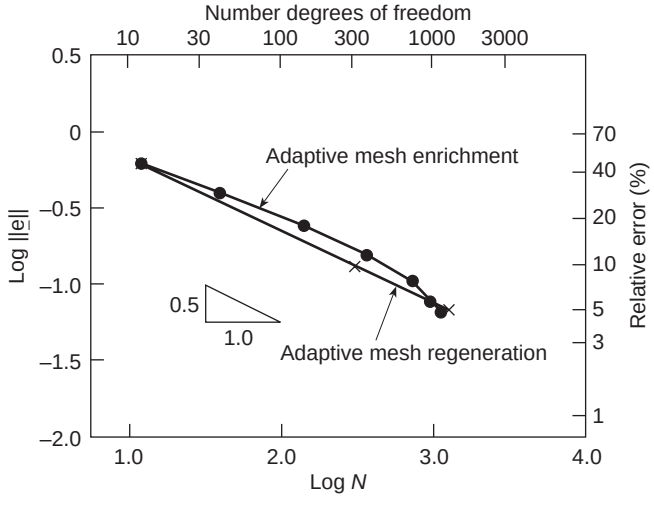


$\lambda/2$ = 0.356, theoretical rate of convergence for uniform refinement

$P/2$ = 1.0, maximum rate of convergence

Fig. 15.5 Experimental rates of convergence for short cantilever beam.

The same problem is also solved by both mesh enrichment and mesh regeneration using linear quadrilateral elements to achieve 5% accuracy. The prescribed accuracy is obtained with optimal rate of convergence being reached by both adaptive refinement processes (Fig. 15.6). However, the mesh enrichment method requires seven



$P/2$ = 0.5, optimal rate of convergence

Fig. 15.6 Short cantilever beam. Mesh enrichment versus mesh regeneration using linear quadrilateral elements.

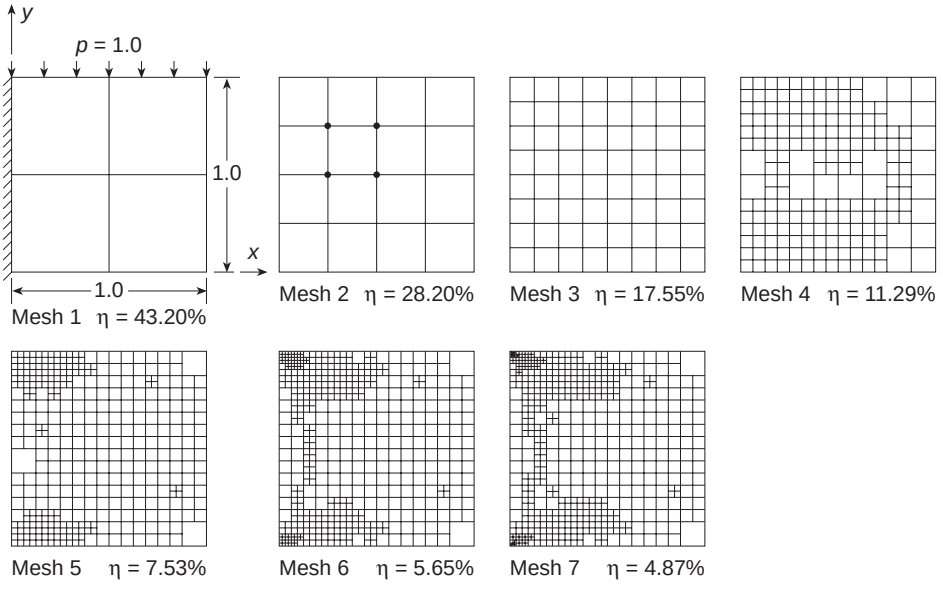


Fig. 15.7 Short cantilever solved by mesh enrichment. Linear quadrilateral elements.

refinements, as shown in Fig. 15.7, while mesh regeneration requires only three (see Fig. 15.8). Here the refinement criterion, Eq. (15.8), is used for the mesh enrichment process.

As we mentioned earlier, the value of the energy norm error is not necessarily the best criterion for practical refinement. Limits on the local stress error can be used effectively. Such errors are quite simply obtained by the recovery processes described in the previous chapter (SPR in Section 14.4 and REP in Section 14.5). In Fig. 15.9 we show a simple exercise recently conducted by Oñate and Bugeda[4] in which a refinement of a stressed cylinder is made using various criteria as described in the caption of Fig. 15.9. It will be observed that the stress tolerance method generally needs a much finer mesh.

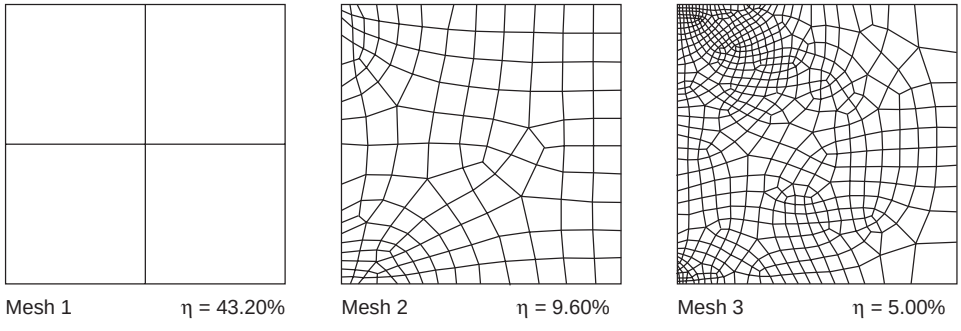


Fig. 15.8 Short cantilever solved by mesh regeneration. Linear quadrilateral elements.

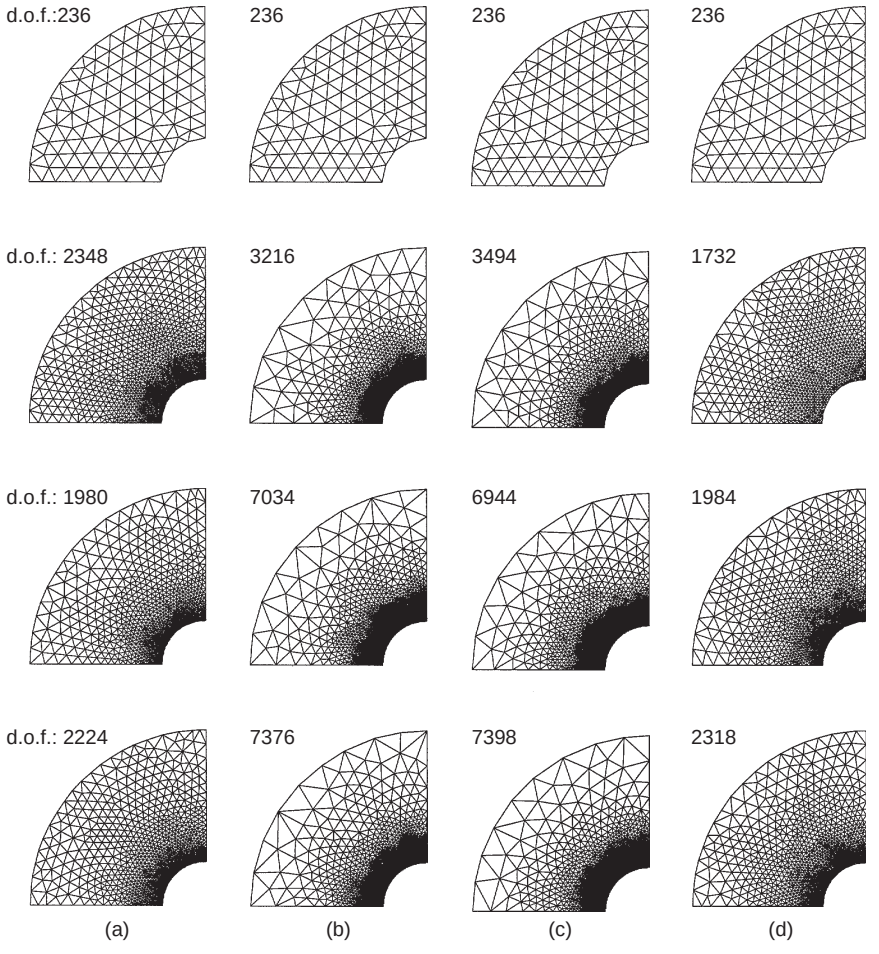


Fig. 15.9 Sequence of adaptive mesh refinement strategies based on (a) equal distribution of the global energetic error between all the elements, (b) equal distribution of the density of energetic error, (c) equal distribution of the maximum error in stresses at each point, and (d) equal distribution of the maximum percentage of the error in stresses at each point. All final meshes have less than 5% energy norm error.

15.2.3 Some further examples

We shall now present further typical examples of *h*-refinement with mesh adaptivity. In all of these, full mesh regeneration is used at every step.

Example 1. A Poisson equation in a square domain This example is fairly straightforward and starts from a simple square domain in which suitable loading terms exist in a Poisson equation to give the solution shown in Fig. 15.10. In Fig. 15.11 we show the first subdivision of this domain into regular linear and quadratic elements and the subsequent refinements. The elements are of both triangular and

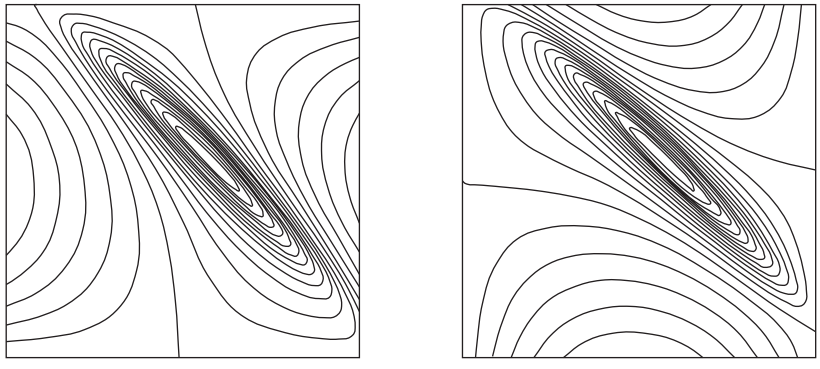


Fig. 15.10 Poisson equation 'exact' solutions. (a) $\partial u/\partial x$ contours; (b) $\partial u/\partial y$ contours.

quadrilateral shape and for the linear ones a target error of 10% in total energy has been set, while for quadratic elements the target error is 1% of total energy. In practically all cases three refinements suffice to reach a very accurate solution satisfying the requirements despite the fact that the original mesh cannot capture in any way the high intensity region illustrated in the previous figure. It is of interest to note that the effectivity indices in all cases are very good – this is true even for the original refinement. Figure 15.12 shows the convergence for various elements with the error plotted against the total number of degrees of freedom. The reader should note that the asymptotic rate of convergence is exceeded when the refinement gets closer to its final objective.

Example 2. An L-shaped domain It is of interest to note the results in Fig. 15.13 which come from an analysis of a re-entrant corner using isoparametric quadratic quadrilaterals. Here a single refinement is shown together with the convergence of the solution.

Example 3. A machine part For this machine part problem plane strain conditions are assumed. A prescribed accuracy of 5% relative error is achieved in one adaptive refinement (see Fig. 15.14) with linear quadrilateral elements. The convergence of the shear stress τ_{xy} is shown in Fig. 15.15.

Example 4. A perforated gravity dam The final example of this section shows a more practical engineering problem of a perforated dam. This dam was analysed in the late 1960s during its construction. More recently, the problem was given to a young engineer to choose a suitable mesh of quadratic triangles. Figure 15.16(a) shows the mesh chosen. Despite the high order of elements the error is quite high, being around 17%. One stage of refinement with a specified value of 5% error in energy norm reaches this in a single operation. As we have seen in previous examples such convergence is not always possible but it is achieved here. We believe this typical

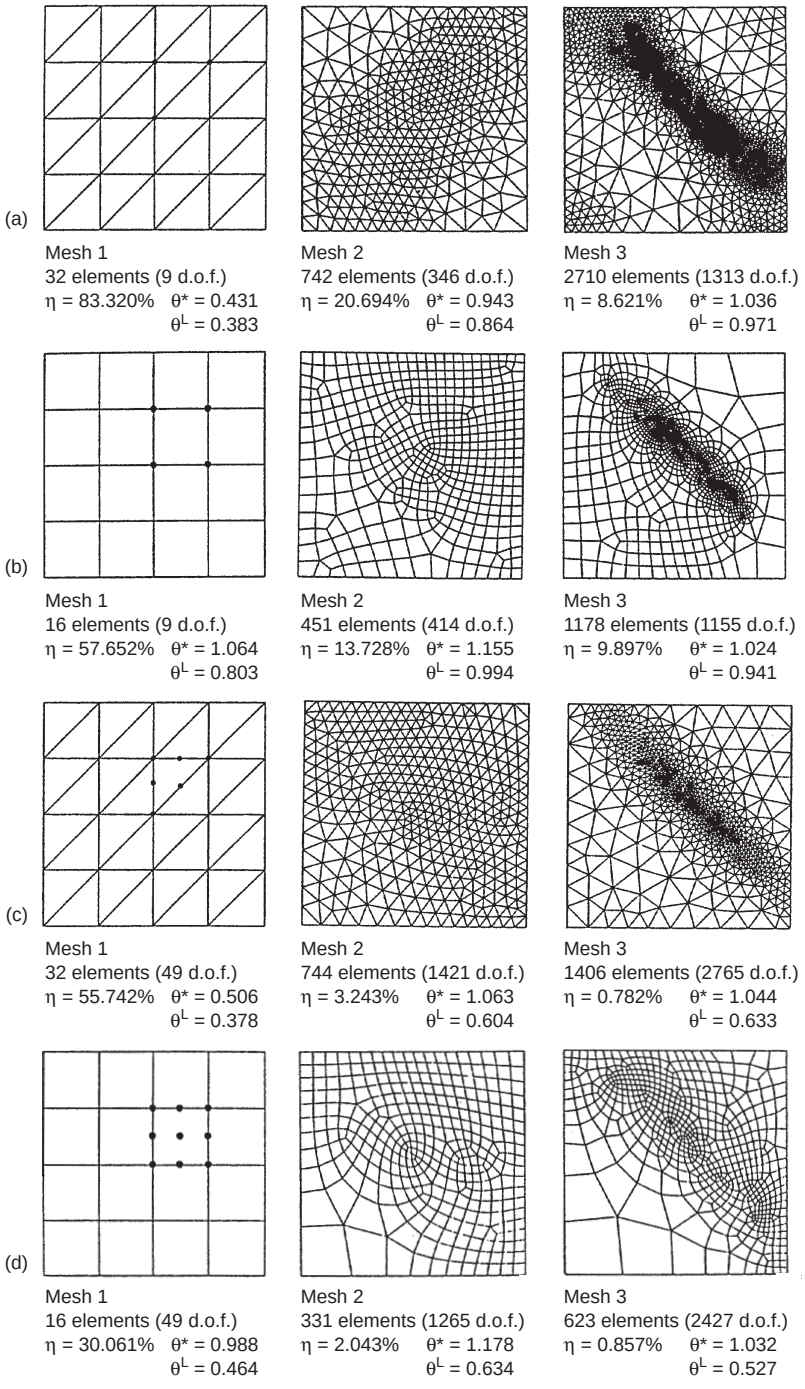


Fig. 15.11 Poisson problem of Fig. 15.10. Adaptive solutions for: (a) linear triangles; (b) linear quadrilaterals; (c) quadratic triangles; (d) quadratic quadrilaterals. θ^* based on SPR, θ^L based on L_2 projection. Target error 10% for linear elements and 1% for quadratic elements.

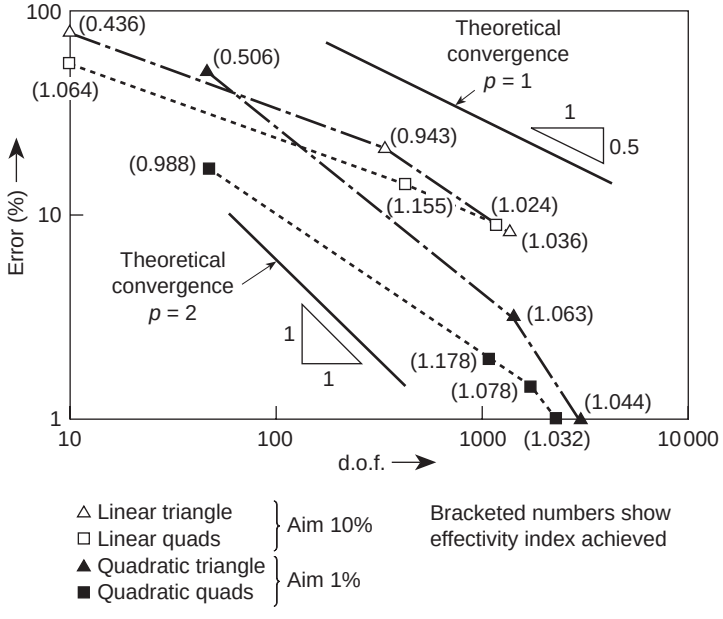


Fig. 15.12 Adaptive refinement for Poisson problem of Fig. 15.10.

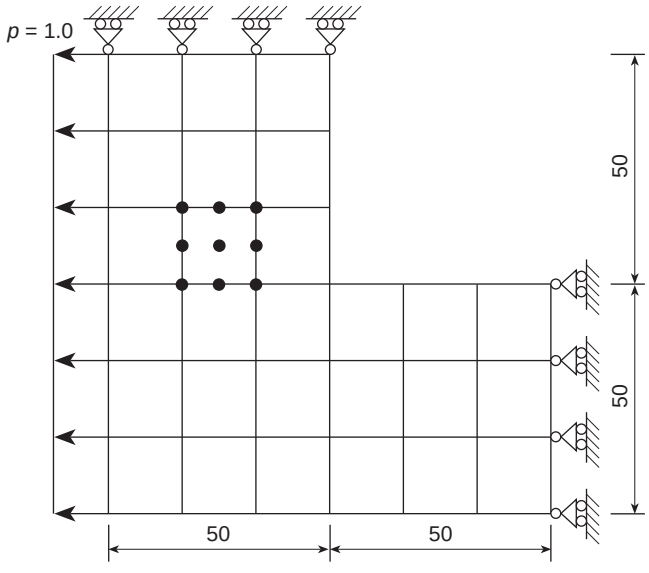
example shows the advantages of adaptivity and the ease with which a final good mesh can be arrived at automatically.

15.3 *p*-refinement and *hp*-refinement

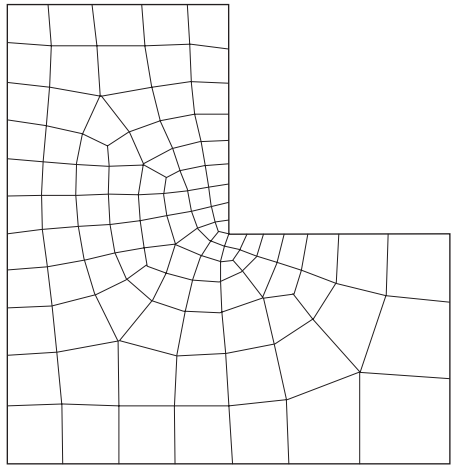
The use of non-uniform *p*-refinement is of course possible if done hierarchically and many attempts have been made to do this efficiently. Some of this was done as early as 1983.[38,39] However, the general process is difficult and necessitates many assumptions about the decrease of error. Certainly, the desired accuracy can seldom be obtained in a single step and most of the work on this requires a sequence of steps. We illustrate such a refinement process in Fig. 15.17 for the perforated dam problem presented in the previous section.

The same applies to *hp*-processes in which much work has been done during the last decade.[8–18] We shall quote here only one particular attempt at *hp*-refinement which seems to be particularly efficient and where the number of resolutions is quite small. The methodology was introduced by Zienkiewicz *et al.* in 1989[19] and we shall quote here some of the procedures suggested.

The first procedure is that of pursuing an *h*-refinement with lower order elements (e.g., linear or quadratic elements) to obtain, say, a 5% accuracy, at which stage the energy norm error is nearly uniformly distributed throughout all elements. From there a *p*-refinement is applied in a uniform manner (i.e., the same *p* is used in all elements). This has very substantial computational advantages as programming is easy and can be readily accomplished, especially if hierarchical functions are used.



Mesh 1: 27 elements (252 DOF)
η = 8.3%, θ* = 1.110



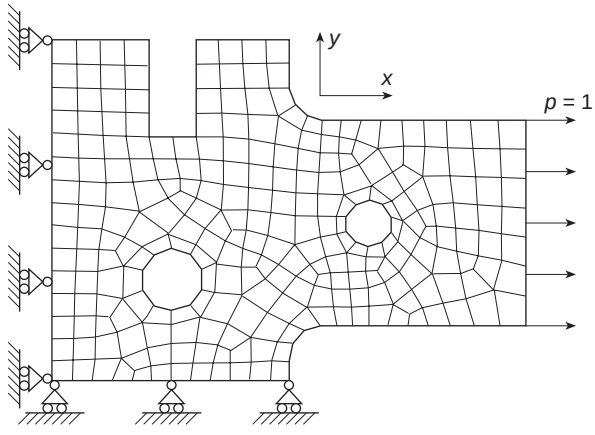
Mesh 2: 101 elements (876 DOF)
η = 3.1%, θ* = 1.057

Fig. 15.13 Adaptive refinement of an L-shaped domain in plane stress with prescribed error of 1%.

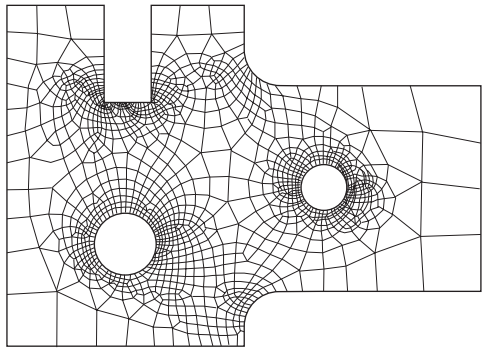
The uniform *p*-refinement also allows the global energy norm error to be approximately extrapolated by three consecutive solutions.[40]

The convergence of the *p*-refinement finite element solution can be written as[41]

$$||\mathbf{e}|| \leqslant CN^{-\beta} \tag{15.11}$$

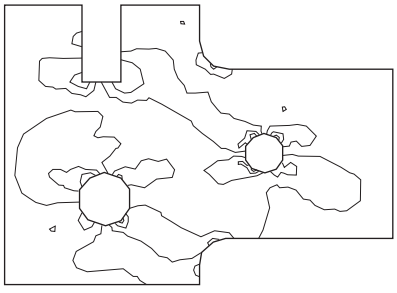


(a) Mesh 1 (565 d.o.f.) $\eta = 9.75\%$

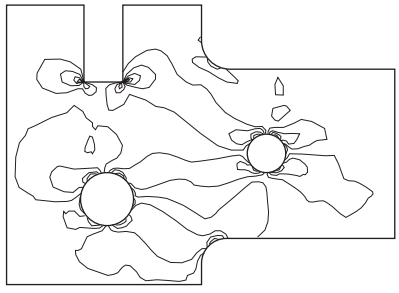


(b) Mesh 2 (3155 d.o.f.) $\eta = 4.85\%$

Fig. 15.14 Adaptive refinement of machine part using linear quadrilateral elements. Target error 5%.

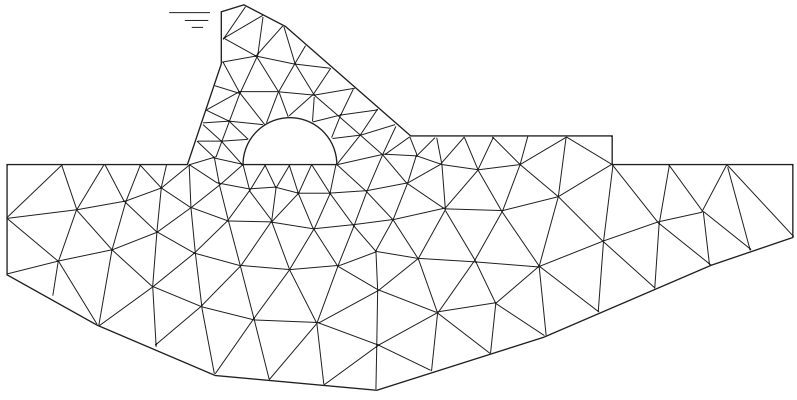


Base contour value = –1.833000
Maximum contour value = 0.586500
Contour interval = 0.163095



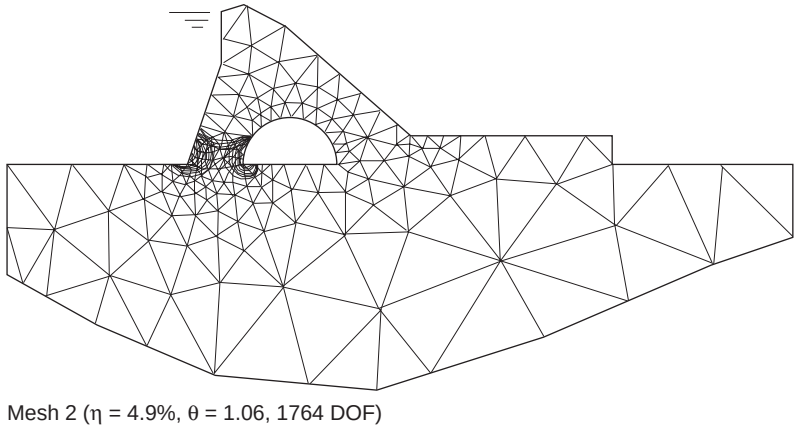
Base contour value = –1.833000
Maximum contour value = 0.586500
Contour interval = 0.163095

Fig. 15.15 Adaptive refinement of machine part. Contours of shear stress for original and final mesh.



Mesh 1 (η = 16.5%, θ = 1.05, 728 DOF)

(a)



Mesh 2 (η = 4.9%, θ = 1.06, 1764 DOF)

(b)

Fig. 15.16 Quadratic triangle. Automatic mesh generation to achieve 5% accuracy. Plane strain analysis of a dam with perforation, water loading only. (a) Original mesh. (b) Refined mesh.

where C and β are positive constants depending on the solution of the problem and N is the number of degrees of freedom. We assume that for each refinement the error is, observing Eq. (15.3),

$$||\mathbf{u}||^2 - ||\hat{\mathbf{u}}_q||^2 = CN_q^{-2\beta} \tag{15.12}$$

with $q = p-2, p-1, p$ for the three solutions. Eliminating the two constants C and β from the above three equations, $||\mathbf{u}||^2$ can be solved by

$$\frac{||\mathbf{u}||^2 - ||\mathbf{u}_p||^2}{||\mathbf{u}||^2 - ||\hat{\mathbf{u}}_{p-1}||^2} = \left(\frac{||\mathbf{u}||^2 - ||\mathbf{u}_{p-1}||^2}{||\mathbf{u}||^2 - ||\hat{\mathbf{u}}_{p-2}||^2}\right)^{\frac{\log(N_{p-1}/N_p)}{\log(N_{p-2}/N_{p-1})}} \tag{15.13}$$

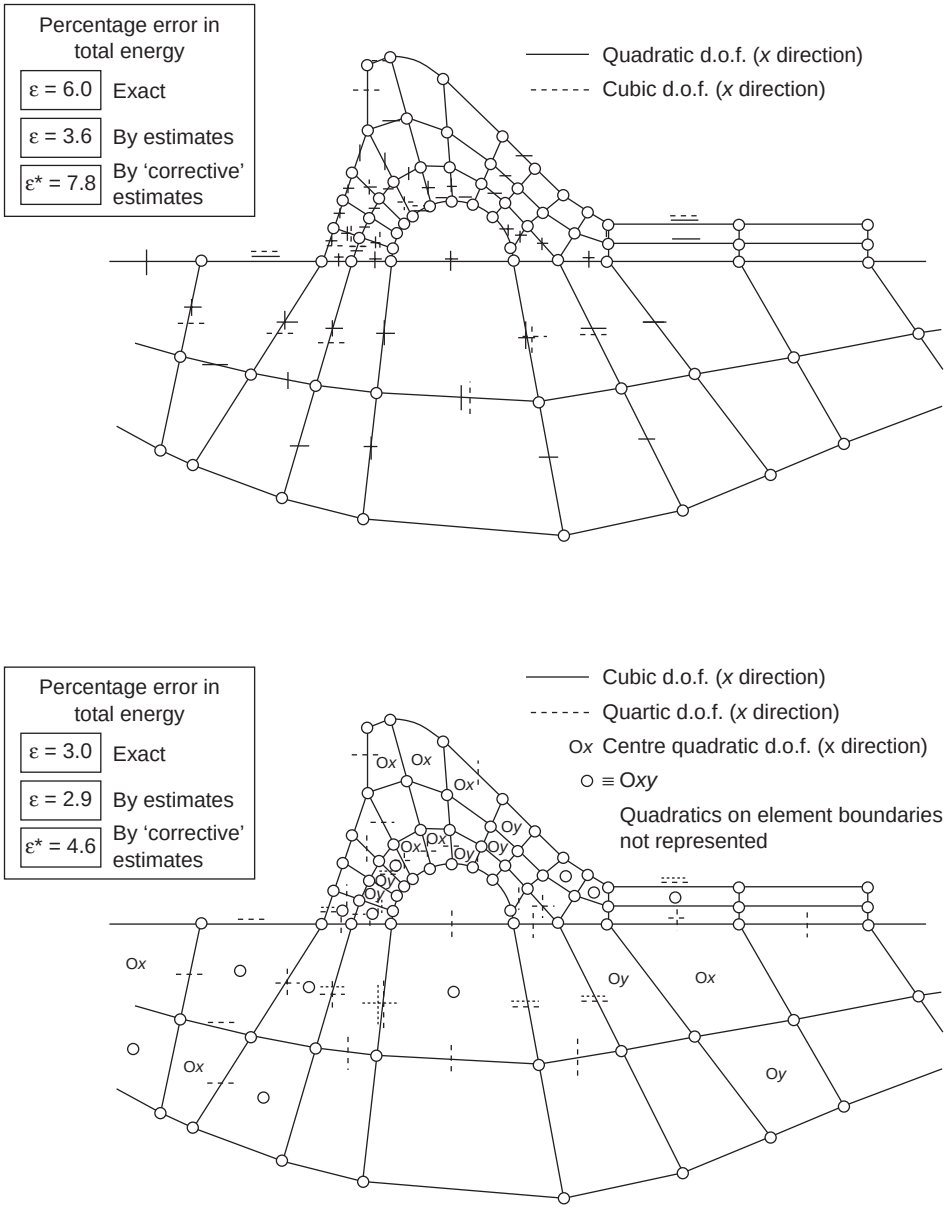
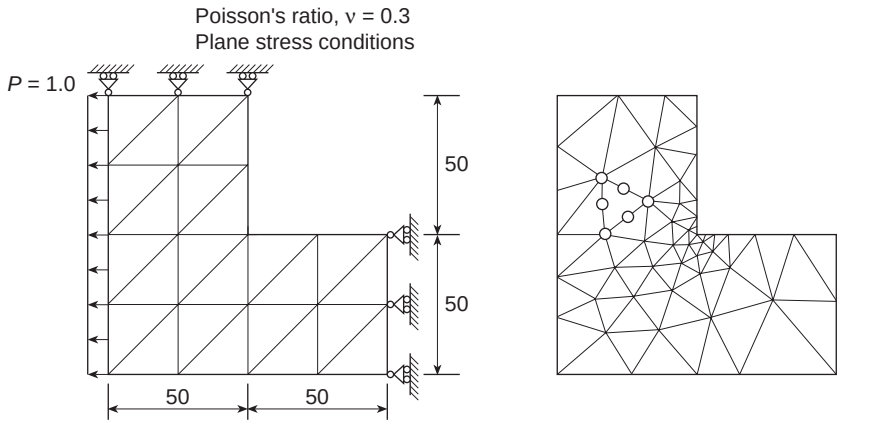


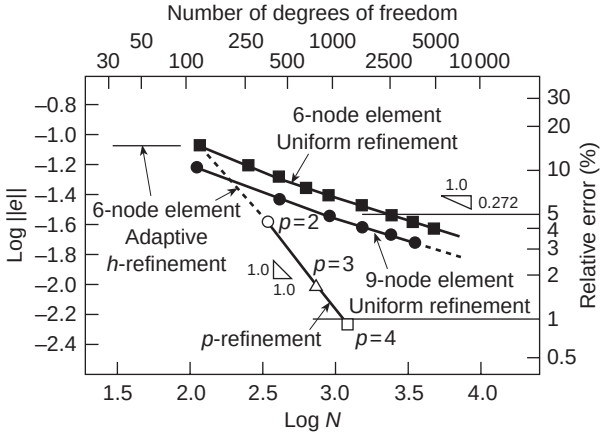
Fig. 15.17 Adaptive solution of perforated dam by *p*-refinement. (a) Stage three, 206 d.o.f.; (b) stage five, 365 d.o.f.

The global energy norm error for the final solution and indeed the error at any stage of the *p*-refinement can be determined using

$$||\mathbf{e}||^2 = ||\mathbf{u}||^2 - ||\hat{\mathbf{u}}_q||^2 \tag{15.14}$$

$q = 1, 2, \ldots, p$.

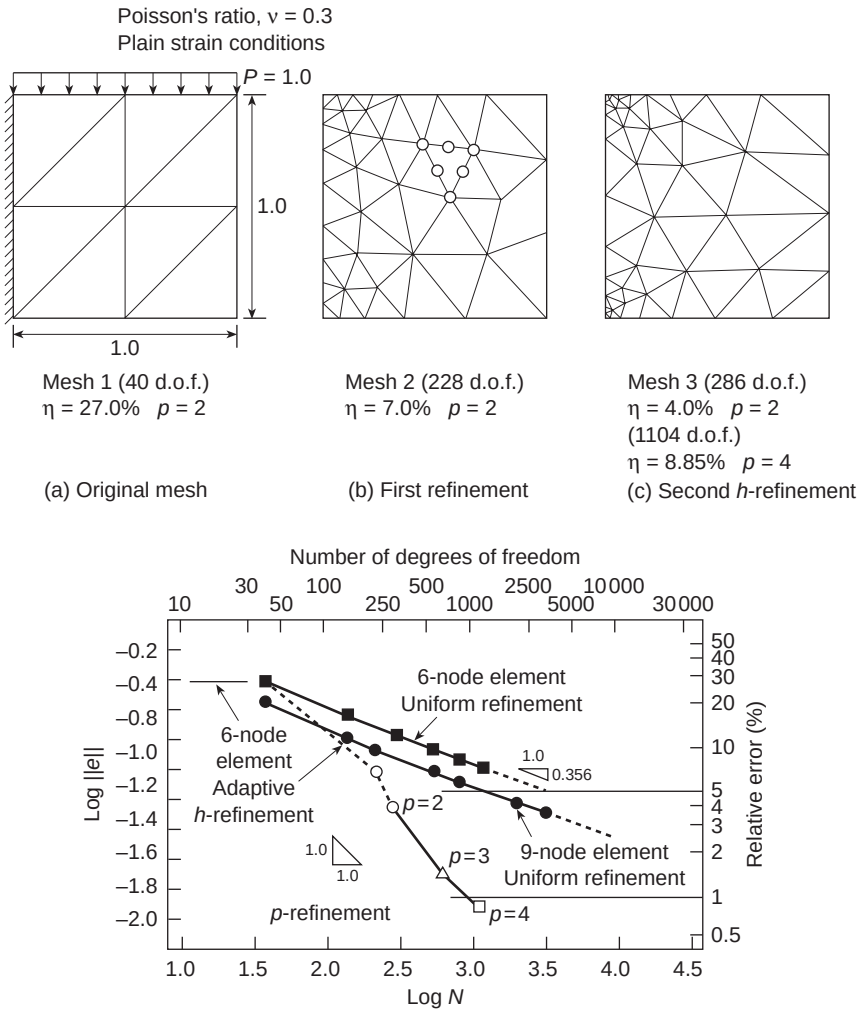




(c) *h-p* refinement. 1% accuracy reached with 1322 d.o.f.

Fig. 15.18 Solution of L-shaped domain by *h-p* refinement (as defined in Example 2 of previous section) using procedure one of reference 19.

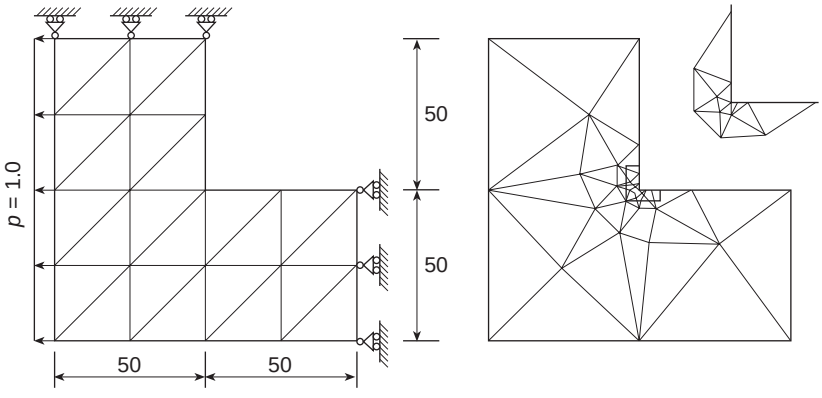
Generally the high accuracy is gained rapidly by refinement, at least from examples performed to date. In Figs 15.18 and 15.19 we show two examples for which we have previously used an *h*-refinement. The first illustrates an *L*-shaped domain with one singularity and the second a short cantilever beam with two strong singularities. In the first both problems are solved using *h*-refinement and target 5% accuracy is reached using quadratic triangles. At this stage the p is increased to third and fourth order so that three solutions are available and when that is reached the error is less than 1%.



(d) *p*-refinement. 1% accuracy reached with 1104 d.o.f.

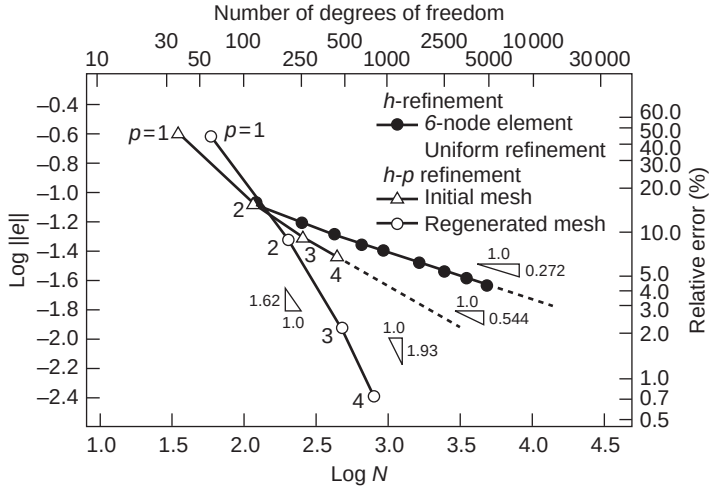
Fig. 15.19 Solution of short cantilever by *h-p* adaptive refinement using procedure one of reference 19.

In the same paper[19] an alternative procedure is suggested. This uses a very coarse mesh at the outset followed by *p*-refinement. In this case the error at the element level is estimated at the last stage of the *p*-refinement as the difference between the last two refinements (e.g., the third and fourth order). The global error estimator is calculated by the extrapolation procedure used in the previous example. The element error estimator is for order $p-1$ rather than the highest order p. It is, however, very accurate. The element error estimator is subsequently used to compute the optimal mesh size as described in Sec. 15.2.2. Nearly optimal rate of convergence is expected to be achieved because the optimal mesh is designed for $p-1$ order



Mesh 1, for $p = 4$ d.o.f. = 430 $\eta = 6.72\%$ Mesh 2, for $p = 4$ d.o.f. = 846 $\eta = 0.76\%$

(a) (b)



Note: 1% accuracy reached with 846 d.o.f.

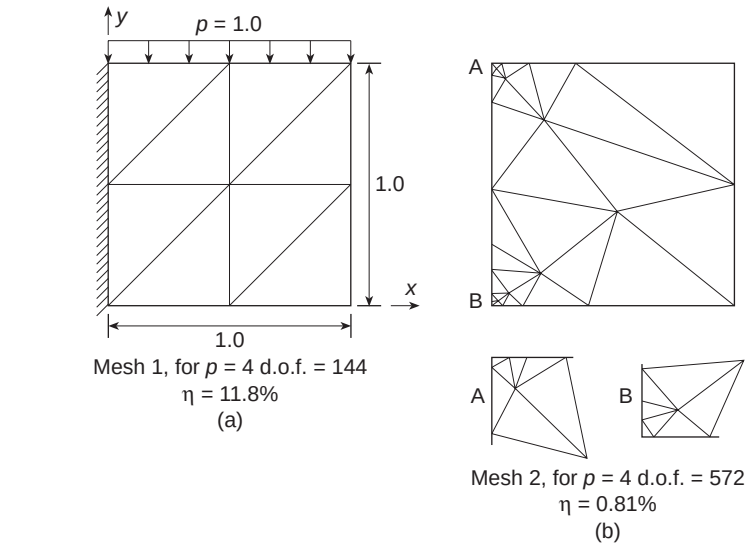
(c)

Fig. 15.20 Solution of L-shaped domain by *h-p* adaptive refinement using alternative procedure of reference 19.

elements. Details of this process will be found again in the reference and will not be discussed further.

At no stage of the *hp*-refinements have we used here any of the estimators quoted in the previous chapter. However, their use would make the optimal mesh design at order p possible, because the element error can be accurately estimated at order p. It will result in an optimal *hp*-refinement.

The two examples we have quoted above are re-analysed using the alternative process described above and presented in Figs 15.20 and 15.21. In both cases



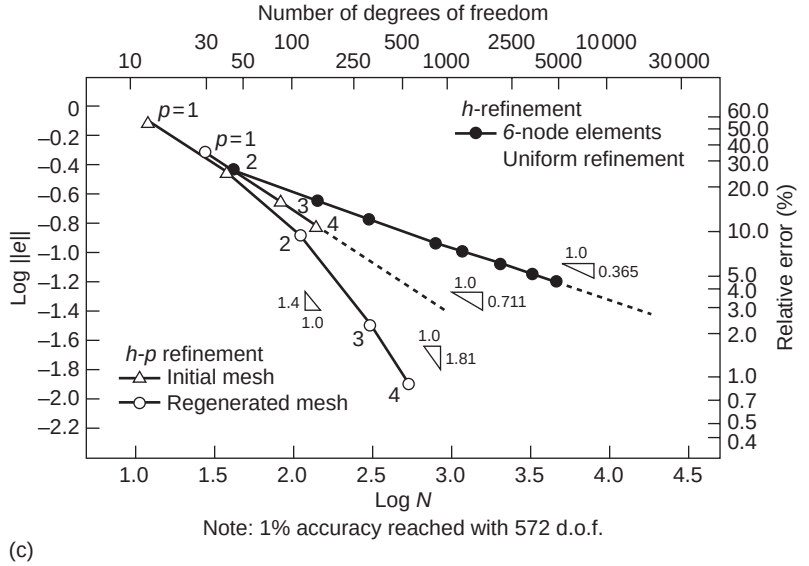
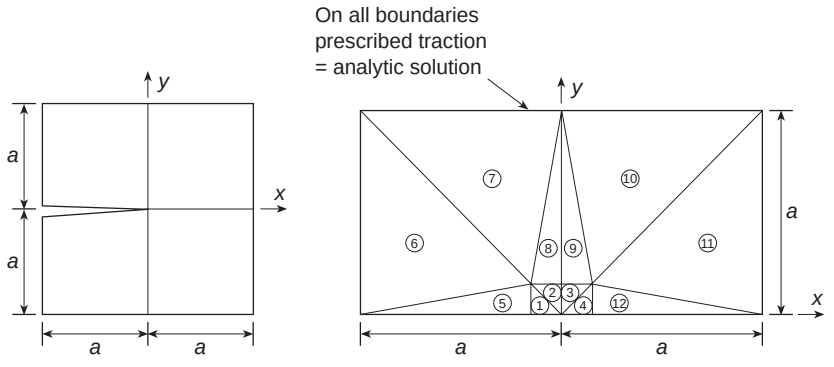
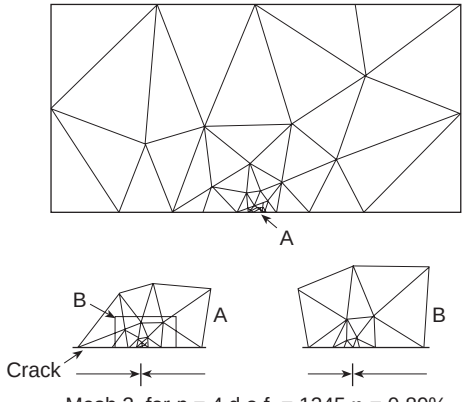


Fig. 15.21 Solution of short cantilever by *h-p* adaptive refinement using alternative procedure of reference 19.

the final accuracy shows an error of less than 1% but it is noteworthy that the total number of degrees of freedom used with the second method is considerably less than that in the first and still achieves a nearly optimal rate of convergence.

We can conclude this section on *hp*-refinement with a final example where a highly singular crack domain is studied. Once again the second procedure is used showing in Fig. 15.22 a remarkable rate of convergence.





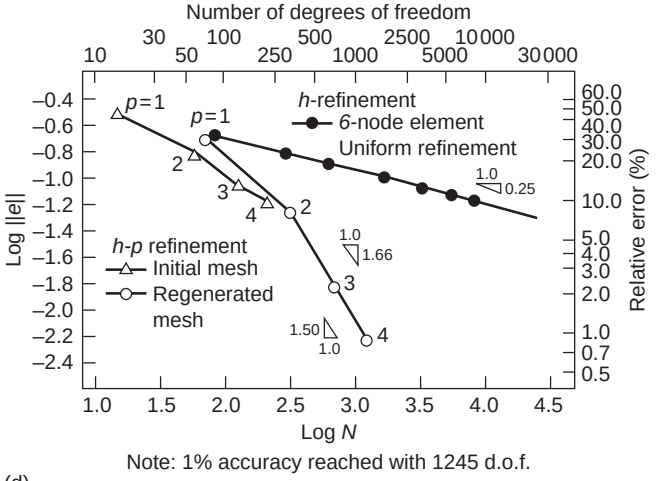
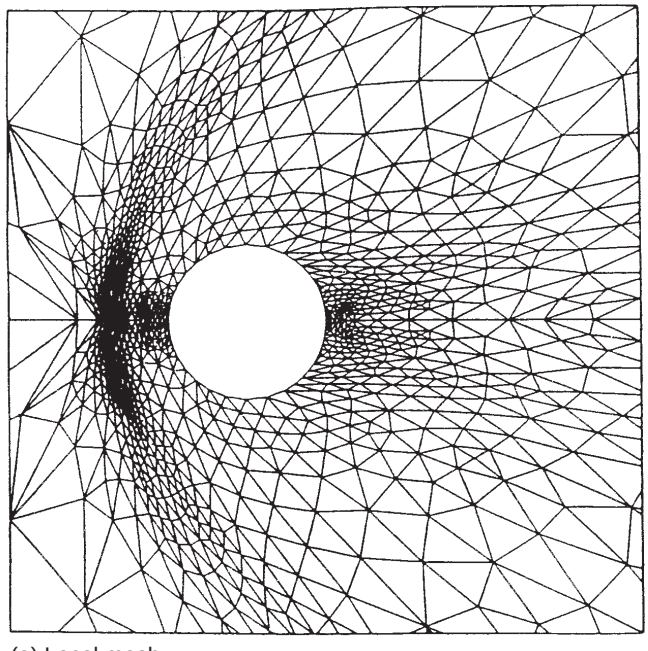
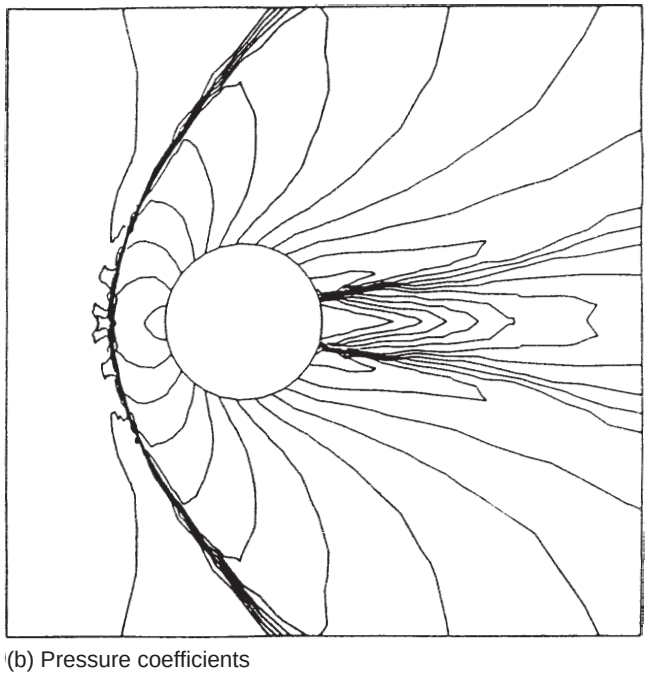


Fig. 15.22 Adaptive *h-p* refinement for a singular crack using alternative procedure of reference 19.



(a) Local mesh



(b) Pressure coefficients

Fig. 15.23 Directional mesh refinement. Gas flow past a circular cylinder – Mach number 3. Third refinement mesh 709 nodes (1348 elements).

15.4 Concluding remarks

The methods of estimating errors and adaptive refinement which are described in this and the previous chapter constitute a very important tool for practical application of finite element methods. The range of applications is large and we have only touched here upon the relatively simple range of linear elasticity and similar self-adjoint problems. A recent survey shows many more areas of application[42] and the reader is referred to this publication for interesting details. At this stage we would like to reiterate that many different norms or measures of error can be used and that for some problems the energy norm is not in fact 'natural'. A good example of this is given by problems of high-speed gas flow, where very steep gradients (shocks) can develop. The formulation of such problems is complex, but this is not necessary for the present argument.

For problems in fluid mechanics which we will discuss in Volume 3 and similarly for problems of strain localization in plastic softening discussed in Volume 2 no global norms can be used effectively. We shall therefore base our refinement on the value of the maximum curvatures developed by the solution of u. On occasion an elongation of the elements will be used to refine the mesh appropriately. Figure 15.23 shows a typical problem of shock capture solved adaptively.

References

1. I. Babuška and C. Rheinboldt. *A-posteriori* error estimates for the finite element method. *Internat. J. Num. Meth. Eng.*, **12**, 1597–615, 1978.
2. I. Babuška and C. Rheinboldt. Adaptive approaches and reliability estimates in finite element analysis. *Comp. Meth. Appl. Mech. Eng.*, **17**/**18**, 519–40, 1979.
3. O.C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Num. Meth. Eng.*, **24**, 337–57, 1987.
4. E. Oñate and G. Bugeda. A study of mesh optimality criteria in adaptive finite element analysis. *Eng. Comp.*, **10**, 307–21, 1993.
5. E.R. de Arantes e Oliveira. Theoretical foundations of the finite element method. *Internat. J. Solids Struct.*, **4**, 929–52, 1968.
6. E.R. de Arantes e Oliveira. Optimization of finite element solutions. In *Proc. 3rd Conf. Matrix Methods in Structural Mechanics*, volume AFFDL-TR-71–160, 423–446, Wright-Patterson Air Force Base, Ohio, 1972.
7. R.L. Taylor and R. Iding. Applications of extended variational principles to finite element analysis. In *Proc. of the International Conference on Variational Methods in Engineering*, volume II, pages 2/54–2/67. Southampton University Press, 1973.
8. W. Gui and I. Babuška. The *h*, *p* and *h-p* version of the finite element method in 1 dimension. Part 1: The error analysis of the p-version. Part 2: The error analysis of the *h*- and *h-p* version. Part 3: The adaptive *h-p* version. *Numerische Math.*, **48**, 557–683, 1986.
9. B. Guo and I. Babuška. The *h-p* version of the finite element method. Part 1: The basic approximation results. Part 2: General results and applications. *Comp. Mech.*, **1**, 21–41, 203–26, 1986.
10. I. Babuška and B. Guo. The *h-p* version of the finite element method for domains with curved boundaries. *SIAM J. Numer. Anal.*, **25**, 837–61, 1988.
11. I. Babuška and B.Q. Guo. Approximation properties of the *hp* version of the finite element method. *Comp. Meth. Appl. Mech. Eng.*, **133**, 319–49, 1996.

12. L. Demkowiez, J.T. Oden, W. Rachowiez, and O. Hardy. Toward a universal *h-p* adaptive finite element strategy. Part 1: Constrained approximation and data structure. *Comp. Meth. Appl. Mech. Eng.*, **77**, 79–112, 1989.
13. W. Rachowicz, T.J. Oden, and L. Demkowicz. Toward a universal *h-p* adaptive finite element strategy. Part 3: Design of *h-p* meshes. *Comp. Meth. Appl. Mech. Eng.*, **77**, 181–211, 1989.
14. K.S. Bey and J.T. Oden. *hp*-version discontinuous Galerkin methods for hyperbolic conservation laws. *Comp. Meth. Appl. Mech. Eng.*, **133**, 259–86, 1996.
15. C.E. Baumann and J.T. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Comp. Meth. Appl. Mech. Eng.*, **175**, 311–41, 1999.
16. P. Monk. On the *p* and *hp* extension of Nedelecs curl-conforming elements. *J. Comput. Appl. Math.*, **53**, 117–37, 1994.
17. L.K. Chilton and M. Suri. On the selection of a locking-free hp element for elasticity problems. *Internat. J. Num. Meth. Eng.*, **40**, 2045–62, 1997.
18. L. Vardapetyan and L. Demkowicz. *hp*-Adaptive finite elements in electromagnetics. *Comp. Meth. Appl. Mech. Eng.*, **169**, 331–44, 1999.
19. O.C. Zienkiewicz, J.Z. Zhu, and N.G. Gong. Effective and practical *h-p*-version adaptive analysis procedures for the finite element method. *Internat. J. Num. Meth. Eng.*, **28**, 879–91, 1989.
20. J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comp. Phys.*, **72**, 449–66, 1987.
21. J.C. Cavendish. Automatic triangulation of arbitrary planar domains for the finite element method. *Internat. J. Num. Meth. Eng.*, **8**, 679–96, 1974.
22. S.H. Lo. A new mesh generation scheme for arbitrary planar domains. *Internat. J. Num. Meth. Eng.*, **21**, 1403–26, 1985.
23. J. Peraire, K. Morgan, M. Vahdati, and O.C. Zienkiewicz. Finite element Euler computations in 3-d. *Internat. J. Num. Meth. Eng.*, **26**, 2135–59, 1988.
24. J.Z. Zhu, O.C. Zienkiewicz, E. Hinton, and J. Wu. A new approach to the development of automatic quadrilateral mesh generation. *Internat. J. Num. Meth. Eng.*, **32**, 849–66, 1991.
25. J.Z. Zhu, E. Hinton, and O.C. Zienkiewicz. Mesh enrichment against mesh regeneration using quadrilateral elements. *Comm. Num. Meth. Eng.*, **9**, 547–54, 1993.
26. E. Rank and O.C. Zienkiewicz. A simple error estimator for the finite element method. *Comp. Meth. Appl. Mech. Eng.*, **3**, 243–50, 1987.
27. E. Rank, M. Schweingruber, and M. Sommer. Adaptive mesh generation. *Comm. Numer. Methods Engrg.*, **9**, 121–29, 1993.
28. T.D. Blacker, M.B. Stephenson, and S. Canann. Analysis automation with paving: A new quadrilateral meshing technique. *Adv. Eng. Software*, Elsevier, **56**(13), 332–37, 1991.
29. T.D. Blacker and M.B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *Internat. J. Num. Meth. Eng.*, **32**, 811–47, 1991.
30. M.A. Price, C.G. Armstrong, and M.A. Sabin. Hexahedral mesh generation by medial axis subdivision, I: solids with convex edges. *Internat. J. Num. Meth. Eng.*, **38**, 3335–59, 1995.
31. T.J. Tautges, T.D. Blacker, and S.A. Mitchell. The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes. *Internat. J. Num. Meth. Eng.*, **39**, 3327–49, 1996.
32. M.A. Price, C.G. Armstrong, and M.A. Sabin. Hexahedral mesh generation by medial axis subdivision, **II**, solids with flat and concave edges. *Internat. J. Num. Meth. Eng.*, **40**, 111–36, 1997.
33. N. Chiba, I. Nishigaki, Y. Yamashita, C. Takizawa, and K. Fujishiro. A flexible automatic hexahedral mesh generation by boundary-fit method. *Comp. Meth. Appl. Mech. Eng.*, **161**, 145–54, 1998.
34. N.P. Weatherill, P.R. Eiseman, J. Hause, and J.F. Thompson. *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*. Pineridge Press, Swansea, 1994.

35. J.F. Thompson, B.K. Soni, and N.P. Weatherill, editors. *Handbook of Grid Generation*. CRC Press, January 1999.
36. P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.
37. J.Z. Zhu and O.C. Zienkiewicz. Adaptive techniques in the finite element method. *Comm. Appl. Num. Math.*, **4**, 197–204, 1988.
38. D.W. Kelly, J.P. De S.R. Gago, O.C. Zienkiewicz, and I. Babuška. *A posteriori* error analysis and adaptive processes in the finite element method: Part I – Error analysis. *Internat. J. Num. Meth. Eng.*, **19**, 1593–619, 1983.
39. J.P. De S.R. Gago, D.W. Kelly, O.C. Zienkiewicz, and I. Babuška. *A posteriori* error analysis and adaptive processes in the finite element method: Part II – Adaptive mesh refinement. *Internat. J. Num. Meth. Eng.*, **19**, 1621–56, 1983.
40. B.A. Szabo. Mesh design for the *p* version of the finite element. *Comp. Meth. Appl. Mech. Eng.*, **55**, 181–97, 1986.
41. I. Babuška, B.A. Szabo, and I.N. Katz. The *p* version of the finite element method. *SIAM J. Numer. Anal.*, **18**, 512–45, 1981.
42. P. Ladevèze and J.T. Oden (Editors), editors. *Advances in Adaptive Computational Methods in Mechanics Studies in Applied Mechanics 47*. Elsevier, 1998.

16

Point-based approximations; element-free Galerkin – and other meshless methods

16.1 Introduction

In all of the preceding chapters, the finite element method was characterized by the subdivision of the total domain of the problem into a set of subdomains called elements. The union of such elements gave the total domain. The subdivision of the domain into such components is of course laborious and difficult necessitating complex mesh generation. Further if adaptivity processes are used, generally large areas of the problem have to be remeshed. For this reason, much attention has been given to devising approximation methods which are based on points without necessity of forming elements.

When we discussed the matter of generalized finite element processes in Chapter 3, we noted that point collocation or in general finite differences did in fact satisfy the requirement of the pointwise definition. However the early finite differences were always based on a regular arrangement of nodes which severely limited their applications. To overcome this difficulty, since the late 1960s the proponents of the finite difference method have worked on establishing the possibility of finite difference calculus being based on an arbitrary disposition of collocation points. Here the work of Girault,[1] Pavlin and Perrone,[2] and Snell *et al.*[3] should be mentioned. However a full realization of the possibilities was finally offered by Liszka and Orkisz,[4,5] and Krok and Orkisz[6] who introduced the use of least square methods to determine the appropriate shape functions.

At this stage Orkisz and coworkers realized not only that collocation methods could be used but also the full finite element, weak formulation could be adopted by performing integration. Questions of course arose as to what areas such integration should be applied. Liszka and Orkisz[4] suggested determining a 'tributary area' to each node providing these nodes were triangulated as shown in Fig. 16.1(a). On the other hand in a somewhat different context Nay and Utku[7] also used the least square approximation including triangular vertices and points of other triangles placed outside a triangular element thus simply returning to the finite element concept. We show this kind of approximation in Fig. 16.1(b). Whichever form of tributary area was used the direct least square approximation centred at each node will lead to discontinuities of the function between the chosen integration areas and

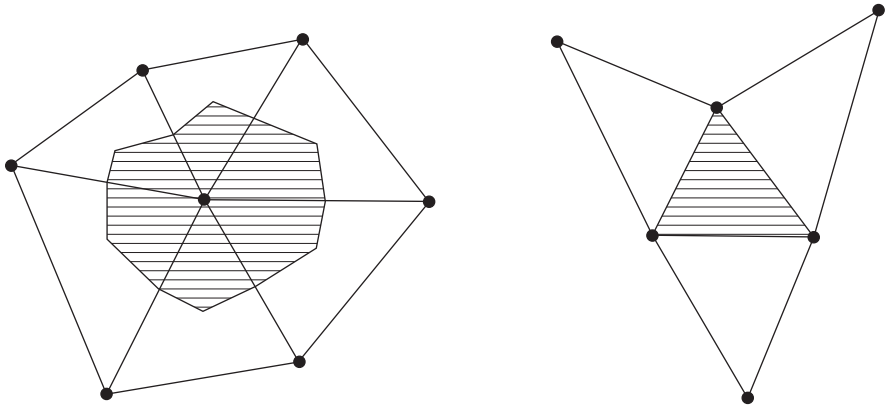


Fig. 16.1 Patches of triangular elements and tributary areas.

thus will violate the rules which we have imposed on the finite element method. However it turns out that such rules could be violated and here the patch test will show that convergence is still preserved.

However the possibility of determining a completely compatible form of approximation existed. This compatible form in which continuity of the function and of its slope if required and even higher derivatives could be accomplished by the use of so-called moving least square methods. Such methods were originated in another context (Shepard,[8] Lancaster and Salkauskas,[9,10]). The use of such interpolation in the meshless approximation was first suggested by Nayroles *et al.*[11–13] This formulation was named by the authors as the *diffuse finite element method.*

Belytschko and coworkers[14,15] quickly realized the advantages offered by such an approach especially when dealing with the development of cracks and other problems for which standard elements presented difficulties. His so-called 'element-free Galerkin' method led to many seminal publications which have been extensively used since.

An alternative use of moving least square procedures was suggested by Duarte and Oden.[16,17] They introduced at the same time a concept of hierarchical forms by noting that all shape functions derived by least squares possess the partition of unity property (viz. Chapter 8). Thus higher order interpolations could be added at each node rather than each element, and the procedures of element-free Galerkin or of the diffuse element method could be extended.

The use of all the above methods still, however, necessitates integration. Now, however, this integration need not be carried out over complex areas. A background grid for integration purposes has to be introduced though internal boundaries were no longer required. Thus such numerical integration on regular grids is currently being used by Belytschko[18,19] and other approaches are being explored. However an interesting possibility was suggested by Babuška and Melenk.[20,21]

Babuška and Melenk use a partition of unity but now the first set of basic shape functions is derived on the simplest element, say the linear triangle. Most of the

approximations then arise through addition of hierarchical variables centred at nodes. We feel that this kind of approach which necessitates very few elements for integration purposes combines well the methodologies of 'element free' and 'standard element' approximation procedures. We shall demonstrate a few examples later on the application of such methods which seem to present a very useful extension of the hierarchical approach.

Incidentally the procedures based on local elements also have the additional advantage that global functions can be introduced in addition to the basic ones to represent special phenomena, for instance the presence of a singularity or waves. Both of these are important and the idea presented by this can be exploited. In Volume 3, we shall show the application of this to certain wave phenomena, see Chapter 8, Volume 3.

This chapter will conclude with reference to other similar procedures which we do not have time to discuss. We shall refer to such procedures in the closure of this chapter.

16.2 Function approximation

We consider here a local set of n points in two (or three) dimensions defined by the coordinates $x_k, y_k, z_k;\ k = 1, 2, \ldots, n$ or simply $\mathbf{x_k} = [x_k, y_k, z_k]$ at which a set of data values of the unknown function $\tilde{u}_k$ are given. It is desired to fit a specified function form to the data points. In order to make a fit it is necessary to:

1. Specify the form of the functions, $\mathbf{p}(\mathbf{x})$, to be used for the approximation. Here as in the standard finite element method, it is essential to include low order polynomials necessary to model the highest derivatives contained in the differential equation or in the weak form approximation being used. Certainly a complete linear and sometimes quadratic polynomial will always be necessary.
2. Define the procedure for establishing the fit.

Here we will consider some *least square fit* methods as the basis for performing the fit. The functions will mostly be assumed to be polynomials, however, in addition other functions can be considered if these are known to model well the solution expected (e.g., see Chapter 8, Volume 3 on use of 'wave' functions).

16.2.1 Least square fit

We shall first consider a least square fit scheme which minimizes the square of the distance between n data values $\tilde{u}_k$ defined at the points $\mathbf{x}_k$ and an approximating function evaluated at the same points $\hat{\mathbf{u}}(\mathbf{x_k})$. We assume the approximation function is given by a set of monomials p_j

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{n} p_j(\mathbf{x})\alpha_j \equiv \mathbf{p}(\mathbf{x})\boldsymbol{\alpha} \tag{16.1}$$

in which $\mathbf{p}$ is a set of *linearly independent* polynomial functions and $\boldsymbol{\alpha}$ is a set of parameters to be determined. A least square scheme is introduced to perform the

fit and this is written as (see Chapter 14 for similar operations): Minimize

$$J = \tfrac{1}{2}\sum_{k=1}^{n}(\hat{u}(\mathbf{x_k}) - \tilde{u}_k)^2 = \min \tag{16.2}$$

where the minimization is to be performed with respect to the values of $\boldsymbol{\alpha}$. Substituting the values of $\hat{u}$ at the points $\mathbf{x}_k$ we obtain

$$\frac{\partial J}{\partial \alpha_j} = \sum_{k=1}^{n} \frac{\partial \hat{u}_k}{\partial \alpha_j} \cdot (\hat{u}(\mathbf{x_k}) - \tilde{u}_k) = 0; \qquad j = 1, 2, \ldots, n \tag{16.3}$$

where

$$\hat{u}_k = \sum_j \mathbf{p}_j(\mathbf{x_k})\alpha_j$$

This set of equations may be written in a compact matrix form as

$$\frac{\partial J}{\partial \boldsymbol{\alpha}} = \sum_{k=1}^{n} \mathbf{p}_k^{\mathrm{T}}(\mathbf{p}_k\boldsymbol{\alpha} - \tilde{u}_k) = \mathbf{0} \tag{16.4}$$

where $\mathbf{p}_k = \mathbf{p}(\mathbf{x}_k)$. We can define the result of the sums as

$$\mathbf{H} = \sum_{k=1}^{n} \mathbf{p}_k^{\mathrm{T}}\mathbf{p}_k = \mathbf{P}^{\mathrm{T}}\mathbf{P} \tag{16.5}$$

$$\mathbf{g} = \sum_{k=1}^{n} \mathbf{p}_k^{\mathrm{T}}\tilde{u}_k = \mathbf{P}^{\mathrm{T}}\tilde{\mathbf{u}} \tag{16.6}$$

in which

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \cdots \\ \mathbf{p}_n \end{bmatrix} \qquad \text{and} \qquad \tilde{\mathbf{u}} = \begin{Bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \cdots \\ \tilde{u}_n \end{Bmatrix}$$

The above process yields the set of linear algebraic equations

$$\mathbf{H}\boldsymbol{\alpha} = \mathbf{g} = \mathbf{P}^{\mathrm{T}}\tilde{\mathbf{u}}$$

which, provided $\mathbf{H}$ is non-singular, has the solution

$$\boldsymbol{\alpha} = \mathbf{H}^{-1}\mathbf{g} = \mathbf{H}^{-1}\mathbf{P}^{\mathrm{T}}\tilde{\mathbf{u}} \tag{16.7}$$

We can now write the approximation for the function as

$$\hat{u} = \mathbf{p}(\mathbf{x})\,\mathbf{H}^{-1}\mathbf{P}^{\mathrm{T}}\tilde{\mathbf{u}} = \mathbf{N}(\mathbf{x})\tilde{\mathbf{u}}$$

where $\mathbf{N}(\mathbf{x})$ are the appropriate shape or basis functions. In general $\mathbf{N_i}(\mathbf{x_i})$ is not unity as it always has been in standard finite element shape functions. However, the partition of unity [viz. Eq. (8.4)] is always preserved provided $\mathbf{p}(\mathbf{x})$ contains a constant.

Example: Fit of a linear polynomial To make the process clear we first consider a dataset, $\tilde{u}_k$, defined at four points, $\mathbf{x_k}$, to which we desire to fit an approximation given by a linear polynomial

$$\hat{u}(\mathbf{x}) = \alpha_1 + x\alpha_2 + y\alpha_3 = \mathbf{p}(\mathbf{x})\boldsymbol{\alpha}$$

If we consider the set of data defined by

$$x_k = [-4.0 \quad -1.0 \quad 0.0 \quad 6.0]$$
$$y_k = [\ 5.0 \quad -5.0 \quad 0.0 \quad 3.0]$$
$$\tilde{u}_k = [-1.5 \quad 5.1 \quad 3.5 \quad 4.3]$$

we can write the arrays as

$$\mathbf{P} = \begin{bmatrix} 1 & -4 & 5 \\ 1 & -1 & -5 \\ 1 & 0 & 0 \\ 1 & 6 & 3 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{u}} = \begin{Bmatrix} -1.5 \\ 5.1 \\ 3.5 \\ 4.3 \end{Bmatrix}$$

Using Eq. (16.5) we obtain the values

$$\mathbf{H} = \mathbf{P}^{\mathrm{T}}\mathbf{P} = \begin{bmatrix} 4 & 1 & 3 \\ 1 & 53 & 3 \\ 3 & 3 & 59 \end{bmatrix} \quad \text{and} \quad \mathbf{g} = \mathbf{P}^{\mathrm{T}}\tilde{\mathbf{u}} = \begin{Bmatrix} 11.4 \\ 26.7 \\ -20.1 \end{Bmatrix}$$

which from Eq. (16.7) has the solution

$$\boldsymbol{\alpha} = \begin{Bmatrix} 3.1241 \\ 0.4745 \\ -0.5237 \end{Bmatrix}$$

Thus, the values for the least square fit at the data points are

$$\hat{\mathbf{u}} = \begin{Bmatrix} -1.5194 \\ 5.0698 \\ 2.9676 \\ 4.2820 \end{Bmatrix}$$

The least square fit for these data points is shown in Fig. 16.2 and the difference between the data points and the values of the fit at $\mathbf{x_k}$ is given in Table 16.1.

16.2.2 Weighted least square fit

Let us now assume that the point at the origin, $\mathbf{x}_0 = \mathbf{0}$, is the point about which we are making the expansion and, therefore, the one where we would like to have the best accuracy. Based on the linear approximation above we observe that the direct least square fit yields at the point in question the *largest* discrepancy. In order to improve the fit we can modify our least square fit for weighting the data in a way that emphasizes the effect of distance from a chosen point. We can write such a *weighted least square fit* as the minimization of

$$J = \tfrac{1}{2}\sum_{k=1}^{n} w(\mathbf{x}_k - \mathbf{x}_0)(\hat{u}(\mathbf{x}_k) - \tilde{u}_k)^2 = \min \tag{16.8}$$

where w is the weighting function. Many choices may be made for the shape of the function w. If we assume that the weight function depends on a radial distance, r,

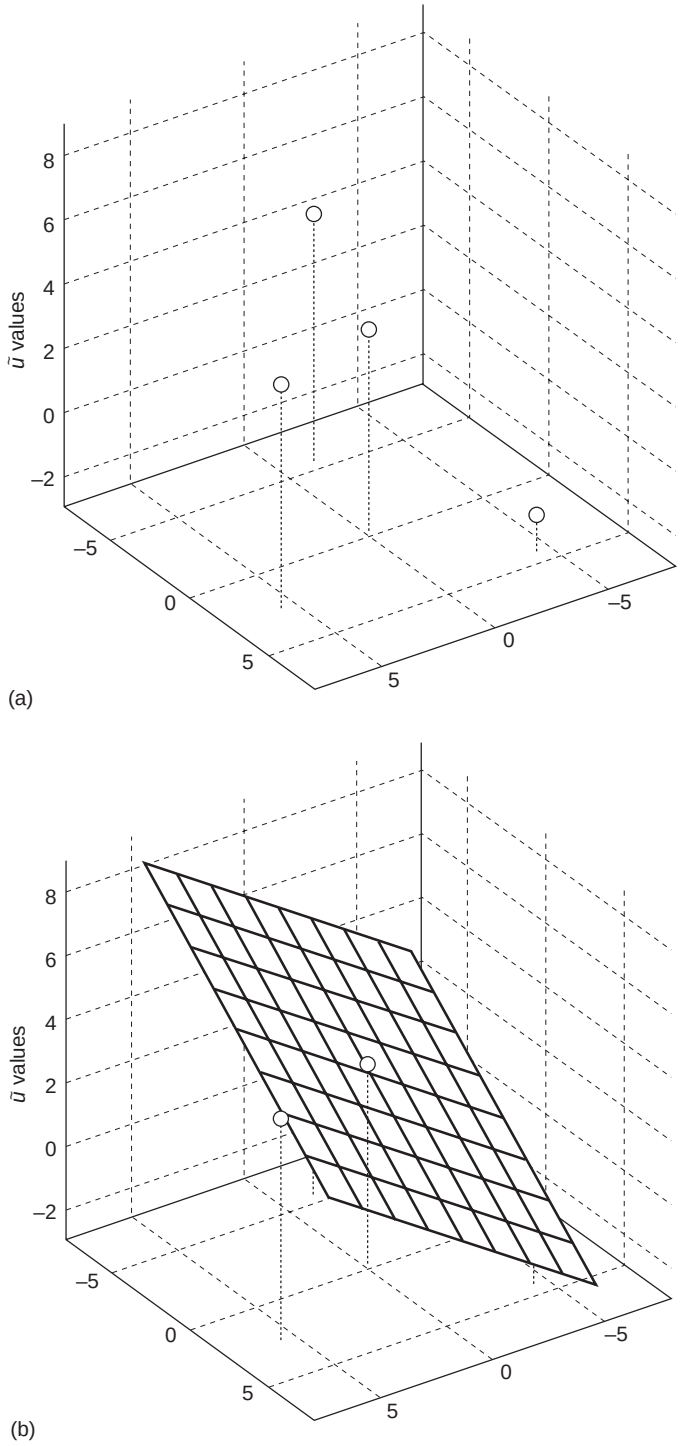


Fig. 16.2 Least square fit: (a) four data points; (b) fit of linear function on the four data points.

Table 16.1 Difference between least square fit and data

x_k	−4	−1	0	6
y_k	5	−5	0	3
$\tilde{u}_k$	−1.500	5.100	3.500	4.300
$\hat{u}_k$	−1.392	5.268	3.124	4.400
Difference	−0.108	−0.168	0.376	−0.100

from the chosen point we have

$$w = w(r); \qquad r^2 = (\mathbf{x} - \mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)$$

One functional form for $w(r)$ is the exponential Gauss function:

$$w(r) = \exp(-cr^2); \qquad c > 0 \quad \text{and} \quad r \geqslant 0 \tag{16.9}$$

For $c = 0.125$ this function has the shape shown in Fig. 16.3 and when used with the previously given four data points yields the linear fit shown in Table 16.2.

16.2.3 Interpolation domains and shape functions

In what follows we shall invariably use the least square procedure to interpolate the unknown function in the vicinity of a particular node i. The first problem is that when approximating to the function it is necessary to include a number of nodes equal at least to the number of parameters of $\boldsymbol{\alpha}$ sought to represent a given polynomial. This number, for instance, in two dimensions is three for linear polynomials and six for quadratic ones. As always the number of nodal points has to be greater than or equal to the bare minimum which is the number of parameters required. We should note in passing that it is always possible to develop a singularity in the equation used for solving $\boldsymbol{\alpha}$, i.e. Eq. (16.7) if the data points lie for instance on a straight line in two or three dimensions. However in general we shall try to avoid such difficulties by reasonable spacing of nodes. The domain of influence can well be defined by making sure that the weighting function is limited in extent so that any point lying beyond a certain distance r_m are weighted by zero and therefore are not taken into account. Commonly used weighting functions are, for instance, in direction r, given by

$$w(r) = \begin{cases} \dfrac{\exp(-cr^2) - \exp(-cr_m^2)}{1 - \exp(-cr_m^2)}; & c > 0 \quad \text{and} \quad 0 \leqslant r \leqslant r_m \\ 0 \qquad\qquad\qquad\qquad\quad ; & r > r_m \end{cases} \tag{16.10}$$

which represents a truncated Gauss function. Another alternative is to use a Hermitian interpolation function as employed for the beam example in Sec. 2.10:

$$w(r) = \begin{cases} 1 - 3\left(\dfrac{r}{r_m}\right)^2 + 2\left(\dfrac{r}{r_m}\right)^3; & 0 \leqslant r \leqslant r_m \\ 0 \qquad\qquad\qquad\qquad\quad ; & r > r_m \end{cases} \tag{16.11}$$

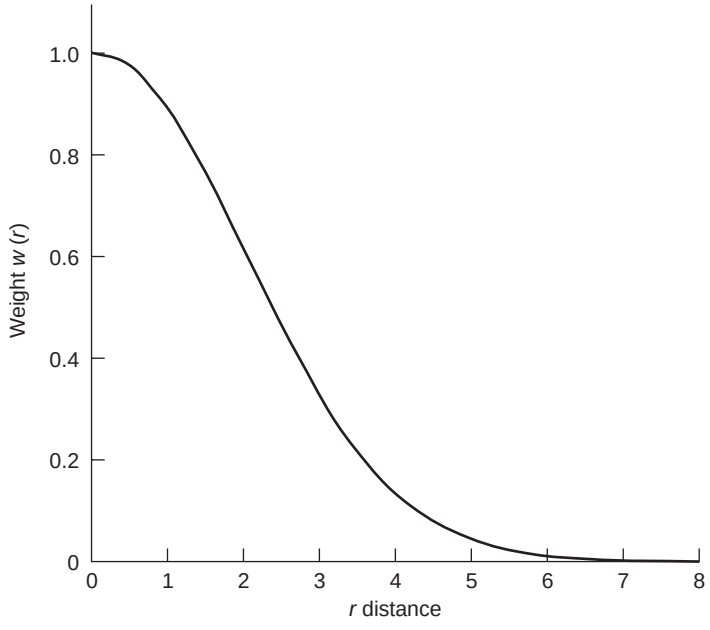


Fig. 16.3 Weighting function for Eq. (16.9): $c = 0.125$.

or alternatively the function

$$w(r) = \begin{cases} \left[1 - \left(\dfrac{r}{r_m}\right)^2\right]^n; & 0 \leqslant r \leqslant r_m \quad \text{and} \quad n \geqslant 2 \\ 0 \qquad\qquad\quad ; & r > r_m \end{cases} \tag{16.12}$$

is simple and has been effectively used. For circular domains, or spherical ones in three dimensions, a simple limitation of r_m suffices as shown in Fig. 16.4(a). However occasionally use of rectangular or hexahedral subdomains is useful as also shown in that figure and now of course the weighting function takes on a different form:

$$w(x,y) = \begin{cases} X_i(x)\,Y_j(y); & 0 \leqslant x \leqslant x_m; \quad 0 \leqslant y \leqslant y_m; \quad \text{and} \quad i,j \geqslant 2 \\ 0 \qquad\qquad ; & x > x_m, y > y_m \end{cases} \tag{16.13}$$

with

$$X_i(x) = \left[1 - \left(\frac{x}{x_m}\right)^2\right]^i; \qquad Y_j(y) = \left[1 - \left(\frac{y}{y_m}\right)^2\right]^j$$

Table 16.2 Difference between weighted least square fit and data

x_k	−4	−1	0	6
y_k	5	−5	0	3
$\tilde{u}_k$	−1.500	5.100	3.500	4.300
$\hat{u}_k$	−0.880	5.247	3.4872	5.246
Error	−0.620	−0.148	0.013	−0.946

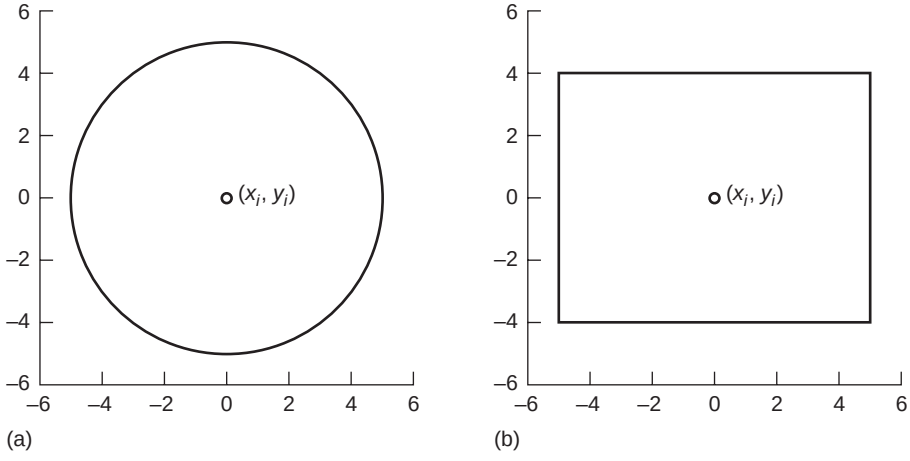


Fig. 16.4 Two-dimensional interpolation domains: (a) circular; (b) rectangular.

The above two possibilities are shown in Fig. 16.4. Extensions to three dimensions using these methods is straightforward.

Clearly the domains defined by the weighting functions will overlap and it is necessary if any of the integral procedures are used such as the Galerkin method to avoid such an overlap by defining the areas of integration. We have suggested a couple of possible ideas in Fig. 16.1 but other limitations are clearly possible. In Fig. 16.5, we show an approximation to a series of points sampled in one dimension. The weighting function here always embraces three or four nodes. Limiting however the domains of their validity to a distance which is close to each of the points provides a unique definition of interpolation. The reader will observe that this interpolation is

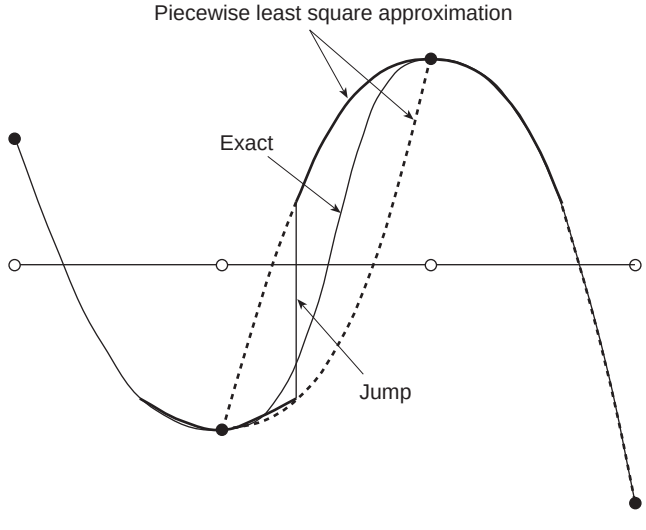


Fig. 16.5 A one-dimensional approximation to a set of data points using parabolic interpolation and direct least square fit to adjacent points.

discontinuous. We have already pointed out such a discontinuity in Chapter 3, but if strictly finite difference approximations are used this does not matter. It can however have serious consequences if integral procedures are used and for this reason it is convenient to introduce a modification to the definition of weighting and method of calculation of the shape function which is given in the next section.

16.3 Moving least square approximations – restoration of continuity of approximation

The method of moving least squares was introduced in the late 1960s by Shepard[8] as a means of generating a smooth surface interpolating between various specified point values. The procedure was later extended for the same reasons by Lancaster and Salkauskas[9,10] to deal with very general surface generation problems but again it was not at that time considered of importance in finite elements. Clearly in the present context the method of moving least squares could be used to replace the local least squares we have so far considered and make the approximation fully continuous.

In moving least square methods, the weighted least square approximation is applied in exactly the same manner as we have discussed in the preceding section but is established for every point at which the interpolation is to be evaluated. The result of course completely smooths the weighting functions used and it also presents smooth derivatives noting of course that such derivatives will depend on the locally specified polynomial.

To describe the method, we again consider the problem of fitting an approximation to a set of data items $\tilde{u}_i$, $i = 1, \ldots, n$ defined at the n points $\mathbf{x}_i$. We again assume the approximating function is described by the relation

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})\alpha_j = \mathbf{p}(\mathbf{x})\boldsymbol{\alpha} \tag{16.14}$$

where p_j are a set of linearly independent (polynomial) functions and α_i are unknown quantities to be determined by the fit algorithm. A generalization to the weighted least square fit given by Eq. (16.8) may be defined for each point $\mathbf{x}$ in the domain by solving the problem

$$J(\mathbf{x}) = \tfrac{1}{2}\sum_{k=1}^{n} w_x(\mathbf{x}_k - \mathbf{x})[\tilde{u}_k - \mathbf{p}(\mathbf{x}_k)\boldsymbol{\alpha}]^2 = \min \tag{16.15}$$

In this form the weighting function is defined for *every* point in the domain and thus can be considered as translating or *moving* as shown in Fig. 16.6. This produces a *continuous* interpolation throughout the whole domain.

Figure 16.7 illustrates the problem previously presented in Fig. 16.5 now showing continuous interpolation. We should note that it is now no longer necessary to specify 'domains of influence' as the shape functions are defined in the whole domain.

The main difficulty with this form is the generation of a *moving* weight function which can change size continuously to match any given distribution of points $\mathbf{x}_k$ with a limited number of points entering each calculation. One expedient method

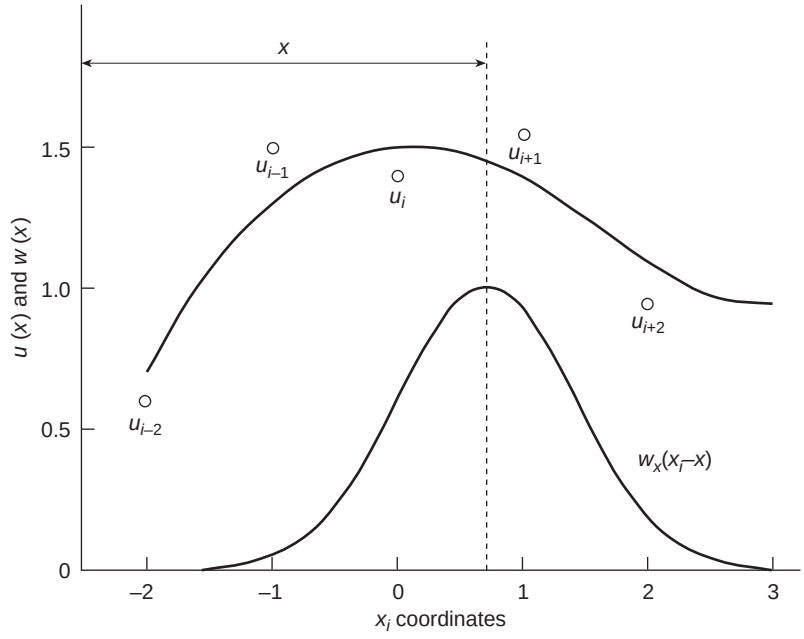


Fig. 16.6 Moving weighting function approximation in MLS.

to accomplish this is to assume the function is *symmetric* so that

$$w_x(\mathbf{x}_k - \mathbf{x}) = w_x(\mathbf{x} - \mathbf{x}_k)$$

and use a weighting function associated with each data point $\mathbf{x}_k$ as

$$w_x(\mathbf{x}_k - \mathbf{x}) = w_k(\mathbf{x} - \mathbf{x}_k)$$

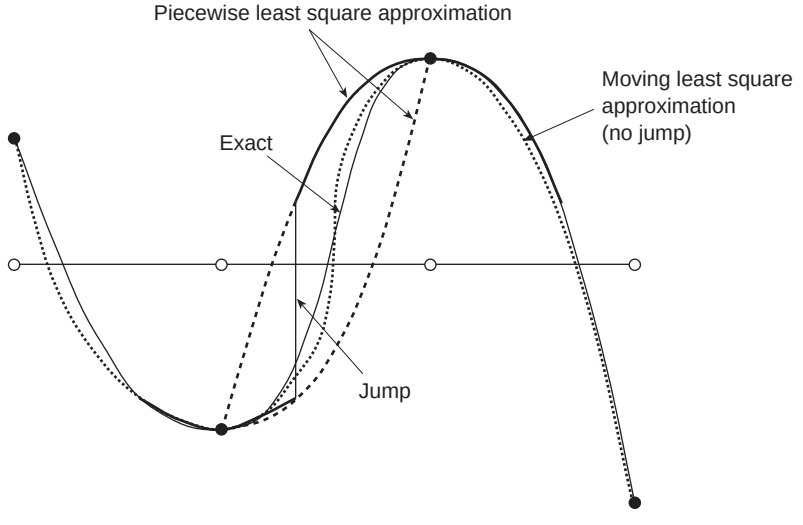


Fig. 16.7 The problem of Fig. 16.5 with moving least square interpolation.

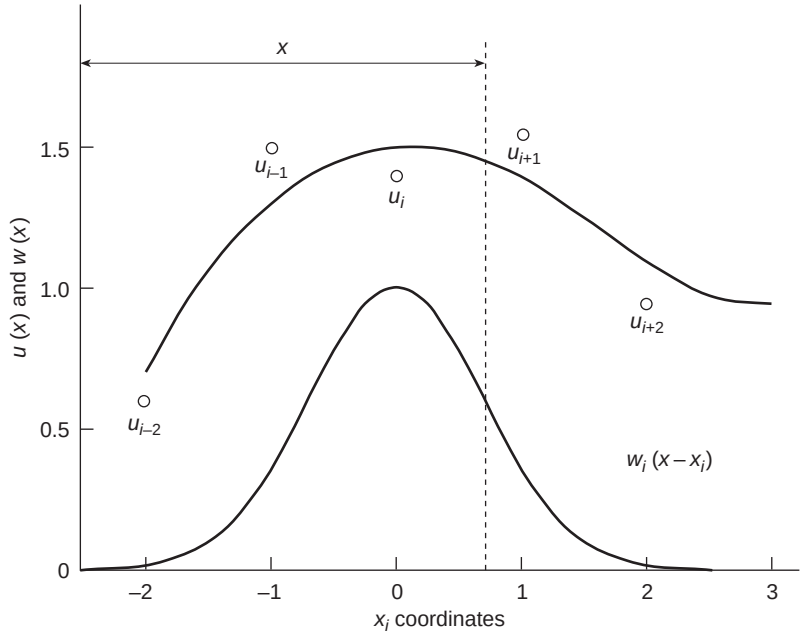


Fig. 16.8 A 'fixed' weighting function approximation to the MLS method.

The function to be minimized now becomes

$$J(\mathbf{x}) = \tfrac{1}{2}\sum_{k=1}^{n} w_k(\mathbf{x}-\mathbf{x}_k)[\tilde{u}_k - \mathbf{p}(\mathbf{x}_k)\boldsymbol{\alpha}]^2 = \min \tag{16.16}$$

In this form the weighting function is *fixed* at a data point $\mathbf{x}_k$ and evaluated at the point $\mathbf{x}$ as shown in Fig. 16.8. Each weighting function may be defined such that

$$w_x(r) = \begin{cases} f_k(r), & \text{if} \quad |r| < r_k \\ 0, & \text{otherwise} \end{cases} \tag{16.17}$$

and the terms in the sum are zero whenever $r^2 = (\mathbf{x}-\mathbf{x}_k)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_k)$ and $|r| > r_k$. The parameter r_k defines the radius of a ball around each point, $\mathbf{x}_k$; inside the ball the weighting function is non-zero while outside the radius it is zero. Each point may have a different weighting function and/or radius of the ball around its defining point. The weighting function should be defined such that it is zero on the boundary of the ball. This class of function may be denoted as $C_q^0(r_k)$, where the superscript denotes the boundary value and the subscript the highest derivative for which C_0 continuity is achieved. Other options for defining the weighting function are available as discussed in the previous section. The solution to the least square problem now leads to

$$\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{H}^{-1}(\mathbf{x})\sum_{j=1}^{n}\mathbf{g}_j(\mathbf{x})\tilde{u}_j = \mathbf{H}^{-1}(\mathbf{x})\mathbf{g}(\mathbf{x})\tilde{\mathbf{u}}_j \tag{16.18}$$

where

$$\mathbf{H}(\mathbf{x}) = \sum_{k=1}^{n} w_k(\mathbf{x} - \mathbf{x}_k)\mathbf{p}(\mathbf{x}_k)^{\mathrm{T}}\mathbf{p}(\mathbf{x}_k) \tag{16.19}$$

and

$$\mathbf{g}_j(\mathbf{x}) = w_j(\mathbf{x} - \mathbf{x}_j)\mathbf{p}(\mathbf{x}_j)^{\mathrm{T}} \tag{16.20}$$

In matrix form the arrays $\mathbf{H}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ may be written as

$$\begin{aligned} \mathbf{H}(\mathbf{x}) &= \mathbf{P}^{\mathrm{T}}\mathbf{w}(\Delta\mathbf{x})\mathbf{P} \\ \mathbf{g}(\mathbf{x}) &= \mathbf{w}(\Delta\mathbf{x})\mathbf{P} \end{aligned} \tag{16.21}$$

in which

$$\mathbf{w}(\Delta\mathbf{x}) = \begin{bmatrix} w_1(\mathbf{x} - \mathbf{x}_1) & 0 & \cdots & \cdots \\ 0 & w_2(\mathbf{x} - \mathbf{x}_2) & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & 0 & w_n(\mathbf{x} - \mathbf{x}_n) \end{bmatrix} \tag{16.22}$$

The moving least square algorithm produces solutions for $\boldsymbol{\alpha}$ which depend continuously on the point selected for each fit. The approximation for the function $u(\mathbf{x})$ now may be written as

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{n} N_j(\mathbf{x})\tilde{u}_j \tag{16.23}$$

where

$$N_j(\mathbf{x}) = \mathbf{p}(\mathbf{x})\mathbf{H}^{-1}(\mathbf{x})\mathbf{g}_j(\mathbf{x}) \tag{16.24}$$

define interpolation functions for each data item $\tilde{u}_j$. We note that in general these 'shape functions' do not possess the Kronecker delta property which we noted previously for finite element methods – that is

$$N_j(\mathbf{x}_i) \neq \delta_{ji} \tag{16.25}$$

It must be emphasized that all least square approximations generally have values at the defining points $\mathbf{x}_j$ in which

$$\tilde{u}_j \neq \hat{u}(\mathbf{x}_j) \tag{16.26}$$

i.e., the local values of the approximating function do not fit the nodal unknown values (e.g., Fig. 16.2). Indeed $\hat{u}$ will be the approximation used in seeking solutions to differential equations and boundary conditions and $\tilde{u}_j$ are simply the unknown parameters defining this approximation.

The main drawback of the least square approach is that the approximation rapidly deteriorates if the number of points used, n, largely exceeds that of the m polynomial terms in $\mathbf{p}$. This is reasonable since a least square fit usually does not match the data points exactly.

A moving least square interpolation as defined by Eq. (16.23) can approximate *globally* all the functions used to define $\mathbf{p}(\mathbf{x})$. To show this we consider the set of

approximations

$$\mathbf{U} = \sum_{j=1}^{n} N_j(x)\tilde{\mathbf{U}}_j \tag{16.27}$$

where

$$\mathbf{U} = [\,\hat{u}_1(x) \quad \hat{u}_2(x) \quad \ldots \quad \hat{u}_n(x)\,]^{\mathrm{T}} \tag{16.28}$$

and

$$\tilde{\mathbf{U}}_j = [\,\tilde{u}_{j1} \quad \tilde{u}_{j2} \quad \ldots \quad \tilde{u}_{jn}\,]^{\mathrm{T}} \tag{16.29}$$

Next, assign to each $\tilde{u}_{jk}$ the value of the polynomial $p_k(\mathbf{x}_j)$ (i.e., the kth entry in $\mathbf{p}$) so that

$$\mathbf{U}_j = \mathbf{p}(x_j) \tag{16.30}$$

Using the definition of the interpolation functions given by Eqs (16.23) and (16.24) we have

$$\mathbf{U} = \sum_{j=1}^{n} N_j(\mathbf{x})\mathbf{p}(\mathbf{x}_j) = \sum_{j=1}^{n} \mathbf{p}(\mathbf{x})\mathbf{H}^{-1}(\mathbf{x})\mathbf{g}_j(\mathbf{x})\mathbf{p}(\mathbf{x}_j) \tag{16.31}$$

which after substitution of the definition of $\mathbf{g}_j(x)$ yields

$$\begin{aligned}\mathbf{U} &= \sum_{j=1}^{n} \mathbf{p}(\mathbf{x})\mathbf{H}^{-1}(\mathbf{x})w_j(\mathbf{x}-\mathbf{x}_j)\mathbf{p}(\mathbf{x}_j)^{\mathrm{T}}\mathbf{p}(\mathbf{x}_j)\\ &= \mathbf{p}(\mathbf{x})\mathbf{H}^{-1}\sum_{j=1}^{n} w_j(\mathbf{x}-\mathbf{x}_j)\mathbf{p}(\mathbf{x}_j)^{\mathrm{T}}\mathbf{p}(\mathbf{x}_j)\\ &= \mathbf{p}(\mathbf{x})\mathbf{H}^{-1}\mathbf{H}(\mathbf{x}) = \mathbf{p}(\mathbf{x})\end{aligned} \tag{16.32}$$

Equation (16.32) shows that a moving least square form can exactly interpolate *any function* included as part of the definition of $\mathbf{p}(\mathbf{x})$. If polynomials are used to define the functions, the interpolation always includes exact representations for each included polynomial. Inclusion of the zero-order polynomial (i.e., 1), implies that

$$\sum_{j=1}^{n} N_j(\mathbf{x}) = 1 \tag{16.33}$$

This is called a *partition of unity* (provided it is true for all points, $\mathbf{x}$, in the domain).[22] It is easy to recognize that this is the same requirement as applies to standard finite element shape functions.

Derivatives of moving least square interpolation functions may be constructed from the representation

$$N_j(\mathbf{x}) = \mathbf{p}(\mathbf{x})\mathbf{v}_j(\mathbf{x}) \tag{16.34}$$

where

$$\mathbf{H}(\mathbf{x})\mathbf{v}_j(\mathbf{x}) = \mathbf{g}_j(\mathbf{x}) \tag{16.35}$$

For example, the first derivatives with respect to x is given by

$$\frac{\partial N_j}{\partial x} = \frac{\partial \mathbf{p}}{\partial x}\mathbf{v}_j + \mathbf{p}\frac{\partial \mathbf{v}_j}{\partial x} \tag{16.36}$$

and

$$\mathbf{H}\frac{\partial \mathbf{v}_j}{\partial x} + \frac{\partial \mathbf{H}}{\partial x}\mathbf{v}_j = \frac{\partial \mathbf{g}_j}{\partial x} \tag{16.37}$$

where

$$\frac{\partial \mathbf{H}}{\partial x} = \sum_{k=1}^{n} \frac{\partial w_k(\mathbf{x}-\mathbf{x}_k)}{\partial x}\mathbf{p}(\mathbf{x}_k)^{\mathrm{T}}\mathbf{p}(\mathbf{x}_k) \tag{16.38}$$

and

$$\frac{\partial \mathbf{g}_j}{\partial x} = \frac{\partial w_j(\mathbf{x}-\mathbf{x}_j)}{\partial x}\mathbf{p}(\mathbf{x}_j) \tag{16.39}$$

Higher derivatives may be computed by repeating the above process to define the higher derivatives of $\mathbf{v}_j$. An important finding from higher derivatives is the order at which the interpolation becomes discontinuous between the interpolation subdomains. This will be controlled by the continuity of the weight function only. For weight functions which are C_q^0 continuous in each subdomain the interpolation will be continuous for all derivatives up to order q. For the truncated Gauss function given by Eq. (16.10) only the approximated function will be continuous in the domain, no matter how high the order used for the $\mathbf{p}$ basis functions. On the other hand, use of the Hermitian interpolation given by Eq. (16.11) produces C_1 continuous interpolation and use of Eq. (16.12) produces C_n continuous interpolation. This generality can be utilized to construct approximations for high order differential equations.

Nayroles *et al*. suggest that approximations ignoring the derivatives of $\boldsymbol{\alpha}$ may be used to define the derivatives of the interpolation functions.[11–13] While this approximation simplifies the construction of derivatives as it is no longer necessary to compute the derivatives for $\mathbf{H}$ and $\mathbf{g}_j$, there is little additional effort required to compute the derivatives of the weighting function. Furthermore, for a constant in $\mathbf{p}$ no derivatives are available. Consequently, there is little to recommend the use of this approximation.

16.4 Hierarchical enhancement of moving least square expansions

The moving least square approximation of the function $u(\mathbf{x})$ was given in the previous section as

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{n} N_j(\mathbf{x})\tilde{u}_j \tag{16.40}$$

where $N_j(\mathbf{x})$ defined the interpolation or shape functions based on linearly independent functions prescribed by $\mathbf{p}(\mathbf{x})$ as given by Eq. (16.24). Here we shall restrict

attention to one-dimensional forms and employ polynomial functions to describe $\mathbf{p}(x)$ up to degree k. Accordingly, we have

$$\mathbf{p}(x) = \begin{bmatrix} 1 & x & x^2 & \dots & x^k \end{bmatrix} \tag{16.41}$$

For this case we will denote the resulting interpolation functions using the notation $N_j^k(x)$, where j is associated with the location of the point where the parameter $\tilde{u}_j$ is given and k denotes the order of the polynomial approximating functions. Duarte and Oden suggest using Legendre polynomials instead of the form given above;[16] however, conceptually the two are equivalent and we use the above form for simplicity. A hierarchical construction based on $N_j^k(x)$ can be established which increases the order of the complete polynomial to degree p. The hierarchical interpolation is written as

$$\begin{aligned} \hat{u}(x) &= \sum_{j=1}^{n} \left(N_j^k(x)\tilde{u}_j + N_j^k(x) \begin{bmatrix} x^{k+1} & x^{k+2} & \dots & x^p \end{bmatrix} \begin{Bmatrix} \tilde{b}_{j1} \\ \tilde{b}_{j2} \\ \vdots \\ \tilde{b}_{jq} \end{Bmatrix} \right) \\ &= \sum_{j=1}^{n} N_j^k(x)\left(\tilde{u}_j + \mathbf{q}(x)\tilde{\mathbf{b}}_j\right) = \sum_{j=1}^{n} N_j^k(x) \begin{bmatrix} 1 & \mathbf{q}(x) \end{bmatrix} \begin{Bmatrix} \tilde{u}_j \\ \tilde{\mathbf{b}}_j \end{Bmatrix} \end{aligned} \tag{16.42}$$

where $q = p - k$ and $\tilde{b}_{jm}$, $m = 1, \dots, q$, are additional parameters for the approximation. Derivatives of the interpolation function may be constructed using the method described by Eqs (16.34)–(16.39).

The advantage of the above method lies in the reduced cost of computing the interpolation function $N_j^k(x)$ compared to that required to compute the p-order interpolations $N_j^p(x)$.

Shepard interpolation

For example, use of the functions $N_j^0(x)$, which are called Shepard interpolations,[8] leads to a scalar matrix $\mathbf{H}$ which is trivial to invert to define the N_j^0. Specifically, the Shepard interpolations are

$$N_j^0(x) = H^{-1}(x)g_j(x) \tag{16.43}$$

where

$$H(x) = \sum_{k=1}^{n} w_k(x - x_k) \tag{16.44}$$

and

$$g_j(x) = w_j(x - x_j) \tag{16.45}$$

The fact that the hierarchical interpolations include polynomials up to order p is easy to demonstrate. Based on previous results from standard moving least squares the interpolation with $\tilde{\mathbf{b}}_j = \mathbf{0}$ contains all the polynomials up to degree k. Higher

degree polynomials may be constructed from

$$\hat{u}(x) = \sum_{j=1}^{n} \left(N_j^k(x)\tilde{u}_j + N_j^k(x) \begin{bmatrix} x^{k+1} & x^{k+2} & \dots & x^p \end{bmatrix} \begin{Bmatrix} \tilde{b}_{j1} \\ \tilde{b}_{j2} \\ \vdots \\ \tilde{b}_{jq} \end{Bmatrix} \right) \tag{16.46}$$

by setting all $\tilde{u}_j$ to zero and for each interpolation term setting one of the $\tilde{b}_{jk}$ to unity with the remaining values set to zero. For example, setting $\tilde{b}_{j1}$ to unity results in the expansion

$$\hat{u}(x) = \sum_{j=1}^{n} N_j^k(x) x^{k+1} = x^{k+1} \tag{16.47}$$

This result requires only the partition of unity property

$$\sum_{j=1}^{n} N_j^k(x) = 1 \tag{16.48}$$

The remaining polynomials are obtained by setting the other values of $\tilde{b}_{jk}$ to unity one at a time. We note further that the same order approximation is obtained using $k = 0, 1$ or p.[16]

The above hierarchical form has parameters which do not relate to approximate values of the interpolation function. For the case where $k = 0$ (i.e., Shepard interpolation), Babuška and Melenk[23] suggest an alternate expression be used in which $\mathbf{q}$ in Eq. (16.42) is taken as $\begin{bmatrix} 1 & x & x^2 & \dots & x^p \end{bmatrix}$ and the interpolation written as

$$\hat{u}(x) = \sum_{j=1}^{n} N_j^0(x) \left(\sum_{k=0}^{p} l_k^p(x)\tilde{u}_{jk} \right) \tag{16.49}$$

In this form the $l_k^p(x)$ are Lagrange interpolation polynomials (e.g., see Sec. 8.5) and $\tilde{u}_{jk}$ are parameters with dimensions of u for the jth term at point x_k of the Lagrange interpolation. The above result follows since Lagrange interpolation polynomials have the property

$$l_k(x_i) = \delta_{ki} = \begin{cases} 1, & \text{if} \quad k = i; \\ 0, & \text{otherwise} \end{cases} \tag{16.50}$$

We should also note that options other than polynomials may be used for the $\mathbf{q}(x)$. Thus, for any function $q_i(x)$ we can set the associated $\tilde{b}_{ji}$ to unity (with all others and $\tilde{u}_j$ set to zero) and obtain

$$\hat{u}(x) = \sum_{j=1}^{n} N_j^k(x) q_i(x) = q_i(x) \tag{16.51}$$

Again the only requirement is that

$$\sum_{j=1}^{n} N_j^k(x) = 1 \tag{16.52}$$

Thus, for any basic functions satisfying the partition of unity a hierarchical enrichment may be added using any type of functions. For example, if one knows the structure of the solution involves exponential functions in x it is possible to include them as members of the $\mathbf{q}(x)$ functions and thus capture the essential part of the solution with just a few terms. This is especially important for problems which involve solutions with different length scales. A large length scale can be included in the basic functions, $N_j^k(x)$, while other smaller length scales may be included in the functions $\mathbf{q}(x)$. This will be illustrated further in Volume 3 in the chapter dealing with waves.

The above discussion has been limited to functions in one space variable, however, extensions to two and three dimensions can be easily constructed. In the process of this extension we shall encounter some difficulties which we address in more detail in the section on partition-of-unity finite element methods. Before doing this we explore in the next section the direct use of least square methods to solve differential equations using collocation methods.

16.5 Point collocation – finite point methods

Finite difference methods based on Taylor formula expansions on regular grids can, as explained in Chapter 3, Sec. 3.13, always be considered as *point collocation methods* applied to the differential equation. They have been used to solve partial differential equations for many decades.[24–26] Classical finite difference methods commonly restrict applications to regular grids. This limits their use in obtaining accurate solutions to general engineering problems which have curved (irregular) boundaries and/or multiple material interfaces. To overcome the boundary approximation and interface problem curvilinear mapping may be used to define the finite difference operators.[27]

The extension of the finite difference methods from regular grids to general arbitrary and irregular grids or sets of point has received considerable attention (Girault,[1] Pavlin and Perrone,[2] Snell *et al.*[3]). An excellent summary of the current state of the art may be found in a recent paper by Orkisz[27] who himself has contributed very much to the subject since the late 1970s (Liszka and Orkisz[4]).

More recently such finite difference approximations on irregular grids have been proposed by Batina[28] in the context of aerodynamics and by Oñate *et al.*[29–31] who introduced the name 'finite point method'. Here both elasticity and fluid mechanics problems have been addressed.

In point collocation methods the set of differential equations, which here is taken in the form described in Sec. 3.1, is used directly without the need to construct a weak form or perform domain integrals. Accordingly, we consider

$$\mathbf{A}(\mathbf{u}) = \mathbf{0} \tag{16.53}$$

as a set of governing differential equations in a domain Ω subject to boundary conditions

$$\mathbf{B}(\mathbf{u}) = \mathbf{0} \tag{16.54}$$

applied on the boundaries Γ. An approximation to the dependent variable $\mathbf{u}$ may be constructed using either a weighted or moving least square approximation since at each collocation point the methods become identical. In this we must first describe

the (collocation) *points* and the *weighting function*. The approximation is then constructed from Eq. (16.23) by assuming a sufficient order polynomial for $\mathbf{p}$ in Eq. (16.14) such that *all* derivatives appearing in Eqs (16.53) and (16.54) may be computed. Generally, it is advantageous to use the same order of interpolation to approximate both the differential and boundary conditions.[27] The resulting discrete form for the differential equations at each collocation point becomes

$$\mathbf{A}(\mathbf{N}(\mathbf{x}_i)\tilde{\mathbf{u}}_i) = \mathbf{0}; \qquad i = 1, 2, \ldots, n_e \tag{16.55}$$

and the discrete form for each boundary condition is

$$\mathbf{B}(\mathbf{N}(\mathbf{x}_i)\tilde{\mathbf{u}}_i) = \mathbf{0}; \qquad i = 1, 2, \ldots, n_b \tag{16.56}$$

The total number of equations must equal the number of collocation points selected. Accordingly,

$$n_e + n_b = n \tag{16.57}$$

It would appear that little difference will exist between continuous approximations involving moving least squares and discontinuous ones as in both locally the same polynomial will be used. This may well account for the convergence of standard least square approximations which we have observed in Chapter 3 for discontinuous least square forms but in view of our previous remarks about differentiation, a slight difference will in fact exist if moving least squares are used and in the work of Oñate *et al.*[29–31] which we mentioned before such moving least squares are adopted.

In addition to the choice for $\mathbf{p}(\mathbf{x})$, a key step in the approximation is the choice of the weighting function for the least square method and the domain over which the weighting function is applied. In the work of Orkisz[32] and Liszka[33] two methods are used:

1. A 'cross' criterion in which the domain at a point is divided into quadrants in a cartesian coordinate system originating at the 'point' where the equation is to be evaluated. The domain is selected such that each quadrant contains a fixed number of points, n_q. The product of n_q and the number of quadrants, q, must equal or exceed the number of polynomial terms in $\mathbf{p}$ less one (the central node point). An example is shown in Fig. 16.9(a) for a two-dimensional problem ($q = 4$ quadrants) and $n_q = 2$.
2. A 'Voronoi neighbour' criterion in which the closest nodes are selected as shown for a two-dimensional example in Fig. 16.9(b).

There are advantages and disadvantages to both approaches – namely, the cross criterion leads to dependence on the orientation of the global coordinate axes while the Voronoi method gives results which are sometimes too few in number to get appropriate order approximations. The Voronoi method is, however, effective for use in Galerkin solution methods or finite volume (subdomain collocation) methods in which only first derivatives are needed.

The interested reader can consult reference 27 for examples of solutions obtained by this approach. Additional results for finite point solutions may be found in work by Oñate *et al.*[29] and Batina.[28]

One advantage of considering moving least square approximations instead of simple fixed point weighted least squares is that approximations at points other

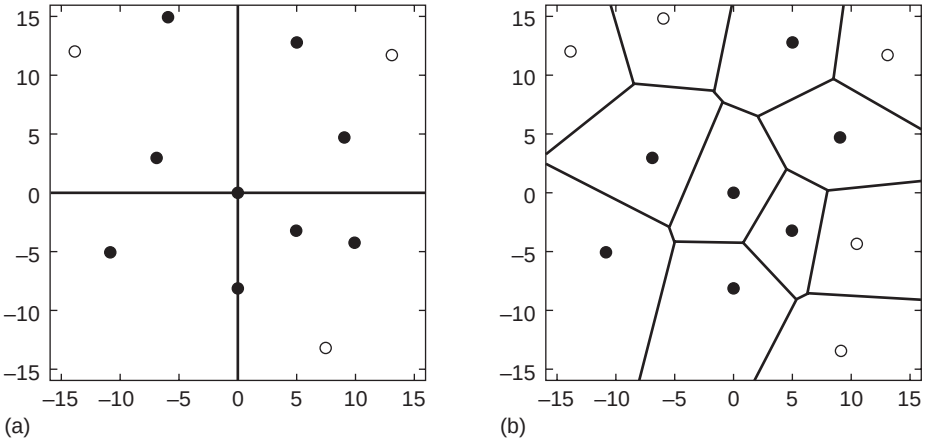


Fig. 16.9 Methods for selecting points: (a) cross; (b) Voronoi.

than those used to write the differential equations and boundary conditions are also continuously available. Thus, it is possible to perform a full post-processing to obtain the contours of the solution and its derivatives.

In the next part of this section we consider the application of the moving least square method to solve a second-order ordinary differential equation using point collocation.

Example: Collocation (point) solution of ordinary differential equations We consider the solution of ordinary differential equations using a point collocation method. The differential equation in our examples is taken as

$$-a\frac{\mathrm{d}^2u}{\mathrm{d}x^2}+b\frac{\mathrm{d}u}{\mathrm{d}x}+cu-f(x)=0 \tag{16.58}$$

on the domain $0<x<L$ with constant coefficients a, b, c, subject to the boundary conditions $u(0)=g_1$ and $u(L)=g_2$. The domain is divided into an equally spaced set of points located at x_i, $i=1,\ldots,n$. The moving least square approximation described in Sec. 16.3 is used to write difference equations at each of the interior points (i.e., $i=2,\ldots,n-1$). The boundary conditions are also written in terms of discrete approximations using the moving least square approximation. Accordingly, for the approximate solution using p-order polynomials to define the $\mathbf{p}(x)$ in the interpolations

$$\hat{u}(x)=\sum_{j=1}^{n}N_i^p(x)\tilde{u}_i \tag{16.59}$$

we have the set of n equations in n unknowns:

$$\sum_{i=1}^{n}N_i^p(x_1)\tilde{u}_i=g_1 \tag{16.60}$$

$$\sum_{i=1}^{n}\left(-a\frac{\mathrm{d}^2N_i^p}{\mathrm{d}x^2}+b\frac{\mathrm{d}^2N_i^p}{\mathrm{d}x^2}+cN_i^p\right)_{x=x_j}\tilde{u}_i-f(x_j)=0;\qquad j=2,\ldots,n-1 \tag{16.61}$$

and

$$\sum_{i=1}^{n} N_i^p(x_n)u_i = g_2 \tag{16.62}$$

The above equations may be written compactly as:

$$\mathbf{Ku} + \mathbf{f} = \mathbf{0} \tag{16.63}$$

where **K** is a square coefficient matrix, **f** is a load vector consisting of the entries from g_i and $f(x_j)$, and **u** is the vector of unknown parameters defining the approximate solution $\hat{u}(x)$. A unique solution to this set of equations requires **K** to be non-singular (i.e., rank(**K**) = n). The rank of **K** depends both on the weighting function used to construct the least square approximation as well as the number of functions used to define the polynomials **p**. In order to keep the least square matrices as well conditioned as possible, a different approximation is used at each node with

$$\mathbf{p}^{(j)}(x) = [\,1 \quad x - x_j \quad (x - x_j)^2 \quad \ldots \quad (x - x_j)^p\,] \tag{16.64}$$

defining the interpolations associated with $N_j^p(x)$. The matrix **K** will be of correct rank provided the weighting function can generate linearly independent equations.

The accurate approximation of second derivatives in the differential equation requires the use of quadratic or higher order polynomials in $\mathbf{p}(x)$.[27] In addition, the span of the weighting function must be sufficient to keep the least squares matrix **H** non-singular *at every collocation point*. Thus, the minimum span needed to define quadratic interpolations of $\mathbf{p}(x)$ (i.e., $p = k = 2$) must include at least three mesh points with non-zero contributions. At the problem boundaries only half of the weighting function span will be used (e.g., the right half at the left boundary). Consequently, for weighting functions which go smoothly to zero at their boundary, a span larger than four mesh spaces is required. The span should not be made too large, however, since the sparse structure of **K** will then be lost and overdiffuse solutions may result.

Use of hierarchical interpolations reduces the required span of the weighting function. For example, use of interpolations with $k = 0$ requires only a span at each point for which the domain is just covered (since any span will include its defining point, x_k, the **H** matrix will always be non-singular). For a uniformly spaced set of points this is any span greater than one mesh spacing.

For the example we use the weighting function described by Eq. (16.12) with a weight span 4.4 ($r_m = 2.2h$) times the largest adjacent mesh space for the quadratic interpolations with $k = p = 2$ and a weight 2.01 times the mesh space for the hierarchical quadratic interpolations with $k = 0$, $p = 2$.

We consider the example of a string on an elastic foundation with the differential equation

$$-a\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + cu + f = 0; \qquad 0 < x < 1 \tag{16.65}$$

with the boundary conditions $u(0) = u(1) = 0$. This is a special form of Eq. (16.58). The parameters for solution are selected as

$$a = 0.01 \qquad c = 1 \qquad f = -1 \tag{16.66}$$

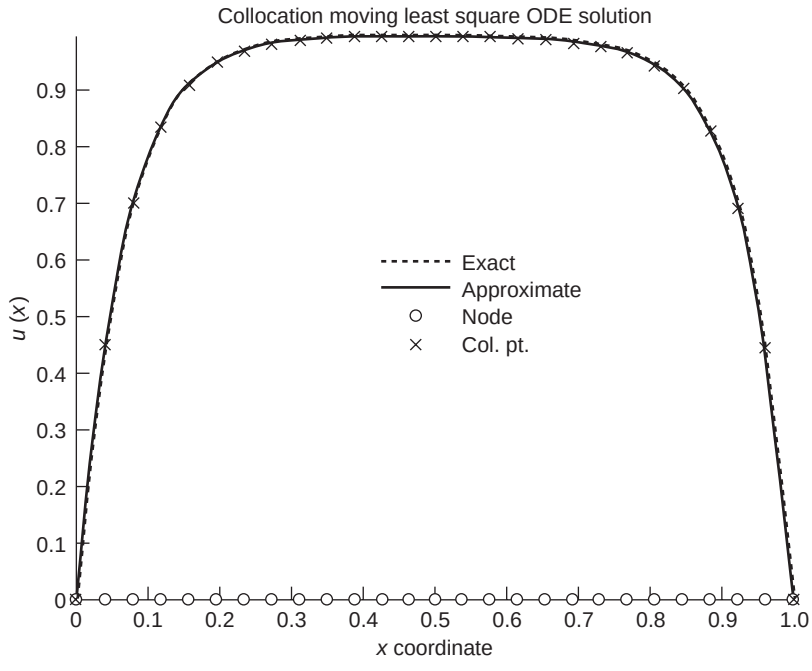


Fig. 16.10 String on elastic foundation solution using MLS form based on nodes: 27 points, $k = 0$, $p = 2$.

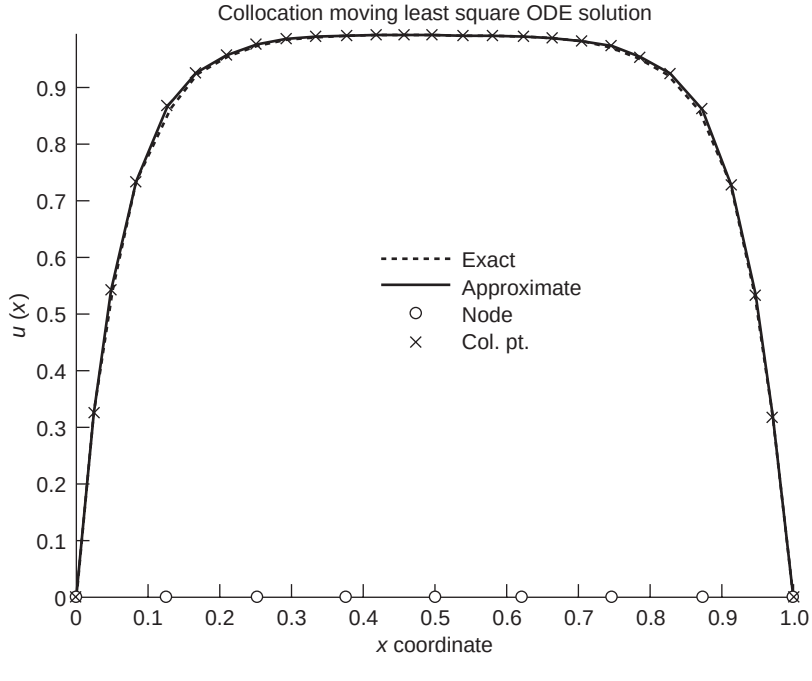


Fig. 16.11 String on elastic foundation hierarchic solution: 9 nodal points, $k = 0$, $p = 2$.

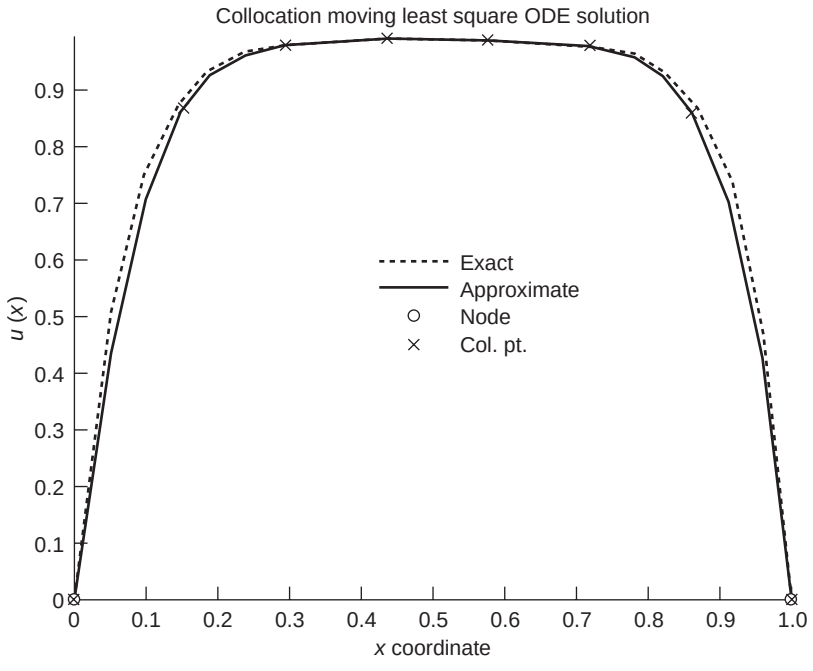


Fig. 16.12 String on elastic foundation hierarchic solution: 2 points, $k = 0$, $p = 3$.

The exact solution is given by

$$u(x) = 1 - \cosh(mx) + (1 - \cosh(m))\frac{\sinh(mx)}{\sinh(m)}, \qquad m = \left(\frac{c}{a}\right)^{1/2} \tag{16.67}$$

The problem is solved using 27 points and $k = p = 2$ producing the results shown in Fig. 16.10.

The process was repeated using the hierarchical interpolations with $k = 0$ and $p = 2$ using nine points (which results in 27 parameters, the same as for the first case). The results are shown in Fig. 16.11.

The hierarchical interpolation permits the solution to be obtained using as few as two points. A solution with two points and interpolations with $k = 0$ and $p = 3$ and 5 is shown in Figs 16.12 and 16.13, respectively. Note however that with the hierarchical form additional collocation points have to be introduced to achieve a sufficient number of equations. We show such collocation points in Fig. 16.11.

16.6 Galerkin weighting and finite volume methods

16.6.1 Introduction

Point collocation methods are straightforward and quite easy to implement, the main task being only the selection of the subdomain on which to perform the fit of the

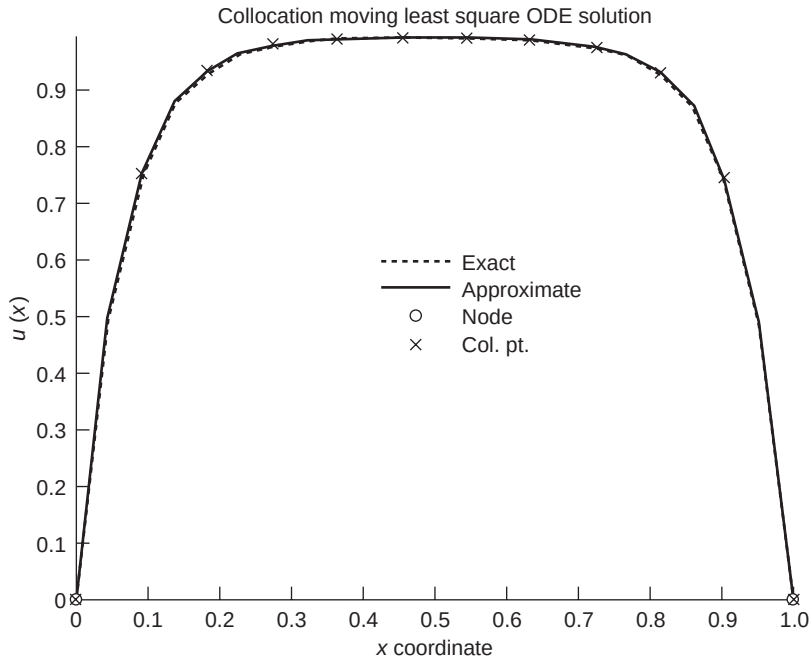


Fig. 16.13 String on elastic foundation solution: 2 points, $k = 0$, $p = 5$.

function from which the derivatives are computed. Disadvantages arise, however, in the need to use high order interpolations such that accurate derivatives of the order of the differential equation may be computed. Further the treatment of boundaries and material interfaces present difficulties.

An alternative, as we have discussed in Chapter 3, is the use of 'weak' or 'variational' forms which are equivalent to the differential equation. Approximations then require functions which have lower order than in the differential equation. In addition, boundary conditions often appear as 'natural' conditions in the weak form – especially for flux (derivative or Neuman) type boundary conditions. This advantage now is balanced by a need to perform integration over the whole domain.

Here, we consider problems of the form given by (see Sec. 3.2)†

$$\int_{\Omega} \mathbf{C}(\mathbf{v})^{\mathrm{T}} \mathbf{D}(\mathbf{u})\, \mathrm{d}\Omega + \int_{\Gamma} \mathbf{E}(\mathbf{v})^{\mathrm{T}} \mathbf{F}(\mathbf{u})\, \mathrm{d}\Gamma = 0 \tag{16.68}$$

in which the operators **C**, **D**, **E** and **F** contain lower derivatives than those occurring in operators **A** and **B** given in Eqs (16.55) and (16.56), respectively. For example, the solution of second-order differential equations (such as those occurring in the quasi-harmonic or linear elasticity equation) have differential operators for **C** to **F** with derivatives no higher than first order.

† We assume that the boundary terms are described such that $\bar{\mathbf{v}} = \mathbf{v}$.

The approximate solution to forms given by Eq. (16.68) may be achieved using moving least squares and alternative methods for performing the domain integrals.

16.6.2 Subdomain collocation – finite volume method

A simple extension of the point collocation method is to use subdomains (elements) defined by the Voronoi neighbour criterion. The integrals for each subdomain are approximated as a *constant* evaluated at the originating point as

$$\sum_{i}^{n_d} \mathbf{C}(\mathbf{v}_i)^{\mathrm{T}} \mathbf{D}(\mathbf{u}_i) \Omega_i + \sum_{i}^{n_b} \mathbf{E}(\mathbf{v}_i)^{\mathrm{T}} \mathbf{F}(\mathbf{u}_i) \Gamma_i = 0 \tag{16.69}$$

where $n_d + n_b = n$, the total number of unknown parameters appearing in the approximations of $\mathbf{u}$ and $\mathbf{v}$.

The validity of the above approximation form can be established using patch tests (see Chapter 10). This approach is often called *subdomain collocation* or the *finite volume* method. This approach has been used extensively in constructing approximations for fluid flow problems.[34–40] It has also been employed with some success in the solution of problems in structural mechanics.[41]

16.6.3 Galerkin methods – diffuse elements

Moving least square approximations have been used with weak forms to construct Galerkin type approximations. The origin of this approach can be traced to the work of Liszka[33] and Orkisz.[27] Additional work, originally called the diffuse element approximation, was presented in the early 1990s by Nayroles *et al.*[11–13] Beginning in the mid-1990s the method has been extensively developed and improved by Belytschko and coauthors under the name *element-free Galerkin*.[14,15,42,43] A similar procedure, call '*hp*-clouds', was also presented by Oden and Duarte.[16,17,44] Each of the methods is also said to be 'meshless', however, in order to implement a true Galerkin process it is necessary to carry out integrations over the domain. What distinguishes each of the above processes is the manner in which these integrations are carried out. In the element-free Galerkin method a background 'grid' is often used to define the integrals whereas in the *hp* cloud method circular subdomains are employed. Differing weights are also used as means to generate the moving least square approximation. The interested reader is referred to the appropriate literature for more details. Another source to consult for implementation of the EFG method is reference 19. Here we present only a simple implementation for solution of an ordinary differential equation.

Example: Galerkin solution of ordinary differential equations The moving least square approximation described in Sec. 16.3; is now used as a Galerkin method to solve a second-order ordinary differential equation. For an arbitrary function $W(x)$, a weak form for the differential equation may be deduced using the procedures

presented in Chapter 3. Accordingly, we obtain

$$\int_0^L \left[\frac{\mathrm{d}W}{\mathrm{d}x} a \frac{\mathrm{d}u}{\mathrm{d}x} + W\left(b\frac{\mathrm{d}u}{\mathrm{d}x} + cu - f(x) \right) \right] \mathrm{d}x = 0 \tag{16.70}$$

subject to the boundary conditions $u(0) = g_1$ and $u(L) = g_2$. Using a *hierarchical moving least square form* a p-order polynomial approximation to the dependent variable may be written as

$$\hat{u}(x) = \sum_{j=1}^{n} N_j^0(x) \bar{\mathbf{q}}_{jp}(x) \tilde{\mathbf{u}}_j^p \tag{16.71}$$

where

$$\bar{\mathbf{q}}_{jp} = [\, 1 \quad x - x_j \quad (x - x_j)^2 \quad \ldots \quad (x - x_j)^p \,] \tag{16.72}$$

Note that in the above form we have used the representation

$$\bar{\mathbf{q}}_{jp}(x) \tilde{\mathbf{u}}_j^p = \tilde{u}_j + \mathbf{q}(x) \tilde{\mathbf{b}}_j$$

The approximation to the weight function is similarly taken as

$$\hat{W}(x) = \sum_{j=1}^{n} N_j^0(x) \bar{\mathbf{q}}_{jp}(x) \tilde{\mathbf{W}}_j^p \tag{16.73}$$

in which $\mathbf{W}_j^p$ are arbitrary parameters satisfying $W(0) = W(L) = 0$. The approximation yields the discrete problem

$$\sum_{i=1}^{n} (\tilde{\mathbf{W}}_i^p)^{\mathrm{T}} \sum_{j=1}^{n} \left\{ \int_0^L \left[\frac{\mathrm{d}(\bar{\mathbf{q}}_{ip}^{\mathrm{T}} N_i^0)}{\mathrm{d}x} a \frac{\mathrm{d}(\bar{\mathbf{q}}_{jp} N_j^0)}{\mathrm{d}x} + \bar{\mathbf{q}}_{ip}^{\mathrm{T}} N_i^0 \left(b \frac{\mathrm{d}(\bar{\mathbf{q}}_{jp} N_j^0)}{\mathrm{d}x} + c \bar{\mathbf{q}}_{jp} N_j^0 \right) \right] \mathrm{d}x \right\} \tilde{\mathbf{u}}_j$$
$$= \sum_{i=1}^{n} (\mathbf{W}_i^p)^{\mathrm{T}} \int_0^L \bar{\mathbf{q}}_{ip}^{\mathrm{T}} N_i^0 f(x)\, \mathrm{d}x \tag{16.74}$$

Since $\mathbf{W}_i^p$ is arbitrary, the solution to the approximate weak form yields the set of equations

$$\sum_{j=1}^{n} \left\{ \int_0^L \left[\frac{\mathrm{d}(\bar{\mathbf{q}}_{ip}^{\mathrm{T}} N_i^0)}{\mathrm{d}x} a \frac{\mathrm{d}(\bar{\mathbf{q}}_{jp} N_j^0)}{\mathrm{d}x} + \bar{\mathbf{q}}_{ip}^{\mathrm{T}} N_i^0 \left(b \frac{\mathrm{d}(\bar{\mathbf{q}}_{jp} N_j^0)}{\mathrm{d}x} + c \bar{\mathbf{q}}_{jp} N_j^0 \right) \right] \mathrm{d}x \right\} \tilde{\mathbf{u}}_j$$
$$= \int_0^L \bar{\mathbf{q}}_{ip}^{\mathrm{T}} N_i^0 f(x)\, \mathrm{d}x; \qquad i = 1, 2, \ldots, n \tag{16.75}$$

The set of equations only needs to be modified to satisfy the essential boundary equations. This is accomplished by replacing the equations corresponding to $W_1 = W_n = 0$ by $\tilde{u}_1 = g_1$ and $\tilde{u}_n = g_2$.

The Galerkin form requires only first derivatives of the approximating functions as opposed to the second derivatives required for the point collocation method. This reduction, however, is accompanied by a need to perform integrals over the domain. For weighting functions given by Eq. (16.12) all functions entering the approximation are polynomial and rational polynomial expressions, thus, a closed form evaluation is impractical. Accordingly, we evaluate integrals using Gauss and

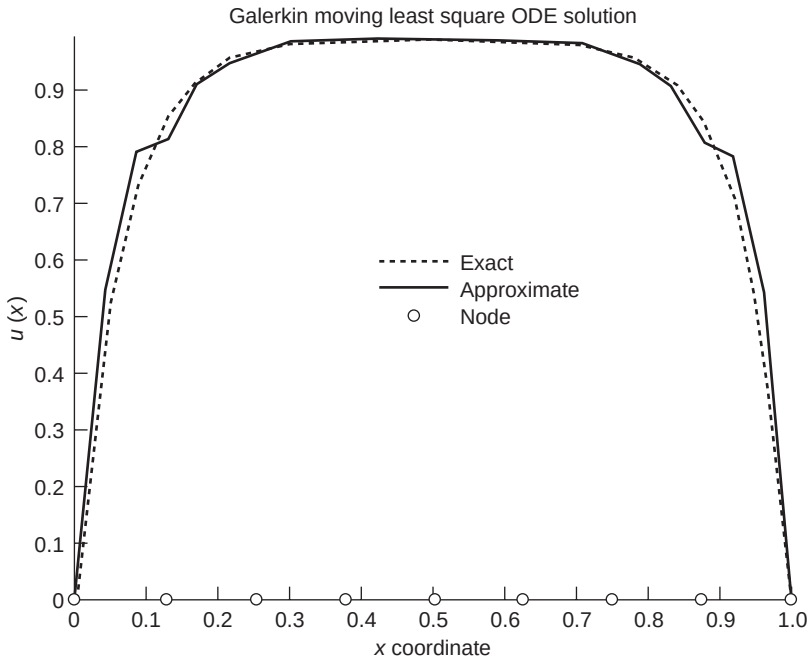


Fig. 16.14 String on elastic foundation solution: 3-point Gauss quadrature.

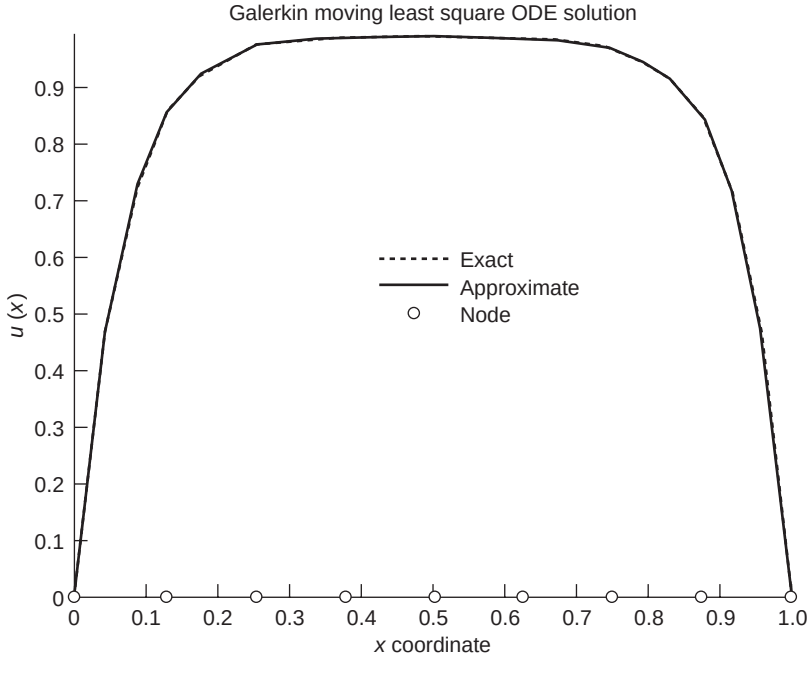


Fig. 16.15 String on elastic foundation solution: 4-point Gauss quadrature.

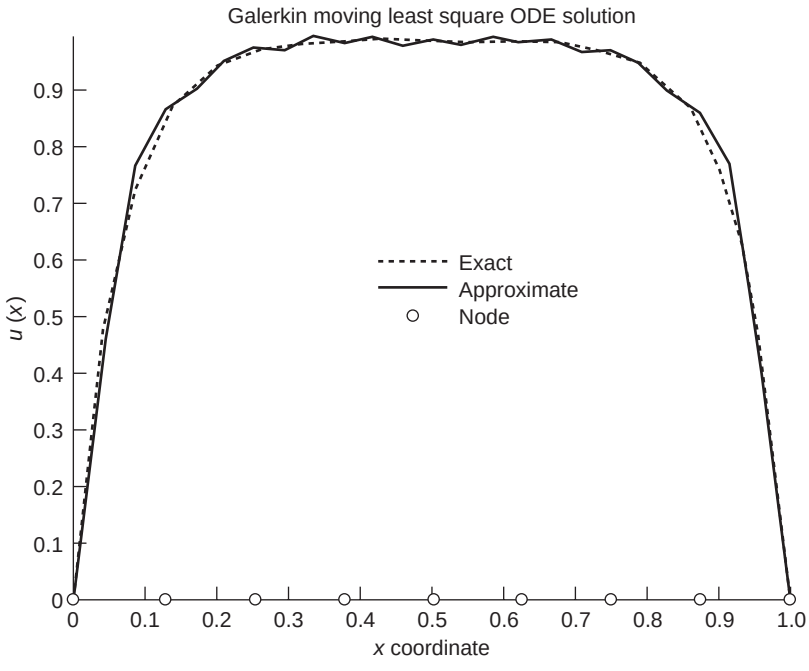


Fig. 16.16 String on elastic foundation solution: 4-point Gauss–Lobatto quadrature.

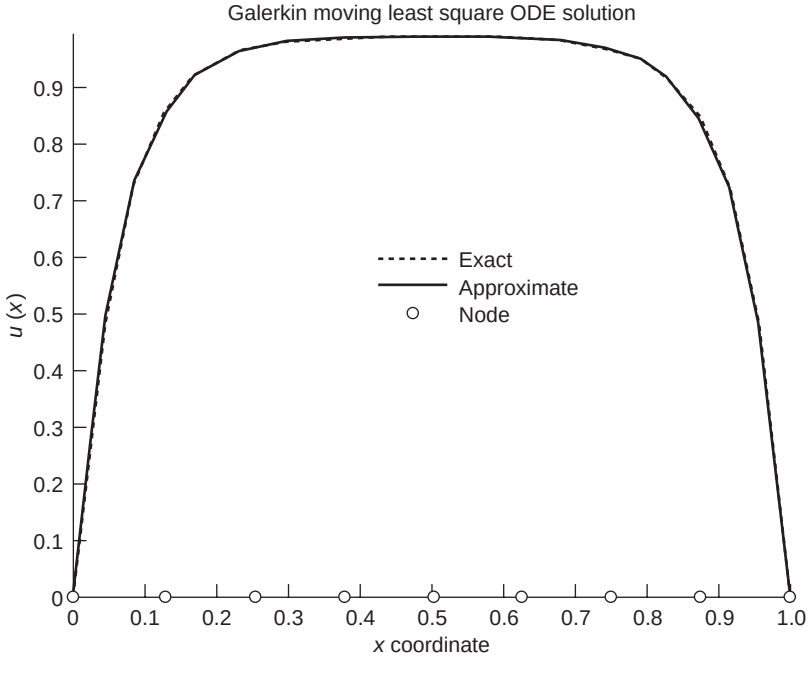


Fig. 16.17 String on elastic foundation solution: 5-point Gauss–Lobatto quadrature.

Gauss–Lobatto quadrature over each *interval* generated by the basis points in the moving least square representation (i.e., x_j for $j = 1, 2, \ldots, n$). As an example of the type of solutions possible we consider the string on elastic foundation problem given in the previous section. For the parameters $a = 0.004$, $c = 1$ with loading $f = -1$ and zero boundary conditions a Galerkin solution using 3 and 4 point Gauss quadrature and 4 and 5 point Gauss–Lobatto quadrature is shown in Figs 16.14–16.17. A mesh consisting of nine equally spaced points is used to define the intervals for the solution and quadrature. The weight function is generated for $k = 0$, $p = 2$ with a span of 2.1 mesh points.

Based upon this elementary example it is evident that the answers for a nine-point mesh depend on accurate evaluation of integrals to produce high-quality answers.

16.7 Use of hierarchic and special functions based on standard finite elements satisfying the partition of unity requirement

16.7.1 Introduction

In Sec. 16.4, we discussed the possibility of introducing hierarchical variables to shape functions based on moving least square interpolations. However a simpler approach to hierarchical forms and indeed to extensions by other functions can be based on simple finite element shape functions.

One important application of the partition of unity method starts from a set of finite element basis functions, $N_i(\mathbf{x})$. An approximation to $u(x)$ is now given by

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_i N_i(\mathbf{x}) \left[\tilde{u}_i + \sum_\alpha q_\alpha^{(i)}(\mathbf{x}) b_{\alpha i} \right] \tag{16.76}$$

where $N_i(\mathbf{x})$ is the conventional (possibly isoparametric) finite element shape function at node i, $q_\alpha^{(i)}$ are *global functions* associated with node i, and $\tilde{u}_i$, and $b_{\alpha i}$ are parameters associated with the added global hierarchical functions. We must note that as before $\tilde{u}_i$ will not represent a local value of the function unless the function q^i become zero at the node i.

Here we assume that conventional shape functions which satisfy the partition of unity condition

$$\sum_i N_i = 1 \tag{16.77}$$

are used. Thus, the above form is a *hierarchic finite element method based on the partition of unity*.[21,45,46] We note in particular that the function $q_\alpha^{(i)}$ may be different for each node and thus the form may be effectively used in an adaptive finite element procedure as described in Chapter 15.

Equation (16.76) provides options for a wide choice of functions for $q_\alpha^{(i)}$:

1. Polynomial functions. In this case the method becomes an alternative hierarchical scheme to that presented in Part 2 of Chapter 8.

2. Harmonic 'wave' functions. This is a multiscale method and will be discussed in detail in Volume 3.
3. Singular functions. These can be used to introduce re-entrant corner or singular load effects in elliptic problems (e.g., heat conduction or elasticity forms).

Derivatives of Eq. (16.76) are computed directly as

$$\frac{\partial \hat{u}}{\partial x_k} = \sum_i \left[\frac{\partial N_i}{\partial x_k} \tilde{u}_i + \sum_\alpha \left(\frac{\partial N_i}{\partial x_k} q_\alpha^{(i)} + N_i \frac{q_\alpha^{(i)}}{\partial x_k} \right) b_{\alpha i} \right] \tag{16.78}$$

The reader will note that the narrow band structure of the standard finite element method will always be maintained as it is determined by the connectivity of N_i. Note also that the standard element on which the shape functions N_i were generated can be used for all subsequent integrations. Such a formulation is very easy to fit into any finite element program.

16.7.2 Polynomial hierarchical method

To give more details of the above hierarchical finite element method we first consider the one-dimensional approximation in a two-noded element where

$$\hat{u} = N_1[\tilde{u}_1 + \mathbf{q}^{(1)}\mathbf{b}_1] + N_2[\tilde{u}_2 + \mathbf{q}^{(2)}\mathbf{b}_2] \tag{16.79}$$

in which

$$N_1 = \frac{x_2 - x}{x_2 - x_1}; \qquad N_2 = \frac{x - x_1}{x_2 - x_1}$$

and

$$\begin{aligned} \mathbf{q}^{(1)} = \mathbf{q}^{(2)} &= [\, x^k, \quad x^{k+1}, \quad \cdots] \\ \mathbf{b}_i &= [\, b_{i1}, \quad b_{i2}, \quad \cdots]^{\mathrm{T}} \end{aligned} \tag{16.80}$$

We recall that $N_1 + N_2 = 1$ and $N_1 x_1 + N_2 x_2 = x$.

Investigation of the term x^k in the approximation

$$\hat{u} = N_1(x)\left[\tilde{u}_1 + x^j a_{k1}\right] + N_2(x)\left[\tilde{u}_2 + x^j a_{k2}\right] \tag{16.81}$$

we observe that a linear dependence with the usual finite element approximation occurs when $\tilde{u}_i = x_i \tilde{b}_0$ and $k = 1$ with $b_{11} = b_{12} = \tilde{b}_1$. In this case Eq. (16.81) becomes

$$\begin{aligned} \hat{u} &= [N_1 x_1 + N_2 x_2]\tilde{b}_0 + [N_1 + N_2] x \tilde{b}_1 \\ &= x\tilde{b}_0 + x\tilde{b}_1 \end{aligned} \tag{16.82}$$

In one dimension linear dependence can be avoided by setting k to 2 in Eqs (16.79) and (16.80). However, in two and three dimensional problems the linear dependence cannot be completely avoided, and we address this next.[45,47]

An approximation over two-dimensional triangles may be expressed as

$$u(x, y) \approx \hat{u}(x, y) = \sum_{i=1}^{3} L_i[\tilde{u}_i + \mathbf{q}^{(i)}\mathbf{b}_i] \tag{16.83}$$

where L_i are the area coordinates defined in Chapter 8. We consider the case where complete quadratic functions are added as

$$\mathbf{q}^{(i)} = \begin{bmatrix} x^2, & xy, & y^2 \end{bmatrix} \tag{16.84}$$

to give a complete second-order polynomial approximation for u. Although this gives a complete second-order polynomial approximation there are two ways in which the cubic term x^2y can be obtained.

1. The first sets

$$\tilde{u}_i = b_{i1} = b_{i3} = 0 \qquad \text{and} \qquad b_{i2} = x_i\tilde{\alpha}$$

giving

$$\hat{u} = \sum_{i=1}^{3} L_i \cdot [xy] \cdot x_i\tilde{\alpha} = x^2y\tilde{\alpha}$$

2. The second alternative to compute the same term sets

$$\tilde{u}_i = b_{i2} = b_{i3} = 0 \qquad \text{and} \qquad b_{i1} = y_i\tilde{\alpha}$$

giving

$$\hat{u} = \sum_{i=1}^{3} L_i \cdot [x^2] \cdot y_i\tilde{\alpha} = x^2y\tilde{\alpha}$$

A similar construction may be made for the polynomial term xy^2.

An alternative is to construct the interpolation to depend on each node as

$$\mathbf{q}^{(i)} = \begin{bmatrix} (x-x_i)^2 & (x-x_i)(y-y_i) & (y-y_i)^2 \end{bmatrix} \tag{16.85}$$

This form, while conceptually the same as the original formulation, appears to be better conditioned and also avoids some of the problems of linear dependency.[47] In Sec. 16.7.4 we will discuss in more detail a methodology to deal with the problem of linear dependency, however, before doing so we illustrate the use of the hierarchical finite element method by an application to two-dimensional problems in linear elasticity.

16.7.3 Application to linear elasticity

In the previous section the form for polynomial interpolation in two dimensions was given. Here we consider the use of the interpolation to model the behaviour of problems in linear elasticity. For simplicity only the displacement model for plane strain as discussed in Chapters 2 and 4 is considered; however, the use of the hierarchic interpolations can easily be extended to other forms and to mixed models.

For a displacement model the finite element arrays may be computed using Eq. (2.24). For two-dimensional plane strain problems, the strain–displacement

relations may be written in matrix form as

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{bmatrix} \tag{16.86}$$

Inserting the interpolations for u and v given by Eq. (16.76) and using Eq. (16.78) to compute derivatives, the strain–displacement relations become

$$\begin{aligned} \boldsymbol{\varepsilon} = & \sum_{i=1}^{N} \begin{bmatrix} \dfrac{\partial N_i^k}{\partial x} & 0 \\ 0 & \dfrac{\partial N_i^k}{\partial y} \\ \dfrac{\partial N_i^k}{\partial y} & \dfrac{\partial N_i^k}{\partial x} \end{bmatrix} \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} \\ & + \sum_{i=1}^{N} \begin{bmatrix} \left(\dfrac{\partial N_i^k}{\partial x} \mathbf{q}_i^k + N_i^k \dfrac{\partial \mathbf{q}_i^k}{\partial x} \right) & 0 \\ 0 & \left(\dfrac{\partial N_i^k}{\partial x} \mathbf{q}_i^k + N_i^k \dfrac{\partial \mathbf{q}_i^k}{\partial y} \right) \\ \left(\dfrac{\partial N_i^k}{\partial x} \mathbf{q}_i^k + N_i^k \dfrac{\partial \mathbf{q}_i^k}{\partial y} \right) & \left(\dfrac{\partial N_i^k}{\partial x} \mathbf{q}_i^k + N_i^k \dfrac{\partial \mathbf{q}_i^k}{\partial x} \right) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{b}}_i^u \\ \tilde{\mathbf{b}}_i^v \end{bmatrix} \end{aligned} \tag{16.87}$$

The first term is identical to the usual finite element strain–displacement matrices [see Eq. (4.10b)] and the second term has identical structure to the usual arrays. Thus, the development of all element arrays follows standard procedures.

A quadratic triangular element

For a triangular element with linear interpolation the shape functions and quadratic polynomial hierarchic terms are given by $N_i = L_i$ and Eq. (16.85), respectively. Using isoparametric concepts the coordinates are given by

$$\mathbf{x} = \sum_{i=1}^{3} N_i^1 \tilde{\mathbf{x}}_i = \sum_{i=1}^{3} L_i \tilde{\mathbf{x}}_i \tag{16.88}$$

and are used to construct all polynomials appearing in hierarchical form (16.85).

A set of patch tests is first performed to assess the stability and consistency of the above hierarchic form. The set consists of one, two, four, and eight element patches as shown in Fig. 16.18. First, we perform a stability assessment by determining the number of zero eigenvalues for each patch. The results for hierarchical interpolation are shown in Table 16.3.

The eigenproblem assessment reveals that the hierarchic interpolation has excess zero eigenvalues (i.e., spurious zero energy modes) only for meshes consisting of

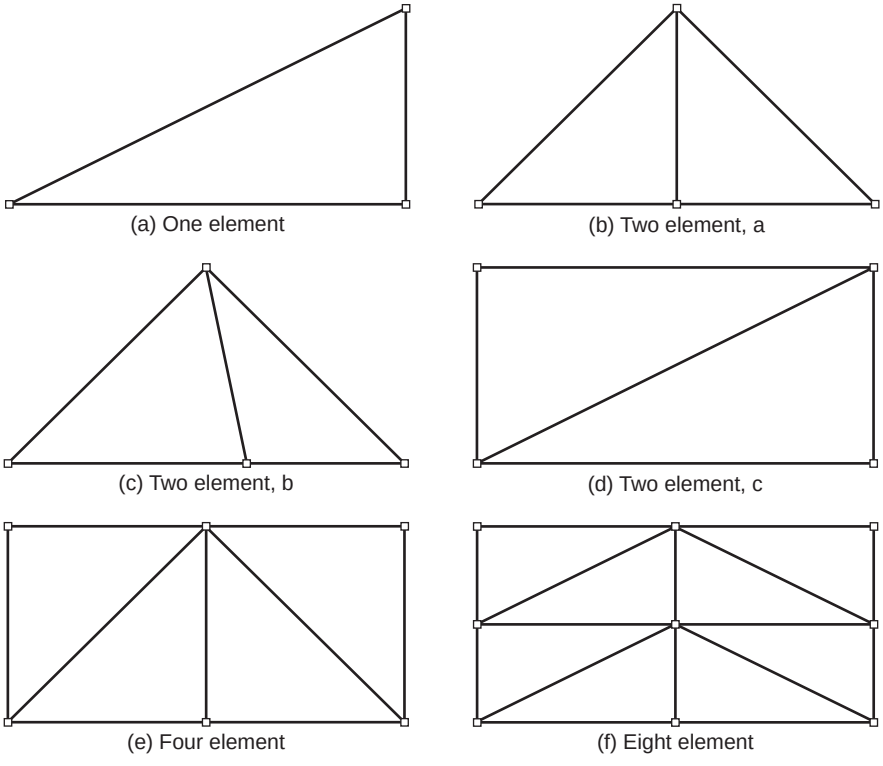


Fig. 16.18 Patches for eigenproblem assessment.

one or two elements. Furthermore, only two element meshes in which one side is a straight line through both elements have excess zero values. Once the mesh has no straight intersections the number of zero modes becomes correct (e.g., contain only the three rigid body modes).

Consistency tests verify that all meshes contain terms of up to quadratic polynomial order – thus also validating the correctness of the coding.

As a simple test problem using the hierarchical finite element method we consider a finite width strip containing a circular hole with diameter half the width of the strip. The strip is subjected to axial extension in the vertical direction and, due to symmetry

Table 16.3 Triangle element patch tests: Number of zero eigenvalues, minimum non-zero value, and maximum value ($k = 2$) – quadratic hierarchical terms

Mesh	No. zero	Min. value	Max. value
1	7	4.7340E + 01	2.0560E + 06
2a	5	4.0689E + 01	2.1543E + 05
2b	5	4.1971E + 02	2.2648E + 05
2c	3	1.5728E + 02	2.3883E + 06
4	3	1.0446E + 02	2.9027E + 05
8	3	9.5560E + 01	3.4813E + 05

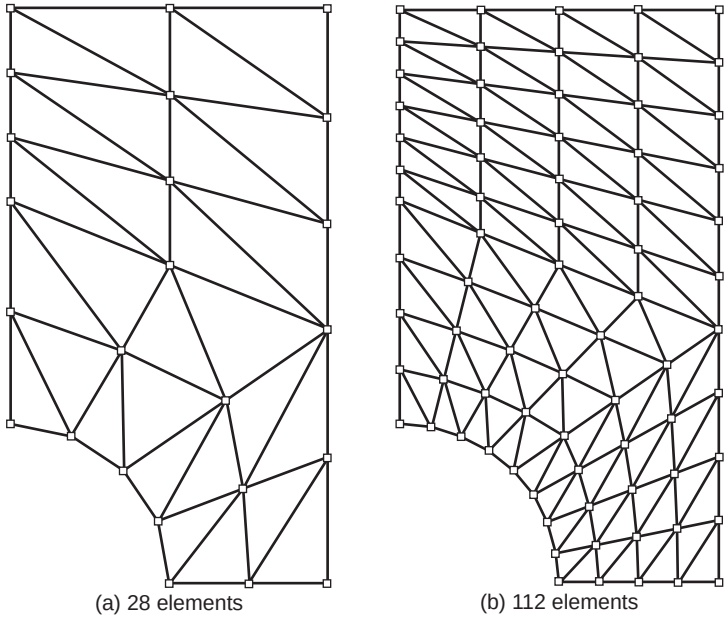


Fig. 16.19 Hierarchic elements: tension strip.

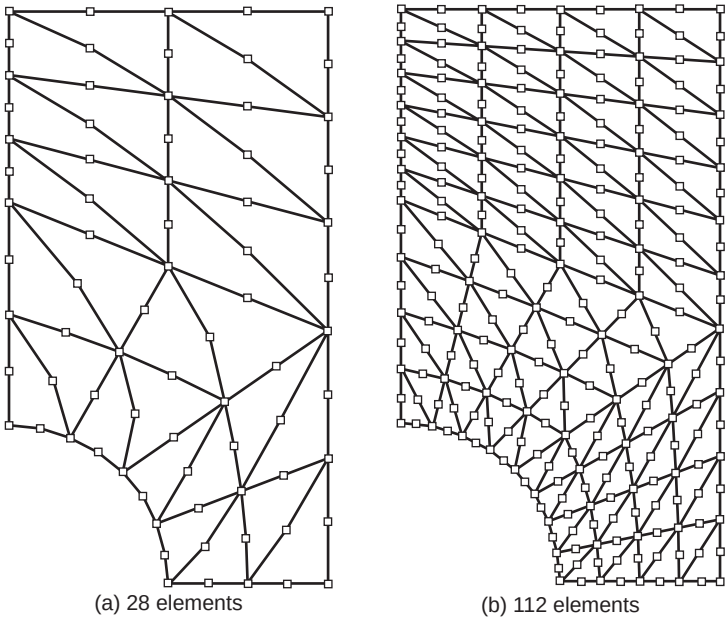


Fig. 16.20 Isoparametric six-noded elements: tension strip.

Table 16.4 Hierarchical element. Boundary segments straight

Nodes	Elements	Equations	Energy
30	28	156	131.7088
85	112	537	127.8260
279	448	1971	126.7641
1003	1792	7527	126.5908

of the loading and geometry, only one quadrant is discretized as shown in Figs 16.19 and 16.20.

The meshes in Fig. 16.19 employ the hierarchical interpolation considered above; whereas those in Fig. 16.20 use standard six-node isoparametric quadratic triangles with two degrees of freedom per node (i.e., u and v). The material is taken as linear elastic with $E = 1000$ and $\nu = 0.25$. The half-width of the strip is 10 units and the half-height is 18 units. The hole has radius 5.

The problem size and computed energy (which indicates solution accuracy) are shown in Table 16.4 for the hierarchical method, in Table 16.5 for the six-node isoparametric formulation and in Table 16.6 for three-node linear triangular elements.

The six-node isoparametric method gives overall the best accuracy; however, the hierarchical element is considerably better than the three-node triangular element and offers great advantages when used in adaptive analysis.[47]

16.7.4 Solution of forms with linearly dependent equations

A typical problem for a steady-state analysis in which the algebraic equations are generated from the hierarchical finite element form described above, such as given

Table 16.5 Isoparametric element. Boundary segments have curved sides

Nodes	Elements	Equations	Energy
30	28	129	127.3350
279	112	483	126.6483
1003	448	1863	126.5661
3795	1792	7311	126.5593

Table 16.6 Linear triangular element

Nodes	Elements	Equations	Energy
30	28	36	137.652
85	112	129	131.065
279	448	483	128.008
1003	1792	1863	126.958
3795	7168	7311	126.662

by Eqs (16.83) and (16.84), produces algebraic equations in the standard form, i.e.,

$$\mathbf{K}\tilde{\mathbf{a}} + \mathbf{f} = \mathbf{0} \tag{16.89}$$

where the parameters $\tilde{\mathbf{a}}$ include both nodal $\tilde{u}_i$ and hierarchical parameters $\mathbf{b}_i$. We assume that occasionally the 'stiffness matrix' $\mathbf{K}$ and 'force' vector $\mathbf{f}$ include equations which are linearly dependent with other equations in the system and, thus, $\mathbf{K}$ can be singular.

If the system is solved by a direct elimination scheme (e.g., as described in Chapter 2 or in books on linear algebra such as references 48 or 49) it is possible to set a tolerance for the pivot below which an equation is assumed to be linearly dependent and can be omitted from the calculations (e.g., see reference 50, 51).

An alternative to the above is to perturb Eq. (16.89) to

$$[\mathbf{K} + \varepsilon \mathbf{D}_K]\Delta\tilde{\mathbf{a}}^k = \mathbf{f} - \mathbf{K}\tilde{\mathbf{a}}^k \tag{16.90}$$

where $\mathbf{D}_K$ are diagonal entries of $\mathbf{K}$, ε is a specified value and

$$\mathbf{a}^{k+1} = \mathbf{a}^k + \Delta\mathbf{a}^k \tag{16.91}$$

is used to define an iterative strategy. An initial guess of zero may be used to start the solution process. Certainly a choice of a small value for ε (e.g., 10^{-6}) leads to rapid convergence.[47]

16.8 Closure

In this chapter we have considered a number of methods which eliminate or reduce our dependence on meshing the total domain. There are a number of other approaches having the same aim which have been pursued with success. These include the *smooth particle hydrodynamics* method (SPH) (Lucy,[52] Gingold and Monaghan,[53] Benz[54]) and the *reproducing kernel* method (RPK) (Liu *et al.*[55,56]) applied to problems in solid and fluid mechanics. Bonet and coworkers[57,58] improve the method of SPH and show its possibilities. Another approach has recently been introduced by Yagawa.[59,60] These are not described here and the reader is referred to the literature for details.

References

1. V. Girault. Theory of a finite difference method on irregular networks. *SIAM J. Num. Anal.*, **11**, 260–82, 1974.
2. V. Pavlin and N. Perrone. Finite difference energy techniques for arbitrary meshes. *Comp. Struct.*, **5**, 45–58, 1975.
3. C. Snell, D.G. Vesey, and P. Mullord. The application of a general finite difference method to some boundary value problems. *Comp. Struct.*, **13**, 547–52, 1981.
4. T. Liszka and J. Orkisz. Finite difference methods of arbitrary irregular meshes in non-linear problems of applied mechanics. In *Proc. 4th Int. Conference on Structural Mechanics in Reactor Technology*, San Francisco, California, 1977.
5. T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its applications in applied mechanics. *Comp. Struct.*, **11**, 83–95, 1980.

6. J. Krok and J. Orkisz. A unified approach to the FE generalized variational FD method in nonlinear mechanics. Concept and numerical approach. In *Discretization methods in structural mechanics*, pages 353–362. Springer-Verlag, Berlin-Heidelberg, 1990. IUTAM/IACM Symposium, Vienna, 1989.
7. R. A. Nay and S. Utku. An alternative for the finite element method. *Variat. Meth. Engng*, **1**, 1972.
8. D. Shepard. A two-dimensional function for irregularly spaced data. In *ACM National Conference*, pages 517–24, 1968.
9. P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Math. Comput.*, **37**, 141–58, 1981.
10. P. Lancaster and K. Salkauskas. *Curve and Surface Fitting*. Academic Press, 1990.
11. B. Nayroles, G. Touzot, and P. Villon. La méthode des éléments diffuse. *C.R. Acad. Sci. Paris*, **313**, 133–38, 1991.
12. B. Nayroles, G. Touzot, and P. Villon. L'approximation diffuse. *C.R. Acad. Sci. Paris*, **313**, 293–96, 1991.
13. B. Nayroles, G. Touzot, and P. Villon. Generalizing the FEM: diffuse approximation and diffuse elements. *Comput. Mech.*, **10**, 307–18, 1992.
14. T. Belytschko, Y. Lu, and L. Gu. Element free Galerkin methods. *Intern. J. Num. Meth. Engng*, **37**, 397–414, 1994.
15. T. Belytschko, Y. Lu, and L. Gu. Crack propagation by element-free Galerkin methods. *Engng. Fracture Mech.*, **51**, 295–315, 1995.
16. A. Duarte and J.T. Oden. *hp* clouds – A meshless method to solve boundary-value problems. Technical Report TICAM Report 95–05, The University of Texas, May 1995.
17. C.A. Duarte and J.T. Oden. An $h-p$ adaptive method using clouds. *Comput. Meth. Appl. Mech. Engng*, **139**(1-4), 237–62, 1996.
18. T. Belytschko, J. Fish, and A. Bayless. The spectral overlay on finite elements for problems with high gradients. *Comput. Meth. Appl. Mech. Engng*, **81**, 71–89, 1990.
19. J. Dolbow and T. Belytschko. An introduction to programming the meshless element free Galerkin method. *Arch. Comput. Meth. Engng*, **5**(3), 207–41, 1998.
20. I. Babuška and J.M. Melenk. The partition of unity finite element method. Technical Report Technical Note BN-1185, Institute for Physical Science and Technology, University of Maryland, April 1995.
21. J.M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Comput. Meth. Appl. Mech. Engng*, **139**, 289–314, 1996.
22. W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.
23. I. Babuška and J.M. Melenk. The partition of unity method. *Intern. J. Num. Meth. Engng*, **40**, 727–58, 1997.
24. L. Collatz. *The Numerical Treatment of Differential Equations*. Springer, Berlin, 1966.
25. G.E. Forsythe and W.R. Wasow. *Finite Difference Methods for Partial Differential Equations*. John Wiley & Sons, New York, 1960.
26. R. D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems*. Wiley (Interscience), New York, 1967.
27. J. Orkisz. Finite difference method. In M. Kleiber, editor, *Handbook of Computational Solid Mechanics*. Springer-Verlag, Berlin, 1998.
28. J. Batina. A gridless Euler/Navier-Stokes solution algorithm for complex aircraft applications. In *AIAA 93–0333*, Reno, NV, January 1993.
29. E. Oñate, S. Idelsohn, and O.C. Zienkiewicz. Finite point methods in computational mechanics. Technical Report CIMNE Report 67, Int. Center for Num. Meth. Engr., Barcelona, July 1995.
30. E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, and C. Sacco. A stabilized finite

point method for analysis of fluid mechanics problems. *Comput. Meth. Appl. Mech. Engng*, **139**, 315–46, 1996.
31. E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, and R.L. Taylor. A finite point method in computational mechanics. Applications to convective transport and fluid flow. *Intern. J. Num. Meth. Engng*, **39**, 3839–66, 1996.
32. J. Orkisz. Computer approach to the finite difference method (in polish). *Mechanika i Komputer*, **2**, 7–69, 1979.
33. T. Liszka. An interpolation method for an irregular net of nodes. *Intern. J. Num. Meth. Engng*, **20**, 1599–1612, 1984.
34. R.B. Pelz and A. Jameson. Transonic flow calculations using triangular finite elements. *AIAA J.*, **23**, 569–76, 1985.
35. J.T. Batina. Vortex-dominated conical-flow computations using unstructured adaptively-refined meshes. *AIAA J.*, **28**(1l), 1925–32, 1990.
36. J.T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA J.*, **28**(8), 1381–88, 1990.
37. D. J. Mavriplis and A. Jameson. Multigrid solution of the navier-stokes equations on triangular meshes. *AIAA J.*, **28**, 1415–25, 1990.
38. R.D. Rausch, J.T. Batina, and H.T.Y. Yang. Spatial adaptation of unstructured meshes for unsteady aerodynamic flow computations. *AIAA J.*, **30**(5), 1243–51, 1992.
39. R.D. Rausch, J.T. Batina, and H.T.Y. Yang. Three-dimensional time-marching aeroelastic analyses using an unstructured-grid euler method. *AIAA J.*, **31**(9), 1626–33, 1993.
40. K. Xu, L. Martinelli, and A. Jameson. Gas-kinetic finite volume methods. In S.M. Deshpande, S.S. Desai, and R. Narasimha (eds), *Proc. 14th Int. Conf. Num. Meth. Fluid Dynamics*, pages 106–111, 1995.
41. E. Oñate, F. Zarate, and F. Flores. A simple triangular element for thick and thin plate and shell analysis. *Intern. J. Num. Meth. Engng*, **37**, 2569–82, 1994.
42. Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element-free. *Comput. Meth. Appl. Mech. Engng*, **113**, 397–414, 1994.
43. M. Tabbara, T. Blacker, and T. Belytschko. Finite element derivative recovery by moving least square interpolates. *Comput. Meth. Appl. Mech. Engng*, **117**, 211–23, 1994.
44. C.A. Duarte. A review of some meshless methods to solve partial differential equations. Technical Report TICAM Report 95-06, The University of Texas, May 1995.
45. R.L. Taylor, O.C. Zienkiewicz, and E. Oñate. A hierarchical finite element method based on the partition of unity. *Comput. Meth. Appl. Mech. Engng*, **152**, 73–84, 1998.
46. J.T. Oden, C.A. Duarte, and O.C. Zienkiewicz. A new cloud-based *hp* finite element method. *Comp. Meth. App. Mech. Eng.*, **152**, 73–84, 1998.
47. C.A. Duarte, I. Babuška, and J.T. Oden. Generalized finite element methods for three dimensional structural mechanics problems, in preparation.
48. G. Strang. *Linear Algebra and its Application*. Academic Press, New York, 1976.
49. J. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
50. I.S. Duff and J.K. Reid. Exploiting zeros on the diagonal in the direct solution of indefinite sparse linear systems. *ACM Transactions on Mathematical Software*, **22**, 227–57, 1996.
51. I.S. Duff and J.A. Scott. Ma62 – a frontal code for sparse positive-definite symmetric systems from finite element applications. In M. Papadrakakis and B.H.V. Topping (eds), *Innovative Computational Methods for Structural Mechanics*, pages 1–25, June 1999.
52. L.B. Lucy. A numerical approach to the testing of fusion process. *The Astron. J.*, 88, 1977.
53. R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of Royal Astron. Sci.*, 181, 1977.
54. W. Benz. Smoothed particle hydrodynamics: A review. Preprint 2884, 1989.

55. W.K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko. Reproducing kernel particle methods for structural dynamics. *Comput. Meth. Appl. Mech. Engng*, **38**, 1655–79, 1995.
56. W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *Intern. J. Num. Meth. Engng*, **20**, 1081–1106, 1995.
57. J. Bonet, S. Kulasetaram, and T.-S.L. Lok. Corrected smooth particle hydrodynamics methods for fluid and solid mechanics application. In W. Wunderlich, editor, *Proceedings First European Conference on Computational Mechanics*, August–September 1999. CD-ROM Version.
58. J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *Intern. J. Num. Meth. Engng*, **47**(to appear), 2000.
59. G. Yagawa and T. Yamada. Free mesh method. a kind of meshless finite element method. *Comput. Mech.*, **18**, 383–86, 1996.
60. G. Yagawa, T. Yamada, and T. Furukawa. Parallel computing with free mesh method: Virtually meshless fem. In H.A. Mang and F.G. Rammerstorfer (eds), *IUTAM Sym., Solid Mechanics and its Applications*, pages 165–172. Kluwer Acd. Pub., 1997.

17

The time dimension – semi-discretization of field and dynamic problems and analytical solution procedures

17.1 Introduction

In all the problems considered so far in this text conditions that do not vary with time were generally assumed. There is little difficulty in extending the finite element idealization to situations that are time dependent.

The range of practical problems in which the time dimension has to be considered is great. Transient heat conduction, wave transmission in fluids and dynamic behaviour of structures are typical examples. While it is usual to consider these various problems separately – sometimes classifying them according to the mathematical structure of the governing equations as 'parabolic' or 'hyperbolic'[1] – we shall group them into one category to show that the formulation is identical.

In the first part of this chapter we shall formulate, by a simple extension of the methods used so far, matrix differential equations governing such problems for a variety of physical situations. Here a finite element discretization in the space dimension only will be used and a semi-discretization process followed (see Chapter 3). In the remainder of this chapter various analytical procedures of the solution for the resulting ordinary linear differential equation system will be dealt with. These form the basic arsenal of steady-state and transient analysis.

Chapter 18 will be devoted to the discretization of the time domain itself.

17.2 Direct formulation of time-dependent problems with spatial finite element subdivision

17.2.1 The 'quasi-harmonic' equation with time differential

In many physical problems the quasi-harmonic equation takes the form in which time derivatives of the unknown function ϕ occur. In the three-dimensional case typically

we might have

$$\frac{\partial}{\partial x}\left(k\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial\phi}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial\phi}{\partial z}\right)+\left(\bar{Q}-\mu\frac{\partial\phi}{\partial t}-\rho\frac{\partial^2\phi}{\partial t^2}\right)=0 \tag{17.1}$$

In the above, quite generally, all the parameters may be prescribed functions of time, or in non-linear cases of ϕ, as well as of space $\mathbf{x}$, i.e.,

$$k=k(\mathbf{x},\phi,t)\qquad \bar{Q}=\bar{Q}(\mathbf{x},\phi,t)\quad \text{etc.} \tag{17.2}$$

If a situation at a particular instant of time is considered, the time derivatives of ϕ and all the parameters can be treated as *prescribed functions of space coordinates*. Thus, at that instant the problem is precisely identified with those treated in Chapter 7 if the whole of the quantity in the last parentheses of Eq. (17.1) is identified as the source term Q.

The finite element discretization of this in terms of *space* elements has already been fully discussed and we found that with the prescription

$$\begin{aligned}\phi&=\sum N_i a_i=\mathbf{N}\mathbf{a}\\ \mathbf{N}&=\mathbf{N}(x,y,z)\qquad \mathbf{a}=\mathbf{a}(t)\end{aligned} \tag{17.3}$$

for each element, the standard form of assembled equations†

$$\mathbf{K}\mathbf{a}+\bar{\mathbf{f}}=\mathbf{0} \tag{17.4}$$

was obtained. Element contributions to the above matrices are defined in Chapter 7 and need not be repeated here except for that representing the 'load' term due to Q. This is given by

$$\bar{\mathbf{f}}=-\int_\Omega \mathbf{N}^{\mathrm{T}}Q\,\mathrm{d}\Omega \tag{17.5}$$

Replacing Q by the last bracketed term of Eq. (17.1) we have

$$\bar{\mathbf{f}}=-\int_\Omega \mathbf{N}^{\mathrm{T}}\left(\bar{Q}-\mu\frac{\partial\phi}{\partial t}-\rho\frac{\partial^2\phi}{\partial t^2}\right)\mathrm{d}\Omega \tag{17.6}$$

However, from Eq. (17.3) it is noted that ϕ is approximated in terms of the nodal parameters $\mathbf{a}$. On substitution of this approximation we have

$$\bar{\mathbf{f}}=-\int_\Omega \mathbf{N}^{\mathrm{T}}Q\,\mathrm{d}\Omega+\left(\int_\Omega \mathbf{N}^{\mathrm{T}}\mu\mathbf{N}\,\mathrm{d}\Omega\right)\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t}+\left(\int_\Omega \mathbf{N}^{\mathrm{T}}\rho\mathbf{N}\,\mathrm{d}\Omega\right)\frac{d^2\mathbf{a}}{\mathrm{d}t^2} \tag{17.7}$$

and on expanding Eq. (17.4) in its final assembled form we get the following *matrix differential equation*:

$$\mathbf{M}\ddot{\mathbf{a}}+\mathbf{C}\dot{\mathbf{a}}+\mathbf{K}\mathbf{a}+\mathbf{f}=\mathbf{0} \tag{17.8}$$

$$\dot{\mathbf{a}}\equiv\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t}\qquad \ddot{\mathbf{a}}\equiv\frac{\mathrm{d}^2\mathbf{a}}{\mathrm{d}t^2} \tag{17.9}$$

† We have replaced the matrix $\mathbf{H}$ of Chapter 7 by $\mathbf{K}$ to facilitate comparison with other transient equations.

in which all the matrices are assembled from element submatrices in the standard manner with submatrices $\mathbf{K}^e$ and $\mathbf{f}^e$ still given by relations (7) in Chapter 7 and

$$C_{ij}^e = \int_\Omega N_i \mu N_j \, d\Omega \tag{17.10}$$

$$M_{ij}^e = \int_\Omega N_i \rho N_j \, d\Omega \tag{17.11}$$

Once again these matrices are symmetric as seen from the above relations.

Boundary conditions imposed at any time instant are treated in the standard manner.

The variety of physical problems governed by Eq. (17.1) is so large that a comprehensive discussion of them is beyond the scope of this book. A few typical examples will, however, be quoted.

Equation (17.1) with $\rho = 0$

This is the standard *transient heat conduction equation*[1,2] which has been discussed in the finite element context by several authors.[3–6] This same equation is applicable in other physical situations – one of these being the *soil consolidation equations*[7] associated with *transient seepage forms*.[8]

Equation (17.1) with $\mu = 0$

Now the relationship becomes the famous *Helmholz wave equation* governing a wide range of physical phenomena. Electromagnetic waves,[9] fluid surface waves[10] and compression waves[11] are but a few cases to which the finite element process has been applied.

Equation (17.1) with $\mu \neq \rho \neq 0$

This damped wave equation is of yet more general applicability and has particular significance in fluid mechanics (wave) problems.

The reader will recognize that what we have done here is simply an application of the process of partial discretization described in Sec. 3.5. It is convenient, however, to perform the operations in the manner suggested above as all the matrices and discretization expressions obtained from steady-state analysis are immediately available.

17.2.2 Dynamic behaviour of elastic structures with linear damping

While in the previous section we have been concerned with, apparently, a purely mathematical problem, identical reasoning can be applied directly to the wide class of dynamic behaviour of elastic structures following precisely the general lines of Chapter 2.

When displacements of an elastic body vary with time two sets of additional forces are called into play. The first is the inertia force, which for an acceleration characterized by $\ddot{\mathbf{u}}$ can be replaced by its static equivalent

$$-\rho\ddot{\mathbf{u}}$$

using the well-known d'Alembert principle. This is a force with components in directions identical to those of the displacement $\mathbf{u}$ and (generally) given per unit of volume. In this context ρ is simply the mass per unit volume.

The second force is that due to (frictional) resistances opposing the motion. These may be due to microstructure movements, air resistance, etc., and are often related in a non-linear way to the velocity $\dot{\mathbf{u}}$. For simplicity of treatment, however, only a linear viscous-type resistance will be considered, resulting again in unit volume forces in an equivalent static problem of magnitude

$$-\boldsymbol{\mu}\dot{\mathbf{u}}$$

In the above $\boldsymbol{\mu}$ is a set of viscosity parameters which can presumably be given numerical values.[12]

The equivalent static problem, at any instant of time, is now discretized precisely in the manner of Chapter 2, but replacing the distributed body force $\mathbf{b}$ by its equivalent

$$\bar{\mathbf{b}} - \rho\ddot{\mathbf{u}} - \boldsymbol{\mu}\dot{\mathbf{u}}$$

The element (nodal) forces given by Eq. (2.13) now become (excluding initial stress and strain contributions)

$$\mathbf{f}^e = -\int_{\Omega^e} \mathbf{N}^{\mathrm{T}}\mathbf{b}\,\mathrm{d}\Omega = -\int_{\Omega^e} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{b}}\,\mathrm{d}\Omega + \int_{\Omega^e} \mathbf{N}^{\mathrm{T}}\rho\ddot{\mathbf{u}}\,\mathrm{d}\Omega + \int_{\Omega^e} \mathbf{N}^{\mathrm{T}}\boldsymbol{\mu}\dot{\mathbf{u}}\,\mathrm{d}\Omega \tag{17.12}$$

in which the first force is that due to an external distributed body load and need not be considered further.

Substituting Eq. (17.12) into the general equilibrium equations we obtain finally, on assembly, the following matrix differential equation:

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{17.13}$$

in which $\mathbf{K}$ and $\mathbf{f}$ are assembled stiffness and force matrices obtained by the usual addition of element stiffness coefficients and of element forces due to external specified loads, initial stresses, etc., in the manner fully described before. The new matrices $\mathbf{C}$ and $\mathbf{M}$ are assembled by the usual rule from element submatrices given by†

$$\mathbf{C}^e_{ij} = \int_{\Omega^e} \mathbf{N}_i^{\mathrm{T}}\boldsymbol{\mu}\mathbf{N}_j\,\mathrm{d}\Omega \tag{17.14}$$

and

$$\mathbf{M}^e_{ij} = \int_{\Omega^e} \mathbf{N}_i^{\mathrm{T}}\rho\mathbf{N}_j\,\mathrm{d}\Omega \tag{17.15}$$

The matrix $\mathbf{M}^e$ is known as the *element mass matrix* and the assembled matrix $\mathbf{M}$ as the system mass matrix. Similarly, the matrix $\mathbf{C}^e$ is known as the *element damping matrix* and the assembled matrix $\mathbf{C}$ as the system damping matrix.

It is of interest to note that in early attempts to deal with dynamic problems of his nature the mass of the elements was usually arbitrarily 'lumped' at nodes, always resulting in a diagonal matrix even if no actual concentrated masses existed. The

† For simplicity we shall only consider *distributed* inertia – concentrated mass and damping forces being a limiting case.

fact that such a procedure was, in fact, unnecessary and apparently inconsistent was simultaneously recognized by Archer[13] and independently by Leckie and Lindberg[14] in 1963. The general presentation of the results given in Eq. (17.15) is due to Zienkiewicz and Cheung.[15] The name *consistent mass matrix* has been coined for the mass matrix defined here, a term which may be considered to be unnecessary since it is the logical and natural consequence of the discretization process. By analogy the matrices $\mathbf{C}^e$ and $\mathbf{C}$ may be called *consistent damping matrices*.

For many computational processes the lumped mass matrix is, however, more convenient and economical. Many practitioners are today using such matrices exclusively – sometimes showing good accuracy. While with simple elements a physically obvious methodology of lumping is easy to devise, this is not the case with higher order elements and we shall return to the process of 'lumping' later.

Determination of the damping matrix $\mathbf{C}$ is in practice difficult as knowledge of the viscous matrix $\boldsymbol{\mu}$ is lacking. It is often assumed, therefore, that the damping matrix is a linear combination of stiffness and mass matrices, i.e.,

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{17.16}$$

Here the parameters α and β are determined experimentally.[12,16] Such damping is known as 'Rayleigh damping' and has certain mathematical advantages which we shall discuss later. On occasion $\mathbf{C}$ may be completely specified and such approximation devices are not necessary.

It is perhaps worth recognizing that on occasion different shape functions need to be used to describe the inertia forces from those specifying the displacements $\mathbf{u}$. For instance, in beams (Chapter 2) (also for plates considered in Chapter 4 of Volume 2) the full strain state is prescribed simply by defining w, the lateral displacement, as additional bending assumptions are introduced. When considering the intertia forces it may be desirable not only to include the simple lateral inertia force given by

$$-\rho A \frac{\partial^2 w}{\partial t^2}$$

(in which ρA is now the mass per unit length of the beam) but also to consider *rotary inertia couples* of the type

$$-\rho I \frac{\partial^2}{\partial t^2}\left(\frac{\partial w}{\partial x}\right)$$

in which ρI is the rotatory inertia. Now it will be necessary to describe a more generalized displacement $\bar{\mathbf{u}}$:

$$\bar{\mathbf{u}} = \begin{Bmatrix} w \\ \dfrac{\partial w}{\partial x} \end{Bmatrix} = \bar{\mathbf{N}}\mathbf{a}^e$$

in which $\bar{\mathbf{N}}$ will follow directly from the definition of $\mathbf{N}$ which specifies only the w component. Relations such as Eq. (17.15) are still valid, providing we replace $\mathbf{N}$ by $\bar{\mathbf{N}}$ and put in place of ρ the matrix

$$\begin{bmatrix} \rho A & 0 \\ 0 & \rho I \end{bmatrix}$$

17.2.3 'Mass' or 'damping' matrices for some typical elements

It is impractical to present in an explicit form all the mass matrices for the various elements discussed in previous chapters. Some selected examples only will be discussed here.

Plane stress and plane strain

Using triangular elements discussed in Chapter 4 the matrix $\mathbf{N}^e$ is defined as

$$\mathbf{N}^e = \begin{bmatrix} \mathbf{N}_i & \mathbf{N}_j & \mathbf{N}_k \end{bmatrix}$$

where

$$\mathbf{N}_i^e = N_i \mathbf{I} \quad \text{etc.}$$

and

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Equation (4.8) gives the shape functions as

$$N_i = \frac{a_i + b_i x + c_i y}{2\Delta}, \quad \text{etc.}$$

where Δ is the area of the triangular element.

If the thickness of the element is h and this is assumed to be constant within the element, we have, for the mass matrix, Eq. (17.15),

$$\mathbf{M}^e = \rho h \int\int \mathbf{N}^{\mathrm{T}} \mathbf{N} \, \mathrm{d}x \, \mathrm{d}y$$

or

$$\mathbf{M}_{rs}^e = \rho h \mathbf{I} \int\int N_r N_s \, \mathrm{d}x \, \mathrm{d}y$$

If the relationships of Eq. (4.8) are substituted, it is easy to verify that

$$\int\int N_r N_s \, \mathrm{d}x \, \mathrm{d}x = \begin{cases} \frac{1}{12}\Delta & \text{when} \quad r \neq s \\ \frac{1}{6}\Delta & \text{when} \quad r = s \end{cases} \tag{17.17}$$

Thus taking the total mass of the element as

$$m = \rho h \Delta$$

the mass matrix becomes

$$\mathbf{M}^e = \frac{m}{12} \left[\begin{array}{cc|cc|cc} 2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{array}\right] \tag{17.18}$$

If the mass is lumped at the nodes in three equal parts the 'lumped' mass matrix contributed by the element is

$$\mathbf{M}^e = \frac{m}{3}\left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{17.19}$$

Certainly both matrices differ considerably and yet in applications the results of the analysis are almost identical.

17.2.4 Mass 'lumping' or diagonalization

We have referred to the computational convenience of lumping of mass matrices and presenting these in diagonal form. On some occasions such lumping is physically obvious (see the linear triangle for instance), in others this is not the case and a 'rational' procedure is required. For matrices of the type given in Eq. (17.15) several alternative approximations have been developed as discussed in Appendix I. In all of these the essential requirement of mass preservation is satisfied, i.e.,

$$\sum_i \tilde{M}_{ii} = \int_\Omega \rho \, \mathrm{d}\Omega \tag{17.20}$$

where $\tilde{M}_{ii}$ is the diagonal of the lumped mass matrix $\tilde{\mathbf{M}}$.

Three main procedures exist (see Fig. 17.1):

1. the row sum method in which

$$\tilde{M}_{ii} = \sum_j M_{ij}$$

2. diagonal scaling in which

$$\tilde{M}_{ii} = aM_{ii}$$

   with a adjusted so that Eq. (17.20) is satisfied,[17,18] and
3. evaluation of M using a quadrature involving only the nodal points and thus automatically yielding a diagonal matrix for standard finite element shape functions[19,20] in which $N_i = 0$ for $\mathbf{x} = \mathbf{x}_j$, $j \neq i$.

It should be remarked that Eq. (17.20) does not hold for hierarchical shape functions where no lumping procedure appears satisfactory.

The quadrature (numerical integration) process is mathematically most appealing but frequently leads to negative or zero lumped masses. Such a loss of positive definiteness is undesirable in some solution processes and cancels out the advantages of lumping. In Fig. 17.1 we show the effect of various lumping procedures on

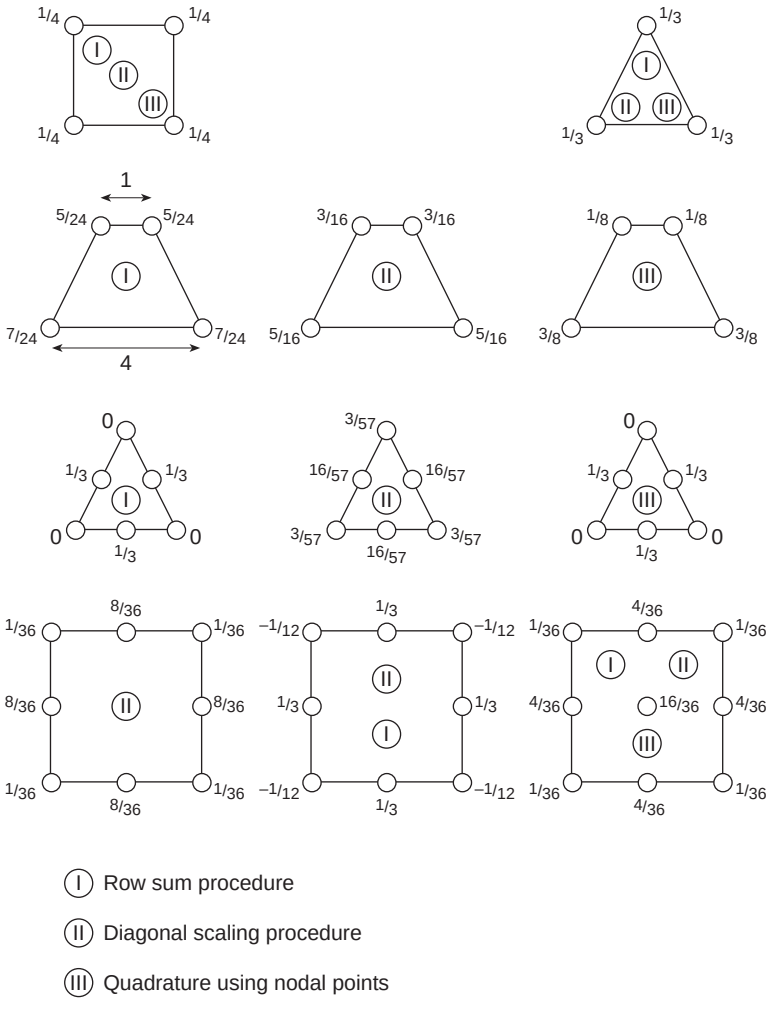


Fig. 17.1 Mass lumping for some two-dimensional elements.

triangular and quadrilateral elements of linear and quadratic type. It is clear from these that the optimal choice to lump the mass is by no means unique.

In general we would recommend the use of lumped matrices only as a convenient numerical device generally paid for by some loss of accuracy. An exception to this is for 'explicit' time integration of dynamics problems where the considerable efficiency of their use more than compensates for any loss in accuracy (see Chapter 18). In some problems of fluid mechanics (Volume 3) we shall indeed use lumping for an intermediate iterative step in getting the consistent solution. However, we note that it has occasionally been shown that lumping can *improve* accuracy of some problem by error cancellation. It can be shown that in the transient approximation the lumping process introduces additional dissipation of the 'stiffness matrix' form and this can help in cancelling out numerical oscillation.

To demonstrate the nature of lumped and consistent mass matrices it is convenient to consider a typical one-dimensional problem specified by the equation

$$\frac{\partial \phi}{\partial t} - \frac{\partial}{\partial x}\left[\mu \frac{\partial}{\partial x}\left(\frac{\partial \phi}{\partial t}\right)\right] - \frac{\partial}{\partial x}\left(k \frac{\partial \phi}{\partial x}\right)$$

Semi-discretization here gives a typical nodal equation i as

$$\left(M_{ij} + H_{ij}\right)\dot{a}_j + K_{ij} a_j = 0$$

where

$$M_{ij} = \int_\Omega N_i N_j \, \mathrm{d}x$$

$$H_{ij} = \int_\Omega \frac{\mathrm{d}N_i}{\mathrm{d}x} \mu \frac{\mathrm{d}N_j}{\mathrm{d}x\,\mathrm{d}x}$$

$$K_{ij} = \int_\Omega \frac{\mathrm{d}N_i}{\mathrm{d}x} k \frac{\mathrm{d}N_j}{\mathrm{d}x\,\mathrm{d}x}$$

and it is observed that **H** and **K** have identical structure. With linear elements of constant size h the approximating equation at a typical node i (and surrounding nodes $i-1$ or $i+1$) can be written as follows (as the reader can readily verify).

$$M_{ij}\dot{a}_j \equiv \frac{h}{6}(\dot{a}_{i-1} + 4\dot{a}_i + \dot{a}_{i+1})$$

$$H_{ij}\dot{a}_j \equiv \frac{\mu}{h}(-\dot{a}_{i-1} + 2\dot{a}_i - \dot{a}_{i+1})$$

$$K_{ij} a_j \equiv \frac{k}{h}(-a_{i-1} + 2a_i - a_{i+1})$$

If a lumped approximation is used for **M**, that is $\tilde{\mathbf{M}}$, we have, simply by adding coefficients using the row sum method,

$$\tilde{M}_{ij}\dot{a}_j = h\dot{a}_i$$

The difference between the two expressions is

$$\tilde{M}_{ij}\dot{a}_j - M_{ij}\dot{a}_j \equiv \frac{h}{6}(-\dot{a}_{i-1} + 2\dot{a}_i - \dot{a}_{i+1})$$

and is clearly identical to that which would be obtained by increasing μ by $h^2/6$. As μ in the above example can be considered as a viscous dissipation we note that the effect of using a lumped matrix is that of adding an extra amount of such viscosity and can often result in smoother (though probably less accurate) solutions.

Eigenvalues and analytical solution procedures

17.3 General classification

We have seen that as a result of semi-discretization many time-dependent problems can be reduced to a system of ordinary differential equations of the characteristic

form given by

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{17.21}$$

In this, in general, all the matrices are symmetric (some cases involving non-symmetric matrices will be discussed in Volume 3, Chapter 2). This second-order system often becomes first order if $\mathbf{M}$ is zero as, for instance, in transient heat conduction problems. We shall now discuss some methods of solution of such ordinary differential equation systems. In general, the above equations can be non-linear (if, for instance, stiffness matrices are dependent on non-linear material properties or if large deformations are involved) but here we shall concentrate on linear cases only.

Systems of ordinary linear differential equations can always in principle be solved analytically without the introduction of additional approximations. The remainder of this chapter will be concerned with such analytical processes. While such solutions are possible they may be so complex that further recourse has to be taken to the process of approximation; we shall deal with this matter in the next chapter. The analytical approach provides, however, an insight into the behaviour of the system which the authors always find helpful.

Some of the matter in this chapter will be an extension of standard well-known procedures used for the solution of differential equations with constant coefficients that are encountered in most studies of dynamics or mathematics. In the following we shall deal successively with:

1. determination of free response ($\mathbf{f} = \mathbf{0}$)
2. determination of periodic response ($\mathbf{f}(t)$ periodic)
3. determination of transient response ($\mathbf{f}(t)$ arbitrary).

In the first two, initial conditions of the system are of no importance and a general solution is simply sought. The last, most important, phase presents a problem to which considerable attention will be devoted.

17.4 Free response – eigenvalues for second-order problems and dynamic vibration

17.4.1 Free dynamic vibration – real eigenvalues

If no damping or forcing terms exist in the dynamic problem of Eq. (17.21) it reduces to

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{K}\mathbf{a} = \mathbf{0} \tag{17.22}$$

A general solution of such an equation may be written as

$$\mathbf{a} = \bar{\mathbf{a}} \exp(\mathrm{i}\omega t)$$

the real part of which simply represents a harmonic response as $\exp(\mathrm{i}\omega t) \equiv \cos \omega t + \mathrm{i} \sin \omega t$. Then on substitution we find that ω can be determined from

$$(-\omega^2 \mathbf{M} + \mathbf{K})\bar{\mathbf{a}} = \mathbf{0} \tag{17.23}$$

This is a *general linear eigenvalue* or *characteristic value problem* and for non-zero solutions the determinant of the above coefficient matrix must be zero:

$$|-\omega^2\mathbf{M} + \mathbf{K}| = 0 \tag{17.24}$$

Such a determinant will in general give n values of ω^2 (or ω_j, $j = 1, 2, \ldots, n$) when the size of the matrices $\mathbf{K}$ and $\mathbf{M}$ is $n \times n$, providing the matrices $\mathbf{K}$ and $\mathbf{M}$ are symmetric positive definite.†

While the solution of Eq. (17.24) cannot determine the actual values of $\mathbf{a}$ we can find n vectors $\bar{\mathbf{a}}_j$ that give the proportions for the various terms. Such vectors are known as the *normal modes of the system* or *eigenvectors* and are made unique by normalizing so that

$$\bar{\mathbf{a}}_j^{\mathrm{T}}\mathbf{M}\bar{\mathbf{a}}_j = 1; \quad j = 1, 2, \ldots, n \tag{17.25}$$

At this stage it is useful to note the property of *modal orthogonality*, i.e., that

$$\bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{M}\bar{\mathbf{a}}_j = 0; \quad (i \neq j) \tag{17.26}$$

$$\bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{K}\bar{\mathbf{a}}_j = 0; \quad (i \neq j) \tag{17.27}$$

The proof of the above statement is simple. As Eq. (17.23) is valid for any mode we can write

$$\omega_i^2\mathbf{M}\bar{\mathbf{a}}_i = \mathbf{K}\bar{\mathbf{a}}_i$$

$$\omega_j^2\mathbf{M}\bar{\mathbf{a}}_j = \mathbf{K}\bar{\mathbf{a}}_j$$

Premultiplying the first by $\bar{\mathbf{a}}_j^{\mathrm{T}}$ and the second by $\bar{\mathbf{a}}_i^{\mathrm{T}}$ and noting the symmetry of $\mathbf{M}$ and $\mathbf{K}$ so that

$$\bar{\mathbf{a}}_j^{\mathrm{T}}\mathbf{M}\bar{\mathbf{a}}_i = \bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{M}\bar{\mathbf{a}}_j$$

$$\bar{\mathbf{a}}_j^{\mathrm{T}}\mathbf{K}\bar{\mathbf{a}}_i = \bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{K}\bar{\mathbf{a}}_j$$

the difference becomes

$$(\omega_i^2 - \omega_j^2)\bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{M}\bar{\mathbf{a}}_j = 0$$

and if $\omega_i \neq \omega_j$ ‡ the orthogonality condition for the matrix $\mathbf{M}$ has been proved. From this the orthogonality of the vectors with $\mathbf{K}$ follows immediately. The final condition

$$\bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{K}\bar{\mathbf{a}}_i = \omega^2$$

follows from Eq. (17.25) and a premultiplication of Eq. (17.23) for equation i by $\bar{\mathbf{a}}_i$.

17.4.2 Determination of eigenvalues

To find the actual eigenvalues it is seldom practicable to write the polynomial expanding the determinant given in Eq. (17.24) and alternative techniques have to

† A symmetric matrix is positive definite if all the diagonals of the triangular factors are positive, this is a usual case with structural problems – all roots of Eq. (17.24) are real positive numbers (for a proof see reference 1). These are known as the natural frequencies of the system. If only the $\mathbf{M}$ matrix is symmetric positive definite while $\mathbf{K}$ is symmetric positive semidefinite the roots are real and positive or zero.

‡ For any case where repeated frequencies occur we merely enforce the orthogonality by construction.

be developed. The discussion of such techniques is best left to specialist texts and indeed many standard computer programs exist as library routines.

Many extremely efficient procedures are available and the reader can find some interesting matter in references.[21–27]

In some processes the starting point is the *standard eigenvalue problem* given by

$$\mathbf{H}\mathbf{x} = \lambda\mathbf{x} \tag{17.28}$$

in which $\mathbf{H}$ is a symmetric matrix and hence has real eigenvalues. Equation (17.23) can be written as

$$\mathbf{M}^{-1}\mathbf{K}\bar{\mathbf{a}} = \omega^2\bar{\mathbf{a}} \tag{17.29}$$

on inverting $\mathbf{M}$ with $\lambda = \omega^2$, but symmetry is in general lost.

If, however, we write in triangular form

$$\mathbf{M} = \mathbf{L}\mathbf{L}^{\mathrm{T}} \qquad \text{and} \qquad \mathbf{M}^{-1} = \mathbf{L}^{-\mathrm{T}}\mathbf{L}^{-1}$$

in which $\mathbf{L}$ is a lower triangular matrix (i.e., has all zero coefficients above the diagonal), Eq. (17.26) may now be written as

$$\mathbf{K}\bar{\mathbf{a}} = \omega^2\mathbf{L}\mathbf{L}^{\mathrm{T}}\bar{\mathbf{a}}$$

Calling

$$\mathbf{L}\bar{\mathbf{a}} = \mathbf{x} \tag{17.30}$$

and multiplying by $\mathbf{L}^{-1}$ we have finally

$$\mathbf{H}\mathbf{x} = \omega^2\mathbf{x} \tag{17.31}$$

in which

$$\mathbf{H} = \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\mathrm{T}} \tag{17.32}$$

which is of the standard form of Eq. (17.30), as $\mathbf{H}$ is now symmetric.

Having determined ω^2 (all, or only a few of the selected smallest values corresponding to fundamental periods) the modes of $\mathbf{x}$ are found, and hence by use of Eq. (17.30) the modes of $\bar{\mathbf{a}}$.

If the matrix $\mathbf{M}$ is diagonal – as it will be if the masses have been 'lumped' – the procedure of deriving the standard eigenvalue problem is simplified and here appears the first advantage of the diagonalization, which we have discussed in Sec. 17.2.4.

17.4.3 Free vibration with the singular K matrix

In static problems we have always introduced a suitable number of *support* conditions to allow the stiffness matrix $\mathbf{K}$ to be inverted, or what is equivalent to solve the static equations uniquely. If such 'support' conditions are in fact not specified, as may well be the case with a rocket travelling in space, the arbitrary fixing of a minimum number of support conditions allows a static solution to be obtained without affecting the stresses. In dynamic situations such a fixing is not permissible and frequently one is faced with the problem of a free oscillation for which $\mathbf{K}$ is singular and therefore does not possess unique triangular factors or an inverse.

To preserve the applicability of methods which require an inverse (e.g., methods based on inverse power iteration[26]) a simple artifice is possible. Equation (17.23) is modified to

$$[(\mathbf{K} + \alpha\mathbf{M}) - (\omega^2 + \alpha)\mathbf{M}]\bar{\mathbf{a}} = \mathbf{0} \tag{17.33}$$

in which α is an arbitrary constant of the same order as the typical ω_2 sought. The new matrix $(\mathbf{K} + \alpha\mathbf{M})$ is no longer singular and can be factored (or inverted) for use in the standard eigensolution procedure to find $(\omega^2 + \alpha)$.

This simple but effective avoidance of an otherwise serious difficulty was first suggested by Cox[28] and Jennings.[29] Alternative methods of dealing with the above problem are given in references 30 and 31.

17.4.4 Reduction of the eigenvalue system

Independent of which technique is used to determine the eigenpairs of the system (17.23), the effort for $n \times n$ matrices is at least one order greater than that involved in an equivalent static situation. Further, while the number of eigenvalues of the real system is infinite, in practice, we are generally interested only in a relatively small number of the lower frequencies and it is possible to simplify the computation by reducing the size of the problem.

To achieve a reduced problem we assume that the unknown $\bar{\mathbf{a}}$ can be expressed in terms of m ($\ll n$) vectors $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_m$ and corresponding participating factors x_i. We now write

$$\bar{\mathbf{a}} = \mathbf{t}_1 x_1 + \mathbf{t}_2 x_2 + \cdots + \mathbf{t}_m x_m = \mathbf{T}\mathbf{x} \tag{17.34}$$

Inserting Eq. (17.34) into Eq. (17.23) and premultiplying by $\mathbf{T}^{\mathrm{T}}$ we have a reduced problem with only m eigenpairs:

$$(\omega^*)^2\mathbf{M}^*\mathbf{x} = \mathbf{K}^*\mathbf{x} \tag{17.35}$$

where

$$\mathbf{M}^* = \mathbf{T}^{\mathrm{T}}\mathbf{M}\mathbf{T} \qquad \mathbf{K}^* = \mathbf{T}^{\mathrm{T}}\mathbf{K}\mathbf{T}$$

and ω^* are now eigenvalues of the *reduced* system, which for the appropriate choice of the $\mathbf{t}_i$ vectors can be good approximations to the eigenvalues of the original system.

If by good fortune the trial vectors were to be chosen as eigenvectors of the original matrix the system would become diagonal and all eigenvalues (i.e., in this case $\omega^* = \omega$) could be determined by a trivial calculation. This indeed is what some iterative eigenproblem strategies attempt (e.g., subspace or Lanczos methods[26,32]). It is also of course possible by physical insight to find vectors $\mathbf{t}$ that correspond closely to the principal modes of the movement (e.g., see reference 33).

17.4.5 Some examples

There are a variety of problems for which practical solutions exist, so only a few simple examples will be shown.

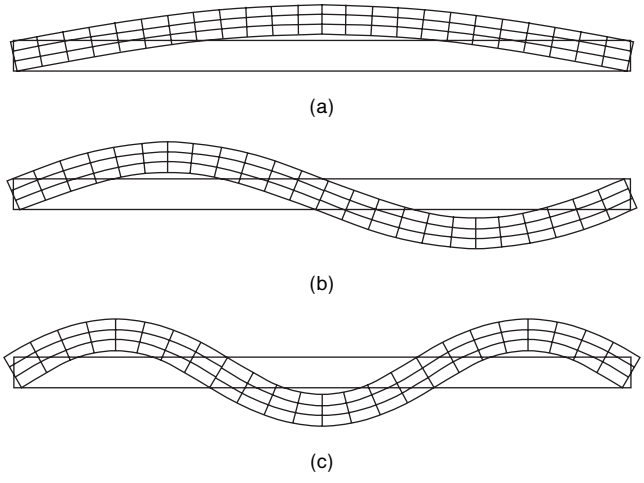


Fig. 17.2 Simply supported beam: (a) $\omega_1 = 3.8050$; (b) $\omega_2 = 59.2236$; (c) $\omega_3 = 290.0804$.

Vibration of a simply supported beam

Figure 17.2 shows the first three vibration modes of a simply supported beam with length 40 and rectangular cross-section of width 1 and depth 2 units. The elastic properties are $E = 30\,000$, $\nu = 0$, and $\rho = 0.1$ units. The beam is modelled using 9-noded quadrilateral elements of lagrangian type with the central node at the left end restrained in the x and y direction and the central node at the right end restrained only in the y direction. The problem is also solved using a mesh with 1000 two-noded beam elements which include effects of transverse shearing deformation. In Table 17.1 we present the values for the first three frequencies obtained from the finite element analysis and compare to the value obtained from an exact solution for the beam without shear deformation.

Vibration of an earth dam

Figure 17.3 shows the vibration of a two-dimensional earth dam resting on a rigid foundation. The earth dam is modelled by linear triangular elements and includes the effects of different material layers.

The 'wave' equation. Electromagnetic and fluid problems

The basic dynamic equation (17.8) can be derived for a variety of non-structural problems. The eigenvalue problem once again occurs with 'stiffness' and 'mass' matrices now having alternate physical meanings.

Table 17.1 Frequencies for a simply supported beam

9-noded element	3.7785	59.2236	290.0804
2-noded element	3.7787	59.2338	290.1774
Beam theory	3.8050	60.8807	308.2080

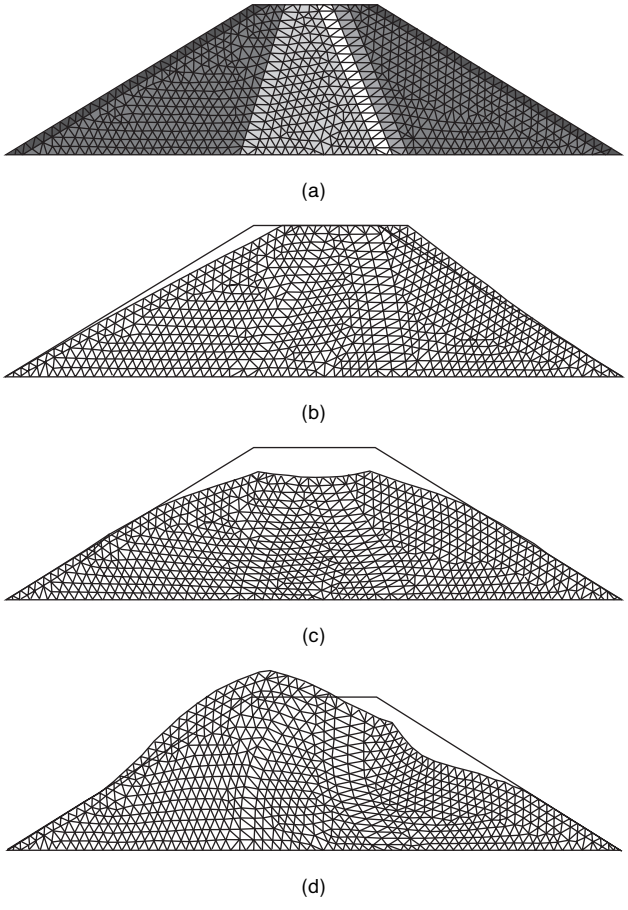


Fig. 17.3 (a) Mesh showing layers considered; (b) earth dam, $\omega_1 = 3.8050$; (c) earth dam, $\omega_2 = 59.2236$; (d) earth dam, $\omega_3 = 290.0804$.

A particular form of the more general equations discussed earlier is the well-known Helmholz wave equation which, in two-dimensional form, is

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{1}{\bar{c}^2} \frac{\partial^2 \phi}{\partial t^2} = 0 \tag{17.36}$$

If the boundary conditions do not force a response, an eigenvalue problem results which has significance in several fields of physical science.

The first application is to *electromagnetic fields*.[9] Figure 17.4 shows a modal shape of a field for a *waveguide problem*. Simple linear triangular elements are used here. More complex three-dimensional oscillations are also discussed in reference 9.

A similar equation also describes to a reasonable approximation the behaviour of shallow water waves in a body of water:

$$\frac{\partial}{\partial x}\left(h \frac{\partial \psi}{\partial x}\right) + \frac{\partial}{\partial y}\left(h \frac{\partial \psi}{\partial y}\right) + \frac{1}{g} \frac{\partial^2 \psi}{\partial t^2} = 0 \tag{17.37}$$

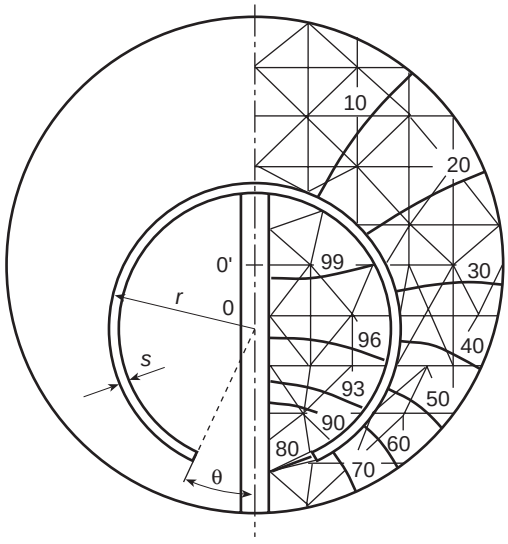


Fig. 17.4 A 'lunar' waveguide;[9] mode of vibration for electromagnetic field. Outer diameter $= d$, $OO' = 0.13d$, $r = 0.29d$, $S = 0.055d$, $\theta = 22^\circ$.

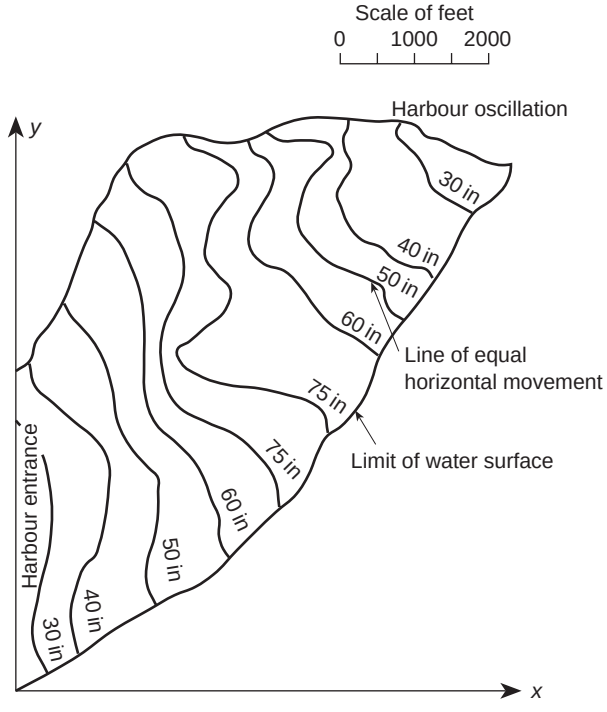


Fig. 17.5 Oscillations of a natural harbour: contours of velocity amplitudes.[10]

in which h is the average water depth, ψ the surface elevation above average and g the gravity acceleration. The formulation of the last two problems are discussed in detail in Volume 3, Chapter 8.

Thus natural frequencies of bodies of water contained in harbours of varying depths may easily be found.[10] Figure 17.5 shows the modal shape for a particular harbour.

17.5 Free response – eigenvalues for first-order problems and heat conduction, etc.

If in Eq. (17.21) $\mathbf{M} = \mathbf{0}$, we have a form typical of the transient heat conduction equation [see Eq. (17.1)]. For free response we seek a solution of the homogeneous equation

$$\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} = \mathbf{0} \tag{17.38}$$

Once again an exponential form can be used:

$$\mathbf{a} = \bar{\mathbf{a}} \exp(-\lambda t)$$

Substituting we have

$$(-\lambda \mathbf{C} + \mathbf{K})\bar{\mathbf{a}} = \mathbf{0} \tag{17.39}$$

which again gives an eigenvalue problem identical to that of Eq. (17.23). As $\mathbf{C}$ and $\mathbf{K}$ are usually positive definite, λ will be positive and real. The solution therefore represents simply an exponential decay term and is not really steady state. Combination of such terms, however, can be useful in the solution of initial value transient problems but is of little value *per se*.

17.6 Free response – damped dynamic eigenvalues

We shall now consider the full equation (17.21) for free response conditions. Writing

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} = \mathbf{0} \tag{17.40}$$

and substituting

$$\mathbf{a} = \bar{\mathbf{a}} \exp(\alpha t) \tag{17.41}$$

we have the characteristic equation

$$(\alpha^2 \mathbf{M} + \alpha \mathbf{C} + \mathbf{K})\bar{\mathbf{a}} = \mathbf{0} \tag{17.42}$$

where α and $\bar{\mathbf{a}}$ will in general be found to be complex. The real part of the solution represents a decaying vibration.

The eigenvalue problem involved in Eq. (17.41) is more difficult than that arising in the previous sections. In solutions to date the problem is usually solved by splitting Eq. (17.40) into two first-order equations. This is accomplished by defining

$$\dot{\mathbf{a}} = \mathbf{v}$$

and writing the split form as

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{a}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{K} \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{17.43}$$

Now substituting

$$\mathbf{a} = \bar{\mathbf{a}} \exp(\alpha t) \qquad \mathbf{v} = \bar{\mathbf{v}} \exp(\alpha t)$$

gives the general linear eigenproblem

$$\left(\alpha \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M} \end{bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{K} \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \right) \begin{Bmatrix} \bar{\mathbf{v}} \\ \bar{\mathbf{a}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{17.44}$$

This form has been studied by Chen *et al.*[34–36] Similar to the first-order problem, no steady-state solution exists and once more the concept of eigenvalues of the above kind is generally of importance only in modal analysis, as we shall see later.

17.7 Forced periodic response

If the forcing term in Eq. (17.21) is periodic or, more generally, if we can express it as

$$\mathbf{f} = \bar{\mathbf{f}} \exp(\alpha t) \tag{17.45}$$

where α is complex, i.e.

$$\alpha = \alpha_1 + \mathrm{i}\,\alpha_2 \tag{17.46}$$

then a general solution can once more be written as

$$\mathbf{a} = \bar{\mathbf{a}} \exp(\alpha t) \tag{17.47}$$

Substituting the above in Eq. (17.21) gives

$$\left(\alpha^2 \mathbf{M} + \alpha \mathbf{C} + \mathbf{K}\right) \bar{\mathbf{a}} \equiv \bar{\mathbf{K}} \bar{\mathbf{a}} = -\bar{\mathbf{f}} \tag{17.48}$$

which is no longer an eigenvalue problem but can be solved formally by inverting the matrix $\bar{\mathbf{K}}$ as

$$\bar{\mathbf{a}} = -\bar{\mathbf{K}}^{-1} \bar{\mathbf{f}} \tag{17.49}$$

The solution is thus precisely of the same form as that used for static problems but now, however, has to be determined in terms of complex quantities. Computer programs are available for operation of complex numbers but computation can be arranged in real numbers directly, noting that

$$\begin{aligned} \exp(\alpha t) &= \exp(\alpha_1 t)[\cos \alpha_2 t + \mathrm{i} \sin \alpha_2 t] \\ \bar{\mathbf{f}} &= \bar{\mathbf{f}}_1 + \mathrm{i}\,\bar{\mathbf{f}}_2 \\ \bar{\mathbf{a}} &= \bar{\mathbf{a}}_1 + \mathrm{i}\,\bar{\mathbf{a}}_2 \end{aligned} \tag{17.50}$$

in which $\alpha_1, \alpha_2, \bar{\mathbf{f}}_1, \bar{\mathbf{f}}_2, \bar{\mathbf{a}}_1$ and $\bar{\mathbf{a}}_2$ are real quantities. Inserting the above into Eq. (17.48) we have

$$\begin{bmatrix} (\alpha_1^2 - \alpha_2^2)\mathbf{M} + \alpha_1 \mathbf{C} + \mathbf{K}, & -2\alpha_1 \alpha_2 \mathbf{M} - \alpha_2 \mathbf{C} \\ -2\alpha_1 \alpha_2 \mathbf{M} - \alpha_2 \mathbf{C}, & -(\alpha_1^2 - \alpha_2^2)\mathbf{M} - \alpha_1 \mathbf{C} - \mathbf{K} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{a}}_1 \\ \bar{\mathbf{a}}_2 \end{Bmatrix} = -\begin{Bmatrix} \bar{\mathbf{f}}_1 \\ -\bar{\mathbf{f}}_2 \end{Bmatrix} \tag{17.51}$$

Equations (17.51) form a system in which all quantities are real and from which the response to any periodic input can be determined by direct solution. The system is no longer positive definite although it has been written in a form which is still symmetric.

With periodic input the solution after an initial transient is not sensitive to the initial conditions and the above solution represents the finally established response. It is valid for problems of dynamic structural and fluid-structure responses as well as for problems typical of heat conduction in which we simply put $\mathbf{M} = \mathbf{0}$.

17.8 Transient response by analytical procedures

17.8.1 General

In the previous sections we have been concerned with steady-state general solutions which took no account of the initial conditions of the system or of the non-periodic form of the forcing terms. The response taking these features into account is essential if we consider, for instance, the earthquake behaviour of structures or the transient behaviour of the heat conduction problem. The solution of such general cases requires either a full-time discretization, which we shall discuss in detail in the next chapter, or the use of special analytical procedures. Here two broad possibilities exist:

1. the frequency response procedure
2. the modal analysis procedure.

We shall discuss these briefly.

17.8.2 Frequency response procedures

In Sec. 17.7 we have shown how the response of the system to any forcing terms of the general periodic type or in particular to a periodic forcing function

$$\mathbf{f} = \bar{\mathbf{f}} \exp(\mathrm{i}\,\omega t) \tag{17.52}$$

can be obtained by solving a simple equation system. As a completely arbitrary forcing function can be represented approximately by a Fourier series or in the limit, exactly, as a Fourier integral, the response to such an input can be obtained by a synthesis of a curve representing the response of any quantity of interest, e.g., the displacement at a particular point, etc., to all frequencies ranging from zero to infinity. In fact only a limited number of such forcing frequencies has to be considered and a result can be synthesized efficiently by fast Fourier transform techniques.[37] We shall not discuss the mathematical details for such procedures which can be found in standard texts on structural dynamics.[12,16]

The technique of frequency response is readily adapted to problems where the damping matrix $\mathbf{C}$ is of an arbitrary specified form. This is not the case with the more widely used modal decomposition procedures which are to be described in the next section.

By way of illustration we show in Fig. 17.6 the frequency response of an artificial harbour [see Eq. (17.37)] to an input of waves with different frequencies and damping due to the radiation of reflected waves which imposes a very particular form on the damping matrix. Details of this problem are given elsewhere[38,39] (see also Volume 3). Similar techniques are frequently used in the analysis for the foundation response of structures where radiation of energy occurs.[40]

17.8.3 Modal decomposition analysis

This procedure is probably the most important and widely used in practice. Further, it provides an insight into the behaviour of the whole system, which is of value where strictly numerical processes are used. We shall therefore describe it in detail in the context of the general problem of Eq. (17.21), i.e.,

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{17.53}$$

where $\mathbf{f}$ is an arbitrary function of time.

We have seen that the general solution for the free response is of the form

$$\mathbf{a} = \sum_{i=1}^{n} \bar{\mathbf{a}}_i \exp(\alpha_i t) \tag{17.54}$$

where α_i are the (complex) eigenvalues and $\bar{\mathbf{a}}_i$ are the (complex) eigenvectors (Sec. 17.6). For forced response we shall assume that the solution can be written in a linear combination of modes as

$$\mathbf{a} = \sum_{i=1}^{n} \bar{\mathbf{a}}_i y_i(t) = [\bar{\mathbf{a}}_1, \quad \bar{\mathbf{a}}_1, \quad \ldots]\mathbf{y}(t) \tag{17.55}$$

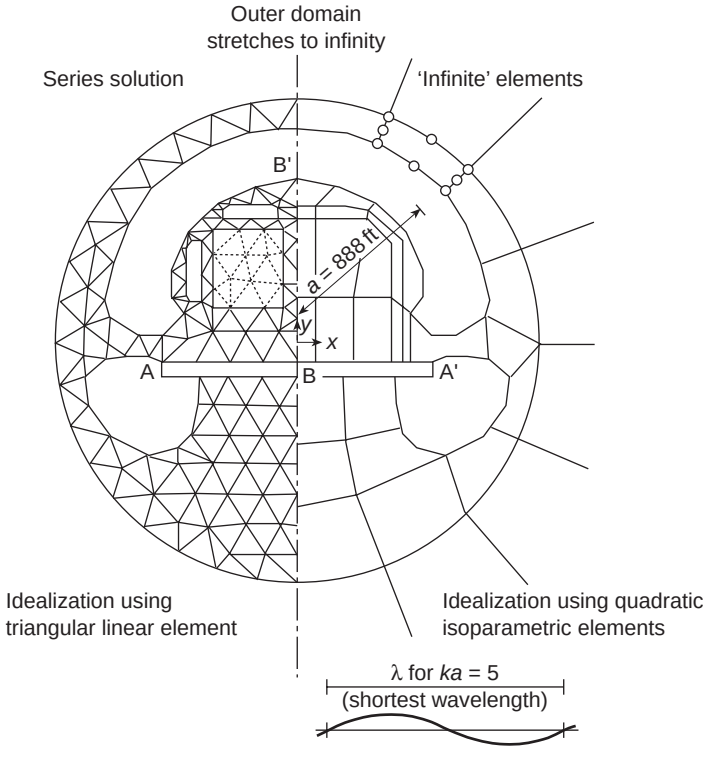
where the scalar mode participation factor y_i is now a function of time. This shows in a clear manner the proportions of each mode occurring. Such a decomposition of an arbitrary vector presents no restriction as all the modes are linearly independent vectors (with those for repeated frequencies being constructed to be linearly independent as mentioned in Sec. 17.4).

If expression (17.55) is substituted into Eq. (17.53) and the result is premultiplied by the complex conjugate transposed, $\bar{\mathbf{a}}_i^{\mathrm{T}}$ $(i = 1, \ldots, n)$, then the result is simply a set of scalar, independent, equations

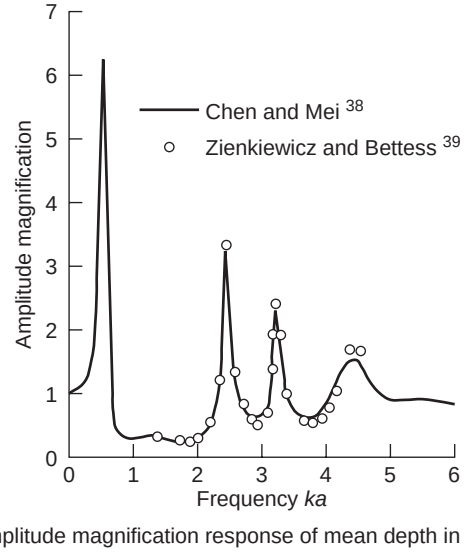
$$m_i \ddot{y}_i + c_i \dot{y}_i + k_i y_i + f_i = 0 \tag{17.56}$$

where

$$\begin{aligned} m_i &= \bar{\mathbf{a}}_i^{\mathrm{T}} \mathbf{M} \bar{\mathbf{a}}_i \\ c_i &= \bar{\mathbf{a}}_i^{\mathrm{T}} \mathbf{C} \bar{\mathbf{a}}_i \\ k_i &= \bar{\mathbf{a}}_i^{\mathrm{T}} \mathbf{K} \bar{\mathbf{a}}_i \\ f_i &= \bar{\mathbf{a}}_i^{\mathrm{T}} \mathbf{f} \end{aligned} \tag{17.57}$$



(a) Geometric details and FEM idealization.
Wave forcing frequency $\omega = k\sqrt{gh} = ka$, h = depth of water



(b) Amplitude magnification response of mean depth in harbour for various frequencies

Fig. 17.6 Frequency response of an artificial harbour to an input of periodic wave.

as for true eigenvectors $\bar{\mathbf{a}}_i$

$$\bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{M}\bar{\mathbf{a}}_j = \bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{C}\bar{\mathbf{a}}_j = \bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{K}\bar{\mathbf{a}}_j = 0 \tag{17.58}$$

when $i \neq j$ (this result was proved in Sec. 17.4 for real eigenpairs but is valid generally for complex pairs, as could be verified by the reader).

Each scalar equation of (17.56) can be solved by elementary procedures independently and the total vector of response obtained by superposition following Eq. (17.57). In the general case, as we have shown in Sec. 17.6, the eigenpairs are complex and their determination is not simple.[30] The more usual procedure is to use real eigenpairs corresponding to the solution of Eq. (17.22):

$$\mathbf{K}\bar{\mathbf{a}} = \omega^2\mathbf{M}\bar{\mathbf{a}} \tag{17.59}$$

Now repetition of procedures using the process described in Eqs (17.55)–(17.58) leads to decoupled equations with real variables $\mathbf{y}$ only if

$$\bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{C}\bar{\mathbf{a}}_j = 0; \qquad i \neq j \tag{17.60}$$

which generally does not occur as the eigenvectors now guarantee only orthogonality with $\mathbf{M}$ and $\mathbf{K}$ and not of the damping matrix. However, if the damping matrix $\mathbf{C}$ is of the form of Eq. (17.16), i.e., a linear combination of $\mathbf{M}$ and $\mathbf{K}$, such orthogonality will obviously occur. Unless the damping is of a definite form which requires special treatment, an assumption of orthogonality is made and Eq. (17.56) is assumed valid in terms of such eigenvectors.

From Eq. (17.59) we have

$$\mathbf{K}\bar{\mathbf{a}}_i = \omega_i^2\mathbf{M}\bar{\mathbf{a}}_i \tag{17.61}$$

and on premultiplying by $\bar{\mathbf{a}}_i^{\mathrm{T}}$ we obtain

$$k_i = \omega_i^2 m_i \tag{17.62}$$

Writing the modal damping in the form

$$c_i = 2\omega\xi_i \tag{17.63}$$

(where ξ_i represents the ratio of damping to its critical value) and assuming that the modes have been normalized so that $m_i = 1$ [see Eq. (17.25)], Eq. (17.56) can be rewritten in standard second order form:

$$\ddot{y}_i + 2\omega_i\xi_i\dot{y}_i + \omega_i^2 y_i + f_i = 0 \tag{17.64}$$

A general solution can then be obtained by writing

$$\begin{aligned} y_i = {} & \exp(-\xi_i\omega_i t)\left[\frac{\dot{y}_{i0} + \xi_i\omega_i y_{i0}}{\bar{\omega}}\sin\bar{\omega}_i t + y_{i0}\cos\bar{\omega}_i t\right] \\ & + \frac{1}{\bar{\omega}i}\int_0^t \exp(-\xi_i\omega_i[t-\tau])\sin\bar{\omega}_i(t-\tau)f_i(\tau)\,\mathrm{d}\tau \end{aligned} \tag{17.65}$$

in which $\bar{\omega}_i = \omega_i\sqrt{1-\xi_i^2}$ and y_{i0}, $\dot{y}_{i0}$ are initial conditions computed from

$$\begin{aligned} y_{i0} &= \bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{M}\mathbf{a}(0) \\ \dot{y}_{i0} &= \bar{\mathbf{a}}_i^{\mathrm{T}}\mathbf{M}\dot{\mathbf{a}}(0) \end{aligned} \tag{17.66}$$

The solution of Eq. (17.65) can be carried out by assuming the forcing function is given by linear interpolation between discrete time points t_k and then evaluating the resulting integrals exactly. Alternatively, a numerical solution can be carried out and the response obtained. In practice, often a single calculation is carried out for each mode to determine the maximum responses and a suitable addition of these results is used. Such processes are described in standard texts and are used as procedures to calculate the bounds on behaviour of structures subjected to seismic loading.[12,16,27]

17.8.4 Damping and participation of modes

The type of calculation implied in modal decomposition apparently necessitates the determination of all modes and eigenvalues, a task of considerable magnitude. In fact only a limited number of modes usually need to be taken into consideration as often the response to higher frequency is critically damped and insignificant.

To show that this is true consider the form of the damping matrices. In Sec. 17.2 [Eq. (17.16)] we have indicated that the damping matrix is often assumed as

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{17.67}$$

Indeed a form of this type is necessary for the use of modal decomposition, although other generalizations are possible.[41,42] From the definition of ξ_i, the critical damping ratio in Eq. (17.63), we see that this can now be written as

$$\xi_i = \frac{1}{2\omega_i}\bar{\mathbf{a}}_i^{\mathrm{T}}(\alpha\mathbf{M} + \beta\mathbf{K})\bar{\mathbf{a}}_i = \frac{1}{2\omega_i}\left(\alpha + \beta\omega_i^2\right) \tag{17.68}$$

Thus if the coefficient β is of greater importance, as is the case with most structural damping, ξ_i grows with ω_i and at high frequency an overdamped condition will arise.[12] This is indeed fortunate as, in general, an infinite number of high frequencies exist which are not modelled by any finite element discretization.

We shall see in the next chapter that in the step-by-step recurrence computation the high frequencies often control the problem, and this effect needs to be 'filtered out' for realistic results.

17.9 Symmetry and repeatability

In concluding this chapter it is worth remarking that in dynamic calculation we have once again encountered all the general principles of assembly, etc., that are applicable to static problems. However, some aspects of symmetry and repeatability which were used previously (see Sec. 9.18) need amending. It is obviously possible for symmetric structures to vibrate in an unsymmetrical manner, for instance, and similarly a repeatable structure contains modes which are themselves non-repeatable. However, even here considerable simplification can still be made; details of this are discussed by Williams,[43] Thomas[44] and Evensen.[45]

References

1. S. Crandall. *Engineering Analysis*. McGraw-Hill, New York, 1956.
2. H.S. Carslaw and J.C. Jaeger. *Conduction of Heat in Solids*. Clarendon Press, Oxford, 2nd edition, 1959.
3. W. Visser. A finite element method for the determination of non-stationary temperature distribution and thermal deformation. In *Proc. 1st Conf. Matrix Methods in Structural Mechanics*, volume AFFDL-TR-66–80, Wright-Patterson Air force Base, Ohio, October 1966.
4. O.C. Zienkiewicz and Y.K. Cheung. *The Finite Element Method in Structural Mechanics*. McGraw-Hill, London, 1967.
5. E.L. Wilson and R.E. Nickell. Application of finite element method to heat conduction analysis. *Nucl. Eng. Design*, **4**, 1–11, 1966.
6. O.C. Zienkiewicz and C.J. Parekh. Transient field problems – two and three dimensional analysis by isoparametric finite elements. *Internat. J. Num. Meth. Eng.*, **2**, 61–71, 1970.
7. K. Terzhagi and R.B. Peck. *Soil Mechanics in Engineering*. John Wiley & Sons, New York, 1948.
8. D.K. Todd. *Ground Water Hydrology*. John Wiley & Sons, New York, 1959.
9. P.L. Arlett, A.K. Bahrani, and O.C. Zienkiewicz. Application of finite elements to the solution of Helmholz's equation. *Proc. IEE*, **115**, 1762–64, 1968.
10. C. Taylor, B.S. Patil, and O.C. Zienkiewicz. Harbour oscillation: a numerical treatment for undamped natural modes. *Proc. Inst. Civ. Eng.,* **43**, 141–56, 1969.
11. O.C. Zienkiewicz and R.E. Newton. Coupled vibration of a structure submerged in a compressible fluid. In *Proc. Int. Symp. on Finite Element Techniques*, pages 1–15, Stuttgart, 1969.
12. A.K. Chopra. *Dynamics of Structures*. Prentice-Hall, Upper Saddle River, NJ, 1995.
13. J.S. Archer. Consistent mass matrix for distributed systems. *Proc. Am. Soc. Civ. Eng.*, **89**(ST4), 161, 1963.
14. F.A. Leckie and G.M. Lindberg. The effect of lumped parameters on beam frequencies. *Aero. Q.*, **14**, 234, 1963.
15. O.C. Zienkiewicz and Y.K. Cheung. The finite element method for analysis of elastic isotropic and orthotropic slabs. *Proc. Inst. Civ. Eng.*, **28**, 471–88, 1964.
16. R.W. Clough and J. Penzien. *Dynamics of Structures*. McGraw-Hill, New York, 2nd edition, 1993.
17. S.W. Key and Z.E. Beisinger. The transient dynamic analysis of thin shells in the finite element method. In *Proc. 1st Conf. Matrix Methods in Structural Mechanics*, volume AFFDL-TR-6680, pages 667–710, Wright-Patterson Air force Base, Ohio, October 1966.
18. E. Hinton, T. Rock, and O.C. Zienkiewicz. A note on mass lumping and related processes in the finite element method. *Earthquake Eng. Struct. Dyn.*, **4**, 245–49, 1976.
19. P. Tong, T.H.H. Pian, and L.L. Bociovelli. Mode shapes and frequencies by the finite element method using consistent and lumped matrices. *Comp. Struct.*, **1**, 623–38, 1971.
20. I. Fried and D.S. Malkus. Finite element mass matrix lumping by numerical integration with the convergence rate loss. *Internat. J. Solids Struct.*, **11**, 461–65, 1975.
21. J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.
22. I. Fried. Gradient methods for finite element eigen problems. *AIAA J.*, **7**, 739–41, 1969.
23. J.H. Wilkinson and C. Reinsch. *Linear Algebra. Handbook for Automatic Computation*, volume II. Springer-Verlag, Berlin, 1971.
24. K.K. Gupta. Solution of eigenvalue problems by Sturm sequence method. *Internat. J. Num. Meth. Eng.*, **4**, 379–404, 1972.
25. A. Jennings. Mass condensation and similarity iterations for vibration problems. *Internat. J. Num. Meth. Eng.*, **6**, 543–52, 1973.

26. B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, NJ, 1980.
27. K.J. Bathe. *Finite Element Procedures*. Prentice-Hall, Englewood Cliffs, NJ, 1996.
28. H.L. Cox. Vibration of missiles. *Aircraft Eng*., **33**, 2–7 and 48–55, 1961.
29. A. Jennings. Natural vibration of a free structure. *Aircraft Eng*., **34**, 8, 1962.
30. W.C. Hurty and M.F. Rubinstein. *Dynamics of Structures*. Prentice-Hall, Englewood Cliffs, NJ, 1974.
31. A. Craig and M.C.C. Bampton. On the iterative solution of semi definite eigenvalue problems. *Aero. J*., **75**, 287–90, 1971.
32. J. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
33. O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*, Volume 2. McGraw-Hill, London, 4th edition, 1991.
34. H.-C. Chen and R.L. Taylor. Using Lanczos vectors and Pitz vectors for computing dynamic responses. *Engrg. Comp*., **6**, 151–57, 1989.
35. A. Ibrabimbegovic, H.C. Chen, E.L. Wilson, and R.L. Taylor. Ritz method for dynamic analysis of large discrete linear systems with non-proportional damping. *Earthquake Eng. Struct. Dyn*., **19**, 877–89, 1990.
36. H.-C. Chen and R.L. Taylor. Properties and solutions of the eigensystem of non-proportionally damped linear dynamic systems. Technical Report UCB/SEMM-86/10, University of California, Berkeley, November 1986.
37. E.O. Brigham. *The Fast Fourier Transform*. Prentice-Hall, Englewood Cliffs, NJ, 1974.
38. H.S. Chen and C.C. Mei. Hybrid-element method for water waves. In *Proc. Modelling Techniques Conf*. (*Modelling 1975*), volume 1, pages 63–81, San Francisco, 1975.
39. O.C. Zienkiewicz and P. Bettess. Infinite elements in the study of fluid-structure interaction problems. In *2nd Int. Symp. on Computing Methods in Applied Science and Engineering*, Versailles, France, December 1975.
40. J. Penzien. Frequency domain analysis including radiation damping and water load coupling. In O.C. Zienkiewicz, R.W. Lewis, and K.G. Stagg, editors, *Numerical Methods in Offshore Engineering*. John Wiley & Sons, 1978.
41. E.L. Wilson and J. Penzien. Evaluation of orthogonal damping matrices. *Internat. J. Num. Meth. Eng*., **4**, 5–10, 1972.
42. H.T. Thomson, T. Collins, and P. Caravani. A numerical study of damping. *Earthquake Eng. Struct. Dyn*., **3**, 97–103, 1974.
43. F.W. Williams. Natural frequencies of repetitive structures. *Q. J. Mech. Appl. Math*., **24**, 285–310, 1971.
44. D.L. Thomas. Standing waves in rotationally periodic structures. *J. Sound Vibr*., **37**, 288–90, 1974.
45. D.A. Evensen. Vibration analysis of multi-symmetric structures. *AIAA J*., **14**, 446–53, 1976.

18

The time dimension – discrete approximation in time

18.1 Introduction

In the last chapter we have shown how semi-discretization of dynamic or transient field problems leads in linear cases to sets of ordinary differential equations of the form

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \qquad \text{where} \qquad \frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} \equiv \dot{\mathbf{a}}, \text{ etc.} \tag{18.1}$$

subject to initial conditions

$$\mathbf{a}(0) = \mathbf{a}_0 \qquad \text{and} \qquad \dot{\mathbf{a}}(0) = \dot{\mathbf{a}}_0$$

for dynamics or

$$\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{18.2}$$

subject to the initial condition

$$\mathbf{a}(0) = \mathbf{a}_0$$

for heat transfer or similar problems.

In many practical situations non-linearities exist, typically altering the above equations by making

$$\mathbf{M} = \mathbf{M}(\mathbf{a}) \qquad \mathbf{C} = \mathbf{C}(\mathbf{a}) \qquad \mathbf{K}\mathbf{a} = \mathbf{P}(\mathbf{a}) \tag{18.3}$$

The analytical solutions previously discussed, while providing much insight into the behaviour patterns (and indispensable in establishing such properties as natural system frequencies), are in general not economical for the solution of transient problems in linear cases and not applicable when non-linearity exists. In this chapter we shall therefore revert to discretization processes applicable directly to the time domain.

For such discretization the finite element method, including in its definition the finite difference approximation, is of course widely applicable and provides the greatest possibilities, though much of the classical literature on the subject uses

only the latter.[1–6] We shall demonstrate here how the finite element method provides a useful generalization unifying many existing algorithms and providing a variety of new ones.

As the time domain is infinite we shall inevitably curtail it to a finite time increment Δt and relate the initial conditions at t_n (and sometimes before) to those at time $t_{n+1} = t_n + \Delta t$, obtaining so-called *recurrence relations*. In all of this chapter, the starting point will be that of the semi-discrete equations (18.1) or (18.2), though, of course, the full space-time domain discretization could be considered simultaneously. This, however, usually offers no advantage, for, with the regularity of the time domain, irregular space-time elements are not required. Indeed, if product-type shape functions are chosen, the process will be identical to that obtained by using first semi-discretization in space followed by time discretization. An exception here is provided in convection dominated problems where simultaneous discretization may be desirable, as we shall discuss in the Volume 3.

The first concepts of space-time elements were introduced in 1969–70[7–10] and the development of processes involving semi-discretization is presented in references 11–20. Full space-time elements are described for convection-type equations in references 21, 22 and 23 and for elastodynamics in references 24, 25 and 26.

The presentation of this chapter will be divided into four parts. In the first we shall derive a set of *single-step* recurrence relations for the linear first- and second-order problems of Eqs (18.2) and (18.1). Such schemes have a very general applicability and are preferable to *multistep schemes* described in the second part as the time step can be easily and adaptively varied. In the third part we briefly describe a *discontinuous Galerkin scheme* and show its application in some simple problems. In the final part we shall deal with generalizations necessary for *non-linear problems*.

When discussing stability problems we shall often revert to the concept of modally uncoupled equations introduced in the previous chapter. Here we recall that the equation systems (18.1) and (18.2) can be written as a set of scalar equations:

$$m_i \ddot{y}_i + c_i \dot{y}_i + k_i y_i + f_i = 0 \tag{18.4}$$

or

$$c_i \dot{y}_i + k_i y_i + f_i = 0 \tag{18.5}$$

in the respective eigenvalue participation factors y_i. We shall find that the stability requirements here are dependent on the eigenvalues associated with such equations, ω_i. It turns out, however, fortunately, that it is never necessary to obtain the system eigenvalues or eigenvectors due to a powerful theorem first stated for finite element problems by Irons and Treharne.[27]

The theorem states simply that the system eigenvalues can be bounded by the eigenvalues of individual elements ω^e. Thus

$$\begin{aligned} \min_j (\omega_j)^2 &\geqslant \min_e (\omega^e)^2 \\ \max_j (\omega_j)^2 &\leqslant \max_e (\omega^e)^2 \end{aligned} \tag{18.6}$$

The stability limits can thus (as will be shown later) be related to Eqs (18.4) or (18.5) written for a single element.

Single-step algorithms

18.2 Simple time-step algorithms for the first-order equation

18.2.1 Weighted residual finite element approach

We shall now consider Eq. (18.2) which may represent a semi-discrete approximation to a particular physical problem or simply be itself a discrete system. The objective is to obtain an approximation for $\mathbf{a}_{n+1}$ given the value of $\mathbf{a}_n$ and the forcing vector $\mathbf{f}$ acting in the interval of time Δt. It is clear that in the first interval $\mathbf{a}_n$ is the initial condition $\mathbf{a}_0$, thus we have an *initial value problem*. In subsequent time intervals $\mathbf{a}_n$ will always be a known quantity determined from the previous step.

In each interval, in the manner used in all finite element approximations, we assume that $\mathbf{a}$ varies as a polynomial and take here the lowest (linear) expansion as shown in Fig. 18.1 writing

$$\mathbf{a} \approx \hat{\mathbf{a}}(t) = \mathbf{a}_n + \frac{\tau}{\Delta t}(\mathbf{a}_{n+1} - \mathbf{a}_n) \tag{18.7}$$

with $\tau = t - t_n$.

This can be translated to the standard finite element expansion giving

$$\hat{\mathbf{a}}(t) = \sum \mathbf{N}_i \mathbf{a}_i = \left(1 - \frac{\tau}{\Delta t}\right)\mathbf{a}_n + \left(\frac{\tau}{\Delta t}\right)\mathbf{a}_{n+1} \tag{18.8}$$

in which the unknown parameter is $\mathbf{a}_{n+1}$.

The equation by which this unknown parameter is provided will be a weighted residual approximation to Eq. (18.2). Accordingly, we write the variational problem

$$\int_0^{\Delta t} \mathbf{w}(\tau)^{\mathrm{T}} [\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f}] \, \mathrm{d}\tau = 0 \tag{18.9}$$

in which $\mathbf{w}(\tau)$ is an arbitrary weighting function We write the approximate form

$$\mathbf{w}(\tau) = W(\tau)\delta\mathbf{a}_{n+1} \tag{18.10}$$

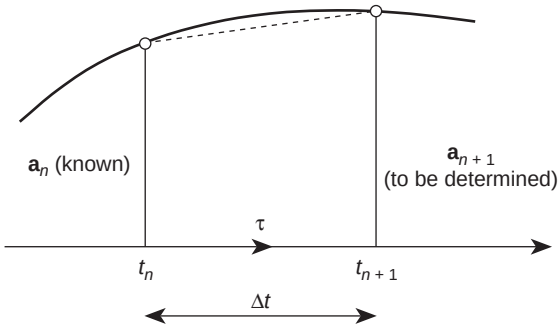


Fig. 18.1 Approximation to $\mathbf{a}$ in the time domain.

in which $\delta\mathbf{a}_{n+1}$ is an arbitrary parameter. With this approximation the weighted residual equation to be solved is given by

$$\int_0^{\Delta t} W(\tau)[\mathbf{C}\dot{\hat{\mathbf{a}}} + \mathbf{K}\hat{\mathbf{a}} + \mathbf{f}]\,\mathrm{d}\tau = \mathbf{0} \tag{18.11}$$

Introducing θ as a weighting parameter given by

$$\theta = \frac{1}{\Delta t}\frac{\int_0^{\Delta t} W\tau\,\mathrm{d}\tau}{\int_0^{\Delta t} W\,\mathrm{d}\tau} \tag{18.12}$$

we can immediately write

$$\frac{1}{\Delta t}\mathbf{C}(\mathbf{a}_{n+1} - \mathbf{a}_n) + \mathbf{K}[\mathbf{a}_n + \theta(\mathbf{a}_{n+1} - \mathbf{a}_n)] + \bar{\mathbf{f}} = \mathbf{0} \tag{18.13}$$

where $\bar{\mathbf{f}}$ represents an average value of $\mathbf{f}$ given by

$$\bar{\mathbf{f}} = \frac{\int_0^{\Delta t} W\mathbf{f}\,\mathrm{d}\tau}{\int_0^{\Delta t} W\,\mathrm{d}\tau} \tag{18.14}$$

or

$$\bar{\mathbf{f}} = \mathbf{f}_n + \theta(\mathbf{f}_{n+1} - \mathbf{f}_n) \tag{18.15}$$

if a linear variation of $\mathbf{f}$ is assumed within the time increment.

Equation (18.13) is in fact almost identical to a finite difference approximation to the governing equation (18.2) at time $t_n + \theta\Delta t$, and in this example little advantage is gained by introducing the finite element approximation. However, the averaging of the forcing term is important, as shown in Fig. 18.2, where a constant W (that is $\theta = 1/2$) is used and a finite difference approximation presents difficulties.

Figure 18.3 shows how different weight functions can yield alternate values of the parameter θ. The solution of Eq. (18.13) yields

$$\mathbf{a}_{n+1} = (\mathbf{C} + \theta\Delta t\mathbf{K})^{-1}[(\mathbf{C} - (1-\theta)\Delta t\mathbf{K})\mathbf{a}_n - \Delta t\bar{\mathbf{f}}] \tag{18.16}$$

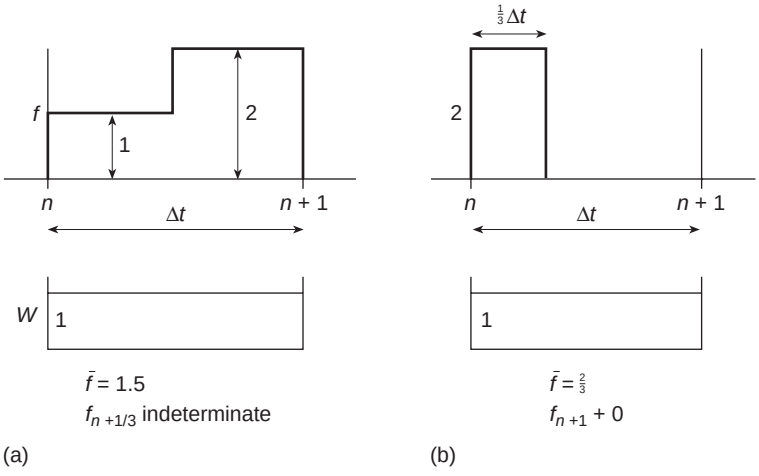


Fig. 18.2 'Averaging' of the forcing term in the finite-element-time approach.

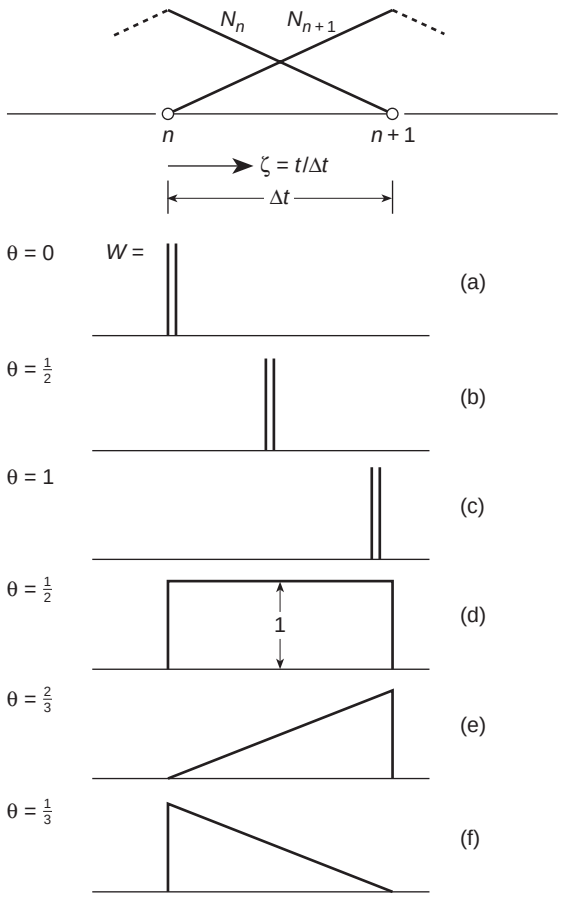


Fig. 18.3 Shape functions and weight functions for two-point recurrence formulae.

and it is evident that in general at each step of the computation a full equation system needs to be solved though of course a single inversion is sufficient for linear problems in which the time increment Δt is held constant. Methods requiring such an inversion are called *implicit*. However, when $\theta = 0$ and the matrix $\mathbf{C}$ is approximated by its lumped equivalent $\mathbf{C}_L$ the solution is called *explicit* and is exceedingly cheap for each time interval. We shall show later that explicit algorithms are *conditionally stable* (requiring the Δt to be less than some critical value Δt_{crit}) whereas implicit methods may be made *unconditionally stable* for some choices of the parameters.

18.2.2 Taylor series collocation

A frequently used alternative to the algorithm presented above is obtained by approximating separately $\mathbf{a}_{n+1}$ and $\dot{\mathbf{a}}_{n+1}$ by truncated Taylor series. We can write, assuming

that $\mathbf{a}_n$ and $\dot{\mathbf{a}}_n$ are known:

$$\mathbf{a}_{n+1} \approx \mathbf{a}_n + \Delta t \dot{\mathbf{a}}_n + \beta \Delta t (\dot{\mathbf{a}}_{n+1} - \dot{\mathbf{a}}_n) \tag{18.17}$$

and use collocation to satisfy the governing equation at t_{n+1} [or alternatively using the weight function shown in Fig. 18.3(c)]

$$\mathbf{C}\dot{\mathbf{a}}_{n+1} + \mathbf{K}\mathbf{a}_{n+1} + \mathbf{f}_{n+1} = \mathbf{0} \tag{18.18}$$

In the above β is a parameter, $0 \leqslant \beta \leqslant 1$, such that the last term of Eq. (18.17) represents a suitable difference approximation to the truncated expansion.

Substitution of Eq. (18.17) into Eq. (18.18) yields a recurrence relation for $\dot{\mathbf{a}}_{n+1}$:

$$\dot{\mathbf{a}}_{n+1} = -(\mathbf{C} + \beta \Delta t \mathbf{K})^{-1} [\mathbf{K}(\mathbf{a}_n + (1-\beta)\Delta t \dot{\mathbf{a}}_n) + \mathbf{f}_{n+1}] \tag{18.19}$$

where $\mathbf{a}_{n+1}$ is now computed by substitution of Eq. (18.19) into Eq. (18.17).

We remark that:

(a) the scheme is not self-starting† and requires the satisfaction of Eq. (18.2) at $t = 0$;
(b) the computation requires, with identification of the parameters $\beta = \theta$, an identical equation-solving problem to that in the finite element scheme of Eq. (18.16) and, finally, as we shall see later, stability considerations are identical.

The procedure is introduced here as it has some advantages in non-linear computations which will be shown later.

18.2.3 Other single-step procedures

As an alternative to the weighted residual process other possibilities of deriving finite element approximations exist, as discussed in Chapter 3. For instance, variational principles in time could be established and used for the purpose. This was indeed done in the early approaches to finite element approximation using Hamilton's or Gurtin's variational principle.[28–31] However, as expected, the final algorithms turn out to be identical. A variant on the above procedures is the use of a least square approximation for minimization of the equation residual.[12,13] This is obtained by insertion of the approximation (18.7) into Eq. (18.2). The reader can verify that the recurrence relation becomes

$$\begin{aligned}
&\left(\frac{1}{\Delta t}\mathbf{C}^{\mathrm{T}}\mathbf{C} + \frac{1}{2}(\mathbf{K}^{\mathrm{T}}\mathbf{C} + \mathbf{C}^{\mathrm{T}}\mathbf{K}) + \frac{1}{3}\Delta t \mathbf{K}^{\mathrm{T}}\mathbf{K}\right)\mathbf{a}_{n+1} \\
&\quad - \left(\frac{1}{\Delta t}\mathbf{C}^{\mathrm{T}}\mathbf{C} + \frac{1}{2}(\mathbf{K}^{\mathrm{T}}\mathbf{C} - \mathbf{C}^{\mathrm{T}}\mathbf{K}) - \frac{1}{6}\Delta t \mathbf{K}^{\mathrm{T}}\mathbf{K}\right)\mathbf{a}_n \\
&\quad + \frac{1}{\Delta t^2}\mathbf{C}^{\mathrm{T}} \int_0^{\Delta t} \mathbf{f}\, \mathrm{d}\tau + \frac{1}{\Delta t}\mathbf{K}^{\mathrm{T}} \int_0^{\Delta t} \mathbf{f}\tau\, \mathrm{d}\tau
\end{aligned} \tag{18.20}$$

requiring a more complex equation solution and always remaining 'implicit'. For this reason the algorithm is largely of purely theoretical interest, though as expected its

† By 'self-starting' we mean an algorithm is directly applicable without solving any subsidiary equations. Other definitions are also in use.

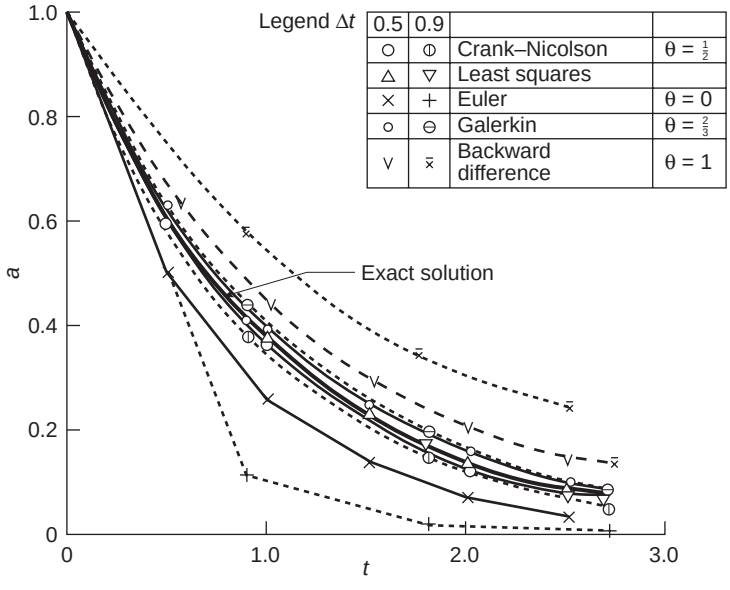


Fig. 18.4 Comparison of various time-stepping schemes on a first-order initial value problem.

accuracy is good, as shown in Fig. 18.4, in which a single degree of freedom equation (18.2) is used with

$$\mathbf{K} \to K = 1 \qquad \mathbf{C} \to C = 1 \qquad \mathbf{f} \to f = 0$$

with initial condition $a_0 = 1$. Here, the various algorithms previously discussed are compared. Now we see from this example that the $\theta = 1/2$ algorithm performs almost as well as the least squares one. It is popular for this reason and is known as the Crank–Nicolson scheme after its originators.[32]

18.2.4 Consistency and approximation error

For the convergence of any finite element approximation, it is necessary and sufficient that it be *consistent* and *stable*. We have discussed these two conditions in Chapter 10 and introduced appropriate requirements for boundary value problems. In the temporal approximation similar conditions apply though the stability problem is more delicate.

Clearly the function **a** itself and its derivatives occurring in the equation have to be approximated with a truncation error of $O(\Delta t^{\alpha})$, where $\alpha \geqslant 1$ is needed for consistency to be satisfied. For the first-order equation (18.2) it is thus necessary to use an approximating polynomial of order $p \geqslant 1$ which is capable of approximating $\dot{\mathbf{a}}$ to at least $O(\Delta t)$.

The *truncation error in the local approximation* of **a** with such an approximation is $O(\Delta t^2)$ and all the algorithms we have presented here using the $p = 1$ approximation of Eq. (18.7) will have at least that *local accuracy*,[33] as at a given time, $t = n\Delta t$, the

total error can be magnified n times and the final accuracy at a given time for schemes discussed here is of order $O(\Delta t)$ in general.

We shall see later that the arguments used here lead to $p \geqslant 2$ for the second-order equation (18.1) and that an increase of accuracy can generally be achieved by use of higher order approximating polynomials.

It would of course be possible to apply such a polynomial increase to the approximating function (18.7) by adding higher order degrees of freedom. For instance, we could write in place of the original approximation a quadratic expansion:

$$\mathbf{a} \approx \hat{\mathbf{a}}(\tau) = \mathbf{a}_n + \frac{\tau}{\Delta t}(\mathbf{a}_{n+1} - \mathbf{a}_n) + \frac{\tau}{\Delta t}\left(1 - \frac{\tau}{\Delta t}\right)\breve{\mathbf{a}}_{n+1} \tag{18.21}$$

where $\breve{\mathbf{a}}$ is a hierarchic internal variable. Obviously now both $\mathbf{a}_{n+1}$ and $\breve{\mathbf{a}}_{n+1}$ are unknowns and will have to be solved for simultaneously. This is accomplished by using the weighting function

$$\mathbf{w} = W(\tau)\delta\mathbf{a}_{n+1} + \breve{W}(\tau)\delta\breve{\mathbf{a}}_{n+1} \tag{18.22}$$

where $W(\tau)$ and $\breve{W}(\tau)$ are two independent weighting functions. This will obviously result in an increased size of the problem.

It is of interest to consider the first of these obtained by using the weighting W alone in the manner of Eq. (18.11). The reader will easily verify that we now have to add to Eq. (18.13) a term involving $\breve{\mathbf{a}}_{n+1}$ which is

$$\left[\frac{1}{\Delta t}(1 - 2\theta)\mathbf{C} + (\theta - \tilde{\theta})\mathbf{K}\right]\breve{\mathbf{a}}_{n+1} \tag{18.23}$$

where

$$\tilde{\theta} = \frac{1}{\Delta t^2}\frac{\int_0^{\Delta t} W\tau^2\,\mathrm{d}\tau}{\int_0^{\Delta t} W\,\mathrm{d}\tau}$$

It is clear that the choice of $\theta = \tilde{\theta} = 1/2$ eliminates the quadratic term and regains the previous scheme, thus showing that the values so obtained have a local truncation error of $O(\Delta t^3)$. This explains why the Crank–Nicolson scheme possesses higher accuracy.

In general the addition of higher order internal variables makes recurrence schemes too expensive and we shall later show how an increase of accuracy can be more economically achieved.

In a later section of this chapter we shall refer to some currently popular schemes in which often sets of $\mathbf{a}$'s have to be solved for simultaneously. In such schemes a discontinuity is assumed at the initial condition and additional parameters ($\tilde{\mathbf{a}}$) can be introduced to keep the same linear conditions we assumed previously. In this case an additional equation appears as a weighted satisfaction of continuity in time.

The procedure is therefore known as the *discontinuous Galerkin process* and was introduced initially by Lesaint and Raviart[34] to solve neutron transport problems. It has subsequently been applied to solve problems in fluid mechanics and heat transfer[22,35,36] and to problems in structural dynamics.[24–26] As we have already stated, the introduction of additional variables is expensive, so somewhat limited use of the concept has so far been made. However, one interesting application is in error estimation and adaptive time stepping.[37]

18.2.5 Stability

If we consider any of the recurrence algorithms so far derived, we note that for the homogeneous form (i.e., with $\mathbf{f} = \mathbf{0}$) all can be written in the form

$$\mathbf{a}_{n+1} = \mathbf{A}\mathbf{a}_n \tag{18.24}$$

where $\mathbf{A}$ is known as the *amplification matrix*.

The form of this matrix for the first algorithm derived is, for instance, evident from Eq. (18.16) as

$$\mathbf{A} = (\mathbf{C} + \theta\Delta t\mathbf{K})^{-1}(\mathbf{C} - (1-\theta)\Delta t\mathbf{K}) \tag{18.25}$$

Any errors present in the solution will of course be subject to amplification by precisely the same factor.

A general solution of any recurrence scheme can be written as

$$\mathbf{a}_{n+1} = \mu\mathbf{a}_n \tag{18.26}$$

and by insertion into Eq. (18.24) we observe that μ is given by eigenvalues of the matrix as

$$(\mathbf{A} - \mu\mathbf{I})\mathbf{a}_n = \mathbf{0} \tag{18.27}$$

Clearly if any eigenvalue μ is such that

$$|\mu| > 1 \tag{18.28}$$

all initially small errors will increase without limit and the solution will be unstable. In the case of complex eigenvalues the above is modified to the requirement that the modulus of μ satisfies Eq. (18.28).

As the determination of system eigenvalues is a large undertaking it is useful to consider only a scalar equation of the form (18.5) (representing, say, one-element performance). The bounding theorems of Irons and Treharne[27] will show why we do so and the results will provide general stability bounds if maximums are used. Thus for the case of the algorithm discussed in Eq. (18.27) we have a scalar A, i.e.

$$A = \frac{c - (1-\theta)\Delta tk}{c + \theta\Delta tk} = \frac{1 - (1-\theta)\omega\Delta t}{1 + \theta\omega\Delta t} = \mu \tag{18.29}$$

where $\omega = k/c$ and μ is evaluated from Eq. (18.27) simply as $\mu = A$ to allow non-trivial a_n. (This is equivalent to making the determinant of $\mathbf{A} - \mu\mathbf{I}$ zero in the more general case.)

In Fig. 18.5 we show how μ (or A) varies with $\omega\Delta t$ for various θ values. We observe immediately that:

(a) for $\theta \geqslant 1/2$

$$|\mu| \leqslant 1 \tag{18.30}$$

and such algorithms are *unconditionally stable*;

(b) for $\theta < 1/2$ we require

$$\omega\Delta t \leqslant \frac{2}{1-2\theta} \tag{18.31}$$

for stability. Such algorithms are therefore only *conditionally stable*. Here of course the explicit form with $\theta = 0$ is typical.

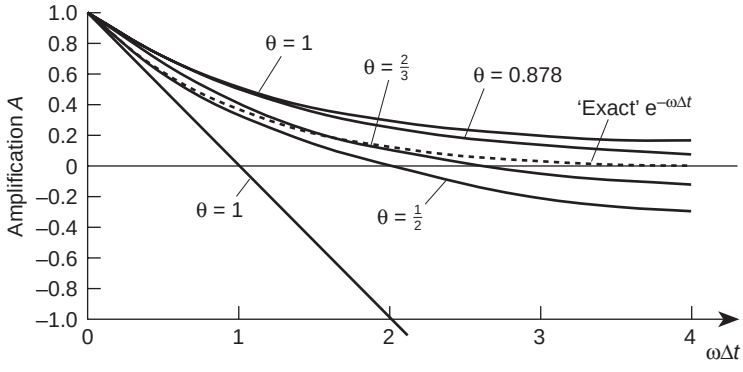


Fig. 18.5 The amplification A for various versions of the θ algorithm.

The critical value of Δt below which the scheme is stable with $\theta < 1/2$ needs the determination of the maximum value of μ from a typical element. For instance, in the case of the thermal conduction problem in which we have the coefficients c_{ii} and k_{ii} defined by expressions

$$c_{ii} = \int_\Omega \tilde{c} N_i^2 \, \mathrm{d}\Omega \qquad \text{and} \qquad k_{ii} = \int_\Omega \nabla N_i \tilde{k} \nabla N_i \, \mathrm{d}\Omega \tag{18.32}$$

we can presuppose uniaxial behaviour with a single degree of freedom and write for a linear element

$$N = \frac{h - x}{h} \qquad c = \int_0^h \tilde{c} N^2 \, \mathrm{d}x = \frac{1}{3}\tilde{c}h \qquad k = \int_0^h \tilde{k} \left(\frac{\mathrm{d}N}{\mathrm{d}x}\right)^2 \mathrm{d}x = \frac{\tilde{k}}{h}$$

Now

$$\omega = \frac{k}{c} = \frac{3\tilde{k}}{\tilde{c}h^2}$$

This gives

$$\Delta t \leqslant \frac{2}{1 - 2\theta} \frac{\tilde{c}h^2}{3\tilde{k}} = \Delta t_{\text{crit}} \tag{18.33}$$

which of course means that the smallest element size, $h_{\min}$, dictates overall stability. We note from the above that:

(a) in first-order problems the critical time step is proportional to h^2 and thus decreases rapidly with element size making explicit computations difficult;
(b) if mass lumping is assumed and therefore $c = \tilde{c}h/2$ the critical time step is larger.

In Fig. 18.6 we show the performance of the scheme described in Sec. 18.2.1 for various values of θ and Δt in the example we have already illustrated in Fig. 18.4, but now using larger values of Δt. We note now that the conditionally stable scheme with $\theta = 0$ and a stability limit of $\Delta t = 2$ shows oscillations as this limit is approached ($\Delta t = 1.5$) and diverges when exceeded.

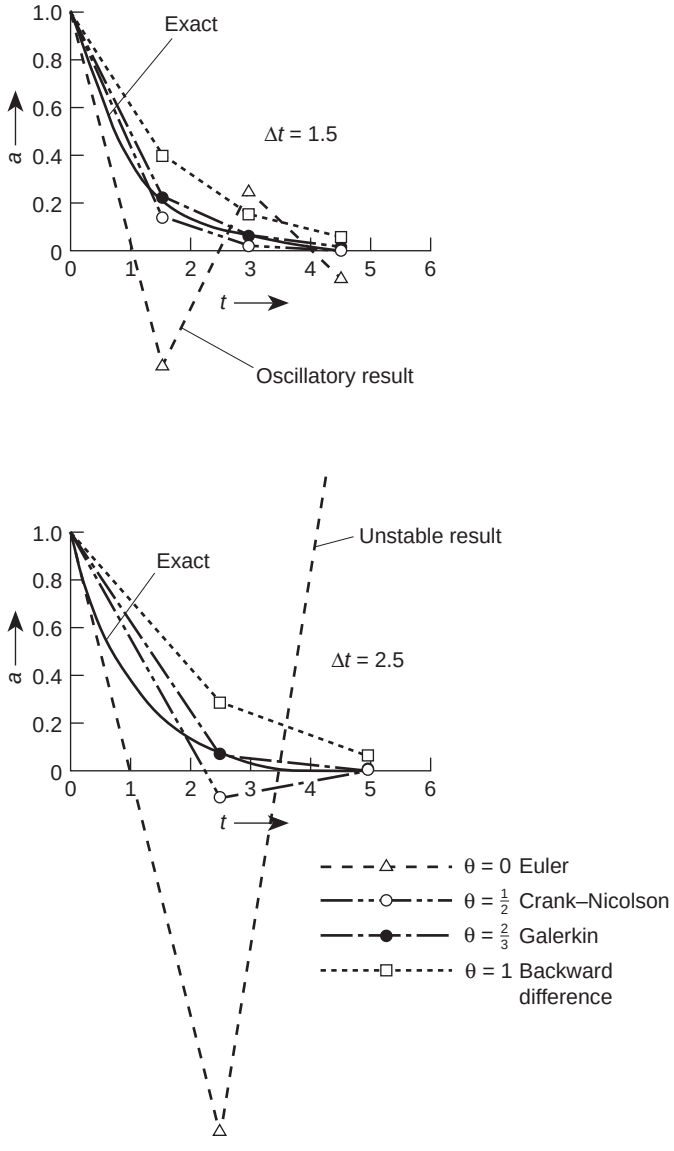


Fig. 18.6 Performance of some θ algorithms in the problem of Fig. 18.4 and larger time steps. Note oscillation and instability.

Stability computations which were presented for the algorithm of Sec. 18.2.1 can of course be repeated for the other algorithms which we have discussed.

If identical procedures are used, for instance on the algorithm of Sec. 18.2.2, we shall find that the stability conditions, based on the determinant of the amplification matrix $(\mathbf{A} - \mu\mathbf{I})$, are identical with the previous one providing we set $\theta = \beta$. Algorithms that give such identical determinants will be called *similar* in the following presentations.

In general, it is possible for different amplification matrices $\mathbf{A}$ to have identical determinants of $(\mathbf{A} - \mu\mathbf{I})$ and hence identical stability conditions, but differ otherwise. If in addition the amplification matrices are the same, the schemes are known as *identical*. In the two cases described here such an identity can be shown to exist despite different derivations.

18.2.6 Some further remarks. Initial conditions and examples

The question of choosing an optimal value of θ is not always obvious from theoretical accuracy considerations. In particular with $\theta = 1/2$ oscillations are sometimes present,[13] as we observe in Fig. 18.6 ($\Delta t = 2.5$), and for this reason some prefer to use[38] $\theta = 2/3$, which is considerably 'smoother' (and which incidentally corresponds to a standard Galerkin approximation). In Table 18.1 we show the results for a one-dimensional finite element problem where a bar at uniform initial temperature is subject to zero temperatures applied suddenly at the ends. Here 10 linear elements are used in the space dimension with $L = 1$. The oscillation errors occurring with $\theta = 1/2$ are much reduced for $\theta = 2/3$. The time step used here is much longer than that corresponding to the lowest eigenvalue period, but the main cause of the oscillation is in the abrupt discontinuity of the temperature change.

For similar reasons Liniger[39] derives θ which minimizes the error in the whole time domain and gives $\theta = 0.878$ for the simple one-dimensional case. We observe in Fig. 18.5 how well the amplification factor fits the exact solution with these values. Again this value will smooth out many oscillations. However, most oscillations are introduced by simply using a physically unrealistic initial condition.

In part at least, the oscillations which for instance occur with $\theta = 1/2$ and $\Delta t = 2.5$ (see Fig. 18.6) in the previous example are due to a sudden jump in the forcing term introduced at the start of the computation. This jump is evident if we consider this simple problem posed in the context of the whole time domain. We can take the problem as implying

$$f(t) = -1 \qquad \text{for} \quad t < 0$$

Table 18.1 Percentage error for finite elements in time: $\theta = 2/3$ and $\theta = 1/2$ (Crank–Nicolson) scheme; $\Delta t = 0.01$

t	$x = 0.1$		$x = 0.2$		$x = 0.3$		$x = 0.4$		$x = 0.5$	
θ	2/3	1/2	2/3	1/2	2/3	1/2	2/3	1/2	2/3	1/2
0.01	10.8	28.2	1.6	3.2	0.5	0.7	0.6	0.1	0.5	0.2
0.02	0.5	3.5	2.1	9.5	0.1	0.0	0.5	0.7	0.7	0.4
0.03	1.3	9.9	0.5	0.7	0.8	3.1	0.5	0.2	0.5	0.6
0.05	0.5	4.5	0.4	0.2	0.5	2.3	0.4	0.8	0.5	1.0
0.10	0.1	1.4	0.1	2.0	0.1	1.4	0.1	1.9	0.1	1.6
0.15	0.3	2.2	0.3	2.1	0.3	2.2	0.3	2.1	0.3	2.2
0.20	0.6	2.6	0.6	2.6	0.6	2.6	0.6	2.6	0.6	2.6
0.30	1.4	3.5	1.4	3.5	1.4	3.5	1.4	3.5	1.4	3.5

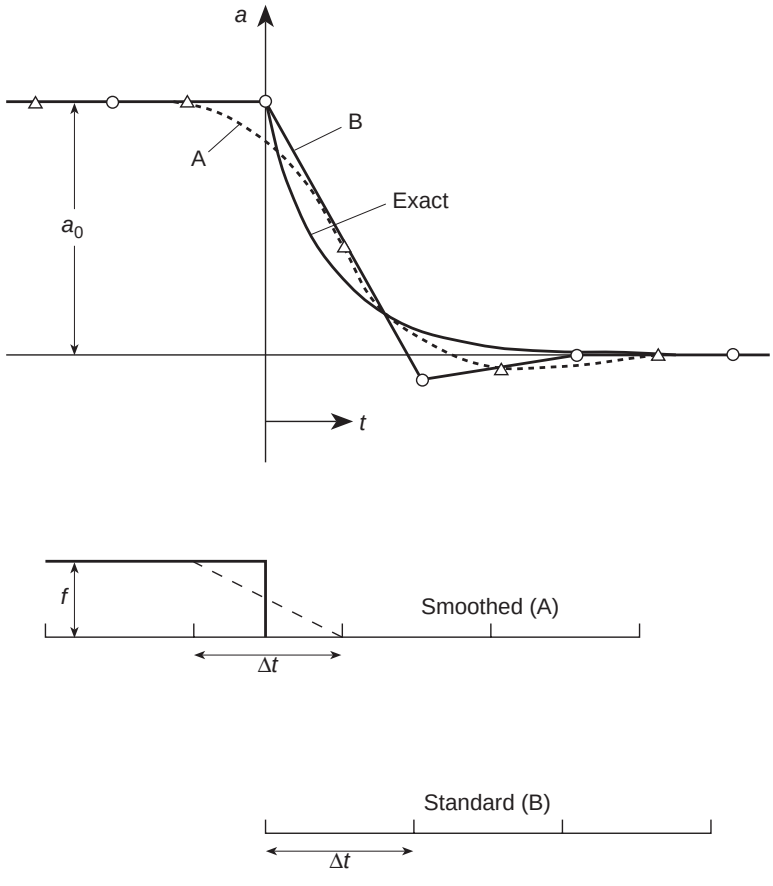


Fig. 18.7 Importance of 'smoothing' the force term in elimination of oscillations in the solution. $\Delta t = 2.5$.

giving the solution $u = 1$ with a sudden change at $t = 0$, resulting in

$$f(t) = 0 \qquad \text{for} \quad t \geqslant 0$$

As shown in Fig. 18.7 this represents a discontinuity of the loading function at $t = 0$.

Although load discontinuities are permitted by the algorithm they lead to a sudden discontinuity of $\dot{u}$ and hence induce undesirable oscillations. If in place of this discontinuity we assume that f varies linearly in the first time step Δt $(-\Delta t/2 \leqslant t \leqslant \Delta t/2)$ then smooth results are obtained with a much improved physical representation of the true solution, even for such a long time step as $t = 2.5$, as shown in Fig. 18.7.

Similar use of smoothing is illustrated in a multidegree of freedom system (the representation of heat conduction in a wall) which is solved using two-dimensional finite elements[40] (Fig. 18.8).

Here the problem corresponds to an instantaneous application of prescribed temperature ($T = 1$) at the wall sides with zero initial conditions. Now again troublesome

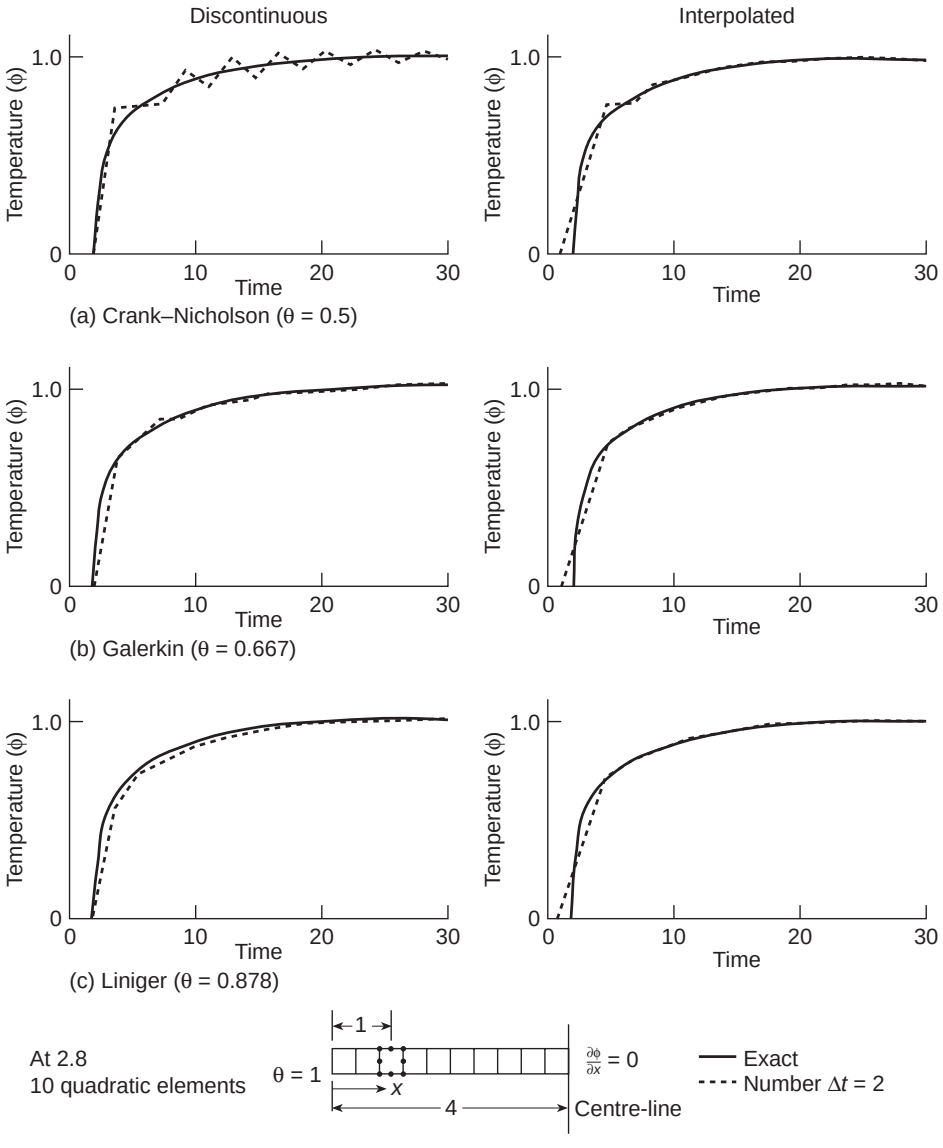
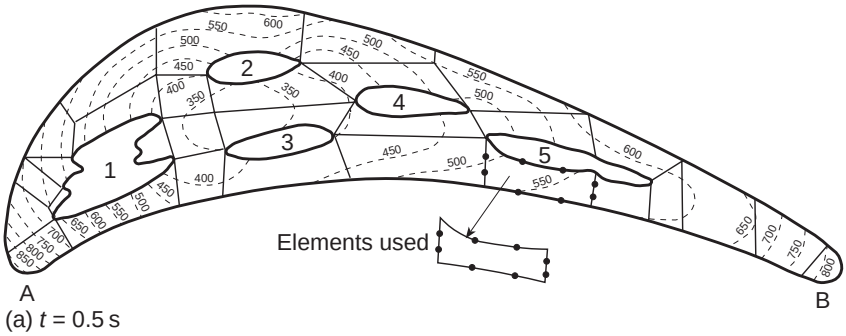


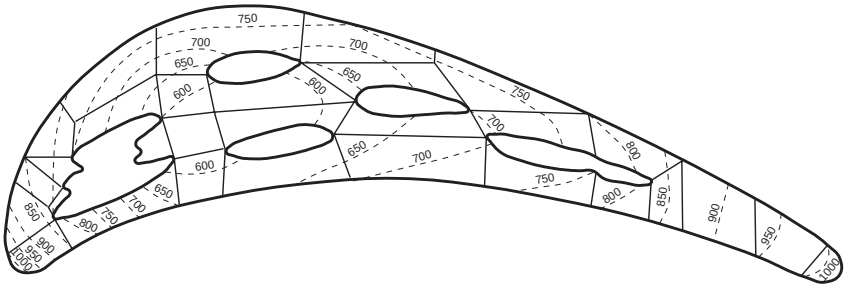
Fig. 18.8 Transient heating of a bar; comparison of discontinuous and interpolated (smoothed) initial conditions for single-step schemes.

oscillations are almost eliminated for $\theta = 1/2$ and improved results are obtained for other values of θ (2/3, 0.878) by assuming the step change to be replaced by a continuous one. Such smoothing is always advisable and a continuous representation of the forcing term is important.

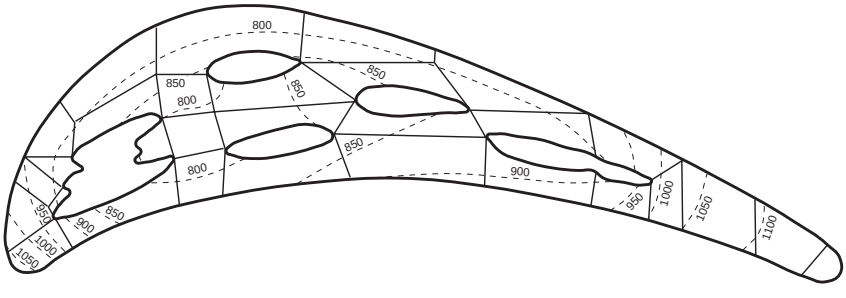
We conclude this section by showing a typical example of temperature distribution in a practical example in which high-order elements are used (Fig. 18.9).



(a) t = 0.5 s



(b) t = 1.0 s



(c) Steady-state solution

Specific heat c = 0.11 cal/gm °C
Density ρ = 7.99 gm/cm^3
Conductivity k = 0.05 cal/s cm °C
Gas temperature around blade = 1145 °C

Heat transfer coefficient α varies from 0.390 to 0.056 on the outside surfaces of the blade (A–B)

Hole number	Cooling hole temperature	α around perimeter of each hole
1	545 °C	0.0980
2	587 °C	0.0871

Fig. 18.9 Temperature distribution in a cooled rotor blade, initially at zero temperature.

18.3 General single-step algorithms for first- and second-order equations

18.3.1 Introduction

We shall introduce in this section two general single-step algorithms applicable to Eq. (18.1):

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0}$$

These algorithms will of course be applicable to the first-order problem of Eq. (18.2) simply by putting $\mathbf{M} = \mathbf{0}$.

An arbitrary degree polynomial p for approximating the unknown function $\mathbf{a}$ will be used and we must note immediately that for the second-order equations $p \geqslant 2$ is required for consistency as second-order derivatives have to be approximated.

The first algorithm SSpj (single step with approximation of degree p for equations of order $j = 1, 2$) will be derived by use of the weighted residual process and we shall find that the algorithm of Sec. 18.2.1 is but a special case. The second algorithm GNpj (generalized Newmark[41] with degree p and order j) will follow the procedures using a truncated Taylor series approximation in a manner similar to that described in Sec. 18.2.2.

In what follows we shall *assume* that at the start of the interval, i.e., at $t = t_n$, we know the values of the unknown function $\mathbf{a}$ and its derivatives, that is $\mathbf{a}_n$, $\dot{\mathbf{a}}_n$, $\ddot{\mathbf{a}}_n$ up to ${}^{p-1}\mathbf{a}_n$ and our objective will be to determine $\mathbf{a}_{n+1}$, $\dot{\mathbf{a}}_{n+1}$, $\ddot{\mathbf{a}}_{n+1}$ up to ${}^{p-1}\mathbf{a}_{n+1}$, where p is the order of the expansion used in the interval.

This is indeed a rather strong presumption as for first-order problems we have already stated that only a single initial condition, $\mathbf{a}(0)$, is given and for second-order problems two conditions, $\mathbf{a}(0)$ and $\dot{\mathbf{a}}(0)$, are available (i.e., the initial displacement and velocity of the system). We can, however, argue that if the system starts from rest we could take $\mathbf{a}(0)$ to ${}^{p-1}\mathbf{a}(0)$ as equal to zero and, providing *that suitably continuous forcing of the system occurs*, the solution will remain smooth in the higher derivatives. Alternatively, we can differentiate the differential equation to obtain the necessary starting values.

18.3.2 The weighted residual finite element form SSpj[18,19]

The expansion of the unknown vector $\mathbf{a}$ will be taken as a polynomial of degree p. With the *known* values of $\mathbf{a}_n$, $\dot{\mathbf{a}}_n$, $\ddot{\mathbf{a}}_n$ up to ${}^{p-1}\mathbf{a}_n$ at the beginning of the time step Δt, we write, as in Sec. 18.2.1,

$$\tau = t - t_n \qquad \Delta t = t_{n+1} - t_n \tag{18.34}$$

and using a polynomial expansion of degree p,

$$\mathbf{a} \approx \hat{\mathbf{a}} = \mathbf{a}_n + \tau\dot{\mathbf{a}}_n + \frac{1}{2!}\tau^2\ddot{\mathbf{a}}_n + \cdots + \frac{1}{(p-1)!}\tau^{p-1}\,{}^{p-1}\mathbf{a}_n + \frac{1}{p!}\tau^p\boldsymbol{\alpha}_n^p \tag{18.35}$$

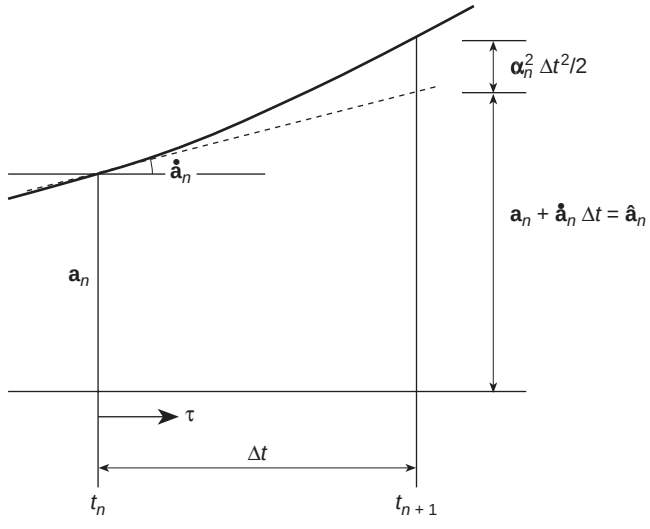


Fig. 18.10 A second-order time approximation.

where the only unknown is the vector $\boldsymbol{\alpha}_n^p$,

$$\boldsymbol{\alpha}_n^p \equiv \overset{p}{\mathbf{a}} = \frac{\mathrm{d}^p}{\mathrm{d}t^p}\mathbf{a} \tag{18.36}$$

which represents some average value of the pth derivative occurring in the interval Δt. The approximation to $\mathbf{a}$ for the case of $p = 2$ is shown in Fig. 18.10.

We recall that in order to obtain a consistent approximation to all the derivatives that occur in the differential equations (18.1) and (18.2), $p \geqslant 2$ is necessary for the full dynamic equation and $p \geqslant 1$ is necessary for the first-order equation. Indeed the lowest approximation, that is $p = 1$, is the basis of the algorithm derived in the previous section.

The recurrence algorithm will now be obtained by inserting $\mathbf{a}$, $\dot{\mathbf{a}}$ and $\ddot{\mathbf{a}}$ obtained by differentiating Eq. (18.35) into Eq. (18.1) and satisfying the weighted residual equation with a single weighting function $W(\tau)$. This gives

$$\begin{aligned}\int_0^{\Delta t} W(\tau)\Bigg[\mathbf{M}\Bigg(\ddot{\mathbf{a}}_n + \tau\dddot{\mathbf{a}}_n + \cdots + \frac{1}{(p-2)!}\tau^{p-2}\boldsymbol{\alpha}_n^p\Bigg) \\ + \mathbf{C}\Bigg(\dot{\mathbf{a}}_n + \tau\ddot{\mathbf{a}}_n + \cdots + \frac{1}{(p-1)!}\tau^{p-1}\boldsymbol{\alpha}_n^p\Bigg) \\ + \mathbf{K}\Bigg(\mathbf{a}_n + \tau\dot{\mathbf{a}}_n + \cdots + \frac{1}{p!}\tau^p\boldsymbol{\alpha}_n^p\Bigg) + \mathbf{f}\Bigg]\,\mathrm{d}t = \mathbf{0}\end{aligned} \tag{18.37}$$

as the basic equation for determining $\boldsymbol{\alpha}_n^p$.

Without specifying the weighting function used we can, as in Sec. 18.2.1, generalize its effects by writing

$$\theta_k = \frac{\int_0^{\Delta t} W\tau^k \,\mathrm{d}\tau}{\int_0^{\Delta t} W \,\mathrm{d}\tau} \qquad k = 0, 1, \ldots, p$$
$$\bar{\mathbf{f}} = \frac{\int_0^{\Delta t} W\mathbf{f} \,\mathrm{d}\tau}{\int_0^{\Delta t} W \,\mathrm{d}\tau} \tag{18.38}$$

where we note θ_0 is always unity. Equation (18.37) can now be written more compactly as

$$\mathbf{A}\boldsymbol{\alpha}_n^p + \mathbf{M}\ddot{\bar{\mathbf{a}}}_{n+1} + \mathbf{C}\dot{\bar{\mathbf{a}}}_{n+1} + \mathbf{K}\bar{\mathbf{a}}_{n+1} + \bar{\mathbf{f}} = \mathbf{0} \tag{18.39}$$

where

$$\mathbf{A} = \frac{\Delta t^{p-2}}{(p-2)!}\mathbf{M} + \frac{\Delta t^{p-1}}{(p-1)!}\mathbf{C} + \frac{\Delta t^p}{p!}\mathbf{K}$$
$$\bar{\mathbf{a}}_{n+1} = \sum_{q=0}^{p-1} \frac{\theta_q \Delta t^q}{q!} {}^q\mathbf{a}_n$$
$$\dot{\bar{\mathbf{a}}}_{n+1} = \sum_{q=1}^{p-1} \frac{\theta_{q-1} \Delta t^{q-1}}{(q-1)!} {}^q\mathbf{a}_n \tag{18.40}$$
$$\ddot{\bar{\mathbf{a}}}_{n+1} = \sum_{q=2}^{p-1} \frac{\theta_{q-2} \Delta t^{q-2}}{(q-2)!} {}^q\mathbf{a}_n$$

As $\bar{\mathbf{a}}_{n+1}$, $\dot{\bar{\mathbf{a}}}_{n+1}$ and $\ddot{\bar{\mathbf{a}}}_{n+1}$ can be computed directly from the initial values we can solve Eq. (18.39) to obtain

$$\boldsymbol{\alpha}_n^p = -\mathbf{A}^{-1}\left[\mathbf{M}\ddot{\bar{\mathbf{a}}}_{n+1} + \mathbf{C}\dot{\bar{\mathbf{a}}}_{n+1} + \mathbf{K}\bar{\mathbf{a}}_{n+1} + \bar{\mathbf{f}}\right] \tag{18.41}$$

It is important to observe that $\bar{\mathbf{a}}_{n+1}$, $\dot{\bar{\mathbf{a}}}_{n+1}$ and $\ddot{\bar{\mathbf{a}}}_{n+1}$ here represent some mean predicted values of $\mathbf{a}_{n+1}$, $\dot{\mathbf{a}}_{n+1}$ and $\ddot{\mathbf{a}}_{n+1}$ in the interval and satisfy the governing Eq. (18.1) in a weighted sense if $\boldsymbol{\alpha}_n^p$ *is chosen as zero*.

The procedure is now complete as knowledge of the vector $\boldsymbol{\alpha}_n^p$ permits the evaluation of $\mathbf{a}_{n+1}$ to ${}^p\bar{\mathbf{a}}_{n+1}$ from the expansion originally used in Eq. (18.35) by putting $\tau = \Delta t$. This gives

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \Delta t\dot{\mathbf{a}}_n + \cdots + \frac{\Delta t^p}{p!}\boldsymbol{\alpha}_n^p = \hat{\mathbf{a}}_{n+1} + \frac{\Delta t^p}{p!}\boldsymbol{\alpha}_n^p$$
$$\dot{\mathbf{a}}_{n+1} = \dot{\mathbf{a}}_n + \Delta t\ddot{\mathbf{a}}_n + \cdots + \frac{\Delta t^{p-1}}{(p-1)!}\boldsymbol{\alpha}_n^p = \dot{\hat{\mathbf{a}}}_{n+1} + \frac{\Delta t^{p-1}}{(p-1)!}\boldsymbol{\alpha}_n^p \tag{18.42}$$
$$\vdots$$
$${}^{p-1}\mathbf{a}_{n+1} = {}^{p-1}\mathbf{a}_n + \Delta t\boldsymbol{\alpha}_n^p$$

In the above $\hat{\mathbf{a}}$, $\dot{\hat{\mathbf{a}}}$, etc., are again quantities that can be written down *a priori* (before solving for $\boldsymbol{\alpha}_n^p$). These represent predicted values at the end of the interval with $\boldsymbol{\alpha}_n^p = \mathbf{0}$.

To summarize, the general algorithm necessitates the choice of values for θ_1 to θ_p and requires

(a) computation of $\bar{\mathbf{a}}$, $\dot{\bar{\mathbf{a}}}$ and $\ddot{\bar{\mathbf{a}}}$ using the definitions of Eqs (18.40);
(b) computation of $\boldsymbol{\alpha}_n^p$ by solution of Eq. (18.41);
(c) computation of $\mathbf{a}_{n+1}$ to ${}^{p-1}\mathbf{a}_{n+1}$ by Eqs (18.42).

After completion of stage (c) a new time step can be started. In first-order problems the computation of $\ddot{\bar{\mathbf{a}}}$ can obviously be omitted.

If matrices $\mathbf{C}$ and $\mathbf{M}$ are diagonal the solution of Eq. (18.41) is trivial providing we choose

$$\theta_p = 0 \tag{18.43}$$

With this choice the algorithms are *explicit* but, as we shall find later, only sometimes *conditionally stable*.

When $\theta_p \neq 0$, *implicit* algorithms of various kinds will be available and some of these will be found to be *unconditionally stable*. Indeed, it is such algorithms that are of great practical use.

Important special cases of the general algorithm are the SS11 and SS22 forms given below.

The SS11 algorithm

If we consider the first-order equation (that is $j = 1$) it is evident that only the value of $\mathbf{a}_n$ is necessarily specified as the initial value for any computation. For this reason the choice of a linear expansion in the time interval is *natural* ($p = 1$) and the SS11 algorithm is for that reason most widely used.

Now the approximation of Eq. (18.35) is simply

$$\mathbf{a} = \mathbf{a}_n + \tau\boldsymbol{\alpha} \qquad (\boldsymbol{\alpha}_n^1 = \boldsymbol{\alpha} = \dot{\mathbf{a}}) \tag{18.44}$$

and the approximation to the average satisfaction of Eq. (18.2) is simply

$$\mathbf{C}\boldsymbol{\alpha} + \mathbf{K}(\bar{\mathbf{a}}_{n+1} + \theta\Delta t\boldsymbol{\alpha}) + \bar{\mathbf{f}} = \mathbf{0} \tag{18.45}$$

with $\bar{\mathbf{a}}_{n+1} = \mathbf{a}_n$. Solution of Eq. (18.45) determines $\boldsymbol{\alpha}$ as

$$\boldsymbol{\alpha} = -(\mathbf{C} + \theta\Delta t\mathbf{K})^{-1}\left(\bar{\mathbf{f}} + \mathbf{K}\mathbf{a}_n\right) \tag{18.46}$$

and finally

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \Delta t\boldsymbol{\alpha} \tag{18.47}$$

The reader will verify that this process is identical to that developed in Eqs (18.7)–(18.13) and hence will not be further discussed except perhaps for noting the more elegant computation form above.

The SS22 algorithm

With Eq. (18.1) we considered a second-order system ($j = 2$) in which the necessary initial conditions require the specification of two quantities, $\mathbf{a}_n$ and $\dot{\mathbf{a}}_n$. The simplest and most natural choice here is to specify the minimum value of p, that is $p = 2$, as this does not require computation of additional derivatives at the start. This algorithm, SS22, is *thus basic for dynamic equations* and we present it here in full.

From Eq. (18.35) the approximation is a quadratic

$$\mathbf{a} = \mathbf{a}_n + \tau\dot{\mathbf{a}}_n + \tfrac{1}{2}\tau^2\boldsymbol{\alpha} \qquad (\boldsymbol{\alpha}_n^2 = \boldsymbol{\alpha} = \ddot{\mathbf{a}}) \tag{18.48}$$

The approximate form of the 'average' dynamic equation is now

$$\mathbf{M}\boldsymbol{\alpha} + \mathbf{C}\big(\dot{\bar{\mathbf{a}}}_{n+1} + \theta_1\Delta t\boldsymbol{\alpha}\big) + \mathbf{K}\big(\bar{\mathbf{a}}_{n+1} + \tfrac{1}{2}\theta_2\Delta t\boldsymbol{\alpha}\big) + \bar{\mathbf{f}} = \mathbf{0} \tag{18.49}$$

with predicted 'mean' values

$$\begin{aligned} \bar{\mathbf{a}}_{n+1} &= \mathbf{a}_n + \theta_1\Delta t\dot{\mathbf{a}}_n \\ \dot{\bar{\mathbf{a}}}_{n+1} &= \dot{\mathbf{a}}_n \end{aligned} \tag{18.50}$$

After evaluation of $\boldsymbol{\alpha}$ from Eq. (18.49), the values of $\mathbf{a}_{n+1}$ are found by Eqs (18.42) which become simply

$$\begin{aligned} \mathbf{a}_{n+1} &= \mathbf{a}_n + \Delta t\dot{\mathbf{a}}_n + \tfrac{1}{2}\Delta t^2\boldsymbol{\alpha} \\ \dot{\mathbf{a}}_{n+1} &= \dot{\mathbf{a}}_n + \Delta t\boldsymbol{\alpha} \end{aligned} \tag{18.51}$$

This completes the algorithm which is of much practical value in the solution of dynamics problems.

In many respects it resembles the Newmark algorithm[41] which we shall discuss in the next section and which is widely used in practice. Indeed, its stability properties turn out to be identical with the Newmark algorithm, i.e.,

$$\begin{aligned} \theta_1 &= \gamma \\ \theta_2 &= 2\beta \\ \theta_1 &\geqslant \theta_2 \geqslant \tfrac{1}{2} \end{aligned} \tag{18.52}$$

for unconditional stability. In the above γ and β are conventionally used Newmark parameters.

For $\theta_2 = 0$ the algorithm is 'explicit' (assuming both $\mathbf{M}$ and $\mathbf{C}$ to be diagonal) and can be made conditionally stable if $\theta_1 \geqslant 1/2$.

The algorithm is clearly applicable to first-order equations described as SS21 and we shall find that the stability conditions are identical. In this case, however, it is necessary to identify an initial condition for $\dot{\mathbf{a}}_0$ and

$$\dot{\mathbf{a}}_0 = -\mathbf{C}^{-1}\big(\mathbf{K}\mathbf{a}\mathbf{a}_0 + \bar{\mathbf{f}}_0\big)$$

is one possibility.

18.3.3 Truncated Taylor series collocation algorithm GNpj

It will be shown that again as in Sec. 18.2.2 a non-self-starting process is obtained, which in most cases, however, gives an algorithm similar to the SSpj one we have derived. The classical Newmark method[41] will be recognized as a particular case together with its derivation process in a form presented generally in existing texts.[42] Because of this similarity we shall term the new algorithm generalized Newmark (GNpj).

In the derivation, we shall now consider the satisfaction of the governing equation (18.1) only at the end points of the interval Δt [collocation which results from the weighting function shown in Fig. 18.3(c)] and write

$$\mathbf{M}\ddot{\mathbf{a}}_{n+1} + \mathbf{C}\dot{\mathbf{a}}_{n+1} + \mathbf{K}\mathbf{a}_{n+1} + \mathbf{f}_{n+1} = \mathbf{0} \tag{18.53}$$

with appropriate approximations for the values of $\mathbf{a}_{n+1}$, $\dot{\mathbf{a}}_{n+1}$ and $\ddot{\mathbf{a}}_{n+1}$.

If we consider a truncated Taylor series expansion similar to Eq. (18.17) for the function $\mathbf{a}$ and its derivatives, we can write

$$\begin{aligned}
\mathbf{a}_{n+1} &= \mathbf{a}_n + \Delta t\dot{\mathbf{a}}_n + \cdots + \frac{\Delta t^p}{p!}\overset{p}{\mathbf{a}}_n + \beta_p \frac{\Delta t^p}{p!}\left(\overset{p}{\mathbf{a}}_{n+1} - \overset{p}{\mathbf{a}}_n\right) \\
\dot{\mathbf{a}}_{n+1} &= \dot{\mathbf{a}}_n + \Delta t\ddot{\mathbf{a}}_n + \cdots + \frac{\Delta t^{p-1}}{(p-1)!}\overset{p}{\mathbf{a}}_n + \beta_{p-1} \frac{\Delta t^{p-1}}{(p-1)!}\left(\overset{p}{\mathbf{a}}_{n+1} - \overset{p}{\mathbf{a}}_n\right) \\
&\vdots \\
\overset{p-1}{\mathbf{a}}_{n+1} &= \overset{p-1}{\mathbf{a}}_n + \Delta t\beta_1 \frac{\Delta t^{p-1}}{(p-1)!}\left(\overset{p}{\mathbf{a}}_{n+1} - \overset{p}{\mathbf{a}}_n\right) \qquad (\beta_0 \equiv 1)
\end{aligned} \tag{18.54}$$

In Eqs (18.44) we have effectively allowed for a polynomial of degree p (i.e., by including terms up to Δt^p) plus a Taylor series remainder term in each of the expansions for the function and its derivatives with a parameter β_j, $j = 1, 2, \ldots, p$, which can be chosen to give good approximation properties to the algorithm.

Insertion of the first three expressions of (18.54) into Eq. (18.53) gives a single equation from which $\mathbf{a}^p_{n+1}$ can be found. When this is determined, $\mathbf{a}_{n+1}$ to $\mathbf{a}^{p-1}_{n+1}$ can be evaluated using Eqs (18.54). Satisfying Eq. (18.53) is almost a 'collocation' which could be obtained by inserting the expressions (18.54) into a weighted residual form (18.37) with $W = \delta(t_{n+1})$ (the Dirac delta function). However, the expansion does not correspond to a unique function $\mathbf{a}$.

In detail we can write the first three expansions of Eqs (18.54) as

$$\begin{aligned}
\mathbf{a}_{n+1} &= \breve{\mathbf{a}}_{n+1} + \beta_p \frac{\Delta t^p}{p!}\overset{p}{\mathbf{a}}_{n+1} \\
\dot{\mathbf{a}}_{n+1} &= \dot{\breve{\mathbf{a}}}_{n+1} + \beta_{p-1} \frac{\Delta t^{p-1}}{(p-1)!}\overset{p}{\mathbf{a}}_{n+1} \\
\ddot{\mathbf{a}}_{n+1} &= \ddot{\breve{\mathbf{a}}}_{n+1} + \beta_{p-2} \frac{\Delta t^{p-2}}{(p-2)!}\overset{p}{\mathbf{a}}_{n+1}
\end{aligned} \tag{18.55}$$

where

$$\begin{aligned}
\breve{\mathbf{a}}_{n+1} &= \mathbf{a}_n + \Delta t\dot{\mathbf{a}}_n + \cdots + (1-\beta_p)\frac{\Delta t^p}{p!}\overset{p}{\mathbf{a}}_n + \cdots \\
\dot{\breve{\mathbf{a}}}_{n+1} &= \dot{\mathbf{a}}_n + \Delta t\ddot{\mathbf{a}}_n + \cdots + (1-\beta_{p-1})\frac{\Delta t^{p-1}}{(p-1)!}\overset{p}{\mathbf{a}}_n + \cdots \\
\ddot{\breve{\mathbf{a}}}_{n+1} &= \ddot{\mathbf{a}}_n + \Delta t\dddot{\mathbf{a}}_n + \cdots + (1-\beta_{p-2})\frac{\Delta t^{p-2}}{(p-2)!}\overset{p}{\mathbf{a}}_n + \cdots
\end{aligned} \tag{18.56}$$

Inserting the above into Eq. (18.53) gives

$$\overset{p}{\mathbf{a}}_{n+1} = -\mathbf{A}^{-1}\left\{\mathbf{M}\ddot{\bar{\mathbf{a}}}_{n+1} + \mathbf{C}\dot{\bar{\mathbf{a}}}_{n+1} + \mathbf{K}\bar{\mathbf{a}}_{n+1} + \mathbf{f}_{n+1}\right\} \tag{18.57}$$

where

$$\mathbf{A} = \frac{\beta_{p-2}\Delta t^{p-2}}{(p-2)!}\mathbf{M} + \frac{\beta_{p-1}\Delta t^{p-1}}{(p-1)!}\mathbf{C} + \frac{\beta_p \Delta t^p}{p!}\mathbf{K}$$

Solving the above equation for $\overset{p}{\mathbf{a}}_{n+1}$, we have

$$\overset{p}{\mathbf{a}}_{n+1} = -\mathbf{A}[\mathbf{M}\breve{\ddot{\mathbf{a}}}_{n+1} + \mathbf{C}\breve{\dot{\mathbf{a}}}_{n+1} + \mathbf{K}\breve{\mathbf{a}}_{n+1} + \mathbf{f}_{n+1}] \tag{18.58}$$

We note immediately that the above expression is formally identical to that of the SSpj algorithm, Eq. (18.41), if we make the substitutions

$$\beta_p = \theta_p \qquad \beta_{p-1} = \theta_{p-1} \qquad \beta_{p-2} = \theta_{p-2} \tag{18.59}$$

However, $\breve{\mathbf{a}}_{n+1}$, $\breve{\dot{\mathbf{a}}}_{n+1}$, etc., in the generalized Newmark, GNpj, are not identical to $\bar{\mathbf{a}}_{n+1}$, $\bar{\dot{\mathbf{a}}}_{n+1}$, etc., in the SSpj algorithms. In the SSpj algorithm these represent predicted mean values in the interval Δt while in the GNpj algorithms they represent predicted values at t_{n+1}.

The computation procedure for the GN algorithms is very similar to that for the SS algorithms, starting now with known values of $\mathbf{a}_n$ to $\overset{p}{\mathbf{a}}_n$. As before we have the given initial conditions and we can usually arrange to use the differential equation and its derivatives to generate higher derivatives for $\mathbf{a}$ at $t = 0$. However, the GN algorithm requires more storage because of the necessity of retaining and using $\overset{p}{\mathbf{a}}_0$ in the computation of the next time step.

An important member of this family is the GN22 algorithm. However, before presenting this in detail we consider another form of the truncated Taylor series expansion which has found considerable use recently, especially in non-linear applications.

An alternative is to use a weighted residual approach with a collocation weight function placed at $t = t_{n+\theta}$ on the governing equation. This gives a generalization to Eq. (18.13) of

$$\frac{1}{\Delta t}\mathbf{M}(\dot{\mathbf{a}}_{n+1} - \dot{\mathbf{a}}_n) + \mathbf{C}\dot{\mathbf{a}}_{n+\theta} + \mathbf{K}\mathbf{a}_{n+\theta} + \mathbf{f}_{n+\theta} = \mathbf{0} \tag{18.60}$$

where an interpolated value for $\mathbf{a}_{n+\theta}$ and $\dot{\mathbf{a}}_{n+\theta}$ may be written as

$$\begin{aligned} \mathbf{a}_{n+\theta} &= \mathbf{a}_n + \theta(\mathbf{a}_{n+1} - \mathbf{a}_n) \\ \dot{\mathbf{a}}_{n+\theta} &= \dot{\mathbf{a}}_n + \theta(\dot{\mathbf{a}}_{n+1} - \dot{\mathbf{a}}_n) \end{aligned} \tag{18.61}$$

This form may be combined with a weighted residual approach as described in reference 16. A collocation algorithm for this form is generalized in references 43–46. An advantage of this latter form is an option which permits the generation of energy and momentum conserving properties in the discrete dynamic problem. These generalizations are similar to the GNpj algorithm described in this section although the optimal parameters are usually different.

The Newmark algorithm (GN22)

We have already mentioned the classical Newmark algorithm as it is one of the most popular for dynamic analysis. It is indeed a special case of the general algorithm of the preceding section in which a quadratic ($p = 2$) expansion is used, this being the minimum required for second-order problems. We describe here the details in view of its widespread use.

The expansion of Eq. (18.54) for $p = 2$ gives

$$\begin{aligned}\mathbf{a}_{n+1} &= \mathbf{a}_n + \Delta t\dot{\mathbf{a}}_n + \tfrac{1}{2}(1-\beta_2)\Delta t^2\ddot{\mathbf{a}}_n + \tfrac{1}{2}\beta_2\Delta t^2\ddot{\mathbf{a}}_{n+1} = \dot{\mathbf{a}}_{n+1} + \tfrac{1}{2}\beta_2\Delta t^2\ddot{\mathbf{a}}_{n+1}\\ \dot{\mathbf{a}}_{n+1} &= \dot{\mathbf{a}}_n + (1-\beta_1)\Delta t\ddot{\mathbf{a}}_n + \beta_1\Delta t\ddot{\mathbf{a}}_{n+1} = \ddot{\mathbf{a}}_{n+1} + \beta_1\Delta t\ddot{\mathbf{a}}_{n+1}\end{aligned} \tag{18.62}$$

and this together with the dynamic equation (18.53),

$$\mathbf{M}\ddot{\mathbf{a}}_{n+1} + \mathbf{C}\dot{\mathbf{a}}_{n+1} + \mathbf{K}\mathbf{a}_{n+1} + \mathbf{f}_{n+1} = \mathbf{0} \tag{18.63}$$

allows the three unknowns $\mathbf{a}_{n+1}$, $\dot{\mathbf{a}}_{n+1}$ and $\ddot{\mathbf{a}}_{n+1}$ to be determined.

We now proceed as we have already indicated and solve first for $\ddot{\mathbf{a}}_{n+1}$ by substituting (18.62) into (18.63). This yields as the first step

$$\ddot{\mathbf{a}}_{n+1} = -\mathbf{A}^{-1}\{\mathbf{f}_{n+1} + \mathbf{C}\check{\dot{\mathbf{a}}}_{n+1} + \mathbf{K}\check{\mathbf{a}}_{n+1}\} \tag{18.64}$$

where

$$\mathbf{A} = \mathbf{M} + \beta_1\Delta t\mathbf{C} + \tfrac{1}{2}\beta_2\Delta t^2\mathbf{K} \tag{18.65}$$

After this step the values of $\mathbf{a}_{n+1}$ and $\dot{\mathbf{a}}_{n+1}$ can be found using Eqs (18.62).

As in the general case, $\beta_2 = 0$ produces an explicit algorithm whose solution is very simple if $\mathbf{M}$ and $\mathbf{C}$ are assumed diagonal.

It is of interest to remark that the accuracy can be slightly improved and yet the advantages of the explicit form preserved for SS/GN algorithms by a simple iterative process within each time increment. In this, for the GN algorithm, we predict $\mathbf{a}_{n+1}^i$, $\dot{\mathbf{a}}_{n+1}^i$ and $\ddot{\mathbf{a}}_{n+1}^i$ using expressions (18.55) with

$$(\overset{p}{\mathbf{a}}_{n+1})^{i-1}$$

setting for $i = 1$

$$(\overset{p}{\mathbf{a}}_{n+1})^0 = \mathbf{0}$$

This is followed by rewriting the governing equation (18.57) as

$$\mathbf{M}\left[\ddot{\mathbf{a}}_{n+1}^{i-1} + \frac{\beta_2\Delta t^{p-2}}{(p-2)!}\overset{p}{\mathbf{a}}_{n+1}^{i}\right] + \mathbf{C}\dot{\mathbf{a}}_{n+1}^{i-1} + \mathbf{K}\mathbf{a}_{n+1}^{i-1} + \mathbf{f}_{n+1} = \mathbf{0} \tag{18.66}$$

and solving for $\overset{p}{\mathbf{a}}_{n+1}^{i}$.

This predictor–corrector iteration has been successfully used for various algorithms, though of course the stability conditions remain unaltered from those of a simple explicit scheme.[47]

For implicit schemes we note that in the general case, Eqs (18.62) have scalar coefficients while Eq. (18.63) has matrix coefficients. Thus, for the implicit case some users prefer a slightly more complicated procedure than indicated above in which the first unknown determined is $\mathbf{a}_{n+1}$. This may be achieved by expressing

Eqs (18.62) in terms of the $\mathbf{a}_{n+1}$ to obtain

$$\begin{aligned}\ddot{\mathbf{a}}_{n+1} &= \ddot{\hat{\mathbf{a}}}_{n+1} + \frac{2}{\beta_2 \Delta t^2}\mathbf{a}_{n+1}\\ \dot{\mathbf{a}}_{n+1} &= \dot{\hat{\mathbf{a}}}_{n+1} + \frac{2\beta_1}{\beta_2 \Delta t}\mathbf{a}_{n+1}\end{aligned} \tag{18.67}$$

where

$$\begin{aligned}\ddot{\hat{\mathbf{a}}}_{n+1} &= -\frac{2}{\beta_2 \Delta t^2}\mathbf{a}_n - \frac{2}{\beta_2 \Delta t}\dot{\mathbf{a}}_n - \frac{1-\beta_2}{\beta_2}\ddot{\mathbf{a}}_n\\ \dot{\hat{\mathbf{a}}}_{n+1} &= -\frac{2\beta_1}{\beta_2 \Delta t}\mathbf{a}_n + \left(1 - \frac{2\beta_1}{\beta_2}\right)\dot{\mathbf{a}}_n + \left(1 - \frac{\beta_1}{\beta_2}\right)\Delta t \ddot{\mathbf{a}}_n\end{aligned} \tag{18.68}$$

These are now substituted into Eq. (18.63) to give the result

$$\mathbf{a}_{n+1} = -\mathbf{A}^{-1}\,(\mathbf{f}_{n+1} + \mathbf{C}\dot{\hat{\mathbf{a}}}_{n+1} + \mathbf{M}\ddot{\hat{\mathbf{a}}}_{n+1}) \tag{18.69}$$

where now

$$\mathbf{A} = \frac{2}{\beta_2 \Delta t^2}\mathbf{M} + \frac{2\beta_1}{\beta_2 \Delta t}\mathbf{C} + \mathbf{K}$$

which again on using Eqs (18.67) and (18.68) gives $\dot{\mathbf{a}}$ and $\ddot{\mathbf{a}}$. The inversion is here identical to within a scalar multiplier but as mentioned before precludes use of the explicit form where β_2 is zero.

18.3.4 Stability of general algorithms

Consistency of the general algorithms of SS and GN type is self-evident and assured by their formulation.

In a similar manner to that used in Sec. 18.2.5 we can conclude from this that the *local truncation error* is $O(\Delta t^{p+1})$ as the expansion contains all terms up to τ^p. However, the total truncation error after n steps is only $O(\Delta t^p)$ for first-order equation system and $O(\Delta t^{p-1})$ for the second-order one. Details of accuracy discussions and reasons for this can be found in reference 6.

The question of stability is paramount and in this section we shall discuss it in detail for the SS type of algorithms. The establishment of similar conditions for the GN algorithms follows precisely the same pattern and is left as an exercise to the reader. It is, however, important to remark here that it can be shown that

(a) the SS and GN algorithms are generally similar in performance;
(b) *their stability conditions are identical when* $\theta_p \equiv \beta_p$.

The proof of the last statement requires some elaborate algebra and is given in reference 6.

The determination of stability requirements follows precisely the pattern outlined in Sec. 18.2.5. However for practical reasons we shall

(a) avoid writing explicitly the amplification matrix $\mathbf{A}$;

(b) immediately consider the scalar equation system implying modal decomposition and no forcing, i.e.,

$$m\ddot{a} + c\dot{a} + ka = 0 \tag{18.70}$$

Equations (18.39), (18.40) and (18.42) written in scalar terms define the recurrence algorithms. For the homogeneous case the general solution can be written down as

$$\begin{aligned} a_{n+1} &= \mu a_n \\ \dot{a}_{n+1} &= \mu \dot{a}_n \\ &\vdots \\ {}^{p-1}a_{n+1} &= \mu\, {}^{p-1}a_n \end{aligned} \tag{18.71}$$

and substitution of the above into the equations governing the recurrence can be written quite generally as

$$\mathbf{S}\mathbf{X}_n = \mathbf{0} \tag{18.72}$$

where

$$\mathbf{X}_n = \begin{Bmatrix} a_n \\ \Delta t \dot{a}_n \\ \vdots \\ \Delta t^p\, {}^{p}a_n \end{Bmatrix} \tag{18.73}$$

The matrix $\mathbf{S}$ is given below in a compact form which can be verified by the reader:

$$\mathbf{S} = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{p-1} & b_p \\ 1-\mu & 1 & \dfrac{1}{2!} & \cdots & \dfrac{1}{(p-1)!} & \dfrac{1}{p!} \\ 0 & 1-\mu & 1 & \cdots & \dfrac{1}{(p-2)!} & \dfrac{1}{(p-1)!} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \dfrac{1}{2!} \\ 0 & 0 & 0 & \cdots & 1-\mu & 1 \end{bmatrix} \tag{18.74}$$

where

$$\begin{aligned} b_0 &= \theta_0 \Delta t^2 k, \qquad \theta_0 = 1 \\ b_1 &= \theta_0 \Delta t c + \theta_1 \Delta t^2 k \\ b_q &= \frac{\theta_{q-2}}{(q-2)!} m + \frac{\theta_{q-1}\Delta t}{(q-1)!} c + \frac{\theta_q \Delta t^2}{q!} k, \qquad q = 2, 3, \ldots, p \end{aligned}$$

For non-trivial solutions for the vector $\mathbf{X}_n$ to exist it is necessary for the determinant of $\mathbf{S}$ to be zero:

$$\det \mathbf{S} = 0 \tag{18.75}$$

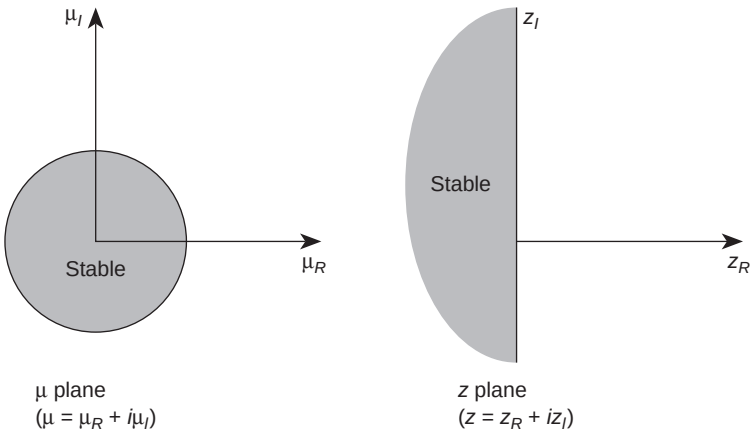


Fig. 18.11 The $\mu = (1+z)/(1-z)$ transformation.

This provides a *characteristic polynomial* of order p for μ which yields the eigenvalues of the amplification matrix. For stability it is sufficient and necessary that the moduli of all eigenvalues [see Eq. (18.28)] satisfy

$$|\mu| \leqslant 1 \tag{18.76}$$

We remark that in the case of repeated roots the equality sign does not apply. The reader will have noticed that the direct derivation of the determinant of $\mathbf{S}$ is much simpler than writing down matrix $\mathbf{A}$ and finding the eigenvalues. The results are, of course, identical.

The calculation of stability limits, even with the scalar (modal) equation system, is non-trivial. For this reason in what follows we shall only do it for $p = 2$ and $p = 3$. However, two general procedures will be introduced here.

The first of these is the so-called *z transformation*. In this we use a change of variables in the polynomial putting

$$\mu = \frac{1+z}{1-z} \tag{18.77}$$

where z as well as μ are in general complex numbers. It is easy to show that the requirement of Eq. (18.76) is identical to that demanding the *real part of z to be negative* (see Fig. 18.11).

The second procedure introduced is the well-known Routh–Hurwitz condition[48–50] which states that for a polynomial with $c_0 > 0$

$$c_0 z^n + c_1 z^{n-1} + \cdots + c_{n-1} z + c_n = 0 \tag{18.78}$$

the real part of all roots will be negative if, for $c_1 > 0$,

$$\det \begin{bmatrix} c_1 & c_3 \\ c_0 & c_2 \end{bmatrix} > 0 \qquad \det \begin{bmatrix} c_1 & c_3 & c_5 \\ c_0 & c_2 & c_4 \\ 0 & c_1 & c_3 \end{bmatrix} > 0 \tag{18.79}$$

and generally

$$\det\begin{bmatrix} c_1 & c_3 & c_5 & c_7 & \cdots & \\ c_0 & c_2 & c_4 & c_6 & \cdots & \\ 0 & c_1 & c_3 & c_5 & \cdots & \\ 0 & 0 & c_2 & c_4 & \cdots & \\ \vdots & & & & \ddots & \\ 0 & 0 & 0 & 0 & \cdots & c_n \end{bmatrix} > 0 \tag{18.80}$$

With these tools in hand we can discuss in detail the stability of specific algorithms.

18.3.5 Stability of SS22/SS21 algorithms

The recurrence relations for the algorithm given in Eqs (18.49) and (18.51) can be written after inserting

$$a_{n+1} = \mu a_n \qquad \dot{a}_{n+1} = \mu \dot{a}_n \qquad f = 0 \tag{18.81}$$

as

$$m\alpha + c(\dot{a}_n + \theta_1 \Delta t \alpha) + k\left(a_n + \Delta t \dot{a}_n + \tfrac{1}{2}\theta_2 \Delta t^2 \alpha\right) = 0$$

$$-\mu a_n + a_n + \Delta t a_n + \tfrac{1}{2}\theta_2 \Delta t^2 \alpha = 0 \tag{18.82}$$

$$-\mu \dot{a}_n + \dot{a}_n + \Delta t \ddot{a}_n + \theta_1 \Delta t \alpha = 0 \tag{18.83}$$

Changing the variable according to Eq. (18.77) results in the characteristic polynomial

$$c_0 z^2 + c_1 z + c_2 = 0 \tag{18.84}$$

with

$$c_0 = 4m + (4\theta_1 - 2)\Delta t c + 2(\theta_2 - \theta_1)\Delta t^2 k$$

$$c_1 = 2\Delta t c + (2\theta_1 - 1)\Delta t^2 k$$

$$c_2 = \Delta t^2 k$$

The Routh–Hurwitz requirements for stability is simply that

$$c_0 > 0 \qquad c_1 \geqslant 0 \qquad \det\begin{bmatrix} c_1 & 0 \\ c_0 & c_2 \end{bmatrix} > 0$$

or simply

$$c_0 > 0 \qquad c_1 \geqslant 0 \qquad c_2 > 0 \tag{18.85}$$

These inequalities give for *unconditional stability* the condition that

$$\theta_2 \geqslant \theta_1 \geqslant \tfrac{1}{2} \tag{18.86}$$

This condition is also generally valid when $m = 0$, i.e., for the SS21 algorithm (the first-order equation) though now $\theta_2 = \theta_1$ must be excluded.

It is possible to satisfy the inequalities (18.85) only at some values of Δt yielding conditional stability. For the explicit process $\theta_2 = 0$ with SS22/SS21 algorithms the inequalities (18.85) demand that

$$\begin{aligned} 2m + (2\theta_1 - 1)\Delta t c - \theta_1 \Delta t^2 k \geqslant 0 \\ 2c + (2\theta_1 - 1)\Delta t k \geqslant 0 \end{aligned} \tag{18.87}$$

The second one is satisfied whenever

$$\theta_1 \geqslant \tfrac{1}{2} \tag{18.88}$$

and for $\theta_1 = 1/2$ the first supplies the requirement that

$$\Delta t^2 \leqslant \frac{4m}{k} \tag{18.89}$$

The last condition does not permit an explicit scheme for SS21, i.e., when $m = 0$. Here, however, if we take $\theta_1 > 1/2$ we have from the first equation of Eq. (18.87)

$$\Delta t < \frac{2\theta_1 - 1}{\theta_1} \frac{c}{k} \tag{18.90}$$

It is of interest for problems of structural dynamics to consider the nature of the bounds in an elastic situation. Here we can use the same process as that described in Sec. 18.2.5 for first-order problems of heat conduction. Looking at a single element with a single degree of freedom and consistent mass yields in place of condition (18.89)

$$\Delta t \leqslant \frac{2}{\sqrt{3}} \frac{h}{C} = \Delta t_{\text{crit}}$$

where h is the element size and

$$C = \sqrt{\frac{E}{\rho}}$$

is the speed of elastic wave propagation. For lumped mass matrices the factor becomes $\sqrt{2}$.

Once again the ratio of the smallest element size over wave speed governs the stability but it is interesting to note that in problems of dynamics the critical time step is proportional to h while, as shown in Eq. (18.33), for first-order problems it is proportional to h^2. Clearly for decreasing mesh size explicit schemes in dynamics are more efficient than in thermal analysis and are exceedingly popular in certain classes of problems.

18.3.6 Stability of various higher order schemes and equivalence with some known alternatives

Identical stability considerations as those described in previous sections can be applied to SS32/SS31 and higher order approximations. We omit here the algebra and simply quote some results.[6]

Table 18.2 SS21 equivalents

Algorithms	Theta values
Zlamal[38]	$\theta_1 = \frac{5}{6}$, $\theta_2 = 2$
Gear[52]	$\theta_1 = \frac{3}{2}$, $\theta_2 = 2$
Liniger[53]	$\theta_1 = 1.0848$, $\theta_2 = 1$
Liniger[53]	$\theta_1 = 1.2184$, $\theta_2 = 1.292$

SS32/31. Here for zero damping ($c = 0$) in SS32 we require for unconditional stability that

$$\theta_1 > \tfrac{1}{2} \qquad \theta_2 \geqslant \theta_1 + \tfrac{1}{6} \qquad \theta_2 \geqslant \tfrac{1}{4} \qquad \theta_3 \geqslant \tfrac{3}{2}$$
$$3\theta_1\theta_2 - 3\theta_1^2 + \theta_1 \geqslant \theta_3 \tag{18.91}$$

For first-order problems ($m = 0$), i.e., SS31, the first requirements are as in dynamics but the last one becomes

$$3\theta_1\theta_1^2 - 3\theta_2 + \theta_1 \geqslant \theta_3 - \frac{[6\theta_1(\theta_1 - 1) + 1]^2}{9(2\theta_1 - 1)} \tag{18.92}$$

With $\theta_3 = 0$, i.e., an explicit scheme when $c = 0$,

$$\Delta t^2 \leqslant \frac{12(2\theta_1 - 1)}{6\theta_2 - 1}\frac{m}{k} \tag{18.93}$$

and when $m = 0$,

$$\Delta t \leqslant \frac{\theta_2 - \theta_1}{6\theta_2 - 1}\frac{c}{k} \tag{18.94}$$

SS42/41. For this (and indeed higher orders) unconditional stability in dynamics problems $m \neq 0$ does not exist. This is a consequence of a theorem by Dahlquist.[51] The SS41 scheme can have unconditional stability but the general expressions for this are cumbersome. We quote one example that is unconditionally stable:

$$\theta_1 = \tfrac{5}{2} \qquad \theta_2 = \tfrac{35}{6} \qquad \theta_3 = \tfrac{25}{2} \qquad \theta_4 = 24$$

This set of values corresponds to a backward difference four-step algorithm of Gear.[52]

It is of general interest to remark that certain members of the SS (or GN) families of algorithms are similar in performance and identical in the stability (and hence recurrence) properties to others published in the large literature on the subject. Each algorithm claims particular advantages and properties. In Tables 18.2–18.4 we show some members of this family.[38–57] Clearly many more algorithms that are

Table 18.3 SS31 equivalents

Algorithms	Theta values
Gear[52]	$\theta_1 = 2$, $\theta_2 = \frac{11}{3}$, $\theta_3 = 6$
Liniger[53]	$\theta_1 = 1.84$, $\theta_2 = 3.07$, $\theta_3 = 4.5$
Liniger[53]	$\theta_1 = 0.8$, $\theta_2 = 1.03$, $\theta_3 = 1.29$

Table 18.4 SS32 equivalents

Algorithms	Theta values
Houbolt[54]	$\theta_1 = 2,\ \theta_2 = \frac{11}{3},\ \theta_3 = 6$
Wilson Θ[55]	$\theta_1 = \Theta,\ \theta_2 = \Theta^2,\ \theta_3 = \Theta^3$ ($\Theta = 1.4$ common)
Bossak–Newmark[56] ($m\ddot{a} + ka = 0$, $\gamma_B = \frac{1}{2} - \alpha_B$)	$\theta_1 = 1 - \alpha_B$ $\theta_2 = \frac{2}{3} - \alpha_B + 2\beta_B$ $\theta_3 = 6\beta_B$
Bossak–Newmark[56] ($m\ddot{a} + c\dot{a} + ka = 0$, $\gamma_B = \frac{1}{2} - \alpha_B$, $\beta_B = \frac{1}{6} - \frac{1}{2}\alpha_B$)	$\theta_1 = 1 - \alpha_B$ $\theta_2 = 1 - 2\alpha_B$ $\theta_3 = 1 - 3\alpha_B$
Hilber–Hughes–Taylor[57] ($m\ddot{a} + ka = 0$, $\gamma_H = \frac{1}{2} - \alpha_H$)	$\theta_1 = 1$ $\theta_2 = \frac{2}{3} + 2\beta_H - 2\alpha_H^2$ $\theta_3 = 6\beta_H(1 + \alpha_H)$

applicable are present in the general formulae and a study of their optimal parameters is yet to be performed.

We remark here that identity of stability and recurrence always occurs with multistep algorithms, which we shall discuss in the next section.

Multistep methods

18.4 Multistep recurrence algorithms

18.4.1 Introduction

In the previous sections we have been concerned with recurrence algorithms valid within a single time step and relating the values of $\mathbf{a}_{n+1}$, $\dot{\mathbf{a}}_{n+1}$, $\ddot{\mathbf{a}}_{n+1}$ to $\mathbf{a}_n$, $\dot{\mathbf{a}}_n$, $\ddot{\mathbf{a}}_n$, etc. It is possible to derive, using very similar procedures to those previously introduced, multistep algorithms in which we relate $\mathbf{a}_{n+1}$ to the values $\mathbf{a}_n$, $\mathbf{a}_{n-1}$, $\mathbf{a}_{n-2}$, etc., without explicitly introducing the derivatives. Much classical work on stability and accuracy has been introduced on such multistep algorithms and hence they deserve mention here.

We shall show in this section that a series of such algorithms may be simply derived using the weighted residual process and that, for constant time increments Δt, this set possesses identical stability and accuracy properties to the SSpj procedures.

18.4.2 The approximation procedure for a general multistep algorithm

As in Sec. 18.3.2 we shall approximate the function $\mathbf{a}$ of the second-order equation

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{18.95}$$

by a polynomial expansion of the order p, now containing a single unknown $\mathbf{a}_{n+1}$. This polynomial assumes knowledge of the value of $\mathbf{a}_n$, $\mathbf{a}_{n-1}$, …, $\mathbf{a}_{n-p+1}$ at appropriate times t_n, t_{n-1}, …, t_{n-p+1} (Fig. 18.12).

We can write this polynomial as

$$\mathbf{a}(t) = \sum_{j=1-p}^{1} N_j(t)\mathbf{a}_{n+j} \tag{18.96}$$

where Lagrange interpolation in time is given by (see Chapter 8)

$$N_j(t) = \prod_{\substack{k=1-p \\ k\neq j}}^{1} \frac{t - t_{n+k}}{t_{n+j} - t_{n+k}} \tag{18.97}$$

The derivatives of the shape functions may be constructed by writing

$$N_j = \frac{n_j(t)}{n_j(t_{n+j})} \tag{18.98}$$

and differentiating the numerator. Accordingly writing

$$n_j(t) = \prod_{\substack{k=1-p \\ k\neq j}}^{1} (t - t_{n+k}) \tag{18.99}$$

the derivative becomes

$$\frac{\mathrm{d}n_j}{\mathrm{d}t} = \sum_{\substack{m=p-1 \\ m\neq j}}^{1} \prod_{\substack{k=p-1 \\ k\neq j \\ k\neq m}}^{1} (t - t_{n+k}) \quad p \geqslant 2 \tag{18.100}$$

Now

$$\frac{\mathrm{d}N_j}{\mathrm{d}t} = \frac{1}{n_j(t_{n+j})}\frac{\mathrm{d}n_j}{\mathrm{d}t} = \dot{N}_j \tag{18.101}$$

These expressions can be substituted into Eq. (18.84) giving

$$\begin{aligned} \dot{\mathbf{a}} &= \sum_{j=1-p}^{1} \dot{N}_j(t)\mathbf{a}_{n+j} \\ \ddot{\mathbf{a}} &= \sum_{j=1-p}^{1} \ddot{N}_j(t)\mathbf{a}_{n+j} \end{aligned} \tag{18.102}$$

Insertion of $\mathbf{a}$, $\dot{\mathbf{a}}$ and $\ddot{\mathbf{a}}$ into the weighted residual equation form of Eq. (18.83) yields

$$\int_{t_n}^{t_{n+1}} W(t) \sum_{j=1-p}^{1} \left[\left(\ddot{N}_j\mathbf{M} + \dot{N}_j\mathbf{C} + N_j\mathbf{K}\right)\mathbf{a}_{n+j} + N_j\mathbf{f}_{n+j}\right] \mathrm{d}t = 0 \tag{18.103}$$

with the forcing functions interpolated similarly from its nodal values.

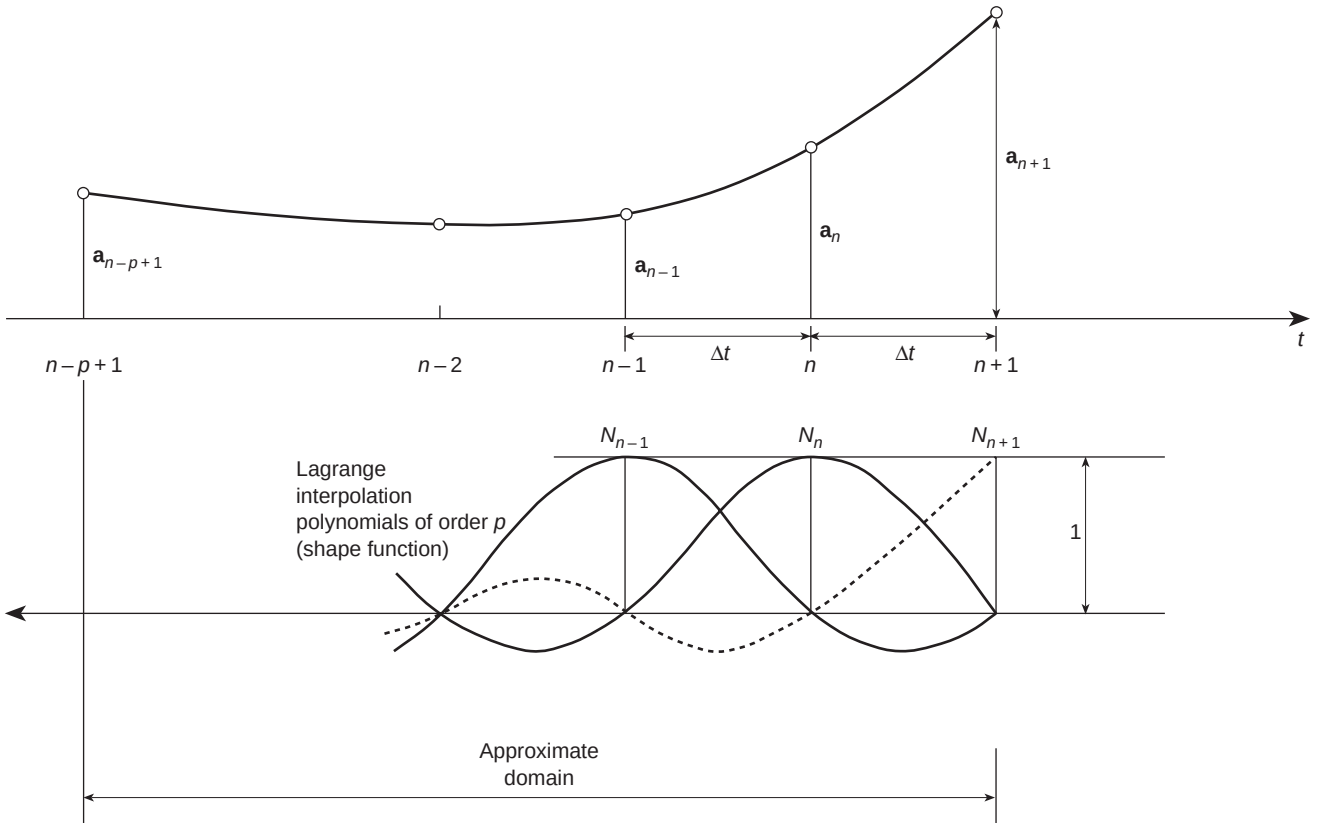


Fig. 18.12 Multistep polynomial approximation.

Two point interpolation: p = 1

Evaluating Eq. (18.85) we obtain

$$N_1 = \frac{t - t_{n+1}}{t_{n+1} - t_n} = \frac{1}{\Delta t}(t - t_{n+1}) = \frac{\tau}{\Delta t}$$

$$N_0 = \frac{t - t_{n+1}}{t_n - t_{n+1}} = \frac{1}{\Delta t}(t_{n+1} - t) = 1 - \frac{\tau}{\Delta t}$$

where $\Delta t = t_{n+1} - t_n$ and $\tau = t - t_n$. Here the derivative is computed directly as

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = -\frac{\mathrm{d}N_0}{\mathrm{d}t} = \frac{1}{\Delta t}$$

Second derivatives are obviously zero, hence this form may only be used for first-order equations.

Three point interpolation: p = 2

Evaluating Eq. (18.85)

$$N_1 = \frac{(t - t_{n-1})(t - t_n)}{(t_{n+1} - t_{n-1})(t_{n+1} - t_n)}$$

$$N_0 = \frac{(t - t_{n-1})(t - t_{n+1})}{(t_n - t_{n-1})(t_n - t_{n+1})}$$

$$N_{-1} = \frac{(t - t_n)(t - t_{n+1})}{(t_{n-1} - t_n)(t_{n-1} - t_{n+1})}$$

The derivatives follow immediately from Eqs (18.87) and (18.88) as

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = \frac{(t - t_n) + (t - t_{n-1})}{(t_{n+1} - t_{n-1})(t_{n+1} - t_n)}$$

$$\frac{\mathrm{d}N_0}{\mathrm{d}t} = \frac{(t - t_{n+1}) + (t - t_{n-1})}{(t_n - t_{n-1})(t_n - t_{n+1})}$$

$$\frac{\mathrm{d}N_{-1}}{\mathrm{d}t} = \frac{(t - t_{n+1}) + (t - t_n)}{(t_{n-1} - t_n)(t_{n-1} - t_{n+1})}$$

This is the lowest order which can be used for second-order equations and has second derivatives

$$\frac{\mathrm{d}^2N_1}{\mathrm{d}t^2} = \frac{2}{(t_{n+1} - t_{n-1})(t_{n+1} - t_n)}$$

$$\frac{\mathrm{d}^2N_0}{\mathrm{d}t^2} = \frac{2}{(t_n - t_{n-1})(t_n - t_{n+1})}$$

$$\frac{\mathrm{d}^2N_{-1}}{\mathrm{d}t^2} = \mathrm{frac}\,2\,\frac{2}{(t_{n-1} - t_n)(t_{n-1} - t_{n+1})}$$

18.4.3 Constant Δt form

For the remainder of our discussion here we shall assume a constant time increment for all steps is given by Δt. To develop the constant increment form we introduce the natural coordinate ξ defined as

$$\begin{aligned} &\xi = \frac{t - t_n}{\Delta t}, \qquad 1 - p \leqslant \xi \leqslant 1 \\ &j = 1 - p, 2 - p, \ldots, 0, 1 \end{aligned} \tag{18.104}$$

and now assume the shape functions N_j in Eq. (18.84) are functions of the natural coordinates and given by

$$N_j(\xi) = \frac{n_j(\xi)}{n_j(j)} \tag{18.105}$$

where

$$n_j(\xi) = \prod_{\substack{k=1-p \\ k \neq j}}^{1} (\xi - k)$$

$$n_j(j) = \prod_{\substack{k=1-p \\ k \neq j}}^{1} (j - k)$$

Derivatives with respect to ξ are given by

$$n_j'(\xi) = \sum_{\substack{k=1-p \\ k \neq j}}^{1} \prod_{\substack{l=1-p \\ l \neq k \\ l \neq j}}^{1} (\xi - l) \tag{18.106}$$

$$n_j''(\xi) = \sum_{\substack{k=1-p \\ k \neq j}}^{1} \sum_{\substack{l=1-p \\ l \neq k \\ l \neq j}}^{1} \prod_{\substack{m=1-p \\ m \neq k \\ m \neq l \\ m \neq j}}^{1} (\xi - m) \tag{18.107}$$

Using the chain rule these derivatives give the time derivatives

$$\dot{\mathbf{a}} = \frac{1}{\Delta t} \sum_{j=1-p}^{1} \frac{n_j'}{n_j(j)} \mathbf{a}_{n+k} \tag{18.108}$$

$$\ddot{\mathbf{a}} = \frac{1}{\Delta t^2} \sum_{j=1-p}^{1} \frac{n_j''}{n_j(j)} \mathbf{a}_{n+k} \tag{18.109}$$

The weighted residual equation may now be written as

$$\int_0^1 W(\xi) \sum_{j=1-p}^{1} \left[\left(N_j'' \mathbf{M} + \Delta t N_j' \mathbf{C} + \Delta t^2 N_j \mathbf{K} \right) \mathbf{a}_{n+j} + \Delta t^2 N_j \mathbf{f}_{n+j} \right] \mathrm{d}\xi = 0 \tag{18.111}$$

Using the parameters

$$\phi_q = \frac{\int_0^1 W\xi^q \,\mathrm{d}\xi}{\int_0^1 W \,\mathrm{d}\xi}, \qquad q = 1, 2, \ldots, p; \quad \phi_0 = 1 \tag{18.112}$$

we now have an algorithm that enables us to compute $\mathbf{a}_{n+1}$ from known values $\mathbf{a}_{n-p+1}$, $\mathbf{a}_{n-p+2}$, …, $\mathbf{a}_n$. [*Note*: so long as the limits of integration are the *same* in Eqs (18.111) and (18.112) it makes no difference what we choose them to be.]

Four-point interpolation: p = 3

For $p = 3$, Eq. (18.93) gives

$$\begin{aligned} N_{-2}(\xi) &= -\tfrac{1}{6}\left(\xi^3 - \xi\right) \\ N_{-1}(\xi) &= \tfrac{1}{2}\left(\xi^3 + \xi^2 - 2\xi\right) \end{aligned} \tag{18.113}$$

$$\begin{aligned} N_0(\xi) &= -\tfrac{1}{2}\left(\xi^3 + 2\xi^2 - \xi - 2\right) \\ N_1(\xi) &= \tfrac{1}{6}\left(\xi^3 + 3\xi^2 + 2\xi\right) \end{aligned} \tag{18.114}$$

Similarly, from Eqs (18.94) and (18.95),

$$\begin{aligned} N'_{-2}(\xi) &= -\tfrac{1}{6}\left(3\xi^2 - 1\right) \\ N'_{-1}(\xi) &= \tfrac{1}{2}\left(3\xi^2 + 2\xi - 2\right) \end{aligned} \tag{18.115}$$

$$\begin{aligned} N'_0(\xi) &= -\tfrac{1}{2}\left(3\xi^2 + 4\xi - 1\right) \\ N'_1(\xi) &= \tfrac{1}{6}\left(3\xi^2 + 6\xi + 2\right) \end{aligned} \tag{18.116}$$

and

$$\begin{aligned} N''_{-2}(\xi) &= -\xi \\ N''_{-1}(\xi) &= 3\xi + 1 \end{aligned} \tag{18.117}$$

$$\begin{aligned} N''_0(\xi) &= -3\xi - 2 \\ N''_1(\xi) &= \xi + 1 \end{aligned} \tag{18.118}$$

We now have a three-step algorithm for the solution of Eq. (18.83) of the form (taking $\mathbf{f} = \mathbf{0}$)

$$\sum_{j=-2}^{1} \left[\alpha_{j+2}\mathbf{M} + \gamma_{j+2}\Delta t\mathbf{C} + \beta_{j+2}\Delta t^2\mathbf{K}\right]\mathbf{a}_{n+j} = \mathbf{0} \tag{18.119}$$

where

$$\begin{aligned} \alpha_{j+2} &= \int_0^1 W(\xi) N''_j \,\mathrm{d}\xi \\ \gamma_{j+2} &= \int_0^1 W(\xi) N'_j \,\mathrm{d}\xi \\ \beta_{j+2} &= \int_0^1 W(\xi) N_j \,\mathrm{d}\xi \end{aligned} \tag{18.120}$$

After integration the above gives

$$\begin{aligned}
&\alpha_0 = -\phi_1 && \gamma_0 = -\tfrac{1}{6}(3\phi_2 - 1) && \beta_0 = -\tfrac{1}{6}(\phi_3 - \phi_1)\\
&\alpha_1 = 3\phi_1 + 1 && \gamma_1 = \tfrac{1}{2}(3\phi_2 + 2\phi_1 - 2) && \beta_1 = \tfrac{1}{2}(\phi_3 + \phi_2 - 2\phi_1)\\
&\alpha_2 = -3\phi_1 - 2 && \gamma_2 = -\tfrac{1}{2}(3\phi_2 + 2\phi_1 - 1) && \beta_2 = -\tfrac{1}{2}(\phi_3 + 2\phi_2 - \phi_1 - 2)\\
&\alpha_3 = \phi_1 + 1 && \gamma_3 = \tfrac{1}{6}(3\phi_2 + 6\phi_1 + 2) && \beta_3 = \tfrac{1}{6}(\phi_3 + 3\phi_2 + 2\phi_1)
\end{aligned} \tag{18.121}$$

An algorithm of the form given in Eq. (18.119) is called a linear three-step method. The general p-step form is

$$\sum_{1-p}^{1}\left[\alpha_{j+p-1}\mathbf{M} + \gamma_{j+p-1}\Delta t\mathbf{C} + \beta_{j+p-1}\Delta t^2\mathbf{K}\right]\mathbf{a}_{n+j} = \mathbf{0} \tag{18.122}$$

This is the form generally given in mathematics texts; it is an extension of the form given by Lambert[2] for $\mathbf{C} = \mathbf{0}$. The weighted residual approach described here derives the α's, β's and γ's in terms of the parameters ϕ_i, $i = 0, 1, \ldots, p$ and thus ensures consistency.

From Eq. (18.122) the unknown $\mathbf{a}_{n+1}$ is obtained in the form

$$\mathbf{a}_{n+1} = \left[\alpha_3\mathbf{M} + \gamma_3\Delta t\mathbf{C} + \beta_3\Delta t^2\mathbf{K}\right]^{-1}\mathbf{F} \tag{18.123}$$

where $\mathbf{F}$ is expressed in terms of known values. For example, for $p = 3$ the matrix to be inverted is

$$\left[(\phi_1 + 1)\mathbf{M} + (\tfrac{1}{2}\phi_2 + \phi_1 + \tfrac{1}{3})\Delta t\mathbf{C} + (\tfrac{1}{6}\phi_3 + \tfrac{1}{2}\phi_2 + \tfrac{1}{3}\phi_1)\Delta t^2\mathbf{K}\right]$$

Comparing this with the matrix to be inverted in the SSpj algorithm given in Eq. (18.41) suggests a correspondence between SSpj and the p-step algorithm above which we explore further in the next section.

18.4.4 The relationship between SSpj and the weighted residual p-step algorithm

For simplicity we now consider the p-step algorithm described in the previous section applied to the homogeneous scalar equation

$$m\ddot{a} + c\dot{a} + ka = 0 \tag{18.124}$$

As in previous stability considerations we can obtain the general solution of the recurrence relation

$$\sum_{j=1-p}^{1}\left[\alpha_{j+p-1}m + \gamma_{j+p-1}\Delta tc + \beta_{j+p-1}\Delta t^2k\right]a_{n+j} = 0 \tag{18.125}$$

by putting $a_{n+j} = \mu^{p-1+j}$ where the values of μ are the roots μ_k of the stability polynomial of the p-step algorithm:

$$\sum_{j=1-p}^{1}\left[\alpha_{j+p-1}m + \gamma_{j+p-1}\Delta tc + \beta_{j+p-1}\Delta t^2k\right]\mu^{p-1+j} = 0 \tag{18.126}$$

Table 18.5 Identities between SSp2 and p-step algorithms

SS22/21	
	$\theta_1 = \phi_1 + \frac{1}{2}$
	$\theta_2 = \phi_2 + \phi_1$
SS32/31	
	$\theta_1 = \phi_1 + 1$
	$\theta_2 = \phi_2 + 2\phi_1 + \frac{2}{3}$
	$\theta_3 = \phi_3 + 3\phi_2 + 2\phi_1$
SS42/41	
	$\theta_1 = \phi_1 + \frac{3}{2}$
	$\theta_2 = \phi_2 + 3\phi_1 + \frac{11}{6}$
	$\theta_3 = \phi_3 + \frac{9}{2}\phi_2 + \frac{11}{2}\phi_1 + \frac{3}{2}$
	$\theta_4 = \phi_4 + 6\phi_3 + 11\phi_2 + 6\phi_1$

This stability polynomial can be quite generally identified with the one resulting from the determinant of Eq. (18.74) as shown in reference 6, by using a suitable set of relations linking θ_i and ϕ_i. Thus, for instance, in the case of $p = 3$ discussed we shall have the identity of stability and indeed of the algorithm when

$$\begin{aligned} \theta_1 &= \phi_1 + 1 \\ \theta_2 &= \phi_2 + 2\phi_1 + \tfrac{2}{3} \\ \theta_3 &= \phi_3 + 3\phi_2 + 2\phi_1 \end{aligned} \tag{18.127}$$

Table 18.5 summarizes the identities of $p = 2$, 3 and 4.

Many results obtained previously with p-step methods[15,58] can be used to give the accuracy and stability properties of the solution produced by the SSpj algorithms. Tables 18.6 and 18.7 give the accuracy of stable algorithms from the SSp1 and SSp2 families respectively for $p = 2$, 3, 4. Algorithms that are only conditionally stable (i.e., only stable for values of the time step less than some critical value) are marked CS. Details are given in reference 2.

We conclude this section by writing in full the second degree (two-step) algorithm that corresponds precisely to SS22 and GN22 methods. Indeed, it is written below in the form originally derived by Newmark[41]:

$$\begin{aligned} &[\mathbf{M} + \gamma\Delta t\mathbf{C} + \beta\Delta t^2\mathbf{K}]\mathbf{a}_{n+1} \\ &\quad + [-2\mathbf{M} + (1 - 2\gamma)\Delta t\mathbf{C} + (\tfrac{1}{2} - 2\beta + \gamma)\Delta t^2\mathbf{K}]\mathbf{a}_n \\ &\quad + [\mathbf{M} - (1 - \gamma)\Delta t\mathbf{C} + (\tfrac{1}{2} + \beta - \gamma)\Delta t^2\mathbf{K}]\mathbf{a}_{n-1} + \Delta t^2\bar{\mathbf{f}} = \mathbf{0} \end{aligned} \tag{18.128}$$

Table 18.6 Accuracy of SSp1 algorithms

Method	parameters	error
SS11	θ_1	$O(\Delta t)$
	$\theta_1 = \frac{1}{2}$	$O(\Delta t^2)$
SS21	θ_1, θ_2	$O(\Delta t^2)$
	$\theta_1 - \theta_2 = \frac{1}{6}$	$O(\Delta t^3)$CS
SS31	$\theta_1, \theta_2, \theta_3$	$O(\Delta t^3)$
	$\theta_1 - 3\theta_2, +2\theta_3 = 0$	$O(\Delta t^4)$CS
SS41	$\theta_1, \theta_2, \theta_3, \theta_4$	$O(\Delta t^4)$

Table 18.7 Accuracy of SSp2 algorithms

Method	Parameters	Error	
		$C = 0$	$C \neq 0$
SS22	θ_1, θ_2	$O(\Delta t)$	$O(\Delta t)$
	$\theta_1, \theta_2 = \frac{1}{2}$	$O(\Delta t^2)$	$O(\Delta t^2)$
	$\theta_1 = \frac{1}{2}, \theta_2 = \frac{1}{6}$	$O(\Delta t^4)$CS	$O(\Delta t^2)$CS
SS32	$\theta_1, \theta_2, \theta_3$	$O(\Delta t^2)$	$O(\Delta t^2)$
	$\theta_2 = \theta_1 - \frac{1}{6}$	$O(\Delta t^3)$CS	$O(\Delta t^3)$CS
	$\theta_3 = \frac{1}{2}\theta_1$	$O(\Delta t^4)$CS	–
SS42	$\theta_1, \theta_2, \theta_3, \theta_4$	$O(\Delta t^4)$CS	$O(\Delta t^4)$CS

Here of course, we have the original Newmark parameters β, γ, which can be changed to correspond with the SS22/GN22 form as follows:

$$\gamma = \theta_1 = \beta_1 \qquad \beta = \tfrac{1}{2}\theta_2 = \tfrac{1}{2}\beta_2$$

The explicit form of this algorithm ($\beta = \theta_2 = \beta_2 = 0$) is frequently used as an alternative to the single-step explicit form. It is then known as the *central difference approximation* obtained by direct differencing. The reader can easily verify that the simplest finite difference approximation of Eq. (18.1) in fact corresponds to the above with $\beta = 0$ and $\gamma = 1/2$.

18.5 Some remarks on general performance of numerical algorithms

In Secs 18.2.5 and 18.3.3 we have considered the *exact* solution of the approximate recurrence algorithm given in the form

$$a_{n+1} = \mu a_n, \qquad \text{etc.} \tag{18.129}$$

for the modally decomposed, single degree of freedom systems typical of Eqs (18.4) and (18.5). The evaluation of μ was important to ensure that its modulus does not exceed unity and stability is preserved.

However, analytical solution of the linear homogeneous differential equations is also easy to obtain in the form

$$a = \bar{a}\, \mathrm{e}^{\lambda t} \tag{18.130}$$

$$a_{n+1} = a_n\, \mathrm{e}^{\lambda \Delta t} \tag{18.131}$$

and comparison of μ with such a solution is always instructive to provide information on the performance of algorithms in the particular range of eigenvalues.

In Fig. 18.5 we have plotted the exact solution $\mathrm{e}^{-\omega \Delta t}$ and compared it with the values of μ for various θ algorithms approximating the first-order equation, noting that here

$$\lambda = -\omega = -\frac{k}{c} \tag{18.132}$$

and is real.

Immediately we see that there the performance error is very different for various values of Δt and obviously deteriorates at large values. Such values in a real multi-variable problem correspond of course to the 'high-frequency' responses which are often less important, and for smooth solutions we favour algorithms where μ tends to values much less than unity for such problems. However, response through the whole time range is important and attempts to choose an optimal value of θ for various time ranges has been performed by Liniger.[53] Table 18.1 of Sec. 18.2.6 illustrates how an algorithm with $\theta = 2/3$ and a higher truncation error than that

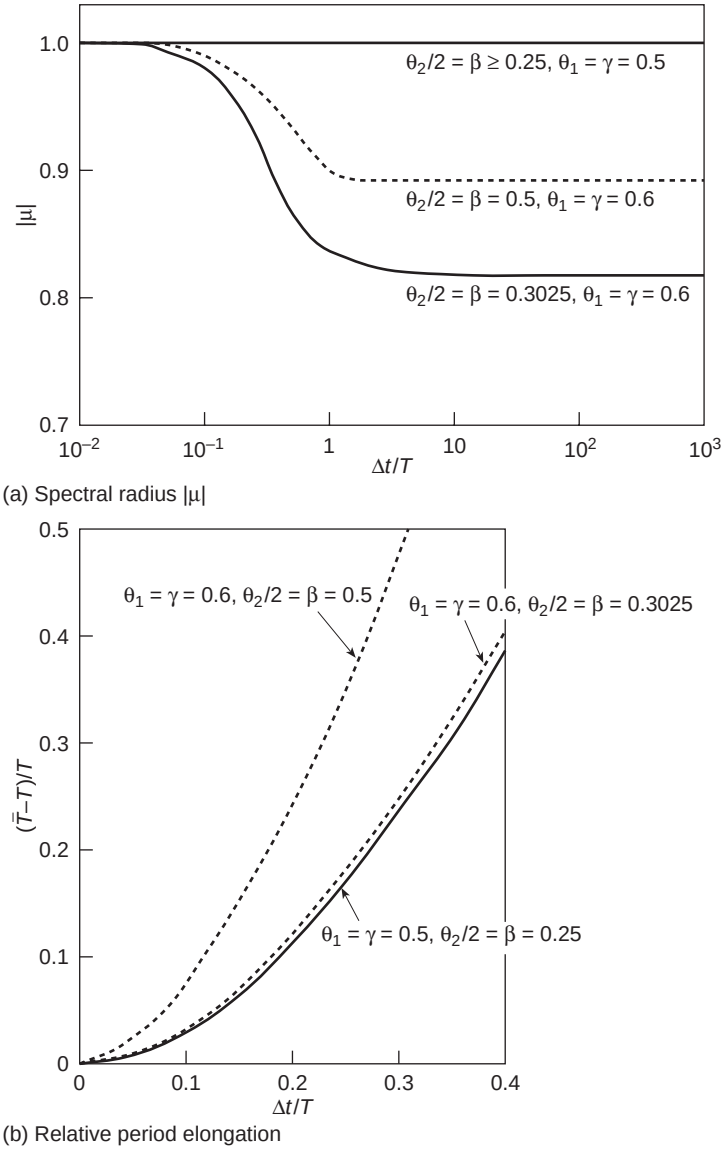
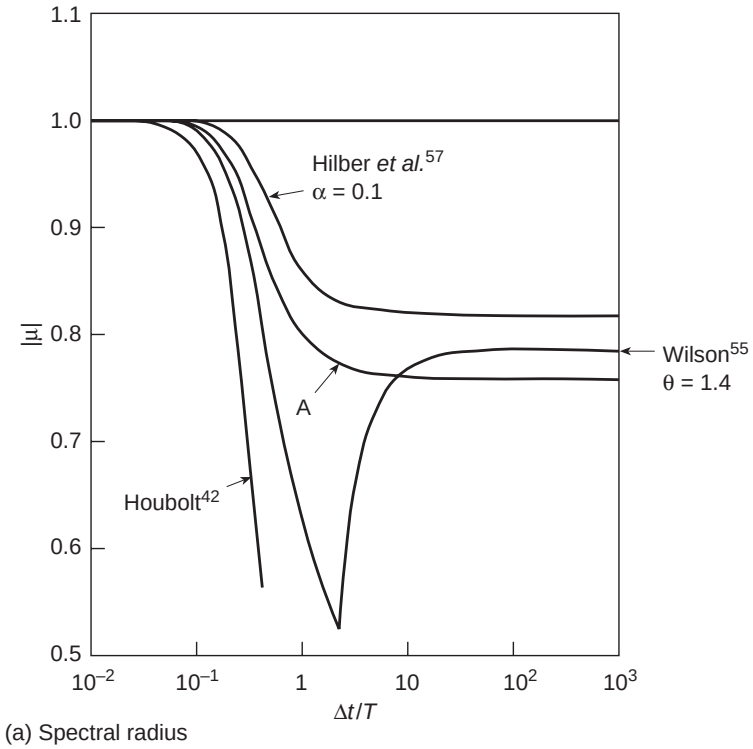
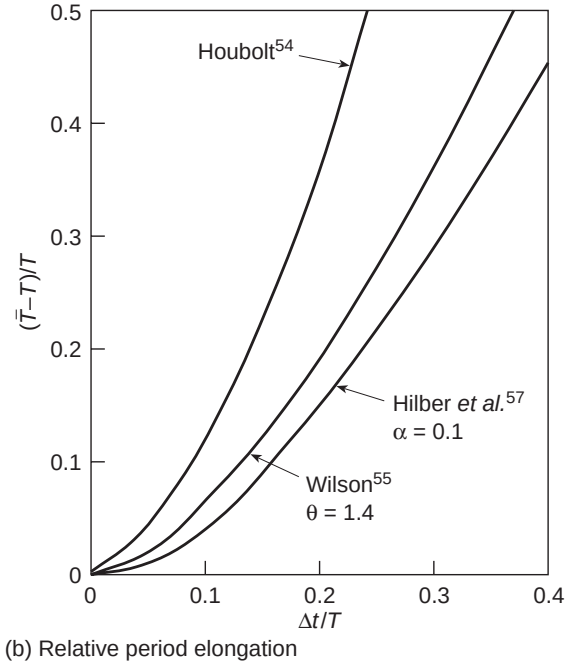


Fig. 18.13 SS22, GN22 (Newmark) or their two-step equivalent.



(a) Spectral radius



(b) Relative period elongation

Fig. 18.14 SS23, GN23 or their two-step equivalent.

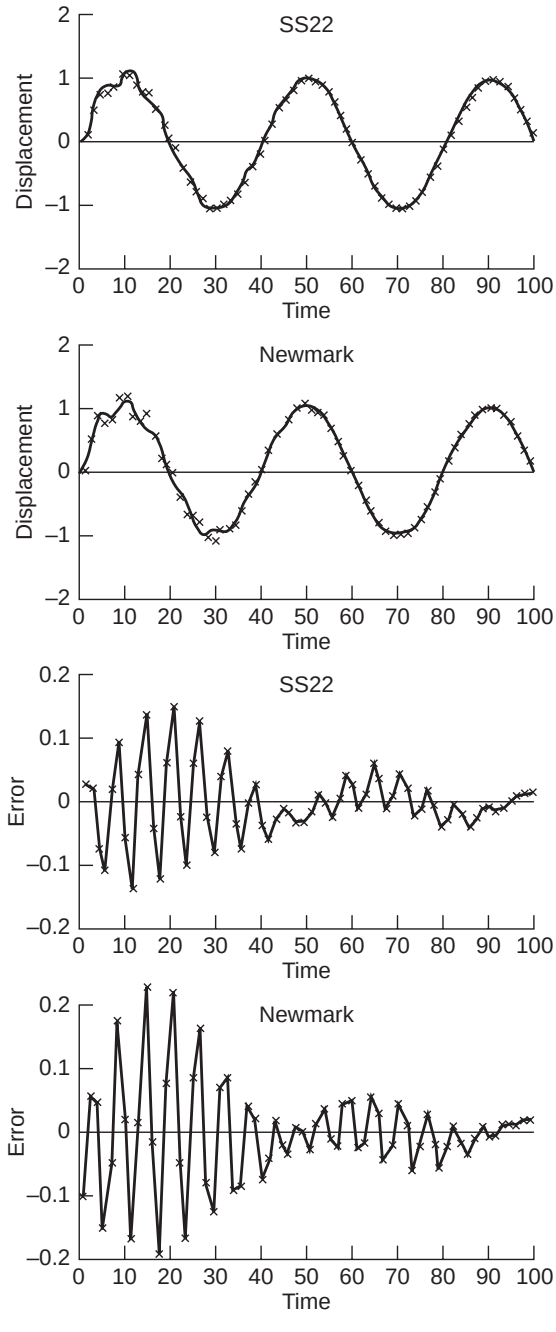


Fig. 18.15 Comparison of the SS22 and GN22 (Newmark) algorithms: a single DOF dynamic equation with periodic forcing term, $\theta_1 = \beta_1 = 1/2$, $\theta_2 = \beta_2 = 0$.

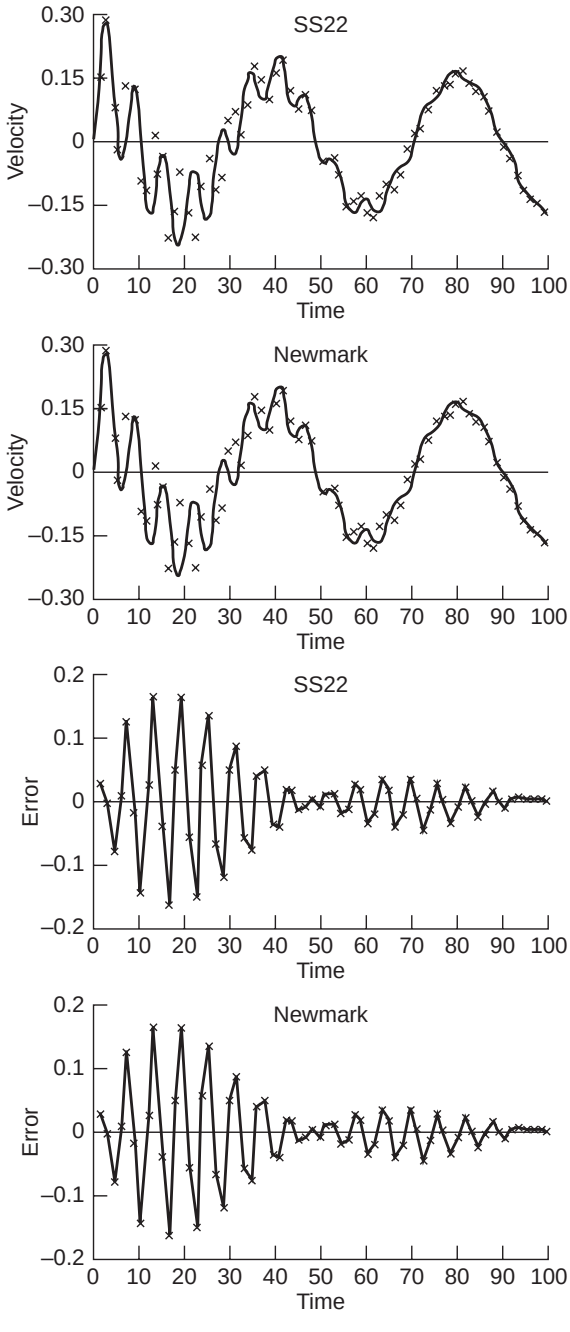


Fig. 18.15 Continued.

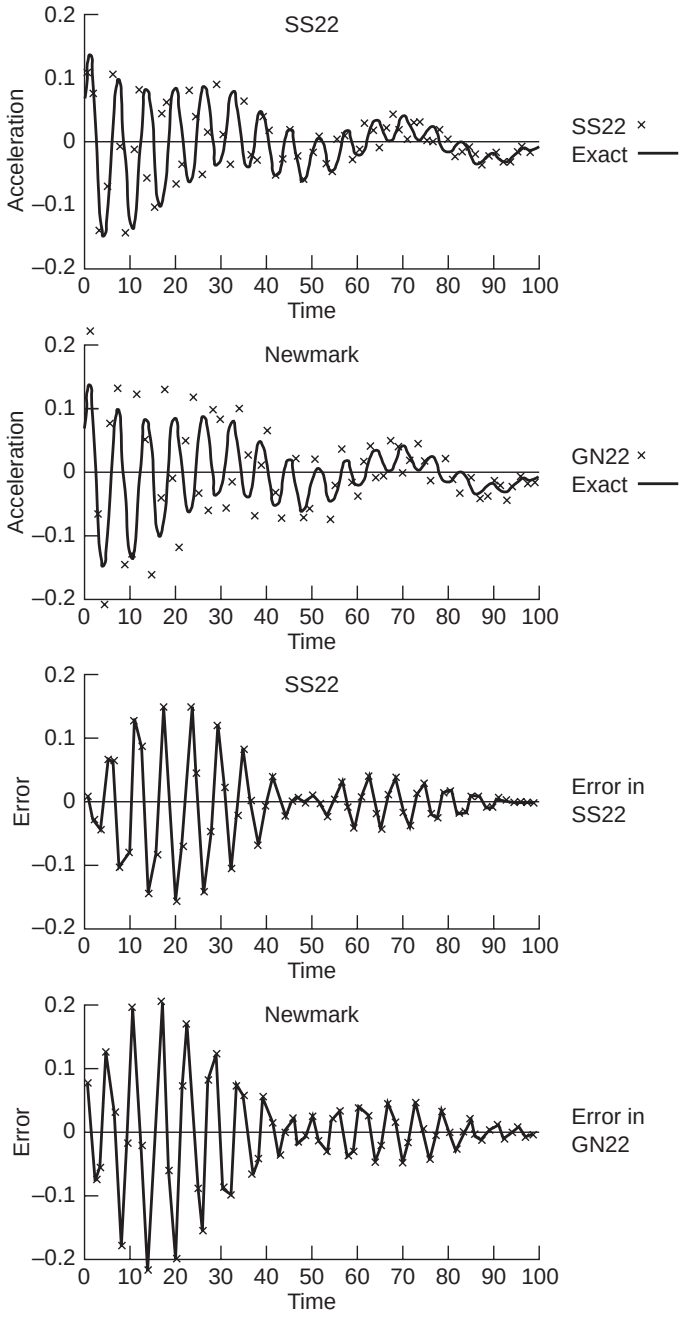


Fig. 18.15 Continued.

of $\theta = 1/2$ can perform better in a multidimensional system because of such properties.

Similar analysis can be applied to the second-order equation. Here, to simplify matters, we consider only the homogeneous undamped equation in the form

$$m\ddot{a} + ka = 0 \tag{18.133}$$

in which the value of λ is purely imaginary and corresponds to a simple oscillator. By examining μ we can find not only the amplitude ratio (which for high accuracy should be unity) but also the phase error.

In Fig. 18.13(a) we show both the variation of the modulus μ (which is called the *spectral radius*) and in Fig. 18.13(b) that of the relative period for the SS22/GN22 schemes, which of course are also applicable to the two-step equivalent. The results are plotted against

$$\frac{\Delta t}{T} \qquad \text{where} \quad T = \frac{2\pi}{\omega}; \quad \omega^2 = \frac{k}{m} \tag{18.134}$$

In Fig. 18.14(a) and (b) similar curves are given for the SS23 and GN23 schemes frequently used in practice and discussed previously.

Here as in the first-order problem we often wish to suppress (or damp out) the response to frequencies in which $\Delta t/T$ is large (say greater than 0.1) in multidegree of freedom systems, as such a response will invariably be inaccurate. At the same time below this limit it is desirable to have amplitude ratios as close to unity as possible. It is clear that the stability limit with $\theta_1 = \theta_2 = 1/2$ giving unit response everywhere is often undesirable (unless physical damping is sufficient to damp high frequency modes) and that some *algorithmic damping* is necessary in these cases. The various schemes shown in Figs 18.13 and 18.14 can be judged accordingly and provide the reason for a search for an optimum algorithm.

We have remarked frequently that although schemes can be identical with regard to stability their performances may differ slightly. In Fig. 18.15 we illustrate the application of SS22 and GN22 to a single degree of freedom system showing results and errors in each scheme.

18.6 Time discontinuous Galerkin approximation

A time discontinuous Galerkin formulation may be deduced from the finite element in time approximation procedure considered in this chapter. This is achieved by assuming the weight function W and solution variables a are approximated within each time interval Δt as

$$\begin{aligned} \mathbf{a} &= \mathbf{a}_n^+ + \Delta\mathbf{a}(t) \quad t_n^- \leqslant t < t_{n+1}^- \\ \mathbf{W} &= \mathbf{W}_n^+ + \Delta\mathbf{W}(t) \quad t_n^- \leqslant t < t_{n+1}^- \end{aligned} \tag{18.135}$$

where the time t_n^- is the limit from times smaller than t_n and t_n^+ is the limit from times larger than t_n and, thus, admit a discontinuity in the approximation to occur at each discrete time location. The functions $\Delta\mathbf{a}$ and $\Delta\mathbf{W}$ are defined to be zero at t_n and continuous up to the time t_{n+1}^- where again a discontinuity can occur during the next time interval.

The discrete form of the governing equations may be deduce starting from the time dependent partial differential equations where standard finite elements in space are combined with the time discontinuous Galerkin approximation and defining a weak form in a space-time slab. Alternatively, we may begin with the semi-discrete form as done previously in this chapter for other finite element in time methods. In this second form, for the first-order case, we write

$$I = \int_{t_n^-}^{t_{n+1}^-} \mathbf{W}^{\mathrm{T}}(\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f})\,\mathrm{d}\tau = 0 \tag{18.136}$$

Due to the discontinuity at t_n it is necessary to split the integral into

$$I = \int_{t_n^-}^{t_n^+} \mathbf{W}^{\mathrm{T}}(\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f})\,\mathrm{d}\tau + \int_{t_n^+}^{t_{n+1}^-} \mathbf{W}^{\mathrm{T}}(\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f})\,\mathrm{d}\tau = 0 \tag{18.137}$$

which gives

$$\begin{aligned} I = (\mathbf{W}_n^+)^{\mathrm{T}}\left[\mathbf{C}\left(\mathbf{a}_n^+ - \mathbf{a}_n^-\right)\right] + (\mathbf{W}_n^+)^{\mathrm{T}} \int_{t_n^+}^{t_{n+1}^-} (\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f})\,\mathrm{d}\tau \\ + \int_{t_n^+}^{t_{n+1}^-} (\Delta\mathbf{W})^{\mathrm{T}}(\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f})\,\mathrm{d}\tau = 0 \end{aligned} \tag{18.138}$$

in which now all integrals involve approximations to functions which are continuous.

To apply the above process to a second-order equation it is necessary first to reduce the equation to a pair of first-order equations. This may be achieved by defining the momenta

$$\mathbf{p} = \mathbf{M}\dot{\mathbf{a}} \tag{18.139}$$

and then writing the pair

$$\mathbf{M}\dot{\mathbf{a}} - \mathbf{p} = \mathbf{0} \tag{18.140}$$

$$\dot{\mathbf{p}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{18.141}$$

The time discrete process may now be applied by introducing two weighting functions as described in reference 37.

Example: Solution of the scalar equation To illustrate the process we consider the simple first-order scalar equation

$$c\dot{u} + ku + f = 0 \tag{18.142}$$

We consider the specific approximations

$$\begin{aligned} u(t) &= u_n^+ + \tau\Delta u_{n+1}^- \\ W(t) &= W_n^+ + \tau\Delta W_{n+1}^- \end{aligned} \tag{18.143}$$

where $\Delta u_{n+1}^- = u_{n+1}^- - u_n^+$, etc., and

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} = \frac{t - t_n}{\Delta t}$$

defines the time interval $0 < \tau < \Delta t$. This approximation gives the integral form

$$I = W_n^+ \left[c\left(u_n^+ - u_n^-\right)\right] + W_n^+ \int_0^{\Delta t} \left[\frac{1}{\Delta t} c\Delta u_{n+1}^- + k\left(u_n^+ + \tau \Delta u_{n+1}^-\right) + f\right] \mathrm{d}\tau$$
$$+ \int_0^{\Delta t} \Delta W_{n+1}^- \tau \left[\frac{1}{\Delta t} c\Delta u_{n+1}^- + k\left(u_n^+ + \tau \Delta u_{n+1}^-\right) + f\right] \mathrm{d}\tau \tag{18.144}$$

Evaluation of the integrals gives the pair of equations

$$\begin{bmatrix} (c + \Delta tk) & \frac{1}{2}\Delta tk \\ \frac{1}{2}\Delta tk & (c + \frac{1}{3}\Delta tk) \end{bmatrix} \begin{Bmatrix} u_n^+ \\ \Delta u_{n+1}^- \end{Bmatrix} + \begin{Bmatrix} \Delta t \bar{f} \\ \Delta t \Delta \bar{f} \end{Bmatrix} = \begin{Bmatrix} cu_n^- \\ 0 \end{Bmatrix} \tag{18.145}$$

where

$$\begin{Bmatrix} \bar{f} \\ \Delta \bar{f} \end{Bmatrix} = \int_0^{\Delta t} \begin{Bmatrix} f \\ \tau f \end{Bmatrix} \mathrm{d}\tau \tag{18.146}$$

Thus, with linear approximation of the variables the time discontinuous Galerkin method gives two equations to be solved for the two unknowns u_n^+ and u_{n+1}^-. It is possible to also perform a solution with *constant* approximation. Based on the above this is achieved by setting Δu_{n+1}^- and ΔW_{n+1}^- to zero yielding the single equation

$$(c + \Delta tk)u_n^+ + \Delta t \bar{f} = cu_n^- \tag{18.147}$$

and now since the approximation is constant over the entire time the u_n^+ also define exactly the u_{n+1}^- value. This form will now be recognized as identical to the *backward difference* implicit scheme defined in Fig. 18.4 for $\theta = 1$.

18.7 Concluding remarks

The derivation and examples presented in this chapter cover, we believe, the necessary tool-kit for efficient solution of many transient problems governed by Eqs (18.1) and (18.2). In the next chapter we shall elaborate further on the application of the procedures discussed here and show that they can be extended to solve coupled problems which frequently arise in practice and where simultaneous solution by time stepping is often needed.

Finally, as we have indicated in Eq. (18.3), many problems have coefficient matrices or other variations which render the problem non-linear. This topic will be addressed further in the second volume where we note also that the issue of stability after many time steps is more involved than the procedures introduced here to investigate *local stability*.

References

1. R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems*. Wiley (Interscience), New York, 1967.
2. T.D. Lambert. *Computational Methods in Ordinary Differential Equations*. John Wiley & Sons, Chichester, 1973.

3. P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations*. John Wiley & Sons, New York, 1962.
4. F.B. Hildebrand. *Finite Difference Equations and Simulations*. Prentice-Hall, Englewood Cliffs, N.J., 1968.
5. G.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, N.J., 1971.
6. W.L. Wood. *Practical Time Stepping Schemes*. Clarendon Press, Oxford, 1990.
7. J.T. Oden. A general theory of finite elements. Part II. Applications. *Internat. J. Num. Meth. Eng.*, **1**, 247–54, 1969.
8. I. Fried. Finite element analysis of time-dependent phenomena. *AIAA J.*, **7**, 1170–73, 1969.
9. J.H. Argyris and D.W. Scharpf. Finite elements in time and space. *Nucl. Eng. Design*, **10**, 456–69, 1969.
10. O.C. Zienkiewicz and C.J. Parekh. Transient field problems – two and three dimensional analysis by isoparametric finite elements. *Internat. J. Num. Meth. Eng.*, **2**, 61–71, 1970.
11. O.C. Zienkiewicz. *The Finite Element Method in Engineering Science*. McGraw-Hill, London, 2nd edition, 1971.
12. O.C. Zienkiewicz and R.W. Lewis. An analysis of various time stepping schemes for initial value problems. *Earthquake Eng. Struct. Dyn.*, **1**, 407–8, 1973.
13. W.L. Wood and R.W. Lewis. A comparison of time marching schemes for the transient heat conduction equation. *Internat. J. Num. Meth. Eng.*, **9**, 679–89, 1975.
14. O.C. Zienkiewicz. A new look at the Newmark, Houbolt and other time stepping formulas. A weighted residual approach. *Earthquake Eng. Struct. Dyn.*, **5**, 413–18, 1977.
15. W.L. Wood. On the Zienkiewicz four-time-level scheme for numerical integration of vibration problems. *Internat. J. Num. Meth. Eng.*, **11**, 1519–28, 1977.
16. O.C. Zienkiewicz, W.L. Wood, and R.L. Taylor. An alternative single-step algorithm for dynamic problems. *Earthquake Eng. Struct. Dyn.*, **8**, 31–40, 1980.
17. W.L. Wood. A further look at Newmark, Houbolt, etc. time-stepping formulae. *Internat. J. Num. Meth. Eng.*, **20**, 1009–17, 1984.
18. O.C. Zienkiewicz, W.L. Wood, N.W. Hine, and R.L. Taylor. A unified set of single-step algorithms. Part 1: general formulation and applications. *Internat. J. Num. Meth. Eng.*, **20**, 1529–52, 1984.
19. W.L. Wood. A unified set of single-step algorithms. Part 2: theory. *Internat. J. Num. Meth. Eng.*, **20**, 2302–09, 1984.
20. M. Katona and O.C. Zienkiewicz. A unified set of single-step algorithms. Part 3: the beta-m method, a generalization of the Newmark scheme. *Internat. J. Num. Meth. Eng.*, **21**, 1345–59, 1985.
21. E. Varoglu and N.D.L. Finn. A finite element method for the diffusion convection equations with concurrent coefficients. *Adv. Water Resources,* **1**, 337–41, 1973.
22. C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Comp. Meth. Appl. Mech. Engng*, **45**, 285–312, 1984.
23. T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Com. Meth. Appl. Mech. Eng.*, **73**, 173–89, 1989.
24. T.J.R. Hughes and G.M. Hulbert. Space-time finite element methods in elastodynamics: Formulation and error estimates. *Com. Meth. Appl. Mech. Eng.*, **66**, 339–63, 1988.
25. G.M. Hulbert and T.J.R. Hughes. Space-time finite element methods for second-order hyperbolic equations. *Com. Meth. Appl. Mech. Eng.*, **84**, 327–48, 1990.
26. G.M. Hulbert. Time finite element methods for structural dynamics. *Internat. J. Num. Meth. Eng.*, **33**, 307–31, 1992.

27. B.M. Irons and C. Treharne. A bound theorem for eigenvalues and its practical application. In *Proc. 3rd Conf. Matrix Methods in Structural Mechanics*, volume AFFDL-TR-71–160, pages 245–54, Wright-Patterson Air Force Base, Ohio, 1972.
28. K. Washizu. *Variational Methods in Elasticity and Plasticity*. Pergamon Press, New York, 3 edition, 1982.
29. M. Gurtin. Variational principles for linear initial-value problems. *Q. Appl. Math.*, **22**, 252–56, 1964.
30. M. Gurtin. Variational principles for linear elastodynamics. *Arch. Rat. Mech. Anal.*, **16**, 34–50, 1969.
31. E.L. Wilson and R.E. Nickell. Application of finite element method to heat conduction analysis. *Nucl. Eng Design,* **4**, 1–11, 1966.
32. J. Crank and P. Nicolson. A practical method for numerical integration of solutions of partial differential equations of heat conduction type. *Proc. Camb. Phil. Soc.,* **43**, 50, 1947.
33. R.L. Taylor and O.C. Zienkiewicz. A note on the Order of Approximation. *Internat. J. Solids Structures*, **21**, 793–838, 1985.
34. P. Lesaint and P.-A. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*. Academic Press, New York, 1974.
35. C. Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge, 1987.
36. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems **I**, A linear model problem. *SIAM J. Numer. Anal.*, **28**, 43–77, 1991.
37. X.D. Li and N.-E. Wiberg. Structural dynamic analysis by a time-discontinuous Galerkin finite element method. *Internat. J. Num. Meth. Eng.*, **39**, 2131–52, 1996.
38. M. Zlamal. Finite element methods in heat conduction problems. In J. Whiteman, editor, *The Mathematics of Finite Elements and Applications*, pages 85–104. Academic Press, London, 1977.
39. W. Liniger. Optimisation of a numerical integration method for stiff systems of ordinary differential equations. Technical Report RC2198, IBM Research, 1968.
40. J.M. Bettencourt, O.C. Zienkiewicz, and G. Cantin. Consistent use of finite elements in time and the performance of various recurrence schemes for heat diffusion equation. *Internat. J. Num. Meth. Eng.*, **17**, 931–38, 1981.
41. N. Newmark. A method of computation for structural dynamics. *J. Eng. Mech. Div.*, **85**, 67–94, 1959.
42. T. Belytschko and T.J.R. Hughes, editors. *Computational Methods for Transient Analysis*. North-Holland, Amsterdam, 1983.
43. J.C. Simo and K. Wong. Unconditionally stable algorithms for rigid body dynamics that exactly conserve energy and momentum. *Internat. J. Num. Meth. Eng.*, **31**, 19–52, 1991.
44. J.C. Simo and N. Tarnow. The discrete energy-momentum method. conserving algorithm for nonlinear elastodynamics. *ZAMP,* **43**, 757–93, 1992.
45. J.C. Simo and N. Tarnow. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Com. Meth. Appl. Mech. Eng.*, **100**, 63–116, 1992.
46. O. Gonzalez. *Design and analysis of conserving integrators for nonlinear Hamiltonian systems with symmetry*. Ph.d thesis, Stanford University, Stanford, California, 1996.
47. I. Miranda, R.M. Ferencz, and T.J.R. Hughes. An improved implicit-explicit time integration method for structural dynamics. *Earthquake Eng. Struct. Dyn.*, **18**, 643–55, 1989.
48. E.J. Routh. *A Treatise on the Stability of a Given State or Motion*. Macmillan, London, 1977.
49. A. Hurwitz. Uber die Bedingungen, unter welchen eine Gleichung nur Würzeln mit negatives reellen teilen besitzt. *Math. Ann.*, **46**, 273–84, 1895.
50. F.R. Gantmacher. *The Theory of Matrices*. Chelsea, New York, 1959.

51. G.G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, **3**, 27–43, 1963.
52. C.W. Gear. The automatic integration of stiff ordinary differential equations. In A.J.H. Morrell, editor, *Information Processing 68*. North Holland, Dordrecht, 1969.
53. W. Liniger. Global accuracy and A-stability of one and two step integration formulae for stiff ordinary differential equations. In *Proc. Conf. on Numerical Solution of Differential Equations*, Dundee University, 1969.
54. J.C. Houbolt. A recurrence matrix solution for dynamic response of elastic aircraft. *J. Aero. Sci.*, **17**, 540–50, 1950.
55. K.J. Bathe and E.L. Wilson. Stability and accuracy analysis of direct integration methods. *Earthquake Eng. Struct. Dyn.*, **1**, 283–91, 1973.
56. W. Wood, M. Bossak, and O.C. Zienkiewicz. An alpha modification of Newmark's method. *Internat. J. Num. Meth. Eng.*, **15**, 1562–66, 1980.
57. H. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for the time integration algorithms in structural dynamics. *Earthquake Eng. Struct. Dyn.*, **5**, 283–92, 1977.
58. W.L. Wood. On the Zienkiewicz three- and four-time-level schemes applied to the numerical integration of parabolic problems. *Internat. J. Num. Meth. Eng.*, **12**, 1717–26, 1978.

19

Coupled systems

19.1 Coupled problems – definition and classification

Frequently two or more physical systems interact with each other, with the independent solution of any one system being impossible without simultaneous solution of the others. Such systems are known as coupled and of course such coupling may be weak or strong depending on the degree of interaction.

An obvious 'coupled' problem is that of dynamic fluid–structure interaction. Here neither the fluid nor the structural system can be solved independently of the other due to the unknown interface forces.

A definition of coupled systems may be generalized to include a wide range of problems and their numerical discretization as:[1]

Coupled systems and formulations are those applicable to multiple domains and dependent variables which usually (*but not always*) *describe different physical phenomena and in which*

(a) *neither domain can be solved while separated from the other*;
(b) *neither set of dependent variables can be explicitly eliminated at the differential equation level*.

The reader may well contrast this with definitions of *mixed* and *irreducible* formulations given in Chapter 11 and find some similarities. Clearly 'mixed' and 'coupled' formulations are analogous, with the main difference being that in the former elimination of some dependent variables is possible at the governing differential equation level. In the coupled system a full analytical solution or inversion of a (discretized) single system is necessary before such elimination is possible.

Indeed, a further distinction can be made. In coupled systems the solution of any single system is a well-posed problem and is possible when the variables corresponding to the other system are prescribed. This is not always the case in mixed formulations.

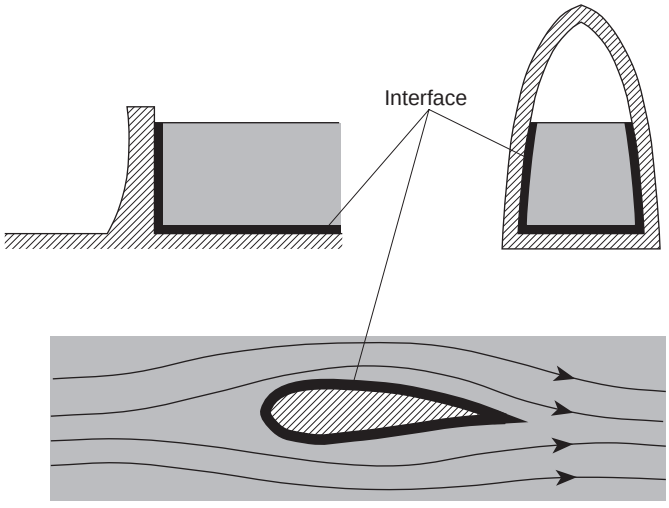
It is convenient to classify coupled systems into two categories:

Class I. This class contains problems in which coupling occurs on domain interfaces via the boundary conditions imposed there. Generally the domains describe different physical situations but it is possible to consider coupling between

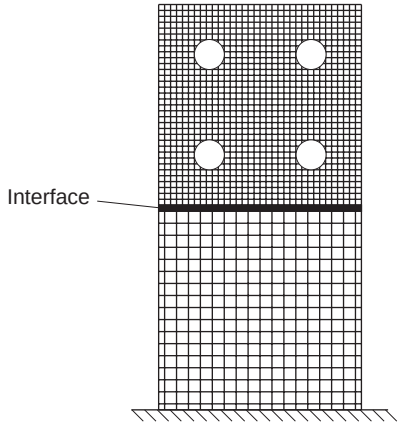
domains that are physically similar in which different discretization processes have been used.

Class II. This class contains problems in which the various domains overlap (totally or partially). Here the coupling occurs through the governing differential equations describing different physical phenomena.

Typical of the first category are the problems of fluid–structure interaction illustrated in Fig. 19.1(a) where physically different problems interact and also

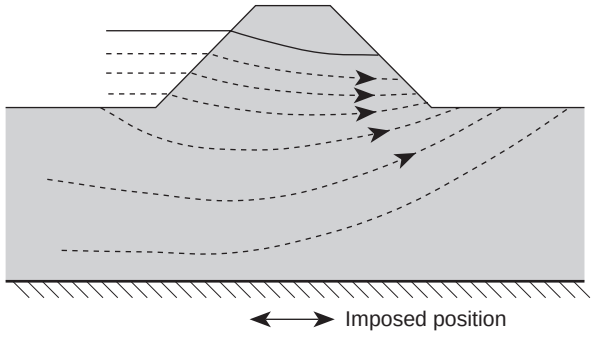


(a) Fluid–structure interaction (physically different domains)

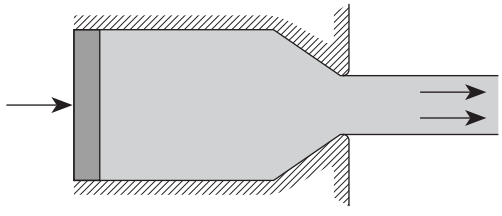


(b) Structure–structure interaction (physically identical domains)

Fig. 19.1 Class I problems with coupling via interfaces (shown as thick line).



(a) Seepage through a porous medium interacts with its dynamic, structural behaviour



(b) Problem of metal extrusion in which the plastic flow is coupled with the thermal field

Fig. 19.2 Class II problems with coupling in overlapping domains.

structure–structure interactions of Fig. 19.1(b) where the interface simply divides arbitrarily chosen regions in which different numerical discretizations are used.

The need for the use of different discretization may arise from different causes. Here for instance:

1. Different finite element meshes may be advantageous to describe the subdomains.
2. Different procedures such as the combination of boundary method and finite elements in respective regions may be computationally desirable.
3. Domains may simply be divided by the choice of different time-stepping procedures, e.g. of an implicit and explicit kind.

In the second category, typical problems are illustrated in Fig. 19.2. One of these is that of metal extrusion where the plastic flow is strongly coupled with the temperature field while at the same time the latter is influenced by the heat generated in the plastic flow. This problem will be considered in more detail in Volume 2 but is included to illustrate a form of coupling that commonly occurs in analyses of solids. The other problem shown in Fig. 19.2 is that of soil dynamics (earthquake response of a dam) in which the seepage flow and pressures interact with the dynamic behaviour of the soil 'skeleton'.

We observe that, in the examples illustrated, motion invariably occurs. Indeed, the vast majority of coupled problems involve such transient behaviour and for this reason the present chapter will only consider this area. It will thus follow and expand the analysis techniques presented in Chapters 17 and 18.

As the problems encountered in coupled analysis of various kinds are similar, we shall focus the presentation on three examples:

1. fluid–structure interaction (confined to small amplitudes);
2. soil–fluid interaction;
3. implicit–explicit dynamic analysis of a structure where the separation involves the process of temporal discretization.

In these problems all the typical features of coupled analysis will be found and extension to others will normally follow similar lines. In Volume 2 we shall, for instance, deal in more detail with the problem of coupled metal forming[2] and the reader will discover the similarities.

As a final remark, it is worthwhile mentioning that problems such as linear thermal stress analysis to which we have referred frequently in this volume are not coupled in the terms defined here. In this the stress analysis problem requires a knowledge of the temperature field but the temperature problem can be solved independently of the stress field.† Thus the problem decouples in one direction. Many examples of truly coupled problems will be found in available books. [4–6]

19.2 Fluid–structure interaction (Class I problem)

19.2.1 General remarks and fluid behaviour equations

The problem of fluid–structure interaction is a wide one and covers many forms of fluid which, as yet, we have not discussed in any detail. The consideration of problems in which the fluid is in substantial motion is deferred until Volume 3 and, thus, we exclude at this stage such problems as flutter where movement of an aerofoil influences the flow pattern and forces around it leading to possible instability. For the same reason we also exclude here the 'singing wire' problem in which the shedding of vortices reacts with the motion of the wire.

However, in a very considerable range of problems the fluid displacement remains small while interaction is substantial. In this category fall the first two examples of Fig. 19.1 in which the structural motions influence and react with the generation of pressures in a reservoir or a container. A number of symposia have been entirely devoted to this class of problems which is of considerable engineering interest, and here fortunately considerable simplifications are possible in the description of the fluid phase. References 7–22 give some typical studies.

† In a general setting the temperature field does depend upon the strain rate. However, these terms are not included in the form presented in this volume and in many instances produce insignificant changes to the solution.[3]

In such problems it is possible to write the dynamic equations of fluid behaviour simply as

$$\frac{\partial(\rho \mathbf{v})}{\partial t} \approx \rho \frac{\partial \mathbf{v}}{\partial t} = \boldsymbol{\nabla} p \tag{19.1}$$

where $\mathbf{v}$ is the fluid velocity, ρ is the fluid density and p the pressure. In postulating the above we have assumed

1. that the density ρ varies by a small amount only so may be considered constant;
2. that velocities are small enough for convective effects to be omitted;
3. that viscous effects by which deviatoric stresses are introduced can be neglected in the fluid.

The reader can in fact note that with the preceding assumption Eq. (19.1) is a special form of a more general relation (described in Chapter 1 of Volume 3).

The continuity equation based on the same assumption is

$$\rho \operatorname{div} \mathbf{v} \equiv \rho \boldsymbol{\nabla}^{\mathrm{T}} \mathbf{v} = -\frac{\partial \rho}{\partial t} \tag{19.2}$$

and noting that

$$\mathrm{d}\rho = \frac{\rho}{K} \mathrm{d}p \tag{19.3}$$

where K is the bulk modulus, we can write

$$\boldsymbol{\nabla}^{\mathrm{T}} \mathbf{v} = -\frac{1}{K} \frac{\partial p}{\partial t} \tag{19.4}$$

Elimination of $\mathbf{v}$ between (19.1) and (19.4) gives the well-known Helmholtz equation governing the pressure p:

$$\boldsymbol{\nabla}^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \tag{19.5}$$

where

$$c = \sqrt{\frac{K}{\rho}} \tag{19.6}$$

denotes the speed of sound in the fluid.

The equations described above are the basis of *acoustic* problems.

19.2.2 Boundary conditions for the fluid. Coupling and radiation

In Fig. 19.3 we focus on the Class I problem illustrated in Fig. 19.1(a) and on the boundary conditions possible for the fluid part described by the governing equation (19.5). As we know well, either normal gradients or values of p now need to be specified.

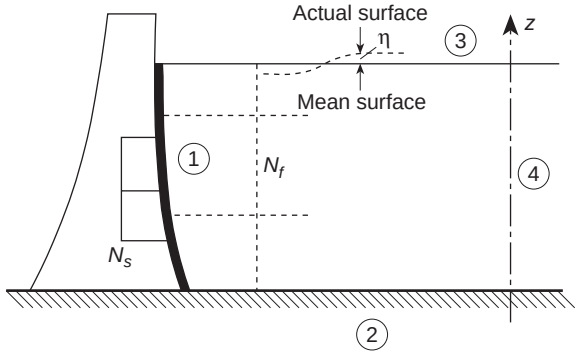


Fig. 19.3 Boundary conditions for the fluid component of the fluid–structure interaction.

Interface with solid

On the boundaries ① and ② in Fig. 19.3 the normal velocities (or their time derivatives) are prescribed. Considering the pressure gradient in the normal direction to the face n we can thus write, by Eq. (19.1),

$$\frac{\partial p}{\partial n} = -\rho \dot{\bar{v}}_n = -\rho \mathbf{n}^{\mathrm{T}} \dot{\bar{\mathbf{v}}} \tag{19.7}$$

where $\mathbf{n}$ is the direction cosine vector for an outward pointing normal to the fluid region and $\dot{\bar{v}}_n$ is prescribed.

Thus, for instance, on boundary ① coupling with the motion of the structure described by displacement $\mathbf{u}$ occurs. Here we put

$$\dot{\bar{v}}_n = \ddot{u}_n = \mathbf{n}^{\mathrm{T}} \ddot{\mathbf{u}} \tag{19.8}$$

while on boundary ② where only horizontal motion exists we have

$$\dot{\bar{v}}_z = 0 \tag{19.9}$$

Coupling with the structure motion occurs only via boundary ①.

Free surface

On the free surface (boundary ③ in Fig. 19.3) the simplest assumption is that

$$p = 0 \tag{19.10}$$

However, this does not allow for any possibility of surface gravity waves. These can be approximated by assuming the actual surface to be at an elevation η relative to the mean surface. Now

$$p = \rho g \eta \tag{19.11}$$

where g is the acceleration due to gravity. From Eq. (19.1) we have, on noting $v_z = \partial \eta / \partial t$ and assuming ρ to be constant,

$$\rho \frac{\partial^2 \eta}{\partial t^2} = -\frac{\partial p}{\partial z} \tag{19.12}$$

and on elimination of η, using Eq. (19.11), we have a specified normal gradient condition

$$\frac{\partial p}{\partial z} = -\frac{1}{g}\frac{\partial^2 p}{\partial t^2} = -\frac{1}{g}\ddot{p} \tag{19.13}$$

This allows for gravity waves to be approximately incorporated in the analysis and is known as the *linearized surface wave condition*.

Radiation boundary

Boundary ④ physically terminates an infinite domain and some approximation to account for the effect of such a termination is necessary. The main dynamic effect is simply that the wave solution of the governing equation (19.5) must here be composed of *outgoing waves only* as no input from the infinite domain exists.

If we consider only variations in x (the horizontal direction) we know that the general solution of Eq. (19.5) can be written as

$$p = F(x - ct) + G(x + ct) \tag{19.14}$$

where c is the wave velocity given by Eq. (19.6) and the two waves F and G travel in positive and negative directions of x, respectively.

The absence of the incoming wave G means that on boundary ④ we have only

$$p = F(x - ct) \tag{19.15}$$

Thus

$$\frac{\partial p}{\partial n} \equiv \frac{\partial p}{\partial x} = F' \tag{19.16}$$

and

$$\frac{\partial p}{\partial t} = -cF' \tag{19.17}$$

where F' denotes the derivative of F with respect to $(x - ct)$. We can therefore eliminate the unknown function F' and write

$$\frac{\partial p}{\partial n} = -\frac{1}{c}\dot{p} \tag{19.18}$$

which is a condition very similar to that of Eq. (19.13). This boundary condition was first presented in reference 7 for radiating boundaries and has an analogy with a damping element placed there.

19.2.3 Weak form for coupled systems

A weak form for each part of the coupled system may be written as described in Chapter 3. Accordingly, for the fluid we can write the differential equation as

$$\delta\Pi_f = \int_{\Omega_f} \delta p \left[\frac{1}{c^2}\ddot{p} - \boldsymbol{\nabla}^2 p\right] \mathrm{d}\Omega = 0 \tag{19.19}$$

which after integration by parts and substitution of the boundary conditions described above yields

$$\int_{\Omega_f} \delta p \left[\frac{1}{c^2} \ddot{p} + (\boldsymbol{\nabla})^{\mathrm{T}} \boldsymbol{\nabla} p \right] \mathrm{d}\Omega + \int_{\Gamma_1} \delta p \mathbf{n}^{\mathrm{T}} \ddot{\mathbf{u}} \, \mathrm{d}\Gamma + \int_{\Gamma_3} \delta p \frac{1}{g} \ddot{p} \, \mathrm{d}\Gamma + \int_{\Gamma_4} \delta p \frac{1}{c} \dot{p} \, \mathrm{d}\Gamma = 0 \tag{19.20}$$

where Ω_f is the fluid domain and Γ_i the integral over boundary part ⓘ.

Similarly for the solid the weak form after integration by parts is given by

$$\int_{\Omega} \delta \mathbf{u} \left[\rho_s \ddot{\mathbf{u}} + \mathbf{S}^{\mathrm{T}} \mathbf{D} \mathbf{S} \mathbf{u} \right] \mathrm{d}\Omega - \int_{\Gamma_t} \delta \mathbf{u}^{\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d}\Gamma = 0 \tag{19.21}$$

where for pressure defined positive in compression the surface traction is defined as

$$\bar{\mathbf{t}} = -p \mathbf{n}_s = p \mathbf{n} \tag{19.22}$$

since the outward normal to the solid is $\mathbf{n}_s = -\mathbf{n}$. The traction integral in Eq. (19.21) is now expressed as

$$\int_{\Gamma_t} \delta \mathbf{u}^{\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d}\Gamma = \int_{\Gamma_t} \delta \mathbf{u}^{\mathrm{T}} \mathbf{n} p \, \mathrm{d}\Gamma \tag{19.23}$$

(1) In complex physical situations, the interaction between compressibility and internal gravity waves (interaction between acoustic modes and sloshing modes) leads to a modified Helmholz equation. The Eq. (19.5) should then be replaced by a more complex equation: in a stratified medium for instance, the irrotationality condition for the fluid is not totally verified (the fluid is irrotational in a plane perpendicular to the stratification axis).[16]

(2) The variational formulation defined by Eq. (19.20) is valid in the static case provided the following constraints conditions are added $\int_{\Omega_f} p \, \mathrm{d}\Omega + \rho c^2 \int_{\partial\Omega_f} n^{\mathrm{T}} u \, \mathrm{d}\Gamma = 0$ for a compressible fluid filling a cavity, $\int_{\Gamma_1} n^{\mathrm{T}} u \, \mathrm{d}\Gamma + \int_{\Gamma_2} p/\rho g \, \mathrm{d}\Gamma = 0$, for an incompressible liquid with a free surface contained inside a reservoir. The static behaviour is important for the modal response of coupled systems when modal truncation need static corrections in order to accelerate the convergence of the method. This static behaviour is also of prime importance for the construction of reduced matrix models when using dynamic substructuring methods for fluid structure interaction problems.[17,18]

19.2.4 The discrete coupled system

We shall now consider the coupled problem discretized in the standard (displacement) manner with the displacement vector approximated as

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_u \tilde{\mathbf{u}} \tag{19.24}$$

and the fluid similarly approximated by

$$p \approx \hat{p} = \mathbf{N}_p \tilde{\mathbf{p}} \tag{19.25}$$

where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{p}}$ are the nodal parameters of each field and $\mathbf{N}_u$ and $\mathbf{N}_p$ are appropriate shape functions.

The discrete structural problem thus becomes

$$\mathbf{M}\ddot{\tilde{\mathbf{u}}} + \mathbf{C}\dot{\tilde{\mathbf{u}}} + \mathbf{K}\tilde{\mathbf{u}} - \mathbf{Q}\tilde{\mathbf{p}} + \mathbf{f} = \mathbf{0} \tag{19.26}$$

where the coupling term arises due to the pressures (tractions) specified on the boundary as

$$\int_{\Gamma_t} \mathbf{N}_u^{\mathrm{T}} \bar{\mathbf{t}}\, \mathrm{d}\Gamma = \int_{\Gamma_t} \mathbf{N}_u^{\mathrm{T}} \mathbf{n} \mathbf{N}_p\, \mathrm{d}\Gamma = \mathbf{Q}\tilde{\mathbf{p}} \tag{19.27}$$

The terms of the other matrices are already well known to the reader as mass, damping, stiffness and force.

Standard Galerkin discretization applied to the weak form of the fluid equation (19.20) leads to

$$\mathbf{S}\ddot{\tilde{\mathbf{p}}} + \tilde{\mathbf{C}}\dot{\tilde{\mathbf{p}}} + \mathbf{H}\tilde{\mathbf{p}} + \mathbf{Q}^{\mathrm{T}}\ddot{\tilde{\mathbf{u}}} + \mathbf{q} = \mathbf{0} \tag{19.28}$$

where

$$\begin{aligned} \mathbf{S} &= \int_\Omega \mathbf{N}_p^{\mathrm{T}} \frac{1}{c^2} \mathbf{N}_p\, \mathrm{d}\Omega + \int_{\Gamma_3} \mathbf{N}_p^{\mathrm{T}} \frac{1}{g} \mathbf{N}_p\, \mathrm{d}\Gamma \\ \tilde{\mathbf{C}} &= \int_{\Gamma_4} \mathbf{N}_p^{\mathrm{T}} \frac{1}{c^2} \mathbf{N}_p\, \mathrm{d}\Gamma \\ \mathbf{H} &= \int_\Omega (\boldsymbol{\nabla}\mathbf{N}_p)^{\mathrm{T}} \boldsymbol{\nabla}\mathbf{N}_p\, \mathrm{d}\Omega \end{aligned} \tag{19.29}$$

and $\mathbf{Q}$ is identical to that of Eq. (19.27).

19.2.5 Free vibrations

If we consider free vibrations and omit all force and damping terms (noting that in the fluid component the damping is strictly that due to radiation energy loss) we can write the two equations (19.26) and (19.28) as a set:

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{Q}^{\mathrm{T}} & \mathbf{S} \end{bmatrix} \begin{Bmatrix} \ddot{\tilde{\mathbf{u}}} \\ \ddot{\tilde{\mathbf{p}}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix} = \mathbf{0} \tag{19.30}$$

and attempt to proceed to establish the eigenvalues corresponding to natural frequencies. However, we note immediately that the system is not symmetric (nor positive definite) and that standard eigenvalue computation methods are not directly applicable. Physically it is, however, clear that the eigenvalues are real and that free vibration modes exist.

The above problem is similar to that arising in vibration of rotating solids and special solution methods are available, though costly.[23] It is possible by various manipulations to arrive at a symmetric form and reduce the problem to a standard eigenvalue one.[14–26]

A simple method proposed by Ohayon proceeds to achieve the symmetrization objective by putting $\tilde{\mathbf{u}} = \breve{\mathbf{u}}\, \mathrm{e}^{\mathrm{i}\omega t}$, $\tilde{\mathbf{p}} = \breve{\mathbf{p}}\, \mathrm{e}^{\mathrm{i}\omega t}$ and rewriting Eq. (19.30) as

$$\begin{aligned} \mathbf{K}\breve{\mathbf{u}} - \mathbf{Q}\breve{\mathbf{p}} - \omega^2 \mathbf{M}\breve{\mathbf{u}} &= \mathbf{0} \\ \mathbf{H}\breve{\mathbf{p}} - \omega^2 \mathbf{S}\breve{\mathbf{p}} - \omega^2 \mathbf{Q}\breve{\mathbf{u}} &= \mathbf{0} \end{aligned} \tag{19.31}$$

and an additional variable $\breve{\mathbf{q}}$ such that

$$\breve{\mathbf{p}} = \omega^2 \breve{\mathbf{q}} \tag{19.32}$$

After some manipulation and substitution we can write the new system as

$$\left\{ \begin{bmatrix} \mathbf{K} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} - \omega^2 \begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{Q} \\ \mathbf{0} & \mathbf{0} & \mathbf{S} \\ \mathbf{Q}^{\mathrm{T}} & \mathbf{S}^{\mathrm{T}} & \mathbf{H} \end{bmatrix} \right\} \begin{Bmatrix} \breve{\mathbf{u}} \\ \breve{\mathbf{p}} \\ \breve{\mathbf{q}} \end{Bmatrix} = \mathbf{0} \tag{19.33}$$

which is a symmetric generalized eigenproblem. Further, the variable $\breve{\mathbf{q}}$ can now be eliminated by static condensation and the final system becomes symmetric and now contains only the basic variables. The system (19.32), with static corrections, may lead to convenient reduced matrix models through appropriate dynamic substructuring methods.[19]

An alternative that has frequently been used is to introduce a new symmetrizing variable at the governing equation level, but this is clearly not necessary.[14,15]

As an example of a simple problem in the present category we show an analysis of a three-dimensional flexible wall vibrating with a fluid encased in a 'rigid' container[27] (Fig. 19.4).

19.2.6 Forced vibrations and transient step-by-step algorithms

The reader can easily verify that the steady-state, linear response to periodic input can be readily computed in the complex frequency domain by the procedures described in Chapter 17. Here no difficulties arise due to the non-symmetric nature of equations and standard procedures can be applied. Chopra and co-workers have, for instance, done many studies of dam/reservoir interaction using such methods.[28,29] However, such methods are not generally economical for very large problems and fail in non-linear response studies. Here time-stepping procedures are required in the manner discussed in the previous chapter. However, simple application of methods developed there leads to an unsymmetric problem for the combined system (with $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{p}}$ as variables) due to the form of the matrices appearing in (19.30) and a modified approach is necessary.[30] In this each of the equations (19.26) and (19.28) is first *discretized in time separately* using the general approaches of Chapter 18.

Thus in the time interval Δt we can approximate $\tilde{\mathbf{u}}$ using, say, the general SS22 procedure as follows. First we write

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}_n + \dot{\tilde{\mathbf{u}}}_n \tau + \boldsymbol{\alpha} \frac{\tau^2}{2} \tag{19.34}$$

with a similar expression for $\mathbf{p}$,

$$\tilde{\mathbf{p}} = \tilde{\mathbf{p}}_n + \dot{\tilde{\mathbf{p}}}_n \tau + \boldsymbol{\beta} \frac{\tau^2}{2} \tag{19.35}$$

where $\tau = t - t_n$.

Insertion of the above into Eqs (19.26) and (19.28) and weighting with two *separate weighting functions* results in two relations in which $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the unknowns. These

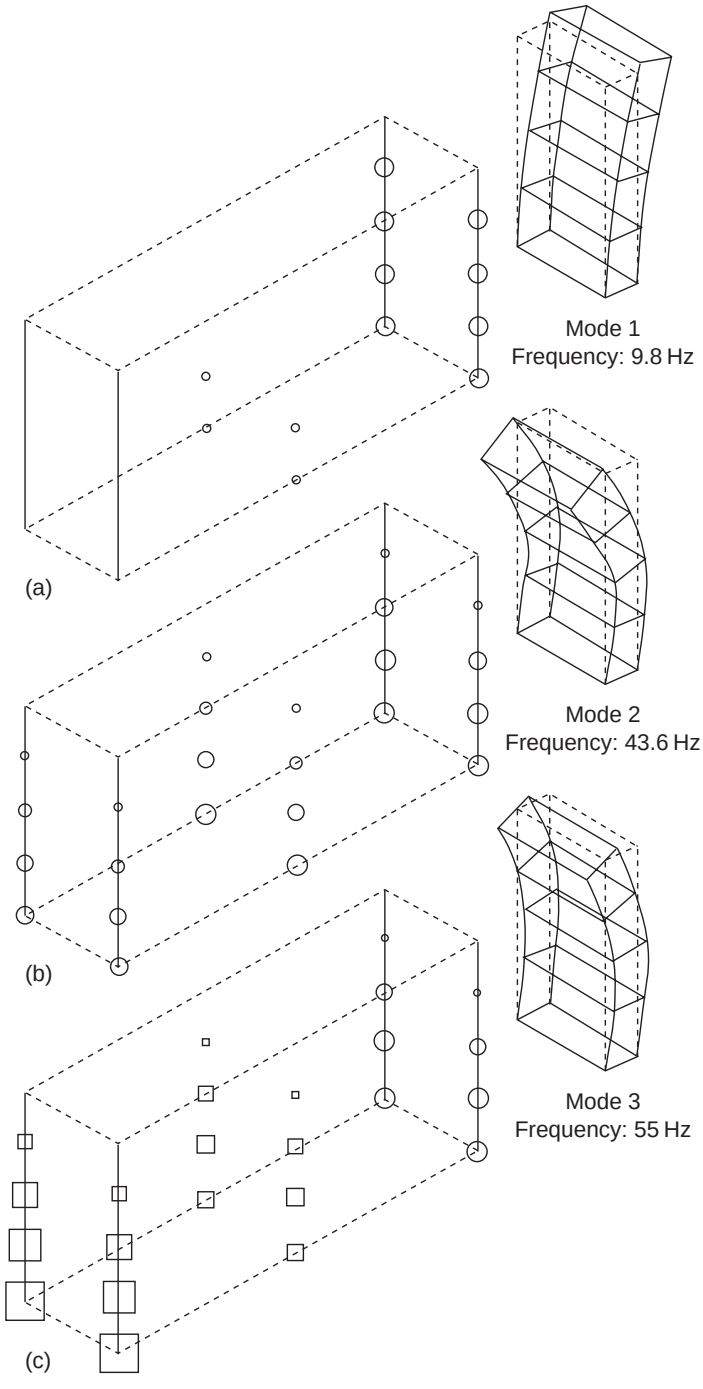


Fig. 19.4 Body of fluid with a free surface oscillating with a wall. Circles show pressure amplitude and squares indicate opposite signs. Three-dimensional approach using parabolic elements.

are

$$\mathbf{M}\boldsymbol{\alpha} + \mathbf{C}\left(\dot{\bar{\mathbf{u}}}_{n+1} + \theta_1 \Delta t \boldsymbol{\alpha}\right) + \mathbf{K}\left(\bar{\mathbf{u}}_{n+1} + \tfrac{1}{2}\theta_2 \Delta t^2 \boldsymbol{\alpha}\right) - \mathbf{Q}\left(\bar{\mathbf{p}}_{n+1} + \tfrac{1}{2}\bar{\theta}_2 \Delta t^2 \boldsymbol{\beta}\right) + \mathbf{f}_{n+1} = \mathbf{0} \tag{19.36}$$

and

$$\mathbf{S}\boldsymbol{\beta} + \mathbf{Q}^{\mathrm{T}}\boldsymbol{\alpha} + \mathbf{H}\left(\bar{\mathbf{p}}_{n+1} + \tfrac{1}{2}\theta_2 \Delta t^2 \boldsymbol{\beta}\right) + \mathbf{q}_{n+1} = \mathbf{0} \tag{19.37}$$

where

$$\begin{aligned} \bar{\mathbf{u}}_{n+1} &= \tilde{\mathbf{u}}_n + \theta_1 \Delta t \dot{\tilde{\mathbf{u}}}_n \\ \dot{\bar{\mathbf{u}}}_{n+1} &= \dot{\tilde{\mathbf{u}}}_n \\ \bar{\mathbf{p}}_{n+1} &= \tilde{\mathbf{p}}_n + \bar{\theta}_1 \Delta t \dot{\tilde{\mathbf{p}}}_n \end{aligned} \tag{19.38}$$

are the predictors for the $n+1$ time step. In the above the parameters θ_i and $\bar{\theta}_i$ are similar to those of Eq. (18.49) and can be chosen by the user. It is interesting to note that the equation system can be put in symmetric form as

$$\begin{bmatrix} (\mathbf{M} + \theta_1 \Delta t \mathbf{C} + \tfrac{1}{2}\theta_2 \Delta t^2 \mathbf{K}) & -\mathbf{Q} \\ -\mathbf{Q}^{\mathrm{T}} & -\left(\dfrac{\bar{\theta}_2}{\theta_2}\mathbf{H} + \dfrac{2}{\theta_2 \Delta t^2}\mathbf{S}\right) \end{bmatrix} \begin{Bmatrix} \boldsymbol{\alpha} \\ \hat{\boldsymbol{\beta}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{Bmatrix} \tag{19.39}$$

where the second equation has been multiplied by -1, the unknown $\boldsymbol{\beta}$ has been replaced by

$$\hat{\boldsymbol{\beta}} = \tfrac{1}{2}\theta_2 \Delta t^2 \boldsymbol{\beta} \tag{19.40}$$

and the forces are given by

$$\begin{aligned} \mathbf{F}_1 &= -\mathbf{f}_{n+1} - \mathbf{C}\dot{\bar{\mathbf{u}}}_{n+1} - \mathbf{K}\bar{\mathbf{u}}_{n+1} + \mathbf{Q}\bar{\mathbf{p}}_{n+1} \\ \mathbf{F}_2 &= \mathbf{q}_{n+1} + \mathbf{H}\bar{\mathbf{p}}_{n+1} \end{aligned} \tag{19.41}$$

It is not necessary to go into detail about the computation steps as these follow the usual patterns of determining $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and then evaluation of the problem variables, that is $\tilde{\mathbf{u}}_{n+1}$, $\tilde{\mathbf{p}}_{n+1}$, $\dot{\tilde{\mathbf{u}}}_{n+1}$ and $\dot{\tilde{\mathbf{p}}}_{n+1}$ at t_{n+1} before proceeding with the next time step. Non-linearity of structural behaviour can readily be accommodated using procedures described in Volume 2.

It is, however, important to consider the stability of the linear system which will, of course, depend on the choice of θ_i and $\bar{\theta}_i$. Here we find, by using procedures described in Chapter 18, that unconditional stability is obtained when

$$\begin{aligned} \theta_2 &\geqslant \theta_1 \qquad & \theta_1 &\geqslant \tfrac{1}{2} \\ \bar{\theta}_2 &\geqslant \theta_2 \qquad & \bar{\theta}_1 &\geqslant \tfrac{1}{2} \end{aligned} \tag{19.42}$$

It is instructive to note that precisely the same result would be obtained if GN22 approximations were used in Eqs (19.34) and (19.35).

The derivation of such stability conditions is straightforward and follows precisely the lines of Sec. 18.3.4 of the previous chapter. However, the algebra is sometimes

tedious. Nevertheless, to allow the reader to repeat such calculations for any case encountered we shall outline the calculations for the present example.

Stability of the fluid–structure time-stepping scheme[30]

For stability evaluations it is always advisable to consider the modally decomposed system with scalar variables. We thus rewrite Eqs (19.36) and (19.37) omitting the forcing terms and putting $\theta_i = \bar{\theta}_i$ as

$$\begin{aligned} m\alpha + c(\dot{u}_n + \theta_1 \Delta t \alpha) + k(u_n + \theta_1 \Delta t \dot{u}_n + \tfrac{1}{2}\theta_2 \Delta t^2 \alpha) \\ - q(p_n + \theta_1 \Delta t \dot{p}_n + \tfrac{1}{2}\theta_2 \Delta t^2 \beta) = 0 \end{aligned} \tag{19.43}$$

and

$$s\beta + q\alpha + h(p_n + \theta_1 \Delta t \dot{p} + \tfrac{1}{2}\theta_2 \Delta t^2 \beta) = 0 \tag{19.44}$$

To complete the recurrence relations we have

$$\begin{aligned} u_{n+1} &= u_n + \Delta t \dot{u}_n + \tfrac{1}{2}\Delta t^2 \alpha \\ \dot{u}_{n+1} &= \dot{u}_n + \Delta t \alpha \\ p_{n+1} &= p_n + \Delta t \dot{p}_n + \tfrac{1}{2}\Delta t^2 \beta \\ \dot{p}_{n+1} &= \dot{p}_n + \Delta t \beta \end{aligned} \tag{19.45}$$

The exact solution of the above system will always be of the form

$$\begin{aligned} u_{n+1} &= \mu u_n \\ \dot{u}_{n+1} &= \mu \dot{u}_n \\ p_{n+1} &= \mu p_n \\ \dot{p}_{n+1} &= \mu \dot{p}_n \end{aligned} \tag{19.46}$$

and immediately we put

$$\mu = \frac{1 - z}{1 + z}$$

knowing that for stability we require the real part of z to be negative.

Eliminating all $n + 1$ values from Eqs (19.45) and (19.46) leads to

$$\begin{aligned} \dot{u}_n &= \frac{2z}{\Delta t} u_n & \dot{p}_n &= \frac{2z}{\Delta t} p_n \\ \alpha &= \frac{4z^2}{(1 - z)\Delta t^2} u_n & \beta &= \frac{4z^2}{(1 - z)\Delta t^2} p_n \end{aligned} \tag{19.47}$$

Inserting (19.47) into the system (19.43) and (19.44) gives

$$\begin{aligned} (a_{11}z^2 + b_{11}z + k)u_n + (a_{12}z^2 + b_{12}z - q)p_n = 0 \\ 4qz^2 u_n + (a_{22}z^2 + b_{22}z + h')p_n = 0 \end{aligned} \tag{19.48}$$

where

$$\begin{aligned} a_{11} &= 4m' - 2(1-2\theta_1)c' - 2k(\theta_1-\theta_2) \\ a_{12} &= 2q(\theta_1-\theta_2) \\ a_{22} &= 4s - 2(\theta_1-\theta_2)h' \\ b_{11} &= 2c' - k(1-2\theta_1) \\ b_{12} &= (1-2\theta_1)q \\ b_{22} &= -(1-2\theta_1)h' \end{aligned} \tag{19.49}$$

in which

$$m' = \frac{m}{\Delta t^2} \qquad c' = \frac{c}{\Delta t} \qquad h' = \Delta t^2 h$$

For non-trivial solutions to exist the determinant of Eq. (19.48) has to be zero. This determinant provides the characteristic equation for z which, in the present case, is a polynomial of fourth order of the form

$$a_0 z^4 + a_1 z^3 + a_2 z^2 + a_3 z + a_4 = 0$$

Thus use of the Routh–Hurwitz conditions given in Sec. 18.3.4 ensures stability requirements are satisfied, i.e., that the roots of z have negative real parts. For the present case the requirements are the following

$$a_0 > 0 \qquad \text{and} \qquad a_i \geqslant 0, \qquad i = 1,2,3,4$$

The inequality

$$a_{11}a_{22} - 8q^2(\theta_1-\theta_2) > 0 \tag{19.50}$$

is satisfied for m', c', k, s, $h' \geqslant 0$ if

$$\theta_1 \geqslant \tfrac{1}{2} \qquad \theta_2 \geqslant \theta_1$$

The inequality

$$a_1 = a_{11}\left[-h'(1-2\theta_1)\right] + \left[2c' - k(1-2\theta_1)\right]a_{22} \geqslant 0 \tag{19.51}$$

is also satisfied if

$$\theta_1 \geqslant \tfrac{1}{2} \qquad \theta_2 \geqslant \theta_1$$

The inequalities

$$a_2 = a_{11}h' + b_{11}b_{22} + a_{22}k + 4q^2 \geqslant 0 \tag{19.52}$$

$$a_3 = b_{11}h' + b_{22}k \geqslant 0 \tag{19.53}$$

are satisfied if (19.50) and (19.51) are satisfied. The inequality

$$a_4 = kh' \geqslant 0 \tag{19.54}$$

is automatically satisfied. Finally the two inequalities

$$a_1a_2 - a_0a_3 \geqslant 0 \tag{19.55}$$

$$a_1a_2a_3 - a_0a_3^2 - a_4a_1^2 \geqslant 0 \tag{19.56}$$

are also satisfied if (19.50) and (19.51) are satisfied.

If all the equalities hold then $m's > 0$ has to be satisfied. In case $m's = 0$ and $c' = 0$ then $\theta_2 > \theta_1$ must be enforced.

19.2.7 Special case of incompressible fluids

If the fluid is incompressible as well as being inviscid, its behaviour is described by a simple laplacian equation

$$\boldsymbol{\nabla}^2 p = 0 \tag{19.58}$$

obtained by putting $c = \infty$ in Eq. (19.5).

In the absence of surface wave effects and of non-zero prescribed pressures the discrete equation (19.28) becomes simply

$$\mathbf{H}\tilde{\mathbf{p}} = -\mathbf{Q}^{\mathrm{T}}\ddot{\tilde{\mathbf{u}}} \tag{19.59}$$

as wave radiation disappears. It is now simple to obtain

$$\tilde{\mathbf{p}} = -\mathbf{H}^{-1}\mathbf{Q}^{\mathrm{T}}\ddot{\tilde{\mathbf{u}}} \tag{19.60}$$

and substitution of the above into the structure equation (19.26) results in

$$\left(\mathbf{M} + \mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^{\mathrm{T}}\right)\ddot{\tilde{\mathbf{u}}} + \mathbf{C}\dot{\tilde{\mathbf{u}}} + \mathbf{K}\tilde{\mathbf{u}} + \mathbf{f} = \mathbf{0} \tag{19.61}$$

This is now a standard structural system in which the mass matrix has been augmented by an *added mass matrix* as

$$\mathbf{M}_u = \mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^{\mathrm{T}} \tag{19.62}$$

and its solution follows the standard procedures of previous chapters.

We have to remark that

1. In general the complete inverse of $\mathbf{H}$ is not required as pressures at interface nodes only are needed.
2. In general the question of when compressibility effects can be ignored is a difficult one and will depend much on the frequencies that have to be considered in the analysis. For instance, in the analysis of the reservoir–dam interaction much debate on the subject has been recorded.[31] Here the fundamental compressible period may be of order H/c where H is a typical dimension (such as height of the dam). If this period is of the same order as that of, say, earthquake forcing motion then, of course, compressibility must be taken into account. If it is much shorter then its neglect can be justified.

19.2.8 Cavitation effects in fluids

In fluids such as water the linear behaviour under volumetric strain ceases when pressures fall below a certain threshold. This is the vapour pressure limit. When this is reached cavities or distributed bubbles form and the pressure remains almost constant. To follow such behaviour a non-linear constitutive law has to be introduced. Although this volume is primarily devoted to linear problems we here indicate some of the steps which are necessary to extend analyses to account for non-linear behaviour.

A convenient variable useful in cavitation analysis was defined by Newton[32]

$$s = \mathrm{div}(\rho\mathbf{u}) \equiv \boldsymbol{\nabla}^{\mathrm{T}}(\rho\mathbf{u}) \tag{19.63}$$

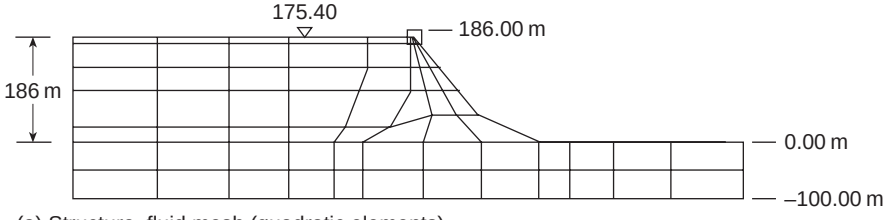
where **u** is the fluid displacement. The non-linearity now is such that

$$\begin{aligned} p &= -K \operatorname{div} \mathbf{u} = c^2 s, && \text{if } s < (p_a - p_v)/c^2 \\ p &= p_a - p_v, && \text{if } s > (p_a - p_v)/c^2 \end{aligned} \tag{19.64}$$

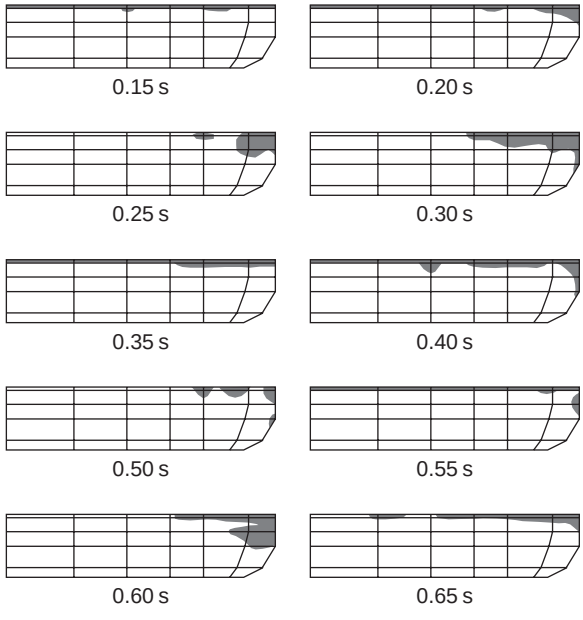
Here p_a is the atmospheric pressure (at which $\mathbf{u} = \mathbf{0}$ is assumed), p_v is the vapour pressure and c is the sound velocity in the fluid.

Clearly monitoring strains is a difficult problem in the formulation using the velocity and pressure variables [Eq. (19.1) and (19.5)]. Here it is convenient to introduce a displacement potential ψ such that

$$\rho \mathbf{u} = -\boldsymbol{\nabla} \psi \tag{19.65}$$



(a) Structure–fluid mesh (quadratic elements)



(b) Zones in which cavitation develops

Fig. 19.5 The Bhakra dam–reservoir system.[33] Interaction during the first second of earthquake motion showing the development of cavitation.

From the momentum equation (19.1) we see that

$$\rho\ddot{\mathbf{u}} = -\boldsymbol{\nabla}\ddot{\psi} = -\boldsymbol{\nabla}p$$

and thus

$$\ddot{\psi} = p \tag{19.66}$$

The continuity equation (19.2) now gives

$$s = \rho \operatorname{div} \mathbf{u} = -\nabla^2\psi = \frac{1}{c^2}p = \frac{1}{c^2}\ddot{\psi} \tag{19.67}$$

in the linear case [with an appropriate change according to conditions (19.64) during cavitation].

Details of boundary conditions, discretization and coupling are fully described in reference 33 and follow the standard methodology previously given. Figure 19.5, taken from that reference, illustrates the results of a non-linear analysis showing the development of cavity zones in a reservoir.

19.3 Soil–pore fluid interaction (Class II problems)

19.3.1 The problem and the governing equations. Discretization

It is well known that the behaviour of soils (and indeed other geomaterials) is strongly influenced by the pressures of the fluid present in the pores of the material. Indeed, the concept of *effective stress* is here of paramount importance. Thus if $\boldsymbol{\sigma}$ describes the total stress (positive in tension) acting on the total area of the soil and the pores, and p is the pressure of the fluid (positive in compression) in the pores (generally of water), the effective stress is defined as

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} + \mathbf{m}p \tag{19.68}$$

Here $\mathbf{m}^{\mathrm{T}} = [1, 1, 1, 0, 0, 0]$ if we use the notation in Chapter 12. Now it is well known that it is only the stress $\boldsymbol{\sigma}'$ which is responsible for the deformations (or failure) of the solid skeleton of the soil (excluding here a very small volumetric grain compression which has to be included in some cases). Assuming for the development given here that the soil can be represented by a linear elastic model we have

$$\boldsymbol{\sigma}' = \mathbf{D}\boldsymbol{\varepsilon} \tag{19.69}$$

Immediately the total discrete equilibrium equations for the soil–fluid mixture can be written in exactly the same form as is done for all problems of solid mechanics:

$$\mathbf{M}\ddot{\tilde{\mathbf{u}}} + \mathbf{C}\dot{\tilde{\mathbf{u}}} + \int_{\Omega} \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}\, \mathrm{d}\Omega + \mathbf{f} = \mathbf{0} \tag{19.70}$$

where $\tilde{\mathbf{u}}$ are the displacement discretization parameters, i.e.

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}\tilde{\mathbf{u}} \tag{19.71}$$

$\mathbf{B}$ is the strain–displacement matrix and $\mathbf{M}$, $\mathbf{C}$, $\mathbf{f}$ have the usual meaning of mass, damping and force matrices, respectively.

Now, however, the term involving the stress must be split as

$$\int_\Omega \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega = \int_\Omega \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}'\,\mathrm{d}\Omega - \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{m}p\,\mathrm{d}\Omega \tag{19.72}$$

to allow the direct relationship between effective stresses and strains (and hence displacements) to be incorporated. For a linear elastic soil skeleton we immediately have

$$\mathbf{M}\ddot{\tilde{\mathbf{u}}} + \mathbf{C}\dot{\tilde{\mathbf{u}}} + \mathbf{K}\tilde{\mathbf{u}} - \mathbf{Q}\tilde{\mathbf{p}} + \mathbf{f} = \mathbf{0} \tag{19.73}$$

where K is the standard stiffness matrix written as

$$\int_\Omega \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}'\,\mathrm{d}\Omega = \left(\int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d}\Omega\right)\tilde{\mathbf{u}} = \mathbf{K}\tilde{\mathbf{u}} \tag{19.74}$$

and $\mathbf{Q}$ couples the field of pressures in the equilibrium equations assuming these are discretized as

$$p \approx \hat{p} = \mathbf{N}_p\tilde{\mathbf{p}} \tag{19.75}$$

Thus

$$\mathbf{Q} = \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{m}\mathbf{N}_p\,\mathrm{d}\Omega \tag{19.76}$$

In the above discretization conventionally the same element shapes are used for the $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{p}}$ variables, though not necessarily identical interpolations. With the dynamic equations coupled to the pressure field an additional equation is clearly needed from which the pressure field can be derived. This is provided by the transient seepage equation of the form

$$-\boldsymbol{\nabla}^{\mathrm{T}}(k\boldsymbol{\nabla}p) + \frac{1}{Q}\dot{p} + \dot{\varepsilon}_v = 0 \tag{19.77}$$

where Q is related to the compressibility of the fluid, k is the permeability and ε_v is the volumetric strain in the soil skeleton, which on discretization of displacements is given by

$$\varepsilon_v = \mathbf{m}^{\mathrm{T}}\boldsymbol{\varepsilon} = \mathbf{m}^{\mathrm{T}}\mathbf{B}\tilde{\mathbf{u}} \tag{19.78}$$

The equation of seepage can now be discretized in the standard Galerkin manner as

$$\mathbf{Q}^{\mathrm{T}}\dot{\tilde{\mathbf{u}}} + \mathbf{S}\dot{\tilde{\mathbf{p}}} + \mathbf{H}\tilde{\mathbf{p}} + \mathbf{q} = \mathbf{0} \tag{19.79}$$

where $\mathbf{Q}$ is precisely that of Eq. (19.76), and

$$\mathbf{S} = \int_\Omega \mathbf{N}_p^{\mathrm{T}}\frac{1}{Q}\mathbf{N}_p\,\mathrm{d}\Omega \qquad \mathbf{H} = \int_\Omega (\boldsymbol{\nabla}\mathbf{N}_p)^{\mathrm{T}}k\boldsymbol{\nabla}\mathbf{N}_p\,\mathrm{d}\Omega \tag{19.80}$$

with $\mathbf{q}$ containing the forcing and boundary terms. The derivation of coupled flow–soil equations was first introduced by Biot[34] but the present formulation is elaborated upon in references 30 to 37 where various approximations, as well as the effect of various non-linear constitutive relations, are discussed.

We shall not comment in detail on any of the boundary conditions as these are of standard type and are well documented in previous chapters.

19.3.2 The format of the coupled equations

The solution of coupled equations often involves non-linear behaviour, as noted previously in the cavitation problem. However, it is instructive to consider the linear version of Eqs (19.73) and (19.79). This can be written as

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\tilde{\mathbf{u}}} \\ \ddot{\tilde{\mathbf{p}}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{Q}^{\mathrm{T}} & \mathbf{S} \end{bmatrix} \begin{Bmatrix} \dot{\tilde{\mathbf{u}}} \\ \dot{\tilde{\mathbf{p}}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{Bmatrix} = - \begin{Bmatrix} \tilde{\mathbf{f}} \\ \tilde{\mathbf{q}} \end{Bmatrix} \tag{19.81}$$

Once again, like in the fluid–structure interaction problem, overall asymmetry occurs despite the inherent symmetry of the **M**, **C**, **K**, **S** and **H** matrices. As the free vibration problem is of no great interest here, we shall not discuss its symmetrization. In the transient solution algorithm we shall proceed in a similar manner to that described in Sec. 19.2.6 and again symmetry will be observed.

19.3.3 Transient step-by-step algorithm

Time-stepping procedures can be derived in a manner analogous to that presented in Sec. 19.2.6. Here we choose to use the GNpj algorithm of lowest order to approximate each variable.

Thus for $\tilde{\mathbf{u}}$ we shall use GN22, writing

$$\begin{aligned} \tilde{\mathbf{u}}_{n+1} &= \tilde{\mathbf{u}}_n + \Delta t \dot{\tilde{\mathbf{u}}}_n + \tfrac{1}{2}\Delta t^2 \ddot{\tilde{\mathbf{u}}}_n + \tfrac{1}{2}\beta_2 \Delta t^2 \Delta \ddot{\tilde{\mathbf{u}}}_{n+1} \\ &\equiv \tilde{\mathbf{u}}^p_{n+1} + \tfrac{1}{2}\beta_2 \Delta t^2 \Delta \ddot{\tilde{\mathbf{u}}}_{n+1} \\ \dot{\tilde{\mathbf{u}}}_{n+1} &= \dot{\tilde{\mathbf{u}}}_n + \Delta t \ddot{\tilde{\mathbf{u}}}_n + \beta_1 \Delta t \Delta \ddot{\tilde{\mathbf{u}}}_{n+1} \\ &\equiv \dot{\tilde{\mathbf{u}}}^p_{n+1} + \beta_1 \Delta t \Delta \ddot{\tilde{\mathbf{u}}}_{n+1} \end{aligned} \tag{19.82}$$

For the variables p that occur in first-order form we shall use GN11, as

$$\begin{aligned} \tilde{\mathbf{p}}_{n+1} &= \tilde{\mathbf{p}}_n + \Delta t \dot{\tilde{\mathbf{u}}}_n + \theta \Delta t \Delta \dot{\tilde{\mathbf{u}}}_{n+1} \\ &\equiv \tilde{\mathbf{p}}^p_{n+1} + \theta \Delta t \Delta \dot{\tilde{\mathbf{p}}}_{n+1} \end{aligned} \tag{19.83}$$

In the above $\tilde{\mathbf{u}}^p_{n+1}$, etc., denote values that can be 'predicted' from known parameters at time t_n and

$$\Delta \ddot{\tilde{\mathbf{u}}}_{n+1} = \ddot{\tilde{\mathbf{u}}}_{n+1} - \ddot{\tilde{\mathbf{u}}}_n \qquad \Delta \dot{\tilde{\mathbf{p}}}_{n+1} = \dot{\tilde{\mathbf{p}}}_{n+1} - \dot{\tilde{\mathbf{p}}}_n \tag{19.84}$$

are the unknowns.

To complete the recurrence algorithm it is necessary to insert the above into the coupled governing equations [(19.70) and (19.79)] written at time t_{n+1}. Thus we require the following equalities

$$\begin{aligned} \mathbf{M}\ddot{\tilde{\mathbf{u}}}_{n+1} + \mathbf{C}\dot{\tilde{\mathbf{u}}}_{n+1} + \int_\Omega \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma}'_{n+1} - \mathbf{Q}\tilde{\mathbf{p}}_{n+1} + \mathbf{f}_{n+1} &= \mathbf{0} \\ \mathbf{Q}^{\mathrm{T}}\dot{\tilde{\mathbf{u}}}_{n+1} + \mathbf{S}\dot{\tilde{\mathbf{p}}}_{n+1} + \mathbf{H}\tilde{\mathbf{p}}_{n+1} + \mathbf{q}_{n+1} &= \mathbf{0} \end{aligned} \tag{19.85}$$

in which $\boldsymbol{\sigma}'_{n+1}$ is evaluated using the constitutive equation (19.69) in incremental form and knowledge of $\boldsymbol{\sigma}'_n$ as

$$\boldsymbol{\sigma}'_{n+1} = \boldsymbol{\sigma}'_n + \mathbf{D}\Delta\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\sigma}'_n + \mathbf{DB}\Delta\tilde{\mathbf{u}}_{n+1} \tag{19.86}$$

In general the above system is non-linear and indeed on many occasions the $\mathbf{H}$ matrix itself may be dependent on the values of $\mathbf{u}$ due to permeability variations with strain. Solution methods of such non-linear systems will be discussed in Volume 2; however, it is of interest to look at the linear form as the non-linear system usually solves a similar form iteratively.

Here insertion of Eqs (19.82), (19.83) and (19.86) into (19.85) results in the equation system

$$\begin{bmatrix} (\mathbf{M} + \beta_1\Delta t\mathbf{C} + \tfrac{1}{2}\beta_2\Delta t^2\mathbf{K}) & -\mathbf{Q} \\ -\mathbf{Q}^{\mathrm{T}} & -\left(\mathbf{H} + \dfrac{1}{\theta\Delta t}\mathbf{S}\right) \end{bmatrix} \begin{Bmatrix} \Delta\ddot{\tilde{\mathbf{u}}}_{n+1} \\ \Delta\dot{\tilde{\mathbf{p}}}_{n+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{Bmatrix} \tag{19.87}$$

where $\mathbf{F}_1$ and $\mathbf{F}_2$ are vectors that can be evaluated from loads and solution values at t_n. Symmetry in the above is obtained by multiplying Eq. (19.37) by -1 and defining

$$\Delta\dot{\tilde{\mathbf{p}}}_{n+1} = \beta_1\Delta t\Delta\dot{\tilde{\mathbf{p}}}_{n+1} \tag{19.88}$$

The solution of Eq. (19.87) and the use of Eqs (19.82) and (19.83) complete the recurrence relation.

The stability of the linear scheme can be found by following identical procedures to those used in Sec. 19.2.6 and the result is[25] that stability is unconditional when

$$\beta_2 \geqslant \beta_1 \qquad \beta_1 \geqslant \tfrac{1}{2} \qquad \theta \geqslant \tfrac{1}{2} \tag{19.89}$$

19.3.4 Special cases and robustness requirements

Frequently the compressibility of the fluid phase, which forms the matrix $\mathbf{S}$, is such that

$$\mathbf{S} \approx \mathbf{0}$$

compared with other terms. Further, the permeability k may on occasion also be very small (as, say, in clays) and

$$\mathbf{H} \approx \mathbf{0}$$

leading to so-called 'undrained' behaviour.

Now the coefficient matrix in (19.87) becomes of the lagrangian constrained form (see Chapter 11), i.e.

$$\begin{bmatrix} \mathbf{A} & -\mathbf{Q} \\ -\mathbf{Q}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \Delta\ddot{\tilde{\mathbf{u}}}_{n+1} \\ \Delta\dot{\tilde{\mathbf{p}}}_{n+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{Bmatrix} \tag{19.90}$$

and is solvable only if

$$n_u \geqslant n_p$$

where n_u and n_p denote the number of $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{p}}$ parameters, respectively.

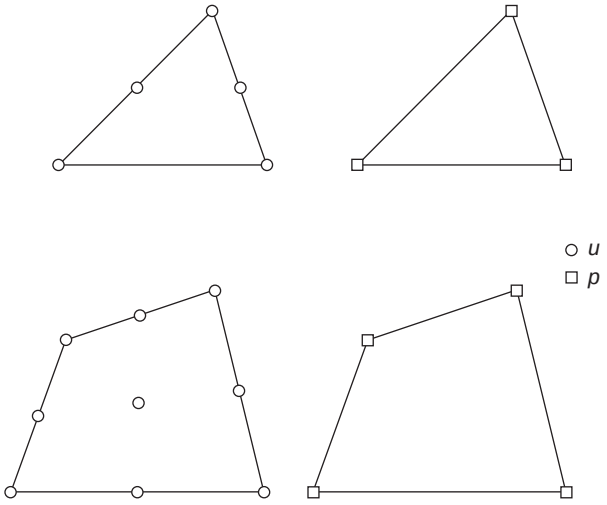


Fig. 19.6 'Robust' interpolations for the coupled soil–fluid problem.

The problem is indeed identical to that encountered in incompressible behaviour and the interpolations used for the $\mathbf{u}$ and p variables have to satisfy identical criteria. As C_0 interpolation for both variables is necessary for the general case, suitable element forms are shown in Fig. 19.6 and can be used with confidence.

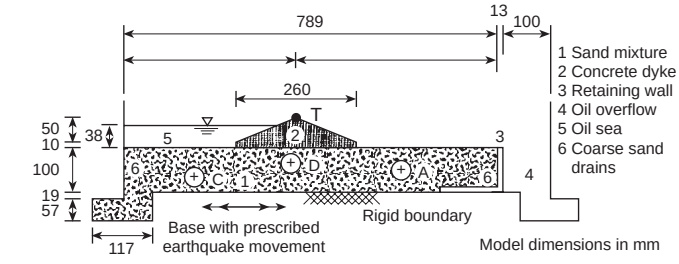
The formulation can of course be used for steady-state solutions but it must be remarked that in such cases an uncoupling occurs as the seepage equation can be solved independently.

Finally, it is worth remarking that the formulation also solves the well-known soil consolidation problem where the phenomena are so slow that the dynamic term $\mathbf{M}\ddot{\mathbf{u}}$ tends to $\mathbf{0}$. However, no special modifications are necessary and the algorithm form is again applicable.

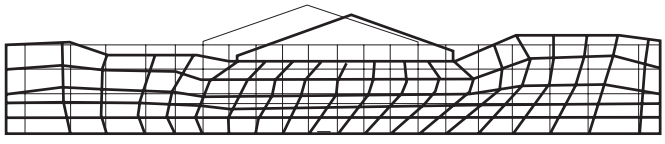
19.3.5 Examples – soil liquefaction

As we have already mentioned, the most interesting applications of the coupled soil–fluid behaviour is when non-linear soil properties are taken into account. In particular, it is a well-known fact that repeated straining of a granular, soil-like material in the absence of the pore fluid results in a decrease of volume (densification) due to particle rearrangement. In Volume 2 we present constitutive equations which include this effect and here we only represent a typical result which they can achieve when used in a coupled soil–fluid solution. When a pore fluid is present, densification will (via the coupling terms) tend to increase the fluid pressures and hence reduce the soil strength. This, as is well known, decreases with the compressive mean effective stress.

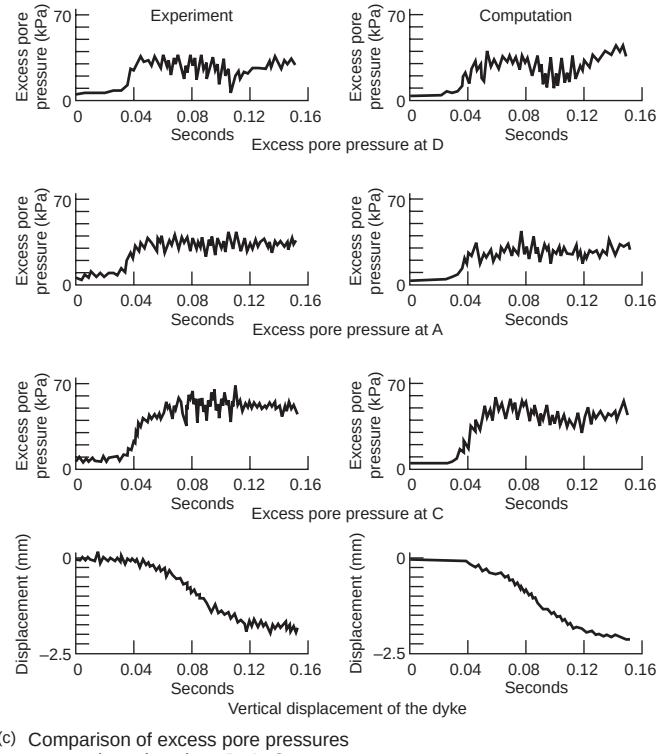
It is not surprising therefore that under dynamic action the soil frequently loses all of its strength (i.e., liquefies) and behaves almost like a fluid, leading occasionally to catastrophic failures of structural foundations in earthquakes. The reproduction of such phenomena with computational models is not easy as a complete constitutive



(a) Outline of centrifuge model.
A, D, C are location of pressure transducers
for which comparisons are shown,
T is displacement transducer

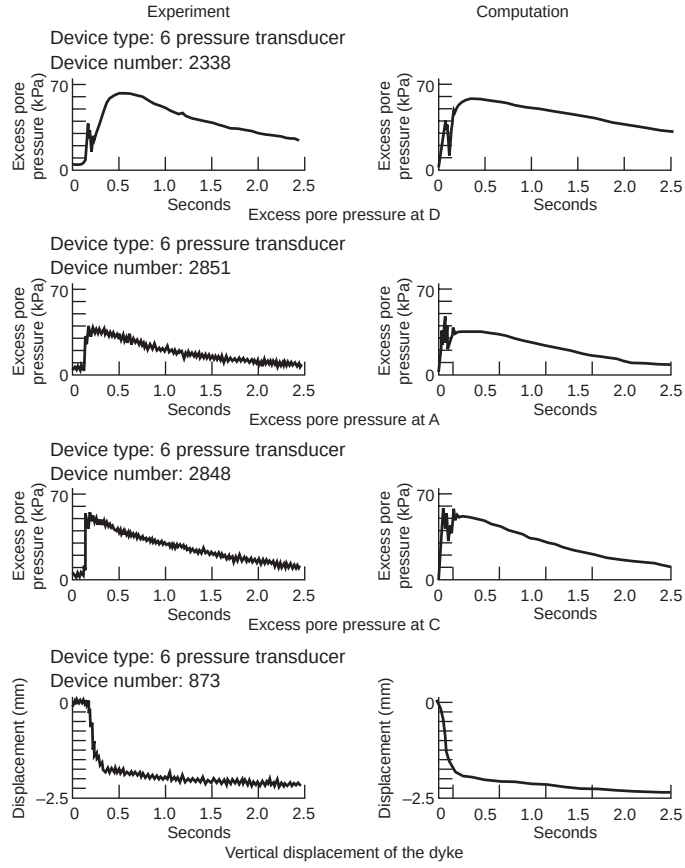


(b) Finite element mesh and final permanent
distorted shape (x 10 magnification)



(c) Comparison of excess pore pressures
at transducer locations D, A, C,
and displacement at T (top of dyke)
0 < time < 0.16 seconds,
earthquake stops at $t = 0.12$ s

Fig. 19.7 Soil–pressure water interaction. Computation and centrifuge model results compared on a problem of a dyke foundation subject to a simulated earthquake.



(d) Comparison of excess pore pressures at transducer locations D, A, C, and displacement at T (top of dyke) 0 < time < 2.5 seconds, Note consolidation process

Fig. 19.7 Continued.

behaviour description for soils is imperfect. However, much effort devoted to the subject has produced good results[35–42] and a reasonable confidence in predictions achieved by comparison with experimental studies exists. One such study is illustrated in Fig. 19.7 where a comparison with tests carried out in a centrifuge is made.[41,42] In particular the close correlation between computed pressure and displacement with experiments should be noted.

19.3.6 Biomechanics, oil recovery and other applications

The interaction between a porous medium and interstitial fluid is not confined to soils. The same equations describe, for instance, the biomechanics problem of bone–fluid interaction *in vivo*. Applications in this field have been documented.[43,44]

On occasion two (or more) fluids are present in the pores and here similar equations can again be written[45,46] to describe the interaction. Problems of ground settlement in oil fields due to oil extraction, or flow of water/oil mixtures in oil recovery are good examples of application of techniques described here.

19.4 Partitioned single-phase systems – implicit–explicit partitions (Class I problems)

In Fig. 19.1(b), describing problems coupled by an interface, we have already indicated the possibility of a structure being partitioned into substructures and linked along an interface only. Here the substructures will in general be of a similar kind but may differ in the manner (or simply size) of discretization used in each or even in the transient recurrence algorithms employed. In Chapter 13 we have described special kinds of mixed formulations allowing the linking of domains in which, say, boundary-type approximations are used in one and standard finite elements in the other. We shall not return to this phase and will simply assume that the total system can be described using such procedures by a single set of equations in time. Here we shall only consider a first-order problem (but a similar approach can be extended to the second-order dynamic system):

$$\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{19.91}$$

which can be partitioned into two (or more) components, writing

$$\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{a}}_1 \\ \dot{\mathbf{a}}_2 \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{Bmatrix} + \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{19.92}$$

Now for various reasons it may be desirable to use in each partition a different time-step algorithm. Here we shall assume the same structure of the algorithm (SS11) and the same time step (Δt) but simply a different parameter θ in each. Proceeding thus as in the other coupled analyses we write

$$\begin{aligned} \mathbf{a}_1 &= \mathbf{a}_{1n} + \tau\boldsymbol{\alpha}_1 \\ \mathbf{a}_2 &= \mathbf{a}_{2n} + \tau\boldsymbol{\alpha}_2 \end{aligned} \tag{19.93}$$

Inserting the above into each of the partitions and using different weight functions, we obtain

$$\mathbf{C}_{11}\boldsymbol{\alpha}_1 + \mathbf{C}_{12}\boldsymbol{\alpha}_2 + \mathbf{K}_{11}(\mathbf{a}_{1n} + \theta\Delta t\boldsymbol{\alpha}_1) + \mathbf{K}_{12}(\mathbf{a}_{2n} + \theta\Delta t\boldsymbol{\alpha}_2) + \bar{\mathbf{f}}_1 = \mathbf{0} \tag{19.94}$$

$$\mathbf{C}_{21}\boldsymbol{\alpha}_1 + \mathbf{C}_{22}\boldsymbol{\alpha}_2 + \mathbf{K}_{21}(\mathbf{a}_{1n} + \bar{\theta}\Delta t\boldsymbol{\alpha}_1) + \mathbf{K}_{22}(\mathbf{a}_{2n} + \bar{\theta}\Delta t\boldsymbol{\alpha}_2) + \bar{\mathbf{f}}_2 = \mathbf{0} \tag{19.95}$$

This system may be solved in the usual manner for $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ and recurrence relations obtained even if θ and $\bar{\theta}$ differ. The remaining details of the time-step calculations follow the obvious pattern but the question of coupling stability must be addressed. Details of such stability evaluation in this case are given elsewhere[47] but the result is interesting.

1. Unconditional stability of the whole system occurs if

$$\theta \geqslant \tfrac{1}{2} \qquad \bar{\theta} \geqslant \tfrac{1}{2}$$

2. Conditional stability requires that

$$\Delta t \leqslant \Delta t_{\text{crit}}$$

where the Δt_{crit} condition is that pertaining to each partitioned system considered *without its coupling terms*.

Indeed, similar results will be obtained for the second-order systems

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{f} = \mathbf{0} \tag{19.96}$$

partitioned in a similar manner with SS22 or GN22 used in each.

The reader may well ask why different schemes should be used in each partition of the domain. The answer in the case of *implicit–implicit* schemes may be simply the desire to introduce different degrees of algorithmic damping. However, much more important is the use of *implicit–explicit* partitions. As we have shown in both 'thermal' and dynamic-type problems the critical time step is inversely proportional to h^2 and h (the element size), respectively. Clearly if a single explicit scheme were to be used with very small elements (or very large material property differences) occurring in one partition, this time step may become too short for economy to be preserved in its use. In such cases it may be advantageous to use an explicit scheme (with $\theta = 0$ in first-order problems, $\theta_2 = 0$ in dynamics) for a part of the domain with larger elements while maintaining unconditional stability with the same time step in the partition in which elements are small or otherwise very 'stiff'. For this reason such implicit–explicit partitions are frequently used in practice.

Indeed, with a lumped representation of matrices $\mathbf{C}$ or $\mathbf{M}$ such schemes are in effect *staggered* as the explicit part can be advanced independently of the implicit part and immediately provides the boundary values for the implicit partition. We shall return to such staggered solutions in the next section.

The use of explicit–implicit partitions was first recorded in 1978.[48–50] In the first reference the process is given in an identical manner as presented here; in the second, a different algorithm is given based on an element split (instead of the implied nodal split above) as described next.

Implicit–explicit solution – element partition

We again consider the first-order problem given in Eq. (19.91) and split as

$$\mathbf{C}_I\dot{\mathbf{a}}_I + \mathbf{C}_E\dot{\mathbf{a}}_E + \mathbf{K}_I\mathbf{a}_I + \mathbf{K}_E\mathbf{a}_E + \mathbf{f} = \mathbf{0} \tag{19.97}$$

where the subscript I denotes an implicit partition and subscript E an explicit one. The recurrence relation for $\mathbf{a}$ is now written using GN11 as

$$\mathbf{a}_{n+1}^{(j)} = \mathbf{a}_n + (1-\theta)\Delta t\dot{\mathbf{a}}_n + \theta\Delta t\dot{\mathbf{a}}_{n+1}^{(j)} \tag{19.98}$$

with

$$\mathbf{a}_{n+1}^{(0)} = \mathbf{a}_n + (1-\theta)\Delta t\dot{\mathbf{a}}_n \tag{19.99}$$

The approximations for the split are now taken as

$$\mathbf{a}_I = \mathbf{a}_{n+1}^{(j)}$$

$$\mathbf{a}_E = \mathbf{a}_{n+1}^{(j-1)}$$

$$\dot{\mathbf{a}}_I = \dot{\mathbf{a}}_E = \dot{\mathbf{a}}_{n+1}^{(j)}$$

thus yielding the system of equations at iteration j as

$$(\mathbf{C} + \theta \Delta t \mathbf{K}_I)\dot{\mathbf{a}}_{n+1}^{(j)} + \mathbf{F}^{(j)} = \mathbf{0} \tag{19.100}$$

where $\mathbf{F}^{(j)}$ contains the loading terms which depend on known values at t_n and previous iterate values $(j-1)$. The above algorithm has stability properties which depend on the choice of θ. For a linear system with $\theta \geqslant 0.5$ the implicit part is unconditionally stable and stability depends on the Δt_{crit} of the explicit elements.[49,50] Performing only one iteration in each time step is permitted; however improved accuracy in the explicit partition can occur if additional iterations are used, although the cost of each time step is obviously increased.

19.5 Staggered solution processes

19.5.1 General remarks

We have observed in the previous section that in the nodal based implicit–explicit partitioning of time stepping it was possible to proceed in a *staggered* fashion, achieving a complete solution of the explicit scheme independently of the implicit one and then using the results to progress with the implicit partition. It is tempting to examine the possibility of such staggered procedures generally even if each uses an independent algorithm.

In such procedures the first equation would be solved with some assumed (predicted) values for the variable of the other. Once the solution for the first system was obtained its values could be substituted in the second system, again allowing its independent treatment. If such procedures can be made stable and reasonably accurate many possibilities are immediately open, for instance:

1. Completely different methodologies could be used in each part of the coupled system.
2. Independently developed codes dealing efficiently with single systems could be combined.
3. Parallel computation with its inherent advantages could be used.
4. Finally, in systems of the same physics, efficient iterative solvers could easily be developed.

The problems of such staggered solutions have been frequently discussed[33,51–54] and on occasion unconditional stability could not be achieved without substantial modification. In the following we shall indicate some options available.

19.5.2 Staggered process of solution in single-phase systems

We shall look at this possibility first, having already mentioned it as a special form arising naturally in the implicit–explicit processes of Sec. 19.4. We return here to

consider the problem of Eq. (19.91) and the partitioning given in Eq. (19.92). Further, for simplicity we shall assume a diagonal form of the **C** matrix, i.e., that the problem is posed as

$$\begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{a}}_1 \\ \dot{\mathbf{a}}_2 \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{Bmatrix} + \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{19.101}$$

As we have already remarked, the use of $\theta = 0$ in the first equation and $\bar{\theta} \geqslant 0.5$ in the second [see Eqs (19.94) and (19.95)] allowed the explicit part to be solved independently of the implicit. Now, however, we shall use the same θ in both equations but in the first of the approximations, analogous to Eq. (19.94), we shall insert a predicted value for the second variable:

$$\mathbf{a}_2 = \mathbf{a}_2^p = \mathbf{a}_{2n} \tag{19.102}$$

This is similar to the treatment of the explicit part in the element split of the implicit–explicit scheme and gives in place of Eq. (19.94)

$$\mathbf{C}_{11}\boldsymbol{\alpha}_1 + \mathbf{K}_{11}(\mathbf{a}_{1n} + \theta\Delta t\boldsymbol{\alpha}_1) = -\mathbf{f}_1 - \mathbf{K}_{12}\mathbf{a}_{2n} \tag{19.103}$$

allowing direct solution for $\boldsymbol{\alpha}_1$.

Following this step, the second equation can be solved for $\boldsymbol{\alpha}_2$ with the previous value of $\boldsymbol{\alpha}_1$ inserted, i.e.

$$\mathbf{C}_{22}\boldsymbol{\alpha}_2 + \mathbf{K}_{22}(\mathbf{a}_{2n} + \theta\Delta t\boldsymbol{\alpha}_2) = -\mathbf{f}_2 - \mathbf{K}_{21}(\mathbf{a}_{1n} + \theta\Delta t\boldsymbol{\alpha}_1) \tag{19.104}$$

This scheme is unconditionally stable if $\theta \geqslant 0.5$, i.e., the total system is stable provided each stagger is unconditionally stable. A similar condition holds for linear second-order dynamic problems.

Obviously, however, some accuracy will be lost as the approximation of Eq. (19.103) is that of the explicit form in $\mathbf{a}_2$. The approximation is consistent and hence convergence will occur.

The advantage of using the staggered process in the above is clear as the equation solving, even though not explicit, is now confined to the magnitude of each partition and computational economy occurs.

Futher, it is obvious that precisely the same procedures can be used for any number of partitions and that again the same stability conditions will apply. Define the arrays

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & & & & & \\ & \mathbf{C}_{22} & & & & \\ & & \ddots & & & \\ & & & \mathbf{C}_{ii} & & \\ & & & & \ddots & \\ & & & & & \mathbf{C}_{kk} \end{bmatrix} \tag{19.105}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & & & \vdots \\ \vdots & & \ddots & & \\ & & & \mathbf{K}_{ii} & \\ \vdots & & & & \ddots \quad \mathbf{0} \\ \mathbf{K}_{k1} & \cdots & & \mathbf{K}_{k,k-1} & \mathbf{K}_{kk} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{K}_{12} & \cdots & & \mathbf{K}_{1k} \\ \mathbf{0} & \mathbf{0} & \cdots & & \vdots \\ \vdots & & \ddots & & \\ & & & \mathbf{0} & \vdots \\ & & & & \ddots \quad \mathbf{K}_{k-1,k} \\ \mathbf{0} & & & \cdots & \mathbf{0} \end{bmatrix}$$

$$= \mathbf{K}_L + \mathbf{K}_U \tag{19.106}$$

and consider the partition

$$\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}_L\mathbf{a} + \mathbf{K}_U\mathbf{a}^p + \mathbf{f} = \mathbf{0} \tag{19.107}$$

Introducing now the approximation

$$\mathbf{a}_i = \mathbf{a}_{in} + \tau\boldsymbol{\alpha}_i \tag{19.108}$$

and using Eq. (19.102) gives the discrete form

$$\begin{aligned} \mathbf{C}\boldsymbol{\alpha} + \mathbf{K}_L(\mathbf{a}_n + \theta\Delta t\boldsymbol{\alpha}) + \mathbf{K}_U\mathbf{a}_n + \bar{\mathbf{f}} = \mathbf{0} \\ (\mathbf{C} + \mathbf{K}_L\theta\Delta t\boldsymbol{\alpha}) + \mathbf{K}\mathbf{a}_n + \bar{\mathbf{f}} = \mathbf{0} \end{aligned} \tag{19.109}$$

In approximating the first equation set it is necessary to use predicted values for $\mathbf{a}_2$, $\mathbf{a}_3, \cdots, \mathbf{a}_k$, writing in place of Eq. (19.103),

$$\mathbf{C}_{11}\boldsymbol{\alpha}_1 + \mathbf{K}_{11}(\mathbf{a}_{1n} + \theta\Delta t\boldsymbol{\alpha}_1) + \mathbf{K}_{12}\mathbf{a}_{2n} + \mathbf{K}_{13}\mathbf{a}_{3n} + \cdots + \mathbf{f}_1 = \mathbf{0} \tag{19.110}$$

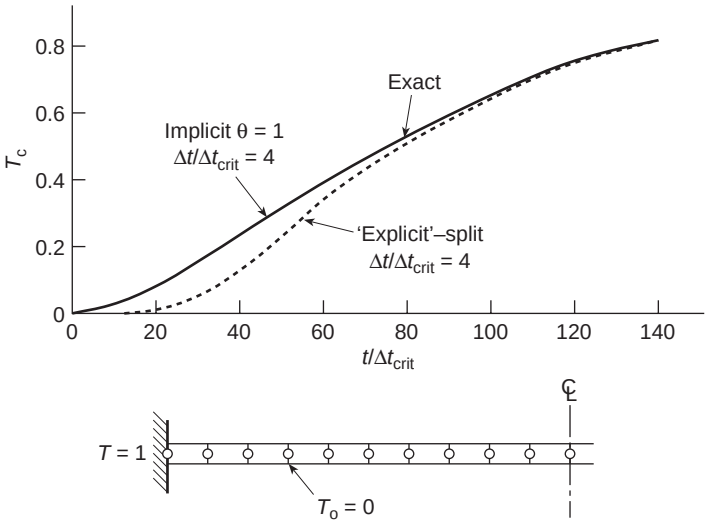
and continue similarly to (19.104), with the predicted values now continually being replaced by better approximations as the solution progresses.

The partitioning of Eq. (19.105) can be continued until only a single equation set is obtained. Then at each step the equation that requires solving for $\boldsymbol{\alpha}_i$ is of the form

$$(\mathbf{C}_{ii} + \theta\Delta t\mathbf{K}_{ii})\boldsymbol{\alpha}_i = \mathbf{F}_i \tag{19.111}$$

where $\mathbf{F}_i$ contains the effects of the load and all the previously computed $\mathbf{a}_i$. For partitions where each submatrix is a scalar Eq. (19.111) is a scalar equation and computation is thus *fully explicit and yet preserves unconditional stability* for $\theta \geqslant 0.5$. This type of partitioning and the derivation of an unconditionally stable explicit scheme was first proposed by Zienkiewicz *et. al.*[55] An alternative and somewhat more limited scheme of a similar kind was given by Trujillo.[56]

Clearly the error in the approximation in the time step decreases as the solution sweeps through the partitions and hence it is advisable to alter the sweep directions during the computation. For instance, in Fig. 19.8 we show quite reasonable accuracy for a one-dimensional heat-conduction problem in which the *explicit-split* process was used with alternating direction of sweeps. Of course the accuracy is much inferior to that exhibited by a standard implicit scheme with the same time step, though the process could be used quite effectively as an iteration to obtain steady-state solutions. Here many other options are also possible.



Δt_{crit} = critical time step for standard explicit form
T_c = temperature on centre-line

Fig. 19.8 Accuracy of an explicit-split procedure compared with a standard implicit process for heat conduction of a bar.

It is, for instance, of interest to consider the system given in Eq. (19.105) as originating from a simple finite difference approximation to, say, a heat-conduction equation on the rectangular mesh of Fig. 19.9.

Here it is well known that the so-called alternating direction implicit (ADI) scheme[57] presents an efficient solution for both transient and steady-state problems. It is fairly obvious that the scheme simply represents the procedure just outlined with partitions representing lines of nodes such as $(1, 5, 9, 13)$, $(2, 6, 10, 14)$, etc., of Fig 19.9 alternating with partitions $(1, 2, 3, 4)$, $(5, 6, 7, 8)$, etc.

Obviously the bigger the partition, the more accurate the scheme becomes, though of course at the expense of computational costs. The concept of the staggered partition clearly allows easy adoption of such procedures in the finite element context. Here irregular partitions arbitrarily chosen could be made but so far applications

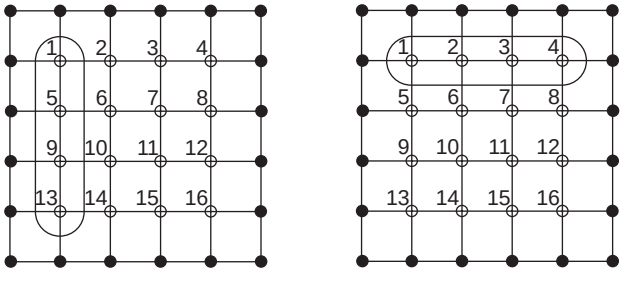


Fig. 19.9 Partitions corresponding to the well-known ADI (alternating direction implicit) finite difference scheme.

have only been recorded in regular mesh subdivisions.[57] The field of possibilities is obviously large. Use in parallel computation is obvious for such procedures.

A further possibility which has many advantages is to use hierarchical variables based on, say, linear, quadratic and higher expansions and to consider each set of these variables as a partition.[58] Such procedures are particularly efficient in iteration if coupled with suitable preconditioning[59] and form a basis of *multigrid procedures*.

19.5.3 Staggered schemes in fluid–structure systems and stabilization processes

The application of staggered solution methods in coupled problems representing different phenomena is more obvious, though, as it turns out, more difficult.

For instance, let us consider the linear discrete fluid–structure equations with damping omitted, written as [see Eqs (19.26) and (19.28)]

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{Q}^{\mathrm{T}} & \mathbf{S} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} + \begin{Bmatrix} \mathbf{f} \\ \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{19.112}$$

where we have omitted the tilde superscript for simplicity.

For illustration purposes we shall use the GN22 type of approximation for both variables and write using Eq. (19.82)

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}^{p}_{n+1} + \tfrac{1}{2}\beta_2 \Delta t^2 \Delta\ddot{\mathbf{u}}_{n+1} \\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}^{p}_{n+1} + \beta_1 \Delta t \Delta\ddot{\mathbf{u}}_{n+1} \\ \mathbf{p}_{n+1} &= \mathbf{p}^{p}_{n+1} + \tfrac{1}{2}\bar{\beta}_2 \Delta t^2 \Delta\ddot{\mathbf{p}}_{n+1} \\ \dot{\mathbf{p}}_{n+1} &= \dot{\mathbf{p}}^{p}_{n+1} + \bar{\beta}_1 \Delta t \Delta\ddot{\mathbf{p}}_{n+1} \end{aligned} \tag{19.113}$$

which together with Eq. (19.112) written at $t = t_{n+1}$ completes the system of equations requiring simultaneous solution for $\Delta\ddot{\mathbf{u}}_{n+1}$ and $\Delta\ddot{\mathbf{p}}_{n+1}$.

Now a staggered solution of a fairly obvious kind would be to write the first set of equations (19.112) corresponding to the structural behaviour with a predicted (approximate) value of $\mathbf{p}_{n+1} = \mathbf{p}^{p}_{n+1}$, as this would allow an independent solution for $\Delta\ddot{\mathbf{u}}_{n+1}$ writing

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + \mathbf{K}\mathbf{u}_{n+1} = -\mathbf{f} + \mathbf{Q}\mathbf{p}^{p}_{n+1} \tag{19.114}$$

This would then be followed by the solution of the fluid problem for $\Delta\ddot{\mathbf{p}}_{n+1}$ writing

$$\mathbf{S}\ddot{\mathbf{p}}_{n+1} + \mathbf{H}\mathbf{u}_{n+1} = -\mathbf{q} - \mathbf{Q}^{\mathrm{T}}\ddot{\mathbf{u}}_{n+1} \tag{19.115}$$

This scheme turns out, however, to be only conditionally stable,[47] even if β_i and $\bar{\beta}_i$ are chosen so that unconditional stability of a simultaneous solution is achieved. (The stability limit is indeed the same as if a fully explicit scheme were chosen for *the fluid phase*.)

Various stabilization schemes can be used here.[25,47] One of these is given below. In this Eq. (19.114) is augmented to

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + \left(\mathbf{K} + \mathbf{Q}\mathbf{S}^{-1}\mathbf{Q}^{\mathrm{T}}\right)\mathbf{u}_{n+1} = -\mathbf{f} + \mathbf{Q}\mathbf{p}^{p}_{n+1} + \mathbf{Q}\mathbf{S}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{u}^{p}_{n+1} \tag{19.116}$$

before solving for $\Delta\ddot{\mathbf{u}}_{n+1}$. It turns out that this scheme is now unconditionally stable provided the usual conditions

$$\beta_2 \geqslant \beta_1 \qquad \beta_1 \geqslant \tfrac{1}{2}$$

are satisfied.

Such stabilization involves the inverse of $\mathbf{S}$ but again it should be noted that this needs to be obtained only for the coupling nodes on the interface. Another stable scheme involves a similar inversion of $\mathbf{H}$ and is useful as incompressible behaviour is automatically given.

Similar stabilization processes have been applied with success to the soil–fluid system.[60,61]

References

1. O.C. Zienkiewicz. Coupled problems and their numerical solution. In *Numerical Methods in Coupled Systems* (Eds R.W. Lewis, P. Bettis and E. Hinton), pp. 65–68, John Wiley and Sons, Chichester, 1984.
2. O.C. Zienkiewicz, E. Oñate, and J.C. Heinrich. A general formulation for coupled thermal flow of metals using finite elements. *Internat. J. Num. Meth. Eng.*, **17**, 1497–514, 1980.
3. B.A. Boley and J.H. Weiner. *Theory of Thermal Stresses*. John Wiley & Sons, New York, 1960. Reprinted by R.E. Krieger Publishing Co., Malabar Florida, 1985.
4. R.W. Lewis, P. Bettess, and E. Hinton, editors. *Numerical Methods in Coupled Systems*, Chichester, 1984. John Wiley & Sons.
5. R.W. Lewis, E. Hinton, P. Bettess, and B.A. Schrefler, editors. *Numerical Methods in Coupled Systems*, Chichester, 1987. John Wiley & Sons.
6. J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*, volume 7 of *Interdisciplinary Applied Mathematics*. Springer-Verlag, Berlin, 1998.
7. O.C. Zienkiewicz and R.E. Newton. Coupled vibration of a structure submerged in a compressible fluid. In *Proc. Int. Symp. on Finite Element Techniques*, pp. 1–15, Stuttgart, 1969.
8. P. Bettess and O.C. Zienkiewicz. Diffraction and refraction of surface waves using finite and infinite elements. *Internat. J. Num. Meth. Eng.*, **11**, 1271–90, 1977.
9. O.C. Zienkiewicz, D.W. Kelly, and P. Bettess. The Sommerfield (radiation) condition on infinite domains and its modelling in numerical procedures. In *Proc. IRIA 3rd Int. Symp. on Computing Methods in Applied Science and Engineering*, Versailles, December 1977.
10. O.C. Zienkiewicz, P. Bettess, and D.W. Kelly. The finite element method for determining fluid loadings on rigid structures. Two- and three-dimensional formulations. In O.C. Zienkiewicz, R.W. Lewis, and K.G. Stagg, editors, *Numerical Methods in Offshore Engineering*, pp. 141–183. John Wiley & Sons, Chichester, 1978.
11. O.C. Zienkiewicz and P. Bettess. Dynamic fluid-structure interaction. Numerical modelling of the coupled problem. In O.C. Zienkiewicz, R.W. Lewis, and K.G. Stagg, editors, *Numerical Methods in Offshore Engineering*, pp. 185–193. John Wiley & Sons, Chichester, 1978.
12. O.C. Zienkiewicz and P. Bettess. Fluid-structure dynamic interaction and wave forces. An introduction to numerical treatment. *Internat. J. Num. Meth. Eng.*, **13**, 1–16, 1978.
13. O.C. Zienkiewicz and P. Bettess. Fluid-structure dynamic interaction and some 'unified' approximation processes. In *Proc. 5th Int. Symp. on Unification of Finite Elements, Finite Differences and Calculus of Variations*, University of Connecticut, May 1980.
14. R. Ohayon. Symmetric variational formulations for harmonic vibration problems coupling primal and dual variables – applications to fluid-structure coupled systems. *La Rechereche Aerospatiale*, **3**, 69–77, 1979.

15. R. Ohayon. True symmetric formulation of free vibrations for fluid-structure interaction in bounded media. In *Numerical Methods in Coupled Systems*, Chichester, 1984. John Wiley & Sons.
16. R. Ohayon. Fluid–structure interaction. *Proc. of the ECCM'99 Conference*, *IACM/ECCM'99*, 31 August–3 September 1999, Munich, Germany.
17. H. Morand and R. Ohayon. *Fluid-Structure Interaction*, Wiley, 1995.
18. M.P. Paidoussis and P.P. Friedmann (eds). *4th International Symposium on Fluid–Structure Interactions*, *Aeroelasticity*, *Flow-Induced Vibration and Noise*, vol. 1, 2, 3, ASME/Winter Annual Meeting, 16–21 November 1997, Dallas, Texas, AD-vol. 52-3.
19. T. Kvamsdal *et al.* (eds). *Computational Methods for Fluid–Structure Interaction*, Tapir Publishers, Trondheim, 1999.
20. R. Ohayon and C.A. Felippa (eds). Computational Methods for Fluid–Structure Interaction and Coupled Problems. *Comp. Meth. in Appl. Mech. Eng.*, special issue, to appear 2000.
21. M. Geradin, G. Roberts, and J. Huck. Eigenvalue analysis and transient response of fluid structure interaction problems. *Eng. Comp.*, **1**, 152–60, 1984.
22. G. Sandberg and P. Gorensson. A symmetric finite element formation of acoustic fluid-structure interaction analysis. *J. Sound Vib.*, **123**, 507–15, 1988.
23. K.K. Gupta. On a numerical solution of the supersonic panel flutter eigenproblem. *Internat. J. Num. Meth. Eng.*, **10**, 637–45, 1976.
24. B.M. Irons. The role of part inversion in fluid-structure problems with mixed variables. *J AIAA*, **7**, 568, 1970.
25. W.J.T. Daniel. Modal methods in finite element fluid-structure eigenvalue problems. *Internat. J. Num. Meth. Eng.*, **15**, 1161–75, 1980.
26. C.A. Felippa. Symmetrization of coupled eigenproblems by eigenvector augmentation. *Commun. Appl. Num. Meth.*, **4**, 561–63, 1988.
27. J. Holbeche. *Ph. D. Thesis*. PhD thesis, University of Wales, Swansea, 1971.
28. A.K. Chopra and S. Gupta. Hydrodynamic and foundation interaction effects in earthquake response of a concrete gravity dam. *J. Struct. Div. Am. Soc. Civ. Eng.*, **578**, 1399–412, 1981.
29. J.F. Hall and A.K. Chopra. Hydrodynamic effects in the dynamic response of concrete gravity dams. *Earthquake Eng. Struct. Dyn.*, **10**, 333–95, 1982.
30. O.C. Zienkiewicz and R.L. Taylor. Coupled problems – a simple time-stepping procedure. *Comm. Appl. Num. Meth.*, **1**, 233–39, 1985.
31. O.C. Zienkiewicz, R.W. Clough, and H.B. Seed. Earthquake analysis procedures for concrete and earth dams – state of the art. Technical Report Bulletin 32, Int. Commission on Large Dams, Paris, 1986.
32. R.E. Newton. Finite element study of shock induced cavitation. In *ASCE Spring Convention*, Portland, Oregon, 1980.
33. O.C. Zienkiewicz, D.K. Paul, and E. Hinton. Cavitation in fluid-structure response (with particular reference to dams under earthquake loading). *Earthquake Eng. Struct. Dyn.*, **11**, 463–81, 1983.
34. M.A. Biot. Theory of propagation of elastic waves in a fluid saturated porous medium, Part I: low frequency range; Part II: high frequency range. *J. Acoust. Soc. Am.*, **28**, 168–91, 1956.
35. O.C. Zienkiewicz, C.T. Chang, and E. Hinton. Non-linear seismic responses and liquefaction. *Internat. J. Num. Anal. Meth. Geomech.*, **2**, 381–404, 1978.
36. O.C. Zienkiewicz and T. Shiomi. Dynamic behaviour of saturated porous media, the generalized Biot formulation and its numerical solution. *Internat. J. Num. Anal. Meth. Geomech.*, **8**, 71–96, 1984.
37. O.C. Zienkiewicz, K.H. Leung, and M. Pastor. Simple model for transient soil loading in earthquake analysis: Part I – basic model and its application. *Internat. J. Num. Anal. Meth. Geomech.*, **9**, 453–76, 1985.

38. O.C. Zienkiewicz, K.H. Leung, and M. Pastor. Simple model for transient soil loading in earthquake analysis: Part II – non-associative models for sands. *Internat. J. Num. Anal. Meth. Geomech.*, **9**, 477–98, 1985.
39. O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, and T. Shiomi. Computational approach to soil dynamics. In A.S. Czamak, editor, *Soil Dynamics and Liquefaction*, volume Developments in Geotechnial Engineering 42. Elsevier, Amsterdam, 1987.
40. O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, D.K. Paul, and T. Shiomi. Static and dynamic behaviour of soils: A rational approach to quantitative solutions, I. *Proc. Roy. Soc. London*, **429**, 285–309, 1990.
41. O.C. Zienkiewicz, Y.M. Xie, B.A. Schrefler, A. Ledesma, and N. Bicanic. Static and dynamic behaviour of soils: A rational approach to quantitative solutions, II. *Proc. Roy. Soc. London*, **429**, 311–21, 1990.
42. O.C. Zienkiewicz, A.H. Chan, M. Pastor, B.A. Schrefler and T. Shiomi. *Computational Geomechanics with Special Reference to Earthquake Engineering*, John Wiley and Sons, Chichester, 1999.
43. B.R. Simon, J. S-S. Wu, M.W. Carlton, L.E. Kazarian, E/P. France, J.H. Evans, and O.C. Zienkiewicz. Poroelastic dynamic structural models of rhesus spinal motion segments. *Spine*, **10**(6), 494–507, 1985.
44. B.R. Simon, J. S-S. Wu, and O.C. Zienkiewicz. Higher order mixed and Hermitian finite element procedures for dynamic analysis of saturated porous media. *Internat. J. Num. Meth. Eng.*, **10**, 483–99, 1986.
45. R.W. Lewis and B.A. Schrefler. *The Finite Element method in the Deformation and Consolidation of Porous Media*. John Wiley & Sons, Chichester, 1987.
46. X.K. Li, O.C. Zienkiewicz, and Y.M. Xie. A numerical model for immiscible two-phase fluid flow in porous media and its time domain solution. *Internat. J. Num. Meth. Eng.*, **30**, 1195–212, 1990.
47. O.C. Zienkiewicz and A.H.C. Chan. Coupled problems and their numerical solution. In *Advanced in Computational Non-linear Mechanics*, chapter 3, pp. 109–176. Springer-Verlag, Berlin, 1988.
48. T. Belytschko and R. Mullen. Stability of explicit-implicit time domain solution. *Internat. J. Num. Meth. Eng.*, **12**, 1575–86, 1978.
49. T.J.R. Hughes and W.K. Liu. Implicit-explicit finite elements in transient analyses. Part I and Part II. *J. Appl. Mech.*, **45**, 371–78, 1978.
50. T. Belytschko and T.J.R. Hughes, editors. *Computational Methods for Transient Analysis*. North-Holland, Amsterdam, 1983.
51. C.A. Felippa and K.C. Park. Staggered transient analysis procedures for coupled mechanical systems: formulation. *Comp. Meth. Appl. Mech. Eng.*, **24**, 61–111, 1980.
52. K.C. Park. Partitioned transient analysis procedures for coupled field problems: stability analysis. *J. Appl. Mech.*, **47**, 370–76, 1980.
53. K.C. Park and C.A. Felippa. Partitioned transient analysis procedures for coupled field problems: accuracy analysis. *J. Appl. Mech.*, **47**, 919–26, 1980.
54. O.C. Zienkiewicz, E. Hinton, K.H. Leung, and R.L. Taylor. Staggered time marching schemes in dynamic soil analysis and selective explicit extrapolation algorithms. In R. Shaw *et al.*, editors, *Proc. Conf. on Innovative Numerical Analysis for the Engineering Sciences*, University of Virginia Press, 1980.
55. O.C. Zienkiewicz, C.T. Chang, and P. Bettess. Drained, undrained, consolidating dynamic behaviour assumptions in soils. *Geotechnique*, **30**, 385–95, 1980.
56. D.M. Trujillo. An unconditionally stable explicit scheme of structural dynamics. *Internat. J. Num. Meth. Eng.*, **11**, 1579–92, 1977.
57. L.J. Hayes. Implementation of finite element alternating-direction methods on non-rectangular regions. *Internat. J. Num. Meth. Eng.*, **16**, 35–49, 1980.

58. A.W. Craig and O.C. Zienkiewicz. A multigrid algorithm using a hierarchical finite element basis. In D.J. Pedolon and H. Holstein, editors, *Multigrid Methods in Integral and Differential Equations*, pp. 310–312. Clarendon Press, Oxford, 1985.
59. I. Babuška, A.W. Craig, J. Mandel, and J. Pitkäranta. Efficient preconditioning for the *p*-inversion finite element method in two dimensions. *SIAM J. Num. Anal*, **28**, 624–61, 1991.
60. K.C. Park. Stabilization of partitioned solution procedures for pore fluid-soil interaction analysis. *Internat. J. Num. Meth. Eng.*, **19**, 1669–73, 1983.
61. O.C. Zienkiewicz, D.K. Paul, and A.H.C. Chan. Unconditionally stable staggered solution procedures for soil-pore fluid interaction problems. *Internat. J. Num. Meth. Eng.*, **26**, 1039–55, 1988.

20

Computer procedures for finite element analysis

20.1 Introduction

In this chapter we consider some of the steps that are involved in the development of a finite element computer program to carry out analyses for the theory presented in previous chapters. The computer program discussed here may be used to solve any one-, two-, or three-dimensional linear steady-state or transient problem. The program may also be used to solve non-linear problems as will be discussed in Volume 2.

Source listings are not included in the book but may be obtained at no charge from the publisher's internet web site (http://www.bh.com/companions/fem). Any errors reported by readers will be corrected frequently so that up-to-date versions will be available.

The program is an extension of the work presented in the 4th edition.[1,2] The version discussed here is called *FEAPpv* to distinguish the current program from that presented earlier. The program name is an acronym for Finite Element Analysis Program – personal version. It is intended mainly for use in learning finite element programming methodologies and in solving small to moderate size problems on single processor computers. A simple memory management scheme is employed to permit efficient use of main memory with limited need to read and write information to disk.

The current version of *FEAPpv* permits both 'batch' and 'interactive' problem solution. The finite element model of the problem is given as an input file and may be prepared using any text editor capable of writing ASCII file output. A simple graphics capability is also included to display the mesh and results from one- and two-dimensional models in either their undeformed or reference configuration. The available versions for graphics is limited to X-window applications and compilers compatible with the current Compac Fortran 95 compiler for Windows based systems. Experienced programmers should be able to easily adapt the routines to other systems.

Finite element programs can be separated into three basic parts:

1. data input module and preprocessor
2. solution module
3. results module

Figure 20.1 shows a simplified schematic for a typical finite element program system. Each of the modules can in practice be very complex. In the subsequent

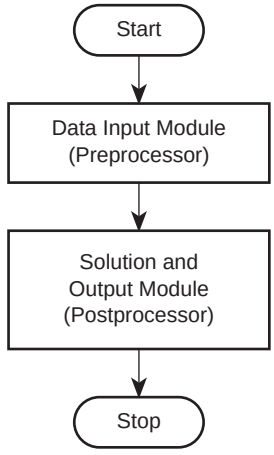


Fig. 20.1 Simplified schematic of finite element program.

sections we shall discuss in some detail the programming aspects for each of the modules. It is assumed that the reader is familiar with the finite element principles presented in this book, linear algebra, and programming in either Fortran or C. Readers who merely intend to use the program may find information in this chapter useful for understanding the solution process; however, for this purpose it is only necessary to read the user instructions available from the web site where the program is downloaded.

This chapter is divided into seven sections. Section 20.2 describes the procedure adopted for data input, necessary to define a finite element problem and basic instructions for data file preparation. The data to be provided consists of nodal quantities (e.g., coordinates, boundary condition data, loading, etc.) and element quantities (e.g., connection data, material properties, etc.).

Section 20.3 describes the memory management routines.

Section 20.4 discusses solution algorithms for various classes of finite element analyses. In order to have a computer program that can solve many types of finite element problems a *command language* strategy is adopted. The command language is associated with a set of compact subprograms, each designed to compute one or at most a few basic steps in a finite element process. Examples in the language are commands to form a global stiffness matrix, as well as commands to solve equations, display results, enter graphics mode, etc. The command language concept permits inclusion of a wide range of solution algorithms useful in solving steady-state and transient problems in either a linear or non-linear mode.

In Section 20.5 we discuss a methodology commonly used to develop element arrays. In particular, numerical integration is used to derive element 'stiffness', 'mass' and 'residual' (load) arrays for problems in linear heat transfer and elasticity. The concept of using basic shape function routines is exploited in these developments (Chapters 8 and 9).

In Section 20.6 we summarize methods for solving the large set of linear algebraic equations resulting from the finite element formulation. The methods may be divided into direct and iterative categories. In a direct solution a variant of Gaussian

elimination is used to factor the problem coefficient matrix (e.g., stiffness matrix) into the product of a lower triangular, diagonal and upper triangular form. A solution (or indeed subsequent resolutions) may then be easily obtained. A direct solution has the advantage that an *a priori* calculation may be made on the number of numerical operations which need to be performed to obtain a solution. On the other hand, a direct solution results in fill-in of the initial, sparse finite element coefficient array – this is especially significant in three-dimensional solutions and results in very large storage and compute times. In the second category iterative strategies are used to systematically reduce a residual equation to zero, and thus yield an acceptable solution to the set of linear algebraic equations. The scheme discussed in this chapter is limited to solution of symmetric equations by a preconditioned conjugate gradient method.

20.2 Data input module

The data input module shown in Fig. 20.1 must obtain sufficient information to permit the definition and solution of each problem by the other modules. In the program discussed in this book the data input module is used to read the necessary geometric, material, and loading data from a file or from information specified by the user using the computer keyboard or mouse. In the program a set of dynamically dimensioned arrays is established which store nodal coordinates, element connection lists, material properties, boundary condition indicators, prescribed nodal forces and displacements, etc. Table 20.1 lists the names of variables which are used in assigning array sizes for mesh data and Table 20.2 indicates some of the main arrays used to store mesh data.

Table 20.1 Control parameters

Variable name	Description	Default
NUMNP	Number of nodal points in mesh	0
NUMEL	Number of elements in mesh	0
NUMMAT	Number of material sets in mesh	0
NDM	Spatial dimension of mesh	none
NDF	Number of degrees of freedom per node (maximum)	none
NEN	Number of nodes per element (maximum)	none
NDD	Number of material property values per set	200

Table 20.2 Variable names used for data storage

Variable name (dimension)	Type	Description
ID(NDF,NUMNP,2)	Integer	(1) Boundary codes; (2) Equation numbers
IE(NIE,NUMMAT)	Integer	Element pointers for degrees of freedom, history pointers, material set type, etc.
IX(NEN1,NUMEL)	Integer	Element connections, set flag, etc.
D(NDD,NUMMAT)	Real	Material property data sets
F(NDF,NUMNP,2)	Real	(1) Nodal forces; (2) and displacements
X(NDM,NUMNP)	Real	Nodal coordinates

The notation used for the arrays often differs from that used in the text. For example, in the text it was found convenient to refer to nodal coordinates as x_i, y_i, z_i, whereas in the program these are called `X(1,i)`, `X(2,i)`, `X(3,i)`, respectively. This change is made so that all arrays used in the program can be dynamically allocated. Thus, if a two-dimensional problem is analysed, space will not be reserved for the `X(3,i)` coordinates. Similarly the nodal displacements in the text were commonly named a_i; in the program these are called `U(1,i)`, `U(2,i)`, etc., where the first subscript refers to the degrees of freedom at a node (from `1` to `NDF`).

20.2.1 Control data and storage allocation

The allocation of the major arrays for storage of mesh and solution variables is performed in a control program as indicated in Fig. 20.2. Since a dynamic memory allocation is used it is not possible to establish absolute values for the maximum number of nodes, elements or material sets. The value for the parameter `NUM_MR` defines the amount of memory available to solve a given problem and is assigned to the main program module; however, if this is not sufficient an error message is given and the program stops execution.

To facilitate the allocation of all the arrays data defining the size of the problem is input by the control program as shown schematically in Fig. 20.2. The required data is shown in Table 20.1; however, the number of nodes, elements and material sets may be omitted and `FEAPpv.f` will use the subsequent input data to determine the actual size required. Using the size data the remaining mesh storage requirements are determined and allocated by the control program.

20.2.2 Element and coordinate data

After a user has determined the mesh layout for a problem solution the data must be transmitted to the analysis program. As an example consider the specification of the nodal coordinate and element connection data for the simple two-dimensional (`NDM = 2`) rectangular region shown in Fig. 20.3, where a mesh of nine four-node rectangular elements (`NUMEL = 9` and `NEN =4`) and 16 nodes (`NUMNP = 16`) has been indicated. To describe the nodal and element data, values must be assigned to each `X(i,j)` for $i = 1, 2$ and $j = 1$ to 16 and to each `IX(k,n)` for $k = 1$ to 4 and $n = 1$ to 9. In the definition of the coordinate array `X`, the subscript i indicates the coordinate direction and the subscript j the node number. Thus, the value of `X(1,3)` is the x coordinate for node 3 and the value of `X(2,3)` is the y coordinate for node 3. Similarly for the element connection array `IX` the subscript k is the local node number of the element and n is the element number. The value of any `IX(k,n)` (for k less than or equal to `NEN`) is the number of a global node. Values of k larger than `NEN` are used to store other data. The convention for the first local node number is somewhat arbitrary. The local node number 1 for element 3 in Fig. 20.3 could be associated with global node 3, 4, 7, or 8. Once the first local node is established the others must follow according to the convention adopted for each particular element type. For example,

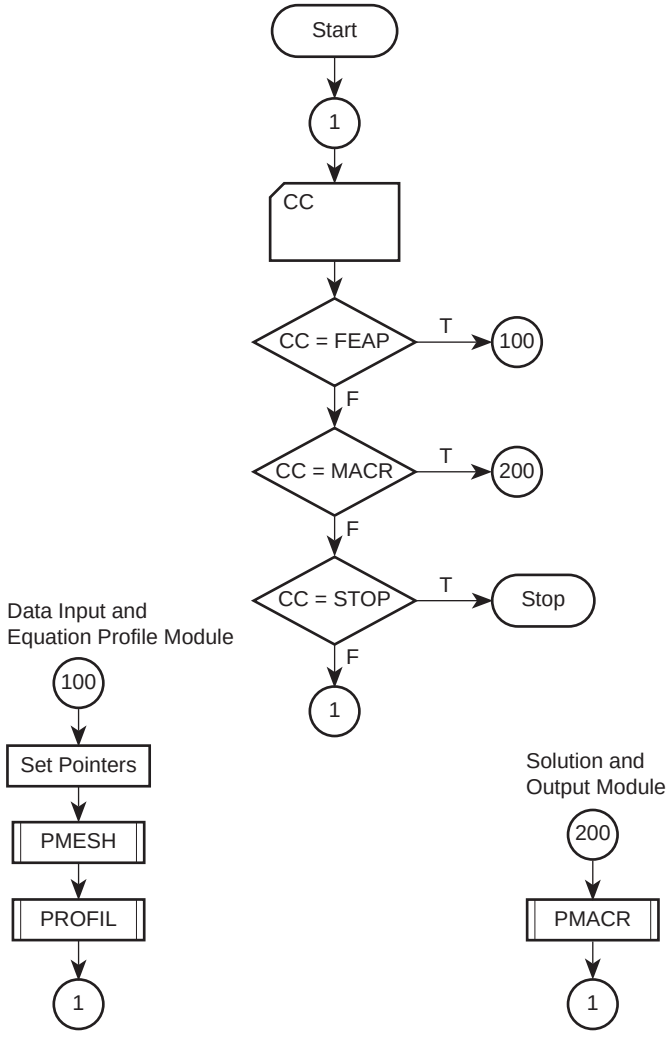


Fig. 20.2 Control program flow chart.

it is conventional to number the connections by a *right-hand rule* and the four-noded quadrilateral element can be numbered according to that shown in Fig. 20.4. If we consider once again element 3 from the mesh in Fig. 20.3 we have four possibilities for specifying the `IX(k,3)` array as shown in Fig. 20.4. The computation of the element arrays from any of the above descriptions must produce the same coefficients for the global arrays and is known as *element invariance* to data input.

Data input modules

In *FEAPpv* two subprograms `PINPUT` and `TINPUT` are available to perform data input operations. For example, all the nodal coordinates may be input using the subprogram

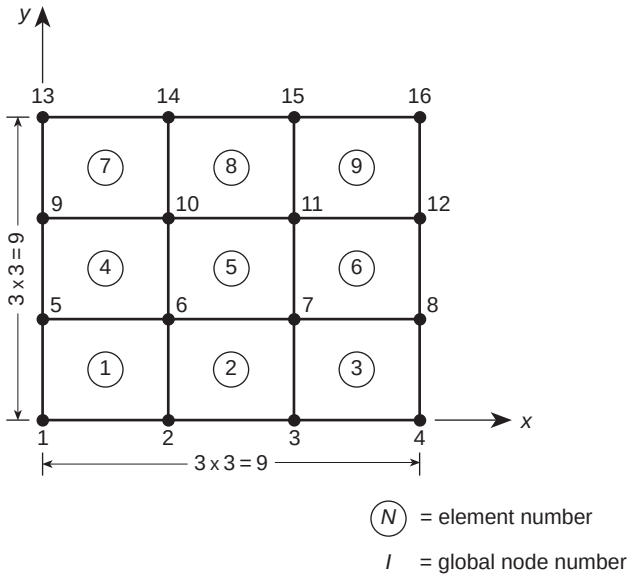
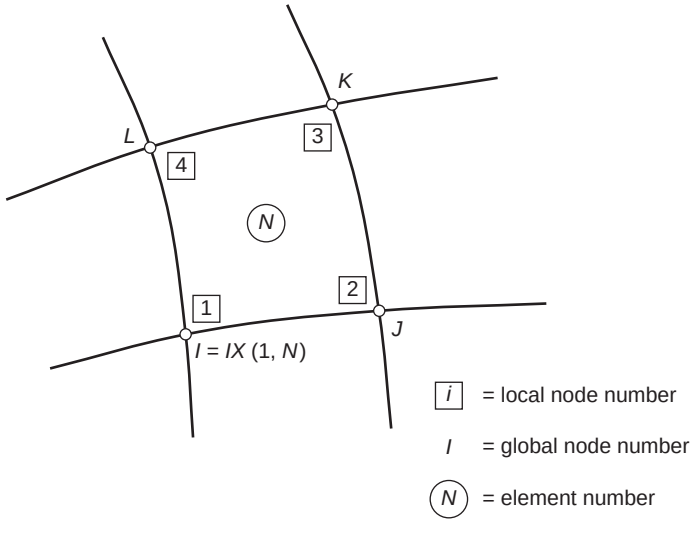


Fig. 20.3 Simple two-dimensional mesh.



Option number	Local node number			
	1	2	3	4
a	3	4	8	7
b	4	8	7	3
c	8	7	3	4
d	7	3	4	8

Fig. 20.4 Typical four-noded element and numbering options.

```
SUBROUTINE XDATA(X,NDM,NUMNP)

IMPLICIT NONE

LOGICAL  ERRCK, PINPUT
INTEGER  NUMNP, NDM , N
REAL*8   X(NDM,NUMNP)

DO N = 1,NUMNP
  ERRCK = PINPUT(X(1,N),NDM}
  IF(ERRCK) THEN
    STOP ‘ Coordinate error: Node:’,N
  ENDIF
END DO ! N

END
```

The above use of the PINPUT routine obtains NDM values from each record and assigns them to the coordinate components of node N. The data input routines obtain their information from the current input file specified by a user. The routines are also used in cases where input is to be provided from the keyboard. These input all data in character mode, and parse the data for embedded function names or parameters (use of functions and parameters is described in the user manual). Users who are extending the capability of the program are encouraged to use the routines to avoid possible errors. The subprogram TINPUT permits character data to precede numerical values use is given as

```
ERRCK = TINPUT(TEXT,M,DATA,N)
```

in which TEXT is a CHARACTER*15 array of size M and DATA is a REAL*8 array of size N.

For cases where integer information is to be input the information must be moved. For example, a simple input routine for the IX data is

```
SUBROUTINE IXDATA(IX,NEN1,NUMEL)

IMPLICIT   NONE

LOGICAL    ERRCK, PINPUT
INTEGER    NUMEL, NEN1 , N, I
INTEGER    IX(NEN1,NUMEL)
REAL*8     RIX(16)

DO N = 1,NUMEL
  ERRCK = PINPUT(RIX,NEN1}
  IF(ERRCK) THEN             ! Stop on error
    STOP ‘ Connection error: ELEMENT:’,N
  ELSE                       ! Move data to IX
    DO I = 1,NEN1
      IX(I,N) = NINT(RIX(I))
    END DO ! I
  ENDIF
END DO ! N

END
```

While the above form is not optimal it is an expedient method to permit the arbitrary mixing of real and integer data on the same record. In the above two examples the node and element numbers are associated with the record number read. The form used in the routines supplied with *FEAPpv* include the node and element numbers as part of the data record. In this form the inputs need not be sequential nor all data input at one instance.

For a very large problem the preparation of each node and element record for the mesh data would be very tedious; consequently, some methods are provided in *FEAPpv* to *generate* missing data. These include simple interpolation between missing numbers of nodes or elements, use of super-elements to perform generation of *blocks* of nodes and elements, and use of blending function methods. Even with these aids the preparation of the mesh data for nodes and coordinates can be time consuming and users should consider the use of mesh generation programs such as GiD[3] to assist in this task. Generally, however, the data input scheme included in the program is sufficient to solve academic and test examples. Moreover the organization of the mesh input module (subprogram `PMESH`) is data driven and permits users to interface their own program directly if desired (see below for more information on adding features). The data-driven format of the mesh input routine is controlled by keywords which direct the program to the specific segment of code to be used. In this form each input segment does not interact with any of the others as shown schematically in the flow chart in Fig. 20.5.

20.2.3 Material property specification – multiple element routines

The above discussion considered the data arrays for nodal coordinates and element connections. It is also necessary to specify the material properties associated with each element, loadings, and the restraints to be applied to each node.

Each element has associated property sets, for example in linear isotropic elastic materials Young's modulus E and Poisson's ratio ν describe the material parameters for an isotropic state. In most situations several elements have the same property sets and it is unnecessary to specify properties for each element individually. In the data structure used in *FEAPpv* an element is associated with a material set by a number on the data record for each element. The material properties are then given once for each number. For example, if the region shown in Fig. 20.3 is all the same material, only one material set is required and each element would reference this set. To accommodate the storage of the material set numbers the `IX` array is increased in size to `NEN1` entries and the material set number is stored in the entry `IX(NEN1,n)` for element n. In *FEAPpv* the material properties are stored in the array `D(NDD,NUMMAT)`, where `NUMMAT` is the number of different material sets and `NDD` is the number of allowable properties for each material set (the default for `NDD` is 200).

Each material set defines the element type to which the properties are to be assigned. In realistic engineering problems several element types may be needed to define the problem to be solved. A simple example involving different element types is shown in

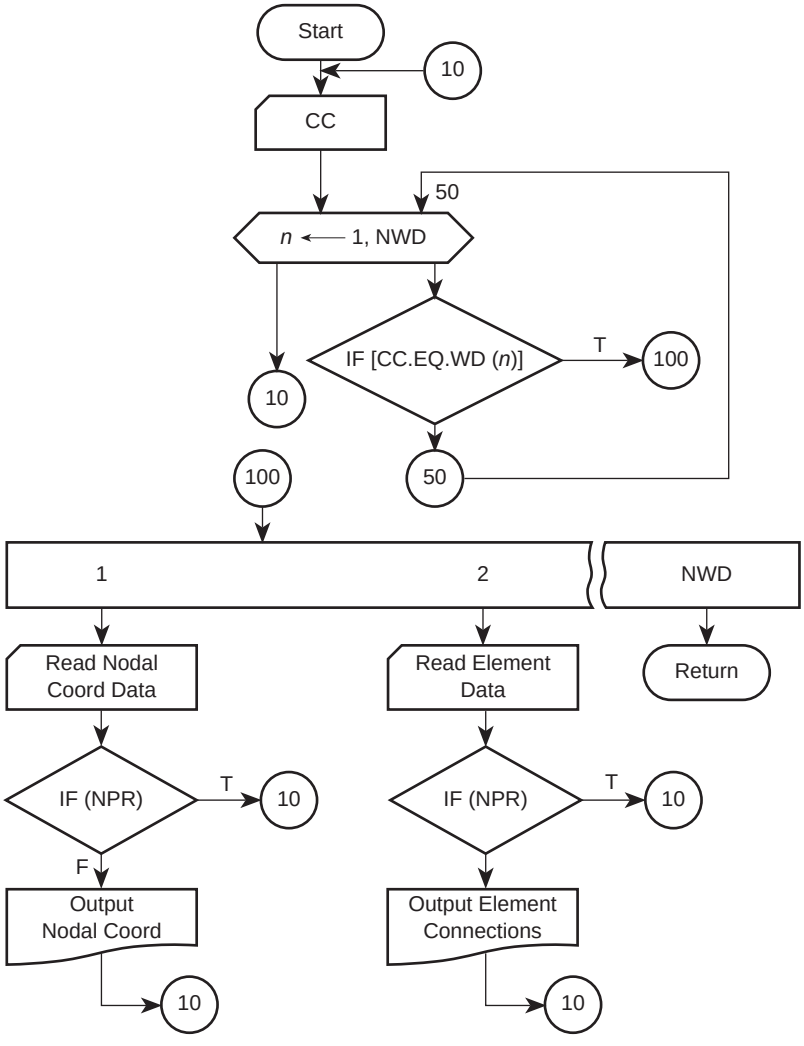


Fig. 20.5 Flow chart for mesh data input.

Fig. 1.4(a) in Chapter 1 where elements 1, 2, 4, and 5 are plane stress elastic elements and element 3 is a truss element. In this case at least two different types of element stiffness formulations must be computed. In *FEAPpv* it is possible to use ten different user provided element formulations in any analysis.† The program has been designed so that all computations associated with each individual element are performed in one element subprogram called `ELMTnn`, where `nn` is between `01` and `10` (see Sec. 20.5.3 for a discussion on the organization of `ELMTnn`). Each element type to be used is specified as part of the material set data. Thus if element type 1, e.g., computations performed by `ELMT01`, is a plane linear elastic three- or four-noded element and element type 4 is a truss element, the data given for example Fig. 1.4(a) would be:

† In addition, some standard element formulations are provided as described in the user instructions.

(a) *Material properties*

Material set number	Element type	Material property data
1	4	E_1, A_1
2	1	E_2, ν_2

(b) *Element connections*

Element	Material set	Connection
1	2	1 3 4
2	2	1 4 2
3	1	2 5
4	2	3 6 7 4
5	2	4 7 8 5

where E is Young's modulus, ν is Poisson's ratio and A is area. Thus, elements 1, 2, 4, and 5 have material property set 2 which is associated with element type 1 and element 3 has a material property set 1 which is associated with element type 4. It will be seen later that the above scheme leads to a simple organization of an element routine which can input material property sets and perform all the necessary computations for the finite element arrays.

More sophisticated schemes could be adopted; however, for the educational and research type of program described here this added complexity is not warranted.

20.2.4 Boundary conditions – equation numbers

The process of specifying the boundary conditions at nodes and the procedure for imposing specified nodal displacements is closely associated with the method adopted to store the global solution matrix, e.g., the stiffness matrix. In *FEAPpv* the direct solution procedure included uses a variable band (profile) storage for the global solution matrix. Accordingly, only those coefficients within the non-zero profiles are stored.

While the nodal displacements associated with boundary restraints may be imposed using the 'penalty' method described in Chapter 1, a more efficient direct solution results if the rows and columns for these equations are deleted. As an example consider the stiffness matrix corresponding to the problem shown in Fig. 1.1; storing all terms within the upper profile leads to the result shown in Fig. 20.6(a) and requires 54 words, whereas if the equations corresponding to the restrained nodes 1 and 6 are deleted the profile shown in Fig. 20.6(b) results and requires only 32 words. In addition to a reduction in storage requirements, the computer time to solve the equations is also reduced.

To facilitate a compact storage operation in forming the global arrays, a boundary condition array is used for each node. The array is named `ID` and is dimensioned as shown in Table 20.2. During input of data, degrees of freedom with known value or where no unknown exists have a non-zero value assigned to `ID(i,j,1)`. All active

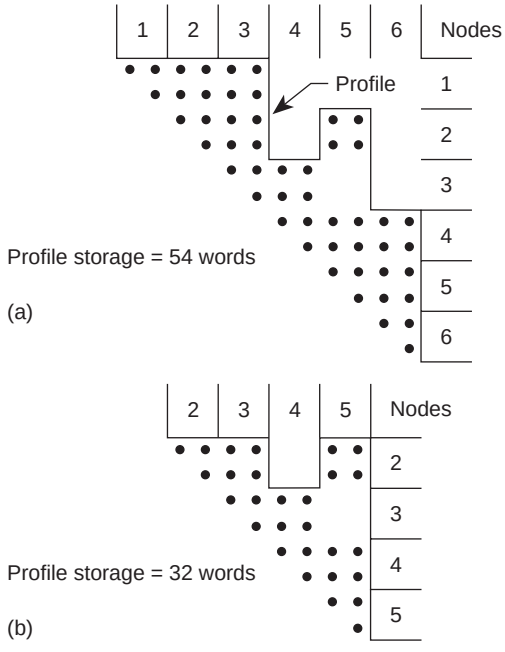


Fig. 20.6 Stiffness matrix: (a) total stiffness storage; (b) storage after deletion of boundary conditions.

degrees of freedom have a zero value in the `ID` array. After the input phase the values in `ID(i,j,2)` are assigned values of the active equation numbers. Restrained DOFs have zero (or negative) values.

Table 20.3 shows the `ID` values for the example shown in Fig. 1.1(a), where it is evident that nodes 1 and 6 are fully restrained.

The numbers for the equations associated with unknowns are constructed from Table 20.3 by replacing each non-zero value with a zero and each zero value by the appropriate equation number. In *FEAPpv* this is performed by subprogram `PROFIL` starting with the degrees of freedom associated with node 1 followed by node 2, etc. The result for the example leads to values shown in Table 20.4, and this information is stored in `ID(i,j,2)`. This information is used to assemble all the global arrays.

Table 20.3 Boundary restraint code values after data input of problem in Fig. 1.1

	Degree of freedom	
Node	1	2
1	1	1
2	0	0
3	0	0
4	0	0
5	0	0
6	1	1

Table 20.4 Compacted equation numbers for problem in Fig. 1.1

	Degree of freedom	
Node	1	2
1	0	0
2	1	2
3	3	4
4	5	6
5	7	8
6	0	0

The above scheme may be modified in a number of ways for either efficiency or to accommodate more general problems. For problems in which the node numbers are input in an order which creates a very large profile it is advisable to employ a program to renumber the nodes for better efficiency (often called bandwidth minimization schemes). Using the renumbered node order the equation numbers may then be constructed.

The solution of mixed formulations which have matrices with zero diagonals requires special care in solving for the parameters. For example in the $\mathbf{q}$, formulation discussed in Sec. 11.2 it is necessary to eliminate all $\tilde{\mathbf{q}}_i$ parameters associated with each $\tilde{\phi}_i$ parameter when a direct method of solution without pivoting is used (e.g., those discussed in Sec. 20.6.1). This may be achieved by numbering the `ID(i,j,2)` entries so that $\tilde{\mathbf{q}}_i$ have smaller equation numbers than the one for the associated $\tilde{\phi}_i$.

The equation number scheme may be further exploited to handle repeating boundaries (see Chapter 9, Sec. 9.18) where nodes on two boundaries are required to have the same displacement but the value is unknown. This is accomplished by setting the equation numbers to the same value (and discard the unused ones). Similarly, regions may be joined by assigning nodes with the same coordinate values the same equation numbers.

All modifications of the above type must be performed prior to computing the profile of the global matrix.

20.2.5 Loading – nodal forces and displacements

In *FEAPpv* the specified nodal forces and displacements associated with each degree of freedom are stored in the array `F(NDF,NUMNP,2)`. The specified force values for degree of freedom *i* at node *j* are retained in `F(i,j,1)` and specified values for the corresponding specified displacements in `F(i,j,2)`. The actual value to be used during each phase of an analysis depends on the current value stored in `ID(i,j,1)`. Thus if the value of the `ID(i,j,1)` is zero a force value is taken from `F(i,j,1)` whereas if the value is non-zero a displacement value is taken from `F(i,j,2)`. For the example of Fig. 1.1, an 0.01 settlement of the node 1 can be input by setting `F(1,2,2) = -0.01`, where it is assumed that the second degree of

freedom is a displacement in the vertical direction. Similarly, a horizontal force at node 4 can be specified by setting `F(1,4,1) = 5`, (i.e., X_4 in the figure).

In many problems the loading may be distributed and in these cases the loading must first be converted to nodal forces. In *FEAPpv* there are some provisions included to perform the computation automatically. Users may develop additional schemes for their own problems and add a new input command in the subprogram `PMESH`. Other options could also be added to compute necessary nodal quantities.

The necessary steps to add a feature in `PMESH` are:

1. Increase the dimensioned size of the array `WD` which is a character array to store the command names.
2. Set the value of `LIST` in the `DATA` statement to the new number of entries in `WD`.
3. Add a new statement label entries to the `GO TO` statement.
4. For each statement label entry add the program statements for the new feature.

The specific instructions to prepare data for *FEAPpv* are contained in the user manual available at the publisher's web site.

20.2.6 Mesh data checking

Once all the data for the geometric, material and loading conditions are supplied *FEAPpv* is ready to initiate execution of the solution module; however prior to this step it is usually preferable to perform some checks on the input data (and any generated values).

After the mesh is input the program will pass to solution mode. During solution additional arrays may be required which can also exceed the available space in the blank common. The most intensive storage requirement is for the global coefficient matrix for the set of linear algebraic equations defining the nodal solution parameters. In direct solution mode a variable band, profile solution scheme is used for simplicity. The solver has the capability of solving both symmetric and unsymmetric coefficient arrays and this is generally adequate for one- and two-dimensional problems of moderate size. However, for three-dimensional applications the storage demands for the coefficient matrix can exceed the capabilities of even the largest computers available at the time of writing this volume. Thus, an alternative iterative scheme is included in *FEAPpv* using a simple preconditioned conjugate gradient solver.

20.3 Memory management for array storage

A single array is partitioned to store all the main data arrays, as well as other arrays needed during the solution and output phases. This is accomplished using a data management system which can define, resize or destroy an integer or real array. Depending on the computer system used real arrays may be defined in the main program module `FEAPpv.F` in either single precision or double precision form. Using the data management system each array indicated in Table 20.2 is dynamically dimensioned to the size and precision required for each problem. The result is a set of pointers defining

the location in a single array located in blank common. Blank common is defined as

```
REAL*8   HR
INTEGER        MR
COMMON   HR(1),MR(NUM_MR)
```

and pointers are assigned into the array NP stored in the named common POINTERS given by

```
INTEGER            NP
COMMON /POINTERS/ NP(NUM_NP)
```

The size of each array is defined by parameters NUM_MR and NUM_NP. While not strictly defined by programming standards the above size for HR is not limited to 1. By working outside the array bound real arrays may be defined up to size NUM_MR/2 for the double precision indicated. Using this artifice of pointers subroutines may be called as

```
CALL SUBX(MR(NP(5)), HR(NP(33)), ... )
```

where the first argument is integer and the second real. The subroutine would then read

```
SUBROUTINE SUBX(I1, R1, .... )
```

and real names associated with each array as determined by a programmer. At this stage the missing ingredient is assignment of values to each specific pointer. In *FEAPpv* this is accomplished by the subprogram PALLOC. This logical function subprogram associates a number with a name for each variable to be defined, changed or deleted. Each programmer must use a listing of this routine to understand which variable is being defined and whether the variable is to be real or integer. A specification of an array action is accomplished using the assignment statement

```
SETVAL = PALLOC{ NUM , NAME , LENGTH , PRECISION )
```

For example the statement

```
SETVAL = PALLOC{ 43 , 'X' , NDM*NUMNP , 2 )
```

defines the real array for the nodal coordinates to have a size as indicated in Table 20.2. Similarly, the statement

```
SETVAL = PALLOC{ 33 , 'IX' , NEN1*NUMEL , 1 )
```

defines an integer array for the element connection array. Repeating the use of the allocation statement with a different size (either larger or smaller) will redefine the size of the array. Similarly, use of the statement with a zero (0) size deletes the array from the allocation table. Accordingly, use of

```
SETVAL = PALLOC{ 33 , 'IX' , 0 , 1 )
```

would destroy the storage (and values) for the connection data. Thus, using the memory management scheme above it is possible to redefine a mesh in an adaptive solution scheme to add or delete specific element data. Alternatively, data may be used in a temporary manner by allocating and then deleting after use.

20.4 Solution module – the command programming language

At the completion of data input and any checks on the mesh we are prepared to initiate a problem solution. It is at this stage that the particular type of solution mode must be available to the user. In many existing programs only a small number of solution modes are generally included. For example, the program may only be able to solve linear steady-state problems, or in addition it may be able to solve linear transient problems for a single method. In a research mode or indeed in practical engineering problems fixed algorithm programs are often too restrictive and continual modification of the program is necessary to solve specific problems that arise – often at the expense of features needed by another user. For this reason it is desirable to have a program that has modules for various algorithm capabilities and, if necessary, can be modified without affecting other users' capabilities. The program form that we discuss here is basic and the reader can undoubtedly find many ways to improve and extend the capabilities to be able to solve other classes of problems.

The command language concept described in this section has been used by the authors for more than 20 years and, to date, has not inhibited our research activities by becoming outdated. Applications are routinely conducted on personal computers and workstations using an identical program except for graphical display modules.

20.4.1 Linear steady-state problems

A basic aspect of the variable algorithm program *FEAPpv* is a command instruction language which is associated with specific program solution modules for specific algorithms as needed. A user needs only to understand the association between specific commands and the operations carried out by the associated solution modules.

In a steady-state problem we are required to solve the problem given, for example, by

$$\mathbf{r}^{(k)} = \mathbf{f} - \mathbf{K}\mathbf{a}^{(k)} \tag{20.1}$$

where k is an index related to the solution iteration number. We call $\mathbf{r}^{(k)}$ the *residual* of the problem for iteration k and note that a solution results when it is zero. In a data-driven solution mode using the command language of *FEAPpv* the formulation of Eq. (20.1) is given by the command `FORM`, which is a mnemonic for *form residual*. In addition an incremental form of the solution of Eq. (20.1) is adopted in *FEAPpv*. Accordingly we let

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} + \Delta\mathbf{a}^{(k)} \tag{20.2}$$

and solve the problem

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \mathbf{K}\Delta\mathbf{a}^{(k)} = \mathbf{0} \tag{20.3}$$

Since the problem given by Eq. (20.1) is linear this iterative form *must* converge in one iteration. That is, if we solve the problem for $k = 0$ for any specified $\mathbf{a}^{(0)}$, the residual for $k = 1$ will be zero (to machine precision). The only exceptions to this will be: (*a*) an improperly formulated or implemented finite element formulation for the stiffness and/or the residual; (*b*) an incorrect setting of the necessary boundary conditions to avoid singularity of the resulting stiffness matrix; or (*c*) the problem is so ill-posed that round-off in computer arithmetic leads to significant error in the resulting solution.

In *FEAPpv* the command language statement to form a symmetric stiffness matrix is `TANG`, which is a mnemonic for *tangent stiffness*. An unsymmetric stiffness matrix can be formed by specifying the command `UTAN`. By now the reader should have observed that commands for *FEAPpv* are given by four-character mnemonics. In general, users can use up to 14 characters to issue any command, however, only the first four are interpreted by the program. Thus, if a user desires, the command to form the tangent may be given as `TANGENT`. Finally, to solve the systems of equations given by Eq. (20.3) the command `SOLV` is used. Thus to solve a steady-state problem the three commands issued are:

```
TANGent
FORM
SOLVe
```

The first two commands can be reversed without affecting the algorithm.

The basic structure for all command language statements is:

```
COMMAND OPTION VALUE_1 VALUE_2 VALUE_3
```

Since the above three statements occur so often in any finite element solution strategy a shorthand command option is provided in *FEAPpv* as

```
TANGent,,1
```

where a comma is used to separate the fields and leave a blank option parameter. Any positive non-zero number may be used for the `VALUE_1` parameter.

A user can check that the solution is correct by including another `FORM` command after the `SOLV` statement.

After a solution has been performed for the steady-state problem it is necessary to issue additional commands in order to obtain the solution results. For example, the commands

```
DISPlacement ALL
STREss ALL
```

will output all the nodal displacements and stresses in an *output file* specified at the initiation of running *FEAPpv*. Table 20.5 lists some of the commands available in the program. A complete list is available in the user manual.

The variable algorithm program described by a command language program can often be extended as necessary without need to reprogram the modules. Additional options are described in the user manual.

Table 20.5 Partial list of solutions commands

Command	Option	Value_1	Value_2	Value_3	Description
CHECk					Perform check of mesh (ISW = 2)[1]
DISP	ALL	N1	N2	N3	Output displacement for nodes N1 to N2 at increments of N3 ALL outputs all
DT		V1			Set time increment to V1
FORM					Form equation residual (ISW = 6)
LOOP		N			Loop N times all instructions to a matching NEXT command
MESH					Input changes to mesh
NEXT					End of LOOP instruction
PLOT	OPTION				Enter graphical mode or perform command OPTION
REAC	ALL	N1	N2	N3	Output reactions at nodes N1 to N2 at increments of N3 ALL outputs all (ISW = 6)
SOLV					Solve for new solution increment (after FORM)
STRE	ALL	N1	N2	N3	Output element variables N1 to N2 at increments of N3 ALL outputs all (ISW = 4)
TANGent		N1			Form symmetric tangent Solve if N1 positive (ISW = 3)
TIME					Advance time by DT value
TOL		V1			Set solution tolerance to V1
UTAN		N1			Form unsymmetric tangent (ISW = 3)

20.4.2 Transient solution methods

The integration of second-order differential equations of motion for time-dependent structural systems can be treated using the command language program. The first-order differential equations resulting from the heat equation may also be similarly integrated. For the transient second-order case the residual equation is modified to

$$\mathbf{r}^{(k)} = \mathbf{f} - \mathbf{K}\mathbf{a}^{(k)} - \mathbf{C}\dot{\mathbf{a}}^{(k)} - \mathbf{M}\ddot{\mathbf{a}}^{(k)} \tag{20.4}$$

where $\mathbf{C}$ and $\mathbf{M}$ are damping and mass matrices, respectively, and $\dot{\mathbf{a}}$ and $\ddot{\mathbf{a}}$ are velocity and acceleration, respectively. To solve this problem it is necessary to:

1. specify the time integration method to be used (see Chapter 18);
2. specify the time increment for the integration;
3. specify the number of time steps to perform;
4. form the residual $\mathbf{r}^{(k)}$;
5. form the tangent matrix for the specific time integration method;
6. solve the equation for each time step;
7. report answers as needed.

As an example we consider the Newmark method (GN22) as described in Chapter 18, Sec. 18.33. Using Eq. (18.12) we can define the updates at iteration k as

$$\mathbf{a}_{n+1}^{(k)} = \bar{\mathbf{a}}_{n+1} + \tfrac{1}{2}\beta_2 \Delta t^2 \ddot{\mathbf{a}}_{n+1}^{(k)} \tag{20.5}$$

$$\dot{\mathbf{a}}_{n+1}^{(k)} = \dot{\bar{\mathbf{a}}}_{n+1} + \beta_1 \Delta t \ddot{\mathbf{a}}_{n+1}^{(k)} \tag{20.6}$$

where $\bar{\mathbf{a}}_{n+1}$ and $\dot{\bar{\mathbf{a}}}_{n+1}$ are expressed in terms of solution variables at time n. These equations may also be written in an incremental form as

$$\mathbf{a}_{n+1}^{(k+1)} = \mathbf{a}_{n+1}^{(k)} + \tfrac{1}{2}\beta_2 \Delta t^2 \Delta\ddot{\mathbf{a}}_{n+1}^{(k)} \tag{20.7}$$

$$\dot{\mathbf{a}}_{n+1}^{(k+1)} = \dot{\mathbf{a}}_{n+1}^{(k)} + \beta_1 \Delta t \Delta\ddot{\mathbf{a}}_{n+1}^{(k)} \tag{20.8}$$

Comparing Eq. (20.7) with Eq. (20.3) we obtain

$$\Delta\mathbf{a}_{n+1}^{(k)} = \tfrac{1}{2}\beta_2 \Delta t^2 \Delta\ddot{\mathbf{a}}_{n+1}^{(k)} \tag{20.9}$$

Similarly

$$\Delta\dot{\mathbf{a}}_{n+1}^{(k)} = \beta_1 \Delta t \Delta\ddot{\mathbf{a}}_{n+1}^{(k)} \tag{20.10}$$

Thus, selecting the incremental nodal displacements as the primary unknown, the residual equation for $k+1$ may be written as

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \mathbf{K}^* \Delta\mathbf{a}_{n+1}^{(k)} \tag{20.11}$$

where

$$\mathbf{K}^* = c_1\mathbf{K} + c_2\mathbf{C} + c_3\mathbf{M} \tag{20.12}$$

with

$$\begin{aligned} c_1 &= 1 \\ c_2 &= \frac{2\beta_1}{\beta_2 \Delta t} \\ c_3 &= \frac{2}{\beta_2 \Delta t^2} \end{aligned} \tag{20.13}$$

obtained from the relations between the incremental displacement, velocity and acceleration vectors. As we have noted in Chapter 18 the changing of the primary unknown from displacement to acceleration or velocity or, indeed, changing the integration algorithm from Newmark to any other method only changes the residual equation by the parameters c_i which define the tangent matrix $\mathbf{K}^*$. The other changes from different integration algorithms appear in the number of vectors required for the algorithm and the way they are initialized and updated within each time increment.

In program *FEAPpv* the parameters c_i are passed to each element routine as `CTAN(i)` together with the values of the localized nodal displacement, velocity and acceleration vectors. This permits an element module to be programmed in a general manner without knowing which integration method will be used during the solution specified in the command language instructions. In Sec. 20.5 we will discuss the steps needed to program the residual terms, as well as the stiffness and mass terms needed to form the global tangent matrix.

Here we note also that the steady-state algorithm discussed in the previous section merely requires that the velocity and acceleration vectors and the parameters c_2 and c_3 be set to zero before calling an element module. Similarly, for a first-order system the acceleration vector and parameter c_3 are set to zero prior to entering the element module.

The command language instructions to solve a linear transient problem over 50 time steps in which all results are reported at each time is given as

```
TRANS,NEWMark   ! Selects Newmark Method
DT,,0.024       ! Sets time increment to 0.024
TANG            ! Form tangent matrix
LOOP,time,50    ! Loop 50 times to NEXT
  FORM          ! Form residual
  SOLVe         ! Solve equations
  DISP,ALL      ! Output nodal displacements
  STRE,ALL      ! Output element variables
NEXT,time       ! End of LOOP
```

The issuing of the instructions `TRANsient` causes the parameters c_i to be set for the Newmark method. The default for the transient option is the steady-state solution algorithm with $c_1 = 1$ and $c_2 = c_3 = 0$.

20.4.3 Non-linear solutions: Newton's methods

The command language programming instructions may also be used to solve non-linear problems. For example, the steady-state set of non-linear algebraic equations given by the residual equation

$$\mathbf{r}^{(k)} = \mathbf{f} - \mathbf{P}(\mathbf{a}^{(k)}) \tag{20.14}$$

in which $\mathbf{P}$ is a non-linear function of $\mathbf{a}$ is considered. A solution may be obtained by writing a linear approximation for the residual at $k+1$ as

$$\mathbf{r}^{(k+1)} \approx \mathbf{r}^{(k)} - \mathbf{K}_T^{(k)} \Delta\mathbf{a}^{(k)} = \mathbf{0} \tag{20.15}$$

in which $\mathbf{K}_T$ is some non-singular coefficient matrix used to obtain the increments $\Delta\mathbf{a}_{(k)}$. Now the update for $\mathbf{a}_{(k+1)}$ using Eq. (20.2) will not in general make $\mathbf{r}^{(k+1)}$ zero in one iteration.

A common method to generate the coefficient matrix is Newton's method where

$$\mathbf{K}_T^{(k)} = \frac{\partial \mathbf{P}}{\partial \mathbf{a}}\Big|_{\mathbf{a}=\mathbf{a}^{(k)}} \tag{20.16}$$

When properly implemented the norm of the residual should converge at a quadratic asymptotic rate. Thus if $||\mathbf{r}||$ is the norm of the residual then for an approximation close to the solution the ratios for two successive iterations should be

$$\frac{||\mathbf{r}^{(k)}||}{||\mathbf{r}^{(0)}||} = C_1 10^{-q}; \qquad \frac{||\mathbf{r}^{(k+1)}||}{||\mathbf{r}^{(0)}||} = C_2 10^{-2q} \tag{20.17}$$

In general, this is the best one can obtain with the type of algorithm given by Eq. (20.15).

In *FEAPpv* a norm of the solution is computed for each iteration and a check of the current norm versus the initial value is performed as indicated in Eq. (20.17). Once the value of the ratio of the norm is below a specified tolerance, convergence is assumed. The solution tolerance is set using the command language instruction `TOL` as indicated in Table 20.5 (the default value for the norm is 10^{-12}). The instructions to perform a solution using the algorithm indicated in Eq. (20.15) is given by

```
LOOP,iteration,10     ! Perform a maximum of 10 iterations
  TANG,,1             ! Compute tangent, residual and solve
NEXT,iteration        ! End for LOOP instruction
```

Once the ratio of the norms is reached, *FEAPpv* will exit the iteration loop and execute the instruction following the `NEXT` statement. If the element module used has a tangent matrix computed using Eq. (20.16) the asymptotic behaviour of Newton's method should be attained. Failure to achieve a quadratic rate of convergence during the last few iterations indicates an incorrect implementation in the element module, a data input error, or extreme sensitivity in the formulation such that round-off prevents the asymptotic rate being reached. One can never achieve convergence beyond that where the round-off limit is reached.

An alternative to the above program is the modified solution method in which the tangent is used from an earlier state. For example, the command language instruction set

```
TANG                  ! Compute tangent
LOOP,iteration,10     ! Perform a maximum of 10 iterations
  FORM                ! Compute residual
  SOLVe               ! Solve equations
NEXT,iteration        ! End for LOOP instruction
```

executes a modified Newton's algorithm and, for general non-linear systems, results in less than a quadratic asymptotic rate of convergence (generally linear or less, so that if iteration k gives a ratio of order 10^{-p}, iteration $k+1$ gives about $10^{-(p+1)}$).

The execution of each `TANG`, `UTAN`, `FORM`, etc. instruction uses the current problem type and time increment to define the parameters c_i along with the current solution values for $\mathbf{a}^{(k)}$, $\dot{\mathbf{a}}^{(k)}$ and $\ddot{\mathbf{a}}^{(k)}$ to calculate a tangent, residual, etc., respectively.

Many additional solution algorithms may be established using the commands available in the program. Some of these are discussed in the user manual where topics ranging from time-dependent loading to general transient, non-linear solution strategies included in *FEAPpv* are described. Authors may be found in Volume 2.

20.4.4 Programming command language statements

The command language module for *FEAPpv* is contained in a set of subprograms whose names begin with `PMAC`. The routine `PMACR` calls the other routines and establishes the limits on the number of commands available to the program. Included

```
      SUBROUTINE UMACR1(LCT,CTL,PRT)
      IMPLICIT   NONE

C     Inputs:
C       LCT    - Command character parameters
C       CTL(3) - Command numerical parameters
C       PRT    - Flag, output if true

C     Outputs:
C       N.B. Users are responsible for command actions.

      IMPLICIT  NONE

      LOGICAL   PCOMP,PRT
      CHARACTER LCT*15
      REAL*8    CTL(3)

      CHARACTER      UCT*4
      COMMON /UMAC1/ UCT

C     Set command word to user selected name

      IF(PCOMP(UCT,'MAC1',4)) THEN
        UCT = 'xxxx'
        RETURN
      ELSE
C       Implement user solution step
      ENDIF

      END
```

Fig. 20.7 Structure of a user command subprogram.

in the current command list is an option to access a set of user subprograms named UMACRn where n ranges from 1 to 5. Each user subprogram has a structure as shown in Fig. 20.7. A user is required to select a four character name for xxxx which does not already exist in the command list in PMACR and to program the desired solution step.

It should be noted that all arrays identified in the subprogram PALLOC can be accessed directly using the data management system described in Sec. 20.3. In addition data may be assigned to space in memory using the TEMPn array names that are also available in PALLOC. Thus it is not necessary to pass the names of arrays through the argument list of the subprograms UMACRn. Quite general routines can be created using these routines; however, if a more involved command is deemed necessary by a user the routines PMACRn may be modified to add additional instructions. This is not an option which should be considered without a thorough study of the new solution option needed, as well as, options already available in the commands included.

If it is decided to modify the PMACRn routines it is necessary to:

1. Increase the size of the WD array in subprogram PMACR by the number of commands to be added.

2. Add the new command name to the list in the data statement for `WD` in subprogram `PMACR` noting which of the routines `PMACRn` will have the solution module added (the continue labels indicate the value of `n`).
3. Increase the value of the variable `NWDn` in the data statement by the number of commands added for each `n`.
4. Add the solution module to the subprograms `PMACRn`. This requires either a modification of a `GO TO` or an `IF-THEN-ELSE` program form in addition to adding the statements.

Again users are reminded that extreme care must be exercised when adding commands in this way. Despite the fact that each command involves a specific solution step or steps there are some interactions between instructions that exist. If these are changed in any way the program may not function properly after new commands are added. This is particularly true for setting the parameters `NWDn` since if these are not correct transfer to incorrect locations in the list can occur.

20.5 Computation of finite element solution modules

20.5.1 Localization of element data

When we want to compute an element array, e.g., an element stiffness matrix, **S**, or an element load or residual vector, **P**, we only need those quantities associated with the one element in question. The nodal and material quantities that are required can be determined from the node and material set numbers stored in the `IX` array for each element. In the program *FEAPpv* the necessary values are moved from each global array to a set of local arrays before the appropriate element routine, `ELMTnn`, is called. The process will be called *localization*. The quantities that are localized are:

1. nodal coordinates which are stored in the local array `XL(NDM,NEN)`;
2. nodal displacements, displacement increments, velocity and acceleration which are stored in the array `UL(NDF,NEN,5)`;
3. nodal T-variables which are stored in the array `TL(NEN)`;
4. equation numbers for assembly which are stored in the destination array `LD(NEN)`.

The `LD` array described in Step 4 above is used to map the element arrays to the global arrays. Accordingly, for the following element array:

$$[\,LD(1) \quad LD(2) \quad LD(3) \quad \cdots\,] \begin{bmatrix} S(1,1) & S(1,2) & S(1,3) & \cdots \\ S(2,1) & S(2,2) & \cdots & \cdots \\ \vdots & \vdots & & \end{bmatrix} \begin{bmatrix} P(1) \\ P(2) \\ \vdots \end{bmatrix}$$

the term `S(i,j)` would be assembled into the global coefficient array (e.g., stiffness matrix) in the position corresponding to row `LD(i)` and column `LD(j)`. Similarly, `P(i)` would be assembled into the position corresponding to the `LD(i)` value. That is, the `LD` array contains the equation numbers of the global arrays. The `LD(i)` assignment of the degrees of freedom for each node is made using the data stored in the `ID(j,k,2)` array as shown in Table 20.2.

The localization process is the same for every type of finite element and is performed in the subprogram `PFORM`, which organizes all computations associated with elements using the connections given in the `IX` array. The maximum number of nodes actually connected to an element is determined and assigned to the parameter `NEL`, which may be less than the maximum `NEN`, and is determined by finding the largest non-zero entry in the `IX` array for each element number. Intermediate zero values are interpreted as no node connected. In this way *FEAPpv* permits the mixing of elements with different numbers of connected nodes, e.g., three-noded triangles can be mixed with four-noded quadrilaterals. Also different types of elements can be mixed such as two-noded shell elements with four-noded quadrilaterals.

Since the current value of the nodal displacements and their increments, as well as the nodal velocities and accelerations for transient problems, is localized for all element computations, the program can be used to solve non-linear problems. This is, in fact, the only additional information required over that needed to solve linear problems and will be discussed further in Volume 2.

20.5.2 Element array computations

The efficient computation of element arrays (in both programmer and computer time) is a crucial aspect of any finite element development. The development of subprograms to evaluate element stiffness and load arrays (or for non-linear problems tangent stiffness and residual arrays) can be efficiently accomplished by a combination of appropriate numerical methods. In order to illustrate a typical development a statement of the essential steps is first given and then some details shown for the two-dimensional linear elastic problem.

A flow chart describing two alternative methods for computing a stiffness matrix is shown in Fig. 20.8. Key steps in the computation are:

1. use of appropriate numerical integration procedures;
2. use of shape function subprograms (which are the same for all problems with the same required continuity);
3. efficient organization of numerical steps.

Gauss–Legendre quadrature formulae are usually utilized to compute element arrays since they provide the highest accuracy for a given number of integration points (see Chapter 9). In some instances it is desirable to use other formulae. For example, if a quadrature formula which samples only at nodes is used, the evaluation of an inertial term leads to a diagonal mass matrix which is more efficient in explicit dynamics calculations.

Shape function subprograms allow a programmer to develop elements for many problems quickly and reliably. A shape function subprogram should evaluate both the shape functions and their derivatives with respect to the global coordinate frame. As an example consider the two-dimensional C_0 problem where we need only first derivatives of each shape function N_i. For the four-noded isoparametric quadrilateral we have

$$N_i = \tfrac{1}{4}(1 + \xi_i\xi)(1 + \eta_i\eta) \tag{20.18}$$

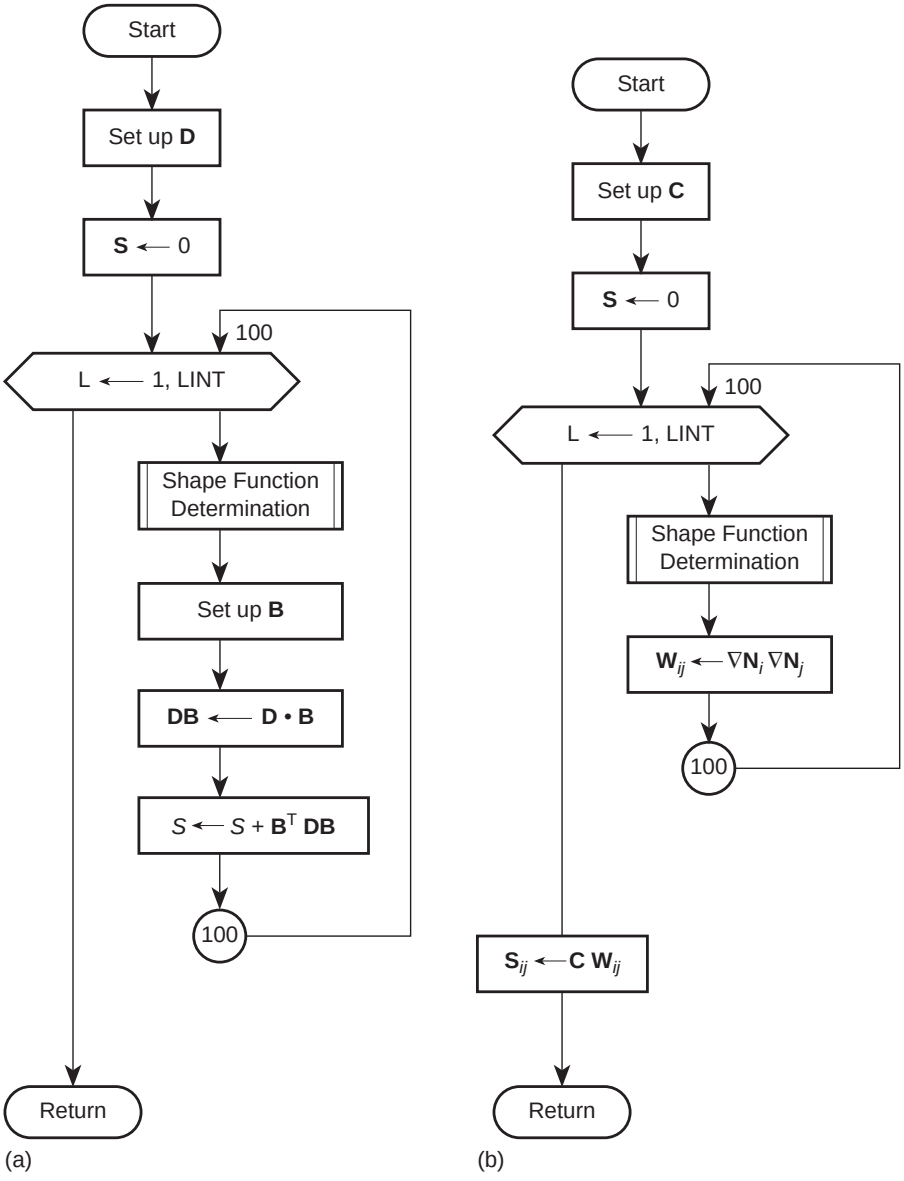


Fig. 20.8 Element stiffness matrix computation: (a) general form; (b) form for constant material properties.

where ξ, η are natural coordinates on the bi-unit square parent element and ξ_i, η_i their values at the four nodes.

Using the isoparametric concept we have

$$\begin{aligned} x &= N_i x_i \\ y &= N_i y_i \end{aligned} \tag{20.19}$$

with derivatives given by

$$\begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix} = \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} \tag{20.20}$$

$$\begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} = \frac{1}{J} \begin{bmatrix} y_{,\eta} & -y_{,\xi} \\ -x_{,\eta} & x_{,\xi} \end{bmatrix} \begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix} \tag{20.21}$$

where J is the jacobian determinant and $(\ )_{,x}$ denotes the partial derivative $\partial(\ )/\partial x$, etc. The above relations define the steps for the shape function subprogram given in Fig. 20.9 where it is assumed that the nodal coordinates have been transferred to the local coordinate array `XL`.

This shape function routine can be used for all two-dimensional C_0 problems which use the four-noded element (e.g., two-dimensional plane and axisymmetric elasticity, heat conduction, flow in porous media, fluid flow, etc.). Shape function subprograms can also be used for the generation of mesh data.[4] It is a simple task to extend the shape function routine to higher order elements (e.g., see the listing for subprogram `SHAP2` in *FEAPpv* which includes options for up to nine-node quadrilaterals). Using such routines permits the use of elements which have individual edges with either linear or quadratic interpolation.

The generation of the matrix products occurring in the stiffness matrix of elasticity problems deserves special attention since zeros often exist in the **B** and **D** matrices. Several methods can be used to reduce the number of operations performed. The first is to form explicitly the matrix products. While this involves extra hand computations it is in fact elementary if performed on a nodal basis. For example, consider the two-dimensional axisymmetric linear elastic problem where

$$\mathbf{B}_i = \begin{bmatrix} N_{i,r} & 0 \\ 0 & N_{i,z} \\ cN_i/r & 0 \\ N_{i,z} & N_{i,r} \end{bmatrix} \tag{20.22}$$

A two-dimensional plane problem may be considered by replacing r, z by x, y and setting the constant c to zero. For axisymmetry the constant c is unity. For an isotropic linear elastic material the moduli are given by

$$\mathbf{D} = \begin{bmatrix} D_{11} & D_{12} & D_{12} & 0 \\ D_{12} & D_{11} & D_{12} & 0 \\ D_{12} & D_{12} & D_{11} & 0 \\ 0 & 0 & 0 & D_{33} \end{bmatrix} \tag{20.23}$$

where D_{33} is the shear modulus given by $(D_{11} - D_{12})/2$. Thus for a typical nodal pair i and j a contribution to the element stiffness $\mathbf{K}_{ij}$ may be computed using

$$\mathbf{Q}_j = \mathbf{D}\mathbf{B}_j \tag{20.24}$$

and

$$\mathbf{K}_{ij} = \mathbf{B}_i^{\mathrm{T}} \mathbf{Q}_j \tag{20.25}$$

```
      SUBROUTINE SHAPE(SS,XL, J,SHP)
C     Shape function routine for 4-node quadrilateral
      IMPLICIT NONE
      INTEGER II   ,JJ     ,KK
      REAL*8  SS(2),XL(2,4),J,SHP(3,4),SI(4),TI(4),XS(2,2),TEMP
      DATA    SI / -0.5D0,  0.5D0, 0.5D0, -0.5D0/
      DATA    TI / -0.5D0, -0.5D0, 0.5D0,  0.5D0/
C     Compute shape functions and natural coordinate derivatives
      DO II = 1,4
        SHP(1,II) = SI(II)*(0.5D0 + TI(II)*SS(2))
        SHP(2,II) = TI(II)*(0.5D0 + SI(II)*SS(1))
        SHP(3,II) = (0.5D0 + SI(II)*SS(1))*(0.5D0 + TI(II)*SS(2))
      END DO ! II
C     Compute Jacobian and Jacobian determinant
      DO II = 1,2
        DO JJ = 1,2
          XS(II,JJ) = 0.0D0
          DO KK = 1,4
            XS(II,JJ) = XS(II,JJ) + XL(II,KK)*SHP(JJ,KK)
          END DO ! KK
        END DO ! JJ
      END DO ! II
      J = XS(1,1)*XS(2,2) - XS(1,2)*XS(2,1)
C     Transform to X,Y derivatives
      DO II = 1,4
        TEMP = ( XS(2,2)*SHP(1,II) - XS(2,1)*SHP(2,II))/J
        SHP(2,II) = (-XS(1,2)*SHP(1,II) + XS(1,1)*SHP(2,II))/J
        SHP(1,II) = TEMP
      END DO ! II
      END
```

Fig. 20.9 Shape function subprogram for four-noded element.

Thus, using Eqs (20.22) and (20.23) and setting

$$n_j = \frac{c}{r} N_j \tag{20.26}$$

we obtain

$$\mathbf{Q}_j = \begin{bmatrix} (D_{11}N_{j,r} + D_{12}n_j) & D_{12}N_{j,z} \\ (D_{12}N_{j,r} + D_{12}n_j) & D_{22}N_{j,z} \\ (D_{12}N_{j,r} + D_{11}n_j) & D_{12}N_{j,z} \\ D_{33}N_{j,z} & D_{33}N_{j,r} \end{bmatrix} \tag{20.27}$$

and finally the stiffness as

$$\mathbf{K}_{ij} = \begin{bmatrix} (N_{i,r}Q_{11} + n_i Q_{31} + N_{i,z}Q_{41}) & (N_{i,r}Q_{12} + n_i Q_{32} + N_{i,z}Q_{42}) \\ (N_{i,z}Q_{21} + N_{i,r}Q_{41}) & (N_{i,z}Q_{22} + N_{i,r}Q_{42}) \end{bmatrix} \tag{20.28}$$

Accordingly, for each nodal pair it is required to perform 21 multiplications to form each $\mathbf{K}_{ij}$, whereas formal multiplication of $B_i^{\mathrm{T}} D B_j$ including all zero operations would require 48 multiplications. When the element stiffness matrix is symmetric it is only necessary to form the upper or lower triangular parts of $\mathbf{K}$ (the other half is formed from the symmetry condition). A typical routine for the stiffness computation is given in Figs 20.10 and 20.11 where it is assumed that the quadrature points are available as `SG(1,L)` equal to ξ_L, `SG(2,L)` equal to η_L, and `SG(3,L)` equal to the quadrature weight.

The increments by `NDF` are to keep the stiffness array stored in nodal order with `NDF`×`NDF` submatrix blocks. This is required by *FEAPpv* to maintain proper compatibility with the routine used to assemble the global arrays.

```
      SUBROUTINE ELSTIF(D, XL, AXI, NDF,NDM,NST, S)
      IMPLICIT   NONE

      LOGICAL    AXI
      INTEGER    II,I1, JJ,J1, L, LINT, NDF,NDM,NST
      REAL*8     DV, D11,D12,D33, J, R
      REAL*8     D(*), XL(NDM,4), S(NST,NST)
      REAL*8     SG(3,4), SHP(3,4), Q(4,2), N(4)

      CALL INT2D(2,LINT, SG) ! Set up 2x2 quadrature points

c     Do numerical integration

      DO L = 1,LINT
        CALL SHAPE(SG(1,L),XL, J,SHP)
        DV = J*SG(3,L)       ! SG(3,L) is quadrature weight
        D11 = D(1)*DV        ! D(1) is D_11 modulus
        D12 = D(2)*DV        ! D(2) is D_12 modulus
        D33 = D(3)*DV        ! D(3) is shear modulus

c       Compute n_i = c*N_i/r

        R = 0.0D0            ! R is radius
        DO II = 1,4
          R = R + SHP(3,II)*XL(1,II)
        END DO ! II
        DO II = 1,4
          IF(AXI) THEN
            N(II) = SHP(3,II)/R
          ELSE
            N(II) = 0.0D0
          ENDIF
        END DO ! II
```

Fig. 20.10 Element stiffness calculation. Part 1.

```
c     Compute Q_j = D * B_j

      J1 = 1
      DO JJ = 1,4
        Q(1,1) = D11*SHP(1,JJ) + D12*N(JJ)
        Q(2,1) = D12*SHP(1,JJ) + D12*N(JJ)
        Q(3,1) = D12*SHP(1,JJ) + D11*N(JJ)
        Q(4,1) = D33*SHP(2,JJ)
        Q(1,2) = D12*SHP(2,JJ)
        Q(2,2) = D11*SHP(2,JJ)
        Q(3,2) = D12*SHP(2,JJ)
        Q(4,2) = D33*SHP(1,JJ)

c       Compute stiffness term: k_ij

        I1 = 1
        DO II = 1,JJ
         S(I1  ,J1  ) = S(I1  ,J1  ) + SHP(1,II)*Q(1,1)+N(II)*Q(3,1)
     &                               + SHP(2,II)*Q(4,1)
         S(I1  ,J1+1) = S(I1  ,J1+1) + SHP(1,II)*Q(1,2)+N(II)*Q(3,2)
     &                               + SHP(2,II)*Q(4,2)
         S(I1+1,J1  ) = S(I1+1,J1  ) + SHP(2,II)*Q(2,1)
     &                               + SHP(1,II)*Q(4,1)
         S(I1+1,J1+1) = S(I1+1,J1+1) + SHP(2,II)*Q(2,2)
     &                               + SHP(1,II)*Q(4,2)
         I1 = I1 + NDF
        END DO ! II
        J1 = J1 + NDF
       END DO ! JJ
     END DO ! L

c    Compute lower part by symmetry

     DO II = 1,NST
       DO JJ = 1,II
         S(II,JJ) = S(JJ,II)
       END DO ! JJ
     END DO ! II

     END
```

Fig. 20.11 Element stiffness calculation. Part 2.

An extension to anisotropic problems can be made by replacing the isotropic **D** matrix by the appropriate anisotropic one and then recomputing the $\mathbf{Q}_j$ matrix.

The computation of element stiffness matrices for two-dimensional plane and three-dimensional problems which have constant material properties within an element can be made more efficient than that given above. This is obtained by

noting from Appendix B that the internal energy may be written in indicial form as

$$W(\mathbf{u}) = \tfrac{1}{2}\tilde{u}_a^i D_{abcd} \int_{V_e} N_{i,b} N_{j,d}\, \mathrm{d}V\, u_c^j \tag{20.29}$$

where a, b, c, d are indices from the elasticity equations and range over the space dimension of the problem and i, j are nodal indices which range from 1 to `NEL` in each element. The element stiffness for the nodal pair i, j may be written as

$$\mathbf{K}_{ac}^{ij} = W_{bd}^{ij} D_{abcd} \tag{20.30}$$

where

$$W_{bd}^{ij} = \int_{V_e} N_{i,b} N_{j,d}\, \mathrm{d}V \tag{20.31}$$

For isotropic materials

$$D_{abcd} = \lambda \delta_{ab} \delta_{cd} + \mu(\delta_{ac}\delta_{bd} + \delta_{ad}\delta_{bc}) \tag{20.32}$$

where λ and μ are the Lamé elastic constants which are related to the usual elastic constants E and μ as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}; \qquad \mu = \frac{E}{2(1+\nu)}$$

Thus, the stiffness matrix for an isotropic material is given as

$$K_{ac}^{ij} = \lambda W_{ac}^{ij} + \mu(W_{ca}^{ij} + \delta_{ac} W_{bb}^{ij}) \tag{20.33}$$

Using this approach the steps to compute the element stiffness matrix for plane elasticity are given in Fig. 20.8(b). This procedure for computing stiffness matrices was noted in reference 5 and for plane problems results in about 25% fewer numerical operations than the procedure shown in Fig. 20.8(a). In three dimensions the savings are even greater.

The computation of other element arrays can also be performed using a shape function routine. For example, the computation of the element consistent and diagonal mass matrices by the row sum method (see Appendix I) for transient or eigenvalue computations can be easily constructed. The consistent mass matrix for two- and three-dimensional problems is obtained from

$$\mathbf{M}_{ij} = \mathbf{I} \int_{V_e} \rho N_i N_j\, \mathrm{d}V \tag{20.34}$$

whereas the diagonal mass is computed from

$$\mathbf{M}_{jj} = \mathbf{I} \int_{V_e} \rho N_j\, \mathrm{d}V \tag{20.35}$$

In the above $\mathbf{I}$ is an identity matrix of size `NDM` and ρ is the mass density. A set of statements to compute the mass matrix for these cases is shown in Fig. 20.12 where the element consistent mass is stored in the square matrix $\mathbf{S}$ and the diagonal mass matrix is stored in the rectangular array $\mathbf{P}$.

The shape function routine may also be used to compute strains, stresses and internal forces in an element. The strains at each point in an element may be

```
C   S(NST,NST) : Consistent mass array
C   P(NDM,NEL) : Diagonal mass array

C   Numerical integration loop

    DO L = 1,LINT
      CALL SHAPE(SG(1,L), XL, J, SHP)
      DMASS = RHO*J*SG(3,L)
      J1 = 1
      DO JJ = 1,NEL
        JMASS   = DMASS*SHP(3,JJ)
        P(1,JJ) = P(1,JJ) + JMASS
        I1 = 1
        DO II = 1,NEL
          S(I1,J1) = S(I1,J1) + SHP((3,II)*JMASS
          I1       = I1 + NDF
        END DO ! II
      J1 = J1 + NDF
      END DO ! JJ
    END DO ! L

C   Copy using identity matrix

    J1 = 0
    DO JJ = 1,NEL
      DO KK = 2,NDM
        P(KK,JJ) = P(1,JJ)
      END DO ! KK
      I1 = 0
      DO II = 1,NEL
        DO KK = 2,NDM
          S(I1+KK,J1+KK) = S(I1+1,J1+1)
        END DO ! KK
        I1 = I1 + NDF
      END DO ! II
      J1 = J1 + NDF
    END DO ! JJ
```

Fig. 20.12 Diagonal (lumped) and consistent mass matrix for an isoparametric element.

computed from

$$\boldsymbol{\varepsilon} = \mathbf{B}_i(\boldsymbol{\xi})_i \tilde{\mathbf{u}}_i \tag{20.36}$$

where $\boldsymbol{\xi}$ is the set of local natural coordinates and $\tilde{\mathbf{u}}_i$ are the nodal displacements at node i. A subprogram to compute the strains for the two-dimensional case given by Eq. (20.22) is shown in Fig. 20.13. Stresses are now computed as usual from

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{20.37}$$

or any other relationship expressed in terms of the strains. The above form is more general and efficient than saving the values in the $\mathbf{Q}_i$ matrices during stiffness

```
      SUBROUTINE STRAIN(XL, UL, SHP, NDM,NDF,NEN,NEL, EPS,R, AXI)

      IMPLICIT NONE

      LOGICAL  AXI
      INTEGER  NDM,NDF,NEN,NEL, II
      REAL*8   XL(NDM,*),UL(NDF,NEN,*),SHP(3,*), EPS(4),R

C     Initialize strains and radius

      DO II = 1,4
        EPS(II) = 0.0D0
      END DO ! II
      R = 0.0D0

C     Sum strains from shape functions and nodal values

      DO II = 1,NEL
        EPS(1) = EPS(1) + SHP(1,II)*UL(1,II,1)
        EPS(2) = EPS(2) + SHP(2,II)*UL(2,II,1)
        EPS(3) = EPS(3) + SHP(3,II)*UL(1,II,1)
        EPS(4) = EPS(4) + SHP(1,II)*UL(2,II,1) + SHP(2,II)*UL(1,II,1)
        R      = R      + SHP(3,II)*XL(1,II)
      END DO ! II

C     Modify hoop strain if axisymmetric; zero for plane problem

      IF(AXI) THEN
        EPS(3) = EPS(3)/R
      ELSE
        EPS(3) = 0.0D0
      ENDIF

      END
```

Fig. 20.13 Strain calculation for isoparametric element.

evaluation and then computing the stresses from

$$\boldsymbol{\sigma} = \mathbf{D}\mathbf{B}_i\tilde{\mathbf{u}}_i = \mathbf{Q}_i\tilde{\mathbf{u}}_i \tag{20.38}$$

This would require significant additional storage or saving and retrieving the $\mathbf{Q}_i$ from backing store as given in reference 6. Moreover, it is often desirable to compute the stresses at points other than those used to compute the stiffness matrix as indicated in Chapter 14 for recovery processes. In non-linear problems the computation of strains and stresses must also be performed directly. Thus, for all the above reasons it is desirable to compute strains as necessary using the technique given in Fig. 20.13.

In *FEAPpv* the stresses must also be determined to compute element residuals. One of the main terms in the element residual is the internal stress term and here again shape function routines are useful. The internal force term for problems in elasticity (and, as will be shown in the Volume 2, also for finite deformation inelastic

```
C     Quadrature loop

      DO L = 1,LINT

C       Compute shape functions

        CALL SHAPE(SG(1,L), XL, J, SHP)
        DV = J*SG(3,L)

C       Compute strains

        CALL STRAIN(XL, UL, SHP, NDM,NDF,NEN,NEL, EPS,R, AXI)
        DO II = 1,NEL
          IF(AXI) THEN
            N(II) = SHP(3,II)/R
          ELSE
            N(II) = 0.0D0
          ENDIF
        END DO ! II

C       Compute stresses

        CALL STRESS(EPS, SIG)

C       Compute internal forces

        DO II = 1,NEL
          P(1,II) = P(1,II) - (SHP(1,II)*SIG(1) + SHP(2,II)*SIG(4)
     &                       + N(II)*SIG(3))*DV
          P(2,II) = P(2,II) - (SHP(2,II)*SIG(2) + SHP(1,II)*SIG(4))*DV
        END DO ! II
      END DO ! L
```


Fig. 20.14 Internal force computation.

problems) is given by

$$\mathbf{P}_i = -\int_{V_e} \mathbf{B}_i^{\mathrm{T}} \boldsymbol{\sigma} \, \mathrm{d}V \tag{20.39}$$

The programming steps to compute are given in Fig. 20.14.

The generality of an isoparametric C_0 shape function routine can be exploited to program element routines for other problems. For example, Fig. 20.15 gives the necessary program instructions to compute the 'stiffness' matrix for problems of the quasi-harmonic equation discussed in Chapters 3 and 7.

20.5.3 Organization of element routines

The previous discussion has focused on procedures for determining element arrays. The reader will note that the element square matrices for stiffness and mass were

```
C     Quadrature loop

      DO L = 1,LINT

C       Compute shape functions

        CALL SHAPE(SG(1,L), XL, J, SHP)
        DV = J*SG(3,L)
        KK = D(1)*DV  ! Conductivity times volume

C       For each JJ-node compute the D*B

        DO JJ = 1,NEL
          DO KK = 1,NDM
            Q(KK) = D1*SHP(KK,JJ)
          END DO ! KK

C         For each II-node compute the coefficient matrix

          DO II = 1,JJ
            DO KK = 1,NDM
              S(II,JJ) = S(II,JJ) + SHP(KK,II)*Q(KK)
            END DO ! KK
          END DO ! II
        END DO ! JJ

      END DO ! L
```

Fig. 20.15 Coefficient matrix for quasi-harmonic operator.

both stored in the square array **S** while element vectors were stored in the rectangular array **P**. This was intentional since all aspects of computing element arrays for the program are to be consolidated into a single subprogram called the *element routine*. An element routine is called by the element library subprogram `ELMLIB`. As given here, the element library provides space for ten element subprograms at any one time, where as noted previously these are named `ELMTnn` with `nn` ranging from `01` to `10`. This can easily be increased by modifying the subprogram `ELMLIB`. The subprogram `ELMLIB` is, in turn, called from the subprogram `PFORM` which is the routine to loop through all elements and perform the localization step to set up local coordinates `XL`, displacements, etc., `UL` and equation numbers for global assembly `LD`. The subprogram `PFORM` also uses subprogram `DASBLE` to assemble element arrays into global arrays and uses subprogram `MODIFY` to perform appropriate modifications for prescribed non-zero displacements. When an element routine is accessed the value of a parameter `ISW` is given a value between 1 and 10. The parameter specifies what action is to be performed in the element routine. Each element routine must provide appropriate transfers for each value of `ISW`. A mock element routine for *FEAPpv* is shown in Fig. 20.16.

```
      SUBROUTINE ELMTnn(D,UL,XL,IX,TL, S,P, NDF,NDM,NST, ISW)

      IMPLICIT   NONE

      INTEGER    NDF,NDM,NST, ISW, IX(NEN1,*)
      REAL*8     D(*),UL(NDF,NEN,*),XL(NDM,*),TL(*), S(NST,*),P(NDF,*)

C     Input and output material set data
      IF(ISW.EQ.1) THEN
        Use D(*) to store input parameters
C     Check element for errors
      ELSEIF(ISW.EQ.2) THEN
        Check element for negative jacobians, etc.
C     Form element coefficient matrix and residual vector
      ELSEIF(ISW.EQ.3 .OR. ISW.EQ.6) THEN
        The S(NST,NST) array stores coefficient matrix
        The P(NDF,NEL) array stores residual vector
C     Output element results (e.g., stress, strain, etc.)
      ELSEIF(ISW.EQ.4) THEN

C     Compute element mass arrays
      ELSEIF(ISW.EQ.5) THEN
        The S(NST,NST) array stores consistent mass
        The P(NDF,NEL) array stores lumped mass
C     Compute element error estimates
      ELSEIF(ISW.EQ.7) THEN

C     Project element results to nodes
      ELSEIF(ISW.EQ.8) THEN

C     Project element error estimator
      ELSEIF(ISW.EQ.9) THEN

C     Augmented lagragian update
      ELSEIF(ISW.EQ.10) THEN

      ENDIF

      END
```

Fig. 20.16 Mock element routine functions.

20.6 Solution of simultaneous linear algebraic equations

A finite element problem leads to a large set of simultaneous linear algebraic equations whose solution provides the nodal and element parameters in the formulation. For example, in the analysis of linear steady-state problems the direct assembly of the element coefficient matrices and load vectors leads to a set of linear algebraic equations. In this section methods to solve the simultaneous algebraic equations are summarized. We consider both *direct* methods where an *a priori* calculation of

the number of numerical operations can be made, and *indirect or iterative* methods where no such estimate can be made.

20.6.1 Direct solution

Consider first the general problem of direct solution of a set of algebraic equations given by

$$\mathbf{K}\mathbf{a} = \mathbf{b} \tag{20.40}$$

where $\mathbf{K}$ is a square coefficient matrix, $\mathbf{a}$ is a vector of unknown parameters and $\mathbf{b}$ is a vector of known values. The reader can associate these with the quantities described previously: namely, the stiffness matrix, the nodal unknowns, and the specified forces or residuals.

In the discussion to follow it is assumed that the coefficient matrix has properties such that row and/or column interchanges are unnecessary to achieve an accurate solution. This is true in cases where $\mathbf{K}$ is symmetric positive (or negative) definite.† Pivoting may or may not be required with unsymmetric, or indefinite, conditions which can occur when the finite element formulation is based on some weighted residual methods. In these cases some checks or modifications may be necessary to ensure that the equations can be solved accurately.[7–9]

For the moment consider that the coefficient matrix can be written as the product of a lower triangular matrix with unit diagonals and an upper triangular matrix. Accordingly,

$$\mathbf{K} = \mathbf{L}\mathbf{U} \tag{20.41}$$

where

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ L_{21} & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & 1 \end{bmatrix} \tag{20.42}$$

and

$$\mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ 0 & U_{22} & \cdots & U_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & U_{nn} \end{bmatrix} \tag{20.43}$$

† For mixed methods which lead to forms of the type given in Eq. (11.14) the solution is given in terms of a positive definite part for $\tilde{\mathbf{q}}$ followed by a negative definite part for $\tilde{\boldsymbol{\phi}}$. Thus, interchanges are not needed providing the ordering of the equation is defined as described in Sec. 20.2.4.

This form is called a *triangular decomposition* of **K**. The solution to the equations can now be obtained by solving the pair of equations

$$\mathbf{L}\mathbf{y} = \mathbf{b} \tag{20.44}$$

and

$$\mathbf{U}\mathbf{a} = \mathbf{y} \tag{20.45}$$

where **y** is introduced to facilitate the separation, e.g., see references 7–11 for additional details.

The reader can easily observe that the solution to these equations is trivial. In terms of the individual equations the solution is given by

$$\begin{aligned} y_1 &= b_1 \\ y_i &= b_i - \sum_{j=1}^{i-1} L_{ij} y_j \qquad i = 2, 3, \ldots, n \end{aligned} \tag{20.46}$$

and

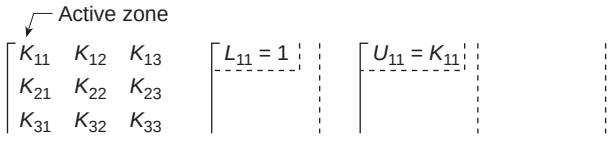
$$\begin{aligned} a_n &= \frac{y_n}{U_{nn}} \\ a_i &= \frac{1}{U_{ii}} \left(y_i - \sum_{j=i+1}^{n} U_{ij} a_j \right) \qquad i = n-2, n-3, \cdots, 1 \end{aligned} \tag{20.47}$$

Equation (20.46) is commonly called *forward elimination* while Eq. (20.47) is called *back substitution*.

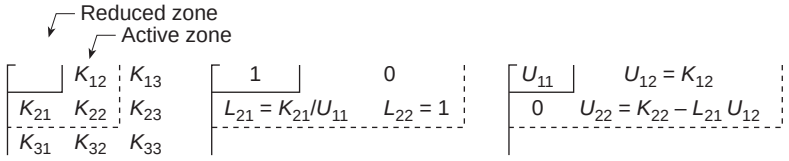
The problem remains to construct the triangular decomposition of the coefficient matrix. This step is accomplished using variations on Gaussian elimination. In practice, the operations necessary for the triangular decomposition are performed directly in the coefficient array; however, to make the steps clear the basic steps are shown in Fig. 20.17 using separate arrays. The decomposition is performed in the same way as that used in the subprogram `DATRI` contained in the *FEAPpv* program; thus, the reader can easily grasp the details of the subprograms included once the steps in Fig. 20.17 are mastered. Additional details on this step may be found in references 9–11.

In `DATRI` the Crout form of Gaussian elimination is used to successively reduce the original coefficient array to upper triangular form. The lower portion of the array is used to store $\mathbf{L} - \mathbf{I}$ as shown in Fig. 20.17. With this form, the unit diagonals for **L** are not stored.

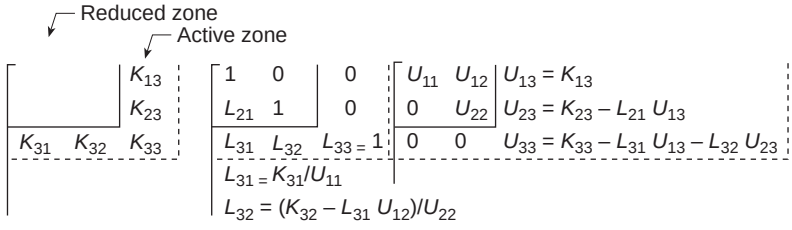
Based on the organization of Fig. 20.17 it is convenient to consider the coefficient array to be divided into three parts: part one being the region that is fully reduced; part two the region which is currently being reduced (called the active zone); and part three the region which contains the original unreduced coefficients. These regions are shown in Fig. 20.18 where the jth column above the diagonal and the jth row to the left of the diagonal constitute the active zone. The algorithm for the triangular decomposition of an $n \times n$ square matrix can be deduced from Fig. 20.17 and



Step 1. Active zone. First row and column to principal diagonal.



Step 2. Active zone. Second row and column to principal diagonal. Use first row of **K** to eliminate $L_{21}\ U_{11}$. The active zone uses only values of **K** from the active zone and values of **L** and **U** which have already been computed in steps 1 and 2.



Step 3. Active zone. Third row and column to principal diagonal. Use first row to eliminate $L_{31}\ U_{11}$; use second row of reduced terms to eliminate $L_{32}\ U_{22}$ (reduced coefficient K_{32}). Reduce column 3 to reflect eliminations below diagonal.

Fig. 20.17 Triangular decomposition of **K**.

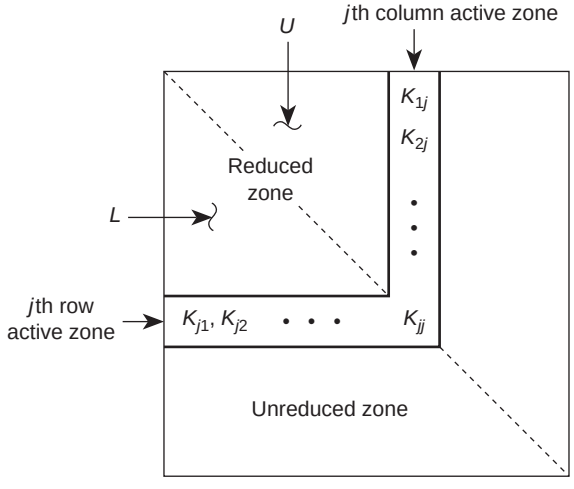


Fig. 20.18 Reduced, active and unreduced parts.

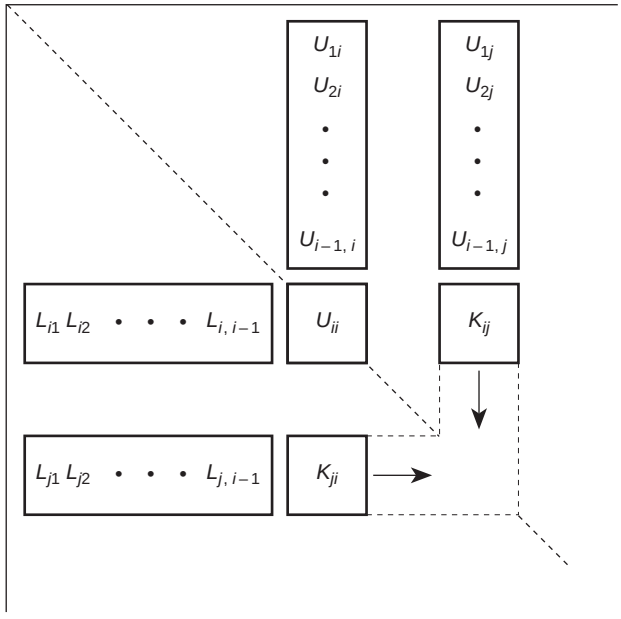


Fig. 20.19 Terms used to construct U_{ij} and L_{ji}.

Fig. 20.19 as follows:

$$U_{11} = K_{11}; \qquad L_{11} = 1 \tag{20.48}$$

For each active zone j from 2 to n,

$$L_{j1} = \frac{K_{j1}}{U_{11}}; \qquad U_{1j} = K_{1j} \tag{20.49}$$

$$\begin{aligned} L_{ji} &= \frac{1}{U_{ii}}\left(K_{ji} - \sum_{m=1}^{i-1} L_{jm}U_{mi}\right) \\ U_{ij} &= K_{ij} - \sum_{m=1}^{i-1} L_{im}U_{mj} \qquad i = 2, 3, \ldots, j-1 \end{aligned} \tag{20.50}$$

and finally

$$\begin{aligned} L_{jj} &= 1 \\ U_{jj} &= K_{jj} - \sum_{m=1}^{j-1} L_{jm}U_{mj} \end{aligned} \tag{20.51}$$

The ordering of the reduction process and the terms used are shown in Fig. 20.19. The results from Fig. 20.17 and Eqs (20.48)–(20.51) can be verified by the reader using the matrix given in the example shown in Table 20.6.

Once the triangular decomposition of the coefficient matrix is computed, several solutions for different right-hand sides **b** can be computed using Eqs (20.46) and (20.47). This process is often called a *resolution* since it is not necessary to recompute

Table 20.6 Example: triangular decomposition of 3×3 matrix

K	**L**	**U**
$\begin{bmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & & \\ & & \\ & & \end{bmatrix}$	$\begin{bmatrix} 4 & & \\ & & \\ & & \end{bmatrix}$

Step 1. $L_{11} = 1$, $U_{11} = 4$

$\begin{bmatrix} & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & & \\ 0.5 & 1 & \\ & & \end{bmatrix}$	$\begin{bmatrix} 4 & 2 & \\ & 3 & \\ & & \end{bmatrix}$

Step 2. $L_{21} = \frac{2}{4} = 0.5$, $U_{12} = 2$, $U_{22} = 1$, $U_{22} = 4 - 0.5 \times 2 = 3$

$\begin{bmatrix} & & 1 \\ & & 2 \\ 1 & 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & & \\ 0.5 & 1 & \\ 0.25 & 0.5 & 1 \end{bmatrix}$	$\begin{bmatrix} 4 & 2 & 1 \\ & 3 & 1.5 \\ & & 3 \end{bmatrix}$

Step 3. $L_{31} = \frac{1}{4} = 0.25$, $U_{13} = 1$, $L_{32} = \dfrac{2 - 0.25 \times 2}{3} = \dfrac{1.5}{3} = 0.5$

$U_{23} = 2 - 0.5 \times 1 = 1.5$, $L_{33} = 1$, $U_{33} = 4 - 0.25 \times 1 - 0.5 \times 1.5 = 3$

$$\begin{bmatrix} 1 & & \\ 0.5 & 1 & \\ 0.25 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 2 & 1 \\ & 3 & 1.5 \\ & & 3 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

Step 4. Check

the **L** and **U** arrays. For large size coefficient matrices the triangular decomposition step is very costly while a resolution is relatively cheap; consequently, a resolution capability is necessary in any finite element solution system using a direct method.

The above discussion considered the general case of equation solving (without row or column interchanges). In coefficient matrices resulting from a finite element formulation some special properties are usually present. Often the coefficient matrix is symmetric ($K_{ij} = K_{ji}$) and it is easy to verify in this case that

$$U_{ij} = L_{ji} U_{ii} \tag{20.52}$$

For this problem class it is not necessary to store the entire coefficient matrix. It is sufficient to store only the coefficients above (or below) the principal diagonal and the diagonal coefficients. Equation (20.52) may be used to construct the missing part. This reduces by almost half the required storage for the coefficient array as well as the computational effort to compute the triangular decomposition.

The required storage can be further reduced by storing only those rows and columns which lie within the region of non-zero entries of the coefficient array. Problems formulated by the finite element method and the Galerkin process normally have a symmetric profile which further simplifies the storage form. Storing the upper and lower parts in separate arrays and the diagonal entries of **U** in a third array is used in `DATRI`. Figure 20.20 shows a typical *profile* matrix and the storage order adopted

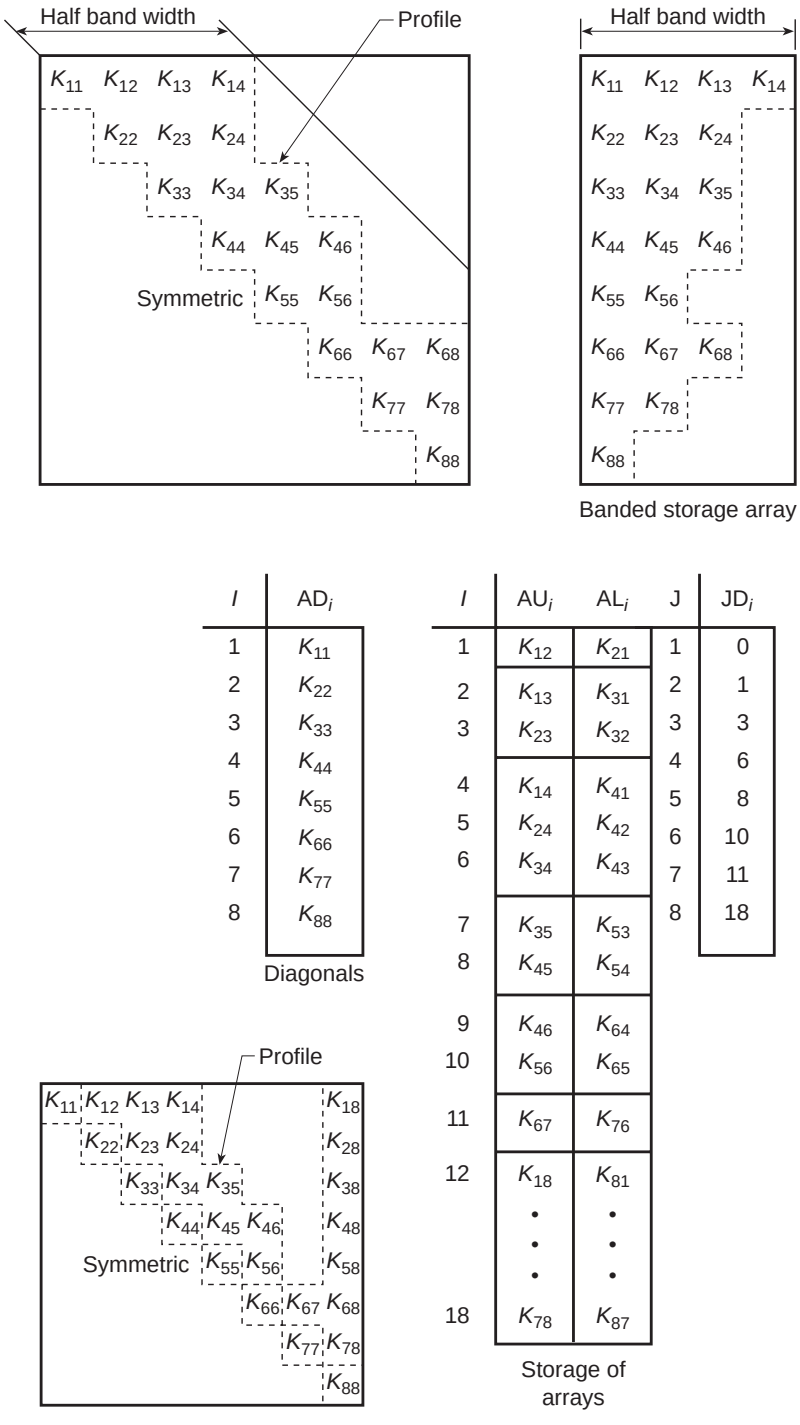


Fig. 20.20 Profile storage for coefficient matrix.

for the upper array AU, the lower array AL and the diagonal array AD. An integer array JD is used to locate the start and end of entries in each column. With this scheme it is necessary to store and compute only within the non-zero profile of the equations. This form of storage does not severely penalize the presence of a few large columns/rows and is also an easy form to program a resolution process (e.g., see subprogram DASOL in *FEAPpv* and reference 10).

The routines included in *FEAPpv* are restricted to problems for which the coefficient matrix can fit within the space allocated in the main storage array. In two-dimensional formulations, problems with several thousand degrees of freedom can be solved on today's personal computers. In three-dimensional cases however problems are restricted to a few thousand equations. To solve larger size problems there are several options. The first is to retain only part of the coefficient matrix in the main array with the rest saved on backing store (e.g., hard disk). This can be quite easily achieved but the size of problem is not greatly increased due to the very large solve times required and the rapid growth in the size of the profile-stored coefficient matrix in three-dimensional problems.

A second option is to use sparse solution schemes. These lead to significant program complexity over the procedure discussed above but can lead to significant savings in storage demands and compute time – especially for problems in three dimensions. Nevertheless, capacity in terms of storage and compute time is again rapidly encountered and alternatives are needed.

20.6.2 Iterative solution

One of the main problems in direct solutions is that terms within the coefficient matrix which are zero from a finite element formulation become non-zero during the triangular decomposition step. While sparse methods are better at limiting this fill than profile methods they still lead to a very large increase in the number of non-zero terms in the factored coefficient matrix. To be more specific consider the case of a three-dimensional linear elastic problem solved using eight-node isoparametric hexahedron elements. In a regular mesh each interior node is associated with 26 other nodes, thus, the equation of such a node has 81 non-zero coefficients – three for each of the 27 associated nodes. On the other hand, for a rectangular block of elements with n nodes on each of the sides the typical column height is approximately proportional to n^2 and the number of equations to n^3. In Table 20.7 we show the size and approximate number of non-zero terms in $\mathbf{K}$ from a finite formulation for linear elasticity (i.e., with three degrees of freedom per node). The table also indicates the size growth with column height and storage requirements for a direct solution based on a profile solution method.

From the table it can be observed that the demands for a direct solution are growing very rapidly (storage is approximately proportional to n^5) while at the same time the demands for storing the non-zero terms in the stiffness matrix grows proportional to the number of equations (i.e., proportional to n^3 for the block).

Iterative solution methods use the terms in the stiffness matrix directly and thus for large problems have the potential to be very efficient for large three-dimensional problems. On the other hand, iterative methods require the resolution of a set of

Table 20.7

Side nodes	Number of equations	Non-zeros in K		Profile storage data		
		Words ($\times 10^{-6}$)	Mbytes	Col. Ht.	Words ($\times 10^{-6}$)	Mbytes
5	375	0.02	0.12	90	0.03	0.27
10	3000	0.12	0.96	330	0.99	7.92
20	24000	0.96	7.68	1260	30.24	241.82
40	192000	7.68	61.44	4920	944.64	7557.12
80	1536000	61.44	491.52	18440	28323.84	226584.72

equations until the residual of the linear equations, given by

$$\mathbf{r}^{(i)} = \mathbf{b} - \mathbf{K}\mathbf{a}^{(i)} \tag{20.53}$$

becomes less than a specified tolerance.

In order to be effective the number of iterations i to achieve a solution must be quite small – generally no larger than a few hundred. Otherwise, excessive solution costs will result. At the time of writing this book the subject of iterative solution for general finite element problems remains a topic of intense research. There are some impressive results available for the case where **K** is symmetric positive (or negative) definite; however, those for other classes (e.g., unsymmetric or indefinite forms) are generally not efficient enough for reliable use in the solution of general problems.

For the symmetric positive definite case methods based on a preconditioned conjugate gradient method have been particularly effective.[12–14] The convergence of the method depends on the condition number of the matrix **K** – the larger the condition number, the slower the convergence (see reference 9 for more discussion). The condition number for a finite element problem with a symmetric positive definite stiffness matrix **K** is defined as

$$\kappa = \frac{\lambda_n}{\lambda_1} \tag{20.54}$$

where λ_1 and λ_n are the smallest and largest eigenvalue from the solution of the eigenproblem (viz. Chapter 17)

$$\mathbf{K}\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{20.55}$$

in which $\boldsymbol{\Lambda}$ is a diagonal matrix containing the individual eigenvalues λ_i and the columns of $\boldsymbol{\Phi}$ are the eigenvectors $_i$ associated with each of the eigenvalues.

Usually, the condition number for an elasticity problem modelled by the finite element method is too large to achieve rapid convergence and a *preconditioned conjugate* gradient (PCG) is used.[12] A symmetric form of preconditioned system is written as

$$\mathbf{K}_p\mathbf{z} = \mathbf{P}\mathbf{K}\mathbf{P}^{\mathrm{T}}\mathbf{z} = \mathbf{P}\mathbf{b} \tag{20.56}$$

where

$$\mathbf{P}^{\mathrm{T}}\mathbf{z} = \mathbf{a} \tag{20.57}$$

Now the convergence of the PCG algorithm depends on the condition number of $\mathbf{K}_p$. The problem remains to construct a preconditioner which adequately reduces

the condition number. In *FEAPpv* the diagonal of **K** is used, however, more efficient schemes incorporating also multigrid methods are discussed in references 13 and 14.

20.7 Extension and modification of computer program *FEAPpv*

The previous sections briefly discussed the basis for the program *FEAPpv* which is available from the publishers web site at no cost. The capabilities of the program are quite significant – mainly due to the flexibility of the command language solution strategy. However, the program can be improved in many ways. Improvements to increase the size of problems which can be solved have already been mentioned. Other improvements include additional command language statements to handle special needs of each user, preprocessors to assist in preparation of input data and postprocessors to permit a wider range of graphical output options. In the latter two instances the program GiD[3] provides features which can greatly assist users in the preparation of mesh data and the display of results†.

In order to facilitate the addition of new input features and/or new command language statements the program *FEAPpv* includes a number of options for users to add subprograms without the need to modify the `PMESH` or the `PMACRn` routines.

References

1. O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*, volume 1. McGraw-Hill, London, 4th edition, 1989.
2. O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*, volume 2. McGraw-Hill, London, 4th edition, 1991.
3. GiD – The Personal Pre/Postprocesor (Version 5.0). Barcelona, Spain, 1999.
4. O.C. Zienkiewicz. *The Finite Element Method in Engineering Science*. McGraw-Hill, London, 2nd edition, 1971.
5. A.K. Gupta and B. Mohraz. A method of computing numerically integrated stiffness matrices. *Internat. J. Num. Meth. Eng.*, **5**, 83–9, 1972.
6. E.L. Wilson. SAP – a general structural analysis program for linear systems. *Nucl. Engr. Des.*, **25**, 257–74, 1973.
7. A. Ralston. *A First Course in Numerical Analysis*. McGraw-Hill, New York, 1965.
8. J.H. Wilkinson and C. Reinsch. *Linear Algebra. Handbook for Automatic Computation*, volume II. Springer-Verlag, Berlin, 1971.
9. J. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
10. R.L. Taylor. Solution of linear equations by a profile solver. *Eng. Comp.*, **2**, 344–50, 1985.
11. G. Strang. *Linear Algebra and its Application*. Academic Press, New York, 1976.
12. R.M. Ferencz. *Element-by-element preconditioning techniques for large-scale, vectorized finite element analysis in nonlinear solid and structural mechanics*. Ph.D thesis, Stanford University, Stanford, California, 1989.

† Options to acquire GiD are also provided at the publishers web sit.

13. M. Adams. Heuristics for automatic construction of coarse grids in multigrid solvers for finite element matrices. Technical Report UCB//CSD-98–994, University of California, Berkeley, 1998.
14. M. Adams. Parallel muiltigrid algorithms for unstructured 3D large deformation elasticity and plasticity finite element problems. Technical Report UCB//CSD-99–1036, University of California, Berkeley, 1999.

Appendix A

Matrix algebra

The mystique surrounding matrix algebra is perhaps due to the texts on the subject requiring a student to 'swallow too much' at one time. It will be found that in order to follow the present text and carry out the necessary computation only a limited knowledge of a few basic definitions is required.

Definition of a matrix

The linear relationship between a set of variables x and b

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \end{aligned} \tag{A.1}$$

can be written, in a shorthand way, as

$$[A]\{x\} = \{b\} \tag{A.2}$$

or

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \tag{A.3}$$

where

$$\mathbf{A} \equiv [A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \tag{A.4}$$

$$\mathbf{x} \equiv \{x\} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix}$$

$$\mathbf{b} \equiv \{b\} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

The above notation contains within it both the definition of a matrix and of the process of multiplication of two matrices. Matrices are *defined* as 'arrays of numbers' of the type shown in Eq. (A.4). The particular form listing a single column of numbers is often referred to as a vector or column matrix, whereas a matrix with multiple columns and rows is called a rectangular matrix. The multiplication of a matrix by a column vector is *defined* by the equivalence of the left and right sides of Eqs (A.1) and (A.2).

The use of bold characters to define both vectors and matrices will be followed throughout the text, generally lower case letters denoting vectors and capital letters matrices.

If another relationship, using the same a-constants, but a different set of x and b, exists and is written as

$$\begin{aligned} a_{11}x'_1 + a_{12}x'_2 + a_{13}x'_3 + a_{14}x'_4 &= b'_1 \\ a_{21}x'_1 + a_{22}x'_2 + a_{23}x'_3 + a_{24}x'_4 &= b'_2 \\ a_{31}x'_1 + a_{32}x'_2 + a_{33}x'_3 + a_{34}x'_4 &= b'_3 \end{aligned} \tag{A.5}$$

then we could write

$$[A][X] = [B] \qquad \text{or} \qquad \mathbf{AX} = \mathbf{B} \tag{A.6}$$

in which

$$\mathbf{X} \equiv [X] = \begin{bmatrix} x_1, & x'_1 \\ x_2, & x'_2 \\ x_3, & x'_3 \\ x_4, & x'_4 \end{bmatrix} \qquad \mathbf{B} \equiv [B] = \begin{bmatrix} b_1, & b'_1 \\ b_2, & b'_2 \\ b_3, & b'_3 \end{bmatrix} \tag{A.7}$$

implying both the statements (A.1) and (A.5) arranged simultaneously as

$$\begin{bmatrix} a_{11}x_1 + \cdots, & a_{11}x'_1 + \cdots \\ a_{21}x_1 + \cdots, & a_{21}x'_1 + \cdots \\ a_{31}x_1 + \cdots, & a_{31}x'_1 + \cdots \end{bmatrix} = \mathbf{B} \equiv [B] = \begin{bmatrix} b_1, & b'_1 \\ b_2, & b'_2 \\ b_3, & b'_3 \end{bmatrix} \tag{A.8}$$

It is seen, incidentally, that matrices can be equal only if each of the individual terms is equal.

The multiplication of full matrices is defined above, and it is obvious that it has a meaning only if the number of columns in **A** is equal to the number of rows in **X** for a relation of the type (A.6).One property that distinguishes matrix multiplication is that, in general,

$$\mathbf{AX} \neq \mathbf{XA}$$

i.e., multiplication of matrices is not commutative as in ordinary algebra.

Matrix addition or subtraction

If relations of the form (A.1) and (A.5) are added then we have

$$\begin{aligned} a_{11}(x_1 + x'_1) + a_{12}(x_2 + x'_2) + a_{13}(x_3 + x'_3) + a_{14}(x_4 + x'_4) &= b_1 + b'_1 \\ a_{21}(x_1 + x'_1) + a_{22}(x_2 + x'_2) + a_{23}(x_3 + x'_3) + a_{24}(x_4 + x'_4) &= b_2 + b'_2 \\ a_{31}(x_1 + x'_1) + a_{32}(x_2 + x'_2) + a_{33}(x_3 + x'_3) + a_{34}(x_4 + x'_4) &= b_3 + b'_3 \end{aligned} \tag{A.9}$$

which will also follow from

$$\mathbf{Ax} + \mathbf{Ax}' = \mathbf{b} + \mathbf{b}'$$

if we define the addition of matrices by simple addition of the individual terms of the array. Clearly this can be done only if the size of the matrices is identical, i.e., for example,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \\ a_{31}+b_{31} & a_{32}+b_{32} \end{bmatrix}$$

or

$$\mathbf{A} + \mathbf{B} = \mathbf{C} \tag{A.10}$$

implies that every term of **C** is equal to the sum of the appropriate terms of **A** and **B**.

Subtraction obviously follows similar rules.

Transpose of a matrix

This is simply a definition for reordering the terms in an array in the following manner:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix} \tag{A.11}$$

and will be indicated by the symbol T as shown.

Its use is not immediately obvious but will be indicated later and can be treated here as a simple prescribed operation.

Inverse of a matrix

If in the relationship (A.2) the matrix **A** is 'square', i.e., it represents the coefficients of simultaneous equations of type (A.1) equal in number to the number of unknowns **x**, then in general it is possible to solve for the unknowns in terms of the known coefficients **b**. This solution can be written as

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{A.12}$$

in which the matrix $\mathbf{A}^{-1}$ is known as the 'inverse' of the square matrix **A**. Clearly $\mathbf{A}^{-1}$ is also square and of the same size as **A**.

We could obtain (A.12) by multiplying both sides of (A.2) by $\mathbf{A}^{-1}$ and hence

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} = \mathbf{A}\mathbf{A}^{-1} \tag{A.13}$$

where **I** is an 'identity' matrix having zero on all off-diagonal positions and unity on each of the diagonal positions.

If the equations are 'singular' and have no solution then clearly an inverse does not exist.

A sum of products

In problems of mechanics we often encounter a number of quantities such as force that can be listed as a matrix 'vector':

$$\mathbf{f} = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{Bmatrix} \tag{A.14}$$

These in turn are often associated with the same number of displacements given by another vector, say,

$$\mathbf{a} = \begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} \tag{A.15}$$

It is known that the work is represented as a sum of products of force and displacement

$$W = \sum_{k=1}^{n} f_k a_k$$

Clearly the transpose becomes useful here as we can write, by the rule of matrix multiplication,

$$W = \begin{bmatrix} f_1 & f_2 \cdots f_n \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} = \mathbf{f}^{\mathrm{T}}\mathbf{a} = \mathbf{a}^{\mathrm{T}}\mathbf{f} \tag{A.16}$$

Use of this fact is made frequently in this book.

Transpose of a product

An operation that sometimes occurs is that of taking the transpose of a matrix product. It can be left to the reader to prove from previous definitions that

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \tag{A.17}$$

Symmetric matrices

In structural problems symmetric matrices are often encountered. If a term of a matrix $\mathbf{A}$ is defined as a_{ij}, then for a symmetric matrix

$$a_{ij} = a_{ji} \qquad \text{or} \qquad \mathbf{A} = \mathbf{A}^{\mathrm{T}}$$

A symmetric matrix must be square. It can be shown that the inverse of a symmetric matrix is also symmetric

$$\mathbf{A}^{-1} = (\mathbf{A}^{-1})^{\mathrm{T}} \equiv \mathbf{A}^{-\mathrm{T}}$$

Partitioning

It is easy to verify that a matrix product **AB** in which for example

$$\mathbf{A} = \left[\begin{array}{ccc|cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{array}\right]$$

$$\mathbf{B} = \left[\begin{array}{cc} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ \hline b_{41} & b_{42} \\ b_{51} & b_{52} \end{array}\right]$$

could be obtained by dividing each matrix into submatrices, indicated by the lines, and applying the rules of matrix multiplication first to each of such submatrices as if it were a scalar number and then carrying out further multiplication in the usual way. Thus, if we write

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$$

then

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_1 & \mathbf{A}_{12}\mathbf{B}_2 \\ \mathbf{A}_{21}\mathbf{B}_1 & \mathbf{A}_{22}\mathbf{B}_2 \end{bmatrix}$$

can be verified as representing the complete product by further multiplication.

The essential feature of partitioning is that the size of subdivisions has to be such as to make the products of the type $\mathbf{A}_{11}\mathbf{B}_1$ meaningful, i.e., the number of columns in $\mathbf{A}_{11}$ must be equal to the number of rows in $\mathbf{B}_1$, etc. If the above definition holds, then all further operations can be conducted on partitioned matrices, treating each partition as if it were a scalar.

It should be noted that any matrix can be multiplied by a scalar (number). Here, obviously, the requirements of equality of appropriate rows and columns no longer apply.

If a symmetric matrix is divided into an equal number of submatrices $\mathbf{A}_{ij}$ in rows and columns then

$$\mathbf{A}_{ij} = \mathbf{A}_{ji}^{\mathrm{T}}$$

The eigenvalue problem

An *eigenvalue* of a symmetric matrix $\mathbf{A}$ of size $n \times n$ is a scalarλ_i, which allows the solution of

$$(\mathbf{A} - \lambda_i \mathbf{I})\boldsymbol{\phi}_i = \mathbf{0} \qquad \text{and} \qquad \det|\mathbf{A} - \lambda_i \mathbf{I}| = 0 \tag{A.18}$$

where $\boldsymbol{\phi}_i$ is called the *eigenvector*.

There are, of course, n such eigenvalues λ_i to each of which corresponds an eigenvector $\boldsymbol{\phi}_i$. Such vectors can be shown to be orthonormal and we write

$$\boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\phi}_j = \delta_{ij} = \begin{cases} 1 & \text{for} \quad i = j \\ 0 & \text{for} \quad i \neq j \end{cases}$$

The full set of eigenvalues and eigenvectors can be written as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \qquad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1, & \ldots & \boldsymbol{\phi}_n \end{bmatrix}$$

Using these the matrix $\mathbf{A}$ may be written in its *spectral form* by noting from the orthonormality conditions on the eigenvectors that

$$\boldsymbol{\Phi}^{-1} = \boldsymbol{\Phi}^{\mathrm{T}}$$

Then from

$$\mathbf{A}\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda}$$

it follows immediately that

$$\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{\mathrm{T}} \tag{A.19}$$

The condition number κ (which is related to equation solution roundoff) is defined as

$$\kappa = \frac{|\lambda_{\max}|}{|\lambda_{\min}|} \tag{A.20}$$

Appendix B

Tensor-indicial notation in the approximation of elasticity problems

Introduction

The matrix type of notation used in this volume for the description of tensor quantities such as stresses and strains is compact and we believe easy to understand. However, in a computer program each quantity will often still have to be identified by appropriate indices (see Chapter 20) and the conciseness of matrix notation does not always carry over to the programming steps. Further, many readers are accustomed to the use of indicial-tensor notation which is a standard tool in the study of solid mechanics. For this reason we summarize here the formulation of finite element arrays in an indicial form.

Some advantages of this reformulation from the matrix setting become apparent when evaluation of stiffness arrays for isotropic materials is considered. Here some multiplication operations previously necessary become redundant and the element module programs can be written more economically.

When finite deformation problems in solid mechanics have to be considered the use of indicial notation is almost essential to form many of the arrays needed for the residual and tangent terms.

This appendix adds little new to the discretization ideas – it merely repeats in a different language the results already presented.

Indicial notation: summation convention

A point P in three-dimensional space may be represented in terms of its cartesian coordinates x_a, $a = 1, 2, 3$. The limits that a can take define its *range*. To define these components we must first establish an oriented orthogonal set of coordinate directions as shown in Fig. B.1. The distance from the origin of the coordinate axes to the point define a position vector $\mathbf{x}$. If along each of the coordinate axes we define the set of unit orthonormal base vectors, $\mathbf{i}_a$, $a = 1, 2, 3$, which have the property

$$\mathbf{i}_a \cdot \mathbf{i}_b = \delta_{ab} = \begin{cases} 1 \text{ for } a = b \\ 0 \text{ for } a \neq b \end{cases} \tag{B.1}$$

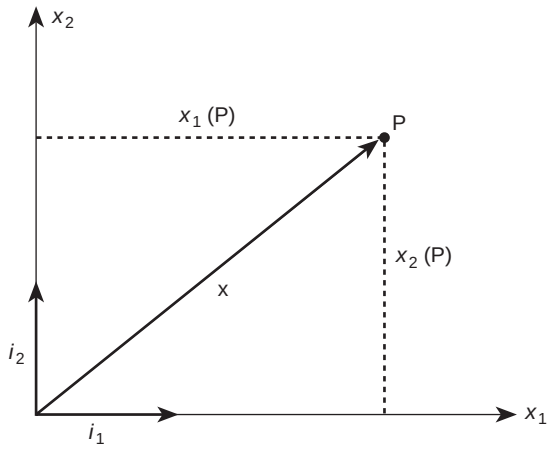


Fig. B.1 Orthogonal axes and a point: cartesian coordinates.

where $(\ )\cdot(\ )$ denotes the vector dot product, the components of the position vector are constructed from the vector dot product

$$x_a = \mathbf{i}_a \cdot \mathbf{x}; \qquad a = 1, 2, 3 \tag{B.2}$$

From this construction it is easy to observe that the vector $\mathbf{x}$ may be represented as

$$\mathbf{x} = \sum_{a=1}^{3} x_a \mathbf{i}_a \tag{B.3}$$

In dealing with vectors, and later tensors, the form $\mathbf{x}$ is called the *intrinsic* notation of the coordinates and $x_a\mathbf{i}_a$ the indicial form. An intrinsic form is a physical entity which is independent of the coordinate system selected, whereas an indicial form depends on a particular coordinate system.

To simplify notation we adopt the common convention that any index which is repeated in any given term implies a summation over the range of the index. Thus, our shorthand notation for Eq. (B.3) is

$$\mathbf{x} = x_a \mathbf{i}_a = x_1 \mathbf{i}_1 + x_2 \mathbf{i}_2 + x_3 \mathbf{i}_3 \tag{B.4}$$

For two-dimensional problems unless otherwise stated it will be understood that the range of the index is two.

Similarly, we can define the components of the displacement vector $\mathbf{u}$ as

$$\mathbf{u} = u_a \mathbf{i}_a \tag{B.5}$$

Note that the components (u_1, u_2, u_3) replace the components (u, v, w) used throughout most of this volume.

To avoid confusion with nodal quantities to which we previously also attached subscripts we shall simply change their position to a superscript. Thus

$$u_2^j \text{ has the same meaning as } v_j \tag{B.6}$$

used previously, etc.

Derivatives and tensorial relations

In indicial notation the derivative of any quantity with respect to a coordinate component x_a is written compactly as

$$\frac{\partial}{\partial x_a} \equiv (\quad)_{,a} \tag{B.7}$$

Thus we can write the *gradient* of the displacement vector as

$$\frac{\partial u_a}{\partial x_b} \equiv u_{a,b} \qquad a,b = 1,2,3 \tag{B.8}$$

In a cartesian coordinate system the base vectors do not change their magnitude or direction along any coordinate direction. Accordingly their derivatives with respect to any coordinate is zero as indicated in Eq. (B.9):

$$\frac{\partial \mathbf{i}_a}{\partial x_b} = \mathbf{i}_{a,b} = 0 \tag{B.9}$$

Thus in Cartesian co-ordinates, the derivative of the intrinsic displacement $\mathbf{u}$ is given by

$$\mathbf{u}_{,b} = u_{a,b}\mathbf{i}_a + u_a\mathbf{i}_{a,b} = u_{a,b}\mathbf{i}_a \tag{B.10}$$

The collection of all the derivatives defines the *displacement gradient* which we write in intrinsic notation as

$$\boldsymbol{\nabla}\mathbf{u} = u_{a,b}\mathbf{i}_a \otimes \mathbf{i}_b \tag{B.11}$$

The symbol $\otimes$ denotes the *tensor product* between two base vectors and since only two vectors are involved the gradient of the displacement is called *second rank*. The notation used to define a tensor product follows that used in reference 31.

Any second rank intrinsic quantity can be split into symmetric and skew symmetric (antisymmetric) parts as

$$\mathbf{A} = \tfrac{1}{2}[\mathbf{A} + \mathbf{A}^{\mathrm{T}}] + \tfrac{1}{2}[\mathbf{A} - \mathbf{A}^{\mathrm{T}}] = \mathbf{A}^{(s)} + \mathbf{A}^{(a)} \tag{B.12}$$

where $\mathbf{A}$ and its transpose have cartesian components

$$\mathbf{A} = A_{ab}\mathbf{i}_a \otimes \mathbf{i}_b; \qquad \mathbf{A}^{\mathrm{T}} = A_{ba}\mathbf{i}_a \otimes \mathbf{i}_b \tag{B.13}$$

The symmetric part of the displacement gradient defines the (small) strain†

$$\begin{aligned} \boldsymbol{\varepsilon} = \boldsymbol{\nabla}\mathbf{u}^{(s)} &= \tfrac{1}{2}\left[\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^{\mathrm{T}}\right] \\ &= \tfrac{1}{2}[u_{a,b} + u_{b,a}]\mathbf{i}_a \otimes \mathbf{i}_b \\ &= \varepsilon_{ab}\,\mathbf{i}_a \otimes \mathbf{i}_b = \varepsilon_{ba}\,\mathbf{i}_a \otimes \mathbf{i}_b \end{aligned} \tag{B.14}$$

and the skew symmetric part gives the (small) rotation

$$\begin{aligned} \boldsymbol{\omega} = \boldsymbol{\nabla}\mathbf{u}^{(a)} &= \tfrac{1}{2}\left[\boldsymbol{\nabla}\mathbf{u} - (\boldsymbol{\nabla}\mathbf{u})^{\mathrm{T}}\right] \\ &= \tfrac{1}{2}[u_{a,b} - u_{b,a}]\mathbf{i}_a \otimes \mathbf{i}_b \\ &= \omega_{ab}\,\mathbf{i}_a \otimes \mathbf{i}_b = -\,\omega_{ba}\,\mathbf{i}_a \otimes \mathbf{i}_b \end{aligned} \tag{B.15}$$

† Note that this definition is slightly different from that occurring in Chapters 2–6. Now $\varepsilon_{ab} = 1/2\gamma_{ab}$ when $i \neq j$.

The strain expression is analogous to Eq. (2.2). The components ε_{ab} and ω_{ab} may be represented by a matrix as

$$\varepsilon_{ab} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{13} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix} \tag{B.16}$$

$$\omega_{ab} = \begin{bmatrix} 0 & \omega_{12} & \omega_{13} \\ \omega_{21} & 0 & \omega_{23} \\ \omega_{31} & \omega_{32} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_{12} & \omega_{13} \\ -\omega_{12} & 0 & \omega_{23} \\ -\omega_{13} & -\omega_{23} & 0 \end{bmatrix} \tag{B.17}$$

Coordinate transformation

Consider now the representation of the intrinsic coordinates in a system which has different orientation than that given in Fig. B.1. We represent the components in the new system by

$$\mathbf{x} = x'_{a'}\,\mathbf{i}'_{a'} \tag{B.18}$$

Using Eq. (B.2) we can relate the components in the prime system to those in the original system as

$$\begin{aligned} x'_{a'} &= \mathbf{i}'_a \cdot \mathbf{x} \\ &= \mathbf{i}'_a \cdot \mathbf{i}_b x_b \\ &= \Lambda_{a'b} x_b \end{aligned} \tag{B.19}$$

where

$$\Lambda_{a'b} = \mathbf{i}'_{a'} \cdot \mathbf{i}_b = \cos(x'_{a'}, x_b) \tag{B.20}$$

define the direction cosines of the coordinate in a manner similar to that of Eq. (1.25).

Equation (B.19) defines how the cartesian coordinate components transform from one coordinate frame to another. Recall that the summation convention implies

$$x'_{a'} = \Lambda_{a'1} x_1 + \Lambda_{a'2} x_2 + \Lambda_{a'3} x_3 \qquad a' = 1, 2, 3 \tag{B.21}$$

In Eq. (B.19) a' is called a *free index* whereas b is called a *dummy index* since it may be replaced by any other unique index without changing the meaning of the term (note that the notation does not permit an index to appear more than twice in any term). The summation convention will be employed throughout the remainder of this discussion and the reader should ensure that the concept is fully understood before proceeding. Some examples will be given occasionally to illustrate its use.

Using the notion of the direction cosines, Eq. (B.19) may be used to transform any vector with three components. Thus, transformation of the components of the displacement vector is given by

$$u'_{a'} = \Lambda_{a'b} u_b \qquad a'b = 1, 2, 3 \tag{B.22}$$

Indeed we can also use the above to express the transformation for the base vectors since

$$\mathbf{i}'_{a'} = \left(\mathbf{i}'_{a'} \cdot \mathbf{i}_b\right)\mathbf{i}_b = \Lambda_{a'b}\mathbf{i}_b \tag{B.23}$$

Similarly, by interchanging the role of the base vectors we obtain

$$\mathbf{i}_b = \left(\mathbf{i}_b \cdot \mathbf{i}'_{a'}\right)\mathbf{i}'_{a'} = \Lambda_{a'b}\mathbf{i}'_{a'} \tag{B.24}$$

which indicates that the *inverse* of the direction cosine coefficient array is the same as its *transpose*.

The strain transformation follows from the intrinsic form written as

$$\begin{aligned}\boldsymbol{\varepsilon} &= \varepsilon'_{a'b'}\mathbf{i}_{a'} \otimes \mathbf{i}_{b'} \\ &= \varepsilon_{cd}\mathbf{i}_c \otimes \mathbf{i}_d\end{aligned} \tag{B.25}$$

Substitution of the base vectors from Eq. (B.24) into Eq. (B.25) gives

$$\boldsymbol{\varepsilon} = \Lambda_{a'c}\varepsilon_{cd}\Lambda_{b'd}\mathbf{i}_{a'} \otimes \mathbf{i}_{b'} \tag{B.26}$$

Comparing Eq. (B.26) with Eq. (B.25) the components of the strain transform according to the relation

$$\varepsilon'_{a'b'} = \Lambda_{a'c}\varepsilon_{cd}\Lambda_{b'd} \tag{B.27}$$

Variables that transform according to Eq. (B.22) are called *first rank cartesian tensors* whereas quantities that transform according to Eq. (B.27) are called *second rank cartesian tensors*. The use of indicial notation in the context of cartesian coordinates will lead naturally to each mechanics variable being defined in terms of a cartesian tensor of appropriate rank.

Stress may be written in terms of its components σ_{ab} which may be written in matrix form similar to Eq. (B.16)

$$\sigma_{ab} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \qquad a, b = 1, 2, 3 \tag{B.28}$$

In intrinsic form the stress is given by

$$\boldsymbol{\sigma} = \sigma_{ab}\mathbf{i}_a \otimes \mathbf{i}_b \tag{B.29}$$

and, using similar logic as was used for strain, can be shown to transform as a second rank cartesian tensor. The symmetry of the components of stress may be established by summing moments (angular momentum balance) about each of the coordinate axes to obtain

$$\sigma_{ab} = \sigma_{ba} \tag{B.30}$$

Equilibrium and energy

Introducing a body force vector

$$\mathbf{b} = b_a\mathbf{i}_a \tag{B.31}$$

we can write the static equilibrium equations (linear momentum balance) for a differential element as

$$\operatorname{div}\boldsymbol{\sigma}+\mathbf{b}\equiv\left(\sigma_{ba,b}+b_a\right)\mathbf{i}_a=\mathbf{0} \tag{B.32}$$

where the repeated index again implies summation over the range of the index, i.e.,

$$\begin{aligned}\sigma_{ba,b}&\equiv\sum_{b=1}^{3}\sigma_{ba,b}\\&=\sigma_{1a,1}+\sigma_{2a,2}+\sigma_{3a,3}\end{aligned}$$

Note that the free index a must appear in each term for the equation to be meaningful.

As a further example of the summation convention consider an internal energy term

$$W=\sigma_{ab}\varepsilon_{ab} \tag{B.33}$$

This expression implies a double summation; hence summing first on a gives

$$W=\sigma_{1b}\varepsilon_{1b}+\sigma_{2b}\varepsilon_{2b}+\sigma_{3b}\varepsilon_{3b}$$

and then summing on b gives finally

$$\begin{aligned}W=&\sigma_{11}\varepsilon_{11}+\sigma_{12}\varepsilon_{12}+\sigma_{13}\varepsilon_{13}\\&+\sigma_{21}\varepsilon_{21}+\sigma_{22}\varepsilon_{22}+\sigma_{23}\varepsilon_{23}\\&+\sigma_{31}\varepsilon_{31}+\sigma_{32}\varepsilon_{32}+\sigma_{33}\varepsilon_{33}\end{aligned}$$

We may use symmetry conditions on σ_{ab} and ε_{ab} to reduce the nine terms to six terms. Accordingly,

$$\begin{aligned}W=&\sigma_{11}\varepsilon_{11}+\sigma_{22}\varepsilon_{22}+\sigma_{33}\varepsilon_{33}\\&+2(\sigma_{12}\varepsilon_{12}+\sigma_{23}\varepsilon_{23}+\sigma_{31}\varepsilon_{31})\end{aligned} \tag{B.34}$$

Following a similar expansion we can also show the result

$$\sigma_{ab}\omega_{ab}\equiv 0 \tag{B.35}$$

Elastic constitutive equations

For an elastic material the most general linear relationship we can write for components of the stress–strain characterization is

$$\sigma_{ab}=D_{abcd}\left(\varepsilon_{cd}-\varepsilon^0_{cd}\right)+\sigma^0_{ab} \tag{B.36}$$

Equation (B.36) is the equivalent of Eq. (2.5) but now written in index notation. We note that the elastic moduli which appear in Eq. (B.36) are components of the fourth rank tensor

$$\mathbf{D}=D_{abcd}\mathbf{i}_a\otimes\mathbf{i}_b\otimes\mathbf{i}_c\otimes\mathbf{i}_d \tag{B.37}$$

The elastic moduli possess the following symmetry conditions

$$D_{abcd}=D_{bacd}=D_{abdc}=D_{cdab} \tag{B.38}$$

the latter, arising from the existence of an internal energy density in the form[2]

$$W(\boldsymbol{\varepsilon})=\tfrac{1}{2}\varepsilon_{ab}D_{abcd}\varepsilon_{cd}+\varepsilon_{ab}\left[\sigma^0_{ab}-D_{abcd}\varepsilon^0_{cd}\right] \tag{B.39}$$

which yields the stress from

$$\sigma_{ab} = \frac{\partial W}{\partial \varepsilon_{ab}} \tag{B.40}$$

By writing the constitutive equation with respect to $x'_{a'}$ and using properties of the base vectors we can deduce the transformation equation for moduli as

$$D'_{a'b'c'd'} = \Lambda_{a'e}\Lambda_{b'f}\,\Lambda_{c'g}\Lambda_{d'h}D_{efgh} \tag{B.41}$$

A common notation for the intrinsic form of Eq. (B.36) is

$$\boldsymbol{\sigma} = \mathbf{D} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^0) + \boldsymbol{\sigma}^0 \tag{B.42}$$

in which : denotes the double summation (contraction) between the elastic moduli and the strains.

The elastic moduli for an isotropic elastic material may be written in indicial form as

$$D_{abcd} = \lambda\delta_{ab}\delta_{cd} + \mu(\delta_{ac}\delta_{bd} + \delta_{ad}\delta_{bc}) \tag{B.43}$$

where λ, μ are the Lamé constants. An isotropic linear elastic material is always characterized by two independent elastic constants. Instead of the Lamé constants we can use Young's modulus, E, and Poisson's ratio, ν, to characterize the material. The Lamé constants may be deduced from

$$\mu = \frac{E}{2(1+\nu)} \qquad \text{and} \qquad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{B.44}$$

Finite element approximation

If we now introduce the finite element displacement approximation given by Eq. (2.1), using indicial notation we may write for a single element

$$u_a \approx \hat{u}_a = N_i\tilde{u}^i_a \qquad a = 1,2,3; \qquad i = 1,2,\ldots,n \tag{B.45}$$

where n is the total number of nodes on an element. The strain approximation in each element is given by the definition of Eq. (B.14) as

$$\hat{\varepsilon}_{ab} = \tfrac{1}{2}\left[N_{i,b}\tilde{u}^i_a + N_{i,a}\tilde{u}^i_b\right] \tag{B.46}$$

The internal virtual work for an element is given as

$$\delta U^I = \int_{V_e} \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma}\,\mathrm{d}V = \int_{V_e} \delta\varepsilon_{ab}\sigma_{ab}\,\mathrm{d}V \tag{B.47}$$

Using Eqs (B.46) and (B.47) and noting the symmetries in D_{abcd} we may write the internal virtual work for a linear elastic material as

$$\begin{aligned}\delta U^I = {} & \delta\tilde{u}^i_a \int_{V_e} N_{i,b}D_{abcd}N_{j,d}\,\mathrm{d}V\tilde{u}^j_c \\ & + \delta\tilde{u}^i_a \int_{V_e} N_{i,b}\left(\sigma^0_{ab} - D_{abcd}\varepsilon^0_{cd}\right)\mathrm{d}V\end{aligned} \tag{B.48}$$

which replaces the terms obtained in Chapter 2 in indicial notation.

In describing a stiffness coefficient two subscripts have been used previously and the submatrix $\mathbf{K}_{ij}$ implied 2×2 or 3×3 entries for the ij nodal pair, depending on

whether two or three dimensional displacement components were involved. Now the scalar components

$$K_{ab}^{ij} \qquad a,b = 1,2,3; \qquad i,j = 1,2,\ldots,n \tag{B.49}$$

define completely the appropriate stiffness coefficient with *ab* indicating the relative submatrix position (in this case for a three-dimensional displacement).

Note that for a symmetric matrix we have previously required that

$$\mathbf{K}_{ij} = \mathbf{K}_{ji}^{\mathrm{T}} \tag{B.50}$$

In indicial notation the same symmetry is implied if

$$K_{ab}^{ij} = K_{ba}^{ji} \tag{B.51}$$

The stiffness tensor is now defined from Eq. (B.48) as

$$K_{ac}^{ij} = \int_{V_e} N_{i,b} D_{abcd} N_{j,d} \, \mathrm{d}V \tag{B.52}$$

When the elastic properties are constant over the element we may separate the integration from the material constants by defining

$$W_{bd}^{ij} = \int_{V_e} N_{i,b} N_{j,d} \, \mathrm{d}V \tag{B.53}$$

and then perform the summations with the material moduli as

$$K_{ac}^{ij} = W_{bd}^{ij} D_{abcd} \tag{B.54}$$

In the case of isotropy a particularly simple result is obtained

$$K_{ac}^{ij} = \lambda W_{ac}^{ij} + \mu [W_{ca}^{ij} + \delta_{ac} W_{bb}^{ij}] \tag{B.55}$$

which allows the construction of the stiffness to be carried out using fewer arithmetic operations as compared with the use of the matrix form.[3]

Using indicial notation the final equilibrium equations of the system are written as

$$K_{ac}^{ij} u_c^j + f_a^i = 0 \qquad a = 1,2,3 \tag{B.56}$$

and in this scalar form every coefficient is simply identified. The reader can, as a simple exercise, complete the derivation of the force terms due to the initial strain ε_{ab}^0, stress σ_{ab}^0,body force b_a and external traction $\bar{t}_a$.

Indicial notation is at times useful in clarifying individual terms, and this introduction should be helpful as a key to reading some of the current literature.

Relation between indicial and matrix notation

The matrix form used throughout most of this volume can be deduced from the indicial form by a simple transformation between the indices. The relationship between the indices of the second rank tensors and their corresponding matrix form can be performed by an inspection of the ordering in the matrix for stress and its representation shown in Eq. (B.28). In the matrix form the stress was given

in Chapter 6 as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{23} & \sigma_{31} \end{bmatrix}^{\mathrm{T}} \tag{B.57}$$

This form includes the use of the symmetry of stress components. The mapping of the indices follows that shown in Table B.1.

Table B.1 Mapping between matrix and tensor indices for second rank symmetric tensors

Form	Index number					
Matrix	1	2	3	4	5	6
Tensor	11 *xx*	22 *yy*	33 *zz*	12 & 21 *xy* & *yx*	23 & 32 *yz* & *zy*	31 & 13 *zx* & *xz*

Table B.1 may also be used to perform the map of the material moduli by noting that the components in the energy are associated with the index pairs from the stress and the strain. Accordingly, the moduli transform as

$$D_{1111} \rightarrow D_{11}; \qquad D_{2233} \rightarrow D_{23}; \qquad D_{1231} \rightarrow D_{46}; \quad \text{etc.} \tag{B.58}$$

The symmetry of the stress and strain is embedded in Table B.1 and the existence of an energy function yields the symmetry of the modulus matrix, i.e., $D_{ij} = D_{ji}$.

References

1. P. Chadwick. *Continuum Mechanics*. John Wiley & Sons, New York, 1976.
2. I.S. Sokolnikoff. *The Mathematical Theory of Elasticity*. McGraw-Hill, New York, 2nd edition, 1956.
3. A.K. Gupta and B. Mohraz. A method of computing numerically integrated stiffness matrices. *Internat. J. Num. Meth. Eng.*, **5**, 83–89, 1972.

Appendix C

Basic equations of displacement analysis (Chapter 2)

Displacement

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum \mathbf{N}_i \mathbf{a}_i = \mathbf{N}\mathbf{a} \tag{C.1}$$

Strain

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u} \tag{C.2}$$

$$\boldsymbol{\varepsilon} = \sum \mathbf{B}_i \mathbf{a}_i = \mathbf{B}\mathbf{a} \tag{C.3}$$

$$\mathbf{B}_i = \mathbf{S}\mathbf{N}_i \tag{C.4}$$

$$\mathbf{B} = \mathbf{S}\mathbf{N}$$

Stress–strain constitutive relation of linear elasticity

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \tag{C.5}$$

Approximate equilibrium equations

$$\mathbf{r} = \mathbf{K}\mathbf{a} + \mathbf{f} \tag{C.6}$$

$$\mathbf{K}_{ij} = \int_V \mathbf{B}_i^{\mathrm{T}} \mathbf{D} \mathbf{B}_j \, \mathrm{d}V \tag{C.7}$$

$$\mathbf{f}_i = -\int_V \mathbf{N}_i^{\mathrm{T}} \mathbf{b} \, \mathrm{d}V - \int_A \mathbf{N}_i^{\mathrm{T}} \bar{\mathbf{t}} \, \mathrm{d}A - \int_V \mathbf{B}_i^{\mathrm{T}} (\mathbf{D}\boldsymbol{\varepsilon}_0 + \boldsymbol{\sigma}_0) \, \mathrm{d}V$$

Appendix D

Some integration formulae for a triangle

Let a triangle be defined in the xy plane by three points (x_i, y_i), (x_j, y_j), (x_m, y_m) with the origin of the coordinates taken at the centroid (or baricentre), i.e.,

$$\frac{x_i + x_j + x_m}{3} = \frac{y_i + y_j + y_m}{3} = 0$$

Then integrating over the triangle area we obtain:

$$\int \mathrm{d}x\,\mathrm{d}y = \frac{1}{2}\begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} = \Delta = \text{area of triangle}$$

$$\int x\,\mathrm{d}x\,\mathrm{d}y = \int y\,\mathrm{d}x\,\mathrm{d}y = 0$$

$$\int x^2\,\mathrm{d}x\,\mathrm{d}y = \frac{\Delta}{12}(x_i^2 + x_j^2 + x_m^2)$$

$$\int y^2\,\mathrm{d}x\,\mathrm{d}y = \frac{\Delta}{12}(y_i^2 + y_j^2 + y_m^2)$$

$$\int xy\,\mathrm{d}x\,\mathrm{d}y = \frac{\Delta}{12}(x_i y_i + x_j y_j + x_m y_m)$$

Appendix E

Some integration formulae for a tetrahedron

Let a tetrahedron be defined in the xyz coordinate system by four points (x_i, y_i, z_i), (x_j, y_j, z_j), (x_m, y_m, z_m), (x_p, y_p, z_p) with the origin of the coordinates taken at the centroid, i.e.,

$$\frac{x_i + x_j + x_m + x_p}{4} = \frac{y_i + y_j + y_m + y_p}{4} = \frac{z_i + z_j + z_m + z_p}{4} = 0$$

Then integrating over the tetrahedron volume gives

$$\int \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{1}{6}\begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_m & y_m & z_m \\ 1 & x_p & y_p & z_p \end{vmatrix} = V = \text{tetrahedron volume}$$

Provided the order of numbering the nodes is as indicated on Fig. 6.1 then also:

$$\int x\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \int y\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \int z\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = 0$$

$$\int x^2\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{V}{20}(x_i^2 + x_j^2 + x_m^2 + x_p^2)$$

$$\int y^2\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{V}{20}(y_i^2 + y_j^2 + y_m^2 + y_p^2)$$

$$\int z^2\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{V}{20}(z_i^2 + z_j^2 + z_m^2 + z_p^2)$$

$$\int xy\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{V}{20}(x_i y_i + x_j y_j + x_m y_m + x_p y_p)$$

$$\int yz\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{V}{20}(y_i z_i + y_j z_j + y_m z_m + y_p z_p)$$

$$\int zx\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{V}{20}(z_i x_i + z_j x_j + z_m x_m + z_p x_p)$$

Appendix F

Some vector algebra

Some knowledge and understanding of basic vector algebra is needed in dealing with complexities of elements oriented in space as occur in beams, shells, etc. Some of the operations are summarized here.

Vectors (in the geometric sense) can be described by their components along the directions of the x, y, z axes.

Thus, the vector $\mathbf{V}_{01}$ shown in Fig. F.1 can be written as

$$\mathbf{V}_{01} = x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k} \tag{F.1}$$

in which $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are unit vectors in the direction of the x, y, z axes.

Alternatively, the same vector could be written as

$$\mathbf{V}_{01} = \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} \tag{F.2}$$

(now a 'vector' in the matrix sense) in which the components are distinguished by positions in the column.

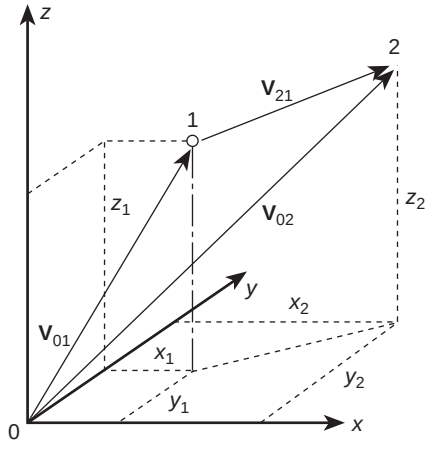


Fig. F.1 Vector addition.

Addition and subtraction

Addition and subtraction is defined by addition and subtraction of components. Thus, for example,

$$\mathbf{V}_{02} - \mathbf{V}_{01} = (x_2 - x_1)\mathbf{i} + (y_2 - y_1)\mathbf{j} + (z_2 - z_1)\mathbf{k} \tag{F.3}$$

The same result is achieved by the definitions of matrix algebra; thus

$$\mathbf{V}_{02} - \mathbf{V}_{01} = \mathbf{V}_{21} = \left\{ \begin{array}{c} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{array} \right\} \tag{F.4}$$

'Scalar' products

The scalar product of two vectors is *defined* as

$$\mathbf{A} \cdot \mathbf{B} = \mathbf{B} \cdot \mathbf{A} = \sum_{k=1}^{3} a_k b_k \tag{F.5}$$

If

$$\begin{aligned} \mathbf{A} &= a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k} \\ \mathbf{B} &= b_x\mathbf{i} + b_y\mathbf{j} + b_z\mathbf{k} \end{aligned} \tag{F.6}$$

then

$$\mathbf{A} \cdot \mathbf{B} = a_x b_x + a_y b_y + a_z b_z \tag{F.7}$$

Using the matrix notation

$$\mathbf{A} = \left\{ \begin{array}{c} a_x \\ a_y \\ a_z \end{array} \right\} \qquad \mathbf{B} = \left\{ \begin{array}{c} b_x \\ b_y \\ b_z \end{array} \right\} \tag{F.8}$$

the scalar product becomes

$$\mathbf{A} \cdot \mathbf{B} = \mathbf{A}^{\mathrm{T}}\mathbf{B} = \mathbf{B}^{\mathrm{T}}\mathbf{A} \tag{F.9}$$

Length of vector

The length of the vector $\mathbf{V}_{21}$ is given, purely geometrically, as

$$l_{21} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{F.10}$$

or in terms of matrix algebra as

$$l_{21} = \sqrt{\mathbf{V}_{21} \cdot \mathbf{V}_{21}} = \sqrt{\mathbf{V}_{21}^{\mathrm{T}}\mathbf{V}_{21}} \tag{F.11}$$

Direction cosines

Direction cosines of a vector are simply given from the definition of the projected component of lengths as (Fig. F.1)

$$\cos\alpha_x = \Lambda_{vx} = \frac{x_2 - x_1}{l_{21}} = \frac{\mathbf{V}\cdot\mathbf{i}}{l_{21}}, \quad \text{etc.} \tag{F.12}$$

The scalar product may also be written as (Fig. F.2)

$$\mathbf{A}\cdot\mathbf{B} = \mathbf{B}\cdot\mathbf{A} = l_a l_b \cos\gamma \tag{F.13}$$

where γ is the angle between the two vectors **A** and **B** and l_a and l_b are their lengths, respectively.

'Vector' or cross product

Another product of vectors is defined as a vector oriented normally to the plane given by two vectors and equal in magnitude to the product of the length of the two vectors multiplied by the sine of the angle between them. Further, the direction of the normal vector follows the right-hand rule as shown in Fig. F.2 in which

$$\mathbf{A}\times\mathbf{B} = \mathbf{C} \tag{F.14}$$

is shown.

Thus, from the right-hand rule, we have

$$\mathbf{A}\times\mathbf{B} = -\mathbf{B}\times\mathbf{A} \tag{F.15}$$

It is worth noting that the magnitude (or length) of **C** is equal to the area of the parallelogram shown in Fig. F.2.

Using the definition of Eq. (F.6) and noting that

$$\begin{aligned} &\mathbf{i}\times\mathbf{i} = \mathbf{j}\times\mathbf{j} = \mathbf{k}\times\mathbf{k} = 0 \\ &\mathbf{i}\times\mathbf{j} = \mathbf{k} \qquad \mathbf{j}\times\mathbf{k} = \mathbf{i} \qquad \mathbf{k}\times\mathbf{i} = \mathbf{j} \end{aligned} \tag{F.16}$$

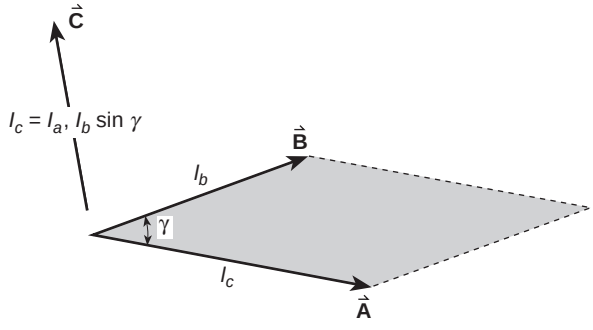


Fig. F.2 Vector multiplication (cross product).

we have

$$\mathbf{A}\times\mathbf{B}=\det\begin{vmatrix}\mathbf{i}&\mathbf{j}&\mathbf{k}\\ a_x&a_y&a_z\\ b_x&b_y&b_z\end{vmatrix}$$
$$=(a_yb_z-a_zb_y)\mathbf{i}+(a_zb_x-a_xb_z)\mathbf{j}+(a_xb_y-a_yb_x)\mathbf{k}$$

There is no simple counterpart in matrix algebra but we can use the above to define the vector $\mathbf{C}$.†

$$\mathbf{C}=\mathbf{A}\times\mathbf{B}=\begin{Bmatrix}a_yb_z-a_zb_y\\ a_zb_x-a_xb_z\\ a_xb_y-a_yb_x\end{Bmatrix} \tag{F.17}$$

The vector product will be found particularly useful when the problem of erecting a normal direction to a surface is considered.

Elements of area and volume

If ξ and η are curvilinear coordinates, then the following vectors in the two-dimensional plane

$$\mathrm{d}\boldsymbol{\xi}=\begin{Bmatrix}\dfrac{\partial x}{\partial\xi}\\ \dfrac{\partial y}{\partial\xi}\end{Bmatrix}\mathrm{d}\xi\qquad \mathrm{d}\boldsymbol{\eta}=\begin{Bmatrix}\dfrac{\partial x}{\partial\eta}\\ \dfrac{\partial y}{\partial\eta}\end{Bmatrix}\mathrm{d}\eta \tag{F.18}$$

defined from the relationship between the cartesian and curvilinear coordinates, are vectors directed tangentially to the ξ and η equal-constant contours, respectively. As the *length* of the vector resulting from a cross product of $\mathrm{d}\boldsymbol{\xi}\times\mathrm{d}\boldsymbol{\eta}$ is equal to the area of the elementary parallelogram we can write

$$\mathrm{d}(\mathit{area})=\det\begin{vmatrix}\dfrac{\partial x}{\partial\xi}&\dfrac{\partial x}{\partial\eta}\\ \dfrac{\partial y}{\partial\xi}&\dfrac{\partial y}{\partial\eta}\end{vmatrix}\mathrm{d}\xi\,\mathrm{d}\eta \tag{F.19}$$

by Eq. (F.17).

† If we rewrite $\mathbf{A}$ as a skew symmetric matrix

$$\hat{\mathbf{A}}=\begin{bmatrix}0&-a_z&a_y\\ a_z&0&-a_x\\ -a_y&a_x&0\end{bmatrix}$$

then an alternative representation of the vector product in matrix form is $\mathbf{C}=\hat{\mathbf{A}}\mathbf{B}$.

Similarly, if we have three curvilinear coordinates ξ, η, ζ in cartesian space, the 'triple' or box product defines a differential volume

$$\mathrm{d}(vol) = (\mathrm{d}\boldsymbol{\xi} \times \mathrm{d}\boldsymbol{\eta}) \cdot \mathrm{d}\boldsymbol{\zeta} = \det \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} & \dfrac{\partial x}{\partial \zeta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial y}{\partial \zeta} \\ \dfrac{\partial z}{\partial \xi} & \dfrac{\partial z}{\partial \eta} & \dfrac{\partial z}{\partial \zeta} \end{vmatrix} \mathrm{d}\xi \, \mathrm{d}\eta \, \mathrm{d}\zeta \tag{F.20}$$

This follows simply from the geometry. The bracketed product, by definition,forms a vector whose length is equal to the parallelogram area with sides tangent to two of the coordinates. The second scalar multiplication by a length and the cosine of the angle between that length and the normal to the parallelogram establishes a differential volume element.

The above equations serve in changing the variables in surface and volume integrals.

Appendix G

Integration by parts in two or three dimensions (Green's theorem)

Consider the integration by parts of the following two-dimensional expression

$$\iint_{\Omega} \phi \frac{\partial \psi}{\partial x} \, \mathrm{d}x \, \mathrm{d}y \tag{G.1}$$

Integrating first with respect to x and using the well-known relation for integration by parts in one-dimension

$$\int_{x_L}^{x_R} u \, \mathrm{d}v = -\int_{x_L}^{x_R} v \, \mathrm{d}u + (uv)_{x=x_R} - (uv)_{x=x_L} \tag{G.2}$$

we have, using the symbols of Fig. G.1

$$\iint_{\Omega} \phi \frac{\partial \psi}{\partial x} \, \mathrm{d}x \, \mathrm{d}y = -\iint_{\Omega} \frac{\partial \phi}{\partial x} \psi \, \mathrm{d}x \, \mathrm{d}y + \int_{y_B}^{y_T} \left[(\phi \psi)_{x=x_R} - (\phi \psi)_{x=x_L} \right] \mathrm{d}y \tag{G.3}$$

If we now consider a direct segment of the boundary $\mathrm{d}\Gamma$ on the right-hand boundary, we note that

$$\mathrm{d}y = n_x \, \mathrm{d}\Gamma \tag{G.4}$$

where n_x is the direction cosine between the outward normal and the x direction. Similarly on the left-hand section we have

$$\mathrm{d}y = -n_x \, \mathrm{d}\Gamma \tag{G.5}$$

The final term of Eq. (G.3) can thus be expressed as the integral taken around an anticlockwise direction of the complete closed boundary:

$$\oint_{\Gamma} \phi \psi n_x \, \mathrm{d}\Gamma \tag{G.6}$$

If several closed contours are encountered this integration has to be taken around each such countour. The general expression in all cases is

$$\iint_{\Omega} \phi \frac{\partial \psi}{\partial x} \, \mathrm{d}x \, \mathrm{d}y = -\iint_{\Omega} \frac{\partial \phi}{\partial x} \psi \, \mathrm{d}x \, \mathrm{d}y + \oint_{\Gamma} \phi \psi n_x \, \mathrm{d}\Gamma \tag{G.7}$$

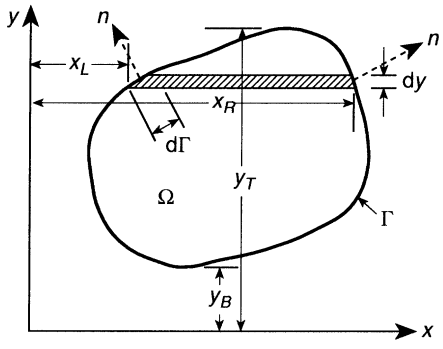


Fig. G.1 Definitions for integrations in two-dimensions.

Similarly, if differentiation in the y direction arises we can write

$$\iint_{\Omega} \phi \frac{\partial \psi}{\partial y} \, \mathrm{d}x \, \mathrm{d}y = - \iint_{\Omega} \frac{\partial \phi}{\partial y} \psi \, \mathrm{d}x \, \mathrm{d}y + \oint_{\Gamma} \phi \psi n_y \, \mathrm{d}\Gamma \tag{G.8}$$

where n_y is the direction cosine between the outward normal and the y axis.

In three dimensions by an identical procedure we can write

$$\iiint_{\Omega} \phi \frac{\partial \psi}{\partial y} \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z = - \iiint_{\Omega} \frac{\partial \phi}{\partial y} \psi \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z + \oint_{\Gamma} \phi \psi n_y \, \mathrm{d}\Gamma \tag{G.9}$$

where $\mathrm{d}\Gamma$ becomes the element of the surface area and the last integral is taken over the whole surface. A similar expression holds for derivatives like Eq. (G.9) in x and z.

Appendix H

Solutions exact at nodes

The finite element solution of ordinary differential equations may be made exact at the interelement nodes by a proper choice of the *weighting* function in the weak (Galerkin) form. To be more specific, let us consider the set of ordinary differential equations given by

$$\mathbf{A}(\mathbf{u}) + \mathbf{f}(x) = \mathbf{0} \tag{H.1}$$

where $\mathbf{u}$ is the set of dependent variables which are functions of the single independent variable 'x' and $\mathbf{f}$ is a vector of specified load functions. The weak form of this set of differential equations is given by

$$\int_{x_L}^{x_R} \mathbf{v}^{\mathrm{T}} [\mathbf{A}(\mathbf{u}) + \mathbf{f}] \, \mathrm{d}x = 0 \tag{H.2}$$

The weak form may be integrated by parts to remove all the derivatives from $\mathbf{u}$ and place them on $\mathbf{v}$. The result of this step may be expressed as

$$\int_{x_L}^{x_R} [\mathbf{u}^{\mathrm{T}} \mathbf{A}^*(\mathbf{v}) + \mathbf{v}^{\mathrm{T}} \mathbf{f}] \, \mathrm{d}x + [\mathbf{B}^*(\mathbf{v})]^{\mathrm{T}} \mathbf{B}(\mathbf{u}) \Big|_{x_L}^{x_R} = 0 \tag{H.3}$$

where $\mathbf{A}^*(\mathbf{v})$ is the *adjoint differential equation* and $\mathbf{B}^*(\mathbf{v})$ and $\mathbf{B}(\mathbf{u})$ are terms on the boundary resulting from integration by parts.

If we can find the general integral to the homogeneous adjoint differential equation

$$\mathbf{A}^*(\mathbf{v}) = \mathbf{0} \tag{H.4}$$

then the weak form of the problem reduces to

$$\int_{x_L}^{x_R} \mathbf{v}^{\mathrm{T}} \mathbf{f} \, \mathrm{d}x + [\mathbf{B}^*(\mathbf{v})]^{\mathrm{T}} \mathbf{B}(\mathbf{u}) \Big|_{x_L}^{x_R} = 0 \tag{H.5}$$

The first term is merely an expression to generate equivalent forces from the solution to the adjoint equation and the last term is used to construct the residual equation for the problem. If the differential equation is linear these lead to a residual which depends linearly on the values of $\mathbf{u}$ at the ends x_L and x_R. If we now let these be the location of the end nodes of a typical element we immediately find an expression to generate a stiffness matrix. Since in this process we have never had to construct an *approximation* for the dependent variables $\mathbf{u}$ it is immediately evident that at the end

points the discrete values of the exact solution must coincide with any admissible approximation we choose. Thus, we always obtain exact solutions at these points.

If we consider that all values of the forcing function are contained in f (i.e., no point loads at nodes), the terms in $\mathbf{B}(\mathbf{u})$ must be continuous between adjacent elements. At the boundaries the terms in $\mathbf{B}(\mathbf{u})$ include a flux term as well as displacements.

As an example problem, consider the single differential equation

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + P\frac{\mathrm{d}u}{\mathrm{d}x} + f = 0 \tag{H.6}$$

with the associated weak form

$$\int_{x_L}^{x_R} v\left[\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + P\frac{\mathrm{d}u}{\mathrm{d}x} + f\right]\mathrm{d}x = 0 \tag{H.7}$$

After integration by parts the weak form becomes

$$\int_{x_L}^{x_R}\left[u\left(\frac{\mathrm{d}^2v}{\mathrm{d}x^2} - P\frac{\mathrm{d}v}{\mathrm{d}x}\right) + vf\right]\mathrm{d}x + \left[v\left(\frac{\mathrm{d}u}{\mathrm{d}x} + Pu\right) - \frac{\mathrm{d}v}{\mathrm{d}x}u\right]_{x_L}^{x_R} = 0 \tag{H.8}$$

The adjoint differential equation is given by

$$a^*(v) = \frac{\mathrm{d}^2v}{\mathrm{d}x^2} - P\frac{\mathrm{d}v}{\mathrm{d}x} = 0 \tag{H.9}$$

and the boundary terms by

$$\mathbf{B}^*(v) = \begin{Bmatrix} v \\ -\dfrac{\mathrm{d}v}{\mathrm{d}x} \end{Bmatrix} \tag{H.10}$$

and

$$\mathbf{B}(u) = \begin{Bmatrix} \dfrac{\mathrm{d}u}{\mathrm{d}x} + Pu \\ u \end{Bmatrix} \tag{H.11}$$

For the above example two cases may be identified:

1. P zero, where the adjoint differential equation is identical to the homogeneous equation in which case the problem is called *self-adjoint*.
2. P non-zero, where we then have the *non-self-adjoint* problem.

The finite element solution for these two cases is often quite different. In the first case an equivalent variational theorem exists, whereas for the second case no such theorem exists.†

In the first case the solution to the adjoint equation is given by

$$v = Ax + B \tag{H.12}$$

which may be written as conventional linear shape functions in each element as

$$N_L = \frac{x_R - x}{x_R - x_L} \qquad N_R = \frac{x - x_L}{x_R - x_L} \tag{H.13}$$

† An integrating factor may often be introduced to make the weak form generate a self-adjoint problem; however, the approximation problem will remain the same. See Sec. 3.9.2.

Thus, for linear shape functions in each element used as the weighting function the interelement nodal displacements for u will always be exact (e.g., see Fig. 3.4) irrespective of the interpolation used for u.

For the second case the exact solution to the adjoint equation is

$$v = Ae^{Px} + B = Az + B \tag{H.14}$$

This yields the shape functions for the weighting function

$$N_L = \frac{z_R - z}{z_R - z_L} \qquad N_R = \frac{z - z_L}{z_R - z_L} \tag{H.15}$$

which when used in the weak form again yield exact answers at the interelement nodes.

After constructing exact nodal solutions for u, exact solutions for the flux at the interelement nodes can also be obtained from the weak form for each element. The above process was first given by Tong for self-adjoint differential equations.†

† P. Tong. Exact solutions of certain problems by the finite element method. *J. AIAA*, **7**, 179–80, 1969.

Appendix I

Matrix diagonalization or lumping

Some of the algorithms discussed in this volume become more efficient if one of the global matrices can be diagonalized (also called 'lumped' by many engineers). For example, the solution of some mixed and transient problems are more efficient if a global matrix to be inverted (or equations solved) is diagonal [see Chapter 12, Eq. (12.95) and Chapter 17, Sec. 17.2.4 and 17.4.2]. Engineers have persisted with purely physical concepts of lumping; however, there is clearly a need for devising a systematic and mathematically acceptable procedure for such lumping.

We shall define the matrix to be considered as

$$\mathbf{A} = \int_{\Omega} \mathbf{N}^{\mathrm{T}} \mathbf{c} \mathbf{N} \, \mathrm{d}\Omega \tag{I.1}$$

where $\mathbf{c}$ is a matrix with small dimension. Often $\mathbf{c}$ is a diagonal matrix (e.g., in mass or simple least square problems $\mathbf{c}$ is an identity matrix times some scalar).When $\mathbf{A}$ is computed exactly it has full rank and is not diagonal – this is called the *consistent* form of $\mathbf{A}$ since it is computed consistently with the other terms in the finite element model. The diagonalized form is defined with respect to 'nodes' or the shape functions, e.g., $\mathbf{N}_i = N_i \mathbf{I}$; hence, the matrix will have small diagonal blocks, each with the maximum dimension of c. Only when $\mathbf{c}$ is diagonal can the matrix $\mathbf{A}$ be completely diagonalized. Four basic lines of argument may be followed in constructing a diagonal form.

The first procedure is to use different shape functions to approximate each term in the finite element discretization. For the $\mathbf{A}$ matrix we use substitute shape functions $\bar{\mathbf{N}}_i$ for the lumping process. No derivatives exist in the definition of $\mathbf{A}$, hence, for this term the shape functions may be piecewise continuous within and between elements and still lead to an acceptable approximation. If the shape functions used to define $\mathbf{A}$ are piecewise constants, such that $\bar{\mathbf{N}}_i$ in a certain part of the element surrounding the node i and zero elsewhere, and such parts are not overlapping or disjoint, then clearly the matrix of Eq. (I.1) becomes nodally diagonal as

$$\int_{\Omega} \bar{\mathbf{N}}_i^{\mathrm{T}} \mathbf{c} \bar{\mathbf{N}}_j \, \mathrm{d}\Omega = \begin{cases} \displaystyle\int_{\Omega_i} \mathbf{c} \, \mathrm{d}\Omega & i = j \\ 0 & i \neq j \end{cases} \tag{I.2}$$

Such an approximation with different shape functions is permissible since the usual finite element criteria of integrability and completeness are satisfied. We can verify

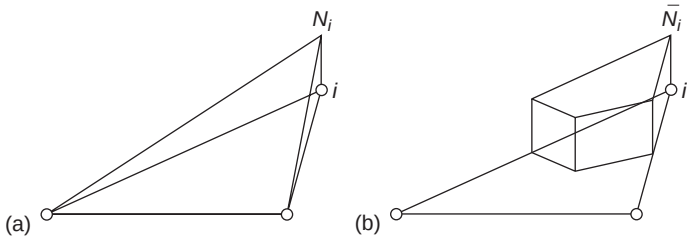


Fig. I.1 (a) Linear and (b) piecewise constant shape functions for a triangle.

this using a patch test to show that consistency is still maintained in the approximation. The functions selected need only satisfy the condition

$$\bar{\mathbf{N}}_i = \bar{N}_i \mathbf{I} \qquad \text{with} \qquad \sum_i \bar{N}_i = 1 \tag{I.3}$$

for all points in the element and this also maintains a partition of unity property in all of Ω. In Fig. I.1 we show the functions N_i and $\bar{N}_i$ for a triangular element.

The second method to diagonalize a matrix is to note that condition (I.1) is simply a requirement that ensures conservation of the quantity $\mathbf{c}$ over the element. For structural dynamics applications this is the conservation of mass at the element level. Accordingly, it has been noted that any lumping that preserves the integral of $\mathbf{c}$ on the element will lead to convergent results, although the rate of convergence may be lower than with use of a consistent $\mathbf{A}$. Many alternatives have been proposed based upon this method. The earliest procedures performed the diagonalization using physical intuition only. Later alternative algorithms were proposed. One suggestion, often called a 'row sum' method, is to compute the diagonal matrix from

$$\mathbf{A}_{ij} = \begin{cases} \sum_k \int_{\Omega_i} \mathbf{N}_i^{\mathrm{T}} \mathbf{c} \mathbf{N}_k \, \mathrm{d}\Omega & i = j \\ \mathbf{0} & i \neq j \end{cases} \tag{I.4}$$

This simplifies to

$$\mathbf{A}_{ij} = \begin{cases} \int_{\Omega_i} \mathbf{N}_i^{\mathrm{T}} \mathbf{c} \, \mathrm{d}\Omega & i = j \\ \mathbf{0} & i \neq j \end{cases} \tag{I.5}$$

since the sum of the shape functions is unity. This algorithm makes sense only when the degrees of freedom of the problem all have the same physical interpretation. An alternative is to scale the diagonals of the consistent mass to satisfy the conservation requirement. In this case the diagonal matrix is deduced from

$$\mathbf{A}_{ij} = \begin{cases} a \int_{\Omega_i} \mathbf{N}_i^{\mathrm{T}} \mathbf{c} \mathbf{N}_i \, \mathrm{d}\Omega & i = j \\ \mathbf{0} & i \neq j \end{cases} \tag{I.6}$$

where a is selected so that

$$\sum_i \mathbf{A}_{ii} = \int_\Omega \mathbf{c} \, \mathrm{d}\Omega \tag{I.7}$$

The third procedure uses numerical integration to obtain a diagonal array without apparently introducing additional shape functions. Use of numerical integration to evaluate the **A** matrix of Eq. (I.1) yields a typical term in the summation form (following Chapter 9)

$$\mathbf{A}_{ij} = \int_{\Omega} \mathbf{N}_i^{\mathrm{T}} \mathbf{c} \mathbf{N}_j \, \mathrm{d}\Omega = \sum_q (\mathbf{N}_i^{\mathrm{T}} \mathbf{c} \mathbf{N}_j)_{\xi_q} J_q W_q \tag{I.8}$$

where ξ_q refers to the quadrature point at which the integrand is evaluated, J is the jacobian volume transformation at the same point and W_q gives the appropriate quadrature weight.

If the quadrature points for the numerical integration are located at nodes then (for standard shape functions) by Eq. (I.3) the diagonal matrix is

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{c} J_i W_i & i = j \\ \mathbf{0} & i \neq j \end{cases} \tag{I.9}$$

where J_i is the jacobian and W_i is the quadrature weight at node i.

Appropriate weighting values may be deduced by requiring the quadrature formula to exactly integrate particular polynomials in the natural coordinate system. In general the quadrature should integrate a polynomial of the highest complete order in the shape functions. Thus, for four-noded quadrilateral elements, linear functions should be exactly integrated. Integrating additional terms may lead to improved accuracy but is not required. Indeed, only conservation of **c** is required.

For low-order elements, symmetry arguments may be used to lump the matrix. It is, for instance, obvious that in a simple triangular element little improvement can be obtained by any other lumping than the simple one in which the total **c** is distributed in three equal parts. For an eight-noded two-dimensional isoparametric element no such obvious procedure is available. In Fig. I.2 we show the case of rectangular

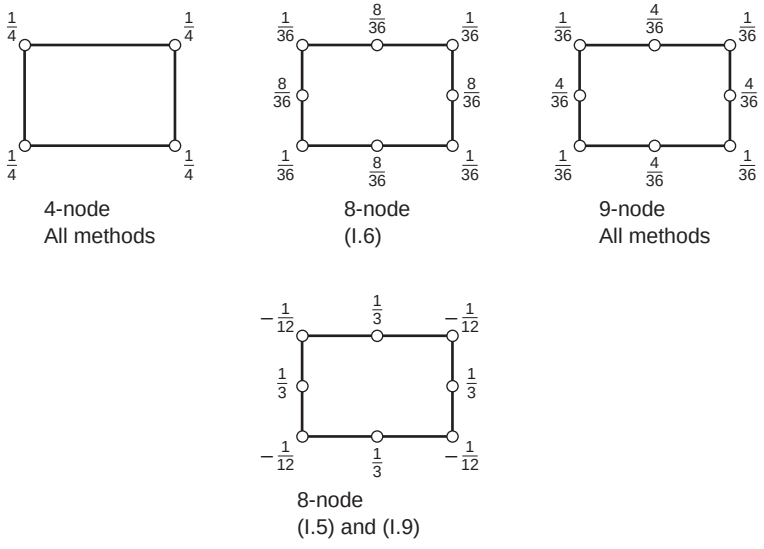


Fig. I.2 Diagonalization of rectangular elements by three methods.

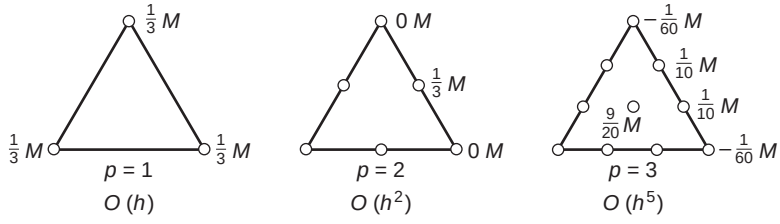


Fig. I.3 Diagonalization of triangular elements by quadrature.

elements of four-, eight-, and nine-noded type and lumping by Eqs (I.5), (I.6) and (I.9).

It is noted that for the eight-noded element some of the lumped quantities are negative when Eq. (I.5) or Eq. (I.9) is used. These will have some adverse effects in certain algorithms (e.g., time-stepping schemes to integrate transient problems) and preclude their use. In Fig. I.3 we show some lumped matrices for triangular elements computed by quadrature (i.e., Eq. (I.9)). It is noted here that the cubic element has negative terms while the quadratic element has zero terms. The zero terms are particularly difficult to handle as the resulting diagonal matrix **A** no longer has full rank and thus may not be inverted.

Another aspect of lumping is the performance of the element when distorted from its parent element shape. For example, as a rectangular element is distorted and approaches a triangular shape it is desirable to have the limit triangular shape case behave appropriately. In the case of a four-noded rectangular element the lumped matrix for all three procedures gives the same answer. However if the element is distorted by a transformation defined by one parameter f as shown in Fig. I.4 then the three lumping procedures discussed so far give different answers. The jacobian transformation is given by

$$J = ab(1 - f) \tag{I.10}$$

and **c** is here taken as the identity matrix.

The form (I.5) gives

$$\mathbf{A}_{ii} = \begin{cases} ab(1 - f/3) & \text{at top nodes} \\ ab(1 + f/3) & \text{at bottom nodes} \end{cases} \tag{I.11}$$

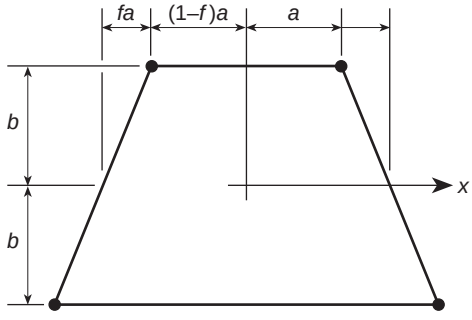


Fig. I.4 Distorted four-noded element.

the form (I.6) gives

$$\mathbf{A}_{ii} = \begin{cases} ab(1-f/2) & \text{at top nodes} \\ ab(1+f/2) & \text{at bottom nodes} \end{cases} \tag{I.12}$$

and the quadrature form (I.9) yields

$$\mathbf{A}_{ii} = \begin{cases} ab(1-f) & \text{at top nodes} \\ ab(1+f) & \text{at bottom nodes} \end{cases} \tag{I.13}$$

The four-noded element has the property that a triangle may be defined by coalescing two nodes and assigning them to the same global node in the mesh. Thus, the quadrilateral is identical to a three-noded triangle when the parameter f is unity. The limit value for the row sum method will give equal lumped terms at the three nodes while method (I.6) yields a lumped value for the coalesced node which is two-thirds the value at the other nodes and the quadrature method (I.9) yields a zero lumped value at the coalesced node. Thus, methods (I.6) and(I.9) give limit cases which depend on how the nodes are numbered to form each triangular element. This lack of invariance is not desirable in computer programs; hence for the four-noded quadrilateral, method (I.5) appears to be superior to the other two. On the other hand,we have observed above that the row sum method (I.5) leads to negative diagonal elements for the eight-noded element; hence there is no universal method for diagonalizing a matrix.

A fourth but not widely used method is available which may be explored to deduce a consistent matrix that is diagonal. This consists of making a mixed representation for the term creating the **A** matrix.

Consider a functional given by

$$\Pi_1 = \tfrac{1}{2} \int_\Omega \mathbf{u}^{\mathrm{T}} \mathbf{c} \mathbf{u} \, \mathrm{d}\Omega \tag{I.14}$$

The first variation of Π_1 yields

$$\delta\Pi_1 = \int_\Omega \delta\mathbf{u}^{\mathrm{T}} \mathbf{c} \mathbf{u} \, \mathrm{d}\Omega \tag{I.15}$$

Approximation using the standard form

$$\mathbf{u} \approx \hat{\mathbf{u}} = \mathbf{N}_i \tilde{\mathbf{u}}_i = \mathbf{N}\tilde{\mathbf{u}} \tag{I.16}$$

yields

$$\delta\Pi_1 = \delta\tilde{\mathbf{u}}^{\mathrm{T}} \int_\Omega \mathbf{N}^{\mathrm{T}} \mathbf{c} \mathbf{N} \, \mathrm{d}\Omega \, \tilde{\mathbf{u}} \tag{I.17}$$

This yields exactly the form for **A** given by Eq. (I.1).

We can construct an alternative mixed form by introducing a momentum type variable given by

$$\mathbf{p} = \mathbf{c}\mathbf{u} \tag{I.18}$$

The Hellinger–Reissner type mixed form may then be expressed as

$$\Pi_2 = \int_\Omega \mathbf{u}^{\mathrm{T}} \mathbf{p} \, \mathrm{d}\Omega - \tfrac{1}{2} \int_\Omega \mathbf{p}^{\mathrm{T}} \mathbf{c}^{-1} \mathbf{p} \, \mathrm{d}\Omega \tag{I.19}$$

and has the first variation

$$\delta\Pi_2 = \int_\Omega \delta\mathbf{u}^{\mathrm{T}}\mathbf{p}\,\mathrm{d}\Omega + \int_\Omega \delta\mathbf{p}^{\mathrm{T}}\left(\mathbf{u} - \mathbf{c}^{-1}\mathbf{p}\right)\mathrm{d}\Omega \tag{I.20}$$

The term with variation on $\mathbf{u}$ will combine with other terms so is not set to zero; however the other term will not appear elsewhere so can be solved separately.

If we now introduce an approximation for $\mathbf{p}$ as

$$\mathbf{p} \approx \hat{\mathbf{p}} = \mathbf{n}_j\tilde{\mathbf{p}}_j = \mathbf{n}\tilde{\mathbf{p}} \tag{I.21}$$

then the variational equation becomes

$$\delta\Pi_2 = \delta\tilde{\mathbf{u}}^{\mathrm{T}}\int_\Omega \mathbf{N}^{\mathrm{T}}\mathbf{n}\,\mathrm{d}\Omega\,\tilde{\mathbf{p}} + \delta\tilde{\mathbf{p}}^{\mathrm{T}}\left(\int_\Omega \mathbf{n}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Omega\,\tilde{\mathbf{u}} - \int_\Omega \mathbf{n}^{\mathrm{T}}\mathbf{c}^{-1}\mathbf{n}\,\mathrm{d}\Omega\,\tilde{\mathbf{p}}\right) \tag{I.22}$$

If we now define the matrices

$$\begin{aligned}\mathbf{G} &= \int_\Omega \mathbf{N}^{\mathrm{T}}\mathbf{n}\,\mathrm{d}\Omega\\ \mathbf{H} &= \int_\Omega \mathbf{n}^{\mathrm{T}}\mathbf{c}^{-1}\mathbf{n}\,\mathrm{d}\Omega\end{aligned} \tag{I.23}$$

then the weak form is

$$\delta\Pi_2 = [\,\delta\tilde{\mathbf{u}}^{\mathrm{T}} \quad \delta\tilde{\mathbf{p}}^{\mathrm{T}}\,]\left(\begin{bmatrix}\bullet & \mathbf{G}^{\mathrm{T}}\\ \mathbf{G} & -\mathbf{H}\end{bmatrix}\begin{Bmatrix}\tilde{\mathbf{u}}\\ \tilde{\mathbf{p}}\end{Bmatrix} = \begin{Bmatrix}\bullet\\ \mathbf{0}\end{Bmatrix}\right) \tag{I.24}$$

Eliminating $\tilde{\mathbf{p}}$ using the second row of Eq. (I.24) gives

$$\mathbf{A} = \mathbf{G}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{G} \tag{I.25}$$

for which diagonal forms may now be sought. This form again has the same options as discussed above but, in addition, forms for the shape functions $\mathbf{n}$ can be sought which also render the matrix diagonal.

Author index

Page numbers in **bold** refer to the list of references at the end of each chapter.

Subject index

SPECIAL OFFER

Purchasers of **Volume 1 of The Finite Element Method** are entitled to a **free one year licence** of the following codes*:

☐ **GiD. The Personal Pre/Post-processor.** The universal, adaptive and user friendly pre and post processing system for computer analysis in science and engineering.

GiD is ideal for generating all the information (structured and unstructured meshes, boundary and loading conditions, material types, visualisation of results etc.) required for the analysis of any problem in science and engineering using numerical methods. Typical problems that can be successfully tackled with GiD include most situations in structural mechanics, fluid dynamics, electromagnetics, heat transfer, geomechanics, etc. using finite element, finite volume, boundary element, finite difference or point-based (meshless) numerical procedures. Documentation available via Internet includes user instructions and tutorial manual. The version of GiD offered is limited to 100.000 elements.

☐ ***FEAP*** A general finite element program for steady-state and transient analysis. Compiled professional version linked to GiD

Executable program includes all modules including contact, library of elements for small and finite deformation analysis of solids with choice of elastic and inelastic constitutive equations, and different linear equation and time integration options. Solution process is command language driven and permits choice of batch and/or interactive solution modes. Documentation includes user instructions, theoretical manual, and examples manual.

The GiD and FEAP codes are operative for a PC under Windows or Linux.
For further information please visit http://www.gid.upc.es

**This offer is valid until 31 December 2001*

Order form

Name _____

Organization _____

Address _____

City _____ Postal Code _____

Country _____

e.mail _____

Please send me via internet **free one year licence** of:

☐ **GiD**: The personal pre/post processor (version limited to 100.000 elements)

- ☐ PC under Windows
- ☐ PC under Linux

☐ ***FEAP*** Finite Element Analysis program compiled professional version linked to GiD.

- ☐ PC under Windows
- ☐ PC under Linux

Please tear off this page and send it by surface mail prior to 31 December 2001 to:

International Center for Numerical Methods in Engineering (CIMNE)
Edicio C1, Campus Norte UPC, Gran Capitán s/n
08034 Barcelona, Spain


POWEREN.IR