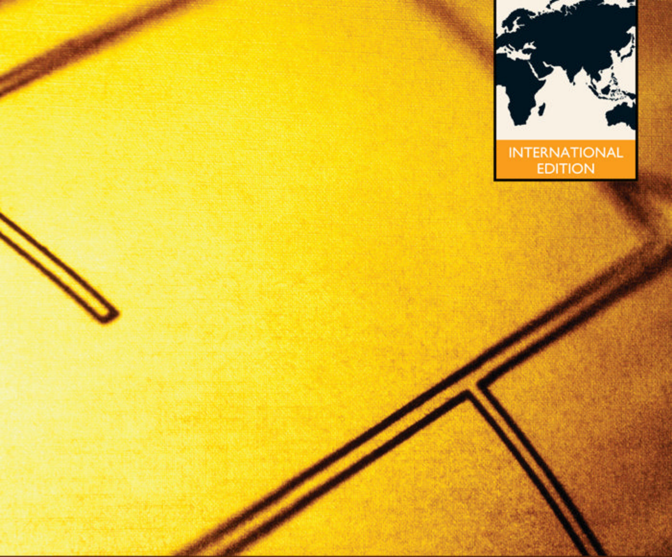


Introduction to Finite Elements in Engineering

Fourth Edition

Tirupathi R. Chandrupatla • Ashok D. Belegundu

ALWAYS LEARNING

PEARSON

Introduction to Finite Elements in Engineering

FOURTH EDITION

TIRUPATHI R. CHANDRUPATLA
Rowan University
Glassboro, New Jersey

ASHOK D. BELEGUNDU
The Pennsylvania State University
University Park, Pennsylvania

International Edition contributions by

T. RAMESH
National Institute of Technology
Tiruchirapalli, India

CHAITALI RAY
Bengal Engineering and Science University
Shibpur, India

PEARSON

Upper Saddle River Boston Columbus San Francisco New York
Indianapolis London Toronto Sydney Singapore Tokyo Montreal
Dubai Madrid Hong Kong Mexico City Munich Paris Amsterdam Cape Town

Vice President and Editorial Director, ECS: *Marcia J. Horton*
Acquisitions Editor: *Norrin Dias*
Vice-President, Production: *Vince O'Brien*
Executive Marketing Manager: *Tim Galligan*
Marketing Assistant: *Jon Bryant*
Permissions Project Manager: *Karen Sanatar*
Senior Managing Editor: *Scott Disanno*
Production Manager: *Pat Brown*
Publisher, International Edition: *Angshuman Chakraborty*
Acquisitions Editor, International Edition: *Somnath Basu*
Publishing Assistant, International Edition: *Shokhi Shah*
Print and Media Editor, International Edition: *Ashwitha Jayakumar*
Project Editor, International Edition: *Jayashree Arunachalam*
Operations Specialist: *Lisa McDowell*
Cover Designer: *Suzanne Behnke*
Full-Service Project Management: *Maheswari PonSaravanan*, TexTech International
Cover Printer: *Lehigh-Phoenix*

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:
www.pearsoninternationaleditions.com

Authorized adaptation from the United States edition, entitled Introduction to Finite Elements in Engineering, 4th edition, *ISBN 978-0-13-216274-6 by Tirupathi R. Chandrupatla and Ashok D. Belegundu published by Pearson Education © 2012*.

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

10 9 8 7 6 5 4 3 2 1
14 13 12 11 10

Typeset in 10/12 Times Ten LT Std by TexTech International
Printed and bound by Courier/Westford in The United States of America

The publisher's policy is to use paper manufactured from sustainable forests.

PEARSON

ISBN-10: 0-273-76368-7
ISBN-13: 978-0-273-76368-0

To our parents

This page intentionally left blank

Contents

This page intentionally left blank

Preface

The first edition of this book appeared over 20 years ago and the second and third editions followed subsequently. Translations of the book appeared in Spanish, Korean, Greek, and Chinese languages. We received positive feedback from professors who taught from the book and from students and practicing engineers who used the book. We also benefited from the feedback received from the students in our courses for the past 30 years. We have incorporated several suggestions in this edition. The underlying philosophy of the book is to provide a clear presentation of theory, aspects of problem modeling and implementation into computer programs. The pedagogy of earlier editions has been retained and enhanced in this edition.

WHAT'S NEW IN THIS EDITION

- Introduction of the superposition principle.
- Treatment of symmetry and antisymmetry.
- Additional examples and exercise problems.
- The patch test.
- Beams and Frames chapter moved to follow the Truss chapter.
- Revised Excel VB programs.
- Programs in JAVASCRIPT to run on web browsers such as IE, Firefox, Google Chrome, or Safari.
- Executable *graphics* programs together with the source codes.
- Additional examples and exercise problems.

New material has been introduced in several chapters. Worked examples and exercise problems have been added to supplement the learning process. Exercise problems stress both fundamental understanding and practical considerations. Problem modeling aspects have been added from early chapters. Principle of superposition is introduced in Chapter 1. Symmetry and antisymmetry considerations in two-dimensional problems are clearly presented. There are additional examples and exercise problems. The patch test is discussed and related problems have been added. The included programs have a common structure, which should enable the users to follow the development easily. Programs in JAVASCRIPT have been added. This will enable the users to solve finite element analysis problems using web browsers such as IE, Firefox, Safari, or Google Chrome. Excel VB programs have been revised. All programs have been thoroughly checked. The downloadable program set includes executable versions for programs involving graphics. The programs have been provided in Visual Basic, Microsoft Excel/Visual Basic, MATLAB, and JAVASCRIPT, together with those provided earlier in QBASIC, FORTRAN, and C. The Solutions Manual has been updated.

Chapter 1 gives a brief historical background and develops the fundamental concepts. Equations of equilibrium, stress–strain relations, strain–displacement relations, and the principles of potential energy are reviewed. The concept of Galerkin's method is introduced.

Properties of matrices and determinants are reviewed in Chapter 2. The Gaussian elimination method is presented, and its relationship to the solution of symmetric banded matrix equations and the skyline solution is discussed. Cholesky decomposition and conjugate gradient method are discussed.

Chapter 3 develops the key concepts of finite element formulation by considering one-dimensional problems. The steps include development of shape functions, derivation of element stiffness, formation of global stiffness, treatment of boundary conditions, solution of equations, and stress calculations. Both the potential energy approach and Galerkin's formulations are presented. Consideration of temperature effects is included.

Finite element formulation for plane and three-dimensional trusses is developed in Chapter 4. The assembly of global stiffness in banded and skyline forms is explained. Computer programs for both banded and skyline solutions are given.

Beams and application of Hermite shape functions are presented in Chapter 5. The chapter covers two-dimensional and three-dimensional frames.

Chapter 6 introduces the finite element formulation for two-dimensional plane stress and plane strain problems using constant strain triangle (CST) elements. Problem modeling and treatment of boundary conditions are presented in detail. Formulation for orthotropic materials is provided.

Chapter 7 treats the modeling aspects of axisymmetric solids subjected to axisymmetric loading. Formulation using triangular elements is presented. Several real-world problems are included in this chapter.

Chapter 8 introduces the concepts of isoparametric quadrilateral and higher-order elements and numerical integration using Gaussian quadrature. Formulation for axisymmetric quadrilateral element and implementation of conjugate gradient method for quadrilateral element are given.

Chapter 9 presents three-dimensional stress analysis. Tetrahedral and hexahedral elements are presented. The frontal method and its implementation aspects are discussed.

Scalar field problems are treated in detail in Chapter 10. While Galerkin as well as energy approaches have been used in every chapter with equal importance, only Galerkin's approach is used in this chapter. This approach directly applies to the given differential equation without the need of identifying an equivalent functional to minimize. Galerkin's formulation for steady-state heat transfer, torsion, potential flow, seepage flow, electric and magnetic fields, fluid flow in ducts, and acoustics are presented.

Chapter 11 introduces dynamic considerations. Element mass matrices are given. Techniques for evaluation of eigenvalues (natural frequencies) and eigenvectors (mode shapes) of the generalized eigenvalue problem are discussed. Methods of inverse iteration, Jacobi, tridiagonalization, and implicit shift approaches are presented.

Preprocessing and postprocessing concepts are developed in Chapter 12. Theory and implementation aspects of two-dimensional mesh generation, least-squares approach to obtain nodal stresses from element values for triangles and quadrilaterals, and contour plotting are presented.

At the undergraduate level some topics may be dropped or delivered in a different order without breaking the continuity of presentation. We encourage the introduction of the Chapter 12 programs at the end of Chapter 6. This helps the students to prepare the data in an efficient manner.

We thank Professor Hongbing Fang, Mechanical Engineering and Engineering Science, UNC Charlotte; Professor Kishore Pochiraju, Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, New Jersey; Professor Subramaniam Rajan, Ira A. Fulton School of Engineering, Arizona State University; Professor Chris H. Reidel, A. Leon Linton Department of Mechanical Engineering, Lawrence Technological University, Michigan; and Professor Nicholas J. Zabaras, Sibley School of Mechanical and Aerospace Engineering, Cornell University, who reviewed our third edition and gave many constructive suggestions that helped us improve the book.

Complete self-contained computer programs with source codes in Visual Basic, Excel-based Visual Basic, MATLAB, FORTRAN, JAVASCRIPT and C to accompany the text are available at www.pearsoninternationaleditions.com/chandrupatla.

Tirupathi Chandrupatla expresses his gratitude to J. Tinsley Oden, whose teaching and encouragement have been a source of inspiration to him throughout his career. He expresses his thanks to many students at Rowan University and Kettering University who took his courses. He expresses his thanks to his colleague Paris von Lockette who gave valuable feedback after teaching the course using the second and third editions.

Ashok D. Belegundu thanks his students at Penn State for their feedback on the course material and programs.

We thank Marcia Horton who provided us guidance with the present and previous editions of the book. We thank our editor Norrin Dias, Tacy Quinn, Debbie Yarnell, and Clare Romeo all from Prentice Hall who turned this project into a pleasant chore. We thank Maheswari PonSaravanan, project manager and her team at TexTech International, Chennai, India for efficient handling of copyediting and page proofs.

TIRUPATHI R. CHANDRUPATLA
ASHOK D. BELEGUNDU

The publishers would like to thank Mr Basavaraj Gudadappanavar of SDMCET Dharwad (India) for reviewing the content of the International Edition.

About the Authors

Tirupathi R. Chandrupatla is Professor and Founding Chair of Mechanical Engineering at Rowan University, Glassboro, New Jersey. He received the B.S. degree from National Institute of Technology (formerly REC), Warangal. He received the M.S. degree in design and manufacturing from the Indian Institute of Technology (IIT), Bombay. He started his career as a design engineer with Hindustan Machine Tools, Bangalore. He then taught in the Department of Mechanical Engineering at IIT, Bombay. He pursued his graduate studies in the Department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin and received his Ph.D. in 1977. He subsequently taught at the University of Kentucky. Prior to joining Rowan, he was a Professor of Mechanical Engineering and Manufacturing Systems Engineering at Kettering University (formerly GMI Engineering and Management Institute/General Motors Institute), where he taught for 16 years.

Dr. Chandrupatla has broad research interests, which include finite element analysis, design, optimization, quality and reliability, and manufacturing engineering. He has published widely in these areas and serves as a consultant to industry. Dr. Chandrupatla is a registered Professional Engineer and also a Certified Manufacturing Engineer. He is a member of ASEE, ASME, SAE, and SME.

Ashok D. Belegundu has been a Professor of Mechanical Engineering at The Pennsylvania State University, University Park, since 1986. Prior to this, he taught at GMI, now Kettering University, in Michigan. He received his B.Tech. degree from IIT, Madras and his Ph.D. from the University of Iowa. He has been a principal investigator on research projects involving finite element based optimization for several agencies, including the National Science Foundation, Army Research Office, NASA, SERC (UK), MacNeal-Schwendler Corporation, Gentex Corporation, and Ingersoll-Rand. A detailed list of his publications and projects can be found at http://www.mne.psu.edu/Directories/Faculty/Belegundu-A.html. He has given short courses on finite elements to the Forging Industry Association, Hazleton Pumps, Infosys (India).

C H A P T E R 1

Fundamental Concepts

1.1 INTRODUCTION

The finite element method has become a powerful tool for the numerical solution of a wide range of engineering problems. Applications range from deformation and stress analysis of automotive, aircraft, building, and bridge structures to field analysis of heat flux, fluid flow, magnetic flux, seepage, and other flow problems. With the advances in computer technology and CAD systems, complex problems can be modeled with relative ease. Several alternative configurations can be tested on a computer before the first prototype is built. All these suggest that we need to keep pace with these developments by understanding the basic theory, modeling techniques, and computational aspects of the finite element method. In this method of analysis, a complex region defining a continuum is discretized into simple geometric shapes called finite elements. The material properties and the governing relationships are considered over these elements and expressed in terms of unknown values at element corners. An assembly process, duly considering the loading and constraints, results in a set of equations. Solution of these equations gives us the approximate behavior of the continuum.

1.2 HISTORICAL BACKGROUND

Basic ideas of the finite element method originated from advances in aircraft structural analysis. In 1941, Hrenikoff presented a solution of elasticity problems using the "frame work method." Courant's paper, which used piecewise polynomial interpolation over triangular subregions to model torsion problems, was published in 1943. Turner et al. derived stiffness matrices for truss, beam, and other elements and presented their findings in 1956. The term *finite element* was first coined and used by Clough in 1960.

In the early 1960s, engineers used the method for approximate solution of problems in stress analysis, fluid flow, heat transfer, and other areas. A book by Argyris in 1955 on energy theorems and matrix methods laid a foundation for further developments in finite element studies. The first book on finite elements by Zienkiewicz and

Cheung was published in 1967. In the late 1960s and early 1970s, finite element analysis was applied to nonlinear problems and large deformations. Oden's book on nonlinear continua appeared in 1972.

Mathematical foundations were laid in the 1970s. New element development, convergence studies, and other related areas fall in this category.

Today, the developments in mainframe computers and availability of powerful microcomputers have brought this method within the reach of students and engineers working in small industries.

1.3 OUTLINE OF PRESENTATION

In this book, we adopt the potential energy and the Galerkin approaches for the presentation of the finite element method. This method originated from the field of solids and structures, and we start our study with these ideas to solidify understanding. For this reason, several early chapters deal with rods, beams, and elastic deformation problems. The same steps are used in the development of material throughout the book, so that the similarity of approach is retained in every chapter. The finite element ideas are then extended to field problems in Chapter 10. Every chapter includes a set of problems and computer programs for interaction.

We now recall some fundamental concepts needed in the development of the finite element method.

1.4 STRESSES AND EQUILIBRIUM

A three-dimensional body occupying a volume V and having a surface S is shown in Fig. 1.1. Points in the body are located by x, y, z coordinates. The boundary is constrained on some region, where displacement is specified. On part of the boundary, distributed force per unit area $\mathbf{T}$, also called traction, is applied. Under the force, the body deforms. The deformation of a point $\mathbf{x}(=[x, y, z]^{\mathrm{T}})$ is given by the three components of its displacement:

$$\mathbf{u} = [u, v, w]^{\mathrm{T}} \tag{1.1}$$

The distributed force per unit volume, for example, the weight per unit volume, is the vector $\mathbf{f}$ given by

$$\mathbf{f} = [f_x, f_y, f_z]^{\mathrm{T}} \tag{1.2}$$

The body force acting on the elemental volume dV is shown in Fig. 1.1. The surface traction $\mathbf{T}$ may be given by its component values at points on the surface:

$$\mathbf{T} = [T_x, T_y, T_z]^{\mathrm{T}} \tag{1.3}$$

Examples of traction are distributed contact force and action of pressure. A load $\mathbf{P}$ acting at a point i is represented by its three components:

$$\mathbf{P}_i = [P_x, P_y, P_z]_i^{\mathrm{T}} \tag{1.4}$$

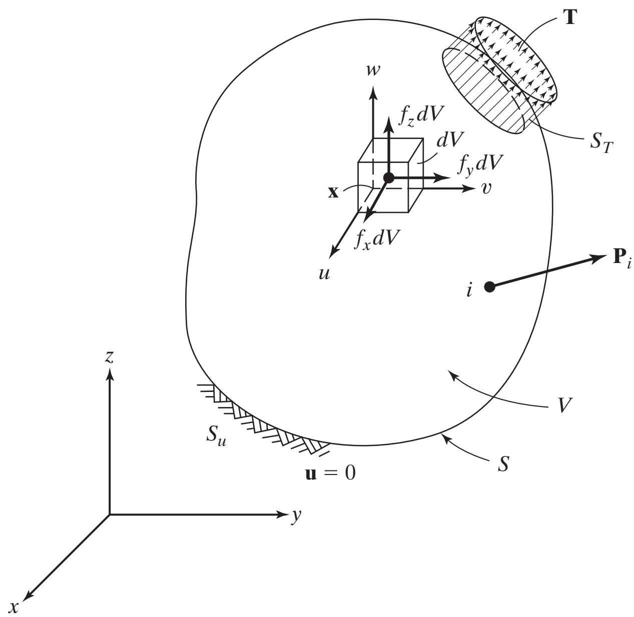


FIGURE 1.1 Three-dimensional body.

The stresses acting on the elemental volume dV are shown in Fig. 1.2. When the volume dV shrinks to a point, the stress tensor is represented by placing its components in a (3×3) symmetric matrix. However, we represent stress by the six independent components as in

$$\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z, \tau_{yz}, \tau_{xz}, \tau_{xy}]^{\mathrm{T}} \tag{1.5}$$

where $\sigma_x, \sigma_y, \sigma_z$ are normal stresses and $\tau_{yz}, \tau_{xz}, \tau_{xy}$ are shear stresses. Let us consider the equilibrium of the elemental volume shown in Fig. 1.2. First, we get forces on faces by multiplying the stresses by the corresponding areas. Writing $\Sigma F_x = 0$, $\Sigma F_y = 0$, and $\Sigma F_z = 0$ and recognizing $dV = dx\, dy\, dz$, we get the equilibrium equations

$$\begin{aligned}
\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + f_x &= 0 \\
\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + f_y &= 0 \\
\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + f_z &= 0
\end{aligned} \tag{1.6}$$

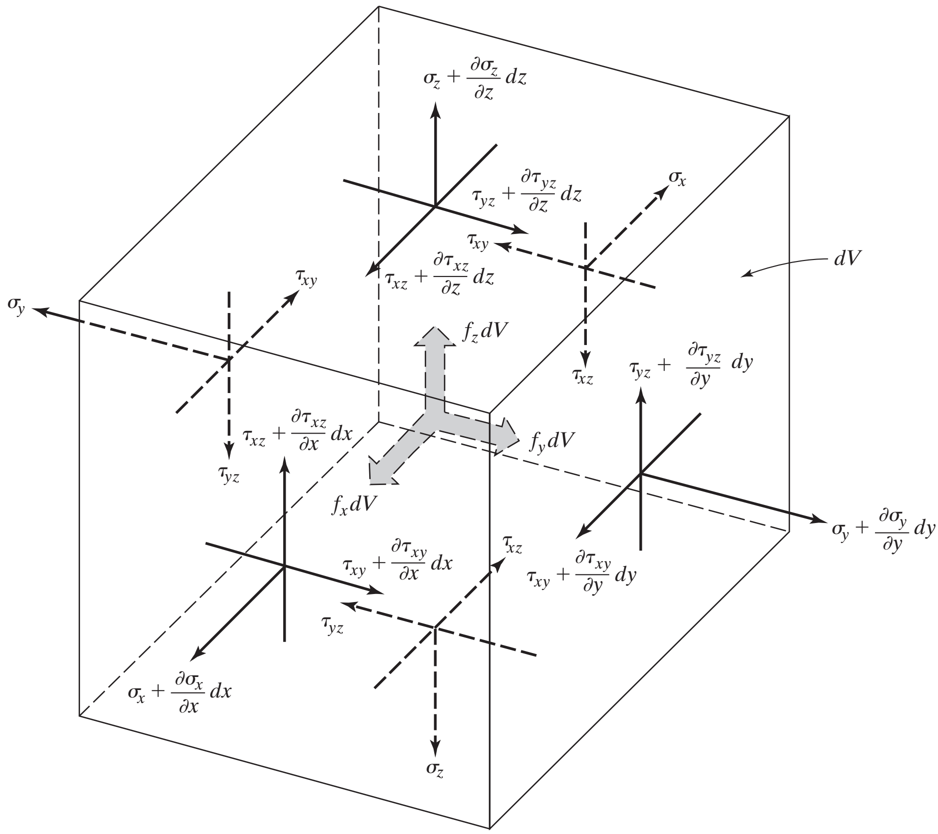


FIGURE 1.2 Equilibrium of elemental volume.

1.5 BOUNDARY CONDITIONS

Referring to Fig. 1.1, we find that there are displacement boundary conditions and surface-loading conditions. If $\mathbf{u}$ is specified on part of the boundary denoted by S_u, we have

$$\mathbf{u} = \mathbf{0} \quad \text{on} \quad S_u \tag{1.7}$$

We can also consider boundary conditions such as $\mathbf{u} = \mathbf{a}$, where $\mathbf{a}$ is a given displacement.

We now consider the equilibrium of an elemental tetrahedron $ABCD$, shown in Fig. 1.3, where DA, DB, and DC are parallel to the x-, y-, and z-axes, respectively, and area ABC, denoted by dA, lies on the surface. If $\mathbf{n} = [n_x, n_y, n_z]^{\mathrm{T}}$ is the unit normal to dA, then area $BDC = n_x dA$, area $ADC = n_y dA$, and area $ADB = n_z dA$. Considering equilibrium along the three axes directions we get

$$\begin{aligned}
\sigma_x n_x + \tau_{xy} n_y + \tau_{xz} n_z &= T_x \\
\tau_{xy} n_x + \sigma_y n_y + \tau_{yz} n_z &= T_y \\
\tau_{xz} n_x + \tau_{yz} n_y + \sigma_z n_z &= T_z
\end{aligned} \tag{1.8}$$

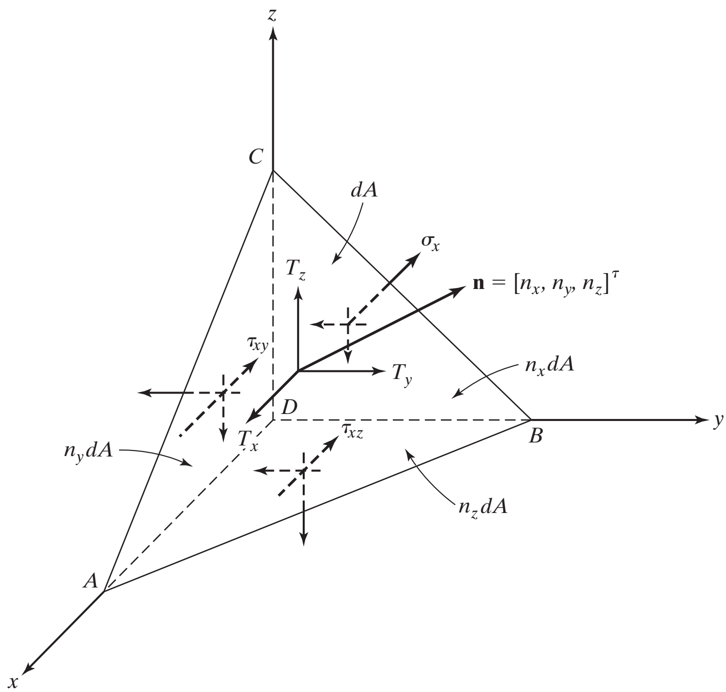


FIGURE 1.3 An elemental volume at surface.

These conditions must be satisfied on the boundary, S_T, where the tractions are applied. In this description, the point loads must be treated as loads distributed over small, but finite areas.

1.6 STRAIN–DISPLACEMENT RELATIONS

We represent the strains in a vector form that corresponds to the stresses in Eq. 1.5,

$$\boldsymbol{\epsilon} = [\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{yz}, \gamma_{xz}, \gamma_{xy}]^{\mathrm{T}} \tag{1.9}$$

where ϵ_x, ϵ_y, and ϵ_z are normal strains and γ_{yz}, γ_{xz}, and γ_{xy} are the engineering shear strains.

Figure 1.4 gives the deformation of the $dx - dy$ face for small deformations, which we consider here. Considering the other faces, we can write

$$\boldsymbol{\epsilon} = \left[\frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial w}{\partial z}, \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}, \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right]^{\mathrm{T}} \tag{1.10}$$

These strain relations hold for small deformations.

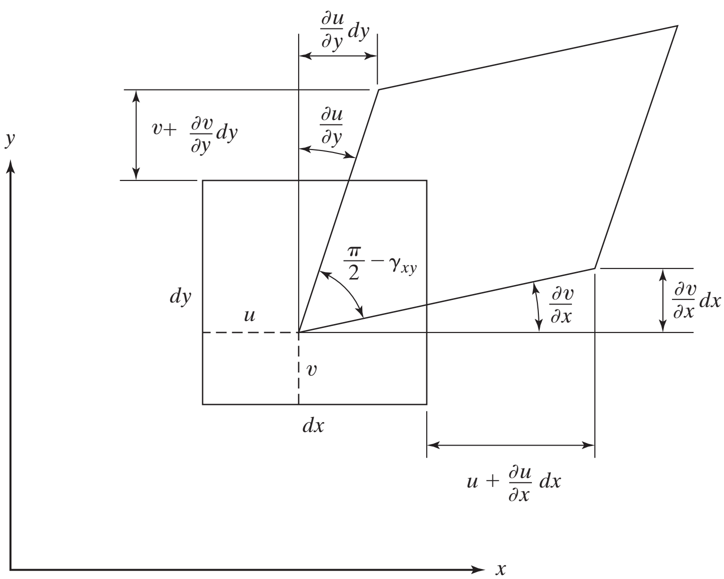


FIGURE 1.4 Deformed elemental surface.

1.7 STRESS–STRAIN RELATIONS

For linear elastic materials, the stress–strain relations come from the generalized Hooke's law. For isotropic materials, the two material properties are Young's modulus (or modulus of elasticity) E and Poisson's ratio ν. Considering an elemental cube inside the body, Hooke's law gives

$$
\begin{aligned}
\epsilon_x &= \frac{\sigma_x}{E} - \nu\frac{\sigma_y}{E} - \nu\frac{\sigma_z}{E} \\
\epsilon_y &= -\nu\frac{\sigma_x}{E} + \frac{\sigma_y}{E} - \nu\frac{\sigma_z}{E} \\
\epsilon_z &= -\nu\frac{\sigma_x}{E} - \nu\frac{\sigma_y}{E} + \frac{\sigma_z}{E} \\
\gamma_{yz} &= \frac{\tau_{yz}}{G} \\
\gamma_{xz} &= \frac{\tau_{xz}}{G} \\
\gamma_{xy} &= \frac{\tau_{xy}}{G}
\end{aligned}
\tag{1.11}
$$

The shear modulus (or modulus of rigidity), G, is given by

$$G = \frac{E}{2(1 + \nu)} \tag{1.12}$$

From Hooke's law relationships (Eq. 1.11), note that

$$\epsilon_x + \epsilon_y + \epsilon_z = \frac{(1 - 2v)}{E}(\sigma_x + \sigma_y + \sigma_z) \tag{1.13}$$

Substituting for $(\sigma_y + \sigma_z)$ and so on into Eq. 1.11, we get the inverse relations

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} \tag{1.14}$$

D is the symmetric (6×6) material matrix given by

$$\mathbf{D} = \frac{E}{(1 + v)(1 - 2v)} \begin{bmatrix} 1 - v & v & v & 0 & 0 & 0 \\ v & 1 - v & v & 0 & 0 & 0 \\ v & v & 1 - v & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 - v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 - v & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 - v \end{bmatrix} \tag{1.15}$$

Special Cases

One Dimension. In one dimension, we have normal stress σ along x and the corresponding normal strain $\boldsymbol{\epsilon}$. Stress–strain relations (Eq. 1.14) are simply

$$\sigma = E\epsilon \tag{1.16}$$

Two Dimensions. In two dimensions, the problems are modeled as plane stress and plane strain.

Plane Stress. A thin planar body subjected to in-plane loading on its edge surface is said to be in plane stress. A ring press fitted on a shaft (Fig. 1.5a) is an example. Here stresses σ_z, τ_{xz}, and τ_{yz} are set as zero. The Hooke's law relations (Eq. 1.11) then give us

$$\begin{aligned} \epsilon_x &= \frac{\sigma_x}{E} - v\frac{\sigma_y}{E} \\ \epsilon_y &= -v\frac{\sigma_x}{E} + \frac{\sigma_y}{E} \\ \gamma_{xy} &= \frac{2(1 + v)}{E}\tau_{xy} \\ \epsilon_z &= -\frac{v}{E}(\sigma_x + \sigma_y) \end{aligned} \tag{1.17}$$

The inverse relations are given by

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1 - v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1 - v}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{1.18}$$

which is used as $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$.

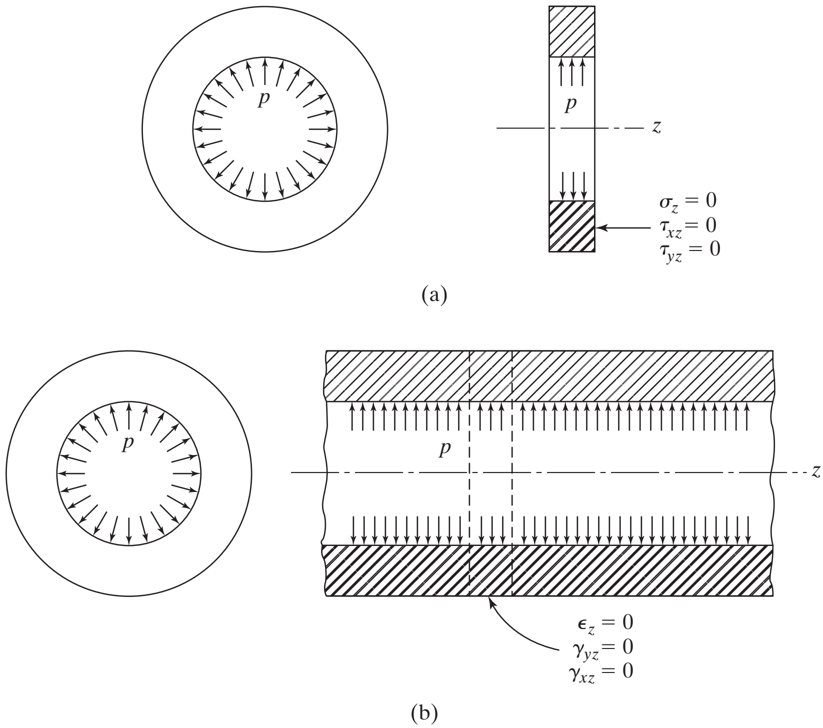


FIGURE 1.5 (a) Plane stress and (b) plane strain.

Plane Strain. If a long body of uniform cross section is subjected to transverse loading along its length, a small thickness in the loaded area, as shown in Fig. 1.5b, can be treated as subjected to plane strain. Here, ϵ_z, γ_{zx}, γ_{yz} are taken as zero. Stress σ_z may not be zero in this case. The stress–strain relations can be obtained directly from Eqs. 1.14 and 1.15:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{1.19}$$

Here, **D** is a (3×3) matrix, which relates three stresses and three strains.

Anisotropic bodies, with uniform orientation, can be considered by using the appropriate **D** matrix for the material.

1.8 TEMPERATURE EFFECTS

If the temperature rise $\Delta T(x, y, z)$ with respect to the original state is known, then the associated deformation can be considered easily. For isotropic materials, the temperature rise ΔT results in a uniform strain, which depends on the coefficient of linear expansion α of the material. α, which represents the change in unit length per unit temperature rise, is assumed to be a constant within the range of variation of the temperature. Also, this strain does not cause any stress when the body is free to deform. The temperature strain is represented as an initial strain:

$$\boldsymbol{\epsilon}_0 = [\alpha\Delta T, \alpha\Delta T, \alpha\Delta T, 0, 0, 0]^{\mathrm{T}} \tag{1.20}$$

The stress–strain relations then become

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) \tag{1.21}$$

In **plane stress**, we have

$$\boldsymbol{\epsilon}_0 = [\alpha\Delta T, \alpha\Delta T, 0]^{\mathrm{T}} \tag{1.22}$$

In **plane strain**, the constraint that $\epsilon_z = 0$ results in a different $\boldsymbol{\epsilon}_0$,

$$\boldsymbol{\epsilon}_0 = (1 + \nu)[\alpha\Delta T, \alpha\Delta T, 0]^{\mathrm{T}} \tag{1.23}$$

For plane stress and plane strain, note that $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \tau_{xy}]^{\mathrm{T}}$ and $\boldsymbol{\epsilon} = [\epsilon_x, \epsilon_y, \gamma_{xy}]^{\mathrm{T}}$, and that **D** matrices are as given in Eqs. 1.18 and 1.19, respectively.

1.9 POTENTIAL ENERGY AND EQUILIBRIUM: THE RAYLEIGH–RITZ METHOD

In mechanics of solids, our problem is to determine the displacement **u** of the body shown in Fig. 1.1, satisfying the equilibrium equations (Eq. 1.6). Note that stresses are related to strains, which in turn are related to displacements. This leads to requiring solution of second-order partial differential equations. Solution of this set of equations is generally referred to as an *exact* solution. Such exact solutions are available for simple geometries and loading conditions, and one may refer to publications in theory of elasticity. For problems of complex geometries and general boundary and loading conditions, obtaining such solutions is an almost impossible task. Approximate solution methods usually employ potential energy or variational methods, which place less stringent conditions on the functions.

Potential Energy, Π

The total potential energy Π of an elastic body is defined as the sum of total strain energy (U) and the work potential (WP):

$$\Pi = \underset{(U)}{\text{Strain energy}} + \underset{(\mathrm{WP})}{\text{Work potential}} \tag{1.24}$$

For linear elastic materials, the strain energy per unit volume in the body is $\frac{1}{2}\boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}$. For the elastic body shown in Fig. 1.1, the total strain energy U is given by

$$U = \frac{1}{2}\int_V \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}\,dV \tag{1.25}$$

The work potential (WP) is given by

$$\mathrm{WP} = -\int_V \mathbf{u}^{\mathrm{T}}\mathbf{f}\,dV - \int_S \mathbf{u}^{\mathrm{T}}\mathbf{T}\,dS - \sum_i \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i \tag{1.26}$$

The total potential for the general elastic body shown in Fig. 1.1 is

$$\Pi = \frac{1}{2}\int_V \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}\, dV - \int_V \mathbf{u}^{\mathrm{T}}\mathbf{f}\, dV - \int_S \mathbf{u}^{\mathrm{T}}\mathbf{T}\, dS - \sum_i \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i \tag{1.27}$$

We consider conservative systems here, where the work potential is independent of the path taken. In other words, if the system is displaced from a given configuration and brought back to this state, the forces do zero work regardless of the path. The potential energy principle is now stated as follows:

Principle of Minimum Potential Energy

For conservative systems, of all the kinematically admissible displacement fields, those corresponding to equilibrium extremize the total potential energy. If the extremum condition is a minimum, the equilibrium state is stable.

Kinematically admissible displacements are those that satisfy the single-valued nature of displacements (compatibility) and the boundary conditions. In problems where displacements are the unknowns, which is the approach used in this book, compatibility is automatically satisfied.

To illustrate the ideas, let us consider an example of a discrete connected system.

Example 1.1

Figure E1.1a shows a system of springs. The total potential energy is given by

$$\Pi = \tfrac{1}{2}k_1\delta_1^2 + \tfrac{1}{2}k_2\delta_2^2 + \tfrac{1}{2}k_3\delta_3^2 + \tfrac{1}{2}k_4\delta_4^2 - F_1q_1 - F_3q_3$$

where δ_1, δ_2, δ_3, and δ_4 are extensions of the four springs. Since $\delta_1 = q_1 - q_2$, $\delta_2 = q_2$, $\delta_3 = q_3 - q_2$, and $\delta_4 = -q_3$, we have

$$\Pi = \tfrac{1}{2}k_1(q_1 - q_2)^2 + \tfrac{1}{2}k_2q_2^2 + \tfrac{1}{2}k_3(q_3 - q_2)^2 + \tfrac{1}{2}k_4q_3^2 - F_1q_1 - F_3q_3$$

where q_1, q_2, and q_3 are the displacements of nodes 1, 2, and 3, respectively.

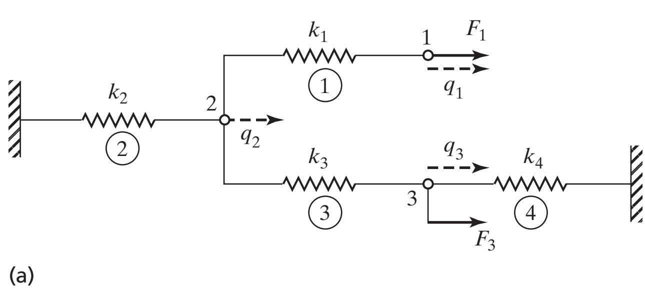


FIGURE E1.1a

For equilibrium of this three degrees of freedom system, we need to minimize Π with respect to q_1, q_2, and q_3. The three equations are given by

$$\frac{\partial \Pi}{\partial q_i} = 0 \quad i = 1, 2, 3 \tag{1.28}$$

which are

$$\frac{\partial \Pi}{\partial q_1} = k_1(q_1 - q_2) - F_1 = 0$$
$$\frac{\partial \Pi}{\partial q_2} = -k_1(q_1 - q_2) + k_2 q_2 - k_3(q_3 - q_2) = 0$$
$$\frac{\partial \Pi}{\partial q_3} = k_3(q_3 - q_2) + k_4 q_3 - F_3 = 0$$

These equilibrium equations can be put in the form of $\mathbf{Kq} = \mathbf{F}$ as follows:

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 + k_4 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ 0 \\ F_3 \end{Bmatrix} \tag{1.29}$$

If, on the other hand, we proceed to write the equilibrium equations of the system by considering the equilibrium of each separate node, as shown in Fig. E1.1b, we can write

$$k_1\delta_1 = F_1$$
$$k_2\delta_2 - k_1\delta_1 - k_3\delta_3 = 0$$
$$k_3\delta_3 - k_4\delta_4 = F_3$$

which is precisely the set of equations represented in Eq. 1.29.

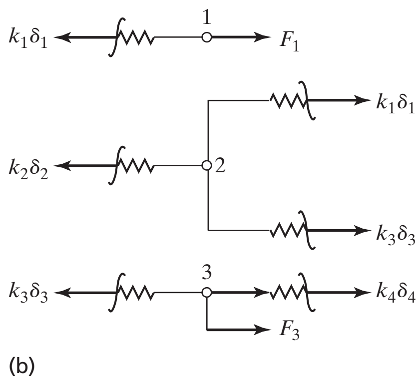


FIGURE E1.1b

We clearly see that the set of equations in Eq. 1.29 is obtained in a routine manner using the potential energy approach, without any reference to the free-body diagrams. This makes the potential energy approach attractive for large and complex problems. ■

Rayleigh–Ritz Method

For continua, the total potential energy Π in Eq. 1.27 can be used for finding an approximate solution. The Rayleigh–Ritz method involves the construction of an assumed displacement field, for example,

$$\begin{aligned}
u &= \sum a_i\phi_i(x, y, z) && i = 1 \text{ to } \ell \\
v &= \sum a_j\phi_j(x, y, z) && j = \ell + 1 \text{ to } m \\
w &= \sum a_k\phi_k(x, y, z) && k = m + 1 \text{ to } n \\
& && n > m > \ell
\end{aligned} \tag{1.30}$$

The functions ϕ_i are usually taken as polynomials. For example, $u = a_1 + a_2x + a_3x^2$ corresponds to $\phi_1 = 1, \phi_2 = x$, and $\phi_3 = x^2$ in Eq. 1.30. Displacements u, v, w must be **kinematically admissible**. That is, u, v, w must satisfy specified boundary conditions. Introducing stress–strain and strain–displacement relations and substituting Eq. 1.30 into Eq. 1.27 give

$$\Pi = \Pi(a_1, a_2, \ldots, a_r) \tag{1.31}$$

where r = number of independent unknowns. Now, the extremum with respect to $a_i, (i = 1 \text{ to } r)$ yields the set of r equations

$$\frac{\partial \Pi}{\partial a_i} = 0 \quad i = 1, 2, \ldots, r \tag{1.32}$$

Example 1.2

The potential energy for the linear elastic one-dimensional rod (Fig. E1.2), with body force neglected, is

$$\Pi = \frac{1}{2}\int_0^L EA\left(\frac{du}{dx}\right)^2 dx - 2u_1$$

where $u_1 = u(x = 1)$.

Let us consider a polynomial function

$$u = a_1 + a_2x + a_3x^2$$

This must satisfy $u = 0$ at $x = 0$ and $u = 0$ at $x = 2$. Thus,

$$\begin{aligned}
0 &= a_1 \\
0 &= a_1 + 2a_2 + 4a_3
\end{aligned}$$

Hence,

$$\begin{aligned}
a_2 &= -2a_3 \\
u &= a_3(-2x + x^2) \quad u_1 = -a_3
\end{aligned}$$

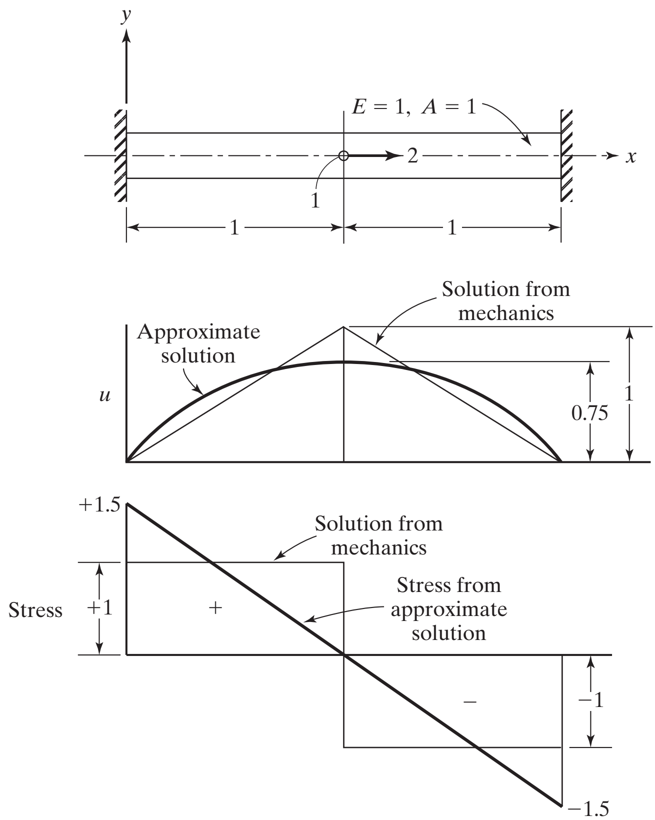


FIGURE E1.2

Then $du/dx = 2a_3(-1 + x)$ and

$$\Pi = \frac{1}{2}\int_0^2 4a_3^2(-1 + x)^2 dx - 2(-a_3)$$

$$= 2a_3^2 \int_0^2 (1 - 2x + x^2)dx + 2a_3$$

$$= 2a_3^2(\tfrac{2}{3}) + 2a_3$$

We set $\partial\Pi/\partial a_3 = 4a_3(\frac{2}{3}) + 2 = 0$, resulting in

$$a_3 = -0.75 \quad u_1 = -a_3 = 0.75$$

The stress in the bar is given by

$$\sigma = E\frac{du}{dx} = 1.5(1 - x)$$ ■

We note here that an exact solution is obtained if piecewise polynomial interpolation is used in the construction of u .

The finite element method provides a systematic way of constructing the basis functions ϕ_i used in Eq. 1.30.

1.10 GALERKIN'S METHOD

Galerkin's method uses the set of governing equations in the development of an integral form. It is usually presented as one of the weighted residual methods. For our discussion, let us consider a general representation of a governing equation on a region V :

$$Lu = P \tag{1.33}$$

For the one-dimensional rod considered in Example 1.2, the governing equation is the differential equation

$$\frac{d}{dx}\left(EA\frac{du}{dx}\right) = 0$$

We may consider L as the operator

$$\frac{d}{dx}EA\frac{d}{dx}()$$

operating on u .

The exact solution needs to satisfy (Eq. 1.33) at every point x. If we seek an approximate solution u, it introduces an error $\epsilon(x)$, called the *residual*:

$$\epsilon(x) = L\tilde{u} - P \tag{1.34}$$

The approximate methods revolve around setting the residual relative to a weighting function, W_i, to zero:

$$\int_V W_i(L\tilde{u} - P)dV = 0 \quad i = 1 \text{ to } n \tag{1.35}$$

The choice of the weighting function W_i leads to various approximation methods. *In Galerkin's method, the weighting functions W_i are chosen from the basis functions used for constructing $\tilde{u}$.* Let $\tilde{u}$ be represented by

$$\tilde{u} = \sum_{i=1}^{n} Q_i G_i \tag{1.36}$$

where $G_i, i = 1$ to n are basis functions (usually polynomials of x, y, z). Here, we choose the weighting functions to be *a linear combination of the basis functions G_i.* Specifically, consider an arbitrary function ϕ, given by

$$\phi = \sum_{i=1}^{n} \phi_i G_i \tag{1.37}$$

where the coefficients ϕ_i are arbitrary, except for requiring that ϕ satisfy homogeneous (zero) boundary conditions where $\tilde{u}$ is prescribed. The fact that ϕ in Eq. 1.37 is constructed in a similar manner as $\tilde{u}$ in Eq. 1.36 leads to simplified derivations in later chapters.

Galerkin's method can be stated as follows:

Choose basis functions G_i. Determine the coefficients Q_i in $\tilde{u} = \sum_{i=1}^{n} Q_i G_i$ such that

$$\int_V \phi(L\tilde{u} - P)dV = 0 \tag{1.38}$$

for every ϕ of the type $\phi = \sum_{i=1}^{n} \phi_i G_i$ where coefficients ϕ_i are arbitrary except for requiring that ϕ satisfy homogeneous (zero) boundary conditions. The solution of the resulting equations for Q_i then yields the approximate solution $\tilde{u}$.

Usually, in the treatment of Eq. 1.38, an integration by parts is involved. The order of the derivatives is reduced and the natural boundary conditions, such as surface force conditions, are introduced.

Example 1.3

Consider the differential equation

$$\frac{du}{dx} + 2u = 1, \quad 0 < x < 1$$

together with the initial condition $u(0) = 1$.

Note that $u(0)$ refers to the value of u at $x = 0$. If we let x to be time t, the above differential equation represents Newton's law of cooling a body with temperature u in an environment with lower temperature. Assume an approximation

$$u = a + bx + cx^2$$

Satisfaction of the initial condition gives $a = 1$. Thus

$$u = 1 + bx + cx^2$$

Galerkin's method requires

$$\int_0^1 W\left(\frac{du}{dx} + 2u - 1\right)dx = 0$$

$$W(0) = 0$$

with W constructed using the same basis functions used in approximating u, or

$$W = A + Bx + Cx^2$$

The weighting function must equal zero at $x = 0$ since u is specified at that point. Thus,

$$W = Bx + Cx^2$$

where B and C are arbitrary. Using integration by parts,

$$\int_0^1 W\frac{du}{dx}\,dx = -\int_0^1 u\frac{dW}{dx}\,dx + W(1)u(1) - W(0)u(0)$$

Since $W(0) = 0$, we have the "variational form"

$$-\int_0^1 u\frac{dW}{dx}\,dx + W(1)u(1) + 2\int_0^1 Wu dx - \int_0^1 W dx = 0$$

with $u(1) = 1 + b + c$. Substituting $W = Bx + Cx^2$ in the above equation, we get

$$Bf(a,b) + Cg(a,b) = 0$$

which must be true for arbitrary values of B and C (one may set $B = 1, C = 0$ and $B = 1$, $C = 0$). This implies that $f(a,b) = 0$ and $g(a,b) = 0$. The two equations are

$$1.667b + 1.667c = -0.5$$

$$0.8333b + 0.9c = -0.3333$$

On solving, we get $b = -0.7857, c = 0.3571$. We can compute the value of u for a given x using $u = 1 + bx + cx^2$ as

$$u(0.5) = 0.6964, \quad u(1) = 0.5714$$

The example shows how Galerkin's method provides approximate solutions starting directly from the differential equation. ■

Galerkin's Method in Elasticity. Let us turn our attention to the equilibrium equations (Eq. 1.6) in elasticity. Galerkin's method requires

$$\begin{aligned}\int_V \Bigg[\left(\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + f_x\right)\phi_x + \left(\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + f_y\right)\phi_y \\ + \left(\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + f_z\right)\phi_z\Bigg] dV = 0\end{aligned} \tag{1.39}$$

where

$$\boldsymbol{\phi} = [\phi_x, \phi_y, \phi_z]^{\mathrm{T}}$$

is an arbitrary displacement consistent with the boundary conditions of $\mathbf{u}$. If $\mathbf{n} = [n_x, n_y, n_z]^{\mathrm{T}}$ is a unit normal at a point $\mathbf{x}$ on the surface, the integration by parts formula is

$$\int_V \frac{\partial \alpha}{\partial x}\theta\, dV = -\int_V \alpha\frac{\partial \theta}{\partial x}\, dV + \int_S n_x \alpha\theta\, dS \tag{1.40}$$

where α and θ are functions of (x, y, z). For multidimensional problems, Eq. 1.40 is usually referred to as the Green–Gauss theorem or the divergence theorem. Using this formula, integrating Eq. 1.39 by parts, and rearranging terms, we get

$$-\int_V \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\epsilon}(\phi)\,dV + \int_V \boldsymbol{\phi}^{\mathrm{T}} \mathbf{f}\,dV + \int_S \big[(n_x\sigma_x + n_y\tau_{xy} + n_z\tau_{xz})\phi_x$$
$$+ (n_x\tau_{xy} + n_y\sigma_y + n_z\tau_{yz})\phi_y + (n_x\tau_{xz} + n_y\tau_{yz} + n_z\sigma_z)\phi_z\big]dS = 0 \quad (1.41)$$

where

$$\boldsymbol{\epsilon}(\phi) = \left[\frac{\partial\phi_x}{\partial x}, \frac{\partial\phi_y}{\partial y}, \frac{\partial\phi_z}{\partial z}, \frac{\partial\phi_y}{\partial z} + \frac{\partial\phi_z}{\partial y}, \frac{\partial\phi_x}{\partial z} + \frac{\partial\phi_z}{\partial x}, \frac{\partial\phi_x}{\partial y} + \frac{\partial\phi_y}{\partial x}\right]^{\mathrm{T}} \quad (1.42)$$

is the strain corresponding to the arbitrary displacement field $\boldsymbol{\phi}$.

On the boundary, from Eq. 1.8, we have $(n_x\sigma_x + n_y\tau_{xy} + n_z\tau_{xz}) = T_x$ and so on. At point loads $(n_x\sigma_x + n_y\tau_{xy} + n_z\tau_{xz})dS$ is equivalent to P_x and so on. These are the natural boundary conditions in the problem. Thus, Eq. 1.41 yields the Galerkin's "variational form" or "weak form" for three-dimensional stress analysis:

$$\int_V \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\epsilon}(\phi)\,dV - \int_V \boldsymbol{\phi}^{\mathrm{T}} \mathbf{f}\,dV - \int_S \boldsymbol{\phi}^{\mathrm{T}} \mathbf{T}\,dS - \sum_i \boldsymbol{\phi}^{\mathrm{T}} \mathbf{P} = 0 \quad (1.43)$$

where $\boldsymbol{\phi}$ is an arbitrary displacement consistent with the specified boundary conditions of $\mathbf{u}$. We may now use Eq. 1.43 to provide us with an approximate solution.

For problems of linear elasticity, Eq. 1.43 is precisely the **principle of virtual work**. $\boldsymbol{\phi}$ is the kinematically admissible virtual displacement. The principle of virtual work may be stated as follows:

Principle of Virtual Work

A body is in equilibrium if the internal virtual work equals the external virtual work for every kinematically admissible displacement field $\langle \boldsymbol{\phi}, \boldsymbol{\epsilon}(\phi) \rangle$.

We note that Galerkin's method and the principle of virtual work result in the same set of equations for problems of elasticity when same basis or coordinate functions are used. Galerkin's method is more general since the variational form of the type Eq. 1.43 can be developed for other governing equations defining boundary-value problems. Galerkin's method works directly from the differential equation and is preferred to the Rayleigh–Ritz method for problems where a corresponding function to be minimized is not obtainable.

Example 1.4

Let us consider the problem of Example 1.2 and solve it by Galerkin's approach. The equilibrium equation is

$$\frac{d}{dx}EA\frac{du}{dx} = 0 \qquad \begin{array}{ll} u = 0 & \text{at } x = 0 \\ u = 0 & \text{at } x = 2 \end{array}$$

Multiplying this differential equation by ϕ, and integrating by parts, we get

$$\int_0^2 -EA\frac{du}{dx}\frac{d\phi}{dx}dx + \left(\phi EA\frac{du}{dx}\right)_0^1 + \left(\phi EA\frac{du}{dx}\right)_1^2 = 0$$

where ϕ is zero at $x = 0$ and $x = 2$. $EA\,(du/dx)$ is the tension in the rod, which takes a jump of magnitude 2 at $x = 1$ (Fig. E1.2). Thus,

$$\int_0^2 -EA\frac{du}{dx}\frac{d\phi}{dx}dx + 2\phi_1 = 0$$

Now we use the same polynomial (basis) for u and ϕ. If u_1 and ϕ_1 are the values at $x = 1$, we have

$$u = (2x - x^2)u_1$$
$$\phi = (2x - x^2)\phi_1$$

Substituting these and $E = 1, A = 1$ in the previous integral yields

$$\phi_1\left[-u_1\int_0^2 (2 - 2x)^2 dx + 2\int_0^2\right] = 0$$

$$\phi_1\left(-\frac{8}{3}u_1 + 2\right) = 0$$

This is to be satisfied for every ϕ_1. We get

$$u_1 = 0.75$$ ■

1.11 SAINT VENANT'S PRINCIPLE

We often have to make approximations in defining boundary conditions to represent a support–structure interface. For example, consider a cantilever beam, free at one end and attached to a column with rivets at the other end. Questions arise as to whether the riveted joint is totally rigid or partially rigid, and as to whether each point on the cross section at the fixed end is specified to have the same boundary conditions. Saint Venant considered the effect of different approximations on the solution to the total problem. *Saint Venant's principle* states that as long as the different approximations are statically equivalent, the resulting solutions will be valid provided we focus on regions sufficiently far away from the support. That is, the solutions may significantly differ only within the immediate vicinity of the support.

1.12 VON MISES STRESS

Von Mises stress is used as a criterion in determining the onset of failure in ductile materials. The failure criterion states that the von Mises stress σ_{VM} should be less than the yield stress σ_Y of the material. In the inequality form, the criterion may be put as

$$\sigma_{VM} \leq \sigma_Y \tag{1.44}$$

The von Mises stress, σ_{VM} is given by

$$\sigma_{VM} = \sqrt{I_1^2 - 3I_2} \tag{1.45}$$

where I_1 and I_2 are the first two invariants of the stress tensor. For the general state of stress given by Eq. 1.5, I_1 and I_2 are given by

$$\begin{aligned} I_1 &= \sigma_x + \sigma_y + \sigma_z \\ I_2 &= \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_z\sigma_x - \tau_{yz}^2 - \tau_{xz}^2 - \tau_{xy}^2 \end{aligned} \tag{1.46}$$

In terms of the principal stresses σ_1, σ_2, and σ_3, the two invariants can be written as

$$\begin{aligned} I_1 &= \sigma_1 + \sigma_2 + \sigma_3 \\ I_2 &= \sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_1 \end{aligned}$$

It is easy to check that von Mises stress given in Eq. 1.45 can be expressed in the form

$$\sigma_{VM} = \frac{1}{\sqrt{2}}\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2} \tag{1.47}$$

For the state of plane stress, we have

$$\begin{aligned} I_1 &= \sigma_x + \sigma_y \\ I_2 &= \sigma_x\sigma_y - \tau_{xy}^2 \end{aligned} \tag{1.48}$$

and for plane strain

$$\begin{aligned} I_1 &= \sigma_x + \sigma_y + \sigma_z \\ I_2 &= \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_z\sigma_x - \tau_{xy}^2 \end{aligned} \tag{1.49}$$

where $\sigma_z = \nu(\sigma_x + \sigma_y)$.

1.13 PRINCIPLE OF SUPERPOSITION

Principle of superposition states that the combined response to a set of stimuli is the sum of individual responses to each of the stimuli. This is an important principle in linear theory. If a linear system is defined by $Au = v$, where v is the input or stimulus, A is the linear operator, and u is the output or response, the principle of superposition may be represented as

$$A(u_1 + u_2 + \cdots + u_n) = Au_1 + Au_2 + \cdots + Au_n \tag{1.50}$$

The principle of superposition satisfies both additive and homogeneous conditions represented by $A(c_1u_1 + c_2u_2) = c_1Au_1 + c_2Au_2$.

The principle of superposition applies to small deformations in elastic systems governed by Hooke's law presented earlier. In elastic structures, the combined effect of a number of loads is equal to the sum of the effects of each load acting separately. If displacements and stresses are known for each of the component loads, the combined displacements and stresses are obtained by taking their algebraic sum.

1.14 COMPUTER PROGRAMS

Computer use is an essential part of the finite element analysis. Well-developed, well-maintained, and well-supported computer programs are necessary in solving engineering problems and interpreting results. Many available commercial finite element packages fulfill these needs. It is also the trend in industry that the results are acceptable only when solved using certain standard computer program packages. The commercial packages provide user-friendly data-input platforms and elegant and easy-to-follow display formats. However, the packages do not provide an insight into the formulations and solution methods. Specially developed computer programs with available source codes enhance the learning process. We follow this philosophy in the development of this book. Every chapter is provided with computer programs that parallel the theory. The curious student should make an effort to see how the steps given in the theoretical development are implemented in the programs. Source codes are provided in C, Excel Visual Basic, FORTRAN, Javascript, MATLAB, and Visual Basic. These programs may be downloaded from the book website given on the back cover. An Excel VB program can be run like a spreadsheet and a Javascript program can be run on any web browser. Example input and output files are provided at the end of every chapter. Refer to the *Readme* included with the programs provides details on use of the programs. We encourage the use of commercial packages to supplement the learning process.

1.15 CONCLUSION

In this chapter, we have discussed the necessary background for the finite element method. We will discuss matrix algebra and techniques for solving a set of linear algebraic equations in the next chapter.

HISTORICAL REFERENCES

1. Hrenikoff, A., "Solution of problems in elasticity by the frame work method." *Journal of Applied Mechanics, Transactions of the ASME* 8: 169–175 (1941).
2. Courant, R., "Variational methods for the solution of problems of equilibrium and vibrations." *Bulletin of the American Mathematical Society* 49: 1–23 (1943).
3. Turner, M. J., R. W. Clough, H. C. Martin, and L. J. Topp, "Stiffness and deflection analysis of complex structures." *Journal of Aeronautical Science* 23(9): 805–824 (1956).

4. Clough, R. W., "The finite element method in plane stress analysis." *Proceedings American Society of Civil Engineers*, 2nd Conference on Electronic Computation, Pittsburgh, PA, 23: 345–378 (1960).
5. Argyris, J. H., "Energy theorems and structural analysis." *Aircraft Engineering*, 26: Oct.–Nov. (1954); 27: Feb.–May (1955).
6. Zienkiewicz, O. C., and Y. K. Cheung, *The Finite Element Method in Structural and Continuum Mechanics.* London: McGraw-Hill (1967).
7. Oden, J. T., *Finite Elements of Nonlinear Continua.* New York: McGraw-Hill (1972).

PROBLEMS

1.1. Obtain the **D** matrix given in Eq. 1.15 using the generalized Hooke's law relations (Eq. 1.11).

1.2. One of the two displacement curves is known to be the equilibrium solution for the cantilever beam shown. Circle the correct one.

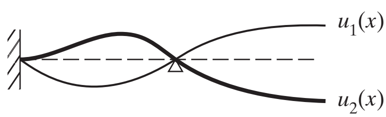


FIGURE P1.2

1.3. In a plane strain problem, we have

$$\sigma_x = 30{,}000 \text{ psi}, \sigma_y = -15{,}000 \text{ psi}$$

$$E = 30 \times 10^6 \text{ psi}, \nu = 0.3$$

Determine the value of the stress σ_z.

1.4. If a displacement field is described by

$$u = (x^2 + 4y^2 - 16xy)10^{-4}$$

$$v = (y^2 - 5x + 8y)10^{-4}$$

determine ϵ_x, ϵ_y, γ_{xy} at the point $x = 1, y = 0$.

1.5. Develop a deformation field $u(x, y), v(x, y)$ that describes the deformation of the finite element shown. From this determine ϵ_x, ϵ_y, γ_{xy}. Interpret your answer.

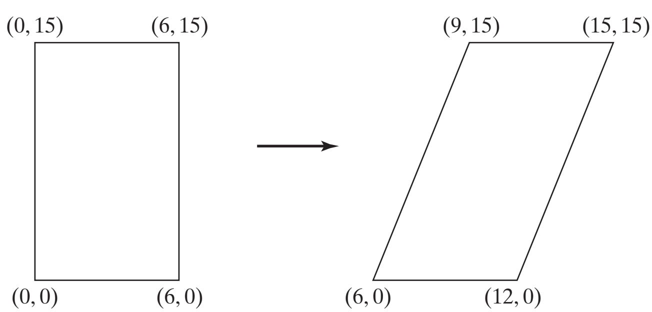


FIGURE P1.5

1.6. A displacement field

$$u = 2 + 2x + 4x^2 + 3xy^2$$
$$v = xy - 8x^2$$

is imposed on the square element shown in Fig. P1.6.

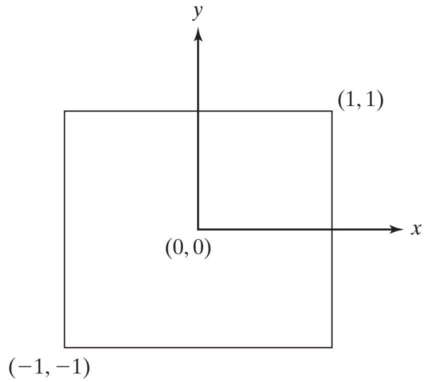


FIGURE P1.6

(a) Write down the expressions for ϵ_x, ϵ_y, and γ_{xy}.
(b) Plot contours of ϵ_x, ϵ_y, and γ_{xy} using, say, MATLAB software.
(c) Find where ϵ_x is maximum within the square.

1.7. A rectangular region $ABCD$ undergoes deformation to the state $A'B'C'D'$, as shown in Fig. P1.7.

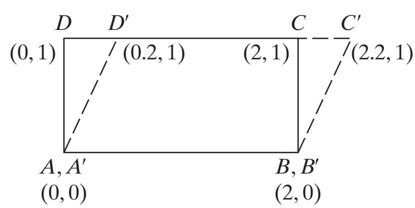


FIGURE P1.7

(a) Develop the displacement field that describes the above deformation by developing the functions $u(x, y)$, $v(x, y)$ that satisfy the specified corner values.
(b) From (a), determine expressions for ϵ_x, ϵ_y, and γ_{xy}.

1.8. In a solid body, the six components of the stress at a point are given by $\sigma_x = 120$ MPa, $\sigma_y = 55$ MPa, $\sigma_z = -85$ MPa, $\tau_{yz} = 33$ MPa, $\tau_{xz} = 75$ MPa, and $\tau_{xy} = -55$ MPa. Determine the normal stress at the point, on a plane for which the normal is $(n_x, n_y, n_z) = (\frac{1}{2}, \frac{1}{2}, 1/\sqrt{2})$. (*Hint:* Note that normal stress $\sigma_n = T_x n_x + T_y n_y + T_z n_z$.)

1.9. For isotropic materials, the stress–strain relations can also be expressed using Lame's constants λ and μ, as follows:

$$\sigma_x = \lambda\epsilon_v + 2\mu\epsilon_x$$

$$\sigma_y = \lambda\epsilon_v + 2\mu\epsilon_y$$
$$\sigma_z = \lambda\epsilon_v + 2\mu\epsilon_z$$
$$\tau_{yz} = \mu\gamma_{yz}, \tau_{xz} = \mu\gamma_{xz}, \tau_{xy} = \mu\gamma_{xy}$$

Here, $\epsilon_v = \epsilon_x + \epsilon_y + \epsilon_z$. Find expressions for λ and μ in terms of E and v.

1.10. A long rod is subjected to loading and a temperature increase of 45°C. The total strain at a point is measured to be 2×10^{-5}. If $E = 123$ GPa and $\alpha = 16.2 \times 10^{-6}/°C$, determine the stress at the point.

1.11. Consider the rod shown in Fig. P1.11, where the strain at any point x is given by $\epsilon_x = 5 + 4x^2$. Find the tip displacement δ.

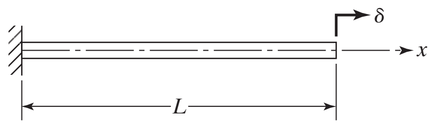


FIGURE P1.11

1.12. Determine the displacements of nodes of the spring system shown in Fig. P1.12.

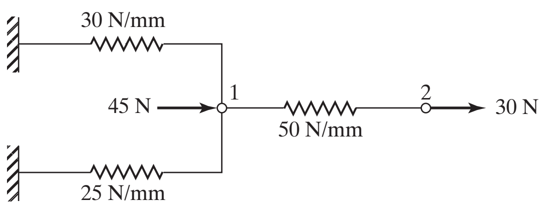


FIGURE P1.12

1.13. Write down the generalized Hooke's Law equations with temperature. Use these to analytically solve the following problem. A block that fits snugly within two rigid walls at room temperature T_0 is heated to $T_0 + \Delta T$. Determine the stress σ_y. Young's modulus of the block is E, Poisson's ratio is v, and the coefficient of linear expansion is α. Solve the problem under two different assumptions.

(a) The block is thin in the z-direction.
(b) The block is very thick in the z-direction.

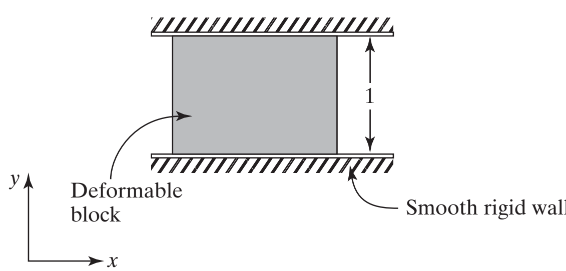


FIGURE P1.13

1.14. A block is press fitted into a rigid wall as shown in Fig. P1.14. Determine using the generalized Hooke's law, the stress σ_y. Take the Young's modulus of the block as E and the Poisson's ratio as ν. Assume that the block is thin in the z-direction. (Hint: Treat "$\alpha\,\Delta T$" in the y-direction as initial strain equal to $+0.1/1$

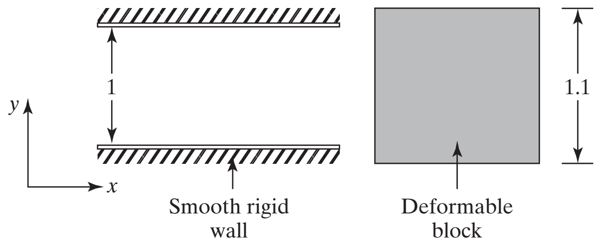


FIGURE P1.14

1.15. Use the Rayleigh–Ritz method to find the displacement of the midpoint of the rod shown in Fig. P1.15.

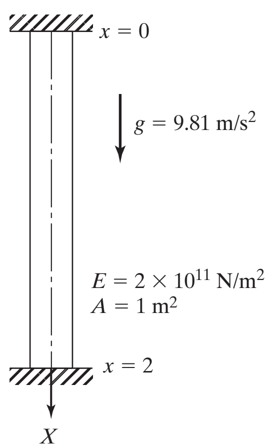


FIGURE P1.15

1.16. A rod fixed at its ends is subjected to a varying body force as shown in Fig. P1.16. Use the Rayleigh–Ritz method with an assumed displacement field $u = a_0 + a_1x + a_2x^2$ to determine displacement $u(x)$ and stress $\sigma(x)$.

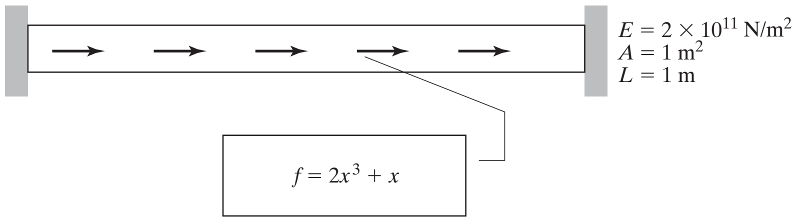


FIGURE P1.16

1.17. Use the Rayleigh–Ritz method to find the displacement field $u(x)$ of the rod in Fig. P1.17. Element 1 is made of aluminum, and element 2 is made of steel. The properties are

$$E_{al} = 97 \text{ GPa}, A_1 = 800 \text{ mm}^2, L_1 = 300 \text{ mm}$$

$$E_{st} = 123 \text{ GPa}, A_2 = 1000 \text{ mm}^2, L_2 = 500 \text{ mm}$$

Load $P = 8000$ N. Assume a piecewise linear displacement field $u = a_1 + a_2x$ for $0 \leq x \leq 300$ mm and $u = a_3 + a_4x$ for $300 \leq x \leq 800$ mm. Compare the Rayleigh–Ritz solution with the analytical strength of materials solution.

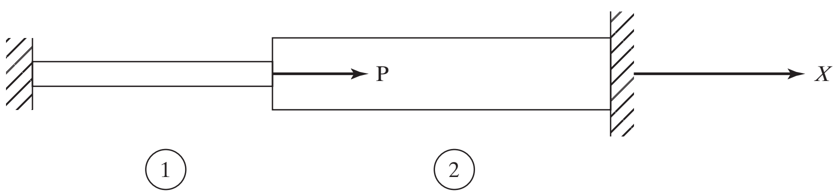
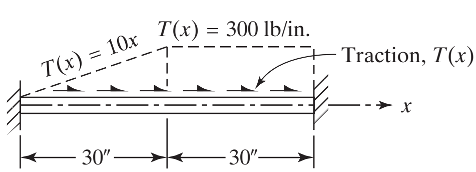


FIGURE P1.17

1.18. Use Galerkin's method to find the displacement at the midpoint of the rod (Fig. P1.15).

1.19. Solve Example 1.2 using the potential energy approach with the polynomial $u = a_1 + a_2x + a_3x^2 + a_4x^3$.

1.20. A steel rod is attached to rigid walls at each end and is subjected to a distributed load $T(x)$ as shown in Fig. P1.20.



$E = 30 \times 10^6$ psi $\quad A = 2$ in.^{2}

FIGURE P1.20

(a) Write the expression for the potential energy, Π.
(b) Determine the displacement $u(x)$ using the Rayleigh–Ritz method. Assume a displacement field $u(x) = a_0 + a_1x + a_2x^2$. Plot u versus x.
(c) Plot σ versus x.

1.21. Consider the functional I for minimization given by

$$I = \int_0^L \tfrac{1}{2}k\left(\frac{dy}{dx}\right)^2 dx + \tfrac{1}{2}h(a_0 - 800)^2$$

with $y = 20$ at $x = 60$. Given $k = 20$, $h = 25$, and $L = 60$, determine a_0, a_1, and a_2 using the polynomial approximation $y(x) = a_0 + a_1x + a_2x^2$ in the Rayleigh–Ritz method.

1.22. For the rod shown in Fig. P1.22, the boundary condition is $u_1 = 0$. Coordinates of the nodes are $x_1 = 0$, $x_2 = 1$, $x_3 = 3$. Two point loads are applied as shown. Use $E = 1$ and $A = 1$. Assume $u = a_0 + a_1x$, and obtain a Rayleigh–Ritz solution. Then plot u versus x and σ versus x.

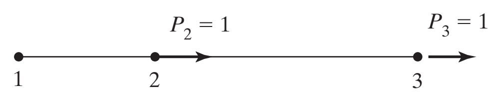


FIGURE P1.22

1.23. Consider the differential equation along with the initial condition

$$\frac{du}{dx} + 3u = x, \quad 0 \le x \le 1$$

$$u(0) = 1$$

Solve this using Galerkin's method assuming an initial approximation $u = a + bx + cx^2$.

1.24. Deflection v and slope v' $(=dv/dx)$ at $x = L$ for a cantilever beam are given in Fig. P1.24a. Determine the deflection and slope at $x = L$ for the loading shown in Fig. 1.24b using the principle of superposition.

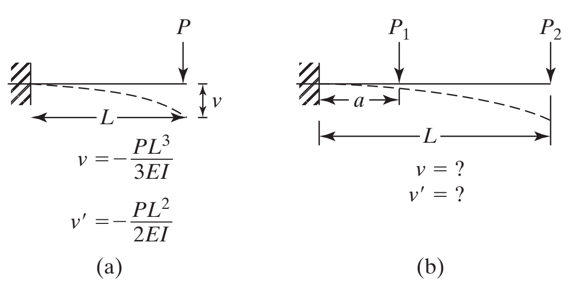


FIGURE P1.24

1.25. A square region undergoes deformation as shown in Fig. P1.25. The displacements x, y at the corners are given as $u = 0, v = 0$ at points A, C and D, and $u = -0.01, v = 0.01$ at B. x, y coordinates of the points are A (0, 0), B (1, 0), C (1, 1), D (0, 1), Q (0.5, 0.5).

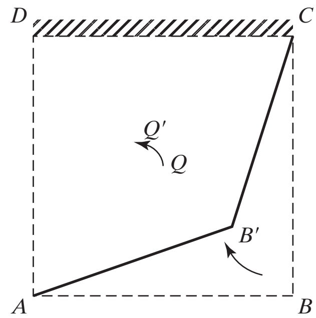


FIGURE P1.25

(a) Develop a displacement field that describes the above deformation. That is develop functions $u(x, y)$, $v(x, y)$ that satisfy the specified corner values.

(b) From (a), determine the shear strain at points B and Q.

C H A P T E R 2

Matrix Algebra and Gaussian Elimination

2.1 MATRIX ALGEBRA

The study of matrices here is largely motivated from the need to solve systems of simultaneous equations of the form

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
\tag{2.1a}
$$

where $x_1, x_2, \cdots, x_n$ are the unknowns. Equations 2.1a can be conveniently expressed in matrix form as

$$\mathbf{Ax} = \mathbf{b} \tag{2.1b}$$

where $\mathbf{A}$ is a square matrix of dimensions $(n \times n)$, and $\mathbf{x}$ and $\mathbf{b}$ are vectors of dimension $(n \times 1)$, given as

$$
\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \dots & a_{m} \end{bmatrix} \qquad \mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \qquad \mathbf{b} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix}
$$

From this information, we see that a matrix is simply an array of elements. The matrix $\mathbf{A}$ is also denoted as $[A]$. An element located at the ith row and jth column of $\mathbf{A}$ is denoted by a_{ij}.

The multiplication of two matrices, $\mathbf{A}$ and $\mathbf{x}$, is also implicitly defined: The dot product of the ith row of $\mathbf{A}$ with the vector $\mathbf{x}$ is equated to b_i, resulting in the ith equation of Eq. 2.1a. The multiplication operation and other operations will be discussed in detail in this chapter.

The analysis of engineering problems by the finite element method involves a sequence of matrix operations. This fact allows us to solve large-scale problems because computers are ideally suited for matrix operations. In this chapter, basic matrix operations are given as needed later in the text. The Gaussian elimination method for solving linear simultaneous equations is also discussed, and a variant of the Gaussian elimination approach, the skyline approach, is presented.

Row and Column Vectors

A matrix of dimension $(1 \times n)$ is called a **row vector**, while a matrix of dimension $(m \times 1)$ is called a **column vector**. For example,

$$\mathbf{d} = [1 \quad -1 \quad 2]$$

is a (1×3) row vector, and

$$\mathbf{e} = \begin{Bmatrix} 2 \\ 2 \\ -6 \\ 0 \end{Bmatrix}$$

is a (4×1) column vector.

Addition and Subtraction

Consider two matrices **A** and **B**, both of dimension $(m \times n)$. Then, the sum $\mathbf{C} = \mathbf{A} + \mathbf{B}$ is defined as

$$c_{ij} = a_{ij} + b_{ij} \tag{2.2}$$

That is, the (ij)th component of **C** is obtained by adding the (ij)th component of **A** to the (ij)th component of **B**. For example,

$$\begin{bmatrix} 2 & -3 \\ -3 & 5 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ -3 & -9 \end{bmatrix}$$

Subtraction is similarly defined.

Multiplication by a Scalar

The multiplication of a matrix **A** by a scalar c is defined as

$$c\mathbf{A} = [ca_{ij}] \tag{2.3}$$

For example, we can write

$$\begin{bmatrix} 10{,}000 & 4500 \\ 4500 & -6000 \end{bmatrix} = 10^3 \begin{bmatrix} 10 & 4.5 \\ 4.5 & -6 \end{bmatrix}$$

Matrix Multiplication

The product of an $(m \times n)$ matrix **A** and an $(n \times p)$ matrix **B** results in an $(m \times p)$ matrix **C**. That is,

$$\underset{(m \times n)}{\mathbf{A}} \quad \underset{(n \times p)}{\mathbf{B}} = \underset{(m \times p)}{\mathbf{C}} \tag{2.4}$$

The (ij)th component of **C** is obtained by taking the dot product

$$c_{ij} = (i\text{th row of } \mathbf{A}) \cdot (j\text{th column of } \mathbf{B}) \tag{2.5}$$

For example,

$$\begin{bmatrix} 2 & 1 & 3 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 5 & -2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 7 & 15 \\ -10 & 7 \end{bmatrix}$$

$$(2 \times 3) \qquad (3 \times 2) \qquad (2 \times 2)$$

It should be noted that $\mathbf{AB} \neq \mathbf{BA}$; in fact, $\mathbf{BA}$ may not even be defined, since the number of columns of $\mathbf{B}$ may not equal the number of rows of $\mathbf{A}$.

Transposition

If $\mathbf{A} = [a_{ij}]$, then the transpose of $\mathbf{A}$, denoted as $\mathbf{A}^{\mathrm{T}}$, is given by $\mathbf{A}^{\mathrm{T}} = [a_{ji}]$. Thus, the rows of $\mathbf{A}$ are the columns of $\mathbf{A}^{\mathrm{T}}$. For example, if

$$\mathbf{A} = \begin{bmatrix} 1 & -5 \\ 0 & 6 \\ -2 & 3 \\ 4 & 2 \end{bmatrix}$$

then

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & -2 & 4 \\ -5 & 6 & 3 & 2 \end{bmatrix}$$

In general, if $\mathbf{A}$ is of dimension $(m \times n)$, then $\mathbf{A}^{\mathrm{T}}$ is of dimension $(n \times m)$.

The transpose of a product is given as the product of the transposes in reverse order:

$$(\mathbf{ABC})^{\mathrm{T}} = \mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \tag{2.6}$$

Differentiation and Integration

The components of a matrix do not have to be scalars; they may also be functions. For example,

$$\mathbf{B} = \begin{bmatrix} x + y & x^2 - xy \\ 6 + x & y \end{bmatrix}$$

In this regard, matrices may be differentiated and integrated. The derivative (or integral) of a matrix is simply the derivative (or integral) of each component of the matrix. Thus,

$$\frac{d}{dx}\mathbf{B}(x) = \left[\frac{db_{ij}(x)}{dx}\right] \tag{2.7}$$

$$\int \mathbf{B}\, dx\, dy = \left[\int b_{ij}\, dx\, dy\right] \tag{2.8}$$

The formula in Eq. 2.7 will now be specialized to an important case. Let $\mathbf{A}$ be an $(n \times n)$ matrix of constants, and $\mathbf{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$ be a column vector of n variables. Then, the derivative of $\mathbf{Ax}$ with respect to variable x_p is given by

$$\frac{d}{dx_p}(\mathbf{Ax}) = \mathbf{a}^p \tag{2.9}$$

where $\mathbf{a}^p$ is the pth column of $\mathbf{A}$. This result follows from the fact that the vector $(\mathbf{Ax})$ can be written out in full as

$$\mathbf{Ax} = \begin{Bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2p}x_p + \cdots + a_{2n}x_z \\ ------------------------------ \\ ------------------------------ \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{np}x_p + \cdots + a_{nn}x_n \end{Bmatrix} \tag{2.10}$$

Now, we see clearly that the derivative of $\mathbf{Ax}$ with respect to x_p yields the pth column of $\mathbf{A}$ as stated in Eq. 2.9.

Square Matrix

A matrix whose number of rows equals the number of columns is called a square matrix.

Diagonal Matrix

A diagonal matrix is a square matrix with nonzero elements only along the principal diagonal. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & -3 \end{bmatrix}$$

Identity Matrix

The identity (or unit) matrix is a diagonal matrix with 1's along the principal diagonal. For example,

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If $\mathbf{I}$ is of dimension $(n \times n)$ and $\mathbf{x}$ is an $(n \times 1)$ vector, then

$$\mathbf{Ix} = \mathbf{x}$$

Symmetric Matrix

A symmetric matrix is a square matrix whose elements satisfy

$$a_{ij} = a_{ji} \tag{2.11a}$$

or equivalently,

$$\mathbf{A} = \mathbf{A}^{\mathrm{T}} \tag{2.11b}$$

That is, elements located symmetrically with respect to the principal diagonal are equal. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 6 & -2 \\ 0 & -2 & 8 \end{bmatrix}$$

Upper Triangular Matrix

An upper triangular matrix is one whose elements below the principal diagonal are all zero. For example,

$$\mathbf{U} = \begin{bmatrix} 2 & -1 & 6 & 3 \\ 0 & 14 & 8 & 0 \\ 0 & 0 & 5 & 1 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

Determinant of a Matrix

The determinant of a square matrix $\mathbf{A}$ is a scalar quantity denoted as det $\mathbf{A}$. The determinants of a (2×2) and a (3×3) matrix are given by the **method of cofactors** as follows:

$$\det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{21}a_{12} \tag{2.12}$$

$$\det\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} = a_{11}(a_{22}a_{33} - a_{32}a_{23}) - a_{12}(a_{23}a_{33} - a_{31}a_{23}) + a_{13}(a_{21}a_{32} - a_{31}a_{22}) \tag{2.13}$$

Matrix Inversion

Consider a square matrix $\mathbf{A}$. If det $\mathbf{A} \neq 0$, then $\mathbf{A}$ has an inverse, denoted by $\mathbf{A}^{-1}$. The inverse satisfies the relations

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \tag{2.14}$$

If det $\mathbf{A} \neq 0$, then we say that $\mathbf{A}$ is **nonsingular**. If det $\mathbf{A} = 0$, then we say that $\mathbf{A}$ is **singular**, for which the inverse is not defined. The minor M_{ij} of a square matrix $\mathbf{A}$ is the determinant of the $(n-1 \times n-1)$ matrix obtained by eliminating the ith row and the jth column of $\mathbf{A}$. The cofactor C_{ij} of matrix $\mathbf{A}$ is given by

$$C_{ij} = (-1)^{i+j} M_{ij}$$

Matrix $\mathbf{C}$ with elements C_{ij} is called the cofactor matrix. The adjoint of matrix $\mathbf{A}$ is defined as

$$\text{Adj}\,\mathbf{A} = \mathbf{C}^{\mathrm{T}}$$

The inverse of a square matrix $\mathbf{A}$ is given as

$$\mathbf{A}^{-1} = \frac{\text{adj}\,\mathbf{A}}{\det\,\mathbf{A}}$$

For example, the inverse of a (2×2) matrix $\mathbf{A}$ is given by

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{\det\mathbf{A}}\begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & -a_{11} \end{bmatrix}$$

Quadratic Forms. Let $\mathbf{A}$ be an $(n \times n)$ matrix and $\mathbf{x}$ be an $(n \times 1)$ vector. Then, the scalar quantity

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} \tag{2.15}$$

is called a quadratic form, since upon expansion we obtain the quadratic expression

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \begin{array}{llll} x_1 a_{11} x_1 & + x_1 a_{12} x_2 & + \cdots & + x_1 a_{1n} x_n \\ + x_2 a_{21} x_1 & + x_2 a_{22} x_2 & + \cdots & + x_2 a_{2n} x_n \\ \hline + x_n a_{n1} x_1 & + x_n a_{n2} x_2 & + \cdots & + x_n a_{nn} x_n \end{array} \tag{2.16}$$

As an example, the quantity

$$u = 3x_1^2 - 4x_1x_2 + 6x_1x_3 - x_2^2 + 5x_3^2$$

can be expressed in matrix form as

$$\begin{aligned} u &= [x_1 \;\; x_2 \;\; x_3]\begin{bmatrix} 3 & -2 & 3 \\ -2 & -1 & 0 \\ 3 & 0 & 5 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \\ &= \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} \end{aligned}$$

Eigenvalues and Eigenvectors

Consider the eigenvalue problem

$$\mathbf{A}\mathbf{y} = \lambda\mathbf{y} \tag{2.17a}$$

where $\mathbf{A}$ is a square matrix, $(n \times n)$. We wish to find a nontrivial solution. That is, we wish to find a nonzero eigenvector $\mathbf{y}$ and the corresponding eigenvalue λ that satisfy Eq. 2.17a. If we rewrite Eq. 2.17a as

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{y} = \mathbf{0} \tag{2.17b}$$

we see that a nonzero solution for $\mathbf{y}$ will occur when $\mathbf{A} - \lambda\mathbf{I}$ is a singular matrix or

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \tag{2.18}$$

Equation 2.18 is called the *characteristic equation.* We can solve Eq. 2.18 for the n roots or eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$. For each eigenvalue λ_i, the associated eigenvector $\mathbf{y}^i$ is then obtained from Eq. 2.17b:

$$(\mathbf{A} - \lambda_i\mathbf{I})\mathbf{y}^i = 0 \tag{2.19}$$

Note that the eigenvector $\mathbf{y}^i$ is determined only to within a multiplicative constant since $(\mathbf{A} - \lambda_i\mathbf{I})$ is a singular matrix.

Example 2.1

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 4 & -2.236 \\ -2.236 & 8 \end{bmatrix}$$

The characteristic equation is

$$\det\begin{bmatrix} 4-\lambda & -2.236 \\ -2.236 & 8-\lambda \end{bmatrix} = 0$$

which yields

$$(4 - \lambda)(8 - \lambda) - 5 = 0$$

Solving this above equation, we get

$$\lambda_1 = 3 \qquad \lambda_2 = 9$$

To get the eigenvector $\mathbf{y}^1 = [y_1^1, y_2^1]^{\mathrm{T}}$ corresponding to the eigenvalue λ_1, we substitute $\lambda_1 = 3$ into Eq. 2.19:

$$\begin{bmatrix} (4-3) & -2.236 \\ -2.236 & (8-3) \end{bmatrix}\begin{Bmatrix} y_1^1 \\ y_2^1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Thus, the components of $\mathbf{y}^1$ satisfy the equation

$$y_1^1 - 2.236y_2^1 = 0$$

We may now *normalalize* the eigenvector, say, by making $\mathbf{y}^1$ a unit vector. This is done by setting $y_2^1 = 1$, resulting in $\mathbf{y}^1 = [2.236, 1]$. Dividing $\mathbf{y}^1$ by its length yields

$$\mathbf{y}^1 = [0.913, 0.408]^{\mathrm{T}}$$

Now, $\mathbf{y}^2$ is obtained in a similar manner by substituting λ_2 into Eq. 2.19. After normalization

$$\mathbf{y}^2 = [-0.408, 0.913]^{\mathrm{T}}$$ ■

Eigenvalue problems in finite element analysis are of the type $\mathbf{Ay} = \lambda\mathbf{By}$. Solution techniques for these problems are discussed in Chapter 11.

Positive Definite Matrix

A symmetric matrix is said to be **positive definite** if all its eigenvalues are strictly positive (greater than zero). In the previous example, the symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 4 & -2.236 \\ -2.236 & 8 \end{bmatrix}$$

had eigenvalues $\lambda_1 = 3 > 0$ and $\lambda_2 = 9 > 0$ and, hence, is positive definite. An alternative definition of a positive definite matrix is as follows:

A symmetric matrix $\mathbf{A}$ of dimension $(n \times n)$ is positive definite if, for *any* nonzero vector $\mathbf{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$,

$$\mathbf{x}^{\mathrm{T}}\mathbf{Ax} > 0 \tag{2.20}$$

Cholesky Decomposition

A positive definite symmetric matrix $\mathbf{A}$ can be decomposed into the form

$$\mathbf{A} = \mathbf{LL}^{\mathrm{T}} \tag{2.21}$$

where $\mathbf{L}$ is a lower triangular matrix, and its transpose $\mathbf{L}^{\mathrm{T}}$ is upper triangular. This is *Cholesky decomposition*. The elements of $\mathbf{L}$ are calculated using the following steps: The evaluation of elements in row k does not affect the elements in the previously evaluated $k - 1$ rows. The decomposition is performed by evaluating rows from $k = 1$ to n as follows:

$$l_{kj} = \frac{\left(a_{kj} - \sum_{i=1}^{j-1} l_{ki} l_{kj}\right)}{l_{jj}} \qquad j = 1 \text{ } to \text{ } k - 1 \tag{2.22}$$

$$l_{kk} = \sqrt{a_{kk} - \sum_{i=1}^{j-1} l_{ki}{}^2}$$

In this evaluation, the summation is not carried out when the upper limit is less than the lower limit.

The inverse of a lower triangular matrix is a lower triangular matrix. The diagonal elements of the inverse $\mathbf{L}^{-1}$ are inverse of the diagonal elements of $\mathbf{L}$. Noting this, for a given $\mathbf{A}$, its decomposition $\mathbf{L}$ can be stored in the lower triangular part of $\mathbf{A}$ and the elements below the diagonal of $\mathbf{L}^{-1}$ can be stored above the diagonal in $\mathbf{A}$.

2.2 GAUSSIAN ELIMINATION

Consider a linear system of simultaneous equations in matrix form as

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ is $(n \times n)$ and $\mathbf{b}$ and $\mathbf{x}$ are $(n \times 1)$. If $\mathbf{A} \neq 0$, then we can premultiply both sides of the equation by $\mathbf{A}^{-1}$ to write the unique solution for $\mathbf{x}$ as $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. However, the explicit construction of $\mathbf{A}^{-1}$, say, by the cofactor approach, is computationally expensive and prone to round-off errors. Instead, an elimination scheme is better. The powerful Gaussian elimination approach for solving $\mathbf{Ax} = \mathbf{b}$ is discussed in the following pages.

Gaussian elimination is the name given to a well-known method of solving simultaneous equations by successively eliminating unknowns. We will first present the method by means of an example, followed by a general solution and algorithm. Consider the simultaneous equations

$$\begin{aligned} x_1 - 2x_2 + 6x_3 &= 0 \qquad &(\text{I}) \\ 2x_1 + 2x_2 + 3x_3 &= 3 \qquad &(\text{II}) \\ -x_1 + 3x_3 \qquad &= 2 \qquad &(\text{III}) \end{aligned} \tag{2.23}$$

The equations are labeled as I, II, and III. Now, we wish to eliminate x_1 from II and III. We have, from Eq. I, $x_1 = +2x_2 - 6x_3$. Substituting for x_1 into Eqs. II and III yields

$$\begin{aligned} x_1 - 2x_2 + 6x_3 &= 0 \qquad &(\text{I}) \\ 0 + 6x_2 - 9x_3 &= 3 \qquad &(\text{II}^{(1)}) \\ -0 + x_2 + 6x_3 &= 2 \qquad &(\text{III}^{(1)}) \end{aligned} \tag{2.24}$$

It is important to realize that Eq. 2.24 can also be obtained from Eq. 2.23 by **row operations**. Specifically, in Eq. 2.23, to eliminate x_1 from II, we subtract 2 times I from II, and to eliminate x_1 from III, we subtract -1 times I from III. The result is Eq. 2.24. Notice the zeroes below the main diagonal in column 1, representing the fact that x_1 has been eliminated from Eqs. II and III. The superscript 1 on the labels in Eqs. 2.24 denotes the fact that the equations have been modified once.

We now proceed to eliminate x_2 from III in Eqs. 2.24. For this, we subtract 1/6 times II from III. The resulting system is

$$\begin{aligned} x_1 - 2x_2 + 6x_3 &= 0 \qquad &(\text{I}) \\ 0 + 6x_2 - 9x_3 &= 3 \qquad &(\text{II}^{(1)}) \\ 0 + 0 \qquad \tfrac{15}{2}x_3 &= \tfrac{3}{2} \qquad &(\text{III}^{(2)}) \end{aligned} \tag{2.25}$$

The coefficient matrix on the left side of Eqs. 2.25 is upper triangular. The solution now is virtually complete, since the last equation yields $x_3 = \frac{1}{5}$, which, upon substitution into the second equation, yields $x_2 = \frac{4}{5}$, and then $x_1 = \frac{2}{5}$ from the first equation. This process of obtaining the unknowns in reverse order is called **back-substitution**.

These operations can be expressed more concisely in matrix form as follows: Working with the augmented matrix $[\mathbf{A}, \mathbf{b}]$, the Gaussian elimination process is

$$\begin{bmatrix} 1 & -2 & 6 & 0 \\ 2 & 2 & 3 & 3 \\ -1 & 3 & 0 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -2 & 6 & 0 \\ 0 & 6 & -9 & 3 \\ 0 & 1 & 6 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -2 & 6 & 0 \\ 0 & 6 & -9 & 3 \\ 0 & 0 & 15/2 & 3/2 \end{bmatrix} \tag{2.26}$$

which, upon back-substitution, yields

$$x_3 = \tfrac{1}{5} \qquad x_2 = \tfrac{4}{5} \qquad x_1 = \tfrac{2}{5} \tag{2.27}$$

General Algorithm for Gaussian Elimination

We just discussed the Gaussian elimination process by means of an example. This process will now be stated as an algorithm, suitable for computer implementation.

Let the original system of equations be as given in Eqs. 2.1, which can be restated as

$$
\begin{array}{rc}
\begin{array}{r} \\ \\ \\ \\ \text{Row } i \\ \\ \\ \end{array} &
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2j} & \cdots & a_{2n} \\
a_{31} & a_{32} & a_{33} & \cdots & a_{3j} & \cdots & a_{3n} \\
\text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \text{—} \\
a_{i1} & a_{i2} & a_{i3} & \cdots & a_{ij} & \cdots & a_{in} \\
\text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \text{—} \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nj} & \cdots & a_{nm}
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{Bmatrix}
=
\begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{Bmatrix} \\
& \text{Column} \\
& j
\end{array}
\tag{2.28}
$$

Gaussian elimination is a systematic approach to successively eliminate variables $x_1, x_2, x_3, \ldots, x_{n-1}$ until only one variable, x_n, is left. This results in an upper triangular matrix with reduced coefficients and reduced right side. This process is called forward elimination. It is then easy to determine $x_n, x_{n-1}, \ldots, x_3, x_2, x_1$ successively by the process of back-substitution. Let us consider the start of step 1, with $\mathbf{A}$ and $\mathbf{b}$ written as follows:

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2j} & \cdots & a_{2n} \\
\vdots & \vdots & \vdots & \text{—} & \vdots & \text{—} & \vdots \\
a_{i1} & a_{i2} & a_{i3} & \cdots & a_{ij} & \cdots & a_{in} \\
\text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \text{—} \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nj} & \cdots & a_{nm}
\end{bmatrix}
\quad
\begin{array}{c} \text{Start} \\ \text{of step} \\ k = 1 \end{array}
\begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{Bmatrix}
\tag{2.29}
$$

The idea at step 1 is to use equation 1 (the first row) in eliminating x_1 from remaining equations. We denote the step number as a superscript set in parentheses. The reduction process at step 1 is

$$
a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}} \cdot a_{1j}
$$

and (2.30)

$$
b_i^{(1)} = b_i - \frac{a_{i1}}{a_{11}} \cdot b_1
$$

We note that the ratios a_{i1}/a_{11} are simply the row multipliers that were referred to in the example discussed previously. Also, a_{11} is referred to as a *pivot*. The reduction is carried out for all the elements in the shaded area in Eq. 2.29 for which i and j range from 2 to n. The elements in rows 2 to n of the first column are zeroes since x_1 is eliminated. In the computer implementation, we need not set them to zero, but they are zeroes for our consideration. At the start of step 2, we thus have

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & a_{1j} & \cdots & a_{1n} \\
0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2j}^{(1)} & \cdots & a_{2n}^{(1)} \\
0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3j}^{(1)} & \cdots & a_{3n}^{(1)} \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
0 & a_{i2}^{(1)} & a_{i3}^{(1)} & \cdots & a_{ij}^{(1)} & \cdots & a_{in}^{(1)}\text{a} \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nj}^{(1)} & \cdots & a_{nn}^{(1)}
\end{bmatrix}
\quad \begin{array}{c}\text{Start of}\\ \text{step } k = 2\end{array} \quad
\begin{bmatrix}
b_1 \\ b_2^{(1)} \\ b_3^{(1)} \\ \vdots \\ b_i^{(1)} \\ \vdots \\ b_n^{(1)}
\end{bmatrix}
\tag{2.31}
$$

The elements in the shaded area in Eq. 2.31 are reduced at step 2. We now show the start of step k and the operations at step k in

$$
\begin{array}{cc}
 & \\
 & \\
 & \\
 & \\
\text{Row } i & \\
 &
\end{array}
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & \cdots & \cdots & a_{1j} & \cdots & a_{1n} \\
0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & \cdots & \cdots & a_{2j}^{(1)} & \cdots & a_{2n}^{(1)} \\
0 & 0 & a_{33}^{(1)} & \cdots & \cdots & \cdots & a_{3j}^{(2)} & \cdots & a_{3n}^{(2)} \\
0 & 0 & 0 & \cdots & a_{k+1,\,k+1}^{(k-1)} & \cdots & a_{k+1,\,j}^{(k-1)} & \cdots & a_{k-1,n}^{(k-1)} \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & a_{i,\,k+1}^{(k-1)} & \cdots & a_{ij}^{(k-1)} & \cdots & a_{in}^{(k-1)} \\
0 & 0 & 0 & \cdots & a_{n,\,k+1}^{(k-1)} & \cdots & a_{nj}^{(k-1)} & \cdots & a_{nn}^{(k-1)}
\end{bmatrix}
\quad \begin{array}{c}\text{Start of}\\ \text{step } k\end{array} \quad
\begin{Bmatrix}
b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_{k+1}^{(k-1)} \\ \vdots \\ b_i^{(k-1)} \\ \cdots \\ b_n^{(k-1)}
\end{Bmatrix}
$$

$$\begin{array}{c}\text{Column}\\ j\end{array} \tag{2.32}$$

At step k, elements in the shaded area are reduced. The general reduction scheme with limits on indices may be put as follows:

In step k,

$$
\begin{aligned}
a_{ij}^{(k)} &= a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} \qquad i, j = k + 1, \ldots, n \\
b_i^{(k)} &= b_{ij}^{(k-1)} - \frac{a_{ij}^{(k-1)}}{a_{kk}^{(k-1)}} b_{kj}^{(k-1)} \qquad i = k + 1, \ldots, n
\end{aligned}
\tag{2.33}
$$

After $(n - 1)$ steps, we get

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\
 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & \cdots & a_{2n}^{(1)} \\
 & & a_{33}^{(2)} & a_{34}^{(2)} & \cdots & a_{3n}^{(2)} \\
 & & & a_{44}^{(3)} & \cdots & a_{4n}^{(3)} \\
 & 0 & & & & \\
 & & & & & a_{nn}^{(n-1)}
\end{bmatrix}
\begin{Bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n
\end{Bmatrix}
=
\begin{Bmatrix}
b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ b_4^{(3)} \\ \vdots \\ b_n^{(n-1)}
\end{Bmatrix}
\tag{2.34}
$$

The superscripts are for the convenience of presentation. In the computer implementation, these superscripts can be avoided. We now drop the superscripts for convenience, and the back-substitution process is given by

$$x_n = \frac{b_n}{a_{nn}} \tag{2.35}$$

and then

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}} \qquad i = n-1, n-2, \cdots, 1 \tag{2.36}$$

This completes the Gauss elimination algorithm.

The algorithm discussed earlier is given below in the form of computer logic.

Algorithm 1: General Matrix

Forward elimination (reduction of A, b)

```
DO  k = 1, n − 1
   DO  i = k + 1, n
          c = a_ik / a_kk
      DO  j = k + 1, n
          a_ij = a_ij − c a_kj
          b_i = b_i − c b_k
```

Back-substitution

```
b_n = b_n / a_nn
DO  ii = 1, n − 1
   i = n − ii
   sum = 0
   DO  j = i + 1, n
      sum = sum + a_ij b_j
   b_i = (b_i − sum) / a_ii
```

[*Note:* **b** contains the solution to **Ax** = **b**.]

The algorithm is implemented in the program GAUSS.

Symmetric Matrix

If **A** is symmetric, then the previous algorithm needs two modifications. One is that the multiplier is defined as

$$c = \frac{a_{ki}}{a_{kk}} \tag{2.37}$$

The other modification is related to the DO LOOP index (the third DO LOOP in the previous algorithm):

$$\text{DO} \quad j = i, n \tag{2.38}$$

Symmetric Banded Matrices

In a **banded** matrix, all of the nonzero elements are contained within a band; outside of the band all elements are zero. The stiffness matrix that we will come across in subsequent chapters is a symmetric and banded matrix.

Consider an $(n \times n)$ symmetric banded matrix:

$$
\left[\begin{array}{ccccccccccc}
\multicolumn{5}{l}{\overbrace{X\;\;X\;\;X\;\;X\;\;X}^{\text{nbw}}} & & & & & & \\
 & X & X & X & X & X & & & 0 & & \\
 & & X & X & X & X & X & & & & \\
 & & & X & X & X & X & X & & & \\
 & & & & X & X & X & X & X & & \\
 & & \text{Symmetric} & & & X & X & X & X & X & \\
 & & & & & & X & X & X & X & X \\
 & & & & & & & X & X & X & X \\
 & & & & & & & & X & X & X \\
 & & & & & & & & & X & X \\
 & & & & & & & & & & X
\end{array}\right]
\begin{array}{l}
\\ \\ \\ \\ \\ \\ \\ \\ \\
\text{2nd diagonal} \\
\text{Main (1st) diagonal}
\end{array}
\tag{2.39}$$

In Eq. 2.39, nbw is called the **half-bandwidth**. Since only the nonzero elements need to be stored, the elements of this matrix are compactly stored in the $(n \times \text{nbw})$ matrix as follows:

$$
\begin{array}{c}
\begin{array}{ccccc}
\text{1st} & \text{2nd} & & & \text{nbw}
\end{array} \\
\begin{bmatrix}
X & X & X & X & X \\
X & X & X & X & X \\
X & X & X & X & X \\
X & X & X & X & X \\
X & X & X & X & X \\
X & X & X & X & X \\
X & X & X & X & X \\
X & X & X & X & \\
X & X & X & & \\
X & X & & 0 & \\
X & & & &
\end{bmatrix}
\end{array}
\tag{2.40}$$

The principal diagonal or 1st diagonal of Eq. 2.39 is the first column of Eq. 2.40. In general, the pth diagonal of Eq. 2.39 is stored as the pth column of Eq. 2.40 as shown. The correspondence between the elements of Eqs. 2.39 and 2.40 is given by

$$\left.\begin{matrix} a_{ij} \\ (j > i) \end{matrix}\right|_{(2.39)} = a_{i(j-i+1)}\Big|_{(2.40)} \tag{2.41}$$

Also, we note that $a_{ij} = a_{ji}$ in Eq. 2.39, and that the number of elements in the kth row of Eq. 2.40 is min $(n - k + 1, \text{nbw})$. We can now present the Gaussian elimination algorithm for symmetric banded matrix.

Algorithm 2: Symmetric Banded Matrix

Forward elimination

DO $k = 1, n - 1$
  nbk $= \min(n - k + 1, \text{nbw})$
  DO $i = k + 1, \text{nbk} + k - 1$
    $i1 = i - k + 1$
    $c = a_{k,\,i1}/a_{k,1}$
    DO $j = i, \text{nbk} + k - 1$
      $j1 = j - i + 1$
      $j2 = j - k + 1$
      $a_{i,j1} = a_{i,j1} - c a_{k,j2}$
    $b_i = b_i - c b_k$

Back-substitution

$$b_n = \frac{b_n}{a_{n,1}}$$

DO $ii = 1, n - 1$
  $i = n - ii$
  nbi $= \min(n - i + 1, \text{nbw})$
  sum $= 0$
  DO $j = 2, \text{nbi}$
    sum $=$ sum $+ a_{i,j} b_{i+j-1}$
  $b_i = \dfrac{b_i - \text{sum}}{a_{i,1}}$

[*Note:* The DO LOOP indices are based on the original matrix as in Eq. 2.39; the correspondence in Eq. 2.41 is then used while referring to elements of the banded matrix **A**. Alternatively, it is possible to express the DO LOOP indices directly as they refer to the banded **A** matrix. Both approaches are used in the computer programs.]

Solution with Multiple Right Sides

Often, we need to solve $\mathbf{Ax} = \mathbf{b}$ with the same $\mathbf{A}$, but with different $\mathbf{b}$s. This happens in the finite element method when we wish to analyze the same structure for different loading conditions. In this situation, it is computationally more economical to separate the calculations associated with $\mathbf{A}$ from those associated with $\mathbf{b}$. The reason for this is that the number of operations in reduction of an $(n \times n)$ matrix $\mathbf{A}$ to its triangular form is proportional to n^3, while the number of operations for reduction of $\mathbf{b}$ and back-substitution is proportional only to n^2. For large n, this difference is significant.

The previous algorithm for a symmetric banded matrix is modified accordingly as follows:

Algorithm 3: Symmetric Banded, Multiple Right Sides

Forward elimination for A

$$
\begin{array}{l}
\text{DO} \quad k = 1, n - 1 \\
\text{nbk} = \min(n - k + 1, \text{nbw}) \\
\quad \text{DO} \quad i = k + 1, \text{nbk} + k - 1 \\
\quad i1 = i - k + 1 \\
\quad c = a_{k,i1}/a_{k,1} \\
\quad\quad \text{DO} \quad j = i, \text{nbk} + k - 1 \\
\quad\quad j1 = j - i + 1 \\
\quad\quad j2 = j - k + 1 \\
\quad\quad a_{i,j1} = a_{i,j1} - c a_{k,j2}
\end{array}
$$

Forward elimination of each b

$$
\begin{array}{l}
\text{DO} \quad k = 1, n - 1 \\
\text{nbk} = \min(n - k + 1, \text{nbw}) \\
\quad \text{DO} \quad i = k + 1, \text{nbk} + k - 1 \\
\quad i1 = i - k + 1 \\
\quad c = a_{k,i1}/a_{k,1} \\
\quad b_i = b_i - c b_k
\end{array}
$$

Back-substitution This algorithm is the same as in Algorithm 2.

Gaussian Elimination with Column Reduction

A careful observation of the Gaussian elimination process shows us a way to reduce the coefficients column after column. This process leads to the simplest procedure for

skyline solution, which we present later. We consider the column-reduction procedure for symmetric matrices. Let the coefficients in the upper triangular matrix and the vector **b** be stored.

We can understand the motivation behind the column approach by referring back to Eq. 2.41, which is

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\
 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & \cdots & a_{2n}^{(1)} \\
 & & a_{33}^{(2)} & a_{34}^{(2)} & \cdots & a_{3n}^{(2)} \\
 & & & a_{44}^{(3)} & \cdots & a_{4n}^{(3)} \\
 & & & \vdots & & \vdots \\
 & & & & & a_{nn}^{(n-1)}
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{Bmatrix}
=
\begin{Bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ b_4^{(3)} \\ \vdots \\ b_n^{(n-1)} \end{Bmatrix}
$$

Let us focus our attention on, say, column 3 of the reduced matrix. The first element in this column is unmodified, the second element is modified once, and the third element is modified twice. Further, from Eq. 2.33, and using the fact that $a_{ij} = a_{ji}$ since **A** is assumed to be symmetric, we have

$$
\begin{aligned}
a_{23}^{(1)} &= a_{23} - \frac{a_{12}}{a_{11}} a_{13} \\
a_{33}^{(1)} &= a_{33} - \frac{a_{13}}{a_{11}} a_{13} \\
a_{33}^{(2)} &= a_{33}^{(1)} - \frac{a_{23}^{(1)}}{a_{22}^{(1)}} a_{23}^{(1)}
\end{aligned}
\tag{2.42}
$$

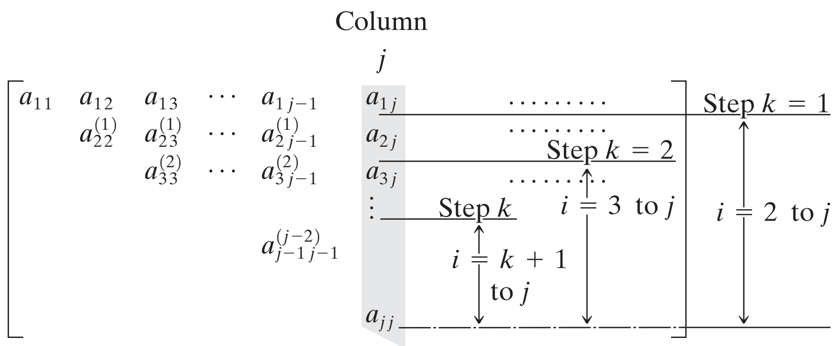
From these equations, we make the critical observation that *the reduction of column 3 can be done using only the elements in columns 1 and 2 and the already reduced elements in column 3.* This idea whereby column 3 is obtained using only elements in previous columns that have already been reduced is shown schematically:

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
 & a_{22}^{(1)} & a_{23} \\
 & & a_{33}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
 & a_{22}^{(1)} & a_{23}^{(1)} \\
 & & a_{33}^{(1)}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
 & a_{22}^{(1)} & a_{23}^{(1)} \\
 & & a_{33}^{(2)}
\end{bmatrix}
\tag{2.43}
$$

The reduction of other columns is similarly completed. For example, the reduction of column 4 can be done in three steps as shown schematically in

$$
\begin{Bmatrix} a_{14} \\ a_{24} \\ a_{34} \\ a_{44} \end{Bmatrix}
\rightarrow
\begin{Bmatrix} a_{14} \\ a_{24}^{(1)} \\ a_{34}^{(1)} \\ a_{44}^{(1)} \end{Bmatrix}
\rightarrow
\begin{Bmatrix} a_{14} \\ a_{24}^{(1)} \\ a_{34}^{(2)} \\ a_{44}^{(2)} \end{Bmatrix}
\rightarrow
\begin{Bmatrix} a_{14} \\ a_{24}^{(1)} \\ a_{34}^{(2)} \\ a_{44}^{(3)} \end{Bmatrix}
\tag{2.44}
$$

We now discuss the reduction of column j, $2 \leq j \leq n$, assuming that columns to the left of j have been fully reduced. The coefficients can be represented in the following form:



(2.45)

The reduction of column j requires only elements from columns to the left of j and appropriately reduced elements from column j. We note that for column j, the number of steps needed is $j - 1$. Also, since a_{11} is not reduced, we need to reduce columns 2 to n only. The logic is now given as

DO $j = 2$ to n
DO $k = 1$ to $j - 1$
DO $i = k + 1$ to j

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ki}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} \tag{2.46}$$

Interestingly, the reduction of the right side **b** can be considered as the reduction of one more column. Thus, we have

DO $k = 1$ to $n - 1$
DO $i = k + 1$ to n

$$b_i^{(k)} = b_i^{(k-1)} - \frac{a_{ki}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)} \tag{2.47}$$

From Eq. 2.46, we observe that if there are a set of zeroes at the top of a column, the operations need to be carried out only on the elements ranging from the first nonzero element to the diagonal. This leads naturally to the skyline solution.

Skyline Solution

If there are zeroes at the top of a column, only the elements starting from the first nonzero value need be stored. The line separating the top zeroes from the first nonzero element is called the *skyline*. Consider the example

$$
\begin{array}{rc}
\overrightarrow{\text{Column height}} & \begin{array}{cccccccc} 1 & 2 & 2 & 4 & 4 & 4 & 3 & 5 \end{array} \\
 & \begin{bmatrix}
a_{11} & a_{12} & 0 & a_{14} & 0 & 0 & 0 & 0 \\
 & a_{22} & a_{23} & a_{24} & a_{25} & 0 & 0 & 0 \\
 & & a_{33} & a_{34} & 0 & a_{36} & 0 & 0 \\
 & & & a_{44} & 0 & a_{46} & 0 & a_{48} \\
 & & & & a_{55} & a_{56} & a_{57} & 0 \\
 & & & & & a_{66} & a_{67} & a_{68} \\
 & & & & & & a_{77} & a_{78} \\
 & & & & & & & a_{88}
\end{bmatrix} \text{Skyline}
\end{array}
\tag{2.48}
$$

For efficiency, only the active columns need be stored. These can be stored in a column vector **A** and a diagonal pointer vector **ID** as

$$
\mathbf{A} = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{22} \\ a_{23} \\ a_{33} \\ a_{14} \\ a_{24} \\ a_{34} \\ a_{44} \\ \vdots \\ a_{88} \end{bmatrix}
\begin{array}{l} \text{Diagonal pointer (ID)} \\ \leftarrow 1 \\ \\ \leftarrow 3 \\ \\ \leftarrow 5 \\ \\ \\ \\ \leftarrow 9 \\ \\ \leftarrow 25 \end{array}
\qquad
\mathbf{ID} = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 9 \\ 13 \\ 17 \\ 20 \\ 25 \end{bmatrix}
\tag{2.49}
$$

The height of column I is given by $\text{ID}(I) - \text{ID}(I - 1)$. The right side, **b**, is stored in a separate column. The column reduction scheme of the Gauss elimination method can be applied for solution of a set of equations. A skyline solver program is given.

Frontal Solution

Frontal method is a variation of the Gaussian elimination method that uses the structure of the finite element problem. The elimination process is handled by writing the eliminated equation to the computer hard disk, thus reducing the need for a large amount of memory. Large finite element problems can thus be solved using small computers. The frontal method is presented in Chapter 9 and implemented for the hexahedral element.

2.3 CONJUGATE GRADIENT METHOD FOR EQUATION SOLVING

The conjugate gradient method is an iterative method for the solution of equations. This method is becoming increasingly popular and is implemented in several computer codes.

We present here the Fletcher–Reeves version of the algorithm for symmetric matrices. Consider the solution of the set of equations

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ is a symmetric positive definite $(n \times n)$ matrix and $\mathbf{b}$ and $\mathbf{x}$ are $(n \times 1)$. The conjugate gradient method uses the following steps for symmetric $\mathbf{A}$.

Conjugate Gradient Algorithm

Start at point $\mathbf{x}_0$:

$$\begin{aligned}
\mathbf{g}_0 &= \mathbf{Ax}_0 - \mathbf{b}, \quad \mathbf{d}_0 = -\mathbf{g}_0 \\
\boldsymbol{\alpha}_k &= \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k}{\mathbf{d}_k^{\mathrm{T}}\mathbf{A}\mathbf{d}_k} \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{d}_k \\
\mathbf{g}_{k+1} &= \mathbf{g}_k + \alpha_k \mathbf{A}\mathbf{d}_k \\
\boldsymbol{\beta}_k &= \frac{\mathbf{g}_{k+1}^T\mathbf{g}_{k+1}}{\mathbf{g}_k^T\mathbf{g}_k} \\
\mathbf{d}_{k+1} &= -\mathbf{g}_{k+1} + \boldsymbol{\beta}_k \mathbf{d}_k
\end{aligned} \tag{2.50}$$

Here $k = 0,1,2,\ldots$. The iterations are continued until $\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k$ reaches a small value. This method is robust and converges in n iterations. This procedure is implemented in the program CGSOLVE, which is included in the downloadable files. This procedure is adaptable to parallel processing in finite element applications and can be accelerated by using preconditioning strategies. The program input and output are as given below.

Input Data/Output

INPUT FOR GAUSS, CGSOLVE

```
EQUATION SOLVING USING GAUSS ELIMINATION
  Number of Equations
  8
  Matrix A() in Ax = B
  6   0   1   2   0   0   2   1
  0   5   1   1   0   0   3   0
  1   1   6   1   2   0   1   2
  2   1   1   7   1   2   1   1
  0   0   2   1   6   0   2   1
  0   0   0   2   0   4   1   0
  2   3   1   1   2   1   5   1
  1   0   2   1   1   0   1   3
  Right hand side B() in Ax = B
  1   1   1   1   1   1   1   1
```

OUTPUT FROM GAUSS

```
Results from Program Gauss
EQUATION SOLVING USING GAUSS ELIMINATION
Solution
1   0.392552899
2   0.639736191
3  -0.1430338
4  -0.217230008
5   0.380186865
6   0.511816433
7  -0.612805716
8   0.447787854
```

PROBLEMS

2.1. Given that

$$\mathbf{A} = \begin{bmatrix} 8 & -2 & 0 \\ -2 & 4 & -3 \\ 0 & -3 & 3 \end{bmatrix} \quad \mathbf{d} = \begin{Bmatrix} 2 \\ -1 \\ 3 \end{Bmatrix}$$

determine the following:

(a) $\mathbf{I} - \mathbf{d}\mathbf{d}^{\mathrm{T}}$

(b) det $\mathbf{A}$

(c) The eigenvalues and eigenvectors of $\mathbf{A}$. Is $\mathbf{A}$ positive definite?

(d) The solution to $\mathbf{Ax} = \mathbf{d}$ using Algorithms 1 and 2, by hand calculation.

2.2. Given that

$$\mathbf{N} = [\xi, 1 - \xi^2]$$

find

(a) $\displaystyle\int_{-1}^{1} \mathbf{N}\, d\xi$

(b) $\displaystyle\int_{-1}^{1} \mathbf{N}^{\mathrm{T}}\mathbf{N}\, d\xi$

2.3. Express $3x_1 - 8x_2 - 5x_1^2 + 5x_1x_2$ in the matrix form $\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Qx} + \mathbf{c}^{\mathrm{T}}\mathbf{x}$.

2.4. Implement Algorithm 3 in BASIC or MATLAB. Hence, solve $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A}$ as in Problem 2.1, and each of the following $\mathbf{b}$s:

$$\mathbf{b} = [5, -10, 3]^{\mathrm{T}}$$
$$\mathbf{b} = [2.2, -1, 3]^{\mathrm{T}}$$

2.5. Using the cofactor approach, determine the inverse of the matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 1 & 2 \end{bmatrix}$$

2.6. Given that the area of a triangle with corners at $(x_1, y_1), (x_2, y_2)$, and (x_3, y_3) can be written in the form

$$\text{Area} = \frac{1}{2}\det\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}$$

determine the area of the triangle with corners at $(2, 2)$, $(7, 8)$, and $(11, 12)$.

2.7. For the triangle shown in Fig. P2.7, the interior point P at $(2, 2)$ divides it into three areas, A_1, A_2, and A_3, as shown. Determine A_1/A, A_2/A, and A_3/A.

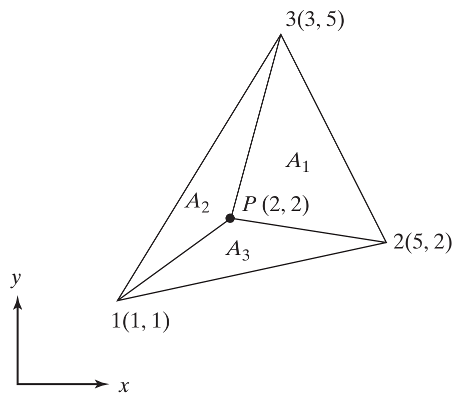


FIGURE P2.7

2.8. A symmetric matrix $[\mathbf{A}]_{n\times n}$ has a bandwidth nbw and is stored in the matrix $[\mathbf{B}]_{n\times \text{nbw}}$.

(a) Find the location in **B** that corresponds to $A_{11,14}$.

(b) Find the location in **A** that corresponds to $B_{6,1}$.

2.9. For a symmetric (10×10) matrix with all nonzero elements, determine the number of locations needed for banded and skyline storage methods.

2.10. Perform the Cholesky decomposition of the positive definite matrix

$$\begin{bmatrix} 4 & 2 & -6 \\ 2 & 10 & 9 \\ -6 & 9 & 26 \end{bmatrix}$$

2.11. A square matrix **A** can be decomposed into $\mathbf{A} = \mathbf{LU}$, where **L** is lower triangular and **U** is upper triangular. Let

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 8 \\ 3 & 8 & 14 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$
$$= \mathbf{LU}$$

Determine **L** and **U**.

2.12. Determine the area of the quadrilateral $ABCD$ with corners A, B, C, D, respectively, at (1,1), (7,2), (6,6), (3,7) by decomposing it into two triangles and using the ideas from Problem 2.6.

2.13. **T** is called an orthogonal matrix if $\mathbf{T}^{-1} = \mathbf{T}^{\mathrm{T}}$. Show that the transformation matrix $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ is orthogonal.

2.14. For the matrix given below, compute (a) all the minors, (b) all the cofactors, (c) the adjoint matrix, (d) the determinant, and (e) the inverse.

$$\begin{bmatrix} 4 & 3 & 1 \\ 2 & 3 & 2 \\ 3 & 4 & 5 \end{bmatrix}$$

Program Listings

```
<< MAIN PROGRAM >>
'*********    PROGRAM GAUSS   ***********
'*      GAUSS ELIMINATION METHOD       *
'*          GENERAL MATRIX             *
'* T.R.Chandrupatla and A.D.Belegundu *
'**************************************
DefInt I-N
DefDbl A-H, O-Z
Dim N
Dim ND, NL, NCH, NPR, NMPC, NBW
Dim A(), B()
Private Sub CommandButton1_Click()
     Call InputData
     Call GaussRow
     Call Output
End Sub
```

```
<< READ DATA FROM SHEET1 (from file in C, FORTRAN, MATLAB, QB, VB) >>
Private Sub InputData()
    N = Worksheets(1).Cells(3, 1)
    ReDim A(N, N), B(N)
      LI = 4
    '----- Read A() -----
    For I = 1 To N
        LI = LI + 1
        For J = 1 To N
        A(I, J) = Worksheets(1).Cells(LI, J)
        Next
    Next
     LI = LI + 2
    '----- Read B() -----
     For J = 1 To N
        B(J) = Worksheets(1).Cells(LI, J)
     Next
End Sub
```

```
<< GAUSS ELIMINATION ROUTINE >>
Private Sub GaussRow()
     '----- Forward Elimination -----
     For K = 1 To N - 1
        For I = K + 1 To N
           C = A(I, K) / A(K, K)
           For J = K + 1 To N
              A(I, J) = A(I, J) - C * A(K, J)
           Next J
           B(I) = B(I) - C * B(K)
        Next I
     Next K
     '----- Back-substitution -----
     B(N) = B(N) / A(N, N)
     For II = 1 To N - 1
        I = N - II
        C = 1 / A(I, I): B(I) = C * B(I)
        For K = I + 1 To N
           B(I) = B(I) - C * A(I, K) * B(K)
        Next K
     Next II
End Sub
```

```
<< OUTPUT TO SHEET2 (to file in C, FORTRAN, MATLAB, QB, VB) >>
Private Sub Output()
   ' Now, writing out the results in a different worksheet
    Worksheets(2).Cells.ClearContents
    Worksheets(2).Cells(1, 1) = "Results from Program Gauss"
    Worksheets(2).Cells(1, 1).Font.Bold = True
    Worksheets(2).Cells(2, 1) = Worksheets(1).Cells(1, 1)
    Worksheets(2).Cells(2, 1).Font.Bold = True
    Worksheets(2).Cells(3, 1) = "Solution"
    Worksheets(2).Cells(3, 1).Font.Bold = True
    LI = 3
    For I = 1 To N
        LI = LI + 1
        Worksheets(2).Cells(LI, 1) = I
        Worksheets(2).Cells(LI, 2) = B(I)
    Next
End Sub
```

CHAPTER 3

One-Dimensional Problems

3.1 INTRODUCTION

The total potential energy and the stress–strain and strain–displacement relationships are now used in developing the finite element method for a one-dimensional problem. The basic procedure is the same for two- and three-dimensional problems discussed later in the book. For the one-dimensional problem, the stress, strain, displacement, and loading depend only on the variable x. That is, the vectors $\mathbf{u}$, $\boldsymbol{\sigma}$, $\boldsymbol{\epsilon}$, $\mathbf{T}$, and $\mathbf{f}$ in Chapter 1 now reduce to

$$\begin{gathered}\mathbf{u} = u(x) \quad \boldsymbol{\sigma} = \sigma(x) \quad \boldsymbol{\epsilon} = u(x) \\ \mathbf{T} = T(x) \quad \mathbf{f} = f(x)\end{gathered} \tag{3.1}$$

Furthermore, the stress–strain and strain–displacement relations are

$$\sigma = E\epsilon \quad \epsilon = \frac{du}{dx} \tag{3.2}$$

For one-dimensional problems, the differential volume dV can be written as

$$dV = Adx \tag{3.3}$$

The loading consists of three types: the **body force** f, the **traction force** T, and the **point load** P_i. These forces are shown acting on a body in Fig. 3.1. A body force is a distributed force acting on every elemental volume of the body and has the units of force per unit volume. The self-weight due to gravity is an example of a body force. A traction force is a distributed load acting on the surface of the body. In Chapter 1, the traction force is defined as force per unit area. For the one-dimensional problem considered here, however, the traction force is defined as force per unit length. This is done by taking the traction force to be the product of the force per unit area and the perimeter of the cross section. Frictional resistance, viscous drag, and surface shear are examples of traction forces in one-dimensional problems. Finally, P_i is a force acting at a point i and u_i is the x displacement at that point.

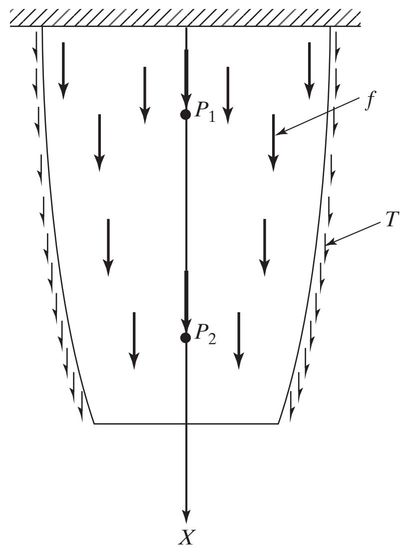


FIGURE 3.1 One-dimensional bar loaded by traction, body, and point loads.

The finite element modeling of a one-dimensional body is considered in Section 3.2. The basic idea is to discretize the region and express the displacement field in terms of values at discrete points. Linear elements are introduced first. Stiffness and load concepts are developed using potential energy and Galerkin approaches. Boundary conditions are then considered. Temperature effects and quadratic elements are discussed later in this chapter.

3.2 FINITE ELEMENT MODELING

The steps of element division and node numbering are discussed here.

Element Division

Consider the bar shown in Fig. 3.1. The first step is to **model** the bar as a *stepped shaft*, consisting of a discrete number of elements, each having a uniform cross section. Specifically, let us model the bar using four finite elements. A simple scheme for doing this is to divide the bar into four regions, as shown in Fig. 3.2a . The average cross-sectional area within each region is evaluated and then used to define an element with uniform cross section. The resulting four-element, five-node finite element model is shown in Fig. 3.2b. In the finite element model, every element connects to two nodes. In Fig. 3.2b, the element numbers are circled to distinguish them from node numbers. In addition to the cross section, traction and body forces are also (normally) treated as constant within each element. However, cross-sectional area, traction, and body forces can differ in magnitude from element to element. Better approximations are obtained by

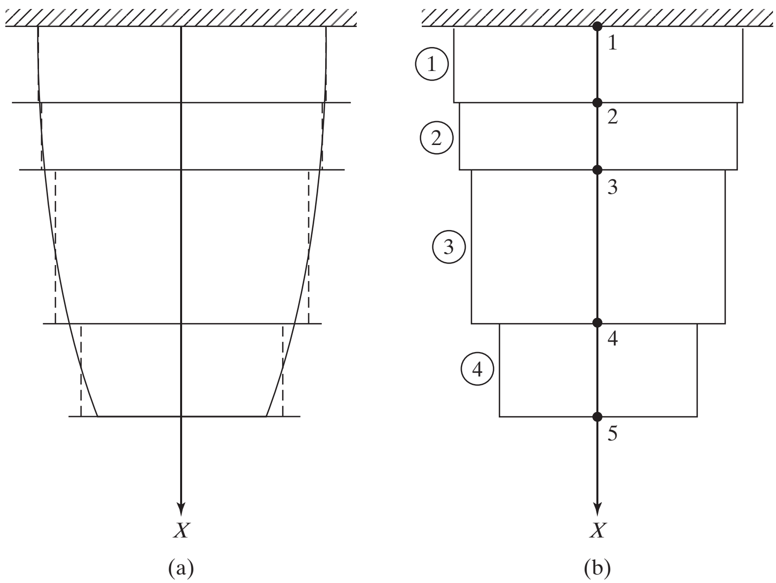


FIGURE 3.2 Finite element modeling of a bar.

increasing the number of elements. It is convenient to define a node at each location where a point load is applied.

Numbering Scheme

We have shown how a rather complicated looking bar has been modeled using a discrete number of elements, each element having a simple geometry. The similarity of the various elements is one reason why the finite element method is easily amenable to computer implementation. For easy implementation, an orderly numbering scheme for the model has to be adopted.

In a one-dimensional problem, every node is permitted to displace only in the $\pm x$ direction. Thus, each node has only one *degree of freedom (dof)*. The five-node finite element model in Fig. 3.2b has five dof. The displacements along each dof are denoted by $Q_1, Q_2, \ldots, Q_5$. In fact, the column vector $\mathbf{Q} = [Q_1, Q_2, \ldots, Q_5]^T$ is called the *global displacement vector*. The *global load vector* is denoted by $\mathbf{F} = [F_1, F_2, \ldots, F_5]^T$. The vectors $\mathbf{Q}$ and $\mathbf{F}$ are shown in Fig. 3.3 . The sign convention used is that a displacement or load has a positive value if acting along the $+x$ direction. At this stage, conditions at the boundary are not imposed. For example, node 1 in Fig. 3.3 is fixed, which implies $Q_1 = 0$. These conditions are discussed later.

Each element has two nodes; therefore, the **element connectivity** information can be conveniently represented as shown in Fig. 3.4. Further, the element connectivity table is also given. In the connectivity table, headings 1 and 2 refer to *local node numbers* of an element, and the corresponding node numbers on the body are called *global numbers*. Connectivity thus establishes the local–global correspondence. In this simple example, the connectivity can be easily generated since local node 1 is the same as the element

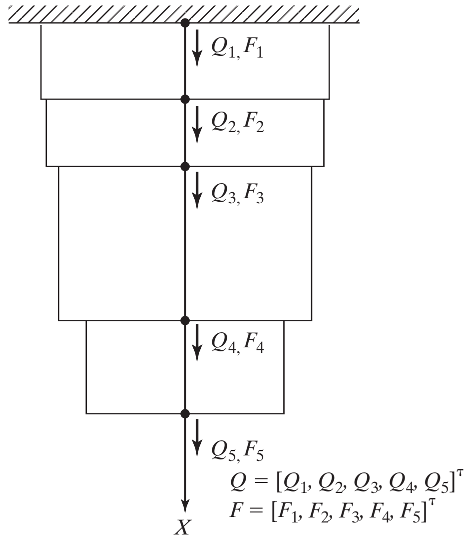


FIGURE 3.3 **Q** and **F** vectors.

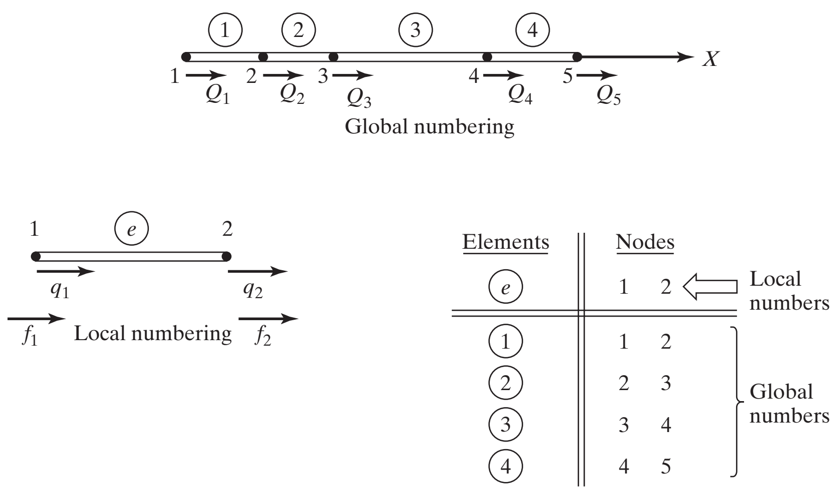


FIGURE 3.4 Element connectivity.

number e, and local node 2 is $e + 1$. Other ways of numbering nodes or more complex geometries suggest the need for a connectivity table. The connectivity is introduced in the program using the array NOC.

The concepts of dof, nodal displacements, nodal loads, and element connectivity are central to the finite element method and should be clearly understood.

3.3 SHAPE FUNCTIONS AND LOCAL COORDINATES

Relationship between the finite element approach in this chapter and the Rayleigh–Ritz method in Chapter 1 can be clearly understood by expressing the displacement using piecewise basis functions. We define piecewise linear basis functions $\Phi_j(\mathbf{x})$, $j = 1$ to 5 as shown in Fig. 3.5. The functions are defined by

$$\Phi_j(\mathbf{x}) = \begin{bmatrix} 1 & \text{at node } j \\ 0 & \text{if node } k \neq j \end{bmatrix} \tag{3.4}$$

Using these functions, we may write

$$u = \sum_{j=1}^{5} Q_j(\mathbf{x})\Phi_j(\mathbf{x}) \tag{3.5}$$

These basis functions may be referred to as *global shape functions*. These functions represent a constant function or a linear function exactly. They also satisfy the relation

$$\sum_j \Phi_j(\mathbf{x}) = 1 \tag{3.6}$$

This is called the *partition of unity* property. Generalized functions of this type are used in mesh free methods referred to as the generalized finite element method (GFEM).

Since the integration of the potential energy is the summation of integration over each element, it is more convenient to look at these basis functions from element point of view. For example in element 2, we have the right portion of Φ_2, and left portion of Φ_3. In a general element e if we call the left node as 1 and the right node as 2, these parts of the basis functions may be viewed as *shape functions* N_1 and N_2. Also if we define the shape functions on an interval (–1, 1) and define a mapping for an element, these functions will be invariant.

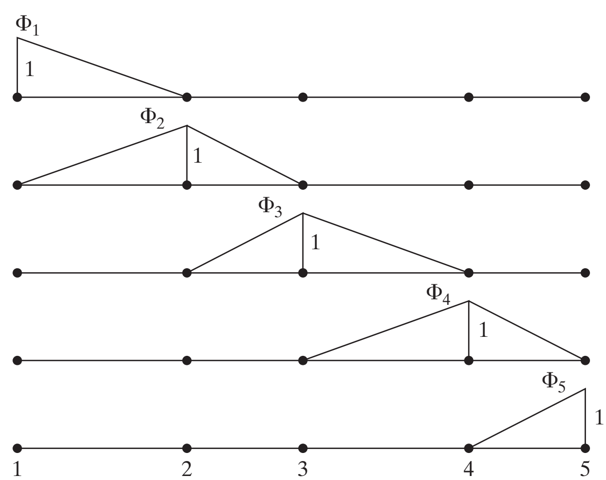


FIGURE 3.5 Global shape functions.

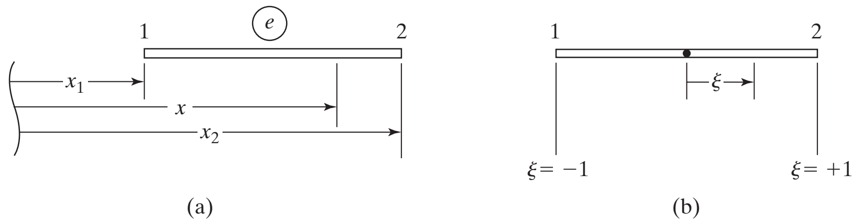


FIGURE 3.6 Typical element in x- and ξ-coordinates.

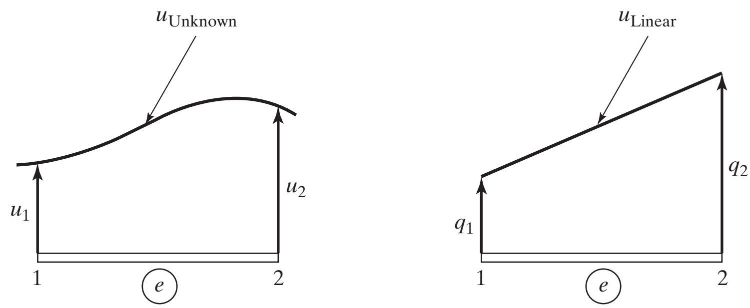


FIGURE 3.7 Linear interpolation of the displacement field within an element.

Consider a typical finite element e in Fig. 3.6a. In the local number scheme, the first node will be numbered 1 and the second node 2. The notation $x_1 = x$-coordinate of node 1, $x_2 = x$-coordinate of node 2 is used. We define a **natural** or **intrinsic** coordinate system, denoted by ξ, as

$$\xi = \frac{2}{x_2 - x_1}(x - x_1) - 1 \tag{3.7}$$

From Fig. 3.6b, we see that $\xi = -1$ at node 1 and $\xi = 1$ at node 2. The length of an element is covered when ξ changes from -1 to 1. We use this system of coordinates in defining shape functions, which are used in interpolating the displacement field.

Now *the unknown displacement field within an element will be interpolated by a linear distribution* (Fig. 3.7). This approximation becomes increasingly accurate as more elements are considered in the model. To implement this linear interpolation, linear **shape functions** will be introduced as

$$N_1(\xi) = \frac{1 - \xi}{2} \tag{3.8}$$

$$N_2(\xi) = \frac{1 + \xi}{2} \tag{3.9}$$

The shape functions N_1 and N_2 are shown in Fig. 3.8a and b, respectively. The graph of the shape function N_1 in Fig. 3.8a is obtained from Eq. 3.8 by noting that $N_1 = 1$ at $\xi = -1$, $N_1 = 0$ at $\xi = 1$, and N_1 is a straight line between the two points. Similarly, the

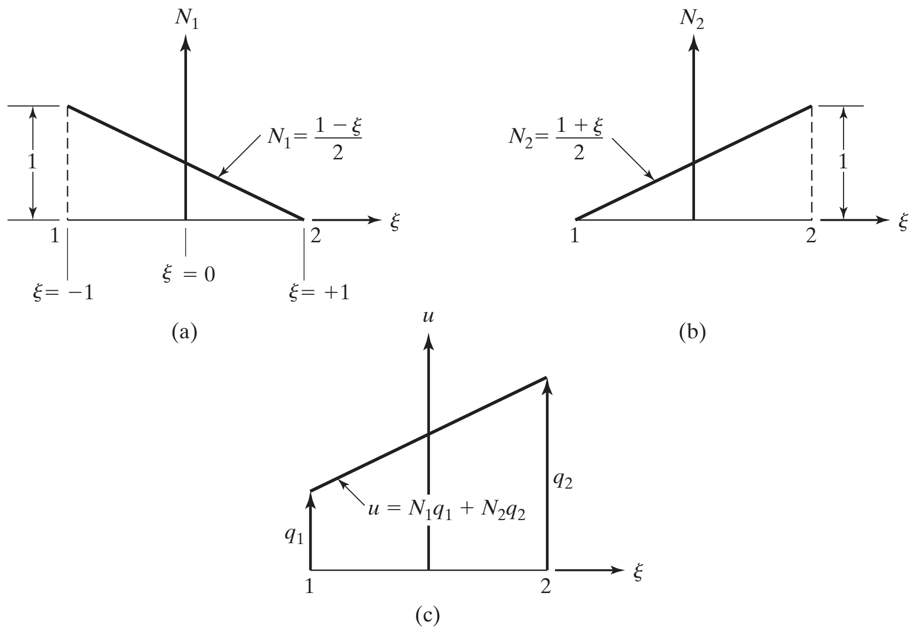


FIGURE 3.8 (a) Shape function N_1, (b) shape function N_2, and (c) linear interpolation using N_1 and N_2.

graph of N_2 in Fig. 3.8b is obtained from Eq. 3.9. Once the shape functions are defined, the linear displacement field within the element can be written in terms of the nodal displacements q_1 and q_2 as

$$u = N_1 q_1 + N_2 q_2 \tag{3.10a}$$

or, in matrix notation, as

$$u = \mathbf{N}\mathbf{q} \tag{3.10b}$$

where

$$\mathbf{N} = [N_1, N_2] \quad \text{and} \quad \mathbf{q} = [q_1, q_2]^{\mathrm{T}} \tag{3.11}$$

In these equations, $\mathbf{q}$ is referred to as the *element displacement vector*. It is readily verified from Eq. 3.10a that $u = q_1$ at node 1, $u = q_2$ at node 2, and that u varies linearly (Fig. 3.8c).

It may be noted that the transformation from x to ξ in Eq. 3.4 can be written in terms of N_1 and N_2 as

$$x = N_1 x_1 + N_2 x_2 \tag{3.12}$$

Comparing Eqs. 3.10a and 3.12, we see that both the displacement u and the coordinate x are interpolated within the element using the *same* shape functions N_1 and N_2. This is referred to as the *isoparametric* formulation in the literature.

Though linear shape functions have been used previously, other choices are possible. Quadratic shape functions are discussed in Section 3.9. In general, shape functions need to satisfy the following:

1. First derivatives must be finite within an element.
2. Displacements must be continuous across the element boundary.

Rigid body motion should not introduce any stresses in the element. Convergence aspects for two-dimensional problems are discussed in Chapter 8.

Example 3.1

Referring to Fig. E3.1, do the following:

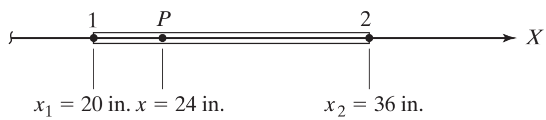


FIGURE E3.1

(a) Evaluate ξ, N_1, and N_2 at point P.
(b) If $q_1 = 0.003$ in. and $q_2 = -0.005$ in., determine the value of the displacement q at point P.

Solution

(a) Using Eq. 3.7, the ξ-coordinate of point P is given by

$$\xi_p = \frac{2}{16}(24 - 20) - 1$$
$$= -0.5$$

Now Eqs. 3.8 and 3.9 yield

$$N_1 = 0.75 \quad \text{and} \quad N_2 = 0.25$$

(b) Using Eq. 3.10a, we get

$$u_p = 0.75(0.003) + 0.25(-0.005)$$
$$= 0.001 \text{ in.}$$

■

The strain–displacement relation in Eq. 3.2 is

$$\epsilon = \frac{du}{dx}$$

Upon using the chain rule of differentiation, we obtain

$$\epsilon = \frac{du}{d\xi}\frac{d\xi}{dx} \tag{3.13}$$

From the relation between x and ξ in Eq. 3.7, we have

$$\frac{d\xi}{dx} = \frac{2}{x_2 - x_1} \tag{3.14}$$

Also, since

$$u = N_1 q_1 + N_2 q_2 = \frac{1 - \xi}{2} q_1 + \frac{1 + \xi}{2} q_2$$

we have

$$\frac{du}{d\xi} = \frac{-q_1 + q_2}{2} \tag{3.15}$$

Thus, Eq. 3.13 yields

$$\epsilon = \frac{1}{x_2 - x_1}(-q_1 + q_2) \tag{3.16}$$

The Eq. 3.16 can be written as

$$\epsilon = \mathbf{Bq} \tag{3.17}$$

where the (1×2) matrix $\mathbf{B}$, called the *element strain–displacement matrix*, is given by

$$\mathbf{B} = \frac{1}{x_2 - x_1}[-1 \quad 1] \tag{3.18}$$

Note: Use of linear shape functions results in a constant $\mathbf{B}$ matrix and, hence, in a constant strain within the element. The stress, from Hooke's law, is

$$\sigma = E\mathbf{Bq} \tag{3.19}$$

The stress given by this equation is also constant within the element. For interpolation purposes, however, the stress obtained from Eq. 3.16 can be considered to be the value at the centroid of the element.

The expressions $u = \mathbf{Nq}$, $\epsilon = \mathbf{Bq}$, and $\sigma = E\mathbf{Bq}$ relate the displacement, strain and stress, respectively, in terms of nodal values. These expressions will now be substituted into the potential-energy expression for the bar to obtain the element stiffness and load matrices.

3.4 THE POTENTIAL-ENERGY APPROACH

The general expression for the potential energy given in Chapter 1 is

$$\Pi = \frac{1}{2}\int_L \sigma^{\mathrm{T}}\epsilon A\,dx - \int_L u^{\mathrm{T}} f A\,dx - \int_L u^{\mathrm{T}} T\,dx - \sum_i u_i P_i \tag{3.20}$$

The quantities σ, ϵ, u, f, and T in Eq. 3.20 are discussed at the beginning of this chapter. In the last term, P_i represents a force acting at point i, and u_i is the x displacement at that point. The summation on i gives the potential energy due to all point loads.

Since the continuum has been discretized into finite elements, the expression for Π becomes

$$\Pi = \sum_e \frac{1}{2}\int_e \sigma^{\mathrm{T}}\epsilon A\,dx - \sum_e \int_e u^{\mathrm{T}} f A\,dx - \sum_e \int_e u^{\mathrm{T}} T\,dx - \sum_i Q_i P_i \qquad (3.21a)$$

The last term in Eq. 3.21a assumes that point loads P_i are applied at the nodes. This assumption makes the present derivation simpler with respect to notation and is also a common modeling practice. Equation 3.21a can be written as

$$\Pi = \sum_e U_e - \sum_e \int_e u^{\mathrm{T}} f A\,dx - \sum_e \int_e u^{\mathrm{T}} T\,dx - \sum_i Q_i P_i \qquad (3.21b)$$

where

$$U_e = \frac{1}{2}\int_e \sigma^{\mathrm{T}}\epsilon A\,dx \qquad (3.22)$$

is the element strain energy.

Element Stiffness Matrix

Consider the strain energy term U_e in Eq. 3.22.
Substituting for $\sigma = E\mathbf{B}\mathbf{q}$ and $\epsilon = \mathbf{B}\mathbf{q}$ into Eq. 3.22 yields

$$U_e = \frac{1}{2}\int_e \mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}E\mathbf{B}\mathbf{q}A\,dx \qquad (3.23a)$$

or

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\int_e [\mathbf{B}^{\mathrm{T}}E\mathbf{B}A\,dx]\mathbf{q} \qquad (3.23b)$$

In the finite element model (Section 3.2), the cross-sectional area of element e, denoted by A_e, is constant. Also, $\mathbf{B}$ is a constant matrix. Further, the transformation from x to ξ in Eq. 3.7 yields

$$dx = \frac{x_2 - x_1}{2}d\xi \qquad (3.24a)$$

or

$$dx = \frac{\ell_e}{2}d\xi \qquad (3.24b)$$

where $-1 \le \xi \le 1$, and $\ell_e = |x_2 - x_1|$ is the length of the element.

The element strain energy U_e is now written as

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\left[A_e\frac{\ell_e}{2}E_e\mathbf{B}^{\mathrm{T}}\mathbf{B}\int_{-1}^{1} d\xi\right]\mathbf{q} \qquad (3.25)$$

where E_e is Young's modulus of element e. Noting that $\int_{-1}^{1} d\xi = 2$ and substituting for $\mathbf{B}$ from Eq. 3.18, we get

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}} A_e \ell_e E_e \frac{1}{\ell_e^2}\begin{Bmatrix} -1 \\ 1 \end{Bmatrix}[-1 \quad 1]\mathbf{q} \tag{3.26}$$

which results in

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}} \frac{A_e E_e}{\ell_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\mathbf{q} \tag{3.27}$$

This equation is of the form

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} \tag{3.28}$$

where the **element stiffness matrix $\mathbf{k}^e$** is given by

$$\mathbf{k}^e = \frac{E_e A_e}{\ell_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.29}$$

We note here the similarity of the strain energy expression in Eq. 3.29 with the strain energy in a simple spring, which is given as $U = \frac{1}{2}kQ^2$. Also, observe that $\mathbf{k}^e$ is linearly proportional to the product $A_e E_e$ and inversely proportional to the length ℓ_e.

Element Stiffness – Direct Approach

Direct approach uses stress–strain relations and strain–displacement relations developed in a first course on solid mechanics. We show this approach here, however the main steps developed using shape functions provide the background needed for general element development.

For the element shown in Fig. 3.4, we have

$$\epsilon = \frac{q_2 - q_1}{\ell}$$

$$\sigma = E\epsilon$$

$$f_1 = -\sigma A = \frac{EA}{\ell}(q_1 - q_2)$$

$$f_2 = \sigma A = \frac{EA}{\ell}(-q_1 + q_2)$$

Putting in the matrix form, we have

$$\frac{EA}{\ell}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix}$$

This is $\mathbf{k}^e\mathbf{q} = \mathbf{f}$, where $\mathbf{k}^e$ is the element stiffness matrix given in Eq. 3.29.

Force Terms

The element body force term $\int_e u^{\mathrm{T}} fA\,dx$ appearing in the total potential energy is considered first. Substituting $u = N_1 q_1 + N_2 q_2$, we have

$$\int_e u^{\mathrm{T}} fA\,dx = A_e f \int_e (N_1 q_1 + N_2 q_2)\,dx \tag{3.30}$$

Recall that the body force f has units of force per unit volume. In Eq. 3.30, A_e and f are constant within the element and were consequently brought outside the integral. This equation can be written as

$$\int_e u^{\mathrm{T}} fA\,dx = \mathbf{q}^{\mathrm{T}} \begin{Bmatrix} A_e f \int_e N_1\,dx \\ A_e f \int_e N_2\,dx \end{Bmatrix} \tag{3.31}$$

The integrals of the shape functions described earlier can be readily evaluated by making the substitution $dx = (\ell_2/2)d\xi$. Thus,

$$\begin{aligned} \int_e N_1\,dx &= \frac{\ell_e}{2} \int_{-1}^{1} \frac{1-\xi}{2}\,d\xi = \frac{\ell_e}{2} \\ \int_e N_2\,dx &= \frac{\ell_e}{2} \int_{-1}^{1} \frac{1+\xi}{2}\,d\xi = \frac{\ell_e}{2} \end{aligned} \tag{3.32}$$

Alternatively, $\int_e N_1\,dx$ is simply the area under the N_1 curve as shown in Fig. 3.9, which equals $\frac{1}{2} \cdot \ell_e \cdot 1 = \ell_e/2$. Similarly, $\int N_2\,dx = \frac{1}{2} \cdot \ell_e \cdot 1 = \ell_e/2$. The body force term in Eq. 3.28 reduces to

$$\int_e u^{\mathrm{T}} fA\,dx = \mathbf{q}^{\mathrm{T}} \frac{A_e}{2} \ell_e f \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{3.33a}$$

which is of the form

$$\int_e u^{\mathrm{T}} fA\,dx = \mathbf{q}^{\mathrm{T}} \mathbf{f}^e \tag{3.33b}$$

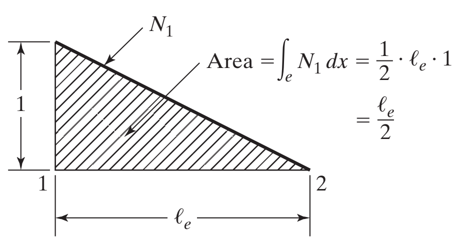


FIGURE 3.9 Integral of a shape function.

The right side of this equation is of the form displacement $\times$ force. Thus, the **element body force vector**, $\mathbf{f}^e$, is identified as

$$\mathbf{f}^e = \frac{A_e \ell_e f}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{3.34}$$

The element body force vector above has a simple physical explanation. Since $A_e \ell_e$ is the volume of the element and f is the body force per unit volume, we see that $A_e \ell_e f$ gives the total body force acting on the element. The factor $\frac{1}{2}$ in Eq. 3.34 tells us that this total body force is equally distributed to the two nodes of the element.

The element *traction force term* $\int_e u^{\mathrm{T}} T\, dx$ appearing in the total potential energy is now considered. We have

$$\int_e u^{\mathrm{T}} T\, dx = \int_e (N_1 q_1 + N_2 q_2) T\, dx \tag{3.35}$$

Since the traction force T is constant within the element, we have

$$\int_e u^{\mathrm{T}} T\, dx = q^{\mathrm{T}} \begin{Bmatrix} T \displaystyle\int_e N_1\, dx \\ T \displaystyle\int_e N_2\, dx \end{Bmatrix} \tag{3.36}$$

We have already shown that $\int_e N_1\, dx = \int_e N_2\, dx = \ell_e/2$. Thus, Eq. 3.33 is of the form

$$\int_e u^{\mathrm{T}} T\, dx = \mathbf{q}^{\mathrm{T}} \mathbf{T}^e \tag{3.37}$$

where the element traction-force vector is given by

$$\mathbf{T}^e = \frac{T \ell_e}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{3.38}$$

We can provide a physical explanation for this equation as was given for the element body force vector.

At this stage, element matrices $\mathbf{k}^e$, $\mathbf{f}^e$, and $\mathbf{T}^e$ have been obtained. After we account for the element connectivity (in Fig. 3.3, for example, $\mathbf{q} = [Q_1, Q_2]^{\mathrm{T}}$ for element 1, $\mathbf{q} = [Q_2, Q_3]^{\mathrm{T}}$ for element 2, etc.), the total potential energy in Eq. 3.21b can be written as

$$\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F} \tag{3.39}$$

where $\mathbf{K}$ is the *global stiffness matrix*, $\mathbf{F}$ is the global load vector, and $\mathbf{Q}$ is the global displacement vector. For example, in the finite element model in Fig. 3.2b, $\mathbf{K}$ is a (5×5) matrix, and $\mathbf{Q}$ and $\mathbf{F}$ are each (5×1) vectors. $\mathbf{K}$ is obtained as follows: Using the element connectivity information, the elements of each $\mathbf{k}^e$ are placed in the appropriate locations in the larger $\mathbf{K}$ matrix, and overlapping elements are then summed. The $\mathbf{F}$ vector is similarly assembled. This process of assembling $\mathbf{K}$ and $\mathbf{F}$ from element stiffness and force matrices is discussed in detail in Section 3.6.

3.5 THE GALERKIN APPROACH

Following the concepts introduced in Chapter 1, we introduce a virtual displacement field

$$\phi = \phi(x) \tag{3.40}$$

and associated virtual strain

$$\epsilon(\phi) = \frac{d\phi}{dx} \tag{3.41}$$

where ϕ is an arbitrary or virtual displacement consistent with the boundary conditions. Galerkin's variational form, given in Eq. 1.43, for the one-dimensional problem considered here, is

$$\int_L \sigma^{\mathrm{T}}\epsilon(\phi)A\,dx - \int_L \phi^{\mathrm{T}} fA\,dx - \int_L \phi^{\mathrm{T}} T\,dx - \sum_i \phi_i P_i = 0 \tag{3.42a}$$

This equation should hold for every ϕ consistent with the boundary conditions. The first term represents the internal virtual work, while the load terms represent the external virtual work.

On the discretized region, Eq. 3.42a becomes

$$\sum_e \int_e \epsilon^{\mathrm{T}} E\epsilon(\phi)A\,dx - \sum_e \int_e \phi^{\mathrm{T}} fA\,dx - \sum_e \int_e \phi^{\mathrm{T}} T\,dx - \sum_i \phi_i P_i = 0 \tag{3.42b}$$

Note that ϵ is the strain due to the actual loads in the problem, while $\epsilon(\phi)$ is a virtual strain. Similar to the interpolation steps in Eqs. 3.10b, 3.17, and 3.19, we express

$$\begin{aligned} \phi &= \mathbf{N}\boldsymbol{\psi} \\ \epsilon(\phi) &= \mathbf{B}\boldsymbol{\psi} \end{aligned} \tag{3.43}$$

where $\boldsymbol{\psi} = [\psi_1, \psi_2]^{\mathrm{T}}$ represents the arbitrary nodal displacements of element e. Also, the global virtual displacements at the nodes are represented by

$$\boldsymbol{\Psi} = [\psi_1, \psi_2, \ldots, \psi_N]^{\mathrm{T}} \tag{3.44}$$

Element Stiffness

Consider the first term, representing internal virtual work, in Eq. 3.42b. Substituting Eq. 3.43 into Eq. 3.42b, and noting that $\epsilon = \mathbf{Bq}$, we get

$$\int_e \epsilon^{\mathrm{T}} E\epsilon(\phi)A\,dx = \int_e \mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} E\,\mathbf{B}\boldsymbol{\psi} A\,dx \tag{3.45}$$

In the finite element model (Section 3.2), the cross-sectional area of element e, denoted by A_e, is constant. Also, $\mathbf{B}$ is a constant matrix. Further, $dx = (\ell_e/2)d\xi$. Thus,

$$\int_e \epsilon^{\mathrm{T}} E\epsilon(\phi)A\,dx = \mathbf{q}^{\mathrm{T}}\left[E_e A_e \frac{\ell_e}{2}\mathbf{B}^{\mathrm{T}}\mathbf{B}\int_{-1}^{1} d\xi\right]\boldsymbol{\psi} \tag{3.46a}$$

$$\begin{aligned} &= \mathbf{q}^{\mathrm{T}}\mathbf{k}^e\boldsymbol{\psi} \\ &= \boldsymbol{\psi}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} \end{aligned} \tag{3.46b}$$

where $\mathbf{k}^e$ is the (symmetric) element stiffness matrix given by

$$\mathbf{k}^e = E_e A_e \ell_e \mathbf{B}^{\mathrm{T}}\mathbf{B} \tag{3.47}$$

Substituting $\mathbf{B}$ from Eq. 3.18, we have

$$\mathbf{k}^e = \frac{E_e A_e}{\ell_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.48}$$

Force Terms

Consider the second term in Eq. 3.42a, representing the virtual work done by the body force in an element. Using $\phi = \mathbf{N}\boldsymbol{\psi}$ and $dx = \ell_e/2d\xi$, and noting that the body force in the element is assumed constant, we have

$$\int_e \phi^{\mathrm{T}} fA\,dx = \int_{-1}^{1} \boldsymbol{\psi}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}} fA_e \frac{\ell_e}{2}\,d\xi \tag{3.49a}$$

$$= \boldsymbol{\psi}^{\mathrm{T}}\mathbf{f}^e \tag{3.49b}$$

where

$$\mathbf{f}^e = \frac{A_e \ell_e f}{2}\begin{Bmatrix} \displaystyle\int_{-1}^{1} N_1 d\xi \\ \displaystyle\int_{-1}^{1} N_2 d\xi \end{Bmatrix} \tag{3.50a}$$

is called the element body force vector. Substituting for $N_1 = (1 - \xi)/2$ and $N_2 = (1 + \xi)/2$, we obtain $\int_{-1}^{1} N_1\,d\xi = 1$. Alternatively, $\int_{-1}^{1} N_1 d\xi$ is the area under the N_1 curve $= \frac{1}{2} \times 2 \times 1 = 1$ and $\int_{-1}^{1} N_2 d\xi = 1$. Thus,

$$\mathbf{f}^e = \frac{A_e \ell_e f}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{3.50b}$$

The element traction term then reduces to

$$\int_e \phi^{\mathrm{T}} T\,dx = \boldsymbol{\psi}^{\mathrm{T}}\mathbf{T}^e \tag{3.51}$$

where the element traction-force vector is given by

$$\mathbf{T}^e = \frac{T\ell_e}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{3.52}$$

At this stage, the element matrices $\mathbf{k}^e$, $\mathbf{f}^e$, and $\mathbf{T}^e$ have been obtained. After accounting for the element connectivity (in Fig. 3.3, for example, $\boldsymbol{\psi} = [\Psi_1, \Psi_2]^{\mathrm{T}}$ for element 1, $\boldsymbol{\psi} = [\Psi_2, \Psi_3]^{\mathrm{T}}$ for element 2, etc.), the variational form

$$\sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} - \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{f}^e - \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{T}^e - \sum_i \Psi_i P_i = 0 \tag{3.53}$$

can be written as

$$\boldsymbol{\Psi}^{\mathrm{T}}(\mathbf{KQ} - \mathbf{F}) = 0 \tag{3.54}$$

which should hold for every $\boldsymbol{\Psi}$ consistent with the boundary conditions. Methods for handling boundary conditions are discussed shortly. The global stiffness matrix $\mathbf{K}$ is assembled from element matrices $\mathbf{k}^e$ using element connectivity information. Likewise, $\mathbf{F}$ is assembled from element matrices $\mathbf{f}^e$ and $\mathbf{T}^e$. This assembly is discussed in detail in the next section.

3.6 ASSEMBLY OF THE GLOBAL STIFFNESS MATRIX AND LOAD VECTOR

We noted earlier that the total potential energy written in the form

$$\Pi = \sum_e \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} - \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{f}^e - \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{T}^e - \sum_i P_i Q_i$$

can be written in the form

$$\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F}$$

by taking element connectivity into account. This step involves assembling $\mathbf{K}$ and $\mathbf{F}$ from element stiffness and force matrices. The assembly of the structural stiffness matrix $\mathbf{K}$ from element stiffness matrices $\mathbf{k}^e$ will first be shown here.

Referring to the finite element model shown in Fig. 3.2b, let us consider the strain energy in, say, element 3. We have

$$U_3 = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^3\mathbf{q} \tag{3.55a}$$

or, substituting for $\mathbf{k}^3$,

$$U_3 = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\frac{E_3A_3}{\ell_3}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\mathbf{q} \tag{3.55b}$$

For element 3, we have $\mathbf{q} = [Q_3, Q_4]^{\mathrm{T}}$. Thus, we can write U_3 as

$$U_3 = \tfrac{1}{2}[Q_1, Q_2, Q_3, Q_4, Q_5]\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{E_3A_3}{\ell_3} & \dfrac{-E_3A_3}{\ell_3} & 0 \\ 0 & 0 & \dfrac{-E_3A_3}{\ell_3} & \dfrac{E_3A_3}{\ell_3} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{Bmatrix} \tag{3.56}$$

From the previous equations, we see that elements of the matrix $\mathbf{k}^3$ occupy the third and fourth rows and columns of the $\mathbf{K}$ matrix. Consequently, when adding element-strain energies, the elements of $\mathbf{k}^e$ are placed in the appropriate locations of the global $\mathbf{K}$ matrix, based on the element connectivity; overlapping elements are simply added. We can denote this assembly symbolically as

$$\mathbf{k} \leftarrow \sum_e \mathbf{k}^e \tag{3.57a}$$

Similarly, the global load vector **F** is assembled from element-force vectors and point loads as

$$\mathbf{F} \leftarrow \sum_e (\mathbf{f}^e + \mathbf{T}^e) + \mathbf{P} \tag{3.57b}$$

The Galerkin approach also gives us the same assembly procedure. An example is now given to illustrate this assembly procedure in detail. In actual computation, **K** is stored in banded or skyline form to take advantage of symmetry and sparsity. This aspect is discussed in Section 3.7 and in greater detail in Chapter 4.

Example 3.2

Consider the bar as shown in Fig. E3.2. For each element i, A_i and ℓ_i are the cross-sectional area and length, respectively. Each element i is subjected to a traction force T_i per unit length and a body force f per unit volume. The units of T_i, f, A_i, and so on are assumed to be consistent. The Young's modulus of the material is E. A concentrated load P_2 is applied at node 2. The structural stiffness matrix and nodal load vector will now be assembled.

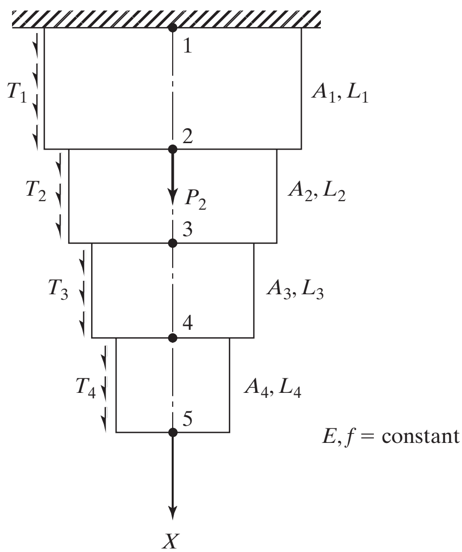


FIGURE E3.2

The element stiffness matrix for each element i is obtained from Eq. 3.26 as

$$[k^{(i)}] = \frac{EA_i}{\ell_i}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

The element connectivity table is given below:

Element	1	2
1	1	2
2	2	3
3	3	4
4	4	5

The element stiffness matrices can be "expanded" using the connectivity table and then summed (or assembled) to obtain the structural stiffness matrix as follows:[1]

$$\mathbf{K} = \frac{EA_1}{\ell_1}\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \frac{EA_2}{\ell_2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+ \frac{EA_3}{\ell_3}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \frac{EA_4}{\ell_4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

which gives

$$\mathbf{K} = E\begin{bmatrix} \frac{A_1}{\ell_1} & -\frac{A_1}{\ell_1} & 0 & 0 & 0 \\ -\frac{A_1}{\ell_1} & \left(\frac{A_1}{\ell_1} + \frac{A_2}{\ell_2}\right) & -\frac{A_2}{\ell_2} & 0 & 0 \\ 0 & -\frac{A_2}{\ell_2} & \left(\frac{A_2}{\ell_2} + \frac{A_3}{\ell_3}\right) & -\frac{A_3}{\ell_3} & 0 \\ 0 & 0 & -\frac{A_3}{\ell_3} & \left(\frac{A_3}{\ell_3} + \frac{A_4}{\ell_4}\right) & -\frac{A_4}{\ell_4} \\ 0 & 0 & 0 & -\frac{A_4}{\ell_4} & \frac{A_4}{\ell_4} \end{bmatrix}$$

The global load vector is assembled as

$$\mathbf{F} = \begin{Bmatrix} \frac{A_1\ell_1 f}{2} + \frac{\ell_1 T_1}{2} \\ \left(\frac{A_1\ell_1 f}{2} + \frac{\ell_1 T_1}{2}\right) + \left(\frac{A_2\ell_2 f}{2} + \frac{\ell_2 T_2}{2}\right) \\ \left(\frac{A_2\ell_2 f}{2} + \frac{\ell_2 T_2}{2}\right) + \left(\frac{A_3\ell_3 f}{2} + \frac{\ell_3 T_3}{2}\right) \\ \left(\frac{A_3\ell_3 f}{2} + \frac{\ell_3 T_3}{2}\right) + \left(\frac{A_4\ell_4 f}{2} + \frac{\ell_4 T_4}{2}\right) \\ \frac{A_4\ell_4 f}{2} + \frac{\ell_4 T_4}{2} \end{Bmatrix} + \begin{Bmatrix} 0 \\ P_2 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

[1]This "expansion" of element stiffness matrices as shown in Example 3.2 is merely for illustration purposes and is never explicitly carried out in the computer, since storing zeroes is inefficient. Instead, **K** is assembled directly from $\mathbf{k}^e$ using the connectivity table. ■

3.7 PROPERTIES OF K

Several important comments will now be made regarding the global stiffness matrix for the linear one-dimensional problem discussed earlier:

1. The dimension of the global stiffness **K** is $(N \times N)$, where N is the number of nodes. This follows from the fact that each node has only one degree of freedom.
2. **K** is symmetric.
3. **K** is a banded matrix. That is, all elements outside of the band are zero. This can be seen in Example 3.2, just considered. In this example, **K** can be compactly represented in banded form as

$$\mathbf{K}_{\text{banded}} = E \begin{bmatrix} \dfrac{A_1}{\ell_1} & -\dfrac{A_1}{\ell_1} \\ \dfrac{A_1}{\ell_1} + \dfrac{A_2}{\ell_2} & -\dfrac{A_2}{\ell_2} \\ \dfrac{A_2}{\ell_2} + \dfrac{A_3}{\ell_3} & -\dfrac{A_3}{\ell_3} \\ \dfrac{A_3}{\ell_3} + \dfrac{A_4}{\ell_4} & -\dfrac{A_4}{\ell_4} \\ \dfrac{A_4}{\ell_4} & 0 \end{bmatrix}$$

Note that $\mathbf{K}_{\text{banded}}$ is of dimension $[N \times \text{NBW}]$, where NBW is the half-bandwidth. In many one-dimensional problems such as the example just considered, the connectivity of element i is $i, i + 1$. In such cases, the banded matrix has only two columns $(\text{NBW} = 2)$. In two and three dimensions, the direct formation of **K** in banded or skyline form from the element matrices involves some bookkeeping. This is discussed in detail at the end of Chapter 4. The reader should verify the following general formula for the half-bandwidth:

$$\text{NBW} = \max\begin{pmatrix}\text{Difference between dof numbers} \\ \text{connecting an element}\end{pmatrix} + 1 \tag{3.58}$$

For example, consider a four-element model of a bar that is numbered as shown in Fig. 3.10a. Using Eq. 3.58, we have

$$\text{NBW} = \max(4 - 1, 5 - 4, 5 - 3, 3 - 2) + 1 = 4$$

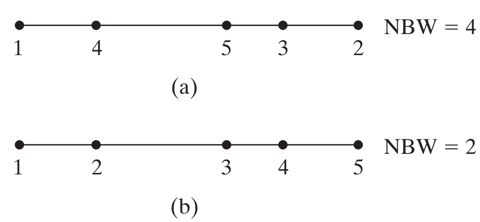


FIGURE 3.10 Node numbering and its effect on the half-bandwidth.

The numbering scheme in Fig. 3.10a is bad since **K** is almost "filled up" and consequently requires more computer storage and computation. Figure 3.10b shows the optimum numbering for minimum NBW.

Now the potential energy or Galerkin approach has to be applied, noting the boundary conditions of the problem, to yield the finite element (equilibrium) equations. Solution of these equations yields the global displacement vector **Q**. The stresses and reaction forces can then be recovered. These steps are discussed in the next section.

3.8 THE FINITE ELEMENT EQUATIONS; TREATMENT OF BOUNDARY CONDITIONS

Finite element equations are now developed after a consistent treatment of the boundary conditions.

Types of Boundary Conditions

After using a discretization scheme to model the continuum, we have obtained an expression for the total potential energy in the body as

$$\Pi = \tfrac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F}$$

where **K** is the structural stiffness matrix, **F** is the global load vector, and **Q** is the global displacement vector. As discussed previously, **K** and **F** are assembled from element stiffness and force matrices, respectively. We now must arrive at the equations of equilibrium, from which we can determine nodal displacements, element stresses, and support reactions.

The minimum potential-energy theorem (Chapter 1) is now invoked. This theorem is stated as follows: *Of all possible displacements that satisfy the boundary conditions of a structural system, those corresponding to equilibrium configurations make the total potential energy assume a minimum value.* Consequently, the equations of equilibrium can be obtained by minimizing, with respect to **Q**, the potential energy $\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F}$ *subject to boundary conditions*. Boundary conditions are usually of the type

$$Q_{P_1} = a_1, Q_{p_2} = a_2, \ \ldots, Q_{p_r} = a_r \tag{3.59}$$

That is, the displacements along dofs $p_1, p_2, \ldots, p_r$ are specified to be equal to $a_1, a_2, \ldots, a_r$, respectively. In other words, there are r number of supports in the structure, with each support node given a specified displacement. For example, consider the bar in Fig. 3.2b. There is only one boundary condition in this problem, $Q_1 = 0$.

It is noted here that *the treatment of boundary conditions in this section is applicable to two- and three-dimensional problems as well.* For this reason, the term dof is used here instead of node, since a two-dimensional stress problem will have two degrees of freedom per node. The steps described in this section will be used in all subsequent chapters. Furthermore, a Galerkin-based argument leads to the same steps for handling boundary conditions as the energy approach used subsequently.

There are *multipoint constraints* of the type

$$\beta_1 Q_{p_1} + \beta_2 Q_{p_2} = \beta_0 \tag{3.60}$$

where β_0, β_1, and β_2 are known constants. These types of boundary conditions are used in modeling inclined roller supports, rigid connections, or shrink fits.

It should be emphasized that improper specification of boundary conditions can lead to erroneous results. Boundary conditions eliminate the possibility of the structure moving as a rigid body. Further, boundary conditions should accurately model the physical system. Two approaches will now be discussed for handling specified displacement boundary conditions of the type given in Eq. 3.59: the **elimination approach** and the **penalty approach**. For multipoint constraints in Eq. 3.60, only the penalty approach will be given, because it is simpler to implement.

Elimination Approach

To illustrate the basic idea, consider the single boundary condition $Q_1 = a_1$. The equilibrium equations are obtained by minimizing Π with respect to $\mathbf{Q}$, subject to the boundary condition $Q_1 = a_1$. For an N-dof structure, we have

$$\mathbf{Q} = [Q_1, Q_2, \ldots, Q_N]^{\mathrm{T}}$$

$$\mathbf{F} = [F_1, F_2, \ldots, F_N]^{\mathrm{T}}$$

The global stiffness matrix is of the form

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & \cdots & K_{2N} \\ \vdots & & & \\ K_{N1} & K_{N2} & \cdots & K_{NN} \end{bmatrix} \tag{3.61}$$

Note that K is a symmetric matrix. The potential energy $\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F}$ can be written in expanded form as

$$\begin{aligned}\Pi = \tfrac{1}{2}(&Q_1K_{11}Q_1 + Q_1K_{12}Q_2 + \cdots + Q_1K_{1N}Q_N \\ &+ Q_2K_{21}Q_1 + Q_2K_{22}Q_2 + \cdots + Q_2K_{2N}Q_N \\ &\text{-----------------------------------} \\ &+ Q_NK_{N1}Q_1 + Q_NK_{N2}Q_2 + \cdots + Q_NK_{NN}Q_N) \\ &- (Q_1F_1 + Q_2F_2 + \cdots + Q_NF_N)\end{aligned} \tag{3.62}$$

If we now substitute the boundary condition $Q_1 = a_1$ into this expression for Π, we obtain

$$\begin{aligned}\Pi = \tfrac{1}{2}(&a_1K_{11}a_1 + a_1K_{12}Q_2 + \cdots + a_1K_{1N}Q_N \\ &+ Q_2K_{21}a_1 + Q_2K_{22}Q_2 + \cdots + Q_2K_{2N}Q_N \\ &\text{-----------------------------------} \\ &+ Q_NK_{N1}a_1 + Q_NK_{N2}Q_2 + \cdots + Q_NK_{NN}Q_N) \\ &- (a_1F_1 + Q_2F_2 + \cdots + Q_NF_N)\end{aligned} \tag{3.63}$$

Note that the displacement Q_1 *has been eliminated* in the potential-energy expression. Consequently, the requirement that Π take on a minimum value implies that

$$\frac{d\Pi}{dQ_i} = 0 \qquad i = 2, 3, \ldots, N \tag{3.64}$$

We thus obtain, from Eqs. 3.60 and 3.61,

$$\begin{aligned} K_{22}Q_2 + K_{23}Q_3 + \cdots + K_{2N}Q_N &= F_2 - K_{21}a_1 \\ K_{32}Q_2 + K_{33}Q_3 + \cdots + K_{3N}Q_N &= F_3 - K_{31}a_1 \\ &\cdots \\ K_{N2}Q_2 + K_{N3}Q_3 + \cdots + K_{NN}Q_N &= F_N - K_{N1}a_1 \end{aligned} \tag{3.65}$$

These *finite element equations* can be expressed in matrix form as

$$\begin{bmatrix} K_{22} & K_{23} & \cdots & K_{2N} \\ K_{32} & K_{33} & \cdots & K_{3N} \\ \vdots & & & \\ K_{N2} & K_{N3} & \cdots & K_{NN} \end{bmatrix} \begin{Bmatrix} Q_2 \\ Q_3 \\ \vdots \\ Q_N \end{Bmatrix} = \begin{Bmatrix} F_2 - K_{21}a_1 \\ F_3 - K_{31}a_1 \\ \vdots \\ F_N - K_{N1}a_1 \end{Bmatrix} \tag{3.66}$$

We now observe that the $(N - 1 \times N - 1)$ stiffness matrix is obtained simply by deleting or *eliminating* the first row and column (in view of $Q_1 = a_1$) from the original $[N \times N]$ stiffness matrix. Equation 3.66 may be denoted as

$$\mathbf{KQ} = \mathbf{F} \tag{3.67}$$

where $\mathbf{K}$ is a reduced stiffness matrix obtained by eliminating the row and column corresponding to the specified or "support" dof. Equation 3.67 can be solved for the displacement vector $\mathbf{Q}$ using Gaussian elimination. Note that the reduced $\mathbf{K}$ matrix is nonsingular, provided the boundary conditions have been specified properly; the original $\mathbf{K}$ matrix, on the other hand, is a singular matrix. Once $\mathbf{Q}$ has been determined, the element stress can be evaluated using Eq. 3.19: $\sigma = E\mathbf{Bq}$, where $\mathbf{q}$ for each element is extracted from $\mathbf{Q}$ using element connectivity information.

Assume that displacements and stresses have been determined. It is now necessary to calculate the **reaction force** R_1 at the support. This reaction force can be obtained from the finite element equation (or equilibrium equation) for node 1:

$$K_{11}Q_1 + K_{12}Q_2 + \cdots + K_{1N}Q_N = F_1 + R_1 \tag{3.68}$$

Here, $Q_1, Q_2, \ldots, Q_N$ are known. F_1, which equals the load applied at the support (if any), is also known. Consequently, the reaction force at the node that maintains equilibrium, is

$$R_1 = K_{11}Q_1 + K_{12}Q_2 + \cdots + K_{1N}Q_N - F_1 \tag{3.69}$$

Note that the elements $K_{11}, K_{12}, \ldots, K_{1N}$, which form the first row of $\mathbf{K}$, need to be *stored separately*. This is because $\mathbf{K}$ in Eq. 3.64 is obtained by deleting this row and column from the original $\mathbf{K}$.

The modifications to $\mathbf{K}$ and $\mathbf{F}$ discussed earlier are also derivable using Galerkin's variational formulation. We have Eq. 3.54 in which

$$\Psi^{\mathrm{T}}(\mathbf{KQ} - \mathbf{F}) = 0 \tag{3.70}$$

for every Ψ consistent with the boundary conditions of the problem. Specifically, consider the constraint

$$Q_1 = a_1 \tag{3.71}$$

Then, we require

$$\Psi_1 = 0 \tag{3.72}$$

Choosing virtual displacements $\Psi = [0, 1, 0, \ldots, 0]$, $\Psi = [0, 0, 1, 0, \ldots, 0]^T, \ldots$, $\Psi = [0, 0, \ldots, 0, 1]^T$, and substituting each of these into Eq. 3.70, we obtain precisely the equilibrium equations given in Eqs. 3.66.

The preceding discussion addressed the boundary condition $Q_1 = a_1$. This procedure can readily be generalized to handle multiple boundary conditions. The general procedure is summarized subsequently. Again, this procedure is also applicable to two- and three-dimensional problems.

Summary: Elimination Approach

Consider the boundary conditions

$$Q_{p_1} = a_1, Q_{p_2} = a_2, \ldots, Q_{p_r} = a_r$$

Step1. Store the p_1th, p_2th, . . ., and p_rth rows of the global stiffness matrix **K** and force vector **F**. These rows will be used subsequently.

Step 2. Delete the p_1th row and column, the p_2th row and column, ..., and the p_rth row and column from the **K** matrix. The resulting stiffness matrix **K** is of dimension $(N - r, N - r)$. Similarly, the corresponding load vector **F** is of dimension $(N - r, 1)$. Modify each load component as

$$F_i = F_i - (K_{i,p_1}a_1 + K_{i,p_2}a_2 + \cdots + K_{i,p_r}a_r) \tag{3.73}$$

for each dof i that is not a support. Solve

$$\mathbf{KQ} = \mathbf{F}$$

for the displacement vector **Q**.

Step 3. For each element, extract the element displacement vector **q** from the **Q** vector, using element connectivity, and determine element stresses.

Step 4. Using the information stored in step 1, evaluate the reaction forces at each support dof from

$$\begin{aligned} R_{p_1} &= K_{p_1 1}Q_1 + K_{p_1 2}Q_2 + \cdots + K_{p_1 N}Q_N - F_{p_1} \\ R_{p_2} &= K_{p_2 1}Q_1 + K_{p_2 2}Q_2 + \cdots + K_{p_2 N}Q_N - F_{p_2} \\ &\text{-----------------------------------} \\ R_{p_r} &= K_{p_r 1}Q_1 + K_{p_r 2}Q_2 + \cdots + K_{p_r N}Q_N - F_{p_r} \end{aligned} \tag{3.74}$$

Example 3.3

Consider the thin (steel) plate in Fig. E3.3a. The plate has a uniform thickness $t = 1$ in., Young's modulus $E = 30 \times 10^6$ psi, and weight density $\rho = 0.2836$ lb/in.^{3}. In addition to its self-weight, the plate is subjected to a point load $P = 100$ lb at its midpoint.

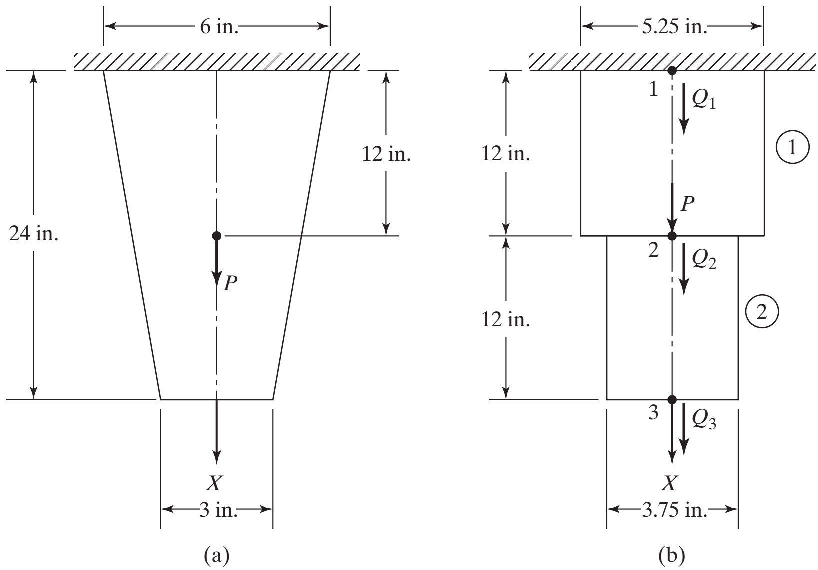


FIGURE E3.3

(a) Model the plate with two finite elements.
(b) Write down expressions for the element stiffness matrices and element body force vectors.
(c) Assemble the structural stiffness matrix **K** and global load vector **F**.
(d) Using the elimination approach, solve for the global displacement vector **Q**.
(e) Evaluate the stresses in each element.
(f) Determine the reaction force at the support.

Solution

(a) Using two elements, each of 12 in. in length, we obtain the finite element model as shown in Fig. E3.3b. Nodes and elements are numbered as shown. Note that the area at the midpoint of the plate in Fig. E3.3a is 4.5 in.^{2}. Consequently, the average area of element 1 is $A_1 = (6 + 4.5)/2 = 5.25$ in.^{2}, and the average area of element 2 is $A_2 = (4.5 + 3)/2 = 3.75$ in.^{2}. The boundary condition for this model is $Q_1 = 0$.

(b) From Eq. 3.29, we can write down expressions for the element stiffness matrices of the two elements as

$$\mathbf{k}^1 = \frac{30 \times 10^6 \times 5.25}{12} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \leftarrow \text{Global dof} \\ \left[\begin{array}{cc} 1 & -1 \\ -1 & 1 \end{array}\right] \begin{array}{c} 1 \\ 2 \end{array} \end{array}$$

and

$$\mathbf{k}^2 = \frac{30 \times 10^6 \times 3.75}{12} \begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 2 \\ 3 \end{array}$$

Using Eq. 3.34, the element body force vectors are

$$\mathbf{f}^1 = \frac{5.25 \times 12 \times 0.2836}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad \begin{array}{c} \text{Global dof} \\ \downarrow \\ 1 \\ 2 \end{array}$$

and

$$\mathbf{f}^2 = \frac{3.75 \times 12 \times 0.2836}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad \begin{array}{c} 2 \\ 3 \end{array}$$

(c) The global stiffness matrix $\mathbf{K}$ is assembled from $\mathbf{k}^1$ and $\mathbf{k}^2$ as

$$\mathbf{K} = \frac{30 \times 10^6}{12} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 5.25 & -5.25 & 0 \\ -5.25 & 9.00 & -3.75 \\ 0 & -3.75 & 3.75 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}$$

The externally applied global load vector $\mathbf{F}$ is assembled from $\mathbf{f}^1$, $\mathbf{f}^2$, and the point load $P = 100$ lb; as

$$\mathbf{F} = \begin{Bmatrix} 8.9334 \\ 15.3144 + 100 \\ 6.3810 \end{Bmatrix}$$

(d) In the elimination approach, the stiffness matrix $\mathbf{K}$ is obtained by deleting rows and columns corresponding to fixed dofs. In this problem, dof 1 is fixed. Thus, $\mathbf{K}$ is obtained by deleting the first row and column of the original $\mathbf{K}$. Also, $\mathbf{F}$ is obtained by deleting the first component of the original $\mathbf{F}$. The resulting equations are

$$\frac{30 \times 10^6}{12} \begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 9.00 & -3.75 \\ -3.75 & 3.75 \end{bmatrix} \end{array} \begin{Bmatrix} Q_2 \\ Q_3 \end{Bmatrix} = \begin{Bmatrix} 115.3144 \\ 6.3810 \end{Bmatrix}$$

Solution of these equations yields

$$Q_2 = 0.9272 \times 10^{-5} \text{ in.}$$
$$Q_3 = 0.9953 \times 10^{-5} \text{ in.}$$

Thus, $Q = [0, 0.9272 \times 10^{-5}, 0.9953 \times 10^{-5}]^{\mathrm{T}}$ in.

(e) Using Eqs. 3.15 and 3.16, we obtain the stress in each element:

$$\sigma_1 = 30 \times 10^6 \times \tfrac{1}{12}[-1 \;\; 1] \begin{Bmatrix} 0 \\ 0.9272 \times 10^{-5} \end{Bmatrix}$$
$$= 23.18 \text{ psi}$$

and

$$\sigma_2 = 30 \times 10^6 \times \tfrac{1}{12}[-1 \;\; 1]\begin{Bmatrix} 0.9272 \times 10^{-5} \\ 0.9953 \times 10^{-5} \end{Bmatrix}$$
$$= 1.70 \text{ psi}$$

(f) The reaction force R_1 at node 1 is obtained from Eq. 3.74. This calculation requires the first row of $\mathbf{K}$ from part (c). Also, from part (c), note that the externally applied load (due to the self-weight) at note 1 is $F_1 = 8.9334$ lb. Thus,

$$R_1 = \frac{30 \times 10^6}{12}[5.25 \;\; -5.25 \;\; 0]\begin{Bmatrix} 0 \\ 0.9272 \times 10^{-5} \\ 0.9953 \times 10^{-5} \end{Bmatrix}^{-8.9334}$$
$$= -130.6 \text{ lb}$$

Evidently, the reaction is equal and opposite to the total downward load on the plate. ■

Penalty Approach

A second approach for handling boundary conditions will now be discussed. This approach is easy to implement in a computer program and retains its simplicity even when considering general boundary conditions as given in Eq. 3.60. Specified displacement boundary conditions will first be discussed. The method will then be shown to apply to problems with multipoint constraints.

Specified Displacement Boundary Conditions. Consider the boundary condition

$$Q_1 = a_1$$

where a_1 is a known specified displacement along dof 1 of the support. The penalty approach for handling this boundary condition is now presented below.

A spring with a large stiffness C is used to model the support. The magnitude of C is discussed subsequently. In this case, one end of the spring is displaced by an amount a_1, as shown in Fig. 3.11. The displacement Q_1 along dof 1 will be approximately equal to a_1, owing to the relatively small resistance offered by the structure. Consequently, the *net* extension of the spring is equal to $(Q_1 - a_1)$. The strain energy in the spring equals

$$U_s = \frac{1}{2}C(Q_1 - a_1)^2 \tag{3.75}$$

This strain energy contributes to the total potential energy. As a result,

$$\Pi_M = \frac{1}{2}\mathbf{Q}^T\mathbf{KQ} + \frac{1}{2}C(Q_1 - a_1)^2 - \mathbf{Q}^T\mathbf{F} \tag{3.76}$$

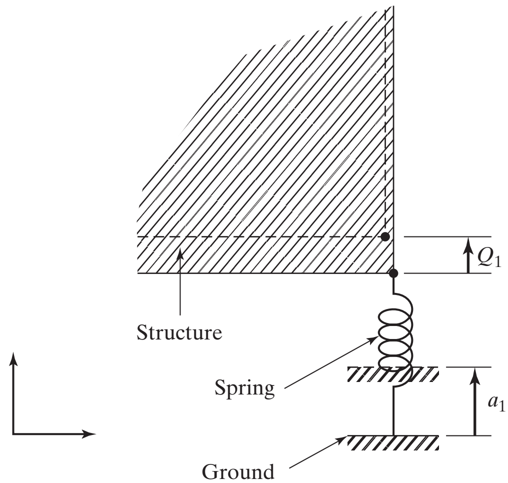


FIGURE 3.11 The penalty approach, where a spring with a large stiffness is used to model the boundary condition $Q_1 = a_1$.

The minimization of Π_M can be carried out by setting $\partial\Pi_M/\partial Q_i = 0, i = 1, 2, \ldots, N$. The resulting finite element equations are

$$\begin{bmatrix} (K_{11} + C) & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & \cdots & K_{2N} \\ \vdots & \vdots & & \vdots \\ K_{N1} & K_{N2} & \cdots & K_{NN} \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{Bmatrix} = \begin{Bmatrix} F_1 + Ca_1 \\ F_2 \\ \vdots \\ F_N \end{Bmatrix} \tag{3.77}$$

Here, we see that the only modifications to handle $Q_1 = a_1$ are that a large number C gets added on to the first diagonal element of **K** and that Ca_1 gets added on to F_1. Solution of Eqs. 3.77 yields the displacement vector **Q**.

The reaction force at node 1 equals the force exerted *by* the spring *on* the structure. Since the net extension of the spring is $(Q_1 - a_1)$, and the spring stiffness is C, the reaction force is given by

$$R_1 = -C(Q_1 - a_1) \tag{3.78}$$

The modifications to **K** and **F** given in Eqs. 3.77 are also derivable using Galerkin approach. Consider the boundary condition $Q_1 = a_1$. To handle this, we introduce a spring with a large stiffness C with the support given a displacement equal to a_1 (Fig. 3.11). The virtual work done by the spring as a result of an arbitrary displacement $\boldsymbol{\Psi}$ is

$$\delta W_s = \text{virtual displacement} \times \text{force in spring}$$

or

$$\delta W_s = \Psi_1 C(Q_1 - a_1) \tag{3.79}$$

Thus, the variational form is

$$\Psi^{\mathrm{T}}(\mathbf{KQ} - \mathbf{F}) + \Psi_1 C(Q_1 - a_1) = 0 \tag{3.80}$$

which should be valid for any $\boldsymbol{\Psi}$. Choosing $\boldsymbol{\Psi} = [1, 0, \ldots, 0]^{\mathrm{T}}$, $\boldsymbol{\Psi} = [0, 1, 0, \ldots, 0]^{\mathrm{T}}, \ldots$ $\boldsymbol{\Psi} = [0, \ldots, 0, 1]^{\mathrm{T}}$ and substituting each in turn into Eq. 3.80, we obtain precisely the modifications shown in Eqs. 3.77. The general procedure is now summarized as follows:

Summary: Penalty Approach

Consider the boundary conditions

$$Q_{p_1} = a_1, Q_{p_2} = a_2, \ldots, Q_{p_r} = a_r$$

Step 1. Modify the structural stiffness matrix **K** by adding a large number C to each of the p_1th, p_2th, ..., and p_rth diagonal elements of **K**. Also, modify the global load vector **F** by adding Ca_1 to F_{p1}, Ca_2 to $F_{p2}, \ldots$, and Ca_r to F_{pr}. Solve **KQ** = **F** for the displacement **Q**, where **K** and **F** are the modified stiffness and load matrices.

Step 2. For each element, extract the element displacement vector **q** from the **Q** vector, using element connectivity, and determine the element stresses.

Step 3. Evaluate the reaction force at each support from

$$R_{p_i} = -C(Q_{p_i} - a_i) \qquad i = 1, 2, \ldots, r \tag{3.81}$$

It should be noted that the penalty approach presented herein is an approximate approach. The accuracy of the solution, particularly the reaction forces, depends on the choice of C.

Choice of C. Let us expand the first equation in Eq. 3.77. We have

$$(K_{11} + C)Q_1 + K_{12}Q_2 + \cdots + K_{1N}Q_N = F_1 + Ca_1 \tag{3.82a}$$

Upon dividing by C, we get

$$\left[\frac{K_{11}}{C} + 1\right]Q_1 + \frac{K_{12}}{C}Q_2 + \cdots + \frac{K_{1N}}{C}Q_N = \frac{F_1}{C} + a_1 \tag{3.82b}$$

From this equation, we see that if C is chosen large enough, then $Q_1 \approx a_1$. Specifically, we see that if C is large compared to the stiffness coefficients $K_{11}, K_{12}, \ldots, K_{1N}$, then $Q_1 \approx a_1$. Note that F_1 is a load applied at the support (if any), and that F_1/C is generally of small magnitude.

A simple scheme suggests itself for choosing the magnitude of C:

$$C = \max|K_{ij}| \times 10^4$$

for

$$\begin{aligned} 1 \le i \le N \\ 1 \le j \le N \end{aligned} \tag{3.83}$$

The choice of 10^4 has been found to be satisfactory on most computers. The reader may wish to choose a sample problem and experiment with this (using, say, 10^5 or 10^6) to check whether the reaction forces differ by much.

Example 3.4

Consider the bar shown in Fig. E3.4. An axial load $P = 200 \times 10^3$ N is applied as shown. Using the penalty approach for handling boundary conditions, do the following:

(a) Determine the nodal displacements.
(b) Determine the stress in each material.
(c) Determine the reaction forces.

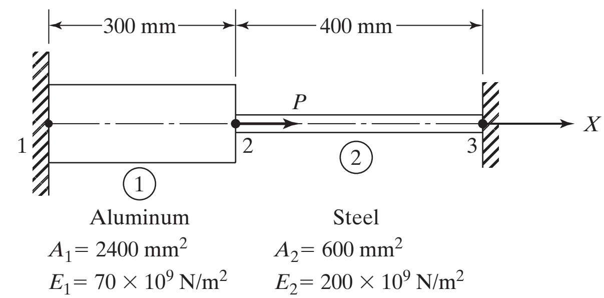


FIGURE E3.4

Solution

(a) The element stiffness matrices are

$$\begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \leftarrow \text{Global dof} \\ \mathbf{k}^1 = \dfrac{70 \times 10^3 \times 2400}{300} & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{array}$$

and

$$\begin{array}{cc} & \begin{array}{cc} 2 & 3 \end{array} \\ \mathbf{k}^2 = \dfrac{200 \times 10^3 \times 600}{400} & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{array}$$

The structural stiffness matrix that is assembled from $\mathbf{k}^1$ and $\mathbf{k}^2$ is

$$\begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \mathbf{K} = 10^6 & \begin{bmatrix} 0.56 & -0.56 & 0 \\ -0.56 & 0.86 & -0.30 \\ 0 & -0.30 & 0.30 \end{bmatrix} \end{array}$$

The global load vector is

$$\mathbf{F} = [0,\ 200 \times 10^3,\ 0]^{\mathrm{T}}$$

Now dofs 1 and 3 are fixed. When using the penalty approach, therefore, a large number C is added to the first and third diagonal elements of **K**. Choosing C based on Eq. 3.83, we get

$$C = [0.86 \times 10^6] \times 10^4$$

Thus, the modified stiffness matrix is

$$\mathbf{K} = 10^6 \begin{bmatrix} 8600.56 & -0.56 & 0 \\ -0.56 & 0.86 & -0.30 \\ 0 & -0.30 & 8600.30 \end{bmatrix}$$

The finite element equations are given by

$$10^6 \begin{bmatrix} 8600.56 & -0.56 & 0 \\ -0.56 & 0.86 & -0.30 \\ 0 & -0.30 & 8600.30 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 200 \times 10^3 \\ 0 \end{Bmatrix}$$

which yields the solution

$$\mathbf{Q} = [15.1432 \times 10^{-6}, 0.23257, 8.1127 \times 10^{-6}]^{\mathrm{T}} \text{ mm}$$

(b) The element stresses (Eq. 3.19) are

$$\sigma_1 = 70 \times 10^3 \times \frac{1}{300}[-1 \quad 1]\begin{Bmatrix} 15.1432 \times 10^{-6} \\ 0.23257 \end{Bmatrix}$$
$$= 54.27 \text{ MPa}$$

where 1 MPa $= 10^6$ N/m^2 $= 1$ N/mm^2. Also,

$$\sigma_2 = 200 \times 10^3 \times \frac{1}{400}[-1 \quad 1]\begin{Bmatrix} 0.23257 \\ 8.1127 \times 10^{-6} \end{Bmatrix}$$
$$= -116.29 \text{ MPa}$$

(c) The reaction forces are obtained from Eq. 3.78 as

$$\begin{aligned} R_1 &= -CQ_1 \\ &= -[0.86 \times 10^{10}] \times 15.1432 \times 10^{-6} \\ &= -130.23 \times 10^3 \end{aligned}$$

Also,

$$\begin{aligned} R_3 &= -CQ_3 \\ &= -[0.86 \times 10^{10}] \times 8.1127 \times 10^{-6} \\ &= -69.77 \times 10^3 \text{ N} \end{aligned}$$

■

Example 3.5

In Fig. E3.5a, a load $P = 60 \times 10^3$ N is applied as shown. Determine the displacement field, stress, and support reactions in the body. Take $E = 20 \times 10^3$ N/mm^2.

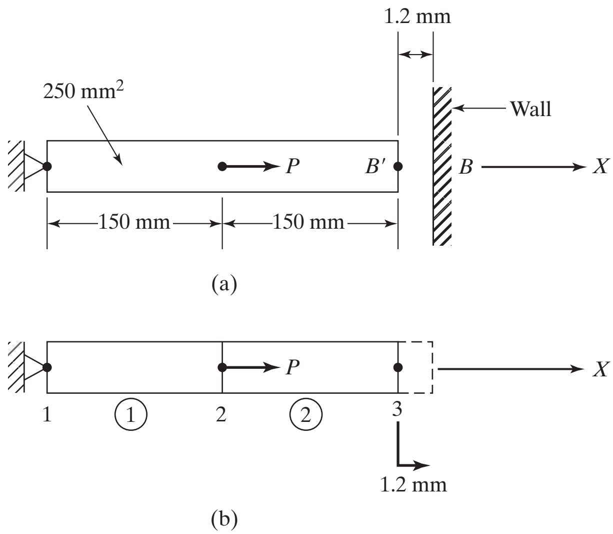


FIGURE E3.5

Solution In this problem, we should first determine whether contact occurs between the bar and the wall, B. To do this, assume that the wall does not exist. Then, the solution to the problem can be verified to be

$$Q_{B'} = 1.8 \text{ mm}$$

where $Q_{B'}$ is the displacement of point B'. From this result, we see that contact does occur. The problem has to be resolved, since the boundary conditions are now different: The displacement at B' is specified to be 1.2 mm. Consider the two-element finite element model in Fig. E3.5b. The boundary conditions are $Q_1 = 0$ and $Q_3 = 1.2$ mm. The structural stiffness matrix **K** is

$$\mathbf{K} = \frac{20 \times 10^3 \times 250}{150} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

and the global load vector **F** is

$$\mathbf{F} = [0, 60 \times 10^3, 0]^{\mathrm{T}}$$

In the penalty approach, the boundary conditions $Q_1 = 0$ and $Q_3 = 1.2$ imply the following modifications: A large number C chosen here as $C = (2/3) \times 10^9$, is added on to the 1st and 3rd diagonal elements of **K**. Also, the number $(C \times 1.2)$ gets added on to the 3rd component of **F**. Thus, the modified equations are

$$\frac{10^5}{3} \begin{bmatrix} 20001 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 20001 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 60.0 \times 10^3 \\ 80.0 \times 10^7 \end{Bmatrix}$$

The solution is

$$\mathbf{Q} = [7.49985 \times 10^{-5},\ 1.500045,\ 1.200015]^{\mathrm{T}}\ \text{mm}$$

The element stresses are

$$\sigma_1 = 200 \times 10^3 \times \frac{1}{150}[-1\ \ 1]\begin{Bmatrix} 7.49985 \times 10^{-5} \\ 1.500045 \end{Bmatrix}$$
$$= 199.996\ \text{MPa}$$

$$\sigma_2 = 200 \times 10^3 \times \frac{1}{150}[-1\ \ 1]\begin{Bmatrix} 1.500045 \\ 1.200015 \end{Bmatrix}$$
$$= -40.004\ \text{MPa}$$

The reaction forces are

$$R_1 = -C \times 7.49985 \times 10^{-5}$$
$$= -49.999 \times 10^3\ \text{N}$$

and

$$R_3 = -C \times (1.200015 - 1.2)$$
$$= -10.001 \times 10^3\ \text{N}$$

The results obtained from the penalty approach have a small approximation error due to the flexibility of the support introduced. In fact, the reader may verify that the elimination approach for handling boundary conditions yields the exact reactions, $R_1 = -50.0 \times 10^3$ N and $R_3 = -10.0 \times 10^3$ N. ■

Multipoint Constraints

In problems where, for example, inclined rollers or rigid connections are to be modeled, the boundary conditions take the form

$$\beta_1 Q_{p_1} + \beta_2 Q_{p_2} = \beta_0$$

where β_0, β_1, and β_2 are known constants. Such boundary conditions are referred to as multipoint constraints in the literature. The penalty approach is shown to apply to this type of boundary condition.

Consider the modified total potential-energy expression

$$\Pi_{\mathrm{M}} = \tfrac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} + \tfrac{1}{2}C(\beta_1 Q_{p_1} + \beta_2 Q_{p_2} - \beta_0)^2 - \mathbf{Q}^{\mathrm{T}}\mathbf{F} \tag{3.84}$$

where C is a large number. Since C is large, Π_{M} takes on a minimum value only when $(\beta_1 Q_{p_1} + \beta_2 Q_{p_2} - \beta_0)$ is very small—that is, when $\beta_1 Q_{p_1} + \beta_2 Q_{p_2} \approx \beta_0$, as desired. Setting $\partial\Pi_{\mathrm{M}}/\partial Q_i = 0$, $i = 1, \ldots, N$ yields the modified stiffness and force matrices. These modifications are given as

$$\begin{bmatrix} K_{p_1p_1} & K_{p_1p_2} \\ K_{p_2p_1} & K_{p_2p_2} \end{bmatrix} \longrightarrow \begin{bmatrix} K_{p_1p_1} + C\beta_1^2 & K_{p_1p_2} + C\beta_1\beta_2 \\ K_{p_2p_1} + C\beta_1\beta_2 & K_{p_2p_2} + C\beta_2^2 \end{bmatrix} \tag{3.85}$$

and

$$\begin{Bmatrix} F_{p_1} \\ F_{p_2} \end{Bmatrix} \longrightarrow \begin{Bmatrix} F_{p_1} + C\beta_0\beta_1 \\ F_{p_2} + C\beta_0\beta_2 \end{Bmatrix} \tag{3.86}$$

If we consider the equilibrium equations $\partial\Pi_M/\partial Q_{p_1} = 0$ and $\partial\Pi_M/\partial Q_{p_2} = 0$ and rearrange these in the form

$$\sum_j K_{p_1 j}Q_j - F_{p_1} = R_{p_1} \quad \text{and} \quad \sum_j K_{p_2 j}Q_j - F_{p_2} = R_{p_2}$$

we obtain the reaction forces R_{p_1} and R_{p_2}, which are the reaction components along dofs p_1 and p_2, respectively, as

$$R_{p_1} = -\frac{\partial}{\partial Q_{p_1}}\left[\frac{1}{2}C(\beta_1 Q_{p_1} + \beta_2 Q_{p_2} - \beta_0)^2\right] \tag{3.87a}$$

and

$$R_{p_2} = -\frac{\partial}{\partial Q_{p_2}}\left[\frac{1}{2}C(\beta_1 Q_{p_1} + \beta_2 Q_{p_2} - \beta_0)^2\right] \tag{3.87b}$$

Upon simplification, Eqs. 3.87 yield

$$R_{p_1} = -C\beta_1(\beta_1 Q_{p_1} + \beta_2 Q_{p_2} - \beta_0) \tag{3.88a}$$

and

$$R_{p_2} = -C\beta_2(\beta_1 Q_{p_1} + \beta_2 Q_{p_2} - \beta_0) \tag{3.88b}$$

We see that the penalty approach allows us to handle multipoint constraints and is again easy to implement in a computer program. A nonphysical argument is used here to arrive at the modified potential energy in Eq. 3.84. Multipoint constraints are the most general types of boundary conditions, from which other types can be treated as special cases. ■

Example 3.6

Consider the structure shown in Fig. E3.6a. A *rigid* bar of negligible mass, pinned at one end, is supported by a steel rod and an aluminum rod. A load $P = 30 \times 10^3$ N is applied as shown.

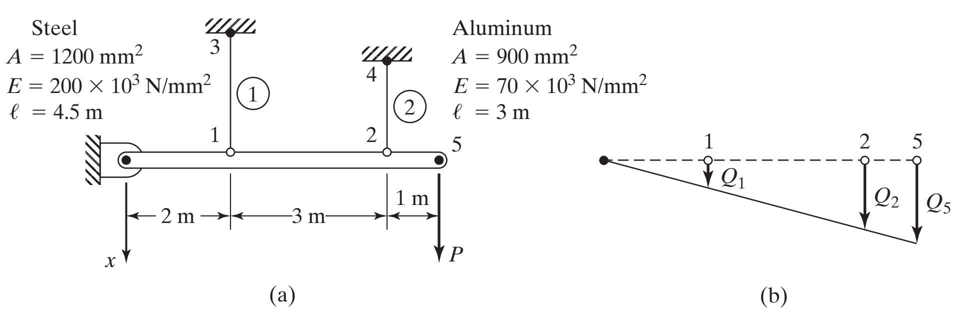


FIGURE E3.6

(a) Model the structure using two finite elements. What are the boundary conditions for your model?
(b) Develop the modified stiffness matrix and modified load vector. Solve the equations for **Q**. Then determine element stresses.

Solution

(a) The problem is modeled using two elements as shown in the following connectivity table:

CONNECTIVITY TABLE

Element no.	Node 1	Node 2
1	3	1
2	4	2

The boundary conditions at nodes 3 and 4 are obvious: $Q_3 = 0$ and $Q_4 = 0$. Now, since the rigid bar has to remain straight, Q_1, Q_2, and Q_5 are related as shown in Fig. E3.6b. The multipoint constraints due to the rigid bar configuration are given by

$$Q_1 - 0.33\dot{3}Q_5 = 0$$
$$Q_2 - 0.83\dot{3}Q_5 = 0$$

(b) First, the element stiffness matrices are given by

$$\mathbf{k}^1 = \frac{200 \times 10^3 \times 1200}{4500}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 10^3 \begin{array}{c} \begin{array}{cc} 3 & 1 \end{array} \\ \begin{bmatrix} 53.3\dot{3} & -53.3\dot{3} \\ -53.3\dot{3} & 53.3\dot{3} \end{bmatrix} \end{array} \begin{array}{c} \\ 3 \\ 1 \end{array}$$

and

$$\mathbf{k}^2 = \frac{70 \times 10^3 \times 900}{3000}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 10^3 \begin{array}{c} \begin{array}{cc} 4 & 2 \end{array} \\ \begin{bmatrix} 21 & -21 \\ -21 & 21 \end{bmatrix} \end{array} \begin{array}{c} \\ 4 \\ 2 \end{array}$$

The global stiffness matrix **K** is

$$\mathbf{K} = 10^3 \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 53.3\dot{3} & 0 & -53.3\dot{3} & 0 & 0 \\ 0 & 21 & 0 & -21 & 0 \\ -53.3\dot{3} & 0 & 53.3\dot{3} & 0 & 0 \\ 0 & -21 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

The **K** matrix is modified as follows: a number $C = [53.3\dot{3} \times 10^3] \times 10^4$, large in comparison to the stiffness values, is chosen. Since $Q_3 = Q_4 = 0$, C is added on to the (3, 3) and (4, 4) locations of **K**. Next, multipoint constraints given in part (a) are considered. For the first constraint, $Q_1 - 0.333Q_5 = 0$, we note that $\beta_0 = 0$, $\beta_1 = 1$, and $\beta_2 = -0.333$. The addition to the stiffness matrix is obtained from Eqs. 3.82 as

$$\begin{bmatrix} C\beta_1^2 & C\beta_1\beta_2 \\ C\beta_1\beta_2 & C\beta_2^2 \end{bmatrix} = 10^7 \begin{array}{c} \begin{array}{cc} 1 & 5 \end{array} \\ \begin{bmatrix} 53.3\dot{3} & -17.7\dot{7} \\ -17.7\dot{7} & 5.925926 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 5 \end{array}$$

The force addition is zero since $\beta_0 = 0$. Similarly, the consideration of the second multipoint constraint $Q_2 - 0.833Q_5 = 0$ yields the stiffness addition

$$\begin{array}{cc}
 & \begin{array}{cc} \quad 2 & \qquad\quad 5 \end{array} \\
10^7 & \begin{bmatrix} 53.3\dot{3} & -44.4\dot{4} \\ -44.4\dot{4} & 37.037037 \end{bmatrix} \begin{array}{c} 2 \\ 5 \end{array}
\end{array}$$

On addition of all the preceding stiffnesses, we obtain the final modified equations as

$$10^3 \begin{bmatrix} 533386.7 & 0 & -53.33 & 0 & -177777.7 \\ 0 & 533354.3 & 0 & -21.0 & -444444.4 \\ -53.33 & 0 & 533386.7 & 0 & 0 \\ 0 & -21.0 & 0 & 533354.3 & 0 \\ -177777.7 & -444444.4 & 0 & 0 & 429629.6 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 30 \times 10^3 \end{Bmatrix}$$

The solution, obtained from a computer program that solves matrix equations, such as the one given in Chapter 2, is

$$\mathbf{Q} = [0.486 \;\; 1.215 \;\; 4.85 \times 10^{-5} \;\; 4.78 \times 10^{-5} \;\; 1.457] \text{ mm}$$

The element stresses are now recovered from Eqs. 3.18 and 3.19 as

$$\sigma_1 = \frac{200 \times 10^3}{4500}[-1 \;\; 1]\begin{Bmatrix} 4.85 \times 10^{-5} \\ 0.486 \end{Bmatrix}$$
$$= 21.60 \text{ MPa}$$

and

$$\sigma_2 = 28.35 \text{ MPa}$$ ■

In this problem, we note that the introduction of the multipoint constraints by the penalty approach makes all the diagonal stiffness values large. Thus, the results become sensitive to errors in the calculations. Double-precision arithmetic on the computer is recommended when there are several multipoint constraints.

3.9 QUADRATIC SHAPE FUNCTIONS

So far, the unknown displacement field was interpolated by linear shape functions within each element. In some problems, however, use of quadratic interpolation leads to far more accurate results. In this section, quadratic shape functions will be introduced, and the corresponding element stiffness matrix and load vectors will be derived. The reader should note that the basic procedure is the same as that used in the linear one-dimensional element earlier.

Consider a typical three-node quadratic element, as shown in Fig. 3.12a. In the local numbering scheme, the left node will be numbered 1, the right node 2, and the midpoint 3. Node 3 has been introduced for the purpose of passing a quadratic fit and is called an *internal node*. The notation x_i = x-coordinate of node i, $i = 1, 2, 3$, is used. Further, $\mathbf{q} = [q_1, q_2, q_3]^{\mathrm{T}}$, where q_1, q_2, and q_3 are the displacements of nodes 1, 2, and 3,

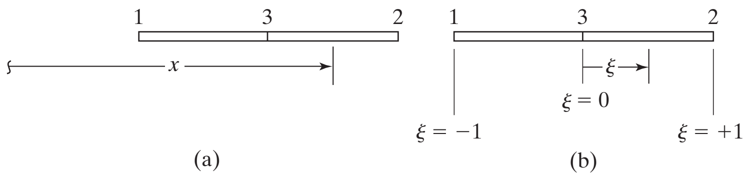


FIGURE 3.12 Quadratic element in x- and ξ-coordinates.

respectively. The x-coordinate system is mapped onto a ξ-coordinate system, which is given by the transformation

$$\xi = \frac{2(x - x_3)}{x_2 - x_1} \tag{3.89}$$

From Eq. 3.89, we see that $\xi = -1, 0,$ and $+1$ at nodes 1, 3, and 2 (Fig. 3.12b). Now, in ξ-coordinates, *quadratic shape functions* N_1, N_2, and N_3 will be introduced as

$$N_1(\xi) = -\tfrac{1}{2}\xi(1 - \xi) \tag{3.90a}$$
$$N_2(\xi) = \tfrac{1}{2}\xi(1 + \xi) \tag{3.90b}$$
$$N_3(\xi) = (1 + \xi)(1 - \xi) \tag{3.90c}$$

The shape function N_1 is equal to unity at node 1 and zero at nodes 2 and 3. Similarly, N_2 equals unity at node 2 and equals zero at the other two nodes; N_3 equals unity at node 3 and equals zero at nodes 1 and 2. The shape functions N_1, N_2, and N_3 are graphed in Fig. 3.13. The expressions for these shape functions can be written down by inspection.

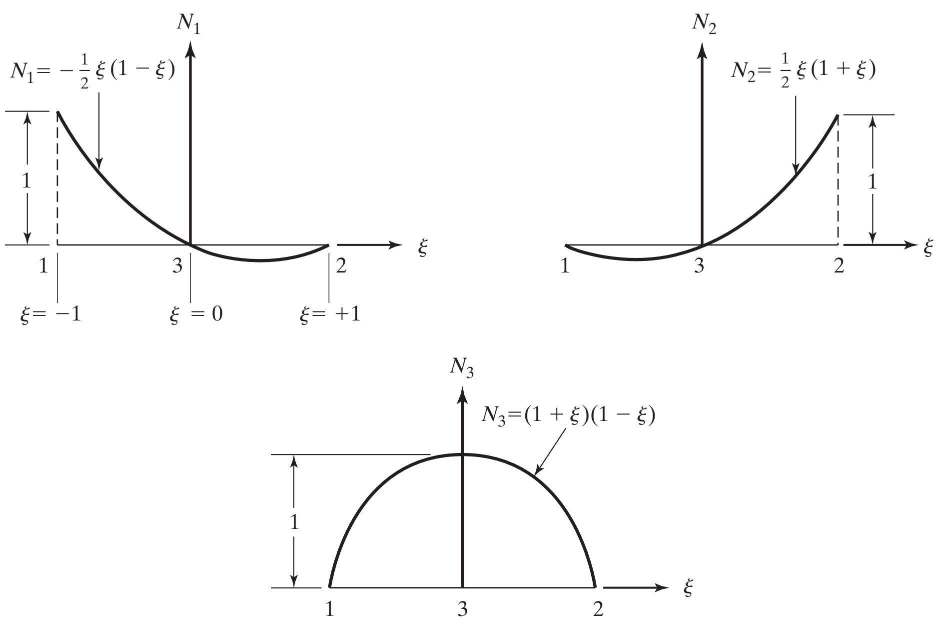


FIGURE 3.13 Shape functions N_1, N_2, and N_3.

For example, since $N_1 = 0$ at $\xi = 0$ and $N_1 = 0$ at $\xi = 1$, we know that N_1 must contain the product $\xi(1 - \xi)$. That is, N_1 is of the form

$$N_1 = c\xi(1 - \xi) \tag{3.91}$$

The constant c is now obtained from the condition $N_1 = 1$ at $\xi = -1$, which yields $c = -\frac{1}{2}$, resulting in the formula given in Eq. 3.90a. These shape functions are called *Lagrange* shape functions.

Now the displacement field within the element is written in terms of the nodal displacements as

$$u = N_1 q_1 + N_2 q_2 + N_3 q_3 \tag{3.92a}$$

or

$$u = \mathbf{N}\mathbf{q} \tag{3.92b}$$

where $\mathbf{N} = [N_1, N_2, N_3]$ is a (1×3) vector of shape functions and $\mathbf{q} = [q_1, q_2, q_3]^{\mathrm{T}}$ is the (3×1) element displacement vector. At node 1, we see that $N_1 = 1$, $N_2 = N_3 = 0$, and hence $u = q_1$. Similarly, $u = q_2$ at node 2 and $u = q_3$ at node 3. Thus, u in Eq. 3.92a is a quadratic interpolation passing through q_1, q_2, and q_3 (Fig. 3.14).

The strain ϵ is now given by

$$\begin{aligned}
\epsilon &= \frac{du}{dx} && \text{(strain−displacement relation)} \\
&= \frac{du}{d\xi}\frac{d\xi}{dx} && \text{(chain rule)} \\
&= \frac{2}{x_2 - x_1}\frac{du}{d\xi} && \text{(using Eq. 3.89)} \\
&= \frac{2}{x_2 - x_1}\left[\frac{dN_1}{d\xi}, \frac{dN_2}{d\xi}, \frac{dN_3}{d\xi}\right] \cdot \mathbf{q} && \text{(using Eq. 3.92)}
\end{aligned} \tag{3.93}$$

Using Eqs. 3.90, we have

$$\epsilon = \frac{2}{x_2 - x_1}\left[-\frac{1 - 2\xi}{2}, \frac{1 + 2\xi}{2}, -2\xi\right]\mathbf{q} \tag{3.94}$$

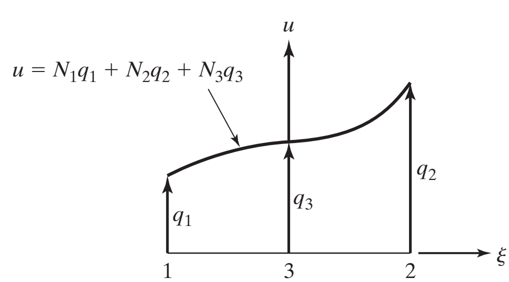


FIGURE 3.14 Interpolation using quadratic shape functions.

which is of the form

$$\epsilon = \mathbf{Bq} \tag{3.95}$$

where $\mathbf{B}$ is given by

$$\mathbf{B} = \frac{2}{x_2 - x_1}\left[-\frac{1-2\xi}{2}, \frac{1+2\xi}{2}, -2\xi\right] \tag{3.96}$$

Using Hooke's law, we can write the stress as

$$\sigma = E\mathbf{Bq} \tag{3.97}$$

Note that since N_i are quadratic shape functions, $\mathbf{B}$ in Eq. 3.96 is linear in ξ. *This means that the strain and stress can vary linearly within the element*. Recall that when using linear shape functions, the strain and stress came out to be constant within the element.

We now have expressions for u, ϵ, and σ in Eqs. 3.92b, 3.95, and 3.97, respectively. Also, we have $dx = (\ell_e/2)d\xi$ from Eq. 3.89.

Again, in the finite element model considered here, it will be assumed that cross-sectional area A_e, body force F, and traction force T are constant within the element. Substituting for u, ϵ, σ, and dx into the potential-energy expression yields

$$\begin{aligned}
\Pi &= \sum_e \frac{1}{2}\int_e \sigma^{\mathrm{T}}\epsilon A\, dx - \sum_e \int_e u^{\mathrm{T}} fA\, dx - \sum_e \int_e u^{\mathrm{T}} T\, dx - \sum_i Q_i P_i \\
&= \sum_e \frac{1}{2}\mathbf{q}^{\mathrm{T}}\left(E_e A_e \frac{\ell_e}{2}\int_{-1}^{1}[\mathbf{B}^{\mathrm{T}}\mathbf{B}]\, d\xi\right)\mathbf{q} - \sum_e \mathbf{q}^{\mathrm{T}}\left(A_e \frac{\ell_e}{2} f \int_{-1}^{1} N^{\mathrm{T}}\, d\xi\right) \\
&\quad - \sum_e \mathbf{q}^{\mathrm{T}}\left(\frac{\ell_e}{2} T \int_{-1}^{1} N^{\mathrm{T}}\, d\xi\right) - \sum_i Q_i P_i
\end{aligned} \tag{3.98}$$

Comparing the above equation with the general form

$$\Pi = \sum_e \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} - \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{f}^e - \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{T}^e - \sum_i Q_i P_i$$

yields

$$\mathbf{k}^e = \frac{E_e A_e \ell_e}{2}\int_{-1}^{1}[\mathbf{B}^{\mathrm{T}}\ \mathbf{B}]d\xi \tag{3.99a}$$

which, upon substituting for $\mathbf{B}$ in Eq. 3.96, yields

$$\mathbf{k}^e = \frac{E_e A_e}{3\ell_e}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \ \leftarrow \text{Local dof} \\ \begin{bmatrix} 7 & 1 & -8 \\ 1 & 7 & -8 \\ -8 & -8 & 16 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \end{array} \tag{3.99b}$$

The element body force vector $\mathbf{f}^e$ is given by

$$\mathbf{f}^e = \frac{A_e \ell_e f}{2} \int_{-1}^{1} \mathbf{N}^{\mathrm{T}} d\xi \tag{3.100a}$$

which, upon substituting for $\mathbf{N}$ in Eqs. 3.87, yields

$$\mathbf{f}^e = A_e \ell_e f \begin{array}{c} \\ \left\{ \begin{array}{c} 1/6 \\ 1/6 \\ 2/3 \end{array} \right\} \end{array} \begin{array}{l} \downarrow \text{Local dof} \\ 1 \\ 2 \\ 3 \end{array} \tag{3.100b}$$

Similarly, the element traction-force vector $\mathbf{T}^e$ is given by

$$\mathbf{T}^e = \frac{\ell_e T}{2} \int_{-1}^{1} \mathbf{N}^{\mathrm{T}} d\xi \tag{3.101a}$$

which results in

$$\mathbf{T}^e = \ell_e T \begin{array}{c} \\ \left\{ \begin{array}{c} 1/6 \\ 1/6 \\ 2/3 \end{array} \right\} \end{array} \begin{array}{l} \downarrow \text{Local dof} \\ 1 \\ 2 \\ 3 \end{array} \tag{3.101b}$$

The total potential energy is again of the form $\Pi = (1/2)\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F}$, where the structural stiffness matrix $\mathbf{K}$ and nodal load vector $\mathbf{F}$ are assembled from element stiffness matrices and load vectors, respectively.

Example 3.7

Consider the rod (a robot arm) in Fig. E3.7a, which is rotating at constant angular velocity $\omega = 30$ rad/s. Determine the axial stress distribution in the rod, using two quadratic elements. Consider only the centrifugal force. Ignore bending of the rod.

Solution A finite element model of the rod, with two quadratic elements, is shown in Fig. E3.7b. The model has a total of five degrees of freedom. The element stiffness matrices are (from Eq. 3.99b)

$$\mathbf{k}^1 = \frac{10^7 \times 0.6}{3 \times 21} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 2 \end{array} \\ \left[\begin{array}{ccc} 7 & 1 & -8 \\ 1 & 7 & -8 \\ -8 & -8 & 16 \end{array} \right] \end{array} \begin{array}{l} \text{Global dof} \\ 1 \\ 3 \\ 2 \end{array}$$

and

$$\mathbf{k}^2 = \frac{10^7 \times 0.6}{3 \times 21} \begin{array}{c} \begin{array}{ccc} 3 & 5 & 4 \end{array} \\ \left[\begin{array}{ccc} 7 & 1 & -8 \\ 1 & 7 & -8 \\ -8 & -8 & 16 \end{array} \right] \end{array} \begin{array}{l} \\ 3 \\ 5 \\ 4 \end{array}$$

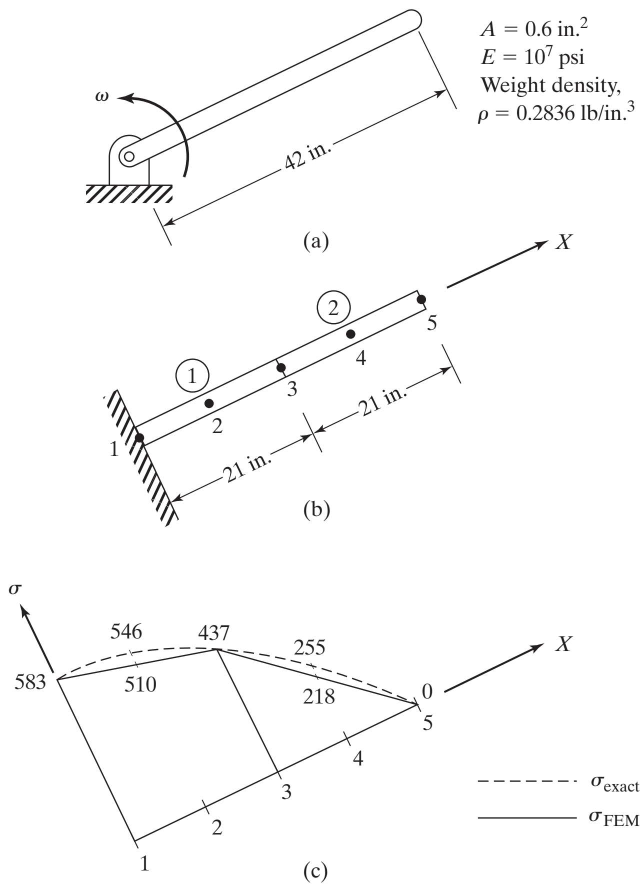


FIGURE E3.7

Thus,

$$
\mathbf{K} = \frac{10^7 \times 0.6}{3 \times 21}
\begin{array}{c}
\begin{array}{ccccc} \mathsf{1} & \mathsf{2} & \mathsf{3} & \mathsf{4} & \mathsf{5} \end{array} \\
\left[\begin{array}{ccccc}
7 & -8 & 1 & 0 & 0 \\
-8 & 16 & -8 & 0 & 0 \\
1 & -8 & 14 & -8 & 1 \\
0 & 0 & -8 & 16 & -8 \\
0 & 0 & 1 & -8 & 7
\end{array}\right]
\end{array}
\begin{array}{c} \\ \mathsf{1} \\ \mathsf{2} \\ \mathsf{3} \\ \mathsf{4} \\ \mathsf{5} \end{array}
$$

The body force $f(\text{lb/in.}^3)$ is given by

$$f = \frac{\rho r \omega^2}{g} \text{lb/in.}^3$$

where ρ = weight density and $g = 32.2\ \text{ft/s}^2$. Note that f is a function of the distance r from the pin. Taking average values of f over each element, we have

$$f_1 = \frac{0.2836 \times 10.5 \times 30^2}{32.2 \times 12} = 6.94$$

and

$$f_2 = \frac{0.2836 \times 31.5 \times 30^2}{32.2 \times 12} = 20.81$$

Thus, the element body force vectors are (from Eq. 3.97b)

$$\mathbf{f}^1 = 0.6 \times 21 \times f_1 \left\{\begin{array}{c} \frac{1}{6} \\ \frac{1}{6} \\ \frac{2}{3} \end{array}\right\} \begin{array}{l} \downarrow \text{Global dof} \\ 1 \\ 3 \\ 2 \end{array}$$

and

$$\mathbf{f}^2 = 0.6 \times 21 \times f_2 \left\{\begin{array}{c} \frac{1}{6} \\ \frac{1}{6} \\ \frac{2}{3} \end{array}\right\} \begin{array}{l} \downarrow \text{Global dof} \\ 3 \\ 5 \\ 4 \end{array}$$

Assembling $\mathbf{f}^1$ and $\mathbf{f}^2$, we obtain

$$\mathbf{F} = [14.57, \quad 58.26, \quad 58.26, \quad 174.79, \quad 43.70]^{\mathrm{T}}$$

Using the elimination method, the finite element equations are

$$\frac{10^7 \times 0.6}{63}\begin{bmatrix} 16 & -8 & 0 & 0 \\ -8 & 14 & -8 & 1 \\ 0 & -8 & 16 & -8 \\ 0 & 1 & -8 & 7 \end{bmatrix}\begin{Bmatrix} Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{Bmatrix} = \begin{Bmatrix} 58.26 \\ 58.26 \\ 174.79 \\ 43.7 \end{Bmatrix}$$

which yields

$$\mathbf{Q} = 10^{-3}[0, \quad .5735, \quad 1.0706, \quad 1.4147, \quad 1.5294]^{\mathrm{T}}\ \text{mm}$$

The stresses can be evaluated from Eqs. 3.96 and 3.97. The element connectivity table is as follows:

Element no.	1	2	3	← Local node nos.
1	1	3	2	↕ Global node nos.
2	3	5	4	

Thus,

$$\mathbf{q} = [Q_1, Q_3, Q_2]^{\mathrm{T}}$$

for element 1, while

$$\mathbf{q} = [Q_3, Q_5, Q_4]^{\mathrm{T}}$$

for element 2. Using Eqs. 3.96 and 3.97, we get

$$\sigma_1 = 10^7 \times \frac{2}{21}\left[-\frac{1-2\xi}{2}, \frac{1+2\xi}{2}, -2\xi\right]\begin{Bmatrix} Q_1 \\ Q_3 \\ Q_2 \end{Bmatrix}$$

where $-1 \leq \xi \leq 1$, and σ_1 denotes the stress in element 1. The stress at node 1 in element 1 is obtained by substituting $\xi = -1$ into the previous equation, which results in

$$\sigma_1|_1 = 10^7 \times \frac{2}{21} \times 10^{-3}[-1.5, -0.5, +2.0]\begin{Bmatrix} 0 \\ 1.0706 \\ .5735 \end{Bmatrix}$$
$$= 583 \text{ psi}$$

The stress at node 2 (which corresponds to the midpoint of element 1) is obtained by substituting for $\xi = 0$:

$$\sigma_1|_3 = 10^7 \times \frac{2}{21} \times 10^{-3}[-0.5, 0.5, 0]\begin{Bmatrix} 0 \\ 1.0706 \\ .5735 \end{Bmatrix}$$
$$= 510 \text{ psi}$$

Similarly, we obtain

$$\sigma_1|_2 = \sigma_2|_1 = 437 \text{ psi} \quad \sigma_2|_3 = 218 \text{ psi} \quad \sigma_2|_2 = 0$$

The axial distribution is shown in Fig. E3.7c. The stresses obtained from the finite element model can be compared with the exact solution, given by

$$\sigma_{\text{exact}}(x) = \frac{\rho\omega^2}{2g}(L^2 - x^2)$$

The exact stress distribution based on this equation is also shown in Fig. E3.7c. ■

3.10 TEMPERATURE EFFECTS

In this section, the stresses induced by temperature changes in an isotropic linearly elastic material will be considered. That is, the **thermal stress** problem will be considered. If the distribution of the change in temperature, $\Delta T(x)$, is known, then the strain due to this temperature change can be treated as an **initial strain**, ϵ_0, given as

$$\epsilon_0 = \alpha \Delta T \tag{3.102}$$

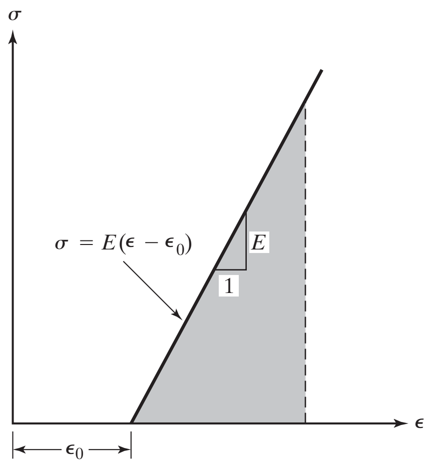


FIGURE 3.15 Stress–strain law in the presence of an initial strain.

where α is the coefficient of thermal expansion. Note that a positive ΔT implies a rise in temperature. The stress–strain law in the presence of ϵ_0 is shown in Fig. 3.15. From this figure, we see that the stress–strain relation is given by

$$\sigma = E(\epsilon - \epsilon_0) \tag{3.103}$$

The strain energy per unit volume, u_0, is equal to the shaded area in Fig. 3.15 and is given by

$$u_0 = \frac{1}{2}\sigma(\epsilon - \epsilon_0) \tag{3.104}$$

By using Eq. 3.103, we find that Eq. 3.104 yields

$$u_0 = \frac{1}{2}(\epsilon - \epsilon_0)^{\mathrm{T}}E(\epsilon - \epsilon_0) \tag{3.105a}$$

The total strain energy U in the structure is obtained by integrating u_0 over the volume of the structure:

$$U = \int_L \frac{1}{2}(\epsilon - \epsilon_0)^{\mathrm{T}}E(\epsilon - \epsilon_0)A\,dx \tag{3.105b}$$

For a structure modeled using one-dimensional linear elements, this equation becomes

$$U = \sum_e \frac{1}{2}A_e\frac{\ell_e}{2}\int_{-1}^{1}(\epsilon - \epsilon_0)^{\mathrm{T}}E_e(\epsilon - \epsilon_0)d\xi \tag{3.105c}$$

Noting that $\boldsymbol{\epsilon} = \mathbf{Bq}$, we get

$$U = \sum_e \frac{1}{2}\mathbf{q}^{\mathrm{T}}\left(E_e A_e \frac{\ell_e}{2}\int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\mathbf{B}\, d\xi\right)\mathbf{q} - \sum_e \mathbf{q}^{\mathrm{T}} E_e A_e \frac{\ell_e}{2}\epsilon_0 \int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\, d\xi$$
$$+ \sum_e \frac{1}{2} E_e A_e \frac{\ell_e}{2}\epsilon_0^2 \tag{3.105d}$$

Examining the strain energy expression, we see that the first term on the right side yields the element stiffness matrix derived earlier in Section 3.4; the last term is a constant term and is of no consequence since it drops out of the equilibrium equations, which are obtained by setting $d\Pi/d\mathbf{Q} = 0$. The second term yields the desired element load vector Θ^e, as a result of the temperature change:

$$\boldsymbol{\Theta}^e = E_e A_e \frac{\ell_e}{2}\epsilon_0 \int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\, d\xi \tag{3.106a}$$

This equation can be simplified by substituting for $\mathbf{B} = [-1 \quad 1]/(x_2 - x_1)$ and noting that $\epsilon_0 = \alpha\Delta T$. Thus,

$$\boldsymbol{\Theta}^e = \frac{E_e A_e \ell_e \alpha\, \Delta T}{x_2 - x_1}\begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \tag{3.106b}$$

In Eq. 3.106b, ΔT is the average change in temperature within the element. The temperature load vector in Eq. 3.106b can be assembled along with the body force, traction-force, and point load vectors to yield the global load vector $\mathbf{F}$, for the structure. This assembly can be denoted as

$$\mathbf{F} = \sum_e (\mathbf{f}^e + \mathbf{T}^e + \boldsymbol{\Theta}^e) + \mathbf{P} \tag{3.107}$$

After solving the finite element equations $\mathbf{KQ} = \mathbf{F}$ for the displacements $\mathbf{Q}$, the stress in each element can be obtained from Eq. 3.103 as

$$\sigma = E(\mathbf{Bq} - \alpha\Delta T) \tag{3.108a}$$

or

$$\sigma = \frac{E}{x_2 - x_1}[-1 \quad 1]\mathbf{q} - E\alpha\Delta T \tag{3.108b}$$

Example 3.8

An axial load $P = 300 \times 10^3$ N is applied at 20°C to the rod as shown in Fig. E3.8. The temperature is then raised to 60°C.

(a) Assemble the $\mathbf{K}$ and $\mathbf{F}$ matrices.
(b) Determine the nodal displacements and element stresses.

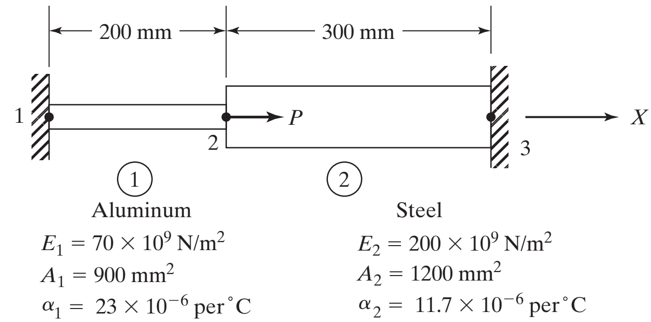


FIGURE E3.8

Solution

(a) The element stiffness matrices are

$$\mathbf{k}^1 = \frac{70 \times 10^3 \times 900}{200} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{N/mm}$$

$$\mathbf{k}^2 = \frac{200 \times 10^3 \times 1200}{300} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{N/mm}$$

Thus,

$$\mathbf{K} = 10^3 \begin{bmatrix} 315 & -315 & 0 \\ -315 & 1115 & -800 \\ 0 & -800 & 800 \end{bmatrix} \text{N/mm}$$

Now, in assembling **F**, both temperature and point load effects have to be considered. The element temperature forces due to $\Delta T = 40°\text{C}$ are obtained from Eq. 3.106b as

$$\boldsymbol{\Theta}^1 = 70 \times 10^3 \times 900 \times 23 \times 10^{-6} \times 40 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \begin{array}{l} \downarrow \text{Global dof} \\ 1 \\ 2 \end{array} \text{N}$$

and

$$\boldsymbol{\Theta}^2 = 200 \times 10^3 \times 1200 \times 11.7 \times 10^{-6} \times 40 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \begin{array}{l} 2 \\ 3 \end{array} \text{N}$$

Upon assembling $\boldsymbol{\Theta}^1$, $\boldsymbol{\Theta}^2$, and the point load, we get

$$\mathbf{F} = 10^3 \begin{Bmatrix} -57.96 \\ 57.96 \quad - 112.32 \ + 300 \\ 112.32 \end{Bmatrix}$$

or

$$\mathbf{F} = 10^3[-57.96, \quad 245.64, \quad 112.32]^{\mathrm{T}}\ \text{N}$$

(b) The elimination approach will now be used to solve for the displacements. Since dof 1 and 3 are fixed, the first and third rows and columns of **K**, together with the first and third components of **F**, are deleted. This results in the scalar equation

$$10^3[1115]Q_2 = 10^3 \times 245.64$$

yielding

$$Q_2 = 0.220 \text{ mm}$$

Thus,

$$\mathbf{Q} = [0, \quad 0.220, \quad 0]^{\mathrm{T}} \text{ mm}$$

In evaluating element stresses, we have to use Eq. 3.108b:

$$\sigma_1 = \frac{70 \times 10^3}{200}[-1 \quad 1]\begin{Bmatrix} 0 \\ 0.220 \end{Bmatrix} - 70 \times 10^3 \times 23 \times 10^{-6} \times 40$$
$$= 12.60 \text{ MPa}$$

and

$$\sigma_2 = \frac{200 \times 10^3}{300}[-1 \quad 1]\begin{Bmatrix} 0.220 \\ 0 \end{Bmatrix} - 200 \times 10^3 \times 11.7 \times 10^{-6} \times 40$$
$$= -240.27 \text{ MPa}$$

■

3.11 PROBLEM MODELING AND BOUNDARY CONDITIONS

We now present some considerations needed in problem modeling.

Problem in Equilibrium

A one-dimensional bar problem in equilibrium shown in Fig. 3.16a does not show any boundary conditions. Modeling without fixing a node gives a singular global stiffness matrix and the program gives a division by zero error. The first step is to check

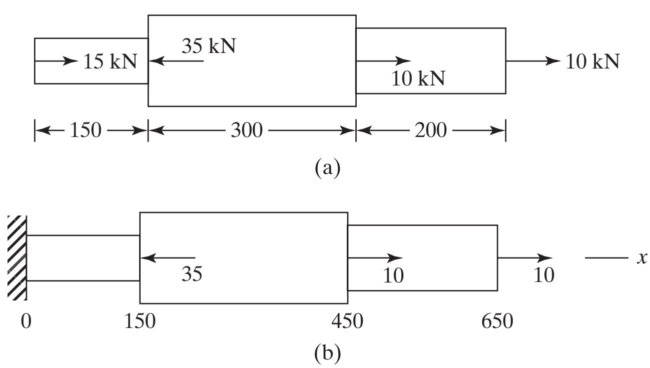


FIGURE 3.16 Bar in equilibrium.

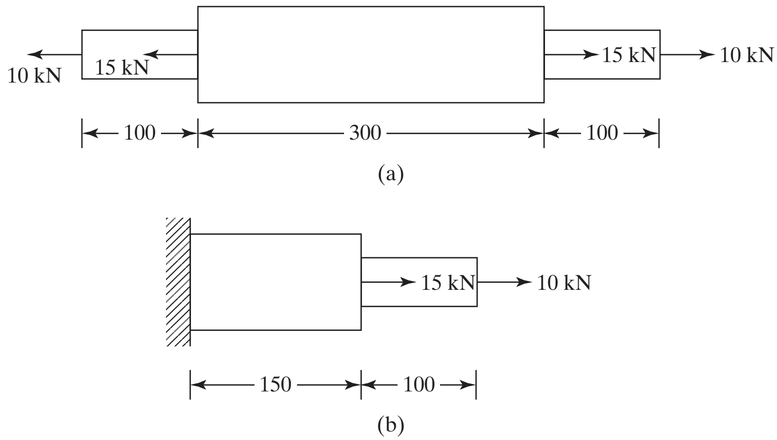


FIGURE 3.17 Symmetric problem.

equilibrium by adding the loads and checking if the sum is zero. We then need to define a coordinate system and fix an end node say 1, as shown in Fig. 3.16b.

Symmetry

If the problem has symmetry of geometry and also symmetry of loading as shown in Fig. 3.17a, complete stresses and displacements can be obtained by considering half the problem as shown in Fig. 3.17b. The node on the line of symmetry is fixed.

Two Elements with Same End Displacements

A problem with two elements having same end positions is shown in Fig. 3.18a. While this problem can be modeled using a multipoint constraint, using the same node numbers for the two elements is a simpler solution as shown in Fig. 3.18b.

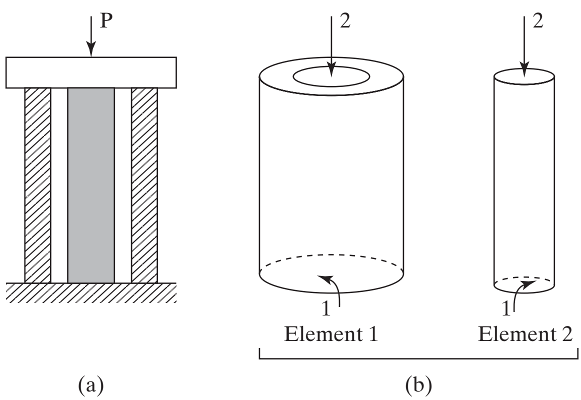


FIGURE 3.18 Elements with same node numbers.

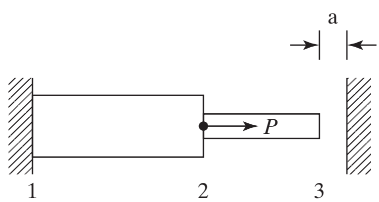


FIGURE 3.19 Problem with gap at a node.

Problems with a Closing Gap

A bar problem with a closing gap is shown in Fig. 3.19. This case has already been presented in Example 3.5. This problem needs to be solved as follows. First we ignore the gap and solve it. If the displacement at the gap node is less than or equal to the gap, we are done. If not, the problem is solved again applying a displacement boundary condition of gap node displacement to be equal to the gap.

Other problems may have a combination of the above situations or other challenges.

Input Data/Output

```
INPUT FOR FEM1D
PROGRAM FEM1D  << BAR ANALYSIS >>
EXAMPLE 3.3
NN      NE      NM      NDIM    NEN     NDN
 3       2      1       1       2       1
ND      NL      NMPC
 1       3      0
NODE#   X-COORD
 1       0
 2      12
 3      24
EL#     N1      N2      MAT#    AREA    TEMP RISE
 1       1      2       1       5.25        0
 2       2      3       1       3.75        0
DOF#    DISP
 1      0
DOF#    LOAD
 1        8.9334
 2      115.3144
 3        6.381
MAT#       E          Alpha
 1      3.00E+07       0
B1      I       B2      J       B3   >== MultiPointContstraints
```

```
OUTPUT FROM FEM1D
Results from Program FEM1D
EXAMPLE 3.3
Node#          Displacement
1              5.80572E-10
2              9.27261E-06
3              9.95325E-06
Element#       Stress
1              23.18007619
2              1.7016
Node#          Reaction
1              -130.6288
```

PROBLEMS

3.1. Consider the bar shown in Fig. P3.1. Cross-sectional area $A_2 = 1\ \text{in.}^2$, and Young's modulus $E = 2.3 \times 10^6$ psi. If $q_1 = 0.05$ in. and $q_2 = 0.06$ in., determine the following (by hand calculation):

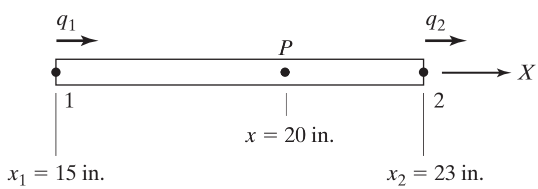


FIGURE P3.1

(a) the displacement at point P,

(b) the strain ϵ and stress σ,

(c) the element stiffness matrix, and

(d) the strain energy in the element.

3.2. Find the bandwidth NBW for the one-dimensional model whose nodes are numbered as shown in Fig. P3.2.

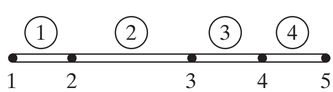


FIGURE P3.2

3.3. A finite element solution using one-dimensional, two-noded elements has been obtained for a rod as shown in Fig. P3.3:

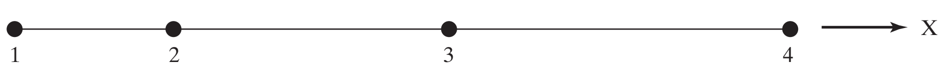


FIGURE P3.3

Displacements are as follows: $\mathbf{Q} = [-0.4, \quad 0, \quad 0.8, \quad -0.3]^T$ mm, $E = 2 \times 10^5$ N/mm^2, Area of each element $= 5$ mm^2, $L_{1-2} = 25$ mm, $L_{2-3} = 60$ mm, $L_{3-4} = 80$ mm.

(a) According to the finite element theory, plot displacement $u(x)$ vs. x.

(b) According to the finite element theory, plot strain $\epsilon(x)$ vs. x.

(c) Determine the $\mathbf{B}$ matrix for element 2–3.

(d) Determine the strain energy in element 1–2.

3.4. Answer concisely:

(a) Element stiffness matrices $[\mathbf{k}]$ are always nonsingular—true or false? Make your choice and justify.

(b) The strain energy in a structure, $U = ½\, \mathbf{Q}^T \mathbf{K} \mathbf{Q}$ is always > 0 for any $\mathbf{Q}$, provided

____

(Fill in the blanks.)

(c) A finite element model of a rod is given a displacement $\mathbf{Q} = [1, 1, \ldots, 1]^T$. The associated strain energy $U = ½\, \mathbf{Q}^T \mathbf{K} \mathbf{Q}$ equals zero. What can you conclude regarding the stiffness matrix $\mathbf{K}$?

(d) Consider a rod fixed at its ends, $x = 0$ and $x = 1$, respectively. Is the axial displacement field $u = (x-1)^2$ "kinematically admissible" ?

3.5. Express the height $h(x)$ in Fig. P3.5 in terms of three unknown coefficients.

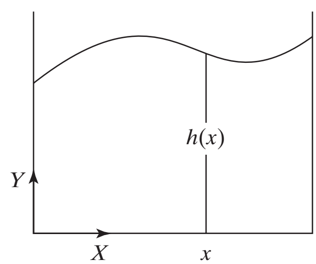


FIGURE P3.5

3.6. Consider a finite element with shape functions $N_1(\xi)$ and $N_2(\xi)$ used to interpolate the displacement field within the element (Fig. P3.6).

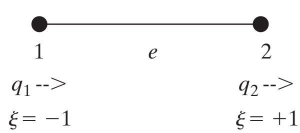


FIGURE P3.6

Derive an expression for the strain–displacement matrix $\mathbf{B}$, where strain $\epsilon = \mathbf{Bq}$, *in terms* of N_1 and N_2. (Do not assume any specific form for N_1 or N_2.) (Note: $\mathbf{q} = [q_1, q_2]^T$.)

3.7. It is desired to attach a spring to node 22 of a structure modeled using one-dimensional elements, as shown in Fig. P3.7. The banded stiffness matrix **S** in program FEM1D can be modified to attach the spring as follows:

$$S(\_\_\_\_, \_\_\_\_) = S(\_\_\_\_, \_\_\_\_) + \_\_\_\_$$

(Fill in the blanks.)

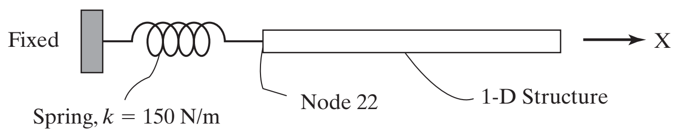


FIGURE P3.7

3.8. Consider the one-dimensional model of the structure shown in Fig. P3.8.

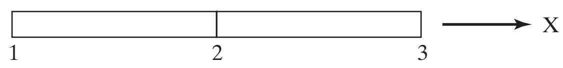


FIGURE P3.8

(a) Show that the assembled stiffness matrix **K** is singular.

(b) Give a sample displacement vector $\mathbf{Q}_0 \neq \mathbf{0}$ that satisfies $\mathbf{KQ}_0 = \mathbf{F} = \mathbf{0}$. With the help of a sketch, discuss the significance of this displacement. What is the strain energy in the structure?

(c) Prove, in general, that any nonzero solution **Q** to a system of equations $\mathbf{KQ} = \mathbf{0}$ implies that **K** is singular.

3.9. Consider the bar in Fig. P3.9 loaded as shown. Determine the nodal displacements, element stresses, and support reactions. Solve this problem by hand calculation, adopting the elimination method for handling boundary conditions. Verify your results using the program FEM1D.

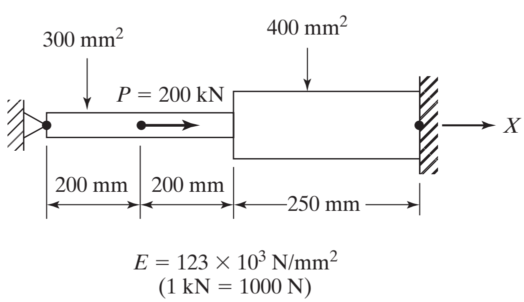


FIGURE P3.9

3.10. Repeat Example 3.5 in the text, but use the elimination approach for handling boundary conditions. Solve by hand calculation.

3.11. An axial load P = 400 kN is applied to the composite block shown in Fig. P3.11. Determine the stress in each material. (*Hint:* You may name the nodes 1–2 for both the elements.)

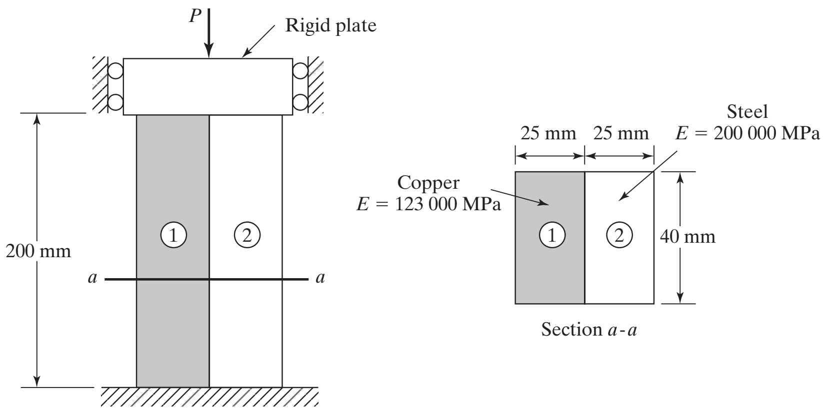


FIGURE P3.11

3.12. Consider the bar shown in Fig. P3.12. Determine the nodal displacements, element stresses, and support reactions.

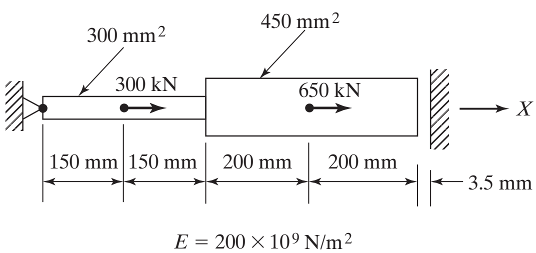


FIGURE P3.12

3.13. Complete Example 3.7 in the text using:

(a) two linear finite elements and

(b) four linear finite elements.

Plot the stress distributions on Fig. E3.7c.

3.14. A tapered rod is subjected to a body force $f = x^2$ acting in the x-direction and also a point load $P = 2$ as shown in Fig. P3.14.

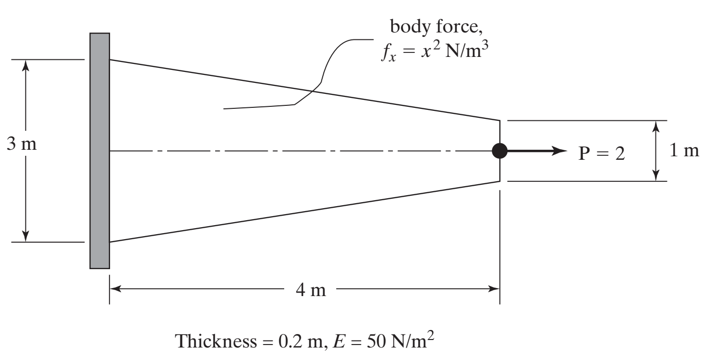


FIGURE P3.14

(a) Use the Rayleigh–Ritz method with an assumed displacement field $u = a_0 + a_1x + a_2x^2$ to determine expressions for displacement $u(x)$ and stress $\sigma(x)$.

(b) Solve this problem using a finite element solution with two (2) two-noded elements. Show all work such as element matrices, assembly, boundary conditions, and solution. Compare finite element and Rayleigh–Ritz solutions by providing plots of $u(x)$ vs. x and $\sigma(x)$ vs. x by the two methods.

3.15. Consider the multipoint constraint $3Q_p - Q_q = 0$, where p and q are the dof numbers. Indicate what modifications need to be made to the banded stiffness matrix **S** to implement this constraint. Also, if the bandwidth of the structure is n_1, what will be the new bandwidth when the constraint is introduced?

3.16. The rigid beam in Fig. P3.16 was level before the load was applied. Find the stress in each vertical member. (*Hint:* The boundary condition is of the multipoint constraint type.)

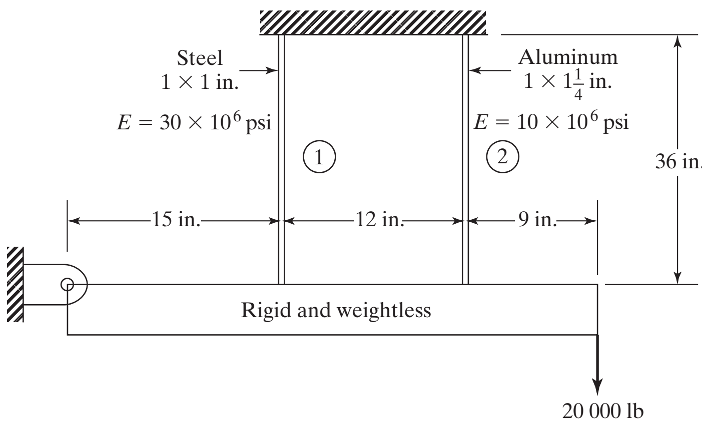


FIGURE P3.16

3.17. A copper bolt is fitted inside an steel tube, as shown in Fig. P3.17. After the nut has been fitted snugly, it is tightened one-quarter of a full turn. Given that the bolt is single threaded with a 2-mm pitch, determine the stress in bolt and tube. (*Hint:* The boundary condition *is* of the multipoint constraint type.)

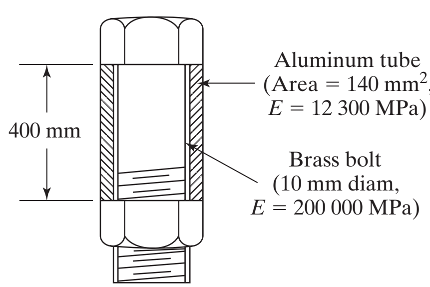


FIGURE P3.17

3.18. This problem reinforces the fact that once the shape functions are assumed, then all other element matrices can be derived. Certain arbitrary shape functions are given, and the reader is asked to derive the **B** and **k** matrices.

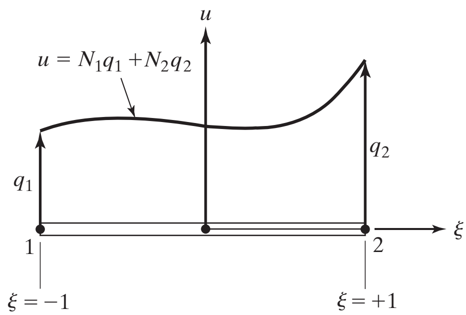


FIGURE P3.18

Consider the one-dimensional element shown in Fig. P3.18. The transformation

$$\xi = \frac{2}{x_2 - x_1}(x - x_1) - 1$$

is used to relate x- and ξ-coordinates. Let the displacement field be interpolated as

$$u(\xi) = N_1 q_1 + N_2 q_2$$

where shape functions N_1 and N_2 are assumed to be

$$N_1 = \cos\frac{\pi(1+\xi)}{4} \qquad N_2 = \cos\frac{\pi(1-\xi)}{4}$$

(a) Develop the relation $\epsilon = \mathbf{Bq}$. That is, develop the **B** matrix.

(b) Develop the stiffness matrix, $\mathbf{k}^e$. (You need not evaluate the integrals.)

3.19. Derive the element stiffness matrix **k** for the one-dimensional tapered elements shown in Fig. P3.19a and b. (*Hint:* Introduce the linearity of width for part (a) and diameter for part (b) using the shape functions used for the displacement interpolation.)

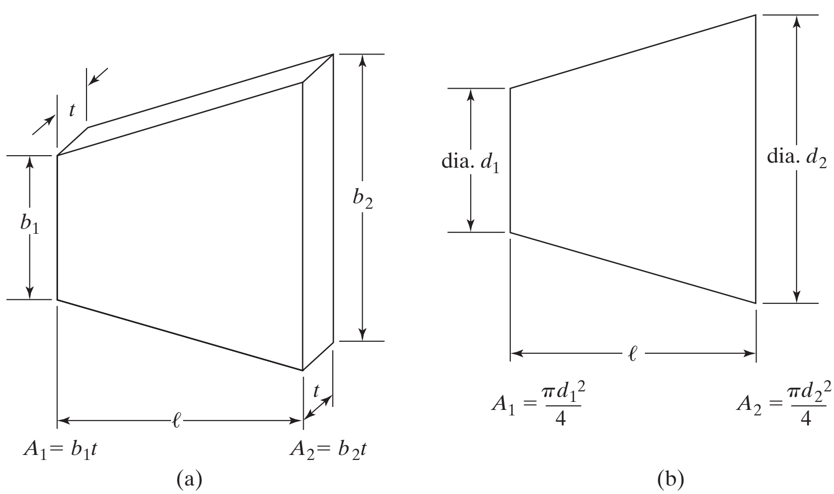


FIGURE P3.19

3.20. For plotting and extrapolation purposes (see Chapter 12), it is sometimes necessary to obtain nodal stress values from element stress values that are obtained from a computer run. Specifically, consider the element stresses, σ_1, σ_2, and σ_3, which are constant within each element, as shown in Fig. P3.20. It is desired to obtain nodal stresses S_i, $i = 1, 2, 3, 4$, which best fit the elemental values. Obtain S_i from the least-squares criterion,

$$\text{Minimize } I = \sum_e \int_e \frac{1}{2}(\sigma - \sigma_e)^2 dx$$

where σ is expressed in terms of the nodal values s_i, using linear shape functions as

$$\sigma = N_1 s_1 + N_2 s_2$$

where 1 and 2 are the local numbers.
Plot the distribution of stress from the nodal values.

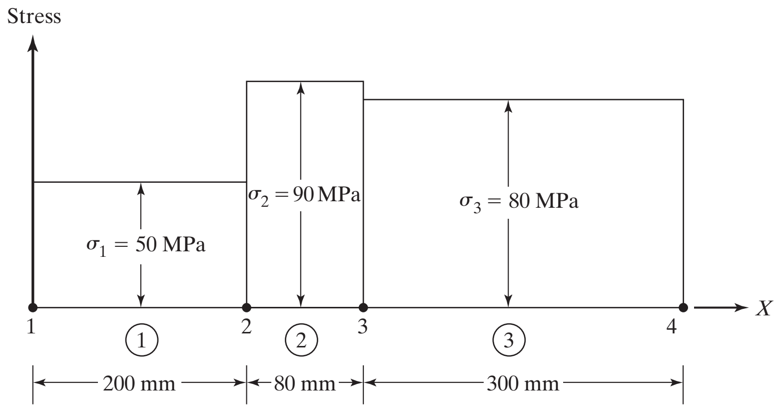


FIGURE P3.20

3.21. Determine the stresses in the 4 in. long bar in Fig. P3.21, using the following models:

(a) One linear element.

(b) Two linear elements. (Note: x in, T kips/in.)

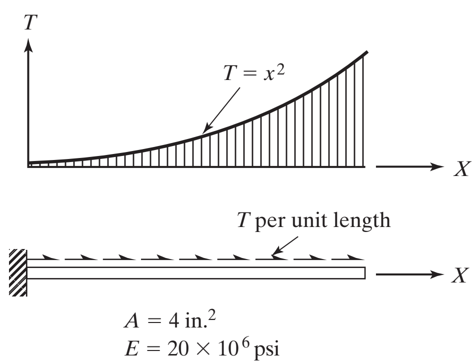


FIGURE P3.21

3.22. For the vertical rod shown in Fig. P3.22, find the deflection at A and the stress distribution. Use $E = 100$ MPa and weight per unit volume $= 0.06$ N/cm^3. (*Hint:* Introduce weight contribution to the nodal loads into the program and solve using two elements and four elements.) Comment on the stress distribution.

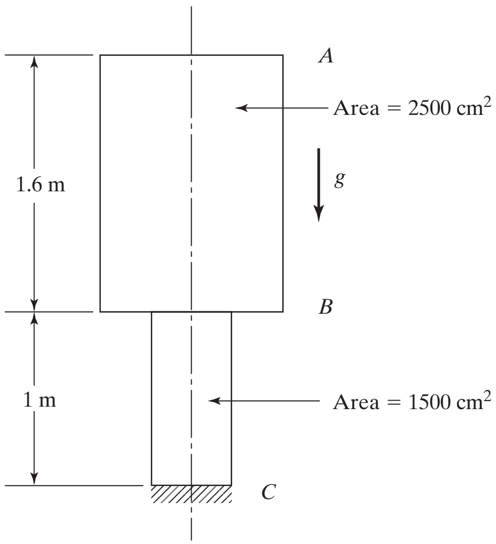


FIGURE P3.22

3.23. For Fig. P3.23, find the deflection at the free end under its own weight, using divisions of

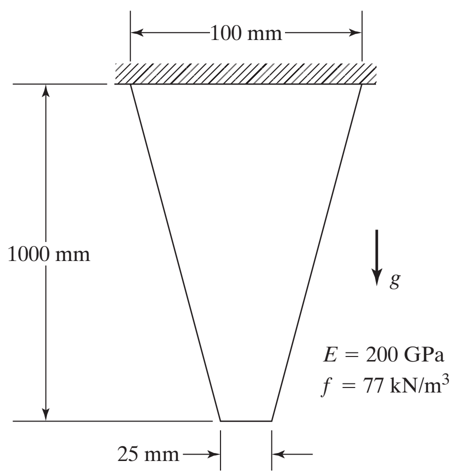


FIGURE P3.23

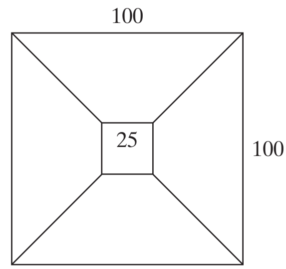


FIGURE P3.23 (continued)

(a) 1 element,

(b) 2 elements,

(c) 4 elements,

(d) 8 elements, and

(e) 16 elements.

Then plot number of elements vs. deflection.

3.24. A two-element finite element solution for a rod in Fig. P3.24 has been obtained as $\mathbf{Q} = [0 \quad 0.5 \quad 0.25]^T$ mm. If the element shape functions used were $N_1 = \dfrac{[1 - \xi]^2}{4}$, $N_2 = \dfrac{[1 + \xi]^2}{4}$, evaluate the displacement u at the midpoint of element 1–2.

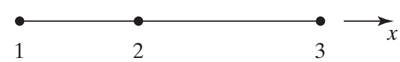


FIGURE P3.24

3.25. For the rod shown in Fig. P3.25, the boundary conditions are $Q_1 = 0, Q_3 = 0.2$. Coordinates of nodes are: $x_1 = 0, x_2 = 1, x_3 = 3$. A point load is applied as shown. Take $E = 1, A = 1$. Obtain the displacement Q_2 using 2 two-noded finite elements.

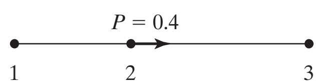


FIGURE P3.25

3.26. The rod shown in Fig. P3.26 is subjected to a body force $f_x = 1$ N/m^3. Take $E = 1$ N/m^2, $A = 1$ m^2 and $L = 3$ m $=$ length of rod. Point load applied is $P = 1$ N, as shown.

(a) Assuming a displacement field as $u = a_0 + a_1 x$, use the Rayleigh–Ritz method to find the displacement and stress distribution, that is, find $u(x)$ and $\sigma(x)$.

(b) Solve the same problem using a two-element finite element model (each element is two-noded).

(c) Plot $u(x)$ and $\sigma(x)$ by both Rayleigh–Ritz and finite element approaches.

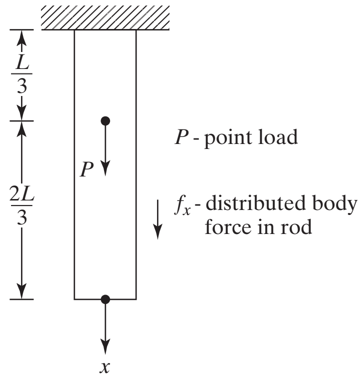


FIGURE P3.26

3.27. Consider the following shape functions (for the one-dimensional problem with two-noded elements): $N_1 = \frac{[1-\xi]^2}{4}, N_2 = \frac{[1+\xi]^2}{4}$

(a) Derive the **B** matrix.

(b) Determine the strain ϵ_x in the element due to end displacements $q_1 = q_2 = 1$. Explain why / why not the strain value you have obtained makes sense.

3.28. The rod shown in Fig. P3.28 is subjected to a set of eight point loads as shown. Each point load $P = 1$ in magnitude.

(a) Assuming a simple displacement field as $u = a_0 + a_1 x$, use the Rayleigh–Ritz method to find the displacement at the tip where $x = 8$.

(b) Compare above solution with FEA using the program FEM1D (include listing of input and output files).

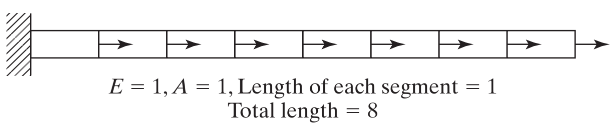


FIGURE P3.28

3.29. The one-dimensional rod shown in Fig. P3.29 is subjected to a linearly varying body force $f = x$. Assume a displacement field as $u = a_0 + a_1 x + a_2 x^2$, and use the Rayleigh–Ritz method to find the displacement field $u(x)$ and stress field $\sigma(x)$. Hence, compute the displacement at the tip, and plot $\sigma(x)$ vs. x. Take $E = 100$, $A = 1$, and $L = 1$.

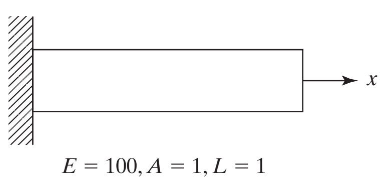


FIGURE P3.29

3.30. Consider the quadratic element shown in Fig. P3.30, subjected to a quadratically varying traction force (which is defined as force per unit length).

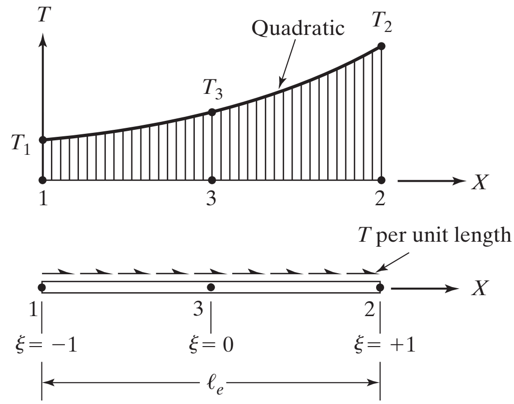


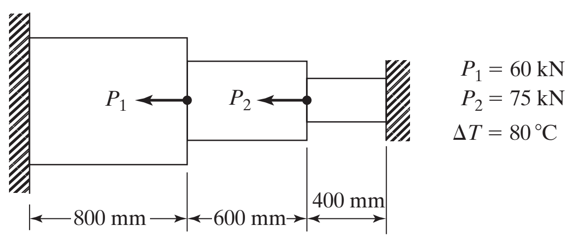
FIGURE P3.30

(a) Express the traction force as a function of ξ, T_1, T_2, and T_3, using the shape functions N_1, N_2, and N_3.

(b) Derive, from the potential term $\int_e u^{\mathrm{T}} T\,dx$, an expression for the element traction force, $\mathbf{T}^e$. Leave your answer in terms of T_1, T_2, T_3, and ℓ_e.

(c) Resolve Problem 3.21 using the exact traction load derived previously, with one quadratic element, by hand calculations.

3.31. The structure in Fig. P3.31 is subjected to an increase in temperature, $\Delta T = 80°\mathrm{C}$. Determine the displacements, stresses, and support reactions. Solve this problem by hand calculation, using the elimination method for handling boundary conditions.



	Bronze	Aluminum	Steel
A =	2400 mm^2	1200 mm^2	600 mm^2
E =	83 GPa	70 GPa	200 GPa
α =	$18.9 \times 10^{-6}/°\mathrm{C}$	$23 \times 10^{-6}/°\mathrm{C}$	$11.7 \times 10^{-6}/°\mathrm{C}$

($1\ \mathrm{GPa} = 10^9\ \mathrm{N/m^2}$)

FIGURE P3.31

3.32. Consider the column shown in Fig. P3.32 acted upon by its own weight.

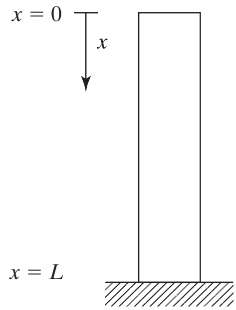


FIGURE P3.32

Take weight density, $\rho g \equiv \gamma = 1\ \text{N/m}^3$. The rod has a uniform cross-section with cross-sectional area $A = 1\ \text{m}^2$, $E = x^{1.2}$ Pa, and $L = 3\text{m}$.

(a) Solve by hand calculations using NE = 2 elements.

(b) Verify the solution for (a) using FEM1D MATLAB code (note that BANSOL.M is used by FEM1D, and they both should be in the same directory).

(c) Modifying the program FEM1D, solve using NE $= 2, 4, 8, 16$ elements and plot tip (that is, at $x = 0$) displacement vs. NE. Include a listing of the code modifications.

(d) Also plot potential energy at equilibrium vs. NE.

3.33. Assemble the 3×1 thermal load contributions to the load vector **F** for the configuration shown in Fig. P3.33.

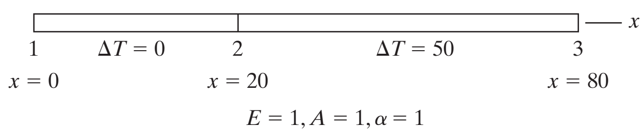


FIGURE P3.33

3.34. A tapered plate of uniform thickness of 1 m shown in Fig. P3.34 is subjected to a body force of $f_x = x$ acting in the x direction. Solve the problem using two elements. Plot element division and stress distribution.

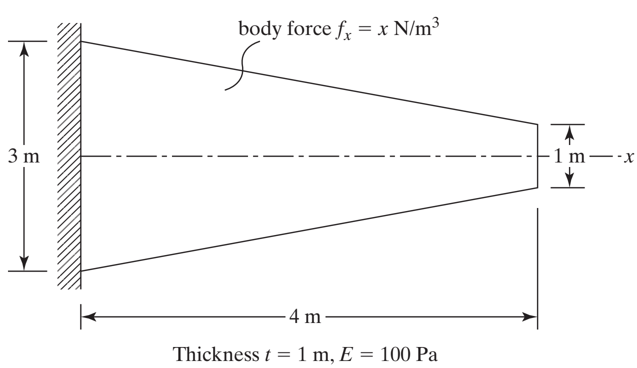


FIGURE P3.34

3.35. Modify the program FEM1D to calculate the element strain energy using $U_e = \frac{1}{2}\int_e \sigma \epsilon dV$. Use AREA (N) to define the cross-sectional area of element N. Modify the following part of the program taken from fem1d.m.

```
%----- Stress Calculation -----
for N = 1 : NE
   N1 = NOC(N, 1); N2 = NOC(N, 2); N3 = MAT(N)
   EPS = (F(N2) - F(N1)) / (X(N2) - X(N1));
   If NPR > 1 Then C = PM(N3, 2)
   Stress(N) = PM(N3, 1) * (EPS - C * DT(N));
End
```

3.36. Express the integral $\int_e u^2 dx$ in the form $\mathbf{q}^T \mathbf{W} \mathbf{q}$, and thus give an expression for the matrix $\mathbf{W}$. (*Hint:* Substitute $u = \mathbf{N}\mathbf{q}$ and write dx in terms of $d\xi$. Denote ℓ_e = element length.)

Program Listing

```
MAIN PROGRAM
'****************************************
'*          PROGRAM FEM1D               *
'*         1-D BAR ELEMENT              *
'*    WITH MULTI-POINT CONSTRAINTS      *
'* T.R.Chandrupatla and A.D.Belegundu   *
'****************************************
DefInt I-N
Dim NN, NE, NM, NDIM, NEN, NDN
Dim ND, NL, NPR, NMPC, NBW
Dim X(), NOC(), F(), AREA(), MAT(), DT(), S()
Dim PM(), NU(), U(), MPC(), BT(), Stress(), React()
Dim CNST, Title$
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call Stiffness
     Call ModifyForBC
     Call BandSolver
     Call StressCalc
     Call ReactionCalc
     Call Output
End Sub
```

DATA INPUT FROM SHEET1 in Excel (from file in C, FORTRAN, MATLAB etc)

```
Private Sub InputData()
    NN = Worksheets(1).Cells(4, 1)
    NE = Worksheets(1).Cells(4, 2)
    NM = Worksheets(1).Cells(4, 3)
    NDIM = Worksheets(1).Cells(4, 4)
    NEN = Worksheets(1).Cells(4, 5)
    NDN = Worksheets(1).Cells(4, 6)
    ND = Worksheets(1).Cells(6, 1)
    NL = Worksheets(1).Cells(6, 2)
    NMPC = Worksheets(1).Cells(6, 3)
    NPR = 2  'Material Properties E, Alpha
    ReDim X(NN), NOC(NE, NEN), F(NN), AREA(NE), MAT(NE), DT(NE)
    ReDim PM(NM, NPR), NU(ND), U(ND), MPC(NMPC, 2), BT(NMPC, 3)
      LI = 7
     '----- Coordinates -----
    For I = 1 To NN
        LI = LI + 1
        N = Worksheets(1).Cells(LI, 1)
        X(N) = Worksheets(1).Cells(LI, 2)
    Next
     LI = LI + 1
    '----- Connectivity -----
     For I = 1 To NE
        LI = LI + 1
        N = Worksheets(1).Cells(LI, 1)
        NOC(N, 1) = Worksheets(1).Cells(LI, 2)
        NOC(N, 2) = Worksheets(1).Cells(LI, 3)
        MAT(N) = Worksheets(1).Cells(LI, 4)
        AREA(N) = Worksheets(1).Cells(LI, 5)
        DT(N) = Worksheets(1).Cells(LI, 6)
     Next
     '----- Specified Displacements -----
     LI = LI + 1
     For I = 1 To ND
        LI = LI + 1
        NU(I) = Worksheets(1).Cells(LI, 1)
        U(I) = Worksheets(1).Cells(LI, 2)
     Next
     '----- Component Loads -----
     LI = LI + 1
     For I = 1 To NL
        LI = LI + 1
        N = Worksheets(1).Cells(LI, 1)
        F(N) = Worksheets(1).Cells(LI, 2)
     Next
     LI = LI + 1
     '----- Material Properties -----
```

```
     For I = 1 To NM
        LI = LI + 1
        N = Worksheets(1).Cells(LI, 1)
        For J = 1 To NPR
            PM(N, J) = Worksheets(1).Cells(LI, J + 1)
        Next
     Next
     '----- Multi-point Constraints B1*Qi+B2*Qj=B0
     If NMPC > 0 Then
         LI = LI + 1
        For I = 1 To NMPC
            LI = LI + 1
            BT(I, 1) = Worksheets(1).Cells(LI, 1)
            MPC(I, 1) = Worksheets(1).Cells(LI, 2)
            BT(I, 2) = Worksheets(1).Cells(LI, 3)
            MPC(I, 2) = Worksheets(1).Cells(LI, 4)
            BT(I, 3) = Worksheets(1).Cells(LI, 5)
        Next
     End If
End Sub
```

BANDWIDTH EVALUATION

```
Private Sub Bandwidth()
     '----- Bandwidth Evaluation -----
     NBW = 0
     For N = 1 To NE
        NABS = Abs(NOC(N, 1) - NOC(N, 2)) + 1
        If NBW < NABS Then NBW = NABS
     Next N
     For I = 1 To NMPC
        NABS = Abs(MPC(I, 1) - MPC(I, 2)) + 1
        If NBW < NABS Then NBW = NABS
     Next I
End Sub
```

ELEMENT STIFFNESS AND ASSEMBLY

```
Private Sub Stiffness()
     ReDim S(NN, NBW)
     '----- Stiffness Matrix -----
     For N = 1 To NE
        N1 = NOC(N, 1): N2 = NOC(N, 2): N3 = MAT(N)
        X21 = X(N2) - X(N1): EL = Abs(X21)
        EAL = PM(N3, 1) * AREA(N) / EL
        TL = PM(N3, 1) * PM(N3, 2) * DT(N) * AREA(N) * EL / X21
        '----- Temperature Loads -----
```

continued

```
        F(N1) = F(N1) - TL
        F(N2) = F(N2) + TL
        '----- Element Stiffness in Global Locations -----
        S(N1, 1) = S(N1, 1) + EAL
        S(N2, 1) = S(N2, 1) + EAL
        IR = N1: If IR > N2 Then IR = N2
        IC = Abs(N2 - N1) + 1
        S(IR, IC) = S(IR, IC) - EAL
     Next N
End Sub
```

```
MODIFICATION FOR BOUNDARY CONDITIONS
Private Sub ModifyForBC()
     '----- Decide Penalty Parameter CNST -----
     CNST = 0
     For I = 1 To NN
        If CNST > S(I, 1) Then CNST = S(I, 1)
     Next I
     CNST = CNST * 10000
     '----- Modify for Boundary Conditions -----
        '--- Displacement BC ---
        For I = 1 To ND
           N = NU(I)
           S(N, 1) = S(N, 1) + CNST
           F(N) = F(N) + CNST * U(I)
        Next I
        '--- Multi-point Constraints ---
        For I = 1 To NMPC
           I1 = MPC(I, 1): I2 = MPC(I, 2)
           S(I1, 1) = S(I1, 1) + CNST * BT(I, 1) * BT(I, 1)
           S(I2, 1) = S(I2, 1) + CNST * BT(I, 2) * BT(I, 2)
           IR = I1: If IR > I2 Then IR = I2
           IC = Abs(I2 - I1) + 1
           S(IR, IC) = S(IR, IC) + CNST * BT(I, 1) * BT(I, 2)
           F(I1) = F(I1) + CNST * BT(I, 1) * BT(I, 3)
           F(I2) = F(I2) + CNST * BT(I, 2) * BT(I, 3)
        Next I
End Sub
```

```
SOLUTION OF EQUATIONS BANDSOLVER
Private Sub BandSolver()
     '----- Equation Solving using Band Solver -----
          N = NN
     '----- Forward Elimination -----
     For K = 1 To N - 1
        NBK = N - K + 1
        If N - K + 1 > NBW Then NBK = NBW
        For I = K + 1 To NBK + K - 1
           I1 = I - K + 1
           C = S(K, I1) / S(K, 1)
           For J = I To NBK + K - 1
              J1 = J - I + 1
              J2 = J - K + 1
              S(I, J1) = S(I, J1) - C * S(K, J2)
           Next J
           F(I) = F(I) - C * F(K)
        Next I
     Next K
     '----- Back Substitution -----
     F(N) = F(N) / S(N, 1)
     For II = 1 To N - 1
        I = N - II
        NBI = N - I + 1
        If N - I + 1 > NBW Then NBI = NBW
        Sum = 0!
        For J = 2 To NBI
           Sum = Sum + S(I, J) * F(I + J - 1)
        Next J
        F(I) = (F(I) - Sum) / S(I, 1)
     Next II
End Sub
```

```
STRESS CALCULATIONS
Private Sub StressCalc()
     ReDim Stress(NE)
     '----- Stress Calculation -----
     For N = 1 To NE
        N1 = NOC(N, 1): N2 = NOC(N, 2): N3 = MAT(N)
        EPS = (F(N2) - F(N1)) / (X(N2) - X(N1))
        Stress(N) = PM(N3, 1) * (EPS - PM(N3, 2) * DT(N))
     Next N
End Sub
```

```
REACTION CALCULATIONS
Private Sub ReactionCalc()
     ReDim React(ND)
     '----- Reaction Calculation -----
     For I = 1 To ND
        N = NU(I)
        React(I) = CNST * (U(I) - F(N))
     Next I
End Sub
```

```
OUTPUT TO SHEET2 in Excel (to file in C,FORTRAN, MATLAB etc)
Private Sub Output()
   ' Now, writing out the results in a different worksheet
    Worksheets(2).Cells.ClearContents
    Worksheets(2).Cells(1, 1) = "Results from Program FEM1D"
    Worksheets(2).Cells(1, 1).Font.Bold = True
    Worksheets(2).Cells(2, 1) = Worksheets(1).Cells(2, 1)
    Worksheets(2).Cells(2, 1).Font.Bold = True
    Worksheets(2).Cells(3, 1) = "Node#"
    Worksheets(2).Cells(3, 1).Font.Bold = True
    Worksheets(2).Cells(3, 2) = "Displacement"
    Worksheets(2).Cells(3, 2).Font.Bold = True
    LI = 3
    For I = 1 To NN
        LI = LI + 1
        Worksheets(2).Cells(LI, 1) = I
        Worksheets(2).Cells(LI, 2) = F(I)
    Next
    LI = LI + 1
    Worksheets(2).Cells(LI, 1) = "Element#"
    Worksheets(2).Cells(LI, 1).Font.Bold = True
    Worksheets(2).Cells(LI, 2) = "Stress"
    Worksheets(2).Cells(LI, 2).Font.Bold = True
    For N = 1 To NE
        LI = LI + 1
        Worksheets(2).Cells(LI, 1) = N
        Worksheets(2).Cells(LI, 2) = Stress(N)
    Next
    LI = LI + 1
    Worksheets(2).Cells(LI, 1) = "Node#"
    Worksheets(2).Cells(LI, 1).Font.Bold = True
    Worksheets(2).Cells(LI, 2) = "Reaction"
    Worksheets(2).Cells(LI, 2).Font.Bold = True
    For I = 1 To ND
        LI = LI + 1
        N = NU(I)
        Worksheets(2).Cells(LI, 1) = N
        Worksheets(2).Cells(LI, 2) = React(I)
     Next
End Sub
```

CHAPTER 4

Trusses

4.1 INTRODUCTION

The finite element analysis of truss structures is presented in this chapter. Two-dimensional (2-D) trusses (or plane trusses) are treated in Section 4.2. In Section 4.3, this treatment is readily generalized to handle three-dimensional (3-D) trusses. A typical plane truss is shown in Fig. 4.1. A truss structure consists only of two-force members. That is, every truss element is in direct tension or compression (Fig. 4.2). In a truss, it is required that all loads and reactions are applied only at the joints and that all members are connected together at their ends by frictionless pin joints. Every engineering student has, in a course on statics, analyzed trusses using the method of joints and the method of sections. These methods, while illustrating the fundamentals of statics, become tedious when applied to large-scale statically indeterminate truss structures. Further, joint displacements are not readily obtainable. The finite element method, on the other hand, is applicable to statically determinate or indeterminate structures alike. This method also provides joint deflections. Effects of temperature changes and support settlements can also be routinely handled.

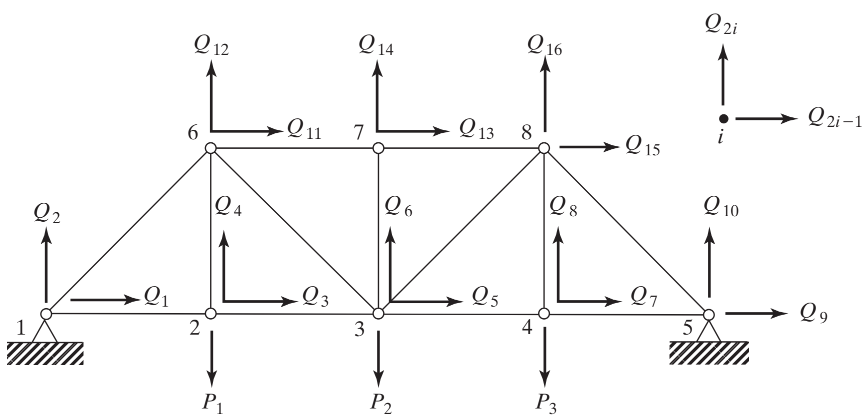


FIGURE 4.1 A two-dimensional truss.

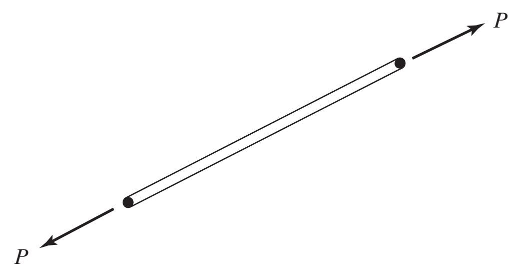


FIGURE 4.2 A two-force member.

4.2 PLANE TRUSSES

Modeling aspects discussed in Chapter 3 are now extended to the 2-D truss. The steps involved are discussed here.

Local and Global Coordinate Systems

The main difference between the one-dimensional (1-D) structures considered in Chapter 3 and trusses is that the elements of a truss have various orientations. To account for these different orientations, **local** and **global** coordinate systems are introduced as follows:

A typical plane-truss element is shown in local and global coordinate systems in Fig. 4.3. In the local numbering scheme, the two nodes of the element are numbered 1 and 2. The local coordinate system consists of the x'-axis, which runs along the element from node 1 toward node 2. All quantities in the local coordinate system will be denoted by a prime ($'$). The global x-, y-coordinate system is fixed and does not depend

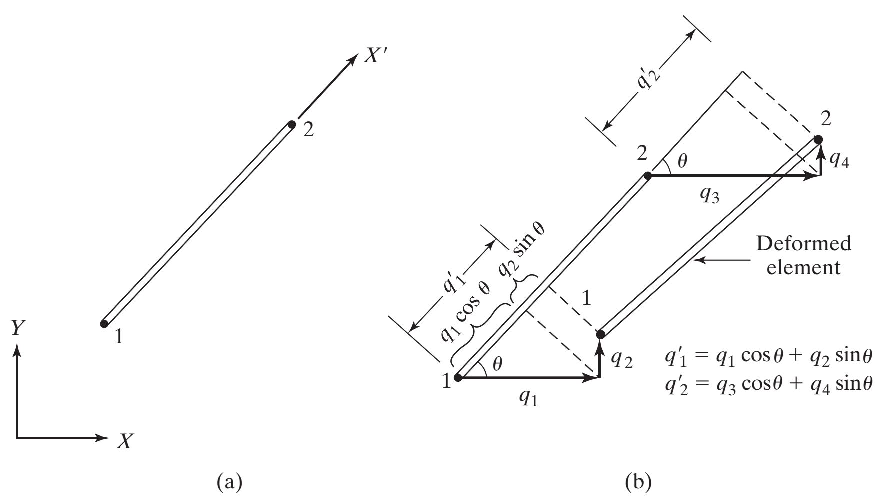


FIGURE 4.3 A two-dimensional truss element in (a) a local coordinate system and (b) a global coordinate system.

on the orientation of the element. Note that x, y, and z form a right-handed coordinate system with the z-axis coming straight out of the paper. In the global coordinate system, every node has two degrees of freedom (dof). A systematic numbering scheme is adopted here: A node whose global node number is j has associated with it dof $2j - 1$ and $2j$. Further, the global displacements associated with node j are Q_{2j-1} and Q_{2j}, as shown in Fig. 4.1.

Let q'_1 and q'_2 be the displacements of nodes 1 and 2, respectively, in the local coordinate system. Thus, the element displacement vector in the local coordinate system is denoted by

$$\mathbf{q}' = [q'_1, q'_2]^{\mathrm{T}} \tag{4.1}$$

The element displacement vector in the global coordinate system is a (4×1) vector denoted by

$$\mathbf{q} = [q_1, q_2, q_3, q_4]^{\mathrm{T}} \tag{4.2}$$

The relationship between $\mathbf{q}'$ and $\mathbf{q}$ is developed as follows: In Fig. 4.3b, we see that q'_1 equals the sum of the projections of q_1 and q_2 onto the x'-axis. Thus,

$$q'_1 = q_1 \cos\theta + q_2 \sin\theta \tag{4.3a}$$

Similarly,

$$q'_2 = q_3 \cos\theta + q_4 \sin\theta \tag{4.3b}$$

At this stage, the **direction cosines** ℓ and m are introduced as $\ell = \cos\theta$ and $m = \cos\phi\,(=\sin\theta)$. These direction cosines are the cosines of the angles that the local x'-axis makes with the global x- and y-axes, respectively. Equations 4.3a and 4.3b can now be written in matrix form as

$$\mathbf{q}' = \mathbf{L}\mathbf{q} \tag{4.4}$$

where the **transformation matrix L** is given by

$$\mathbf{L} = \begin{bmatrix} \ell & m & 0 & 0 \\ 0 & 0 & \ell & m \end{bmatrix} \tag{4.5}$$

Formulas for Calculating ℓ and m

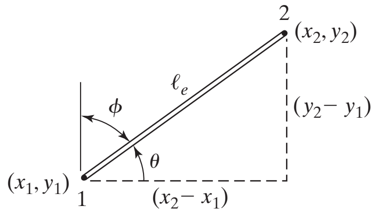
Simple formulas are now given for calculating the direction cosines ℓ and m from nodal coordinate data. Referring to Fig. 4.4, let (x_1, y_1) and (x_2, y_2) be the coordinates of nodes 1 and 2, respectively. We then have

$$\ell = \frac{x_2 - x_1}{\ell_e} \qquad m = \frac{y_2 - y_1}{\ell_e} \tag{4.6}$$

where the length ℓ_e is obtained from

$$\ell_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.7}$$

The expressions in Eqs. 4.6 and 4.7 are obtained from nodal coordinate data and can readily be implemented in a computer program.



$$\ell = \cos\theta = \frac{x_2 - x_1}{\ell_e}$$

$$m = \cos\phi = \frac{y_2 - y_1}{\ell_e} (= \sin\theta)$$

$$\ell_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

FIGURE 4.4 Direction cosines.

Element Stiffness Matrix

An important observation will now be made: *The truss element is a 1-D element when viewed in the local coordinate system.* This observation allows us to use previously developed results in Chapter 3 for 1-D elements. Consequently, from Eq. 3.29, the element stiffness matrix for a truss element in the local coordinate system is given by

$$\mathbf{k} = \frac{E_e A_e}{\ell_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{4.8}$$

where A_e is the element cross-sectional area and E_e is Young's modulus. The problem at hand is to develop an expression for the element stiffness matrix in the global coordinate system. This is obtainable by considering the strain energy in the element. Specifically, the element strain energy in local coordinates is given by

$$U_e = \tfrac{1}{2}\mathbf{q}'^{\mathrm{T}}\mathbf{k}'\mathbf{q}' \tag{4.9}$$

Substituting for $\mathbf{q}' = \mathbf{L}\mathbf{q}$ into Eq. 4.9, we get

$$U_e = \tfrac{1}{2}\mathbf{q}^{\mathrm{T}}[\mathbf{L}^{\mathrm{T}}\mathbf{k}'\mathbf{L}]\mathbf{q} \tag{4.10}$$

The strain energy in global coordinates can be written as

$$U_e = \tfrac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}\mathbf{q} \tag{4.11}$$

where $\mathbf{k}$ is the element stiffness matrix in global coordinates. From the previous equation, we obtain the element stiffness matrix in global coordinates as

$$\mathbf{k} = \mathbf{L}^{\mathrm{T}}\mathbf{k}'\mathbf{L} \tag{4.12}$$

Substituting for $\mathbf{L}$ from Eq. 4.5 and for $\mathbf{k}'$ from Eq. 4.8, we get

$$\mathbf{k} = \frac{E_e A_e}{\ell_e}\begin{bmatrix} \ell^2 & \ell m & -\ell^2 & -\ell m \\ \ell m & m^2 & -\ell m & -m^2 \\ -\ell^2 & -\ell m & \ell^2 & \ell m \\ -\ell m & -m^2 & \ell m & m^2 \end{bmatrix} \tag{4.13}$$

The element stiffness matrices are assembled in the usual manner to obtain the structural stiffness matrix. This assembly is illustrated in Example 4.1. The computer logic for directly placing element stiffness matrices into global matrices for banded and skyline solutions is explained in Section 4.4.

The derivation of the result $\mathbf{k} = \mathbf{L}^{\mathrm{T}}\mathbf{k}'\mathbf{L}$ also follows from Galerkin's variational principle. The virtual work δW as a result of virtual displacement $\boldsymbol{\Psi}'$ is

$$\delta W = \boldsymbol{\Psi}'^{\mathrm{T}}(\mathbf{k}'\mathbf{q}') \tag{4.14a}$$

Since $\boldsymbol{\Psi}' = \mathbf{L}\boldsymbol{\Psi}$ and $\mathbf{q}' = \mathbf{L}\mathbf{q}$, we have

$$\begin{aligned} \delta W &= \boldsymbol{\Psi}^{\mathrm{T}}[\mathbf{L}^{\mathrm{T}}\mathbf{k}'\mathbf{L}]\mathbf{q} \\ &= \boldsymbol{\Psi}^{\mathrm{T}}\mathbf{k}\mathbf{q} \end{aligned} \tag{4.14b}$$

Stress Calculations

Expressions for the element stresses can be obtained by noting that a truss element in local coordinates is a simple two-force member (Fig. 4.2). Thus, the stress σ in a truss element is given by

$$\sigma = E_e \epsilon \tag{4.15a}$$

Since the strain ϵ is the change in length per unit original length,

$$\begin{aligned} \sigma &= E_e \frac{q'_2 - q'_1}{\ell_e} \\ &= \frac{E_e}{\ell_e}[-1 \quad 1]\begin{Bmatrix} q'_1 \\ q'_2 \end{Bmatrix} \end{aligned} \tag{4.15b}$$

This equation can be written in terms of the global displacements $\mathbf{q}$ using the transformation $\mathbf{q}' = \mathbf{L}\mathbf{q}$ as

$$\sigma = \frac{E_e}{\ell_e}[-1 \quad 1]\mathbf{L}\mathbf{q} \tag{4.15c}$$

Substituting for $\mathbf{L}$ from Eq. 4.5 yields

$$\sigma = \frac{E_e}{\ell_e}[-\ell \quad -m \quad \ell \quad m]\mathbf{q} \tag{4.16}$$

Once the displacements are determined by solving the finite element equations, the stresses can be recovered from Eq. 4.16 for each element. Note that a positive stress implies that the element is in tension and a negative stress implies compression.

Example 4.1

Consider the four-bar truss shown in Fig. E4.1a. It is given that $E = 29.5 \times 10^6$ psi and $A_e = 1 \text{ in.}^2$ for all elements. Complete the following:

(a) Determine the element stiffness matrix for each element.
(b) Assemble the structural stiffness matrix $\mathbf{K}$ for the entire truss.

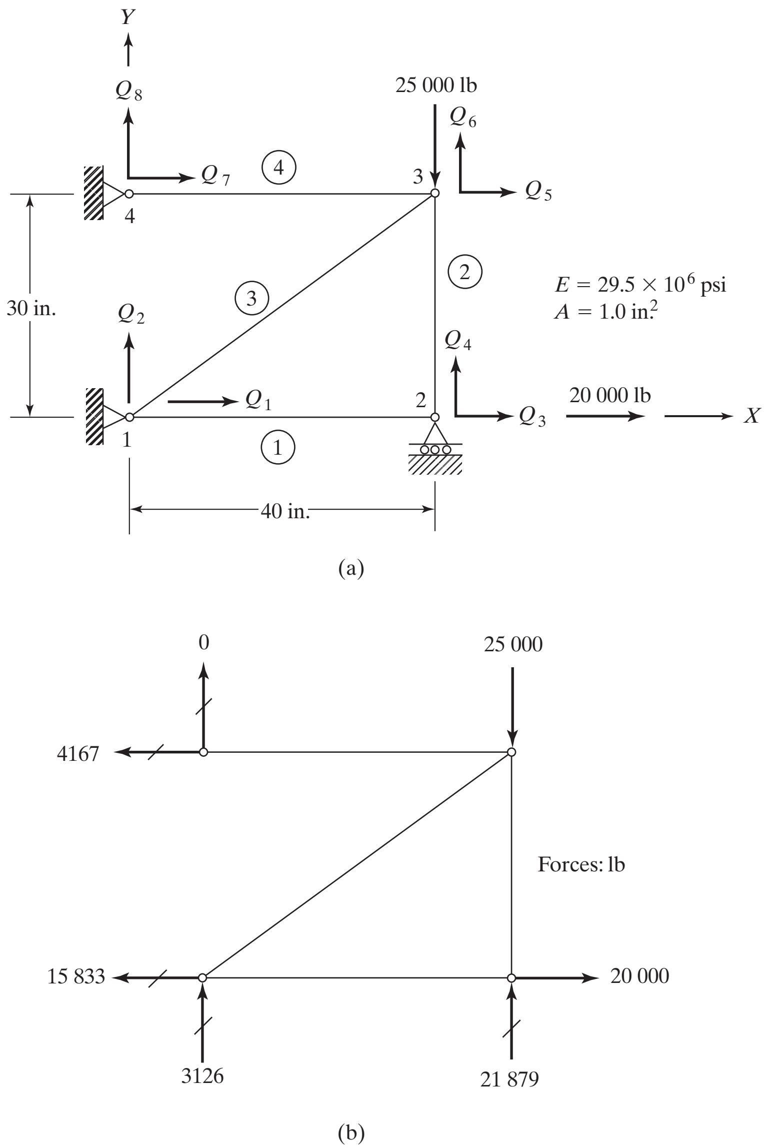


FIGURE E4.1

(c) Using the elimination approach, solve for the nodal displacement.
(d) Recover the stresses in each element.
(e) Calculate the reaction forces.

Solution

(a) It is recommended that a *tabular* form be used for representing nodal coordinate data and element information. The nodal coordinate data are as follows:

Node	x	y
1	0	0
2	40	0
3	40	30
4	0	30

The element connectivity table is

Element	1	2
1	1	2
2	3	2
3	1	3
4	4	3

Note that the user has a choice in defining element connectivity. For example, the connectivity of element 2 can be defined as 2–3 instead of 3–2 as in the previous table. However, calculations of the direction cosines will be consistent with the adopted connectivity scheme. Using formulas in Eqs. 4.6 and 4.7, together with the nodal coordinate data and the given element connectivity information, we obtain the direction cosines table:

Element	ℓ_e	ℓ	m
1	40	1	0
2	30	0	−1
3	50	0.8	0.6
4	40	1	0

For example, the direction cosines of elements 3 are obtained as $\ell = (x_3 - x_1)/\ell_e = (40 - 0)/50 = 0.8$ and $m = (y_3 - y_1)/\ell_e = (30 - 0)/50 = 0.6$. Now, using Eq. 4.13, the element stiffness matrices for element 1 can be written as

$$\mathbf{k}^1 = \frac{29.5 \times 10^6}{40} \begin{array}{c} \begin{matrix} 1 & 3 & 3 & 4 \end{matrix} \leftarrow \text{Global dof} \\ \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \end{array}$$

The global dof associated with element 1, which is connected between nodes 1 and 2, are indicated in $\mathbf{k}^1$ earlier. These global dof are shown in Fig. E4.1a and assist in assembling the various element stiffness matrices.

The element stiffness matrices of elements 2, 3, and 4 are as follows:

$$\mathbf{k}^2 = \frac{29.5 \times 10^6}{30} \begin{array}{c} \begin{matrix} 5 & 6 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{matrix} 5 \\ 6 \\ 3 \\ 4 \end{matrix} \end{array}$$

$$\mathbf{k}^3 = \frac{29.5 \times 10^6}{50} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 5 & 6 \end{array} \\ \begin{bmatrix} .64 & .48 & -.64 & -.48 \\ .48 & .36 & -.48 & -.36 \\ -.64 & -.48 & .64 & .48 \\ .48 & -.36 & .48 & .36 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 5 \\ 6 \end{array}$$

$$\mathbf{k}^4 = \frac{29.5 \times 10^6}{40} \begin{array}{c} \begin{array}{cccc} 7 & 8 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ 7 \\ 8 \\ 5 \\ 6 \end{array}$$

(b) The structural stiffness matrix **K** is now assembled from the element stiffness matrices. By adding the element stiffness contributions, noting the element connectivity, we get

$$\mathbf{K} = \frac{29.5 \times 10^6}{600} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{bmatrix} 22.68 & 5.76 & -15.0 & 0 & -7.68 & -5.76 & 0 & 0 \\ 5.76 & 4.32 & 0 & 0 & -5.76 & -4.32 & 0 & 0 \\ -15.0 & 0 & 15.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20.0 & 0 & -20.0 & 0 & 0 \\ -7.68 & -5.76 & 0 & 0 & 22.68 & 5.76 & -15.0 & 0 \\ -5.76 & -4.32 & 0 & -20.0 & 5.76 & 24.32 & 0 & 0 \\ 0 & 0 & 0 & 0 & -15.0 & 0 & 15.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}$$

(c) The structural stiffness matrix **K** given above needs to be modified to account for the boundary conditions. The elimination approach discussed in Chapter 3 will be used here. The rows and columns corresponding to dof 1, 2, 4, 7, and 8, which correspond to fixed supports, are deleted from the **K** matrix. The reduced finite element equations are given as

$$\frac{29.5 \times 10^6}{600} \begin{bmatrix} 15 & 0 & 0 \\ 0 & 22.68 & 5.76 \\ 0 & 5.76 & 24.32 \end{bmatrix} \begin{Bmatrix} Q_3 \\ Q_5 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} 20000 \\ 0 \\ -25000 \end{Bmatrix}$$

Solution of these equations yields the displacements

$$\begin{Bmatrix} Q_3 \\ Q_5 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} 27.12 \times 10^{-3} \\ 5.65 \times 10^{-3} \\ -22.25 \times 10^{-3} \end{Bmatrix} \text{in.}$$

The nodal displacement vector for the entire structure can therefore be written as

$$\mathbf{Q} = [0, 0, 27.12 \times 10^{-3}, 0, 5.65 \times 10^{-3}, 22.25 \times 10^{-3}, 0, 0]^{\mathrm{T}} \text{in.}$$

(d) The stress in each element can now be determined from Eq. 4.16, as shown below.

The connectivity of element 1 is $1 - 2$. Consequently, the nodal displacement vector for element 1 is given by $\mathbf{q} = [0, 0, 27.12 \times 10^{-3}, 0]^{\mathrm{T}}$, and Eq. 4.16 yields

$$\sigma_1 = \frac{29.5 \times 10^6}{40}[-1 \;\; 0 \;\; 1 \;\; 0]\begin{Bmatrix} 0 \\ 0 \\ 27.12 \times 10^{-3} \\ 0 \end{Bmatrix}$$

$$= 20\,000.0 \text{ psi}$$

The stress in member 2 is given by

$$\sigma_2 = \frac{29.5 \times 10^6}{30}[0 \;\; 1 \;\; 0 \;\; -1]\begin{Bmatrix} 5.65 \times 10^{-3} \\ -22.25 \times 10^{-3} \\ +27.12 \times 10^{-3} \\ 0 \end{Bmatrix}$$

$$= -21\,880.0 \text{ psi}$$

Following similar steps, we get

$$\sigma_3 = -5208.0 \text{ psi}$$
$$\sigma_4 = 4167.0 \text{ psi}$$

(e) The final step is to determine the support reactions. We need to determine the reaction forces along dof 1, 2, 4, 7, and 8, which correspond to fixed supports. These are obtained by substituting for **Q** into the original finite element equation **R** = **KQ** − **F**. In this substitution, only those rows of **K** corresponding to the support dof are needed, and **F** = **O** for these dof. Thus, we have

$$\begin{Bmatrix} R_1 \\ R_2 \\ R_4 \\ R_7 \\ R_8 \end{Bmatrix} = \frac{29.5 \times 10^6}{600}\begin{bmatrix} 22.68 & 5.76 & -15.0 & 0 & -7.68 & -5.76 & 0 & 0 \\ 5.76 & 4.32 & 0 & 0 & -5.76 & -4.32 & 0 & 0 \\ 0 & 0 & 0 & 20.0 & 0 & -20.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -15.0 & 0 & 15.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} 0 \\ 0 \\ 27.12 \times 10^{-3} \\ 0 \\ 5.65 \times 10^{-3} \\ -22.25 \times 10^{-3} \\ 0 \\ 0 \end{Bmatrix}$$

which results in

$$\begin{Bmatrix} R_1 \\ R_2 \\ R_4 \\ R_7 \\ R_8 \end{Bmatrix} = \begin{Bmatrix} -15833.0 \\ 3126.0 \\ 21879.0 \\ -4167.0 \\ 0 \end{Bmatrix} \text{lb}$$

A free body diagram of the truss with reaction forces and applied loads is shown in Fig. E4.1b. ■

Temperature Effects

The thermal stress problem is considered here. Since a truss element is simply a 1-D element when viewed in the local coordinate system, the element temperature load in the local coordinate system is given by (see Eq. 3.106b)

$$\mathbf{\Theta}' = E_e A_e \epsilon_0 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \tag{4.17}$$

where the initial strain ϵ_0 associated with a temperature change is given by

$$\epsilon_0 = \alpha \Delta T \tag{4.18}$$

in which α is the coefficient of thermal expansion, and ΔT is the average change in temperature in the element. It may be noted that the initial strain ϵ_0 can also be induced by forcing members into places that are either too long or too short, due to fabrication errors.

We will now express the load vector in Eq. 4.17 in the global coordinate system. Since the potential energy associated with this load is the same in magnitude whether measured in the local or global coordinate systems, we have

$$\mathbf{q}'^{\mathrm{T}}\mathbf{\Theta}' = \mathbf{q}^{\mathrm{T}}\mathbf{\Theta} \tag{4.19}$$

where $\mathbf{\Theta}$ is the load vector in the global coordinate system. Substituting for $\mathbf{q}' = \mathbf{Lq}$ into Eq. 4.19, we get

$$\mathbf{q}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{\Theta}' = \mathbf{q}^{\mathrm{T}}\mathbf{\Theta} \tag{4.20}$$

Comparing the left and right sides of this equation, we obtain

$$\mathbf{\Theta} = \mathbf{L}^{\mathrm{T}}\mathbf{\Theta}' \tag{4.21}$$

Substituting for $\mathbf{L}$ from Eq. 4.5, we can write down the expression for the element temperature load as

$$\mathbf{\Theta}^e = E_e A_e \epsilon_0 \begin{Bmatrix} -\ell \\ -m \\ \ell \\ m \end{Bmatrix} \tag{4.22}$$

The temperature loads, along with other externally applied loads, are assembled in the usual manner to obtain the nodal load vector $\mathbf{F}$. Once the displacements are obtained by solving the finite element equations, the stress in each truss element is obtained from (see Eq. 3.103)

$$\sigma = E(\epsilon - \epsilon_0) \tag{4.23}$$

This equation for the element stress can be simplified by using Eq. 4.16 and noting that $\epsilon_0 = \alpha\Delta T$, to obtain

$$\sigma = \frac{E_e}{\ell_e}[-\ell \quad -m \quad \ell \quad m]\mathbf{q} - E_e\alpha\,\Delta T \tag{4.24}$$

Example 4.2

The four-bar truss in Example 4.1 is considered here, but the loading is different. Take $E = 29.5 \times 10^6$ psi and $\alpha = 1/150000$ per °F.

(a) There is an increase in temperature of 50°F in bars 2 and 3 only (Fig. E4.2a). There are no other loads on the structure. Determine the nodal displacements and element stresses as a result of this temperature increase. Use the elimination approach.

(b) A support settlement effect is considered here. Node 2 settles by 0.12 in. vertically down, and in addition, two point loads are applied on the structure (Fig. E4.2b). Write down (without solving) the equilibrium equations $\mathbf{KQ} = \mathbf{F}$, where $\mathbf{K}$ and $\mathbf{F}$ are the modified structural stiffness matrix and load vector, respectively. Use the penalty approach.

(c) Use the program TRUSS2D to obtain the solution to part (b).

Solution

(a) The stiffness matrix for the truss structure has already been developed in Example 4.1. Only the load vector needs to be assembled due to the temperature increase. Using Eq. 4.22, the temperature load as a result of temperature increases in elements 2 and 3, respectively,

$$\mathbf{\Theta}^2 = \frac{29.5 \times 10^6 \times 50}{150{,}000}\begin{array}{l} \\ \left\{\begin{array}{r} 0 \\ 1 \\ 0 \\ -1 \end{array}\right\} \end{array}\begin{array}{l} \downarrow \text{ Global dof} \\ 5 \\ 6 \\ 3 \\ 4 \end{array}$$

and

$$\mathbf{\Theta}^3 = \frac{29.5 \times 10^6 \times 50}{150{,}000}\left\{\begin{array}{r} -0.8 \\ -0.6 \\ 0.8 \\ 0.6 \end{array}\right\}\begin{array}{l} 1 \\ 2 \\ 5 \\ 6 \end{array}$$

The $\mathbf{\Theta}^2$ and $\mathbf{\Theta}^3$ vectors contribute to the global load vector $\mathbf{F}$. Using the elimination approach, we can delete all rows and columns corresponding to support dof in $\mathbf{K}$ and $\mathbf{F}$. The resulting finite element equations are

$$\frac{29.5 \times 10^6}{600}\begin{bmatrix} 15.0 & 0 & 0 \\ 0 & 22.68 & 5.76 \\ 0 & 5.76 & 24.32 \end{bmatrix}\begin{Bmatrix} Q_3 \\ Q_5 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 7866.7 \\ 15733.3 \end{Bmatrix}$$

which yield

$$\begin{Bmatrix} Q_3 \\ Q_5 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0.003951 \\ 0.01222 \end{Bmatrix} \text{ in.}$$

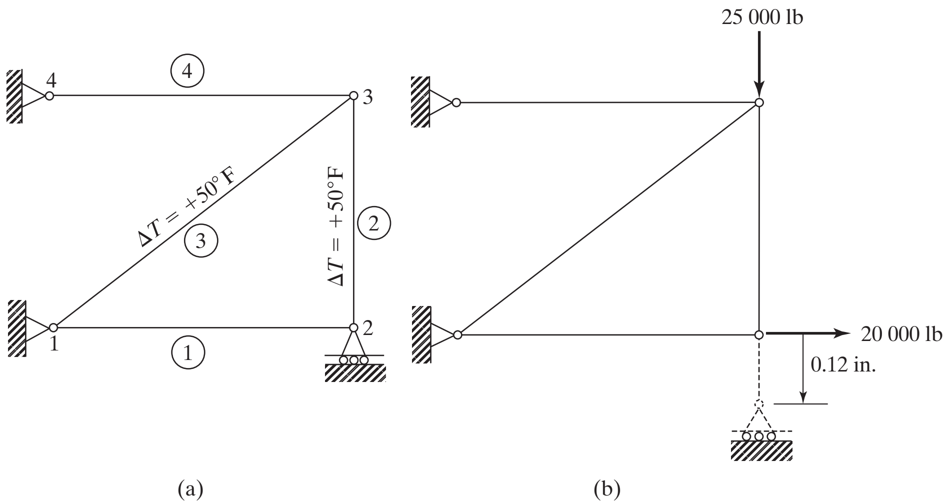


FIGURE E4.2

The element stresses can now be obtained from Eq. 4.24. For example, the stress in element 2 is given as

$$\sigma_2 = \frac{29.5 \times 10^6}{30}[0 \quad 1 \quad 0 \quad -1]\begin{Bmatrix} 0.003951 \\ 0.01222 \\ 0 \\ 0 \end{Bmatrix} - \frac{29.5 \times 10^6 \times 50}{150{,}000}$$

$$= 2183 \text{ psi}$$

The complete stress solution is

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 2183 \\ -3643 \\ 2914 \end{Bmatrix} \text{psi}$$

(b) Support 2 settles by 0.12 in. vertically down, and two concentrated forces are applied (Fig. E4.2b). In the penalty approach for handling boundary conditions (Chapter 3), recall that a large spring constant C is added to the diagonal elements in the structural stiffness matrix at those dof where the displacements are specified. Typically, C may be chosen 10^4 times the largest diagonal element of the unmodified stiffness matrix (see Eq. 3.83). Further, a force Ca is added to the force vector, where a is the specified displacement. In this example, for degree of freedom (dof) 4, $a = -0.12$ in., and consequently, a force equal to $-0.12C$ gets added to the fourth location in the force vector. Consequently, the modified finite element equations are given by

$$\frac{29.5 \times 10^6}{600}\begin{bmatrix} 22.68 + C & 5.76 & -15.0 & 0 & -7.68 & -5.76 & 0 & 0 \\ & 4.32 + C & 0 & 0 & -5.76 & -4.32 & 0 & 0 \\ & & 15.0 & 0 & 0 & 0 & 0 & 0 \\ & & & 20.0 + C & 0 & -20.0 & 0 & 0 \\ & & & & 22.68 & 5.76 & -15.0 & 0 \\ & & & & & 24.32 & 0 & 0 \\ & & & & & & 15.0 + C & 0 \\ \text{Symmetric} & & & & & & & C \end{bmatrix}\begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ Q_7 \\ Q_8 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 20\,000 \\ -0.12C \\ 0 \\ -25\,000.0 \\ 0 \\ 0 \end{Bmatrix}$$

(c) Obviously, the equations in (b) are too large for hand calculations. In the program TRUSS, that is provided, these equations are automatically generated and solved from the user's input data. The output from the program is

$$\begin{Bmatrix} Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} 0.0271200 \\ -0.1200145 \\ 0.0323242 \\ -0.1272606 \end{Bmatrix} \text{in.}$$

and

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{Bmatrix} = \begin{Bmatrix} 20\,000.0 \\ -7\,125.3 \\ -29\,791.7 \\ 23\,833.3 \end{Bmatrix} \text{psi}$$

■

4.3 THREE-DIMENSIONAL TRUSSES

The 3-D truss element can be treated as a straightforward generalization of the 2-D truss element discussed earlier. The local and global coordinate systems for a 3-D truss element are shown in Fig. 4.5. Note that the local coordinate system is again the x'-axis running along the element, since a truss element is simply a two-force member. Consequently, the nodal displacement vector in local coordinates is

$$\mathbf{q}' = [q'_1, q'_2]^{\mathrm{T}} \tag{4.25}$$

The nodal displacement vector in global coordinates is (Fig. 4.5b)

$$\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^{\mathrm{T}} \tag{4.26}$$

Referring to Fig. 4.5, we find that the transformation between local and global coordinates is

$$\mathbf{q}' = \mathbf{L}\mathbf{q} \tag{4.27}$$

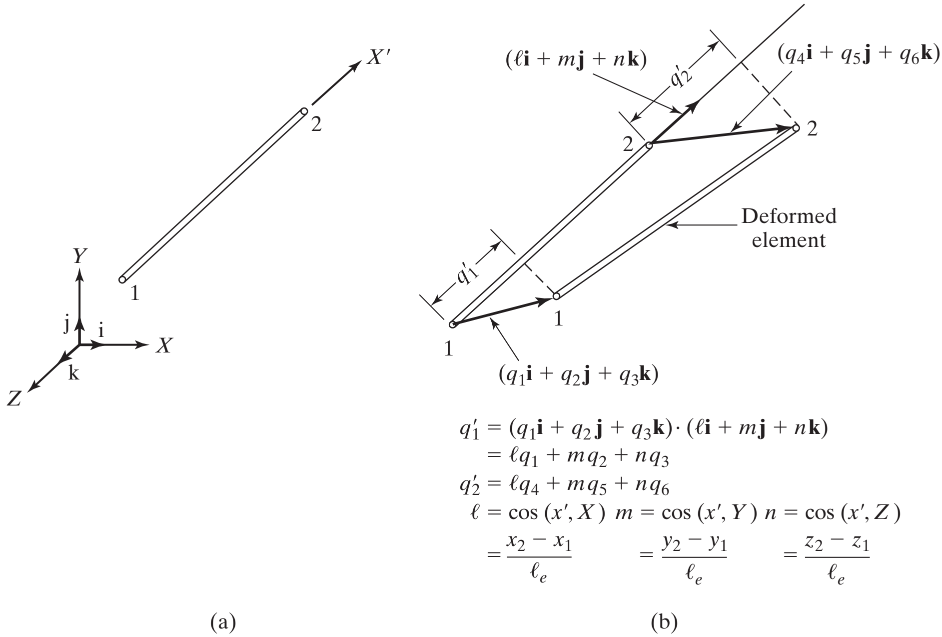


FIGURE 4.5 A three-dimensional truss element in local and global coordinate systems.

where the transformation matrix $\mathbf{L}$ is given by

$$\mathbf{L} = \begin{bmatrix} \ell & m & n & 0 & 0 & 0 \\ 0 & 0 & 0 & \ell & m & n \end{bmatrix} \tag{4.28}$$

in which ℓ, m, and n are the direction cosines of the local x'-axis with respect to the global x-, y-, and z-axes, respectively. The element stiffness matrix in global coordinates is given by Eq. 4.12, which yields

$$\mathbf{k} = \frac{E_e A_e}{\ell_e} \begin{bmatrix} \ell^2 & \ell m & \ell n & -\ell^2 & -\ell m & -\ell n \\ \ell m & m^2 & mn & -\ell m & -m^2 & -mn \\ \ell n & mn & n^2 & -\ell n & -mn & -n^2 \\ -\ell^2 & -\ell m & -\ell n & \ell^2 & \ell m & \ell n \\ -\ell m & -m^2 & -mn & \ell m & m^2 & mn \\ -\ell n & -mn & -n^2 & \ell n & mn & n^2 \end{bmatrix} \tag{4.29}$$

The formulas for calculating ℓ, m, and n are

$$\ell = \frac{x_2 - x_1}{\ell_e} \quad m = \frac{y_2 - y_1}{\ell_e} \quad n = \frac{z_2 - z_1}{\ell_e} \tag{4.30}$$

where the length ℓ_e of the element is given by

$$\ell_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{4.31}$$

Generalizations of the element stress and element temperature load expressions are left as an exercise.

4.4 ASSEMBLY OF GLOBAL STIFFNESS MATRIX FOR THE BANDED AND SKYLINE SOLUTIONS

The solution of the finite element equations should take advantage of symmetry and sparsity of the global stiffness matrix. Two methods, the banded approach and the skyline approach, are discussed in Chapter 2. In the banded approach, the elements of each element stiffness matrix $\mathbf{k}^e$ are directly placed in a banded matrix $\mathbf{S}$. In the skyline approach, the elements of $\mathbf{k}^e$ are placed in a vector form with certain identification pointers. The bookkeeping aspects of this assembly procedure for banded and skyline solution are discussed in the sections that follow.

Assembly for Banded Solution

The assembly of elements of $\mathbf{k}^e$ into a banded global stiffness matrix $\mathbf{S}$ is now discussed for a 2-D truss element. Consider an element e whose connectivity is indicated as follows:

Element	1	2	← Local Node Nos.
e	i	j	← Global Node Nos.

The element stiffness with its associated dof are

$$\mathbf{k}^e = \begin{array}{c} \begin{array}{cccc} 2i-1 & 2i & 2j-1 & 2j \end{array} \\ \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ & k_{22} & k_{23} & k_{24} \\ & & k_{33} & k_{34} \\ \text{Symmetric} & & & k_{44} \end{bmatrix} \end{array} \begin{array}{l} \text{Global dofs} \\ 2i-1 \\ 2i \\ 2j-1 \\ 2j \end{array} \tag{4.32}$$

The principal diagonal of $\mathbf{k}^e$ is placed in the first column of $\mathbf{S}$, the next-to-principal diagonal is placed in the second column, and so on. Thus, the correspondence between elements in $\mathbf{k}^e$ and $\mathbf{S}$ is given by (see Eq. 2.39)

$$k^e_{\alpha,\beta} \longrightarrow S_{p,q-p+1} \tag{4.33}$$

where α and β are the local dof taking on values 1, 2, 3, and 4, while p and q are global dof taking on values of $2i - 1, 2i, 2j - 1, 2j$. For instance,

$$k^e_{1,3} \longrightarrow S_{2i-1,2(j-i)+1}$$

and

$$k^e_{4,4} \longrightarrow S_{2j,1} \tag{4.34}$$

This assembly is done only for elements in the upper triangle owing to symmetry. Thus, Eq. 4.33 is valid only for $q \geq p$. We can now follow the assembly steps given in program TRUSS2D.

A formula for the half-bandwidth, NBW, in 2-D truss structures can be readily derived. Consider a truss element e connected to, say, nodes 4 and 6. The dof for the element are 7, 8, 11, and 12. Thus, the entries in the global stiffness matrix for this element will be

$$\begin{array}{cc}
\begin{array}{cccccccccc} 1 & 2 & \cdots & 7 & 8 & \cdots & 11 & 12 & \cdots & N \end{array} & \\
\left[\begin{array}{cccccccccc}
 & & & & & & & & & \\
 & & & \longleftarrow & & m & & \longrightarrow & & \\
 & & & & & & & & & \\
 & & & \times & \times & \cdots & \times & \times & & \\
 & & & & \times & \cdots & \times & \times & & \\
 & & & & & \cdots & \cdot & \cdot & & \\
 \text{Symmetric} & & & & & & \times & \times & & \\
 & & & & & & & \times & & \\
 & & & & & & & & & \\
 & & & & & & & & &
\end{array}\right] & \begin{array}{c} 1 \\ 2 \\ \vdots \\ 7 \\ 8 \\ \vdots \\ 11 \\ 12 \\ \vdots \\ N \end{array}
\end{array} \tag{4.35}$$

We see that the span m of nonzero entries is equal to 6, which also follows from the connecting node numbers: $m = 2[6 - 4 + 1]$. In general, the span associated with an element e connecting nodes i and j is

$$m_e = 2[|i - j| + 1] \tag{4.36}$$

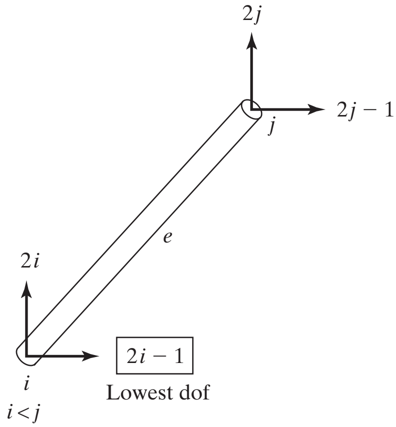
Thus, the maximum span or half-bandwidth is

$$\text{NBW} = \max_{1 \leq e \leq \text{NE}} m_e \tag{4.37}$$

In the banded approach, we see that differences in node numbers connecting an element should be kept to a minimum for computational efficiency.

Skyline Assembly

As discussed in Chapter 2, the first step in skyline assembly involves the evaluation of the skyline height or the column height for each diagonal location. Consider the element e with the end nodes i and j shown in Fig. 4.6. Without loss of generality, let i be the smaller node number; that is, $i < j$. Then, starting with a vector of identifiers, **ID**, we look at the four dof $2i - 1$, $2i$, $2j - 1$, and $2j$. At the location corresponding to one of these four dof represented by I, the previous value is replaced by the larger of the two numbers ID(I) and $I - (2i - 1) + 1$. This is precisely represented in the table given in Fig. 4.6. The process is repeated over all the elements. At this stage all the skyline heights



Element e

Location No. I	Skyline Height $\mathrm{ID}(I)$
$2i-1$	max (1, OLD)
$2i$	max (2, OLD)
$2j-1$	max $(2j-2i+1, \text{OLD})$
$2j$	max $(2j-2i+2, \text{OLD})$

max $(X, \text{OLD}) \equiv$ REPLACE by X if $X >$ OLD
(start value of OLD = 0)

FIGURE 4.6 Skyline heights.

have been determined and placed in the vector **ID**. Then, starting from location $I = 2$, replacing the location at I by the sum $\mathrm{ID}(I) + \mathrm{ID}(I-1)$ gives the pointer numbers as discussed in Chapter 2.

The next step involves assembling the element stiffness values into the column vector **A**. The correspondence of the global locations of the square stiffness matrix coming from an element shown in Fig. 4.6 is clearly presented in Fig. 4.7, using the diagonal

Column

Column vector A ⇨

$2i-1$	$2i$	$2j-1$	$2j$	Row
$A[\mathrm{ID}(2i-1)]$	$A[\mathrm{ID}(2i)-1]$	$A[\mathrm{ID}(2j-1)-(2j-2i)]$	$A[\mathrm{ID}(2j)-(2j-2i+1)]$	$2i-1$
	$A[\mathrm{ID}(2i)]$	$A[\mathrm{ID}(2j-1)-(2j-1-2i)]$	$A[\mathrm{ID}(2j)-(2j-2i)]$	$2i$
		$A[\mathrm{ID}(2j-1)]$	$A[\mathrm{ID}(2j)-1]$	$2j-1$
			$A[\mathrm{ID}(2j)]$	$2j$

FIGURE 4.7 Stiffness locations in column vector form for skyline solution.

pointers discussed previously. The details presented earlier have been implemented in program TRUSSKY. Other programs provided may be similarly modified for skyline solution instead of banded solution.

4.5 PROBLEM MODELING AND BOUNDARY CONDITIONS

Boundary conditions must be properly imposed in order to solve the truss problem. A general 2-D truss needs to have a minimum of three boundary conditions, for example, one joint restricted in both x and y directions, and another joint fixed along, say, y direction. If this condition is not satisfied, a division by zero error occurs when the finite element solution is attempted. In some situations, we have inclined roller supports, which need special treatment. Problem size can be reduced using symmetry and antisymmetry. These are discussed now.

Inclined Support in Two Dimensions

An inclined roller support where the node j on the truss moves along a line along $\mathbf{n} = \cos\theta\ \mathbf{i} + \sin\theta\ \mathbf{j}$ is shown in Fig. 4.8. The constraint yields $\mathbf{Q}_{2j-1} = c\cos\theta$, $\mathbf{Q}_{2j} = c\sin\theta$, where c is a constant of proportionality. This results in the equation

$$-Q_{2j-1}\sin\theta + Q_{2j}\cos\theta = 0 \tag{4.38}$$

This is clearly a multipoint constraint, which can be implemented as discussed in Chapter 3.

Inclined Support in Three Dimensions—Line Constraint

Problems of this type may occur in 3-D trusses. The node j is restricted to move along a line having a direction $\mathbf{t} = \ell\mathbf{i} + m\mathbf{j} + n\mathbf{k}$ as shown in Fig. 4.9. ℓ, m, n are the direction cosines. The condition for the line constraint is $\mathbf{q} = \alpha\mathbf{t}$, which is equivalent to

$$\begin{aligned} mq_1 &= \ell q_2 \\ nq_2 &= mq_3 \\ \ell q_3 &= nq_1 \end{aligned} \tag{4.39}$$

This may be handled by the penalty approach by adding the following stiffness terms at appropriate locations indicated to the global stiffness matrix.

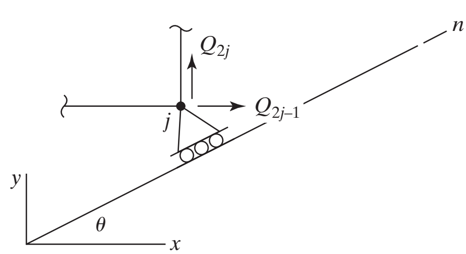


FIGURE 4.8 Inclined roller support.

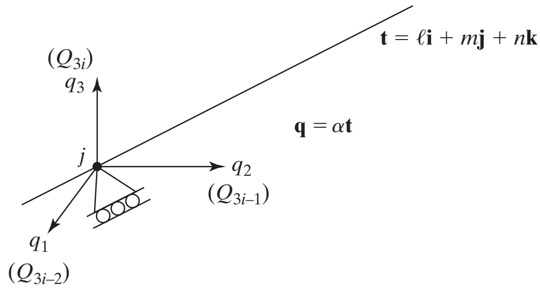


FIGURE 4.9 Line constraint in three dimensions.

$$
\begin{array}{c|ccc}
 & 3j-2 & 3j-1 & 3j \\
\hline
3j-2 & C(1-\ell^2) & -C\ell m & -C\ell n \\
3j-1 & -C\ell m & C(1-m^2) & -Cmn \\
3j & -C\ell n & -Cmn & C(1-n^2)
\end{array}
\tag{4.40}
$$

C is the penalty constant, which is large in comparison to the largest diagonal stiffness value as discussed earlier.

Inclined Support in Three Dimensions—Plane Constraint

The node j may be restricted to move on a plane having a normal direction $\mathbf{t} = \ell\mathbf{i} + m\mathbf{j} + n\mathbf{k}$ as shown in Fig. 4.10. ℓ, m, n are the direction cosines. The condition for the line constraint is

$$\mathbf{q} \cdot \mathbf{t} = \ell q_1 + mq_2 + nq_3 = 0 \tag{4.41}$$

This may be handled by the penalty approach by adding the following stiffness terms at appropriate locations indicated to the global stiffness matrix.

$$
\begin{array}{c|ccc}
 & 3j-2 & 3j-1 & 3j \\
\hline
3j-2 & C\ell^2 & C\ell m & C\ell n \\
3j-1 & C\ell m & Cm^2 & -Cmn \\
3j & C\ell n & Cmn & Cn^2
\end{array}
\tag{4.42}
$$

Penalty constant C is as defined above.

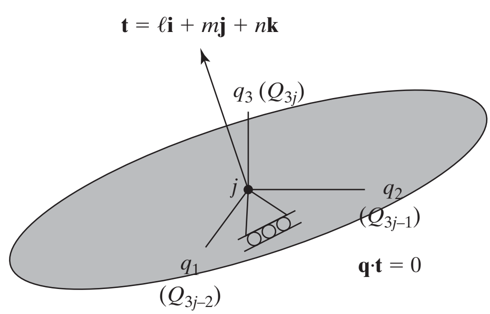


FIGURE P4.10 Plane constraint in three dimensions.

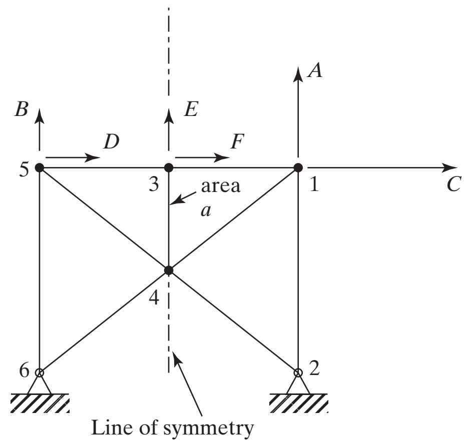


FIGURE 4.11 Symmetric geometry.

Symmetry and Antisymmetry

We consider here symmetry of the geometry about y-axis and symmetric and antisymmetric loads. Let the loading on a symmetric nine-bar truss be as shown in Fig. 4.11. By splitting the load at a point into two halves and placing two equal and opposite loads of half the magnitude of the load at its symmetric point, we can break the loading into symmetric and antisymmetric parts, as shown in Fig. 4.12. It is easy to see that by superposition of the antisymmetric load on the symmetric load, we get the original loading. If results are obtained for the two cases, the combined result

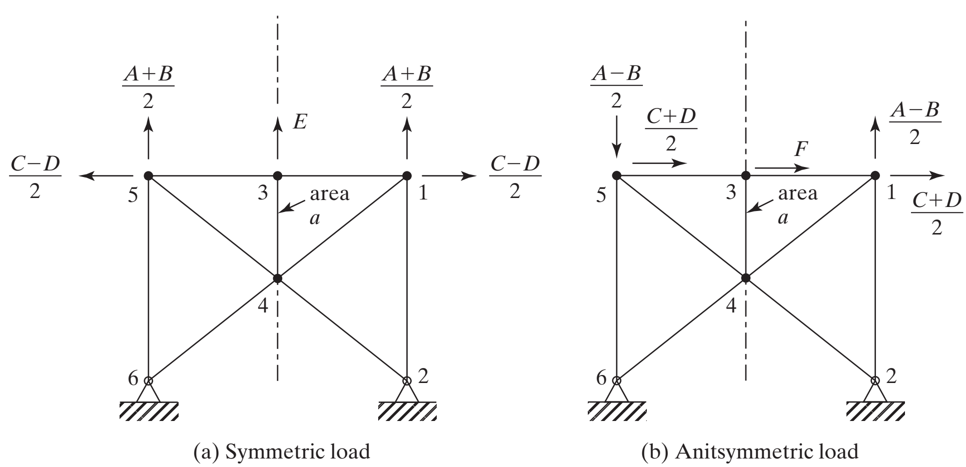


FIGURE 4.12 Symmetry and antisymmetry.

can be found by superposition. If $u(x, y)$ and $v(x, y)$ are the x and y displacements of the node at x, y, then we have the following relations for symmetric and antisymmetric cases

$$\text{Symmetric loading} \quad \begin{aligned} u(-x, y) &= -u(x, y) \\ v(-x, y) &= v(x, y) \end{aligned} \tag{4.43}$$

$$\text{Antisymmetric loading} \quad \begin{aligned} u(-x, y) &= u(x, y) \\ v(-x, y) &= -v(x, y) \end{aligned} \tag{4.44}$$

If $\sigma(x, y)$ is stress in a member, then

$$\text{Symmetric loading} \quad \sigma(-x, y) = \sigma(x, y) \tag{4.43}$$

$$\text{Antisymmetric loading} \quad \sigma(-x, y) = -\sigma(x, y) \tag{4.44}$$

The points on the line of symmetry are constrained to move in the y direction (x displacement is zero) for symmetric loading. The points on the line of symmetry are constrained to move in the x direction (y displacement is zero) for antisymmetric loading. Using this idea, we only need to model one-half of the problem, as shown in Figs. 4.13a and 4.13b. We show the right half of the problem for solving. We then superpose and use Eqs. 4.43 and 4.44 to obtain values in the other half.

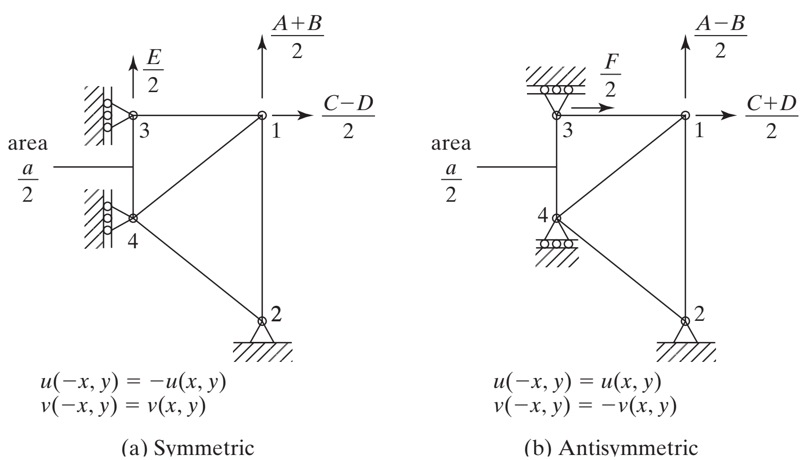


FIGURE 4.13 Half geometry model.

Input Data/Output

```
INPUT TO TRUSS2D, TRUSSKY
<< 2D TRUSS ANALYSIS USING BAND SOLVER>>
Example 4.1
   NN  NE NM NDIM  NEN  NDN
    4   4  1   2     2    2
   ND  NL NMPC
    5   2  0
   Node#   X    Y
    1      0    0
    2     40    0
    3     40   30
    4      0   30
   Elem# N1    N2  Mat#  Area  TempRise
    1      1     2   1      1      0
    2      3     2   1      1      0
    3      3     1   1      1      0
    4      4     3   1      1      0
   DOF#  Displacement
    1      0
    2      0
    4      0
    7      0
    8      0
   DOF#  Load
    3     20000
    6    -25000
   MAT#   E          Alpha
    1     2.95E+07  1.20E-05
   B1     i     B2   j      B3 (Multi-point constr. B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM TRUSS2D
Results from Program TRUSS2D
Example 4.1
Node#   X-Displ
     1  1.32411E-06 -2.61375E-07
     2  0.027119968 -1.82939E-06
     3  0.005650704 -0.022247233
     4  3.48501E-07 0
Elem#   Stress
     1  20000
     2  -21874.64737
     3  -5208.921046
     4  4167.136837
DOF#     Reaction
     1  -15832.86316
     2   3125.352627
     4   21874.64737
     7  -4167.136837
     8  0
```

PROBLEMS

4.1. Consider the truss element shown in Fig. P4.1. The x-, y-coordinates of the two nodes are indicated in the figure. If $\mathbf{q} = [2, 0.7, 2.5, 5] \times 10^{-2}$ in., determine the following:

(a) the vector $\mathbf{q}'$
(b) the stress in the element
(c) the $\mathbf{k}$ matrix
(d) the strain energy in the element.

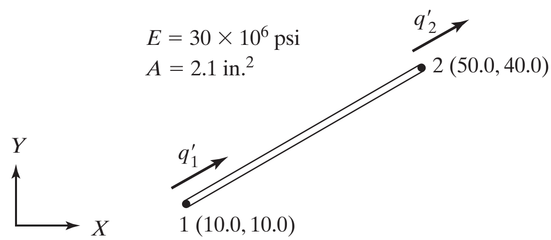


FIGURE P4.1

4.2. A truss element, with local node numbers 1 and 2, is shown in Fig. P4.2.

(a) What are the direction cosines ℓ and m.
(b) Show the x'-axis, $q_1, q_2, q_3, q_4, q'_1, q'_2$ on the figure.
(c) If $\mathbf{q} = [0., 0.01, -0.025, -0.05]^{\mathrm{T}}$, determine q'_1, q'_2.

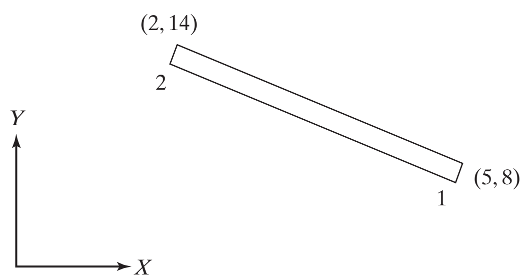


FIGURE P4.2

4.3. For the pin-jointed configuration shown in Fig. P4.3, determine the stiffness values K_{11}, K_{12}, and K_{22} of the global stiffness matrix.

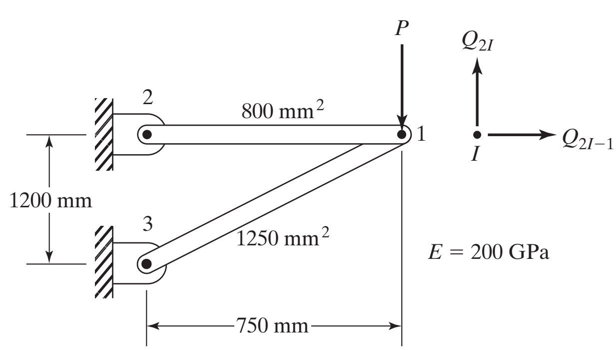


FIGURE P4.3

4.4. For the truss in Fig. P4.4, a horizontal load of $P = 2500$ lb is applied in the x direction at node 2.

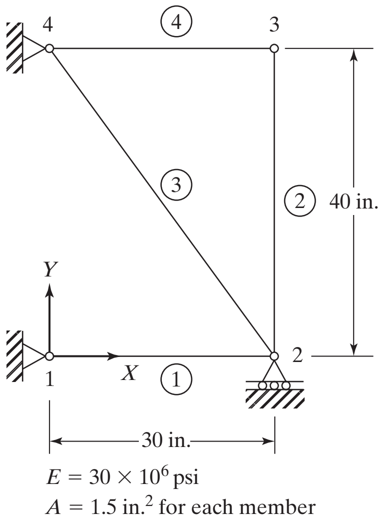


FIGURE P4.4

(a) Write down the element stiffness matrix **k** for each element.
(b) Assemble the **K** matrix.
(c) Using the elimination approach, solve for **Q**.
(d) Evaluate the stress in elements 2 and 3.
(e) Determine the reaction force at node 2 in the y direction.

4.5. For Fig. P4.4, determine the stresses in each element due to the following support movement: support at node 2, 0.36 in. down.

4.6. For the two-bar truss shown in Fig. P4.6, determine the displacements of node 1 and the stress in elements 1–3.

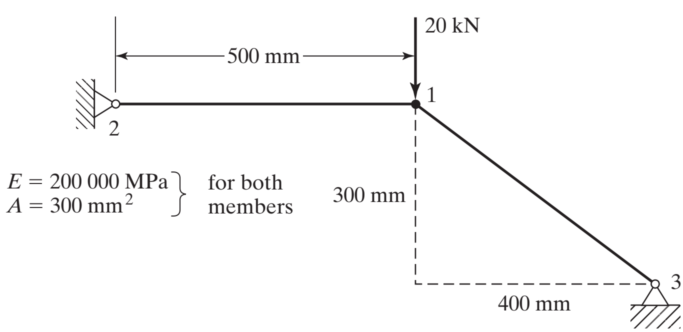


FIGURE P4.6

4.7. For the three-bar truss shown in Fig. P4.7, determine the displacements of node 1 and the stress in element 3.

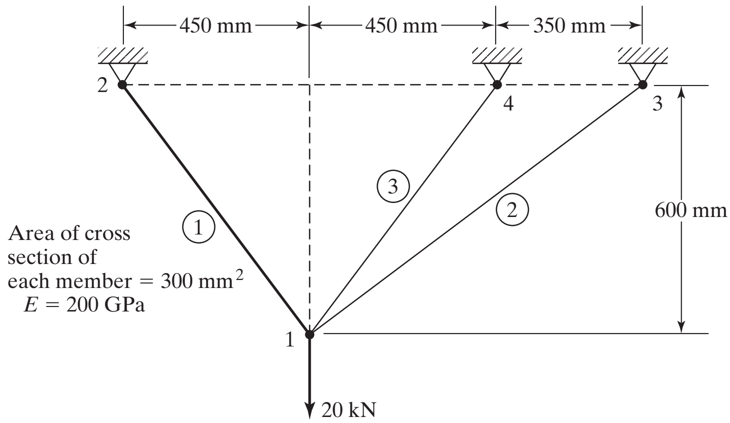


FIGURE P4.7

4.8. For the 2-D truss configuration shown in Fig. P4.8, determine the bandwidth for stiffness storage in a banded form. Choose an alternative numbering scheme and determine the corresponding bandwidth. Comment on the strategy that you use for decreasing the bandwidth.

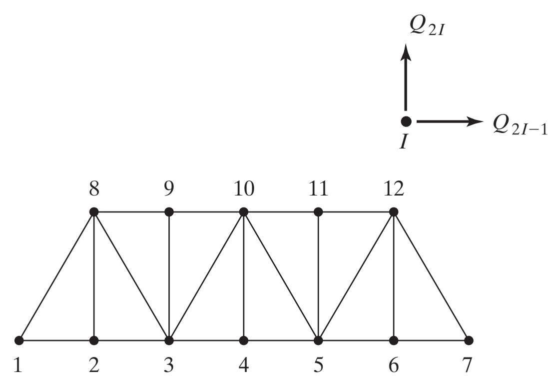


FIGURE P4.8

4.9. A small railroad bridge is constructed of steel members, all of which have a cross-sectional area of 3000 mm^2. A train stops on the bridge, and the loads applied to the truss on one side of the bridge are as shown in Fig. P4.9. Estimate how much the point R moves horizontally because of this loading. Also determine the nodal displacements and element stresses.

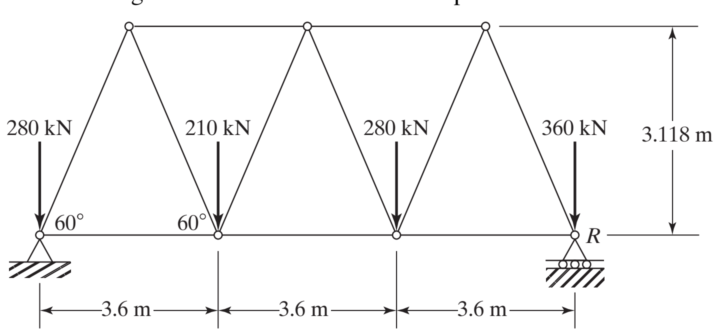


FIGURE P4.9

4.10. Consider the truss in Fig. P4.10 loaded as shown. Cross-sectional areas in square inches are shown in parentheses. Consider symmetry and model only one-half of the truss shown. Determine displacements and element stresses. Let $E = 30 \times 10^6$ psi.

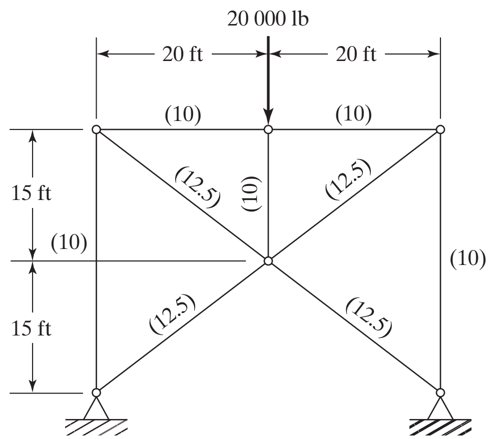


FIGURE P4.10

4.11. Determine the nodal displacements and element stresses in the truss in Fig. P4.11, due to each of the following conditions:

(a) Increase of temperature of 50°F in elements 1, 3, 7, and 8.

(b) Elements 9 and 10 are $\frac{1}{4}$ in. too short and element 6 is $\frac{1}{8}$ in. too long, owing to errors in fabrication, and it was necessary to force them into place.

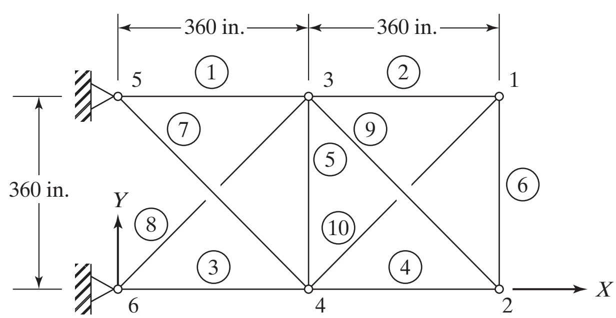


FIGURE P4.11

(c) Support at node 6 moves 0.12 in. down. *Data*: Take $E = 30 \times 10^6$ psi, $\alpha = 1/150\ 000$ per °F. Cross-sectional areas for each element are as follows:

Element	Area (in.^{2})
1, 3	25
2, 4	12
5	1
6	4
7, 8, 9	17
10	5

4.12. A two-member truss is subjected to a load $P = 8000$ N. Member 1–2 is 400 mm long. Member 1–3 was manufactured to be 505 mm long instead of 500 mm. However, it was forced into place. Determine

(a) the stresses in the members assuming that member 1–3 was manufactured to its correct length of 500 mm

(b) the stresses in the members as a result of member 1–3 being forced into place (and the load P, of course).

(*Hint*: Treat this as an initial strain problem and use the temperature load vector expression in the text.) Take cross-sectional areas = 750 mm^2, $E = 200$ GPa.

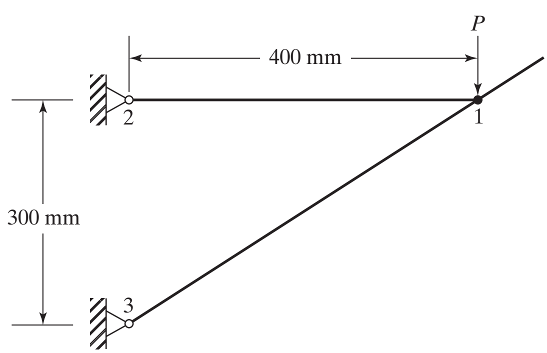


FIGURE P4.12

4.13. Expressions for the element stress (Eq. 4.16), and element temperature load (Eq. 4.22) were derived for a 2-D truss element. Generalize these expressions for a 3-D truss element.

4.14. Find deflections at nodes, stresses in members, and reactions at supports for the truss shown in Fig. P4.14 when the 200-kip load is applied.

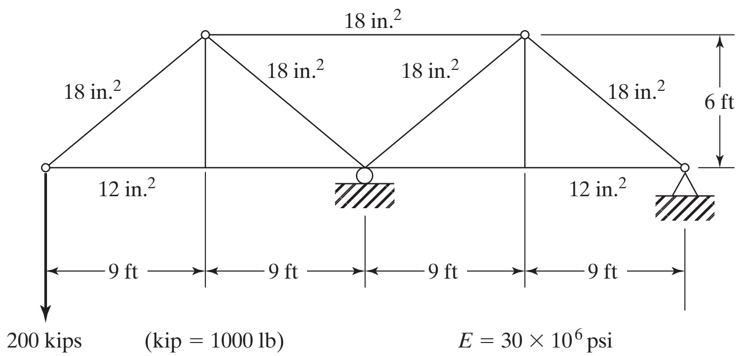


FIGURE P4.14

4.15. Find the deflections at the nodes for the truss configuration shown in Fig. P4.15. Area = 8 in.^{2} for each member.

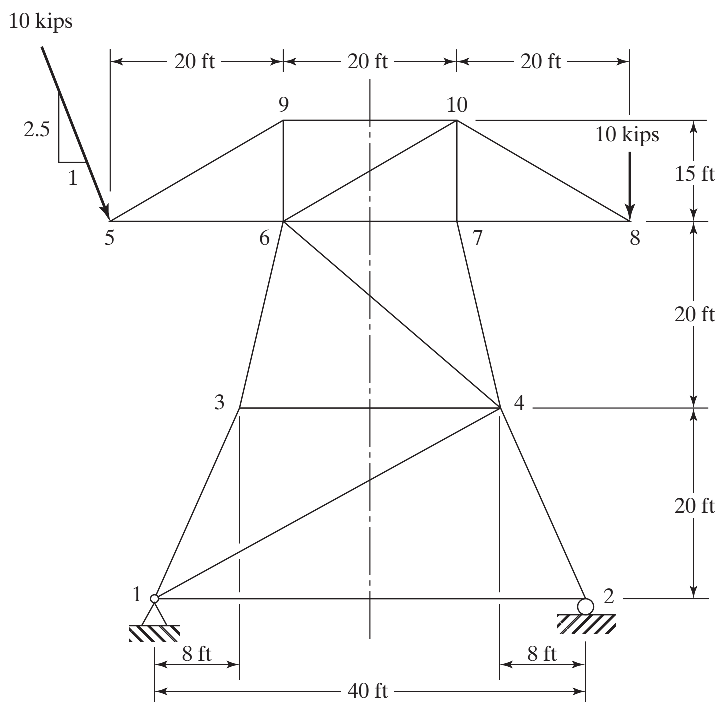


FIGURE P4.15

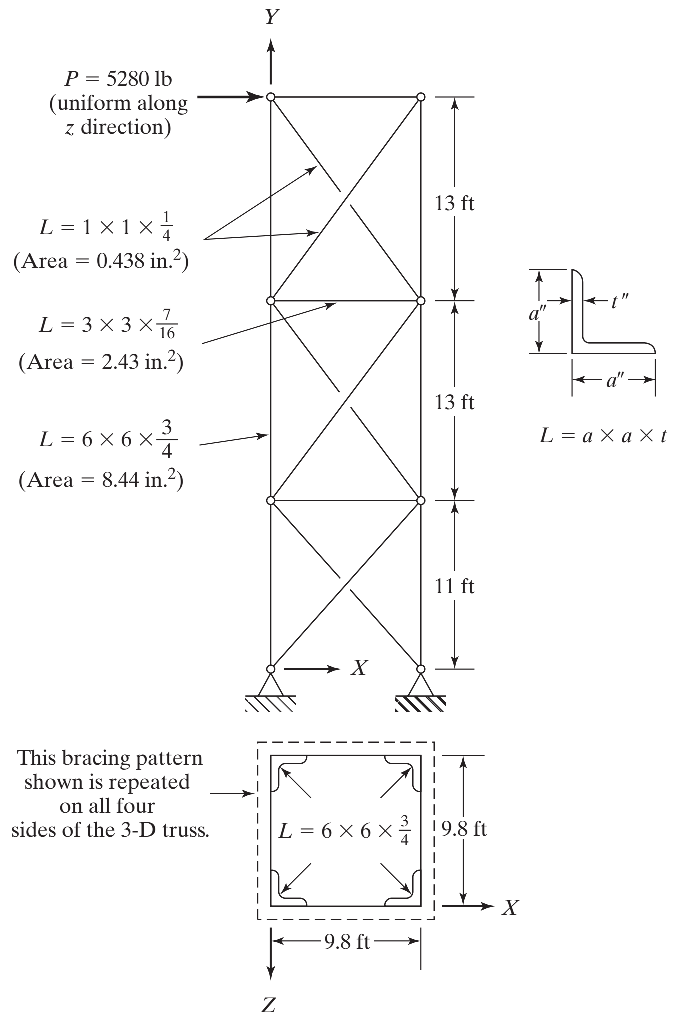
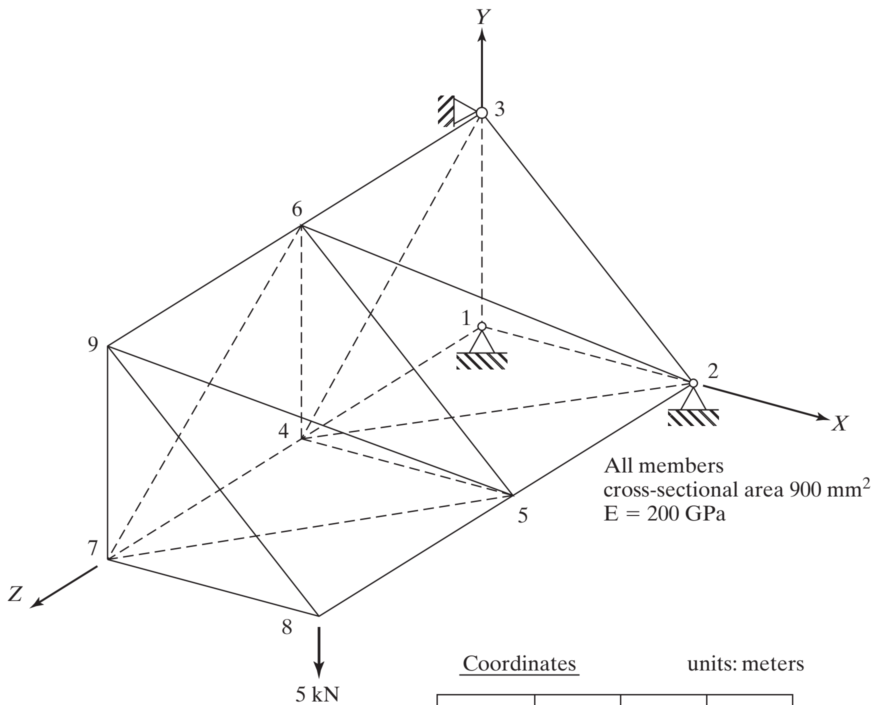


FIGURE P4.16 Three-dimensional truss model of a steel tower, supporting a water tank, and subjected to wind loads.

4.16. Modify program TRUSS2D to handle 3-D trusses and solve the problem in Fig. P4.16.

4.17. If the members in the truss in Problem 4.9 have a moment of inertia I of $8.4 \times 10^5\,\text{mm}^4$ about the axis perpendicular to the plane of the truss, check the compression members for Euler buckling. The Euler buckling load P_{ct} is given by $(\pi^2 EI)/\ell^2$. If σ_c is the compressive stress in a member, then the factor of safety for buckling may be taken as $P_{cr}/A\sigma_c$. Introduce this into the computer program TRUSS2D to calculate the factors of safety in compression members and print them in the output file.

4.18. **(a)** Analyze the 3-D truss shown in Fig. P4.18. Identify the tetrahedral patterns in the truss.
(b) Generate the coordinates and connectivity if the two-stage truss shown is extended to 10 stages.



Connectivity

1 - 2	4 - 5	7 - 8
1 - 3	4 - 6	7 - 9
2 - 3	5 - 6	8 - 9
1 - 4	4 - 7	
2 - 5	5 - 8	
3 - 6	6 - 9	
2 - 4	5 - 7	
3 - 4	6 - 7	
2 - 6	5 - 9	

Coordinates units: meters

Node	X	Y	Z
1	0	0	0
2	0.25	0	0
3	0	0.25	0
4	0	0	0.5
5	0.25	0	0.5
6	0	0.25	0.5
7	0	0	1.0
8	0.25	0	1.0
9	0	0.25	1.0

FIGURE P4.18

4.19. For the K truss shown in Fig. P4.19, find displacements of nodes and stresses in elements. All horizontal members are 20 mm diameter aluminum rods and all other members are of 25 mm square section. Use symmetric geometry of the problem.

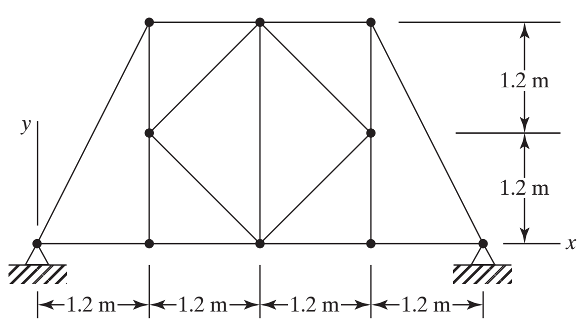


FIGURE P4.19

4.20. In the 11-member truss shown in Fig. P4.20, the horizontal and vertical members have a cross-sectional area of 200 mm^2, and all other members have a cross-sectional area of 90 mm^2. All members are made of steel. For the loading shown, determine nodal displacements and element stresses using symmetry/antisymmetry. If Euler critical load (buckling load) of a member of length l is $\pi^2 EI/l^2$, check if all members in compression are safe in buckling.

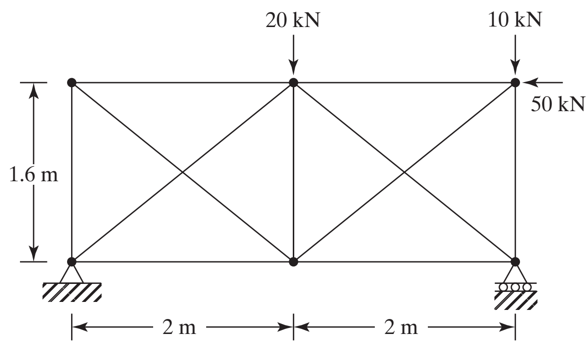


FIGURE P4.20

Program Listing

```
MAIN PROGRAM
'****************************************
'* PROGRAM TRUSS2D                      *
'* TWO DIMENSIONAL TRUSSES              *
'* T.R.Chandrupatla and A.D.Belegundu   *
'****************************************
Private Sub CommandButton1_Click()
   Call InputData
   Call Bandwidth
   Call Stiffness
   Call ModifyForBC
   Call BandSolver
   Call StressCalc
   Call ReactionCalc
   Call Output
End Sub
```

```
ELEMENT STIFFNESS AND ASSEMBLY OF GLOBAL STIFFNESS
Private Sub Stiffness()
     ReDim S(NQ, NBW)
     '----- Global Stiffness Matrix -----
     For N = 1 To NE
        I1 = NOC(N, 1): I2 = NOC(N, 2)
        I3 = MAT(N)
        X21 = X(I2, 1) - X(I1, 1)
        Y21 = X(I2, 2) - X(I1, 2)
        EL = Sqr(X21 * X21 + Y21 * Y21)
        EAL = PM(I3, 1) * AREA(N) / EL
        CS = X21 / EL: SN = Y21 / EL
     '----------- Element Stiffness Matrix SE() -----------
        SE(1, 1) = CS * CS * EAL
        SE(1, 2) = CS * SN * EAL: SE(2, 1) = SE(1, 2)
        SE(1, 3) = -CS * CS * EAL: SE(3, 1) = SE(1, 3)
        SE(1, 4) = -CS * SN * EAL: SE(4, 1) = SE(1, 4)
        SE(2, 2) = SN * SN * EAL
        SE(2, 3) = -CS * SN * EAL: SE(3, 2) = SE(2, 3)
        SE(2, 4) = -SN * SN * EAL: SE(4, 2) = SE(2, 4)
        SE(3, 3) = CS * CS * EAL
        SE(3, 4) = CS * SN * EAL: SE(4, 3) = SE(3, 4)
        SE(4, 4) = SN * SN * EAL
     '-------------- Temperature Load TL() ---------------
        EE0 = PM(I3, 2) * DT(N) * PM(I3, 1) * AREA(N)
        TL(1) = -EE0 * CS: TL(2) = -EE0 * SN
        TL(3) = EE0 * CS: TL(4) = EE0 * SN
     '----- Stiffness Assmbly -----
        For II = 1 To NEN
           NRT = NDN * (NOC(N, II) - 1)
           For IT = 1 To NDN
              NR = NRT + IT
              I = NDN * (II - 1) + IT
              For JJ = 1 To NEN
                 NCT = NDN * (NOC(N, JJ) - 1)
                 For JT = 1 To NDN
                    J = NDN * (JJ - 1) + JT
                    NC = NCT + JT - NR + 1
                    If NC > 0 Then
                      S(NR, NC) = S(NR, NC) + SE(I, J)
                    End If
                 Next JT
              Next JJ
              F(NR) = F(NR) + TL(I)
           Next IT
        Next II
     Next N
End Sub
```

```
STRESS CALCULATIONS
Private Sub StressCalc()
     ReDim Stress(NE)
     '----- Stress Calculations
     For I = 1 To NE
         I1 = NOC(I, 1)
         I2 = NOC(I, 2)
         I3 = MAT(I)
         X21 = X(I2, 1) - X(I1, 1): Y21 = X(I2, 2) - X(I1, 2)
         EL = Sqr(X21 * X21 + Y21 * Y21)
         CS = X21 / EL
         SN = Y21 / EL
         J2 = 2 * I1
         J1 = J2 - 1
         K2 = 2 * I2
         K1 = K2 - 1
         DLT = (F(K1) - F(J1)) * CS + (F(K2) - F(J2)) * SN
         Stress(I) = PM(I3, 1) * (DLT / EL - PM(I3, 2) * DT(I))
     Next I
End Sub
```

C H A P T E R 5

Beams and Frames

5.1 INTRODUCTION

Beams are slender members that are used for supporting transverse loading. Long horizontal members used in buildings and bridges, and shafts supported in bearings are some examples of beams. Complex structures with rigidly connected members are called frames and may be found in automobile and airplane structures and motion- and force-transmitting machines and mechanisms. In this chapter, we first present the finite element formulation for beams and extend these ideas to formulate and solve two-dimensional frame problems.

Beams with cross sections that are symmetric with respect to plane of loading are considered here. A general horizontal beam is shown in Fig. 5.1. Figure 5.2 shows the

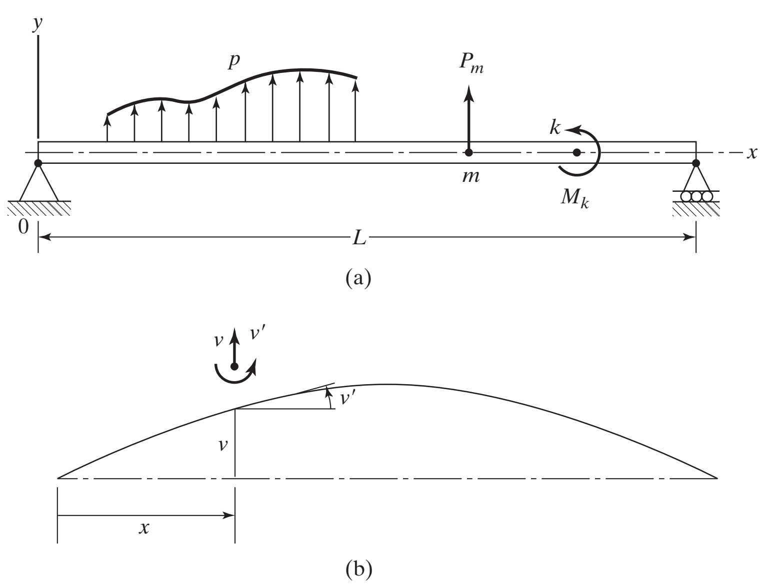


FIGURE 5.1 (a) Beam loading and (b) deformation of the neutral axis.

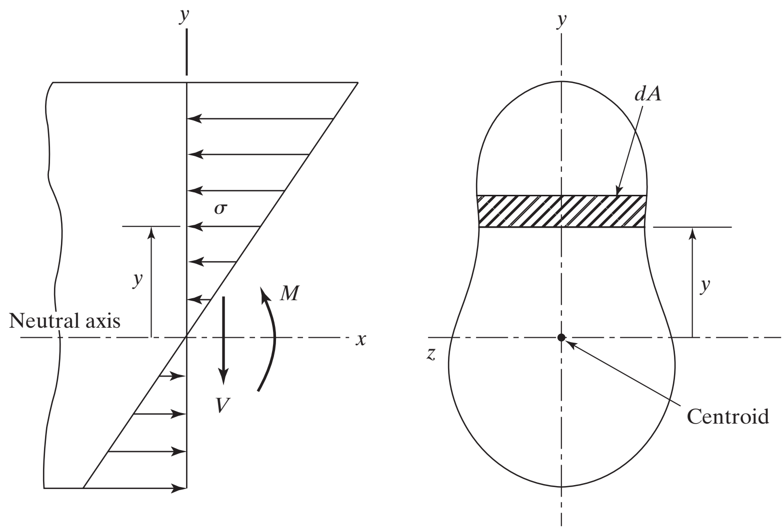


FIGURE 5.2 Beam section and stress distribution.

cross section and the bending stress distribution. For small deflections, we recall from elementary beam theory that

$$\sigma = -\frac{M}{I}y \tag{5.1}$$

$$\epsilon = \frac{\sigma}{E} \tag{5.2}$$

$$\frac{d^2v}{dx^2} = \frac{M}{EI} \tag{5.3}$$

where σ is the normal stress, ϵ is the normal strain, M is the bending moment at the section, v is the deflection of the centroidal axis at x, and I is the moment of inertia of the section about the neutral axis (z-axis passing through the centroid).

Potential Energy Approach

The strain energy in an element of length dx is

$$dU = \frac{1}{2}\int_A \sigma\epsilon\, dA\, dx$$

$$= \frac{1}{2}\left(\frac{M^2}{EI^2}\int_A y^2\, dA\right) dx$$

Noting that $\int_A y^2 dA$ is the moment of inertia I, we have

$$dU = \frac{1}{2}\frac{M^2}{EI}dx \tag{5.4}$$

When Eq. 5.3 is used, the total strain energy in the beam is given by

$$U = \frac{1}{2}\int_0^L EI\left(\frac{d^2v}{dx^2}\right)^2 dx \tag{5.5}$$

The potential energy of the beam is then given by

$$\Pi = \frac{1}{2}\int_0^L EI\left(\frac{d^2v}{dx^2}\right)^2 dx - \int_0^L pv\,dx - \sum_m P_m v_m - \sum_k M_k v'_k \tag{5.6}$$

where p is the distributed load per unit length, P_m is the point load at point m, M_k is the moment of the couple applied at point k, v_m is the deflection at point m, and v'_k is the slope at point k.

Galerkin Approach

For the Galerkin formulation, we start from equilibrium of an elemental length. From Fig. 5.3, we recall that

$$\frac{dV}{dx} = p \tag{5.7}$$

$$\frac{dM}{dx} = V \tag{5.8}$$

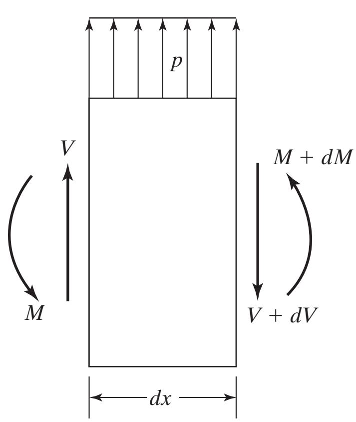


FIGURE 5.3 Free body diagram of an elemental length dx.

When Eqs. 5.3, 5.7, and 5.8 are combined, the equilibrium equation is given by

$$\frac{d^2}{dx^2}\left(EI\frac{d^2v}{dx^2}\right) - p = 0 \tag{5.9}$$

For approximate solution by the Galerkin approach, we look for the approximate solution v constructed using finite element shape functions such that

$$\int_0^L \left[\frac{d}{dx^2}\left(EI\frac{d^2v}{dx^2}\right) - p\right]\phi\, dx = 0 \tag{5.10}$$

where ϕ is an arbitrary function using same basis functions as v. Note that ϕ is zero where v has a specified value. We integrate the first term of Eq. 5.10 by parts. The integral from 0 to L is split into intervals: 0 to x_m, x_m to x_k, and x_k to L. We obtain

$$\begin{aligned}\int_0^L EI\frac{d^2v}{dx^2}\frac{d^2\phi}{dx^2}dx - \int_0^L p\phi\, dx + \frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)\phi\Big|_0^{x_m} + \frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)\phi\Big|_{x_m}^{L} \\ - EI\frac{d^2v}{dx^2}\frac{d\phi}{dx}\Big|_0^{x_k} - EI\frac{d^2v}{dx^2}\frac{d\phi}{dx}\Big|_{x_k}^{L} = 0\end{aligned} \tag{5.11}$$

We note that $EI(d^2v/dx^2)$ equals the bending moment M from Eq. 5.3 and $(d/dx)[EI(d^2v/dx^2)]$ equals the shear force V from Eq. 5.8. Also, ϕ and M are zero at the supports. At x_m, the jump in shear force is P_m and at x_k, the jump in bending moment is $-M_k$. Thus, we get

$$\int_0^L EI\frac{d^2v}{dx^2}\frac{d^2\phi}{dx^2}dx - \int_0^L p\phi\, dx - \sum_m P_m\phi_m - \sum_k M_k\phi'_k = 0 \tag{5.12}$$

For the finite element formulation based on Galerkin approach, v and ϕ are constructed using the same shape functions. Equation 5.12 is precisely the statement of the principle of virtual work.

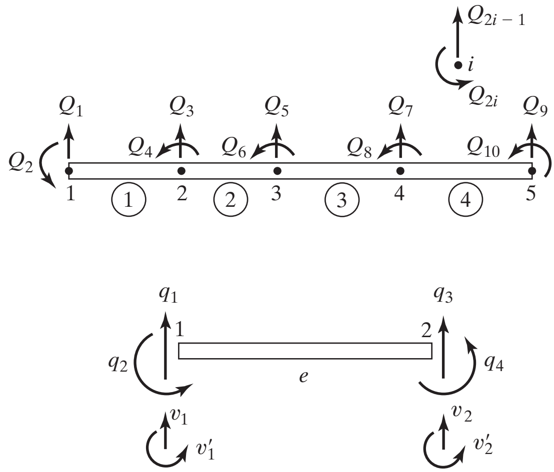
5.2 FINITE ELEMENT FORMULATION

The beam is divided into elements, as shown in Fig. 5.4. Each node has two degrees of freedom (dof). Typically, the dof of node i are Q_{2i-1} and Q_{2i}. The dof Q_{2i-1} is transverse displacement and Q_{2i} is slope or rotation. The vector

$$\mathbf{Q} = [Q_1, Q_2, \ldots, Q_{10}]^{\mathrm{T}} \tag{5.13}$$

represents the global displacement vector. For a single element, the local dof are represented by

$$\mathbf{q} = [q_1, q_2, q_3, q_4]^{\mathrm{T}} \tag{5.14}$$



	Local	
e	1	2
1	1	2
2	2	3
3	3	4
4	4	5

Global

FIGURE 5.4 Finite element discretization.

The local–global correspondence is easy to see from the table given in Fig. 5.4. $\mathbf{q}$ is same as $[v_1, v'_1, v_2, v'_2]^{\mathrm{T}}$. The shape functions for interpolating v on an element are defined in terms of ξ on -1 to $+1$ as shown in Fig. 5.5. *The shape functions for beam elements differ from those discussed earlier.* Since nodal values and nodal slopes are involved, we define Hermite shape functions, which satisfy nodal value and slope continuity requirements. Each of the shape functions is of cubic order represented by

$$H_i = a_i + b_i\xi + c_i\xi^2 + d_i\xi^3, \qquad i = 1, 2, 3, 4 \tag{5.15}$$

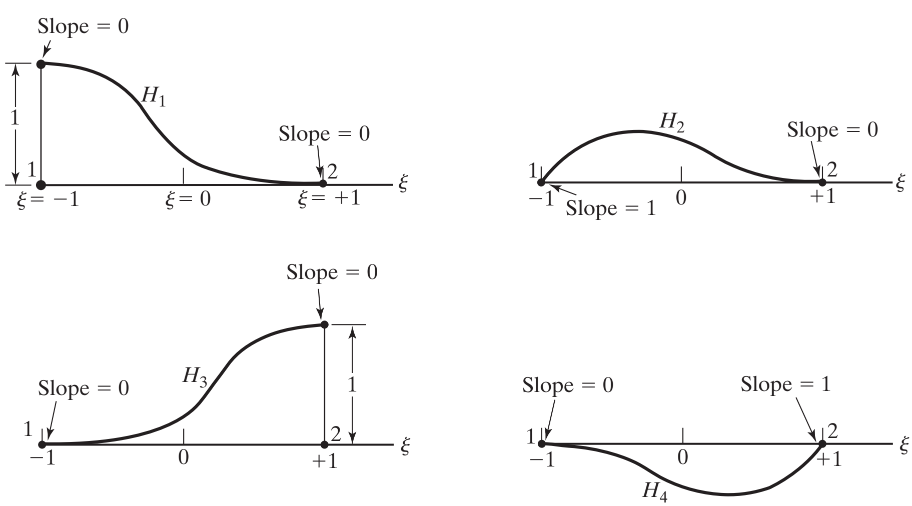


FIGURE 5.5 Hermite shape functions.

The conditions given in the following table must be satisfied:

	H_1	H'_1	H_2	H'_2	H_3	H'_3	H_4	H'_4
$\xi = -1$	1	0	0	1	0	0	0	0
$\xi = 1$	0	0	0	0	1	0	0	1

The coefficients $a_i, b_i, c_i,$ and d_i can be easily obtained by imposing these conditions. Thus,

$$\begin{aligned} H_1 &= \tfrac{1}{4}(1-\xi)^2(2+\xi) \quad \text{or} \quad \tfrac{1}{4}(2-3\xi+\xi^3) \\ H_2 &= \tfrac{1}{4}(1-\xi)^2(\xi+1) \quad \text{or} \quad \tfrac{1}{4}(1-\xi-\xi^2+\xi^3) \\ H_3 &= \tfrac{1}{4}(1+\xi)^2(2-\xi) \quad \text{or} \quad \tfrac{1}{4}(2+3\xi-\xi^3) \\ H_4 &= \tfrac{1}{4}(1+\xi)^2(\xi-1) \quad \text{or} \quad \tfrac{1}{4}(-1-\xi+\xi^2+\xi^3) \end{aligned} \tag{5.16}$$

The Hermite shape functions can be used to write v in the form

$$v(\xi) = H_1 v_1 + H_2\left(\frac{dv}{d\xi}\right)_1 + H_3 v_2 + H_4\left(\frac{dv}{d\xi}\right)_2 \tag{5.17}$$

The coordinates transform by the relationship

$$\begin{aligned} x &= \frac{1-\xi}{2}x_1 + \frac{1+\xi}{2}x_2 \\ &= \frac{x_1+x_2}{2} + \frac{x_2-x_1}{2}\xi \end{aligned} \tag{5.18}$$

Since $\ell_e = x_2 - x_1$ is the length of the element, we have

$$dx = \frac{\ell_e}{2}d\xi \tag{5.19}$$

The chain rule $dv/d\xi = (dv/dx)(dx/d\xi)$ gives us

$$\frac{dv}{d\xi} = \frac{\ell_e}{2}\frac{dv}{dx} \tag{5.20}$$

Noting that dv/dx evaluated at nodes 1 and 2 is q_2 and q_4, respectively, we have

$$v(\xi) = H_1 q_1 + \frac{\ell_e}{2}H_2 q_2 + H_3 q_3 + \frac{\ell_e}{2}H_4 q_4 \tag{5.21}$$

which may be denoted as

$$v = \mathbf{Hq} \tag{5.22}$$

where

$$\mathbf{H} = \left[H_1, \frac{\ell_e}{2}H_2, H_3, \frac{\ell_e}{2}H_4\right] \tag{5.23}$$

In the total potential energy of the system, we consider the integrals as summations over the integrals over the elements. The element strain energy is given by

$$U_e = \tfrac{1}{2} EI \int_e \left(\frac{d^2 v}{dx^2} \right)^2 dx \tag{5.24}$$

From Eq. 5.20,

$$\frac{dv}{dx} = \frac{2}{\ell_e} \frac{dv}{d\xi} \quad \text{and} \quad \frac{d^2 v}{dx^2} = \frac{4}{\ell_e^2} \frac{d^2 v}{d\xi^2}$$

Then, substituting $v = \mathbf{Hq}$, we obtain

$$\left(\frac{d^2 v}{dx} \right)^2 = \mathbf{q}^{\mathrm{T}} \frac{16}{\ell_e^4} \left(\frac{d^2 \mathbf{H}}{d\xi^2} \right)^{\mathrm{T}} \left(\frac{d^2 \mathbf{H}}{d\xi^2} \right) \mathbf{q} \tag{5.25}$$

$$\left(\frac{d^2 \mathbf{H}}{d\xi^2} \right) = \left[\frac{3}{2}\xi, \frac{-1 + 3\xi}{2} \frac{\ell_e}{2}, -\frac{3}{2}\xi, \frac{1 + 3\xi}{2} \frac{\ell_e}{2} \right] \tag{5.26}$$

On substituting $dx = (\ell_e/2)d\xi$ and Eqs. 5.25 and 5.26 in Eq. 5.24, we get

$$U_e = \frac{1}{2} \mathbf{q}^{\mathrm{T}} \frac{8EI}{\ell_e^3} \int_{-1}^{+1} \begin{bmatrix} \frac{9}{4}\xi^2 & \frac{3}{8}\xi(-1 + 3\xi)\ell_e & -\frac{9}{4}\xi^2 & \frac{3}{8}\xi(1 + 3\xi)\ell_e \\ & \left(\frac{-1 + 3\xi}{4} \right)^2 \ell_e^2 & -\frac{3}{8}\xi(-1 + 3\xi)\ell_e & \frac{-1 + 9\xi^2}{16} \ell_e^2 \\ & \text{Symmetric} & \frac{9}{4}\xi^2 & -\frac{3}{8}\xi(1 + 3\xi)\ell_e \\ & & & \left(\frac{1 + 3\xi}{4} \right)^2 \ell_e^2 \end{bmatrix} d\xi \, \mathbf{q} \tag{5.27}$$

Each term in the matrix needs to be integrated. Note that

$$\int_{-1}^{+1} \xi^2 \, d\xi = \frac{2}{3} \qquad \int_{-1}^{+1} \xi \, d\xi = 0 \qquad \int_{-1}^{+1} d\xi = 2$$

This results in the element strain energy given by

$$U_e = \frac{1}{2} \mathbf{q}^{\mathrm{T}} \mathbf{k}^e \mathbf{q} \tag{5.28}$$

where the element stiffness matrix is

$$\mathbf{k}^e = \frac{EI}{\ell_e^3} \begin{bmatrix} 12 & 6\ell_e & -12 & 6\ell_e \\ 6\ell_e & 4\ell_e^2 & -6\ell_e & 2\ell_e^2 \\ -12 & -6\ell_e & 12 & -6\ell_e \\ 6\ell_e & 2\ell_e^2 & -6\ell_e & 4\ell_e^2 \end{bmatrix} \tag{5.29}$$

which is symmetric.

In the development based on Galerkin approach (see Eq. 5.12), we note that

$$EI\frac{d^2\phi}{dx^2}\frac{d^2v}{dx^2} = \boldsymbol{\psi}^{\mathrm{T}}EI\frac{16}{\ell_e^4}\left(\frac{d^2\mathbf{H}}{d\xi^2}\right)^{\mathrm{T}}\left(\frac{d^2\mathbf{H}}{d\xi^2}\right)\mathbf{q} \tag{5.30}$$

where

$$\boldsymbol{\psi} = [\psi_1 \;\; \psi_2 \;\; \psi_3 \;\; \psi_4]^{\mathrm{T}} \tag{5.31}$$

is the set of generalized virtual displacements on the element, $v = \mathbf{Hq}$, and $\phi = \mathbf{H}\boldsymbol{\psi}$. Equation 5.30 yields the same element stiffness as Eq. 5.28 on integration, with $\psi^{\mathrm{T}}\mathbf{k}^e\mathbf{q}$ being the internal virtual work in an element.

Element Stiffness—Direct Approach

We present here the direct stiffness approach in deriving the element stiffness (Eq. 5.29) using relations learnt in the first course in solid mechanics. Figure 5.6 shows deflection v and slope v' relations for a fixed beam of length ℓ (subscript e is avoided in this derivation) with end load P and end moment M.

$$\text{End load } P \qquad v = \frac{P\ell^3}{3EI} \quad v' = \frac{P\ell^2}{2EI} \tag{5.32}$$

$$\text{End moment } M \qquad v = \frac{M\ell^2}{2EI} \quad v' = \frac{M\ell}{EI} \tag{5.33}$$

Using these relations, the relative deformations of the node 2 of the element shown in Fig. 5.7 can be written as

$$q_3 - q_1 - q_2\ell = \frac{f_3\ell^3}{3EI} + \frac{f_4\ell^2}{2EI} \tag{5.34}$$

$$q_4 - q_2 = \frac{f_3\ell^2}{2EI} + \frac{f_4\ell}{EI} \tag{5.35}$$

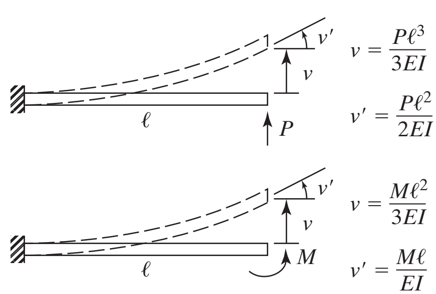


FIGURE 5.6 Slope-deflection relations.

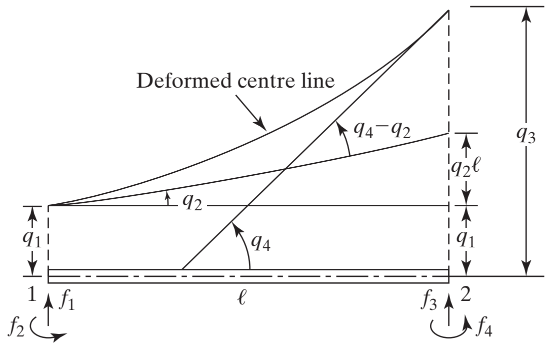


FIGURE 5.7 Element deformed configuration.

By adding 2 × Eq. 5.34 to –l × Eq. 5.35, we get

$$\frac{EI}{\ell^3}(-12q_1 - 6\ell q_2 + 12q_3 - 6\ell q_4) = f_3 \tag{5.36}$$

Similarly, adding –3 × Eq. 5.34 to 2 ℓ × Eq. 5.35, we get

$$\frac{EI}{\ell^2}(6q_1 + 2\ell q_2 - 6q_3 + 4\ell q_4) = f_4 \tag{5.37}$$

Equilibrium of the element shown in Fig. 5.7 gives the relations

$$\begin{aligned} f_1 &= -f_3 \\ f_2 &= -f_4 - \ell f_3 \end{aligned} \tag{5.38}$$

Equations 5.36 to 5.38 yield the relation

$$\mathbf{k}^e\mathbf{q} = \mathbf{f} \tag{5.39}$$

$\mathbf{k}^e$ is the element stiffness matrix given in Eq. 5.29.

5.3 LOAD VECTOR

The load contributions from the distributed load p in the element are first considered. We assume that the distributed load is uniform over the element:

$$\int_{\ell_e} pv\,dx = \left(\frac{p\ell_e}{2}\int_{-1}^{1}\mathbf{H}\,d\xi\right)\mathbf{q} \tag{5.40}$$

On substituting for $\mathbf{H}$ from Eqs. 5.16 and 5.23 and integrating, we obtain

$$\int_{l_e} pv\,dx = \mathbf{f}^{e\mathrm{T}}\mathbf{q} \tag{5.41}$$

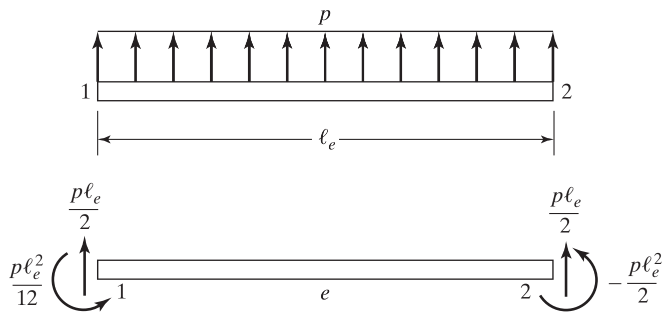


FIGURE 5.8 Distributed load on an element.

where

$$\boxed{\mathbf{f}^e = \left[\frac{p\ell_e}{2}, \frac{p\ell_e^2}{12}, \frac{p\ell_e}{2}, \frac{-p\ell_e^2}{12}\right]^{\mathrm{T}}} \tag{5.42}$$

This equivalent load on an element is shown in Fig. 5.8. The same result is obtained by considering the term $\int_e p\phi\, dx$ in Eq. 5.12 for the Galerkin formulation. For linearly distributed load, see problem 5.15. The point loads P_m and M_k are readily taken care of by introducing nodes at the points of application. On introducing the local–global correspondence, from the potential energy approach, we get

$$\Pi = \tfrac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{KQ} - \mathbf{Q}^{\mathrm{T}}\mathbf{F} \tag{5.43}$$

and from Galerkin approach, we get

$$\mathbf{\Psi}^{\mathrm{T}}\mathbf{KQ} - \mathbf{\Psi}^{\mathrm{T}}\mathbf{F} = 0 \tag{5.44}$$

where $\mathbf{\Psi}$ = arbitrary admissible global virtual displacement vector.

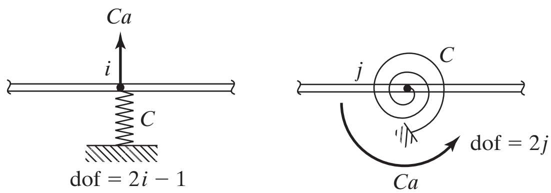
5.4 BOUNDARY CONSIDERATIONS

When the generalized displacement value is specified as a for the dof r, we follow the penalty approach and add $\frac{1}{2}C(Q_r - a)^2$ to Π and $\mathbf{\Psi}_j C(Q_r - a)$ to the left-hand side of the Galerkin formulation and place no restrictions on the dof. The number C represents stiffness and is large in comparison with beam stiffness terms. This amounts to adding stiffness C to K_{rr} and load Ca to F_r (see Fig. 5.9). Both Eqs. 5.43 and 5.44 independently yield

$$\mathbf{KQ} = \mathbf{F} \tag{5.45}$$

These equations are now solved to get the nodal displacements.

Reactions at constrained dof may be calculated using Eq. 3.74 or 3.78.



a = known generalized displacement

FIGURE 5.9 Boundary conditions for a beam.

5.5 SHEAR FORCE AND BENDING MOMENT

Using the bending moment and shear force equations

$$M = EI\frac{d^2v}{dx^2} \quad V = \frac{dM}{dx} \quad \text{and} \quad v = \mathbf{Hq}$$

we get the element bending moment and shear force:

$$M = \frac{EI}{\ell_e^2}\left[6\xi q_1 + (3\xi - 1)\ell_e q_2 - 6\xi q_3 + (3\xi + 1)\ell_e q_4\right] \tag{5.46}$$

$$V = \frac{6EI}{\ell_e^3}(2q_1 + \ell_e q_2 - 2q_3 + \ell_e q_4) \tag{5.47}$$

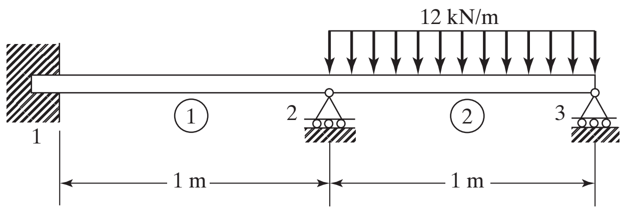
These bending moment and shear force values are for the loading as modeled using equivalent point loads. Denoting element end equilibrium loads as R_1, R_2, R_3, and R_4, we note that

$$\begin{Bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{Bmatrix} = \frac{EI}{\ell_e^3}\begin{bmatrix} 12 & 6\ell_e & -12 & 6\ell_e \\ 6\ell_e & 4\ell_e^2 & -6\ell_e & 2\ell_e^2 \\ -12 & -6\ell_e & 12 & -6\ell_e \\ 6\ell_e & 2\ell_e^2 & -6\ell_e & 4\ell_e^2 \end{bmatrix}\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} + \begin{Bmatrix} \dfrac{-p\ell_e}{2} \\ \dfrac{-p\ell_e^2}{12} \\ \dfrac{-p\ell_e}{2} \\ \dfrac{p\ell_e^2}{12} \end{Bmatrix} \tag{5.48}$$

It is easily seen that the first term on the right-hand side is $\mathbf{k}^e\mathbf{q}$. Also note that the second term needs to be added only on elements with distributed load. In books on matrix structural analysis, the previous equations are written directly from element equilibrium. Also, the last vector on the right-hand side of the equation consists of terms that are called *fixed-end reactions*. The shear forces at the two ends of the element are $V_1 = R_1$ and $V_2 = -R_3$. The end bending moments are $M_1 = -R_2$ and $M_2 = R_4$.

Example 5.1

For the beam and loading shown in Fig. E5.1, determine (1) the slopes at 2 and 3 and (2) the vertical deflection at the midpoint of the distributed load.



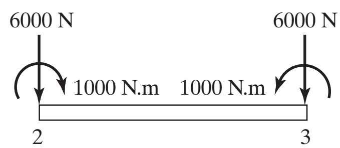


FIGURE E5.1

Solution We consider the two elements formed by the three nodes. Displacements Q_1, Q_2, Q_3, and Q_5 are constrained to be zero, and Q_4 and Q_6 need to be found. Since the lengths and sections are equal, the element matrices are calculated from Eq. 5.29 as follows:

$$\frac{EI}{\ell^3} = \frac{(200 \times 10^9)(4 \times 10^{-6})}{1^3} = 8 \times 10^5 \mathrm{N/m}$$

$$\mathbf{k}^1 = \mathbf{k}^2 = 8 \times 10^5 \begin{bmatrix} 12 & 6 & -12 & 6 \\ 6 & 4 & -6 & 2 \\ -12 & -6 & 12 & -6 \\ 6 & 2 & -6 & 4 \end{bmatrix}$$

$$\begin{array}{llcccc} e = 1 & & Q_1 & Q_2 & Q_3 & Q_4 \\ e = 2 & & Q_3 & Q_4 & Q_5 & Q_6 \end{array}$$

We note that global applied loads are $F_4 = -1000$ N·m and $F_6 = +1000$ N·m obtained from $p\ell^2/12$, as seen in Fig. 5.8. We use here the elimination approach presented in Chapter 3. Using the connectivity, we obtain the global stiffness after elimination:

$$\mathbf{K} = \begin{bmatrix} k_{44}^{(1)} + k_{22}^{(2)} & k_{24}^{(2)} \\ k_{42}^{(2)} & k_{44}^{(2)} \end{bmatrix}$$

$$= 8 \times 10^5 \begin{bmatrix} 8 & 2 \\ 2 & 4 \end{bmatrix}$$

The set of equations is given by

$$8 \times 10^5 \begin{bmatrix} 8 & 2 \\ 2 & 4 \end{bmatrix} \begin{Bmatrix} Q_4 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} -1000 \\ +1000 \end{Bmatrix}$$

The solution is

$$\begin{Bmatrix} Q_4 \\ Q_6 \end{Bmatrix} = \begin{Bmatrix} -2.679 \times 10^{-4} \\ 4.464 \times 10^{-4} \end{Bmatrix}$$

For element 2, $q_1 = 0, q_2 = Q_4, q_3 = 0$, and $q_4 = Q_6$. To get vertical deflection at the midpoint of the element, use $v = \mathbf{Hq}$ at $\xi = 0$:

$$\begin{aligned} v &= 0 + \frac{\ell_e}{2} H_2 Q_4 + 0 + \frac{\ell_e}{2} H_4 Q_6 \\ &= \left(\frac{1}{2}\right)\left(\frac{1}{4}\right)(-2.679 \times 10^{-4}) + \left(\frac{1}{2}\right)\left(-\frac{1}{4}\right)(4.464 \times 10^{-4}) \\ &= -8.93 \times 10^{-5}\,\text{m} \\ &= -0.0893\,\text{mm} \end{aligned}$$

■

5.6 BEAMS ON ELASTIC SUPPORTS

In many engineering applications, beams are supported on elastic members. Shafts are supported on ball, roller, or journal bearings. Large beams are supported on elastic walls. Beams supported on soil form a class of applications known as Winkler foundations.

Single-row ball bearings can be considered by having a node at each bearing location and adding the bearing stiffness k_B to the diagonal location of vertical dof (Fig. 5.10a). Rotational (moment) stiffness has to be considered for roller bearings and journal bearings.

In wide journal bearings and Winkler foundations, we use stiffness per unit length, s, of the supporting medium (Fig. 5.10b). Over the length of the support, this adds the following term to the total potential energy:

$$\frac{1}{2}\int_0^{\ell} sv^2\,dx \tag{5.49}$$

In Galerkin approach, this term is $\int_0^{\ell} sv\phi\,dx$. When we substitute $v = \mathbf{Hq}$ in the discretized model, the previous term becomes

$$\frac{1}{2}\sum_e \mathbf{q}^{\mathrm{T}} s \int_e \mathbf{H}^{\mathrm{T}}\mathbf{H}\,dx\,\mathbf{q} \tag{5.50}$$

We recognize the stiffness term in this summation, namely,

$$\mathbf{k}_s^e = s\int_e \mathbf{H}^{\mathrm{T}}\mathbf{H}\,dx = \frac{s\ell_e}{2}\int_{-1}^{+1} \mathbf{H}^{\mathrm{T}}\mathbf{H}\,d\xi \tag{5.51}$$

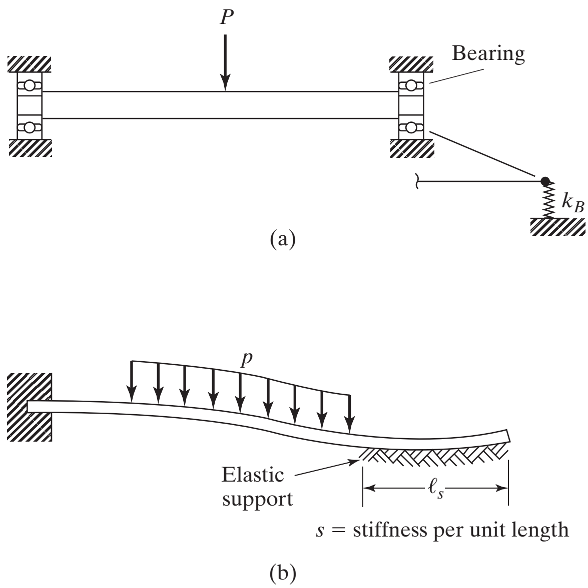


FIGURE 5.10 Elastic support.

On integration, we get

$$\mathbf{k}_s^e = \frac{s\ell_e}{420}\begin{bmatrix} 156 & 22\ell_e & 54 & -13\ell_e \\ 22\ell_e & 4\ell_e^2 & 13\ell_e & -3\ell_e^2 \\ 54 & 13\ell_e & 156 & -22\ell_e \\ -13\ell_e & -3\ell_e^2 & -22\ell_e & 4\ell_e^2 \end{bmatrix} \tag{5.52}$$

For elements supported on an elastic foundation, this stiffness has to be added to the element stiffness given by Eq. 5.29. Matrix $\mathbf{k}_s^e$ is the consistent stiffness matrix for the elastic foundation.

5.7 PLANE FRAMES

Here, we consider plane structures with rigidly connected members. These members will be similar to the beams except that axial loads and axial deformations are present. The elements also have different orientations. Figure 5.11 shows a frame element. We have two displacements and a rotational deformation for each node. The nodal displacement vector is given by

$$\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^{\mathrm{T}} \tag{5.53}$$

We also define the local or body coordinate system x', y', such that x' is oriented along 1 to 2, with direction cosines ℓ, m (where $\ell = \cos\theta$, $m = \sin\theta$). These are evaluated

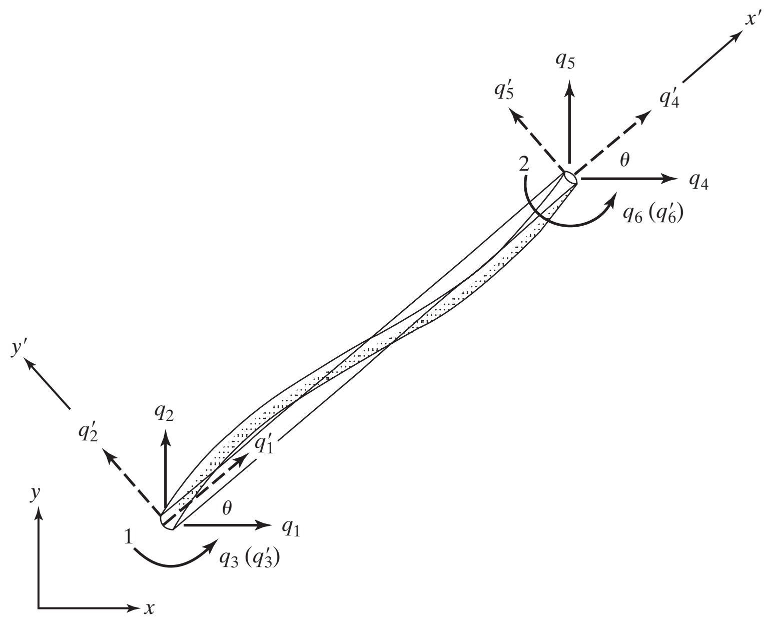


FIGURE 5.11 Frame element.

using relationships given for the truss element, shown in Fig. 4.4. The nodal displacement vector in the local system is

$$\mathbf{q}' = [q'_1, q'_2, q'_3, q'_4, q'_5, q'_6]^{\mathrm{T}} \tag{5.54}$$

Recognizing that $q'_3 = q_3$ and $q'_6 = q_6$, which are rotations with respect to the body, we obtain the local–global transformation

$$\mathbf{q}' = \mathbf{L}\mathbf{q} \tag{5.55}$$

where

$$\mathbf{L} = \begin{bmatrix} \ell & m & 0 & 0 & 0 & 0 \\ -m & \ell & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ell & m & 0 \\ 0 & 0 & 0 & -m & \ell & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.56}$$

It is now observed that q'_2, q'_3, q'_5, and q'_6 are similar to the beam dof, while q'_1 and q'_4 are similar to the displacements of a rod element, as discussed in Chapter 3. Combining the two stiffnesses and arranging in proper locations, we get the element stiffness for a frame element as

$$\mathbf{k}'^{e} = \begin{bmatrix} \frac{EA}{\ell_e} & 0 & 0 & \frac{-EA}{\ell_e} & 0 & 0 \\ 0 & \frac{12EI}{\ell_e^3} & \frac{6EI}{\ell_e^2} & 0 & \frac{-12EI}{\ell_e^3} & \frac{6EI}{\ell_e^2} \\ 0 & \frac{6EI}{\ell_e^2} & \frac{4EI}{\ell_e} & 0 & \frac{-6EI}{\ell_e^2} & \frac{2EI}{\ell_e} \\ \frac{-EA}{\ell_e} & 0 & 0 & \frac{EA}{\ell_e} & 0 & 0 \\ 0 & \frac{-12EI}{\ell_e^3} & \frac{-6EI}{\ell_e^2} & 0 & \frac{12EI}{\ell_e^3} & \frac{-6EI}{\ell_e^2} \\ 0 & \frac{6EI}{\ell_e^2} & \frac{2EI}{\ell_e} & 0 & \frac{-6EI}{\ell_e^2} & \frac{4EI}{\ell_e} \end{bmatrix} \tag{5.57}$$

As discussed in the development of a truss element in Chapter 4, we recognize that the element strain energy is given by

$$U_e = \tfrac{1}{2}\mathbf{q}'^{\mathrm{T}}\mathbf{k}'^{e}\mathbf{q}' = \tfrac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{k}'^{e}\mathbf{L}\mathbf{q} \tag{5.58}$$

or in Galerkin approach, the internal virtual work of an element is

$$W_e = \mathbf{\Psi}'^{\mathrm{T}}\mathbf{k}'^{e}\mathbf{q}' = \mathbf{\Psi}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{k}'^{e}\mathbf{L}\mathbf{q} \tag{5.59}$$

where $\mathbf{\Psi}'$ and $\mathbf{\Psi}$ are virtual nodal displacements in local and global coordinate systems, respectively. From Eq. 5.58 or 5.59, we recognize the element stiffness matrix in global coordinates to be

$$\boxed{\mathbf{k}^{e} = \mathbf{L}^{\mathrm{T}}\mathbf{k}'^{e}\mathbf{L}} \tag{5.60}$$

In the finite element program implementation, $\mathbf{k}'^{e}$ can first be defined, and then this matrix multiplication can be carried out.

If there is distributed load on a member, as shown in Fig. 5.12, we have

$$\mathbf{k}'^{\mathrm{T}}\mathbf{f}' = \mathbf{q}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{f}' \tag{5.61}$$

where

$$\mathbf{f}' = \left[0,\ \frac{p\ell_e}{2},\ \frac{p\ell_e^2}{12},\ 0,\ \frac{p\ell_e}{2},\ -\frac{p\ell_e^2}{12}\right]^{\mathrm{T}} \tag{5.62}$$

The nodal loads due to the distributed load p are given by

$$\mathbf{f} = \mathbf{L}^{\mathrm{T}}\mathbf{f}' \tag{5.63}$$

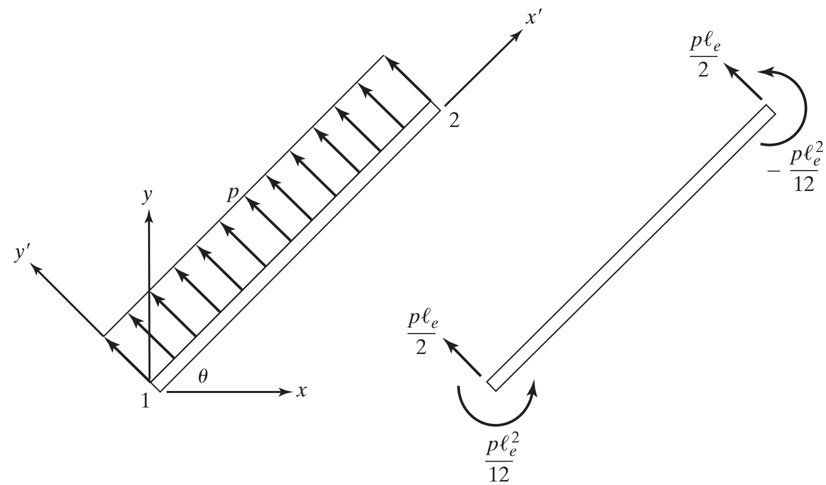


FIGURE 5.12 Distributed load on a frame element.

The values of **f** are added to the global load vector. Note that here positive p is in the y' direction. Positive p is along $\mathbf{z} \times \mathbf{x}'$.

The point loads and couples are simply added to the global load vector. On gathering stiffnesses and loads, we get the system of equations

$$\mathbf{KQ} = \mathbf{F}$$

where the boundary conditions are considered by applying the penalty terms in the energy or Galerkin formulations.

Example 5.2

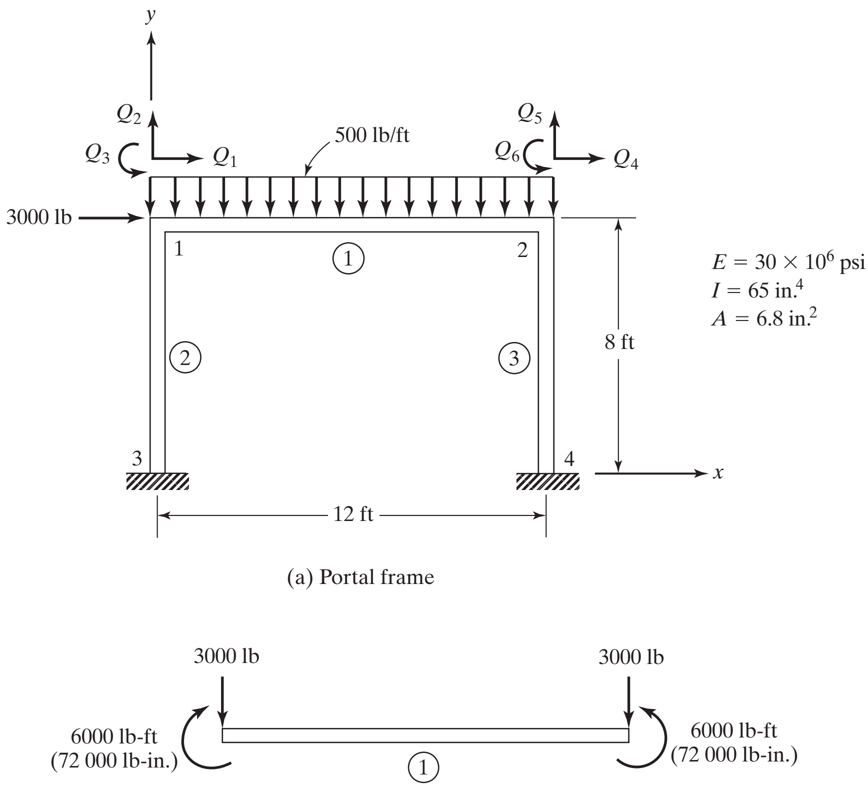
Determine the displacements and rotations of the joints for the portal frame shown in Fig. E5.2.

Solution We follow the steps given below:

Step 1. Connectivity

The connectivity is as follows:

	Node	
Element no.	1	2
1	1	2
2	3	1
3	4	2



(a) Portal frame

(b) Equivalent load for element 1

FIGURE E5.2 (a) Portal frame. (b) Equivalent load for element 1.

Step 2. Element Stiffnesses

Element 1. Using the matrix given in Eq. 5.57 and noting that $\mathbf{k}^1 = \mathbf{k}'^1$, we find that

$$\mathbf{k}^1 = 10^4 \times \begin{array}{c} \begin{array}{cccccc} Q_1 & Q_2 & Q_3 & Q_4 & Q_5 & Q_6 \end{array} \\ \begin{bmatrix} 141.7 & 0 & 0 & -141.7 & 0 & 0 \\ 0 & 0.784 & 56.4 & 0 & -0.784 & 56.4 \\ 0 & 56.4 & 5417 & 0 & -56.4 & 2708 \\ -141.7 & 0 & 0 & 141.7 & 0 & 0 \\ 0 & -0.784 & -56.4 & 0 & 0.784 & -56.4 \\ 0 & 56.4 & 2708 & 0 & -56.4 & 5417 \end{bmatrix} \end{array}$$

Elements 2 and 3. Local element stiffnesses for elements 2 and 3 are obtained by substituting for E, A, I and ℓ_2 in matrix $\mathbf{k}'$ of Eq. 5.57:

$$\mathbf{k}'^{2} = 10^4 \times \begin{bmatrix} 212.5 & 0 & 0 & -212.5 & 0 & 0 \\ 0 & 2.65 & 127 & 0 & -2.65 & 127 \\ 0 & 127 & 8125 & 0 & -127 & 4063 \\ -212.5 & 0 & 0 & 212.5 & 0 & 0 \\ 0 & -2.65 & -127 & 0 & 2.65 & -127 \\ 0 & 127 & 4063 & 0 & -127 & 8125 \end{bmatrix}$$

Transformation matrix **L**. We have noted that for element 1, $\mathbf{k} = \mathbf{k}^1$. For elements 2 and 3, which are oriented similarly with respect to the x- and y-axes, we have $\ell = 0, m = 1$. Then,

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Noting that $\mathbf{k}^2 = \mathrm{L}^{\mathrm{T}}\, \mathbf{k}'^{2}\mathbf{k}$ and dof of interest for elements 3 and 2, we get

$$\begin{array}{llll} e = 3 & Q_4 & Q_5 & Q_6 \\ e = 2 \rightarrow & Q_1 & Q_2 & Q_3 \end{array}$$

$$\mathbf{k} = 10^4 \times \begin{bmatrix} 2.65 & 0 & -127 & -2.65 & 0 & -127 \\ 0 & 212.5 & 0 & 0 & -212.5 & 0 \\ -127 & 0 & 8125 & 127 & 0 & 4063 \\ -2.65 & 0 & 127 & 2.65 & 0 & 127 \\ 0 & -212.5 & 0 & 0 & 212.5 & 0 \\ -127 & 0 & 4063 & 127 & 0 & 8125 \end{bmatrix}$$

Stiffness $\mathbf{k}^1$ has all its elements in the global locations. For elements 2 and 3, the shaded part of the stiffness matrix shown previously is added to the appropriate global locations of **K**. The global stiffness matrix is given by

$$\mathbf{K} = 10^4 \times \begin{bmatrix} 144.3 & 0 & 127 & -141.7 & 0 & 0 \\ 0 & 213.3 & 56.4 & 0 & -0.784 & 56.4 \\ 127 & 56.4 & 13542 & 0 & -56.4 & 2708 \\ -141.7 & 0 & 0 & 144.3 & 0 & 127 \\ 0 & -0.784 & -56.4 & 0 & 213.3 & -56.4 \\ 0 & 56.4 & 2708 & 127 & -56.4 & 13542 \end{bmatrix}$$

From Fig. E5.2, the load vector can easily be written as

$$\mathbf{F} = \begin{Bmatrix} 3\,000 \\ -3\,000 \\ -72\,000 \\ 0 \\ -3\,000 \\ +72\,000 \end{Bmatrix}$$

The set of equations is given by

$$\mathbf{KQ} = \mathbf{F}$$

On solving, we get

$$\mathbf{Q} = \begin{Bmatrix} 0.092 \text{ in.} \\ -0.00104 \text{ in.} \\ -0.00139 \text{ rad} \\ 0.0901 \text{ in.} \\ -0.0018 \text{ in.} \\ -3.88 \times 10^{-5} \text{ rad} \end{Bmatrix}$$

■

5.8 THREE-DIMENSIONAL FRAMES

Three-dimensional frames, also called as *space frames*, are frequently encountered in the analysis of multistory buildings. They are also found in the modeling of car body and bicycle frames. A typical three-dimensional frame is shown in Fig. 5.13. Each node has

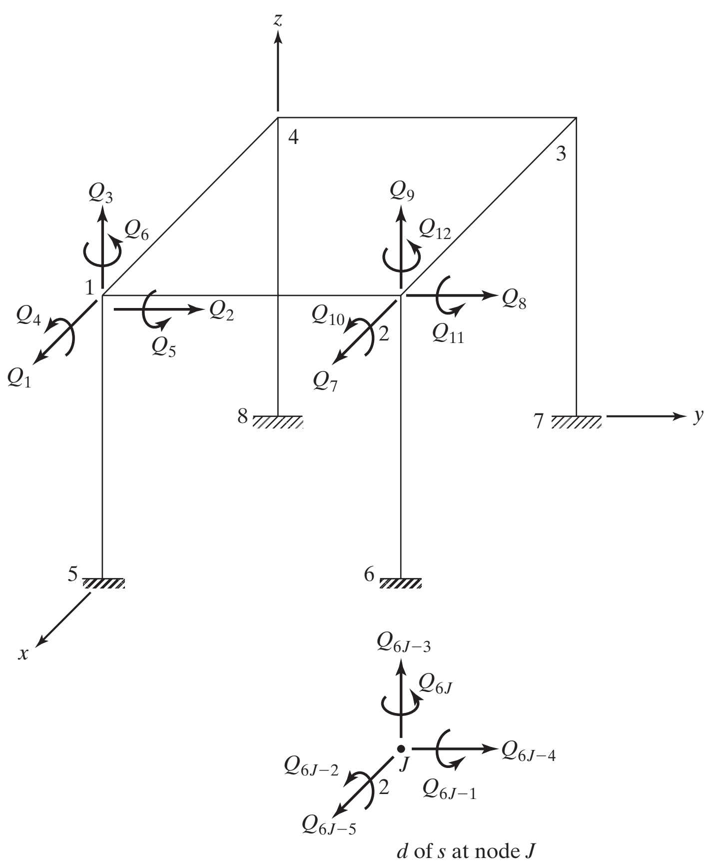
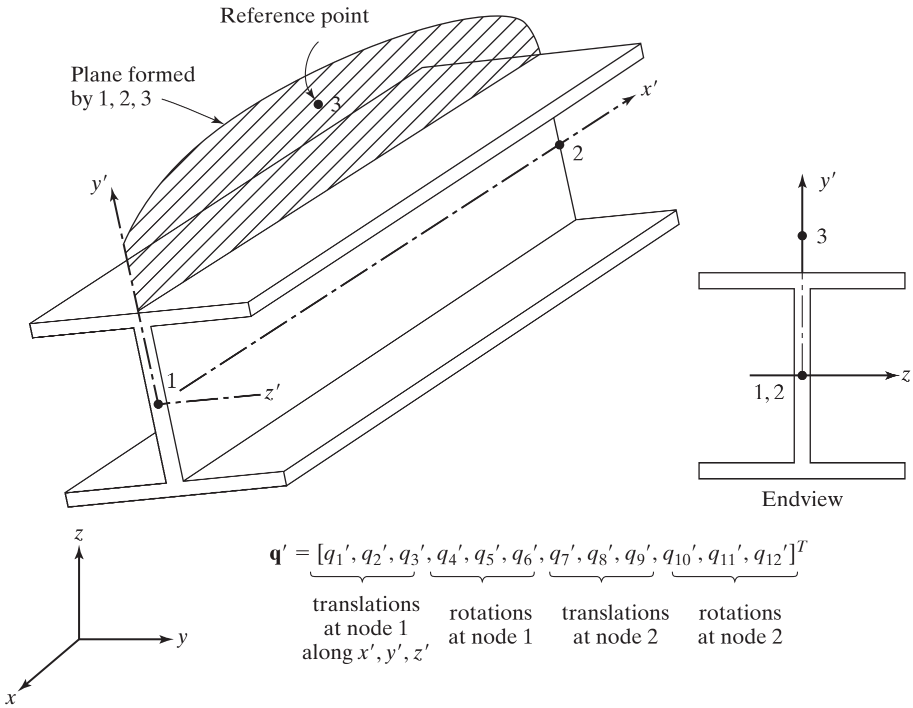


FIGURE 5.13 Degrees of freedom numbering for a three-dimensional frame.

six dof (as opposed to only three dof in a plane frame). The dof numbering is shown in Fig. 5.13: for node J, dof $6J$–5, $6J$–4, and $6J$–3 represent the x-, y-, and z-translational dofs, while $6J$–2, $6J$–1, and $6J$ represent the rotational dofs along the x-, y-, and z-axes. The element displacement vectors in the local and global coordinate systems are denoted as $\mathbf{q}'$ and $\mathbf{q}$, respectively. These vectors are of dimension (12×1) as shown in Fig. 5.14.

Orientation of the local x'-, y'-, and z'-coordinate system is established with the use of three points. Points 1 and 2 are the ends of the element; the x'-axis is along the line from point 1 to point 2, just as in the case of two-dimensional frames. Point 3 is any *reference point* not lying along the line joining points 1 and 2. The y'-axis is to lie in the plane defined by points 1, 2, and 3. This is shown in Fig. 5.14. The z'-axis is then automatically defined from the fact that x', y' and z' form a right-handed system. We note that y' and z' are the principal axes of the cross section, with $I_{y'}$ and $I_{z'}$ the principal moments of inertia. The cross-sectional properties are specified by four parameters: area A and moments of inertia $I_{y'}$, $I_{z'}$, and J. The product GJ is the torsional stiffness, where G is the shear modulus. For circular or tubular cross sections, J is the polar moment of inertia. For other cross-sectional shapes, such as an I section, the torsional stiffness is given in strength of materials texts.



$$\mathbf{q}' = [\underbrace{q_1', q_2', q_3'}_{\substack{\text{translations}\\ \text{at node 1}\\ \text{along } x', y', z'}}, \underbrace{q_4', q_5', q_6'}_{\substack{\text{rotations}\\ \text{at node 1}}}, \underbrace{q_7', q_8', q_9'}_{\substack{\text{translations}\\ \text{at node 2}}}, \underbrace{q_{10}', q_{11}', q_{12}'}_{\substack{\text{rotations}\\ \text{at node 2}}}]^T$$

$$\mathbf{q} = [q_1, q_2, \ldots, q_{12}]^T = \text{displacement vector in global } (x, y, z) \text{ system}$$

FIGURE 5.14 Three-dimensional frame element in local and global coordinate systems.

The (12 × 12) element stiffness matrix $\mathbf{k}'$ in the local coordinate system is obtained by a straightforward generalization of Eq. 5.49 as

$$
\mathbf{k}' = \left[\begin{array}{cccccccccccc}
AS & 0 & 0 & 0 & 0 & 0 & -AS & 0 & 0 & 0 & 0 & 0 \\
 & a_{z'} & 0 & 0 & 0 & b_{z'} & 0 & -a_{z'} & 0 & 0 & 0 & b_{z'} \\
 & & a_{y'} & 0 & -b_{y'} & 0 & 0 & 0 & -a_{y'} & 0 & -b_{y'} & 0 \\
 & & & TS & 0 & 0 & 0 & 0 & 0 & -TS & 0 & 0 \\
 & & & & c_{y'} & 0 & 0 & 0 & b_{y'} & 0 & d_{y'} & 0 \\
 & & & & & c_{z'} & 0 & -b_{z'} & 0 & 0 & 0 & d_{z'} \\
 & & & & & & AS & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & a_{z'} & 0 & 0 & 0 & -b_{z'} \\
 & & & & & & & & c_{y'} & 0 & b_{y'} & 0 \\
 & & & & & & & & & TS & 0 & 0 \\
\text{Symmetric} & & & & & & & & & & c_{y'} & 0 \\
 & & & & & & & & & & & c_{z'}
\end{array}\right] \tag{5.64}
$$

where $AS = EA/\ell_e$, ℓ_e = length of the element, $TS = GJ/\ell_e$, $a_{z'} = 12EI_{z'}/\ell_e^3$, $b_{z'} = 6EI_{z'}/\ell_e^3$, $c_{z'} = 4EI_{z'}/\ell_e$, $d_{z'} = 2EI_{z'}/\ell'_e$, $a_{y'} = 12EI_{y'}/\ell_e^3$ and so on. The global–local transformation matrix is given by

$$\mathbf{q}' = \mathbf{L}\mathbf{q} \tag{5.65}$$

The (12 × 12) transformation matrix $\mathbf{L}$ is defined from a (3 × 3)$\boldsymbol{\lambda}$ matrix as

$$
\mathbf{L} = \begin{bmatrix} \boldsymbol{\lambda} & & & \mathbf{0} \\ & \boldsymbol{\lambda} & & \\ & & \boldsymbol{\lambda} & \\ \mathbf{0} & & & \boldsymbol{\lambda} \end{bmatrix} \tag{5.66}
$$

The $\boldsymbol{\lambda}$ is a matrix of direction cosines:

$$
\boldsymbol{\lambda} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \tag{5.67}
$$

Here, l_1, m_1, and n_1 are the cosines of the angles between the x'-axis and the global x-, y-, and z-axes, respectively. Similarly, l_2, m_2, and n_2, are the cosines of the angles between the y'-axis and the x-, y-, and z-axes, and l_3, m_3, and n_3 are associated with the z'-axis. These direction cosines and hence the $\boldsymbol{\lambda}$ matrix are obtainable from the coordinates of the points 1, 2, and 3 as follows. We have

$$
l_1 = \frac{x_2 - x_1}{\ell_e} \quad m_1 = \frac{y_2 - y_1}{\ell_e} \quad n_1 = \frac{z_2 - z_1}{\ell_e}
$$

$$
l_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}
$$

Now, let $\mathbf{V}_{x'} = [l_1 \quad m_1 \quad n_1]^{\mathrm{T}}$ denote the unit vector along the x'-axis. Also, let

$$\mathbf{V}_{13} = \left[\frac{x_3 - x_1}{l_{13}} \; \frac{y_3 - y_1}{l_{13}} \; \frac{z_3 - z_1}{l_{13}}\right]$$

where l_{13} = distance between points 1 and 3. The unit vector along the z'-axis is now given by

$$\mathbf{V}_{z'} = [l_3 \;\; m_3 \;\; n_3]^{\mathrm{T}} = \frac{\mathbf{V}_{x'} \times \mathbf{V}_{13}}{|\mathbf{V}_{x'} \times \mathbf{V}_{13}|}$$

The cross product of any two vectors is given by the determinant

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{vmatrix} = \begin{vmatrix} u_y v_z - v_y u_z \\ v_x u_z - u_x v_z \\ u_x v_y - v_x u_y \end{vmatrix}$$

Finally, we have the direction cosines of the y'-axis given by

$$\mathbf{V}_{y'} = [l_2 \quad m_2 \quad n_2]^{\mathrm{T}} = \mathbf{V}_{z'} \times \mathbf{V}_{x'}$$

These calculations to define the $\mathbf{L}$ matrix are coded in program FRAME3D. The element stiffness matrix in global coordinates is

$$\mathbf{k} = \mathbf{L}^{\mathrm{T}}\mathbf{k}'\mathbf{L} \tag{5.68}$$

where $\mathbf{k}'$ has been defined in Eq. 5.64.

If a distributed load with components $w_{y'}$ and $w_{z'}$ (unit of force/unit length) is applied on the element, then the equivalent point loads at the end of the member are

$$\mathbf{f}' = \left[0, \frac{w_{y'}\ell_e}{2}, \frac{w_{z'}\ell_e}{2}, 0, \frac{-w_{z'}\ell_e^2}{12}, \frac{w_{y'}\ell_e^2}{12}, 0, \frac{w_{y'}\ell_e}{2}, \frac{w_{z'}\ell_e}{2}, 0, \frac{w_{z'}\ell_e^2}{12}, \frac{-w_{y'}\ell_e^2}{12}\right] \tag{5.69}$$

These loads are transferred into global components by $\mathbf{f} = \mathbf{L}^{\mathrm{T}}\mathbf{f}'$. After enforcing boundary conditions and solving the system of equations $\mathbf{KQ} = \mathbf{F}$, we can compute the member end forces from

$$\mathbf{R}' = \mathbf{k}'\mathbf{q}' + \text{ fixed-end reactions} \tag{5.70}$$

where the fixed-end reactions are the negative of the $\mathbf{f}'$ vector and are only associated with those elements having distributed loads acting on them. The member end forces provide the bending moments and shear forces from which the beam stresses can be determined.

Example 5.3

Figure E5.3 shows a three-dimensional frame subjected to various loads. Our task is to run program FRAME3D to obtain the maximum bending moments in the structure. The input and output files are as given in the program FRAME3D. From the output, we obtain the maximum $M_{y'} = 3.680\text{E} + 0.5\ \text{N}\cdot\text{m}$ occurring in member 1 at node 1 (the first node) and maximum $M_{Z'} = 1.413\text{E} + 0.5\ \text{N}\cdot\text{m}$ occurring in member 3 at node 4.

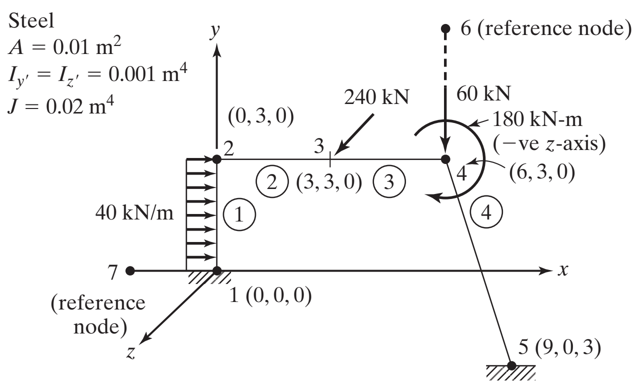


FIGURE E5.3 ■

5.9 PROBLEM MODELING AND BOUNDARY CONDITIONS

Some modeling aspects have already been discussed through examples. Few other situations are presented here. Problem of two overhanging beams with a gap is shown in Fig. 5.15. The suggested strategy is to solve the problem ignoring the gap. If $-Q_3 > a$ or $(Q_3 + a) < 0$ then we need to solve the problem imposing the condition $Q_9 = Q_3 + a$. This may be handled as a multipoint constraint. A symmetric plane frame problem is shown in Fig. 5.16. The load may be decomposed into symmetric and antisymmetric parts using the superposition principle as shown in Fig. 5.16a. The problem may be modeled as solving two half problems as shown in Fig. 5.16b. Displacements and member loads are then superposed and values in the other half can be obtained using symmetry as discussed in the previous chapter.

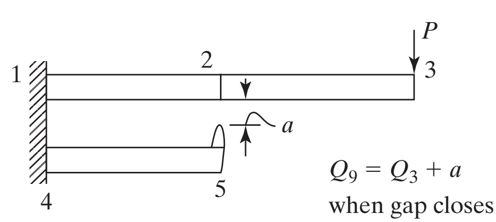


FIGURE 5.15 Beam with a gap.

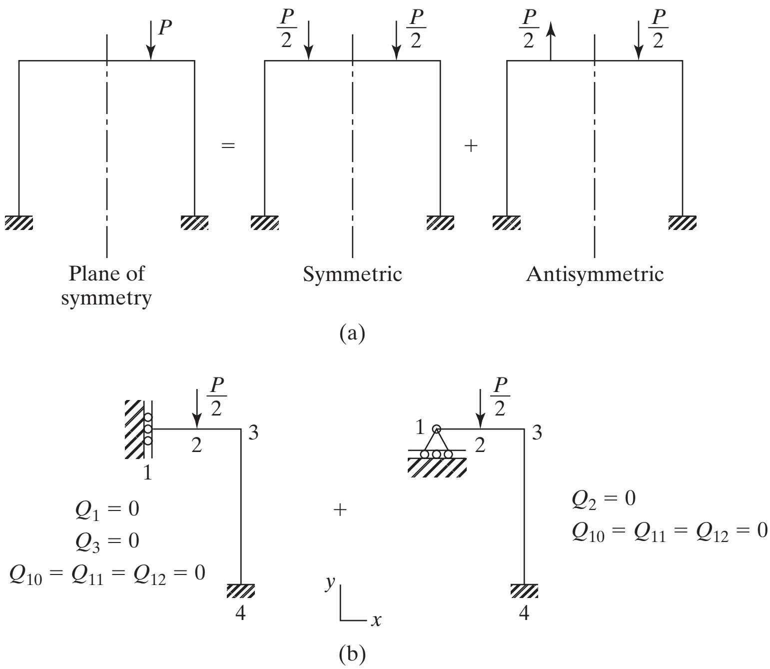


FIGURE 5.16 Symmetric plane frame.

5.10 SOME COMMENTS

Symmetric beams and plane and space frames have been discussed in this chapter. In engineering applications, there are several challenging problems, such as frames and mechanisms with pin-jointed members, unsymmetric beams, buckling of members due to axial loads, shear considerations, and structures with large deformations. For help in formulating and analyzing such problems, the reader may refer to some advanced publications in mechanics of solids, structural analysis, elasticity and plasticity, and finite element analysis.

Input Data/Output

```
INPUT TO BEAM
 << BEAM ANALYSIS >>
EXAMPLE 5.1
NN NE NM NDIM NEN NDN
 3  2  1   1    2   2
ND NL  NMPC
 4  4   0
NODE#  X-COORD
 1      0
 2      1000
 3      2000
```

Continued

```
EL#    N1  N2  MAT# Mom_Inertia
 1     1    2   1     4.00E+06
 2     2    3   1     4.00E+06
DOF#   Displ.
 1     0
 2     0
 3     0
 5     0
DOF#   LOAD
 3     -6000
 4     -1.00E+06
 5     -6000
 6     1.00E+06
MAT#   E
 1     200000
B1     i       B2  j  B3  (Multi-point constr. B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM BEAM
Results from Program BEAM
EXAMPLE 5.1
Node#  Displ.          Rotation
   1    2.00889E-11    6.69614E-09
   2   -1.27232E-10   -0.000267859
   3   -8.03572E-11    0.00044643
DOF#   Reaction
   1   -1285.691327
   2   -428553.0615
   3    8142.829592
   5    5142.861735
```

```
INPUT TO FRAME2D
<<2-D FRAME ANALYSIS >>
EXAMPLE 5.2
NN  NE  NM  NDIM  NEN  NDN
 4   3   1   2     2    3
ND  NL  NMPC
 6   1   0
Node#    X     Y
 1       0     96
 2       144   96
 3       0      0
 4       144    0
Elem#    N1    N2   Mat#   Area    Inertia  Distr_Load
 1       2      1    1     6.8      65      41.6666
 2       3      1    1     6.8      65        0
 3       4      2    1     6.8      65        0
DOF#     Displ.
 7       0
 8       0
```

Continued

```
   9   0
  10   0
  11   0
  12   0
DOF#  Load
 1    3000
MAT#  E
 1    3.00E+07
B1    i     B2  j  B3  (Multi-point constr. B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM FRAME2D
Results from Program Frame2D
EXAMPLE 5.2
Node#   X-Displ        Y-Displ       Z-Rotation
    1   0.09177       -0.0010358    -0.0013874
    2   0.09012       -0.0017877    -3.88368E-05
    3   4.91667E-10   -1.62547E-09  -4.44102E-08
    4   1.72372E-09   -2.80529E-09  -8.33197E-08
Member  End-Forces
Member#     1
2334.2004  -798.8360  -39254.5538
-2334.2004  798.8360  -75777.8342
Member#     2
2201.1592   665.7995981   60138.812
-2201.1592 -665.7995981   3777.950
Member#     3
 3798.831   2334.2004   112828.8
-3798.831  -2334.2004   111254.439
DOF#         Reaction
7           -665.800
8            2201.159
9            60138.812
10          -2334.200
11           3798.831
12           112828.8
```

PROBLEMS

5.1. Find the deflection at the load and the slopes at the ends for the steel shaft shown in Fig. P5.1. Consider the shaft to be simply supported at bearings A and B.

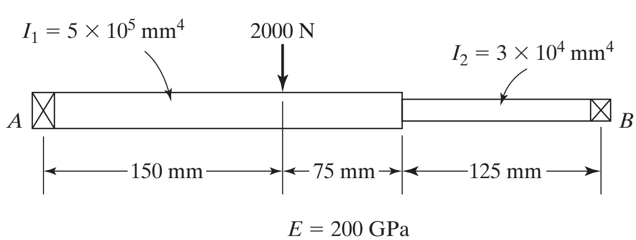


FIGURE P5.1 Problems 5.1 and 5.4.

5.2. A three-span beam is shown in Fig. P5.2. Determine the deflection curve of the beam and evaluate the reactions at the supports.

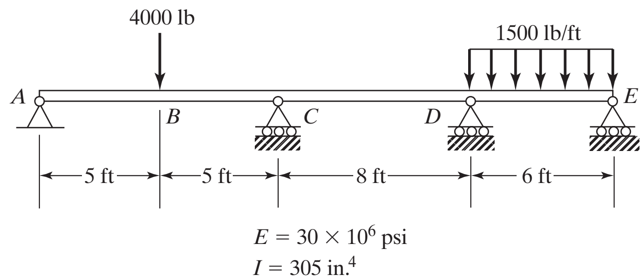


FIGURE P5.2

5.3. A reinforced concrete slab floor is shown in Fig. P5.3. Using a unit width of the slab in the z direction, determine the deflection curve of the neutral surface under its own weight.

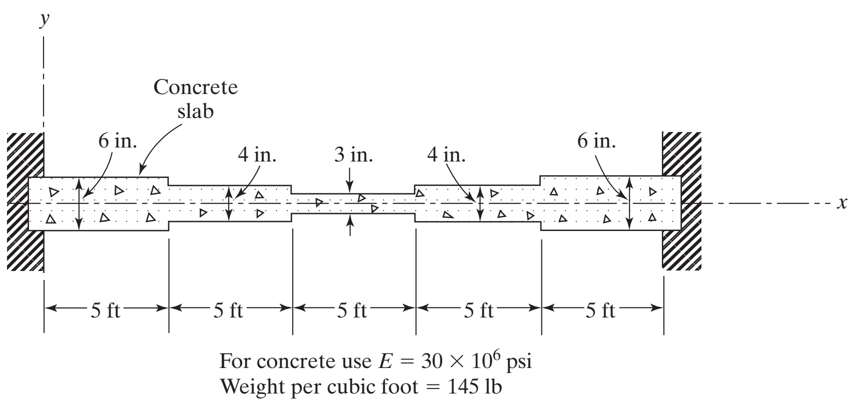


FIGURE P5.3

5.4. In the shaft shown in Fig. P5.1, determine the deflection at the loads and the slopes at the ends if the bearings at A and B have radial stiffnesses of 20 and 12 kN/mm, respectively.

5.5. Figure P5.5 shows a beam AD pinned at A and welded at B and C to long and slender rods BE and CF. A load of 4000 lb is applied at D as shown. Model the beam AD using beam elements and determine deflections at B, C, and D and stresses in rods BE and CF.

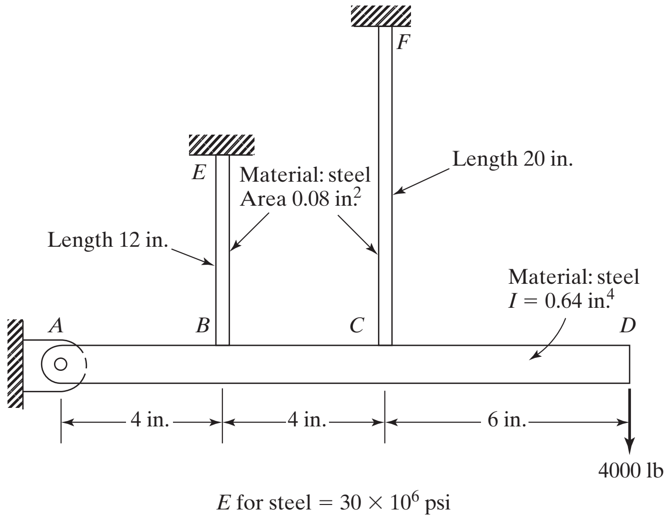


FIGURE P5.5

5.6. Figure P5.6 shows a cantilever beam with three rectangular openings. Find the deflections for the beam shown and compare the deflections with a beam without openings. Assume the Young's modulus of the material is 4.5×10^6 psi.

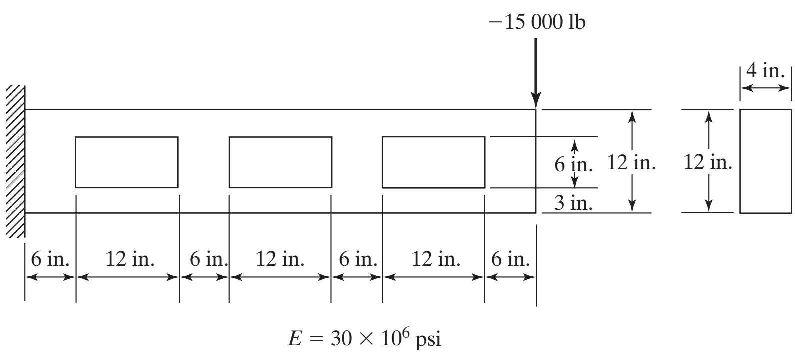


FIGURE P5.6

5.7. A simplified section of a machine tool spindle is shown in Fig. P5.7. Bearing B has a radial stiffness of 60 N/μm and a rotational stiffness (against moment) of 8×10^5 N·m/rad. Bearing C has a radial stiffness of 20 N/μm and its rotational stiffness can be neglected. For a load of 1000 N, as shown, determine the deflection and slope at A. Also, give the deflected shape of the spindle center line (1 μm = 10^{-6} m).

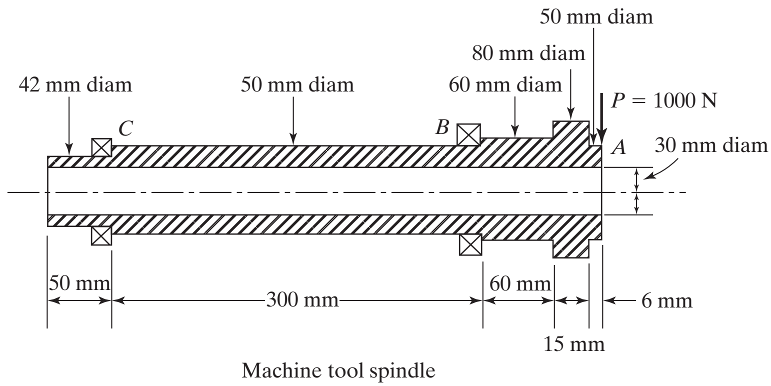


FIGURE P5.7

5.8. Determine the deflection at the center of BC for the frame shown in Fig. P5.8 using program FRAME2D. Also determine the reactions at A and D.

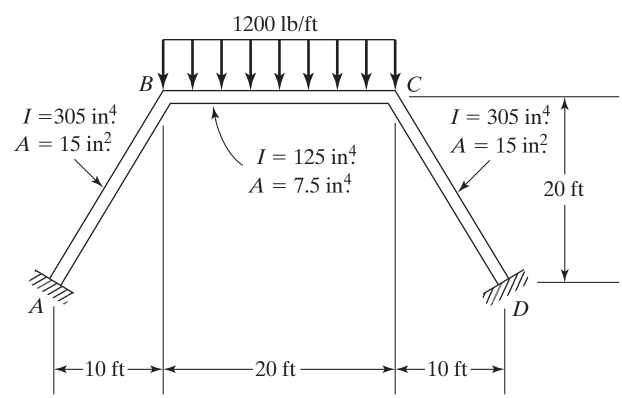


FIGURE P5.8

5.9. Figure P5.9 shows a hollow square section with two loading conditions. Using a 1-in. width perpendicular to the section, determine the deflection at the load for each of the two cases.

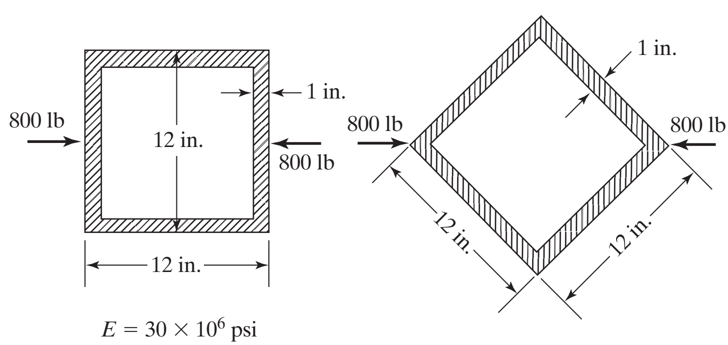


FIGURE P5.9

5.10. Figure P5.10 shows a five-member steel frame subjected to loads at the free end. The cross section of each member is a tube of wall thickness $t = 1$ cm and mean radius $R = 6$ cm. Determine the following for $E = 3 \times 10^5$ N/mm^2:

(a) The displacement of node 3

(b) The maximum axial compressive stress in a member.

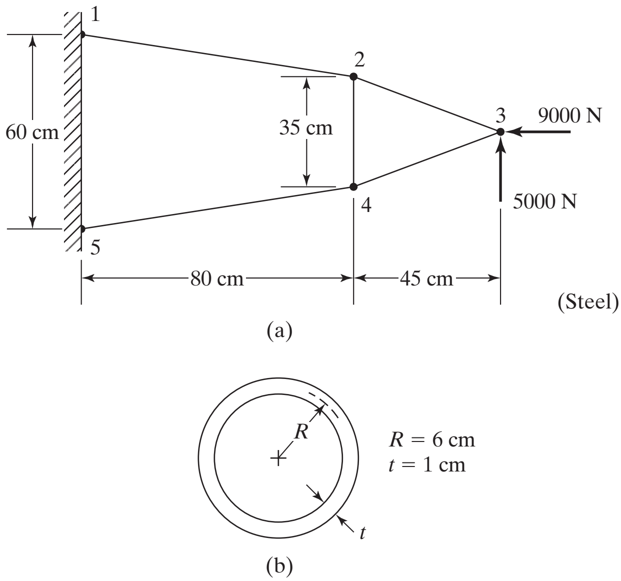


FIGURE P5.10

5.11. Dimensions of a common paper staple are shown in Fig. P5.11. While the staple is penetrating into the paper, a force of about 120 N is applied. Find the deformed shape for the following cases:

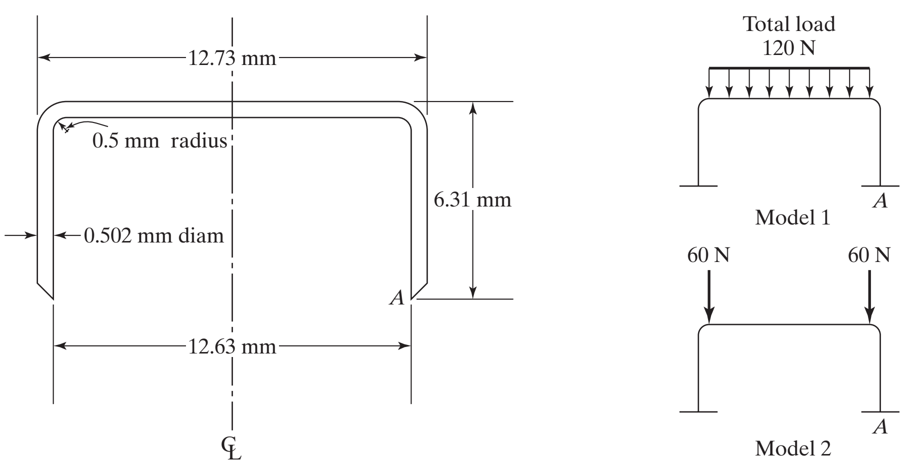


FIGURE P5.11

(a) Load uniformly distributed on the horizontal member and pinned condition at A at entry
(b) Load as in (a) with fixed condition at A after some penetration
(c) Load divided into two point loads, with A pinned
(d) Load as in (c) with A fixed.

5.12. A commonly used street light arrangement is shown in Fig. P5.12. Assuming fixed condition at A, compare the deformed shapes for the following two cases:
(a) Without the rod BC (i.e., only member ACD supports the light)
(b) With tie rod BC.

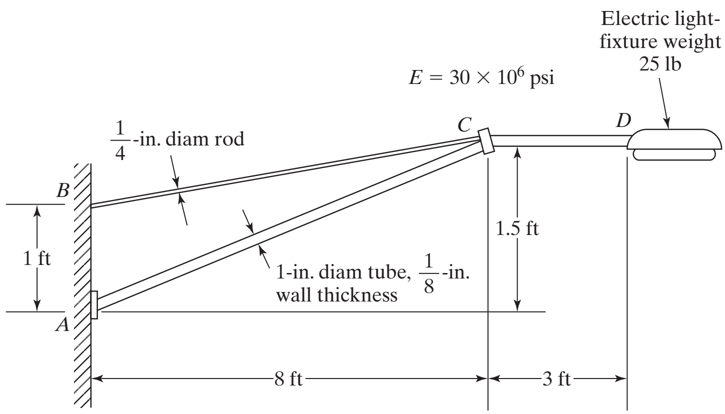


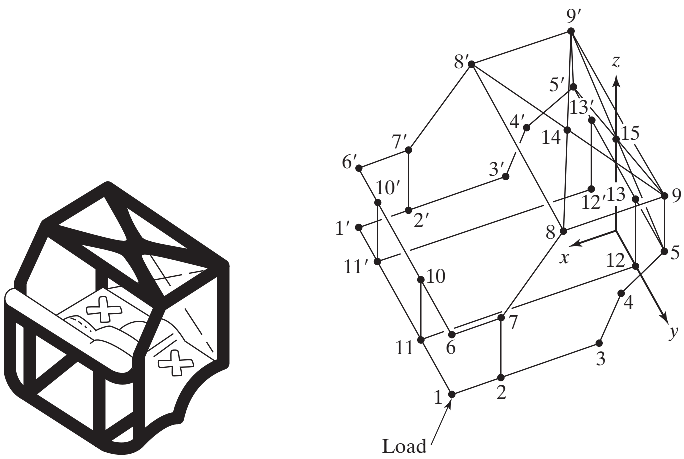
FIGURE P5.12

5.13. Figure P5.13a shows a cab of a van. A simplified finite element frame model is shown in Fig. P5.13b. The model consists of 28 nodes. x–z is a plane of symmetry; thus, nodes 1′–13′ have the same x- and z-coordinates as nodes 1–13, with y-coordinates reversed in sign. Each beam element is made of steel with $A = 0.2$ in.^{2}, $I_{y'} = I_{z'} = 0.003$ in.^{4}, and $J = 0.006$ in.^{4}. The loading corresponds to a frontal impact test based on Swedish standards and consists of a load at node 1 (only) with components $Fx = -3194.0$ lb and $Fy = -856.0$ lb. Treat nodes 11, 11′, 12, and 12′ as fixed (boundary conditions). Nodal coordinates in inches are as follows:

Node	x	y	z	Node	x	y	z
1	58.0	38.0	0	9	0	38.0	75.0
2	48.0	38.0	0	10	58.0	17.0	42.0
3	31.0	38.0	0	11	58.0	17.0	0
4	17.0	38.0	22.0	12	0	17.0	0
5	0	38.0	24.0	13	0	17.0	24.0
6	58.0	38.0	42.0	14	18.0	0	72.0
7	48.0	38.0	42.0	15	0	0	37.5
8	36.0	38.0	70.0				

Note: Number the nodes to keep bandwidth to a minimum.

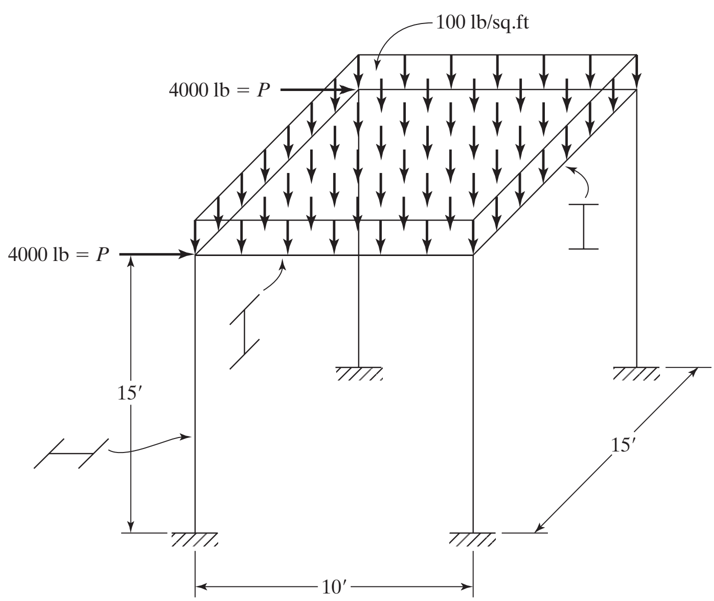
Determine the deflections at nodes 1, 2, 6, 7, 10, and 11 and the location and magnitude of the maximum bending moments using program FRAME3D.



(a) (b)

FIGURE P5.13 (a) Van frame. (b) Frame finite element model.

5.14. Consider the steel frame in Fig. P5.14, which is subjected to a wind load and roof load as shown. Determine the bending moments in the structure (maximum $M_{y'}$ and $M_{z'}$).



	Area (in.^{2})	$I_{y'}$ (in.^{4})	$I_{z'}$ (in.^{4})	J (in.^{4})
Columns	6.0	3.75	51.0	0.24
Beams	3.0	1.26	17.0	0.08

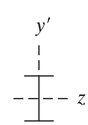


FIGURE P5.14

5.15. For a linearly distributed load for a beam element with end distributed values p_1 and p_2 as shown in Fig. P5.15, show that the equivalent nodal loads are given by

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} \dfrac{(7p_1 + 3p_2)\ell}{20} \\ \dfrac{(3p_1 + 2p_2)\ell^2}{60} \\ \dfrac{(3p_1 + 7p_2)\ell}{20} \\ \dfrac{-(2p_1 + 3p_2)\ell^2}{60} \end{bmatrix}$$

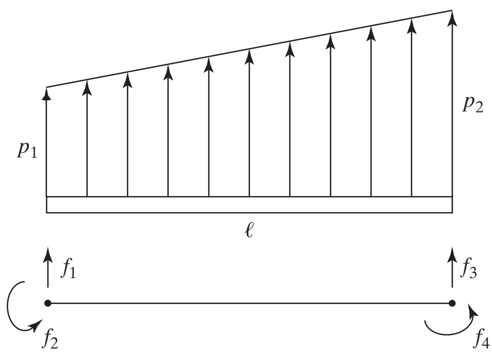


FIGURE P5.15

By setting $p_1 = p_2 = p$, check if it is consistent with the values for uniform distributed load case.

Program Listings

```
MAIN PROGRAM BEAM
'**************************************
'*             PROGRAM BEAM           *
'*        Beam Bending Analysis       *
'* T.R.Chandrupatla and A.D.Belegundu *
'**************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call Stiffness
     Call ModifyForBC
     Call BandSolver
     Call ReactionCalc
     Call Output
End Sub
```

```
ELEMENT STIFFNESS BEAM
Private Sub Stiffness()
     ReDim S(NQ, NBW)
     '----- Global Stiffness Matrix -----
     For N = 1 To NE
         N1 = NOC(N, 1)
         N2 = NOC(N, 2)
         M = MAT(N)
         EL = Abs(X(N1) - X(N2))
         EIL = PM(M, 1) * SMI(N) / EL ^ 3
         SE(1, 1) = 12 * EIL
         SE(1, 2) = EIL * 6 * EL
         SE(1, 3) = -12 * EIL
         SE(1, 4) = EIL * 6 * EL
            SE(2, 1) = SE(1, 2)
            SE(2, 2) = EIL * 4 * EL * EL
            SE(2, 3) = -EIL * 6 * EL
            SE(2, 4) = EIL * 2 * EL * EL
         SE(3, 1) = SE(1, 3)
         SE(3, 2) = SE(2, 3)
         SE(3, 3) = EIL * 12
         SE(3, 4) = -EIL * 6 * EL
            SE(4, 1) = SE(1, 4)
            SE(4, 2) = SE(2, 4)
            SE(4, 3) = SE(3, 4)
            SE(4, 4) = EIL * 4 * EL * EL
'Stiffness assembly routine is common to all programs Truss etc
```

```
MAIN PROGRAM FRAME2D
'********    PROGRAM FRAME2D      ********
'*      2-D   FRAME ANALYSIS BY FEM      *
'*  T.R.Chandrupatla and A.D.Belegundu   *
'*****************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call Stiffness
     Call AddLoads
     Call ModifyForBC
     Call BandSolver
     Call EndActions
     Call ReactionCalc
     Call Output
End Sub
```

```
ELEMENT STIFFNESS FRAME2D
Private Sub Elstif(N)
     '----- Element Stiffness Matrix -----
     I1 = NOC(N, 1): I2 = NOC(N, 2): M = MAT(N)
     X21 = X(I2, 1) - X(I1, 1)
     Y21 = X(I2, 2) - X(I1, 2)
     EL = Sqr(X21 * X21 + Y21 * Y21)
     EAL = PM(M, 1) * ARIN(N, 1) / EL
     EIZL = PM(M, 1) * ARIN(N, 2) / EL
     For I = 1 To 6
     For J = 1 To 6
       SEP(I, J) = 0!
     Next J: Next I
     SEP(1, 1) = EAL: SEP(1, 4) = -EAL: SEP(4, 4) = EAL
     SEP(2, 2) = 12 * EIZL / EL ^ 2: SEP(2, 3) = 6 * EIZL / EL
     SEP(2, 5) = -SEP(2, 2): SEP(2, 6) = SEP(2, 3)
     SEP(3, 3) = 4 * EIZL
     SEP(3, 5) = -6 * EIZL / EL: SEP(3, 6) = 2 * EIZL
     SEP(5, 5) = 12 * EIZL / EL ^ 2: SEP(5, 6) = -6 * EIZL / EL
     SEP(6, 6) = 4 * EIZL
     For I = 1 To 6
     For J = I To 6
       SEP(J, I) = SEP(I, J)
     Next J: Next I
'CONVERT ELEMENT STIFFNESS MATRIX TO GLOBAL SYSTEM
     DCOS(1, 1) = X21 / EL: DCOS(1, 2) = Y21 / EL: DCOS(1, 3) = 0
     DCOS(2, 1) = -DCOS(1, 2): DCOS(2, 2) = DCOS(1, 1): DCOS(2, 3) = 0
     DCOS(3, 1) = 0: DCOS(3, 2) = 0: DCOS(3, 3) = 1
     For I = 1 To 6
     For J = 1 To 6
       ALAMBDA(I, J) = 0!
     Next J: Next I
     For K = 1 To 2
       IK = 3 * (K - 1)
       For I = 1 To 3
       For J = 1 To 3
         ALAMBDA(I + IK, J + IK) = DCOS(I, J)
       Next J: Next I
     Next K
     If ISTF = 1 Then Exit Sub
     For I = 1 To 6
     For J = 1 To 6
       SE(I, J) = 0
       For K = 1 To 6
         SE(I, J) = SE(I, J) + SEP(I, K) * ALAMBDA(K, J)
       Next K
     Next J: Next I
     For I = 1 To 6: For J = 1 To 6: SEP(I, J) = SE(I, J): Next J: Next
     For I = 1 To 6
     For J = 1 To 6
       SE(I, J) = 0
       For K = 1 To 6
         SE(I, J) = SE(I, J) + ALAMBDA(K, I) * SEP(K, J)
       Next K
     Next J: Next I
End Sub
```

```
UNIFORMLY DISTRIBUTED LOAD TO POINT LOADS
Private Sub AddLoads()
'----- Loads due to uniformly distributed load on element
     For N = 1 To NE
     If Abs(UDL(N)) > 0 Then
       ISTF = 1
       Call Elstif(N)
       I1 = NOC(N, 1): I2 = NOC(N, 2)
       X21 = X(I2, 1) - X(I1, 1)
       Y21 = X(I2, 2) - X(I1, 2)
       EL = Sqr(X21 * X21 + Y21 * Y21)
       ED(1) = 0: ED(4) = 0
       ED(2) = UDL(N) * EL / 2: ED(5) = ED(2)
       ED(3) = UDL(N) * EL ^ 2 / 12: ED(6) = -ED(3)
       For I = 1 To 6
         EDP(I) = 0
         For K = 1 To 6
           EDP(I) = EDP(I) + ALAMBDA(K, I) * ED(K)
         Next K
       Next I
       For I = 1 To 3
         F(3 * I1 - 3 + I) = F(3 * I1 - 3 + I) + EDP(I)
         F(3 * I2 - 3 + I) = F(3 * I2 - 3 + I) + EDP(I + 3)
       Next I
     End If
     Next N
End Sub
```

```
MEMBER END FORCES
Private Sub EndActions()
     ReDim EF(NE, 6)
     '----- Calculating Member End-Forces
     For N = 1 To NE
       ISTF = 1
       Call Elstif(N)
       I1 = NOC(N, 1): I2 = NOC(N, 2)
       X21 = X(I2, 1) - X(I1, 1)
       Y21 = X(I2, 2) - X(I1, 2)
       EL = Sqr(X21 * X21 + Y21 * Y21)
       For I = 1 To 3
         ED(I) = F(3 * I1 - 3 + I): ED(I + 3) = F(3 * I2 - 3 + I)
       Next I
       For I = 1 To 6
         EDP(I) = 0
         For K = 1 To 6
           EDP(I) = EDP(I) + ALAMBDA(I, K) * ED(K)
         Next K
       Next I
```

Continued

```
'----- END FORCES DUE TO DISTRIBUTED LOADS
       If Abs(UDL(N)) > 0 Then
         ED(1) = 0: ED(4) = 0
         ED(2) = -UDL(N) * EL / 2: ED(5) = ED(2)
         ED(3) = -UDL(N) * EL ^ 2 / 12: ED(6) = -ED(3)
       Else
         For K = 1 To 6: ED(K) = 0: Next K
       End If
       For I = 1 To 6
         EF(N, I) = ED(I)
         For K = 1 To 6
           EF(N, I) = EF(N, I) + SEP(I, K) * EDP(K)
         Next K
       Next I
     Next N
End Sub
```

CHAPTER 6

Two-Dimensional Problems Using Constant Strain Triangles

6.1 INTRODUCTION

The two-dimensional finite element formulation in this chapter follows the steps used in the one-dimensional problem. The displacements, traction components, and distributed body force values are functions of the position indicated by (x, y). The displacement vector $\mathbf{u}$ is given as

$$\mathbf{u} = [u \quad v]^{\mathrm{T}} \tag{6.1}$$

where u and v are the x and y components of $\mathbf{u}$ respectively. The stresses and strains are given by

$$\boldsymbol{\sigma} = [\sigma_x \quad \sigma_y \quad \tau_{xy}]^{\mathrm{T}} \tag{6.2}$$

$$\boldsymbol{\epsilon} = [\epsilon_x \quad \epsilon_y \quad \gamma_{xy}]^{\mathrm{T}} \tag{6.3}$$

From Fig. 6.1, representing the two-dimensional problem in a general setting, the body force, traction vector, and elemental volume are given by

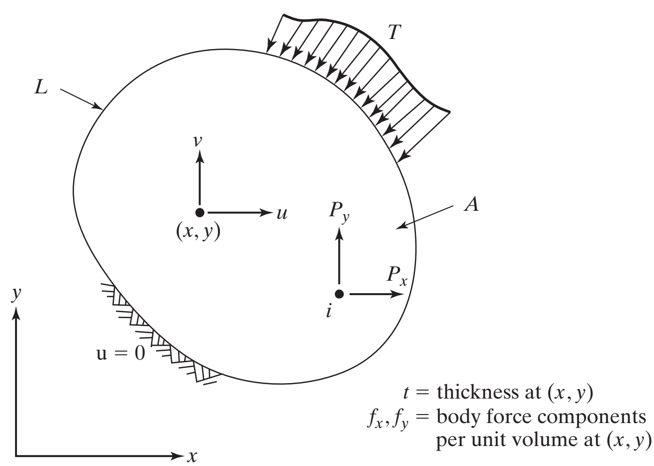


FIGURE 6.1 Two-dimensional problem.

$$\mathbf{f} = [f_x \quad f_y]^{\mathrm{T}} \qquad \mathbf{T} = [T_x \quad T_y]^{\mathrm{T}} \qquad \text{and} \qquad dV = t\,dA \tag{6.4}$$

where t is the thickness along the z direction. The body force $\mathbf{f}$ has the units force/unit volume, while the traction force $\mathbf{T}$ has the units force/unit area. The strain–displacement relations are given by

$$\boldsymbol{\epsilon} = \left[\frac{\partial u}{\partial x} \quad \frac{\partial v}{\partial y} \quad \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right]^{\mathrm{T}} \tag{6.5}$$

Stresses and strains are related by (see Eqs. 1.18 and 1.19)

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} \tag{6.6}$$

The region is discretized with the idea of expressing the displacements in terms of values at discrete points. Triangular elements are introduced first. Stiffness and load concepts are then developed using energy and Galerkin approaches.

6.2 FINITE ELEMENT MODELING

The two-dimensional region is divided into straight-sided triangles. Figure 6.2 shows a typical triangulation. The points where the corners of the triangles meet are called *nodes*, and each triangle formed by three nodes and three sides is called an *element*. The elements fill the entire region except a small region at the boundary. This unfilled region exists for curved boundaries, and it can be reduced by choosing smaller elements

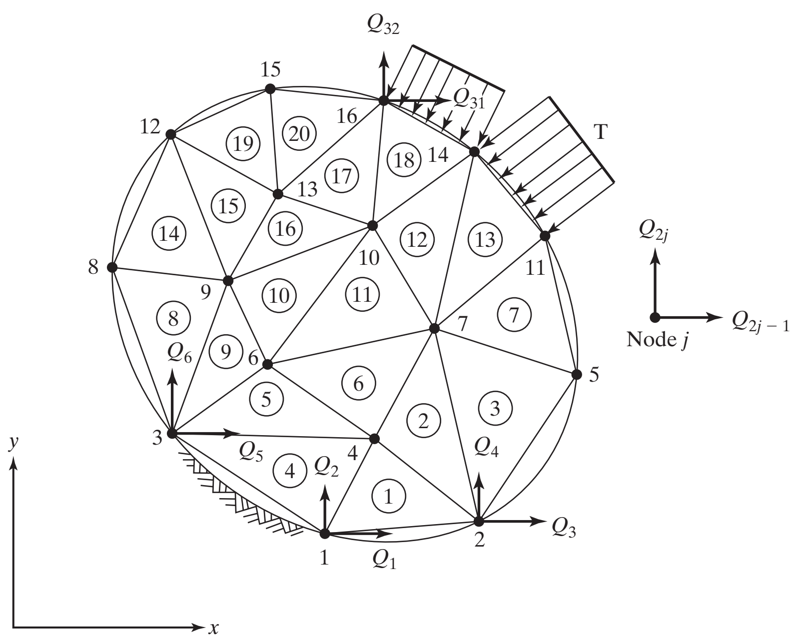


FIGURE 6.2 Finite element discretization.

or elements with curved boundaries. The idea of the finite element method is to solve the continuous problem approximately, and this unfilled region contributes to some part of this approximation. For the triangulation shown in Fig. 6.2, the node numbers are indicated at the corners and element numbers are circled.

In the two-dimensional problem discussed here, each node is permitted to displace in the x and y directions. Thus, each node has two degrees of freedom (dof). As seen from the numbering scheme used in trusses, the displacement components of node j are taken as Q_{2j-1} in the x direction and Q_{2j} in the y direction. We denote the global displacement vector as

$$\mathbf{Q} = [Q_1 \quad Q_2 \quad \ldots \quad Q_N]^{\mathrm{T}} \tag{6.7}$$

where N is the number of dof.

Computationally, the information on the triangulation is to be represented in the form of *nodal coordinates* and *connectivity*. The nodal coordinates are stored in a two-dimensional array represented by the total number of nodes and the two coordinates per node. The connectivity may be clearly seen by isolating a typical element, as shown in Fig. 6.3. For the three nodes designated locally as 1, 2, and 3, the corresponding global node numbers are defined in Fig. 6.2. This element connectivity information becomes an array of the size of number of elements and three nodes per element. A typical connectivity representation is shown in Table 6.1. Most standard finite element codes use the convention of going around the element in a counterclockwise direction to avoid calculating a negative area. However, in the program that accompanies this chapter, ordering is not necessary.

Table 6.1 establishes the correspondence of local and global node numbers and the corresponding dof. The displacement components of a local node j in Fig. 6.3 are represented as q_{2j-1} and q_{2j} in the x and y directions, respectively. We denote the element displacement vector as

$$\mathbf{q} = [q_1 \quad q_2 \quad \ldots \quad q_6]^{\mathrm{T}} \tag{6.8}$$

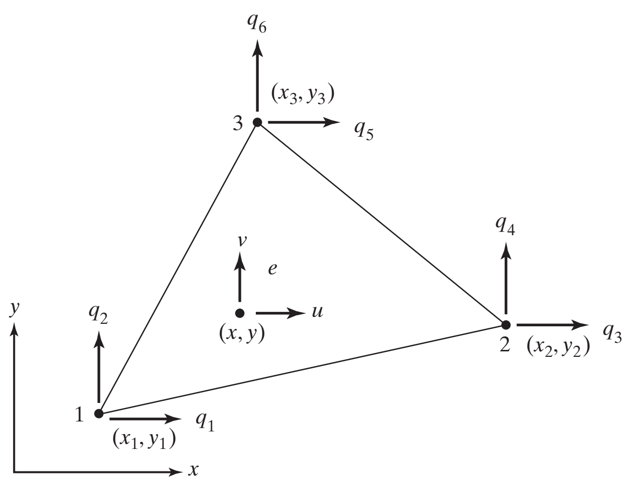


FIGURE 6.3 Triangular element.

TABLE 6.1 Element Connectivity

Element number e	Three nodes		
	1	2	3
1	1	2	4
2	4	2	7
⋮			
11	6	7	10
⋮			
20	13	16	15

Note that from the connectivity matrix in Table 6.1, we can extract the **q** vector from the global **Q** vector, an operation performed frequently in a finite element program. Also, the nodal coordinates designated by (x_1, y_1), (x_2, y_2), and (x_3,y_3) have the global correspondence established through Table 6.1. The local representation of nodal coordinates and dof provides a setting for a simple and clear representation of element characteristics.

6.3 CONSTANT STRAIN TRIANGLE (CST)

The displacements at points inside an element need to be represented in terms of the nodal displacements of the element. As discussed earlier, the finite element method uses the concept of shape functions in systematically developing these interpolations. For the CST, the shape functions are linear over the element. The three shape functions N_1, N_2, and N_3 corresponding to nodes 1, 2, and 3, respectively, are shown in Fig. 6.4. Shape function N_1 is 1 at node 1 and linearly reduces to 0 at nodes 2 and 3. The values of shape function N_1 thus define a plane surface shown shaded in Fig. 6.4a. N_2 and N_3 are represented by similar surfaces having values of 1 at nodes 2 and 3, respectively, and dropping to 0 at the opposite edges. Any linear combination of these shape functions also represents a plane surface. In particular, $N_1 + N_2 + N_3$ represents a plane at a height of 1 at nodes 1, 2, and 3, and, thus, it is parallel to the triangle 123. Consequently, for every N_1, N_2, and N_3,

$$N_1 + N_2 + N_3 = 1 \tag{6.9}$$

N_1, N_2, and N_3 are therefore not linearly independent; only two of these are independent. The independent shape functions are conveniently represented by the pair ξ, η as

$$N_1 = \xi \qquad N_2 = \eta \qquad N_3 = 1 - \xi - \eta \tag{6.10}$$

where ξ, η are natural coordinates (Fig. 6.4). At this stage, the similarity with the one-dimensional element (Chapter 3) should be noted: in the one-dimensional problem, the x-coordinates were mapped onto the ξ-coordinates, and shape functions were defined as functions of ξ. Here, in the two-dimensional problem, the x-, y-coordinates are mapped onto the ξ-, η-coordinates, and shape functions are defined as functions of ξ and η.

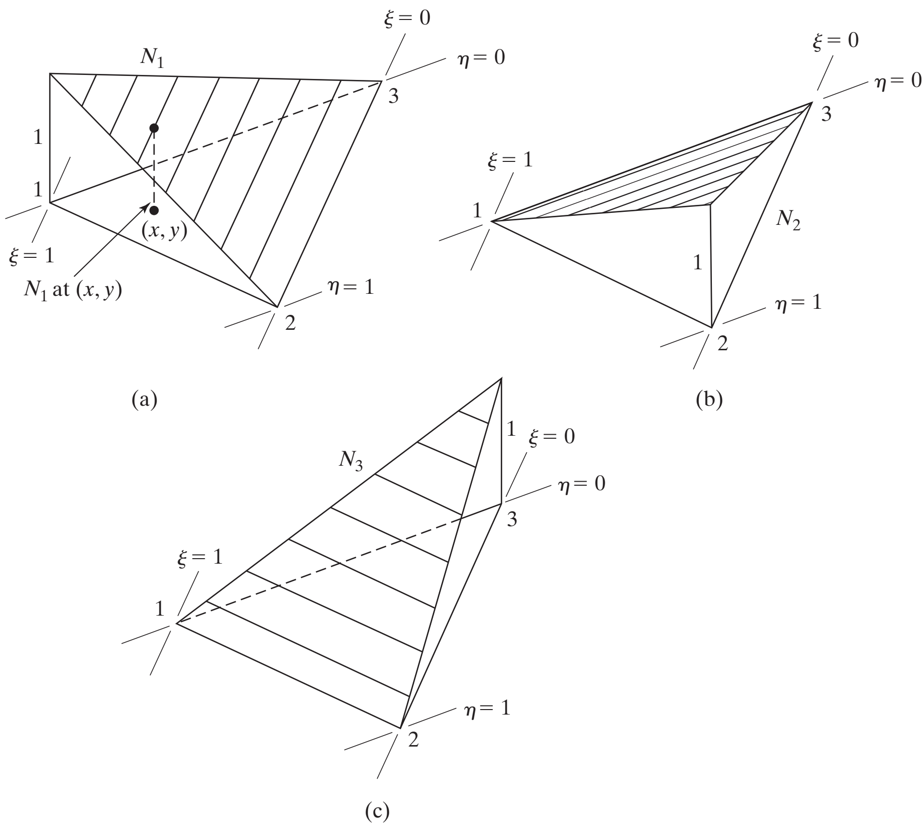


FIGURE 6.4 Shape functions.

The shape functions can be physically represented by **area coordinates**. A point (x, y) in a triangle divides it into three areas, A_1, A_2, and A_3, as shown in Fig. 6.5. The shape functions N_1, N_2, and N_3 are precisely represented by

$$N_1 = \frac{A_1}{A} \qquad N_2 = \frac{A_2}{A} \qquad N_3 = \frac{A_3}{A} \tag{6.11}$$

where A is the area of the element. Clearly, $N_1 + N_2 + N_3 = 1$ at all points inside the triangle.

Isoparametric Representation

The displacements inside the element are now written using the shape functions and the nodal values of the unknown displacement field. We have

$$\begin{aligned} u &= N_1 q_1 + N_2 q_3 + N_3 q_5 \\ v &= N_1 q_2 + N_2 q_4 + N_3 q_6 \end{aligned} \tag{6.12a}$$

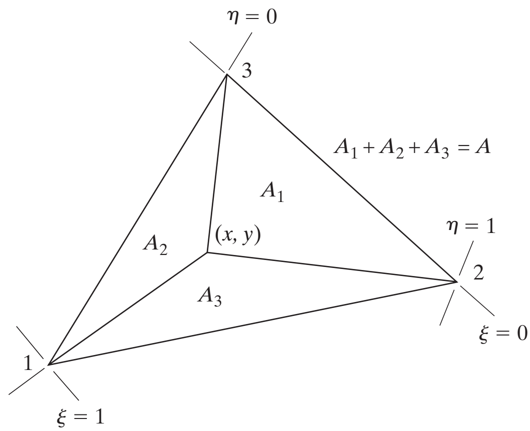


FIGURE 6.5 Area coordinates.

or, using Eq. 6.10,

$$
\begin{aligned}
u &= (q_1 - q_5)\xi + (q_3 - q_5)\eta + q_5 \\
v &= (q_2 - q_6)\xi + (q_4 - q_6)\eta + q_6
\end{aligned}
\tag{6.12b}
$$

The relations in Eq. 6.12a can be expressed in a matrix form by defining a shape function matrix

$$
\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}
\tag{6.13}
$$

and

$$
\mathbf{u} = \mathbf{N}\mathbf{q}
\tag{6.14}
$$

For the triangular element, the coordinates x, y can also be represented in terms of nodal coordinates using the same shape functions. This is **isoparametric representation**. This approach lends to simplicity of development and retains the uniformity with other complex elements. We have

$$
\begin{aligned}
x &= N_1x_1 + N_2x_2 + N_3x_3 \\
y &= N_1y_1 + N_2y_2 + N_3y_3
\end{aligned}
\tag{6.15a}
$$

or

$$
\begin{aligned}
x &= (x_1 - x_3)\xi + (x_2 - x_3)\eta + x_3 \\
y &= (y_1 - y_3)\xi + (y_2 - y_3)\eta + y_3
\end{aligned}
\tag{6.15b}
$$

Using the notations, $x_{ij} = x_i - x_j$ and $y_{ij} = y_i - y_j$, we can write Eq. 6.15b as

$$
\begin{aligned}
x &= x_{13}\xi + x_{23}\eta + x_3 \\
y &= y_{13}\xi + y_{23}\eta + y_3
\end{aligned}
\tag{6.15c}
$$

This equation relates x- and y-coordinates to the ξ- and η-coordinates. Equation 6.12 expresses u and v as functions of ξ and η.

Example 6.1

Evaluate the shape functions N_1, N_2, and N_3 at the interior point P for the triangular element shown in Fig. E6.1.

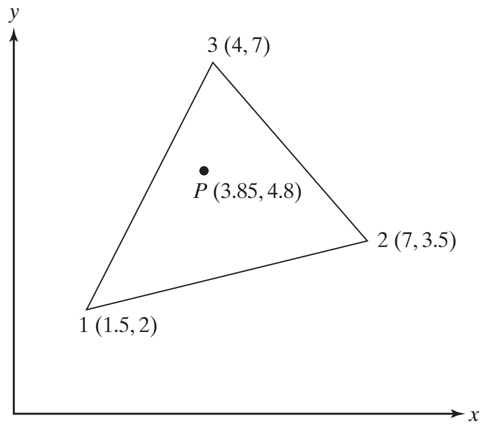


FIGURE E6.1 Examples 6.1 and 6.2.

Solution Using the isoparametric representation (Eqs. 6.15a, b, c), we get

$$3.85 = 1.5N_1 + 7N_2 + 4N_3 = -2.5\xi + 3\eta + 4$$
$$4.8 = 2N_1 + 3.5N_2 + 7N_3 = -5\xi - 3.5\eta + 7$$

These two equations are rearranged in the form

$$2.5\xi - 3\eta = 0.15$$
$$5\xi + 3.5\eta = 2.2$$

Solving the equations, we obtain $\xi = 0.3$ and $\eta = 0.2$, which imply that

$$N_1 = 0.3 \qquad N_2 = 0.2 \qquad N_3 = 0.5$$ ■

In evaluating the strains, partial derivatives of u and v are to be taken with respect to x and y. From Eqs. 6.12 and 6.15, we see that u, v and x, y are functions of ξ and η. That is, $u = u[x(\xi, \eta), y(\xi, \eta)]$ and similarly $v = v[x(\xi, \eta), y(\xi, \eta)]$. Using the chain rule for partial derivatives of u, we get

$$\frac{\partial u}{\partial \xi} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \xi}$$
$$\frac{\partial u}{\partial \eta} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \eta}$$

which can be written in matrix notation as

$$\left\{\begin{matrix}\dfrac{\partial u}{\partial \xi}\\[2ex]\dfrac{\partial u}{\partial \eta}\end{matrix}\right\} = \begin{bmatrix}\dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi}\\[2ex]\dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta}\end{bmatrix}\left\{\begin{matrix}\dfrac{\partial u}{\partial x}\\[2ex]\dfrac{\partial u}{\partial y}\end{matrix}\right\} \tag{6.16}$$

where the (2×2) square matrix is denoted as the *Jacobian* of the transformation, **J**:

$$\mathbf{J} = \begin{bmatrix}\dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi}\\[2ex]\dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta}\end{bmatrix} \tag{6.17}$$

Some additional properties of the Jacobian are given in the appendix. On taking the derivative of x and y,

$$\mathbf{J} = \begin{bmatrix}x_{13} & y_{13}\\ x_{23} & y_{23}\end{bmatrix} \tag{6.18}$$

Also, from Eq. 6.16,

$$\left\{\begin{matrix}\dfrac{\partial u}{\partial x}\\[2ex]\dfrac{\partial u}{\partial y}\end{matrix}\right\} = \mathbf{J}^{-1}\left\{\begin{matrix}\dfrac{\partial u}{\partial \xi}\\[2ex]\dfrac{\partial u}{\partial \eta}\end{matrix}\right\} \tag{6.19}$$

where $\mathbf{J}^{-1}$ is the inverse of the Jacobian **J**, given by

$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}}\begin{bmatrix}y_{23} & -y_{13}\\ -x_{23} & x_{13}\end{bmatrix} \tag{6.20}$$

$$\det \mathbf{J} = x_{13}\,y_{23} - x_{23}\,y_{13} \tag{6.21}$$

From the knowledge of the area of the triangle, it can be seen that the magnitude of det **J** is twice the area of the triangle. If the points 1, 2, and 3 are ordered in a counterclockwise manner, det **J** is positive in sign. We have

$$A = \tfrac{1}{2}|\det \mathbf{J}| \tag{6.22}$$

where $|\ |$ represents the magnitude. Most computer codes use a counterclockwise order for the nodes and use det **J** for evaluating the area.

Example 6.2

Determine the Jacobian of the transformation **J** for the triangular element shown in Fig. E6.1.

Solution We have

$$\mathbf{J} = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix} = \begin{bmatrix} -2.5 & -5.0 \\ 3.0 & -3.5 \end{bmatrix}$$

Thus, det $\mathbf{J}$ = 23.75 units. This is twice the area of the triangle. If 1, 2, 3 are in a clockwise order, then det $\mathbf{J}$ will be negative. ■

From Eqs. 6.19 and 6.20, it follows that

$$\begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{Bmatrix} y_{23}\dfrac{\partial u}{\partial \xi} - y_{13}\dfrac{\partial u}{\partial \eta} \\ -x_{23}\dfrac{\partial u}{\partial \xi} + x_{13}\dfrac{\partial u}{\partial \eta} \end{Bmatrix} \tag{6.23a}$$

Replacing u by the displacement v, we get a similar expression

$$\begin{Bmatrix} \dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{Bmatrix} y_{23}\dfrac{\partial v}{\partial \xi} - y_{13}\dfrac{\partial v}{\partial \eta} \\ -x_{23}\dfrac{\partial v}{\partial \xi} + x_{13}\dfrac{\partial v}{\partial \eta} \end{Bmatrix} \tag{6.23b}$$

Using the strain–displacement relations (Eq. 6.5) and Eqs. 6.12b and 6.23, we get

$$\boldsymbol{\epsilon} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix}$$

$$= \frac{1}{\det \mathbf{J}} \begin{Bmatrix} y_{23}(q_1 - q_5) \quad - y_{13}(q_3 - q_5) \\ -x_{23}(q_2 - q_6) + x_{13}(q_4 - q_6) \\ -x_{23}(q_1 - q_5) + x_{13}(q_3 - q_5) + y_{23}(q_2 - q_6) - y_{13}(q_4 - q_6) \end{Bmatrix} \tag{6.24a}$$

From the definition of x_{ij} and y_{ij}, we can write $y_{31} = -y_{13}$ and $y_{12} = y_{13} - y_{23}$ and so on. The foregoing equation can be written in the form

$$\boldsymbol{\epsilon} = \frac{1}{\det \mathbf{J}} \begin{Bmatrix} y_{23}q_1 + y_{31}q_3 + y_{12}q_5 \\ x_{32}q_2 + x_{13}q_4 + x_{21}q_6 \\ x_{32}q_1 + y_{23}q_2 + x_{13}q_3 + y_{31}q_4 + x_{21}q_5 + y_{12}q_6 \end{Bmatrix} \tag{6.24b}$$

This equation can be written in matrix form as

$$\boldsymbol{\epsilon} = \mathbf{Bq} \tag{6.25}$$

where $\mathbf{B}$ is a (3×6) element strain–displacement matrix relating the three strains to the six nodal displacements and is given by

$$\mathbf{B} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix} \tag{6.26}$$

It may be noted that all the elements of the $\mathbf{B}$ matrix are constants expressed in terms of the nodal coordinates.

Example 6.3

Find the strain–nodal displacement matrices $\mathbf{B}^e$ for the elements shown in Fig. E6.3. Use local numbers given at the corners.

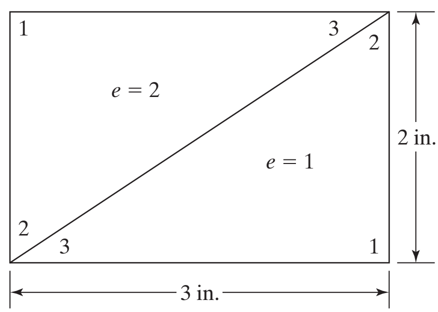


FIGURE E6.3

Solution We have

$$\mathbf{B}^1 = \frac{1}{\det \mathbf{J}} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}$$

$$= \frac{1}{6} \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & -3 & 0 & 3 & 0 & 0 \\ -3 & 2 & 3 & 0 & 0 & -2 \end{bmatrix}$$

where $\det \mathbf{J}$ is obtained from $x_{13}y_{23} - x_{23}y_{13} = (3)(2) - (3)(0) = 6$. Using the local numbers at the corners, $\mathbf{B}^2$ can be written using the relationship as

$$\mathbf{B}^2 = \frac{1}{6} \begin{bmatrix} -2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 3 & 0 & -3 & 0 & 0 \\ 3 & -2 & -3 & 0 & 0 & 2 \end{bmatrix}$$

■

Potential Energy Approach

The potential energy of the system, Π, is given by

$$\Pi = \frac{1}{2}\int_A \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}\, t\, dA - \int_A \mathbf{u}^{\mathrm{T}}\mathbf{f} t\, dA - \int_L \mathbf{u}^{\mathrm{T}}\mathbf{T} t\, d\ell - \sum_i \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i \tag{6.27}$$

In the last term in Eq. 6.27, i indicates the point of application of a point load $\mathbf{P}_i$ and $\mathbf{P}_i = [P_x, P_y]_i^{\mathrm{T}}$. The summation over i gives the potential energy due to all point loads.

Using the triangulation shown in Fig. 6.2, the total potential energy can be written in the form

$$\Pi = \sum_e \frac{1}{2}\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}\, t\, dA - \sum_e \int_e \mathbf{u}^{\mathrm{T}}\mathbf{f} t\, dA - \int_L \mathbf{u}^{\mathrm{T}}\mathbf{T} t\, d\ell - \sum_i \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i \tag{6.28a}$$

or

$$\Pi = \sum_e U_e - \sum_e \int_e \mathbf{u}^{\mathrm{T}}\mathbf{f} t\, dA - \int_L \mathbf{u}^{\mathrm{T}}\mathbf{T} t\, d\ell - \sum_i \mathbf{u}_i^{T}\mathbf{P}_i \tag{6.28b}$$

where $U_e = \frac{1}{2}\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}\, t\, dA$ is the element strain energy.

Element Stiffness

We now substitute for the strain from the element strain–displacement relationship in Eq. 6.25 into the element strain energy U_e in Eq. 6.28b, to obtain

$$U_e = \frac{1}{2}\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}\, t\, dA \tag{6.29a}$$

$$= \frac{1}{2}\int_e \mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\mathbf{q}\, t\, dA \tag{6.29a}$$

Taking the element thickness t_e as constant over the element and remembering that all terms in the $\mathbf{D}$ and $\mathbf{B}$ matrices are constants, we get

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B} t_e\left(\int_e dA\right)\mathbf{q} \tag{6.29b}$$

Now, $\int_e dA = A_e$, where A_e is the area of the element. Thus,

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}} t_e A_e \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\mathbf{q} \tag{6.29c}$$

or

$$U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} \tag{6.29d}$$

where $\mathbf{k}^e$ is the element stiffness matrix given by

$$\mathbf{k}^e = t_e A_e \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \tag{6.30}$$

For plane stress or plane strain, the element stiffness matrix can be obtained by taking the appropriate material property matrix $\mathbf{D}$ (defined in Chapter 1) and carrying out the previous multiplication on the computer. We note that $\mathbf{k}^e$ is symmetric since $\mathbf{D}$ is symmetric. The element connectivity as established in Table 6.1 is now used to add the element stiffness values in $\mathbf{k}^e$ into the corresponding global locations in the global stiffness matrix $\mathbf{K}$, so that

$$\begin{aligned} U &= \sum_e \frac{1}{2} \mathbf{q}^{\mathrm{T}} \mathbf{k}^e \mathbf{q} \\ &= \frac{1}{2} \mathbf{Q}^{\mathrm{T}} \mathbf{K} \mathbf{Q} \end{aligned} \tag{6.31}$$

The global stiffness matrix $\mathbf{K}$ is symmetric and banded or sparse. The stiffness value K_{ij} is zero when the dof i and j are not connected through an element. If i and j are connected through one or more elements, stiffness values accumulate from these elements. For the global dof numbering shown in Fig. 6.2, the bandwidth is related to the maximum difference in node numbers of an element, over all the elements. If i_1, i_2, and i_3 are node numbers of an element e, the maximum element node number difference is given by

$$m_e = \max\left(|i_1 - i_2|, |i_2 - i_3|, |i_3 - i_1|\right) \tag{6.32a}$$

The half-bandwidth is then given by

$$\mathrm{NBW} = 2\left(\max_{1 \le e \le \mathrm{NE}} (m_e) + 1\right) \tag{6.32b}$$

where NE is the number of elements and 2 is the number of dof per node.

The global stiffness $\mathbf{K}$ is in a form where all the dof $\mathbf{Q}$ are free. It needs to be modified to account for the boundary conditions.

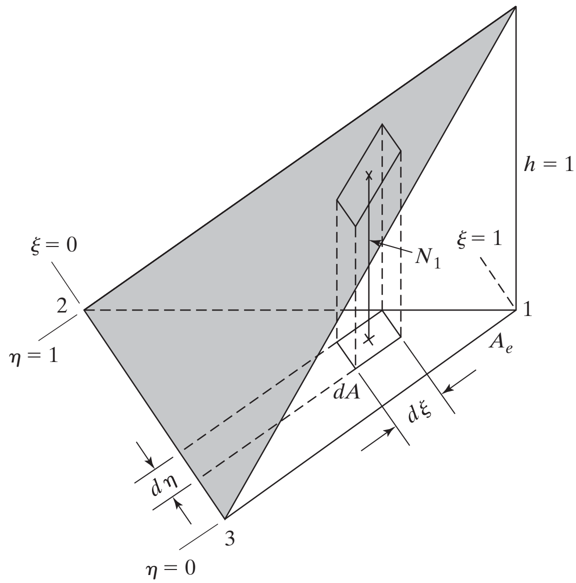
Force Terms

The **body force term** $\int_e \mathbf{u}^{\mathrm{T}} \mathbf{f} t\, dA$ appearing in the total potential energy in Eq. 6.28b is considered first. We have

$$\int_e \mathbf{u}^{\mathrm{T}} \mathbf{f} t\, dA = t_e \int_e (u f_x + v f_y)\, dA$$

Using the interpolation relations given in Eq. 6.12a, we find that

$$\begin{aligned} \int_e \mathbf{u}^{\mathrm{T}} \mathbf{f} t\, dA = {} & q_1\left(t_e f_x \int_e N_1 dA\right) + q_2\left(t_e f_y \int_e N_1\, dA\right) \\ & + q_3\left(t_e f_x \int_e N_2\, dA\right) + q_4\left(t_e f_y \int_e N_2\, dA\right) \\ & + q_5\left(t_e f_x \int_e N_3\, dA\right) + q_6\left(t_e f_y \int_e N_3\, dA\right) \end{aligned} \tag{6.33}$$



$$\int_e N_1\,dA = \frac{1}{3}A_e h = \frac{1}{3}A_e$$

$$\text{or} \int_e N_1\,dA = \int_0^1 \int_0^{1-\xi} N_1 \det \mathbf{J}\, d\eta d\xi = 2A_e \int_0^1 \int_0^{1-\xi} \xi d\eta d\xi = \frac{1}{3}A_e$$

FIGURE 6.6 Integral of a shape function.

From the definition of shape functions on a triangle, shown in Fig. 6.4, $\int_e N_1\,dA$ represents the volume of a tetrahedron with base area A_e and height of corner equal to 1 (nondimensional). The volume of this tetrahedron is given by $1/3 \times$ base area $\times$ height (Fig. 6.6) as in

$$\int_e N_i\,dA = \frac{1}{3}A_e \tag{6.34}$$

Similarly, $\int_e N_2\,dA = \int_e N_3\,dA = 1/3\,A_e$. Equation 6.33 can now be written in the form

$$\int_e \mathbf{u}^{\mathrm{T}}\mathbf{f}t\,dA = \mathbf{q}^{\mathrm{T}}\mathbf{f}^e \tag{6.35}$$

where $\mathbf{f}^e$ is the element body force vector, given as

$$\mathbf{f}^e = \frac{t_e A_e}{3}[f_x \quad f_y \quad f_x \quad f_y \quad f_x \quad f_y]^{\mathrm{T}} \tag{6.36}$$

These element nodal forces contribute to the global load vector $\mathbf{F}$. The connectivity in Table 6.1 needs to be used again to add $\mathbf{f}^e$ to the global force vector $\mathbf{F}$. The vector $\mathbf{f}^e$ is of dimension (6×1), whereas $\mathbf{F}$ is $(N \times 1)$. This assembly procedure is discussed in

Chapters 3 and 4. This is symbolically stated as

$$\mathbf{F} \leftarrow \sum_{e} \mathbf{f}^{e} \tag{6.37}$$

A **traction force** is a distributed load acting on the surface of the body. Such a force acts on edges connecting boundary nodes. A traction force acting on the edge of an element contributes to the global load vector $\mathbf{F}$. This contribution can be determined by considering the traction force term $\int \mathbf{u}^{\mathrm{T}}\mathbf{T}t\,d\ell$. Considering an edge ℓ_{1-2}, acted on by traction T_x, T_y in units of force per unit surface area (Fig. 6.7a), we get

$$\int_L \mathbf{u}^{\mathrm{T}}\mathbf{T}t\,d\ell = \int_{\ell_{1-2}} (uT_x + vT_y)t\,d\ell \tag{6.38}$$

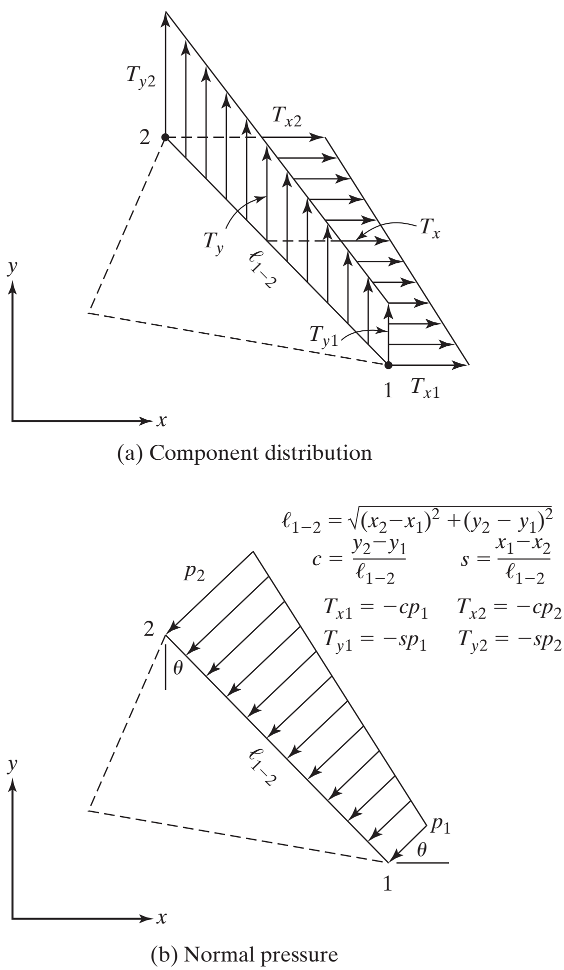


FIGURE 6.7 Traction load.

Using the interpolation relations involving the shape functions

$$\begin{aligned} u &= N_1 q_1 + N_2 q_3 \\ v &= N_1 q_2 + N_2 q_4 \\ T_x &= N_1 T_{x1} + N_2 T_{x2} \\ T_y &= N_1 T_{y1} + N_2 T_{y2} \end{aligned} \tag{6.39}$$

and noting that

$$\int_{\ell_{1-2}} N_1^2 \, d\ell = \frac{1}{3}\ell_{1-2}, \qquad \int_{\ell_{1-2}} N_2^2 \, d\ell = \frac{1}{3}\ell_{1-2}, \qquad \int_{\ell_{1-2}} N_1 N_2 \, d\ell = \frac{1}{6}\ell_{1-2}$$

$$\ell_{1-2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6.40}$$

we get

$$\int_{\ell_{1-2}} \mathbf{u}^{\mathrm{T}}\mathbf{T} t \, d\ell = [q_1, q_2, q_3, q_4]\mathbf{T}^e \tag{6.41}$$

where $\mathbf{T}^e$ is given by

$$\mathbf{T}^e = \frac{t_e \ell_{1-2}}{6}[2T_{x_1} + T_{x_2} \quad 2T_{y_1} + T_{y_2} \quad T_{x_1} + 2T_{x_2} \quad T_{y_1} + 2T_{y_2}]^{\mathrm{T}} \tag{6.42}$$

If p_1 and p_2 are pressures acting normal to the line directed to the right as we move from 1 to 2, as shown in Fig. 6.7b, then

$$T_{x1} = -cp_1, \qquad T_{x2} = -cp_2, \qquad T_{y1} = -sp_1, \qquad T_{y2} = -sp_2$$

where

$$s = \frac{(x_1 - x_2)}{\ell_{1-2}} \qquad \text{and} \qquad c = \frac{(y_2 - y_1)}{\ell_{1-2}}.$$

In Eq. 6.42, both normal and tangential distributed loads can be considered. The contributions of traction load need to be added to the global force vector **F**.

The programs given in this book expect the loads in component point load form. For distributed loads, we need to determine the equivalent point load components as illustrated in the following examples.

Example 6.4

Consider a traction load on the finite element model shown in Fig. E6.4. Determine the equivalent point loads at the nodes.

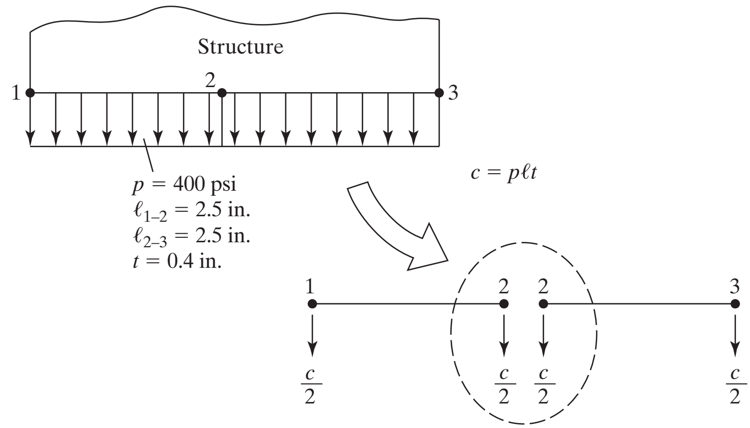


FIGURE E6.4

Solution The formula in Eq. 6.42 is applied to each edge, viz., 1–2 and 2–3, one at a time. Defining $c = (t_e)\,(\ell_{\text{edge}})\,(p\,) = 0.4 \times 2.5 \times 400 = 400$, we get, from Fig. E6.4,

$$[F_{1y} \quad F_{2y} \quad F_{3y}] = [200 \quad 400 \quad 200]\text{N}$$

These nodal loads must be input as negative quantities if the y-axis points upward. ■

Example 6.5

A two-dimensional plate is shown in Fig. E6.5. Determine the equivalent point loads at nodes 7, 8, and 9 for the linearly distributed pressure load acting on the edge 7–8–9.

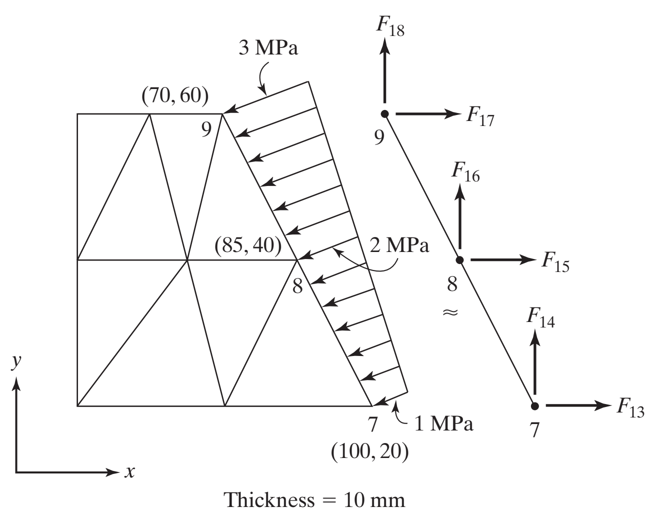


FIGURE E6.5

Solution We consider the two edges 7–8 and 8–9 separately and then merge them.

For edge 7–8

$p_1 = 1$ MPa, $p_2 = 2$ MPa, $x_1 = 100$ mm, $y_1 = 20$ mm, $x_2 = 85$ mm, $y_2 = 40$ mm,

$$\ell_{1-2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = 25 \text{ mm}$$

$$c = \frac{y_2 - y_1}{\ell_{1-2}} = 0.8, \qquad s = \frac{x_1 - x_2}{\ell_{1-2}} = 0.6$$

$$T_{x_1} = -p_1 c = -0.8, \qquad T_{y_1} = -p_1 s = -0.6, \qquad T_{x_1} = -p_2 c = -1.6,$$

$$T_{y_2} = -p_2 s = -1.2$$

$$\mathbf{T}^1 = \frac{10 \times 25}{6}[2T_{x_1} + T_{x_2} \quad 2T_{y_1} + T_{y_2} \quad T_{x_1} + 2T_{x_2} \quad T_{y_1} + 2T_{y_2}]^{\mathrm{T}}$$

$$= [-133.3 \quad -100 \quad -166.7 \quad -125]^{\mathrm{T}} \text{ N}$$

These loads add to F_{13}, F_{14}, F_{15}, and F_{16}, respectively.

For edge 8–9

$p_1 = 2$ MPa, $p_2 = 3$ MPa, $x_1 = 85$ mm, $y_1 = 40$ mm, $x_2 = 70$ mm, $y_2 = 60$ mm,

$$\ell_{1-2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = 25 \text{ mm}$$

$$c = \frac{y_2 - y_1}{\ell_{1-2}} = 0.8, \qquad s = \frac{x_1 - x_2}{\ell_{1-2}} = 0.6$$

$$T_{x_1} = -p_1 c = -1.6, \qquad T_{y_1} = -p_1 s = -1.2, \qquad T_{x_2} = -p_2 c = -2.4,$$

$$T_{y_2} = -p_2 s = -1.8$$

$$\mathbf{T}^2 = \frac{10 \times 25}{6}[2T_{x_1} + T_{x_2} \quad 2T_{y_1} + T_{y_2} \quad T_{x_1} + 2T_{x_2} \quad T_{y_1} + 2T_{y_2}]^{\mathrm{T}}$$

$$= [-233.3 \quad -175 \quad -266.7 \quad -200]^{\mathrm{T}} \text{ N}$$

These loads add to F_{15}, F_{16}, F_{17}, and F_{18}, respectively. Thus,

$$[F_{13} \quad F_{14} \quad F_{15} \quad F_{16} \quad F_{17} \quad F_{18}] = [-133.3 \quad -100 \quad -400 \quad -300 \quad -266.7 \quad -200] \text{ N}$$

■

The **point load** term is easily considered by having a node at the point of application of the point load. If i is the node at which $\mathbf{P}_i = [P_x, P_y]^{\mathrm{T}}$ is applied, then

$$\mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i = Q_{2i-1}P_x + Q_{2i}P_y \tag{6.43}$$

Thus, P_x and P_y, the x and y components of $\mathbf{P}_i$, get added to the $(2i - 1)$th and $(2i)$th components of the global force $\mathbf{F}$.

The contribution of body forces, traction forces, and point loads to the global force $\mathbf{F}$ can be represented as $\mathbf{F} \leftarrow \sum_e(\mathbf{f}^e + \mathbf{T}^e) + \mathbf{P}$.

Consideration of the strain energy and the force terms gives us the total potential energy in the form

$$\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{f} \tag{6.44}$$

The stiffness and force modifications are made to account for the boundary conditions. Using the methods presented in Chapters 3 and 4, we have

$$\mathbf{KQ} = \mathbf{F} \tag{6.45}$$

where **K** and **F** are modified stiffness matrix and force vector, respectively. These equations are solved by Gaussian elimination or other techniques, to yield the displacement vector **Q**.

Example 6.6

A CST element is shown in Fig. E6.6. The element is subjected to a body force $f_x = x^2\,\mathrm{N/m^3}$. Determine the nodal force vector $\mathbf{f}^e$. Take element thickness as 1 m.

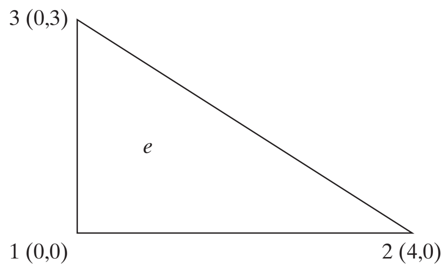


FIGURE E6.6

The work potential is $-\int_e \mathbf{f}^{\mathrm{T}}\mathbf{u}\,dV$, where $\mathbf{f}^{\mathrm{T}} = [f_x, 0]$. Substituting for $\mathbf{u} = \mathbf{N}\mathbf{q}$, we obtain the work potential in the form $-\mathbf{q}^{\mathrm{T}}\mathbf{f}^e$, where $\mathbf{f}^e = \int_e \mathbf{N}^{\mathrm{T}}\mathbf{f}\,dV$, where **N** is defined in Eq. 6.13.

All y components of $\mathbf{f}^e$ are zero. The x components at nodes 1, 2, 3 are given, respectively, by

$$\int_e \xi f_x\,dV, \quad \int_e \eta f_x\,dV, \quad \int_e (1 - \xi - \eta) f_x\,dV$$

We now make the following substitutions: $f_x = x^2$, $x = \xi x_1 + \eta x_2 + (1 - \xi - \eta)x_3 = 4\eta$, $dV = \det \mathbf{J}\,d\eta\,d\xi$, $\det \mathbf{J} = 2A_e$, and $A_e = 6$. Now, integration over a triangle is illustrated in Fig. 6.6. Thus,

$$\int_e \xi f_x\,dV = \int_0^1 \int_0^{1-\xi} \xi(16\eta^2)(12)\,d\eta\,d\xi = 3.2\,\mathrm{N}$$

Similarly, other integrations result in 9.6 N and 3.2 N. Thus,

$$\mathbf{f}^e = [3.2,\ 0,\ 9.6,\ 0,\ 3.2,\ 0]^{\mathrm{T}}\,\mathrm{N} \qquad \blacksquare$$

Integration Formula on a Triangle

Integration of various terms on a triangle has been carried out in the above example from the first steps. For other polynomial terms of the type $\xi^a\eta^b(1-\xi-\eta)^c$, a general formula is available. We have

$$\int_0^1\int_0^{1-\xi}\xi^a\eta^b(1-\xi-\eta)^c d\xi\,d\eta = \frac{a!b!c}{(a+b+c+2)} \tag{6.46}$$

where for integer values of a, b, and c[1], a ! is the factorial of a given by $a(a-1)(a-2)\ldots 1$, and $0! = 1$.

The integral on a triangle is evaluated using

$$\int_A f(\xi,\eta)\,dA = \int_0^1\int_0^{1-\xi} f(\xi,\eta)\det\mathbf{J}\,d\xi\,d\eta$$

$$= 2A\int_0^1\int_0^{1-\xi} f(\xi,\eta)\,d\xi\,d\eta \tag{6.47}$$

Equations (6.46) and (6.47) may be used for evaluating other integrals.

Galerkin Approach

Following the steps presented in Chapter 1, we introduce

$$\boldsymbol{\varphi} = [\phi_x \quad \phi_y]^{\mathrm{T}} \tag{6.48}$$

and

$$\boldsymbol{\epsilon}(\boldsymbol{\varphi}) = \left[\frac{\partial\phi_x}{\partial x} \quad \frac{\partial\phi_y}{\partial y} \quad \frac{\partial\phi_x}{\partial y}+\frac{\partial\phi_y}{\partial x}\right]^{\mathrm{T}} \tag{6.49}$$

where $\boldsymbol{\varphi}$ is an arbitrary (virtual) displacement vector, consistent with the boundary conditions. The variational form is given by

$$\int_A \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}(\boldsymbol{\varphi})t\,dA - \left(\int_A \boldsymbol{\varphi}^{\mathrm{T}}\mathbf{f}t\,dA + \int_L \boldsymbol{\varphi}^{\mathrm{T}}\mathbf{T}t\,d\ell + \sum_i \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{P}_i\right) = 0 \tag{6.50}$$

where the first term represents the internal virtual work. The expression in parentheses represents the external virtual work. On the discretized region, the previous equation becomes

$$\sum_e\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}(\boldsymbol{\varphi})t\,dA - \left(\sum_e\int_e \boldsymbol{\varphi}^{\mathrm{T}}\mathbf{f}t\,dA + \int_L \boldsymbol{\varphi}^{\mathrm{T}}\mathbf{T}t\,d\ell + \sum_i \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{P}_i\right) = 0 \tag{6.51}$$

[1]The formula holds for nonnegative real values of a, b, and c. For this general case, the integral is given by $[\Gamma(a+1)\Gamma(b+1)\Gamma(c+1)]/[\Gamma(a+b+c+3)]$, where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is the complete gamma function. For integer x, $\Gamma(x+1) = x!$

Using the interpolation steps of Eqs. 6.12–6.14, we express

$$\boldsymbol{\varphi} = \mathbf{N}\boldsymbol{\psi} \tag{6.52}$$

$$\boldsymbol{\epsilon}(\boldsymbol{\varphi}) = \mathbf{B}\boldsymbol{\psi} \tag{6.53}$$

where

$$\boldsymbol{\psi} = [\psi_1, \psi_2, \psi_3, \psi_4, \_\psi_5, \psi_6]^{\mathrm{T}} \tag{6.54}$$

represents the arbitrary nodal displacements of element e. The global nodal displacement variations $\boldsymbol{\Psi}$ are represented by

$$\boldsymbol{\Psi} = [\Psi_1, \Psi_2, \ldots, \Psi_N]^{\mathrm{T}} \tag{6.55}$$

The internal work term of the element in Eq. 6.49 can be expressed as

$$\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}(\boldsymbol{\varphi})t\,dA = \int_e \mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\psi\,t\,dA$$

Noting that all terms of $\mathbf{B}$ and $\mathbf{D}$ are constant and denoting t_e and A_e as thickness and area of element, respectively, we find that

$$\begin{aligned}\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}(\boldsymbol{\varphi})t\,dA &= \mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}t_{\mathbf{e}}\int_e dA\,\boldsymbol{\psi} \\ &= \mathbf{q}^{\mathrm{T}}t_eA_e\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\boldsymbol{\psi} \\ &= \mathbf{q}^{\mathrm{T}}\mathbf{k}^e\boldsymbol{\psi}\end{aligned} \tag{6.56}$$

where $\mathbf{k}^e$ is the element stiffness matrix given by

$$\mathbf{k}^e = t_e\,A_e\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B} \tag{6.57}$$

The material property matrix $\mathbf{D}$ is symmetric, and, hence, the element stiffness matrix is also symmetric. The element connectivity as presented in Table 6.1 is used in adding the stiffness values of $\mathbf{k}^e$ to the global locations. Thus,

$$\begin{aligned}\sum_e \int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}(\boldsymbol{\varphi})t\,dA &= \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{k}^e\psi = \sum_e \psi^{\mathrm{T}}\mathbf{k}^e\mathbf{q} \\ &= \psi^{\mathrm{T}}\mathbf{K}\mathbf{Q}\end{aligned} \tag{6.58}$$

The global stiffness matrix $\mathbf{K}$ is symmetric and banded. The treatment of external virtual work terms follows the steps involved in the treatment of force terms in the potential energy formulation, where $\mathbf{u}$ is replaced by $\boldsymbol{\varphi}$. Thus,

$$\int_e \boldsymbol{\varphi}^{\mathrm{T}}\mathbf{f}t\,dA = \boldsymbol{\Psi}^{\mathrm{T}}\mathbf{f}^e \tag{6.59}$$

which follows from Eq. 6.33, with $\mathbf{f}^e$ given by Eq. 6.36. Similarly, the traction and point load treatment follows from Eqs. 6.38 and 6.43. The terms in the variational form are given by

$$\text{Internal virtual work} = \Psi^{\mathrm{T}}\mathbf{KQ} \tag{6.60a}$$

$$\text{External virtual work} = \Psi^{\mathrm{T}}\mathbf{F} \tag{6.60b}$$

The stiffness and force matrices are modified to use the full size (all dof), using methods suggested in Chapter 3. From the Galerkin form (Eq. 6.51), the arbitrariness of Ψ gives

$$\mathbf{KQ} = \mathbf{F} \tag{6.61}$$

where **K** and **F** are modified to account for boundary conditions. Equation 6.61 turns out to be the same as Eq. 6.45, obtained in the potential energy formulation.

Stress Calculations

Since strains are constant in a CST element, the corresponding stresses are constant. The stress values need to be calculated for each element. Using the stress–strain relations in Eq. 6.6 and element strain–displacement relations in Eq. 6.25, we have

$$\boldsymbol{\sigma} = \mathbf{DBq} \tag{6.62}$$

The connectivity in Table 6.1 is again needed to extract the element nodal displacements **q** from the global displacements vector **Q**. Equation 6.62 is used to calculate the element stresses. For interpolation purposes, the calculated stress may be used as the value at the centroid of the element.

Principal stresses and their directions are calculated using Mohr's circle relationships. The program at the end of the chapter includes the principal stress calculations.

Detailed calculations in Example 6.7 illustrate the steps involved. However, it is expected that the exercise problems at the end of the chapter will be solved using a computer.

Example 6.7

For the two-dimensional loaded plate shown in Fig. E6.7, determine the displacements of nodes 1 and 2 and the element stresses using plane stress conditions. Body force may be neglected in comparison with the external forces.

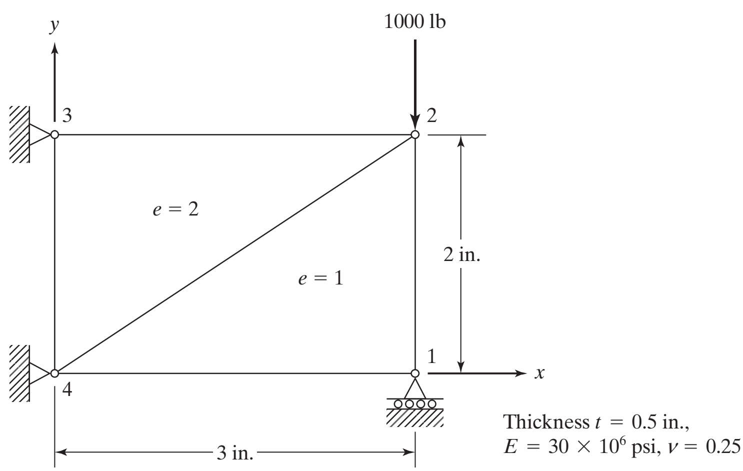


FIGURE E6.7

Solution For plane stress conditions, the material property matrix is given by

$$\mathbf{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} = \begin{bmatrix} 3.2 \times 10^7 & 0.8 \times 10^7 & 0 \\ 0.8 \times 10^7 & 3.2 \times 10^7 & 0 \\ 0 & 0 & 1.2 \times 10^7 \end{bmatrix}$$

Using the local numbering pattern used in Fig. E6.3, we establish the connectivity as follows:

	Nodes		
Element No.	1	2	3
1	1	2	4
2	3	4	2

On performing the matrix multiplication $\mathbf{DB}^e$, we get

$$\mathbf{DB}^1 = 10^7 \begin{bmatrix} 1.067 & -0.4 & 0 & 0.4 & -1.067 & 0 \\ 0.267 & -1.6 & 0 & 1.6 & -0.267 & 0 \\ -0.6 & 0.4 & 0.6 & 0 & 0 & -0.4 \end{bmatrix}$$

and

$$\mathbf{DB}^2 = 10^7 \begin{bmatrix} -1.067 & 0.4 & 0 & -0.4 & 1.067 & 0 \\ -0.267 & 1.6 & 0 & -1.6 & 0.267 & 0 \\ 0.6 & -0.4 & -0.6 & 0 & 0 & 0.4 \end{bmatrix}$$

These two relationships will be used later in calculating stresses using $\boldsymbol{\sigma}^e = \mathbf{DB}^e\mathbf{q}$. The multiplication $t_e A_e \mathbf{B}^{e^{\mathrm{T}}}\mathbf{DB}^e$ gives the element stiffness matrices,

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 7 & 8 \leftarrow \text{Global dof} \end{array}$$

$$\mathbf{k}^1 = 10^7 \begin{bmatrix} 0.983 & -0.5 & -0.45 & 0.2 & -0.533 & 0.3 \\ & 1.4 & 0.3 & -1.2 & 0.2 & -0.2 \\ & & 0.45 & 0 & 0 & -0.3 \\ & & & 1.2 & -0.2 & 0 \\ \text{Symmetric} & & & & 0.533 & 0 \\ & & & & & 0.2 \end{bmatrix}$$

$$\begin{array}{cccccc} 5 & 6 & 7 & 8 & 3 & 4 \leftarrow \text{Global dof} \end{array}$$

$$\mathbf{k}^2 = 10^7 \begin{bmatrix} 0.983 & -0.5 & -0.45 & 0.2 & -0.533 & 0.3 \\ & 1.4 & 0.3 & -1.2 & 0.2 & -0.2 \\ & & 0.45 & 0 & 0 & -0.3 \\ & & & 1.2 & -0.2 & 0 \\ \text{Symmetric} & & & & 0.533 & 0 \\ & & & & & 0.2 \end{bmatrix}$$

In the previous element matrices, the global dof association is shown on top. In the problem under consideration, Q_2, Q_5, Q_6, Q_7, and Q_8, are all zero. Using the elimination approach discussed in Chapter 3, it is now sufficient to consider the stiffnesses associated with the dof Q_1, Q_3, and Q_4. Since the body forces are neglected, the first vector has the component $F_4 = -1000$ lb. The set of equations is given by the matrix representation

$$10^7 \begin{bmatrix} 0.983 & -0.45 & 0.2 \\ -0.45 & 0.983 & 0 \\ 0.2 & 0 & 1.4 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_3 \\ Q_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -1000 \end{Bmatrix}$$

Solving for Q_1, Q_3, and Q_4, we get

$$Q_1 = 1.913 \times 10^{-5}\text{ in.} \qquad Q_3 = 0.875 \times 10^{-5}\text{ in.} \qquad Q_4 = -7.436 \times 10^{-5}\text{ in.}$$

For element 1, the element nodal displacement vector is given by

$$\mathbf{q}^1 = 10^{-5}[1.913 \quad 0 \quad 0.875 \quad -7.436 \quad 0 \quad 0]^{\mathrm{T}}$$

The element stresses $\boldsymbol{\sigma}^1$ are calculated from $\mathbf{DB}^1\mathbf{q}$ as

$$\boldsymbol{\sigma}^1 = [-93.3 \quad -1138.7 \quad -62.3]^{\mathrm{T}}\text{ psi}$$

Similarly,

$$\mathbf{q}^2 = 10^{-5}[0 \quad 0 \quad 0 \quad 0 \quad 0.875 \quad -7.436]^{\mathrm{T}}$$
$$\boldsymbol{\sigma}^2 = [93.4 \quad 23.4 \quad -297.4]^{\mathrm{T}}\text{ psi}$$

The computer results may differ slightly since the penalty approach is used in the computer program for handling boundary conditions. ■

Temperature Effects

If the distribution of the change in temperature $\Delta T(x, y)$ is known, the strain due to this change in temperature can be treated as an initial strain $\boldsymbol{\epsilon}_0$. From the theory of mechanics of solids, $\boldsymbol{\epsilon}_0$ can be represented by

$$\boldsymbol{\epsilon}_0 = [\alpha\Delta T \quad \alpha\Delta T \quad 0]^{\mathrm{T}} \tag{6.63}$$

for plane stress and

$$\boldsymbol{\epsilon}_0 = (1 + \nu)[\alpha\Delta T \quad \alpha\Delta T \quad 0]^{\mathrm{T}} \tag{6.64}$$

for plane strain. The stresses and strains are related by

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) \tag{6.65}$$

The effect of temperature can be accounted for by considering the strain energy term. We have

$$\begin{aligned} U &= \frac{1}{2}\int (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0)^{\mathrm{T}}\mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0)t\,dA \\ &= \frac{1}{2}\int (\boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon} - 2\boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}_0 + \boldsymbol{\epsilon}_0^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}_0)t\,dA \end{aligned} \tag{6.66}$$

The first term in the previous expansion gives the stiffness matrix derived earlier. The last term is a constant, which has no effect on the minimization process. The middle term, which yields the temperature load, is now considered in detail. Using the strain–displacement relationship $\boldsymbol{\epsilon} = \mathbf{Bq}$.

$$\int_A \boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{D} \boldsymbol{\epsilon}_0 t\, dA = \sum_e \mathbf{q}^{\mathrm{T}} (\mathbf{B}^{\mathrm{T}} \mathbf{D} \boldsymbol{\epsilon}_0) t_e A_e \tag{6.67}$$

This step is directly obtained in the Galerkin approach where $\boldsymbol{\epsilon}^{\mathrm{T}}$ will be $\boldsymbol{\epsilon}^{\mathrm{T}}(\boldsymbol{\phi})$ and $\mathbf{q}^{\mathrm{T}}$ will be $\boldsymbol{\psi}^{\mathrm{T}}$.

It is convenient to designate the element temperature load as

$$\boldsymbol{\Theta}^e = t_e A_e \mathbf{B}^{\mathrm{T}} \mathbf{D} \boldsymbol{\epsilon}_0 \tag{6.68}$$

where

$$\boldsymbol{\Theta}^e = [\Theta_1 \quad \Theta_2 \quad \Theta_3 \quad \Theta_4 \quad \Theta_5 \quad \Theta_6]^{\mathrm{T}} \tag{6.69}$$

The vector $\boldsymbol{\epsilon}_0$ is the strain in Eq. 6.63 or 6.64 due to the average temperature change in the element. $\boldsymbol{\Theta}^e$ represents the element nodal load contributions that must be added to the global force vector using the connectivity.

The stresses in an element are then obtained by using Eq. 6.65 in the form

$$\boldsymbol{\sigma} = \mathbf{D}(\mathbf{Bq} - \boldsymbol{\epsilon}_0) \tag{6.70}$$

Example 6.8

Consider the two-dimensional loaded plate shown in Fig. E6.5. In addition to the conditions defined in Example 6.6, there is an increase in temperature of the plate of 80°F. The coefficient of linear expansion of the material α is 7×10^{-6}/°F. Determine the additional displacements due to temperature. Also, calculate the stresses in element 1.

Solution We have $\alpha = 7 \times 10^{-6}$/°F and $\Delta T = 80$°F. So

$$\boldsymbol{\epsilon}_0 = \begin{bmatrix} \alpha \Delta T \\ \alpha \Delta T \\ 0 \end{bmatrix} = 10^{-4} \begin{bmatrix} 5.6 \\ 5.6 \\ 0 \end{bmatrix}$$

Thickness t equals 0.5, and the area of the element A is 3 in.^{2}. The element temperature loads are

$$\boldsymbol{\Theta}^1 = tA(\mathbf{DB}^1)^{\mathrm{T}} \boldsymbol{\epsilon}_0$$

where $\mathbf{DB}^1$ is calculated in the solution of Example 6.5. On evaluation, we get

$$(\boldsymbol{\Theta}^1)^{\mathrm{T}} = [11206 \quad -16800 \quad 0 \quad 16800 \quad -11206 \quad 0]^{\mathrm{T}}$$

with associated dof 1, 2, 3, 4, 7, 8, and

$$(\boldsymbol{\Theta}^2)^{\mathrm{T}} = [-11206 \quad 16800 \quad 0 \quad -16800 \quad 11206 \quad 0]^{\mathrm{T}}$$

with associated dof 5, 6, 7, 8, 3, and 4.

Picking the forces for dof 1, 3, and 4 from the previous equations, we get

$$\mathbf{F}^{\mathrm{T}} = [F_1 \quad F_3 \quad F_4] = [11206 \quad 11206 \quad 16800]$$

On solving $\mathbf{KQ} = \mathbf{F}$, we get

$$[Q_1 \quad Q_3 \quad Q_4] = [1.862 \times 10^{-3} \quad 1.992 \times 10^{-3} \quad 0.934 \times 10^{-3}] \text{ in.}$$

The displacements of element 1 due to temperature are

$$\mathbf{q}^1 = [1.862 \times 10^{-3} \quad 0 \quad 1.992 \times 10^{-4} \quad 0.934 \times 10^{-3} \quad 0 \quad 0]^{\mathrm{T}}$$

The stresses are calculated using Eq. 6.68 as

$$\boldsymbol{\sigma}^1 = (\mathbf{DB}^1)^{\mathrm{T}}\mathbf{q}^1 - \mathbf{D}\boldsymbol{\epsilon}_0$$

On substituting for the terms on the right-hand side, we get

$$\boldsymbol{\sigma}^1 = 10^4[1.204 \quad -2.484 \quad 0.78]^{\mathrm{T}} \text{ psi}$$

We note that the displacements and stresses calculated above are due to temperature change. ■

6.4 PROBLEM MODELING AND BOUNDARY CONDITIONS

The finite element method is used for computing displacements and stresses for a wide variety of problems. The physical dimensions, loading, and boundary conditions are clearly defined in some problems, similar to what we discussed in Example 6.5. In other problems, these are not clear at the outset.

An example is the problem illustrated in Fig. 6.8a. A plate with such a loading can exist anywhere in space. Since we are interested in the deformation of the body, the symmetry of the geometry and the symmetry of the loading can be used effectively. Let x and y represent the axes of symmetry as shown in Fig. 6.8b. The points along the x-axis move along x and are constrained in the y direction, and points along the y-axis are constrained along the x direction. This suggests that the part, which is one-quarter of the full area, with the loading and boundary conditions as shown is all that is needed to solve the deformation and stresses.

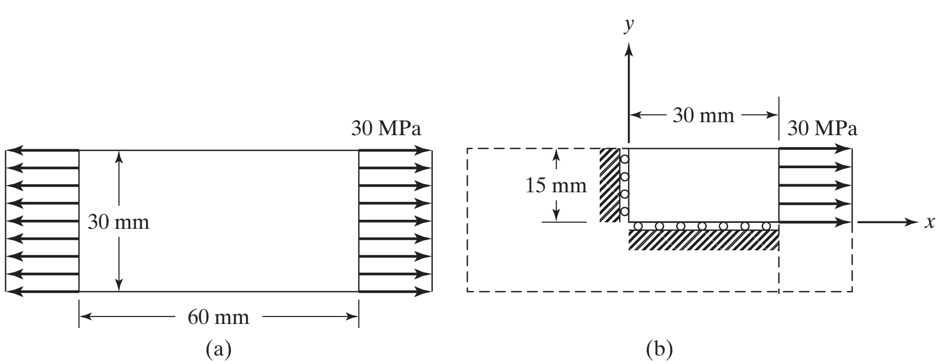


FIGURE 6.8 Rectangular plate.

A linearly distributed load symmetric with respect to a plane can be divided into symmetric–symmetric and symmetric–antisymmetric loading as shown in Fig. 6.9. The symmetric–symmetric part can be treated as discussed above. The symmetric–antisymmetric part can be treated with boundary conditions as shown. Displacements and stresses need to be computed using the principle of superposition. The values for the entire body need to consider the geometry of deformation for symmetry and antisymmetry.

As another example, consider an octagonal pipe under internal pressure, shown in Fig. 6.10a. By symmetry, we observe that it is sufficient to consider the 22.5° segment shown in Fig. 6.10b. The boundary conditions require that points along x and n are constrained normal to the two lines, respectively. Note that for a *circular* pipe under internal or external pressure, by symmetry, all points move radially. In this case, any radial segment may be considered. The boundary conditions for points along the x-axis in Fig. 6.10b are easily considered by using the penalty approach (discussed in Chapter 3). The boundary conditions for points along the inclined direction n, which are considered perpendicular to n, are now discussed in detail. If node i with dof Q_{2i-1} and Q_{2i} moves along n as seen in Fig. 6.11 and θ is the angle of inclination of n with respect to x-axis, we get

$$Q_{2i-1}\sin\theta - Q_{2i}\cos\theta = 0 \tag{6.71}$$

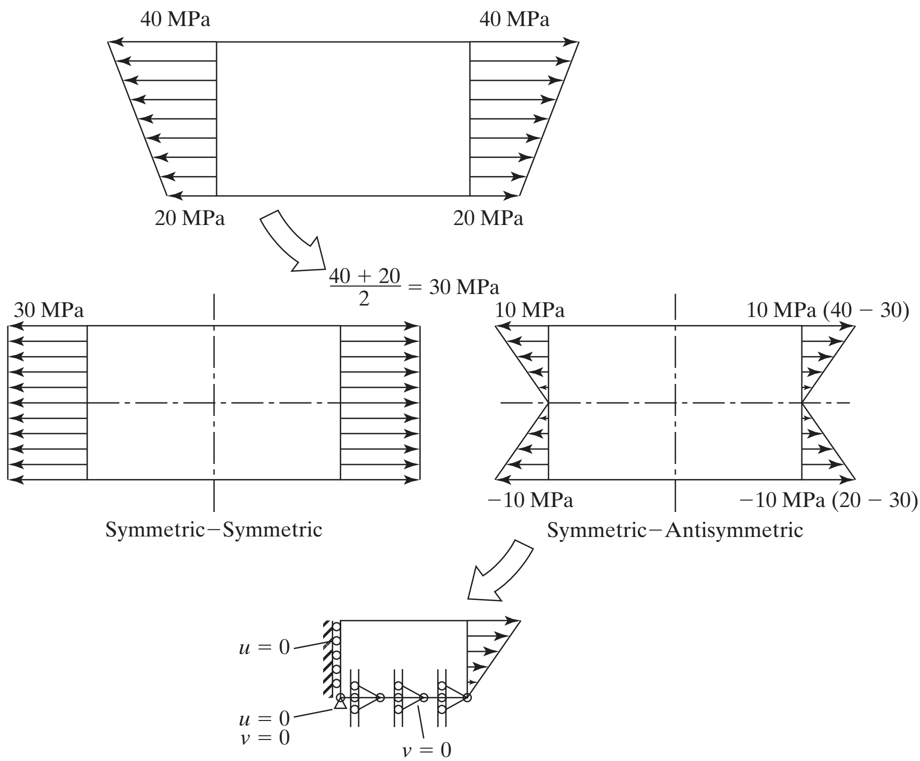


FIGURE 6.9 Symmetric–antisymmetric components.

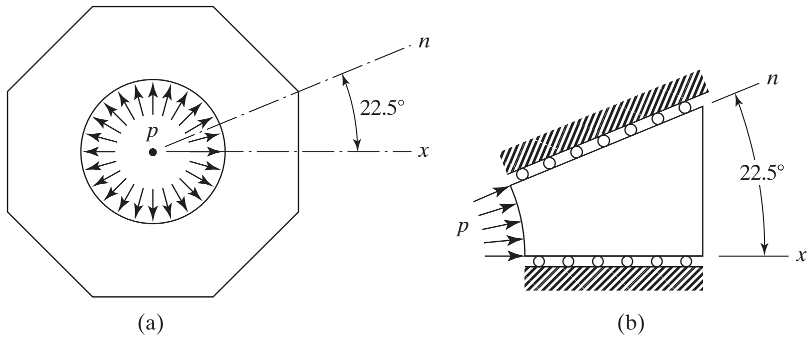


FIGURE 6.10 Octagonal pipe.

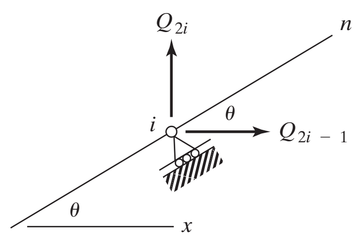


FIGURE 6.11 Inclined roller support.

This boundary condition is seen to be a multipoint constraint (MPC), which is discussed in Chapter 3. Using the penalty approach (presented in Chapter 3), this amounts to adding a term to the potential energy as in

$$\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F} + \frac{1}{2}C(Q_{2i-1}\sin\theta - Q_{2i}\cos\theta)^2 \tag{6.72}$$

where C is a large number.

The squared term in Eq. 6.72 can be written in the form

$$\frac{1}{2}C(Q_{2i-1}\sin\theta - Q_{2i}\cos\theta)^2 = \frac{1}{2}[Q_{2i-1},\ Q_{2i}]\begin{bmatrix} C\sin^2\theta & -C\sin\theta\cos\theta \\ -C\sin\theta\cos\theta & C\cos^2\theta \end{bmatrix}\begin{Bmatrix} Q_{2i-1} \\ Q_{2i} \end{Bmatrix} \tag{6.73}$$

The terms $C\sin^2\theta$, $-C\sin\theta\cos\theta$, and $C\cos^2\theta$ get added to the global stiffness matrix, for every node on the incline, and the new stiffness matrix is used to solve for the displacements. Note that these modifications can also be directly obtained from Eq. 3.85 by substituting $\beta_0 = 0$, $\beta_1 = \sin\theta$, and $\beta_2 = -\cos\theta$. The contributions to the banded stiffness matrix $\mathbf{S}$ are made in the locations $(2i - 1, 1)$, $(2i - 1, 2)$, and $(2i, 1)$ by adding $C\sin^2\theta$, $-C\sin\theta\cos\theta$, and $C\cos^2\theta$, respectively.

Some General Comments on Dividing into Elements

When dividing an area into triangles, avoid large aspect ratios. Aspect ratio is defined as the ratio of maximum to minimum characteristic dimensions. Observe that the best elements are those that approach an equilateral triangular configuration. Such configurations are not usually possible. A good practice may be to choose corner angles in the range of 30–120°.

In problems where the stresses change widely over an area, such as in notches and fillets, it is good practice to decrease the size of elements in that area to capture the stress variations. The CST, in particular, gives constant stresses on the element. This suggests that smaller elements will better represent the distribution. Better estimates of maximum stress may be obtained even with coarser meshes by plotting and extrapolating. For this purpose, the constant element stresses may be interpreted as the values at centroids of the triangle. A method for evaluating nodal values from constant element values is presented in the postprocessing section of Chapter 12.

Coarse meshes are recommended for initial trials to check data and reasonableness of results. Errors may be fixed at this stage, before running larger numbers of elements. Increasing the number of elements in those regions where stress variations are high should give better results. This is called *convergence*. One should get a feel for convergence by successively increasing the number of elements in finite element meshes.

6.5 PATCH TEST AND CONVERGENCE

When a finite element problem is solved repeatedly by increasing the number of elements (mesh refinement), we expect convergence which implies that the solution approaches the exact solution. A test proposed for ensuring convergence is called the *patch test*.

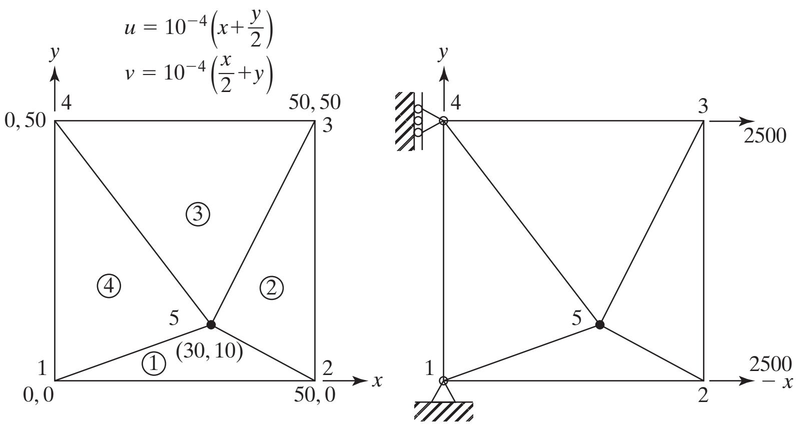
Patch Test

We follow the steps given below to perform a patch test.

- Define a small number of connected elements called a *patch*.
- Have at least one node internal to the *patch*.
- Restrain the patch by subjecting it to minimum displacement boundary conditions to eliminate all rigid body motions.
- Apply consistent equilibrium loads or displacements on the boundary nodes, which should result in a constant stress within the patch.
- At the interior node, no loads are applied, displacements are not specified.
- Compute the displacements, strains, and stresses.

The patch test is *passed* if the computed stresses and strains match the exact values within the limit of computer precision.

A successful patch test is a check for validation of the element. The test shows that the element is able to display a state of constant strain or stress, rigid body motion without strain, and its compatibility with neighboring elements.



(a) Displacement test

BC	Node	u	v
	1	0	0
	2	0.005	0.0025
	3	0.0075	0.0075
	4	0.0025	0.0050

$E = 10^6$ $\quad\nu = 0.25$

Exact solution

$u_5 = 0.0035$ $\quad v_5 = 0.0025$

Plane stress $\quad \sigma_x = 133.3$, $\sigma_y = 133.3$, $\tau_{xy} = 40$

Plane strain $\quad \sigma_x = 160$, $\sigma_y = 160$, $\tau_{xy} = 40$

(b) Load test

Distributed load 10 on edge 2–3

Thickness 10

$$\text{Point loads} = \frac{50 \times 10 \times 10}{2} = 2500$$

Exact solution

$\sigma_x = 10$
$\sigma_y = 10$
$\tau_{xy} = 0$

FIGURE 6.12 Patch test.

Figure 6.12a and b shows possible patch tests for triangular element for displacement and load conditions. The boundary conditions and exact solutions are given in the figure. The computation is easily carried out by preparing data for program CST.

6.6 ORTHOTROPIC MATERIALS

Certain naturally occurring materials such as crystals of topaz and barite are orthotropic. Wood may also be considered to be orthotropic as a first approximation. Unidirectional fiber-reinforced composites also exhibit orthotropic behavior. Orthotropic materials have three mutually perpendicular planes of elastic symmetry. We will denote 1, 2, and 3 as the *principal material axes* that are normal to the planes of symmetry. For example, Fig. 6.13 shows a cross section of a tree, with 1 being the axis along the wood fibers (grain), 2 being the axis tangential to the annual rings, and 3 the axis along the radial

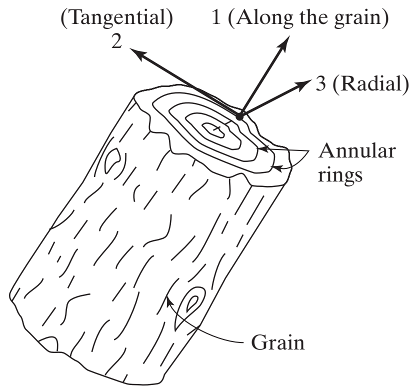


FIGURE 6.13 Wood as an orthotropic material.

direction. The generalized Hooke's law as referred to coordinate system 1, 2, 3 can be written as[2]

$$
\begin{aligned}
\epsilon_1 &= \frac{1}{E_1}\sigma_1 - \frac{\nu_{21}}{E_2}\sigma_2 - \frac{\nu_{31}}{E_3}\sigma_3, & \gamma_{23} &= \frac{1}{G_{23}}\tau_{23} \\
\epsilon_2 &= -\frac{\nu_{12}}{E_1}\sigma_1 + \frac{1}{E_2}\sigma_2 - \frac{\nu_{32}}{E_3}\sigma_3, & \gamma_{13} &= \frac{1}{G_{13}}\tau_{13} \\
\epsilon_3 &= -\frac{\nu_{13}}{E_1}\sigma_1 - \frac{\nu_{23}}{E_2}\sigma_2 + \frac{1}{E_3}\sigma_3, & \gamma_{12} &= \frac{1}{G_{12}}\tau_{12}
\end{aligned}
\tag{6.74}
$$

where E_1, E_2, and E_3 are the Young's moduli along the principal material axes; ν_{12} is the Poisson's ratio that characterizes the decrease in the 2-direction during tension applied in the 1-direction; ν_{21} is the Poisson's ratio that characterizes the decrease in the 1-direction due to tension in the 2-direction and so on, and G_{23}, G_{13}, and G_{12} are the shear moduli that characterize changes of angles between principal directions 2 and 3, 1 and 3, and 1 and 2, respectively. Due to the symmetry of Eq. 6.72, the following relations are obtained:

$$
E_1\nu_{21} = E_2\nu_{12}, \qquad E_2\nu_{32} = E_3\nu_{13}, \qquad E_3\nu_{23} = E_1\nu_{31} \tag{6.75}
$$

Thus, there are nine independent material constants. In this chapter, we will consider only the problem of plane stress. Thus, we consider a thin body that lies in the 1, 2 plane. Examples of such thin bodies are shown in Fig. 6.14a and b. Figure 6.14a shows how thin planks are obtained from a tree. Figure 6.14b shows a unidirectional composite that can be modeled as a plane stress orthotropic problem. In actual design, many layers of these unidirectional composites are stacked at different fiber orientations to form a *laminate*. A single-layer composite may be viewed as a building block for laminate constructions. In a unidirectional composite, the Young's modulus along the fibers is greater

[2] S. G. Lekhnitskii, *Anisotropic Plates*, NY: Gordon and Breach Science Publishers (1968) (translated by S. W. Tsai and T. Cheron).

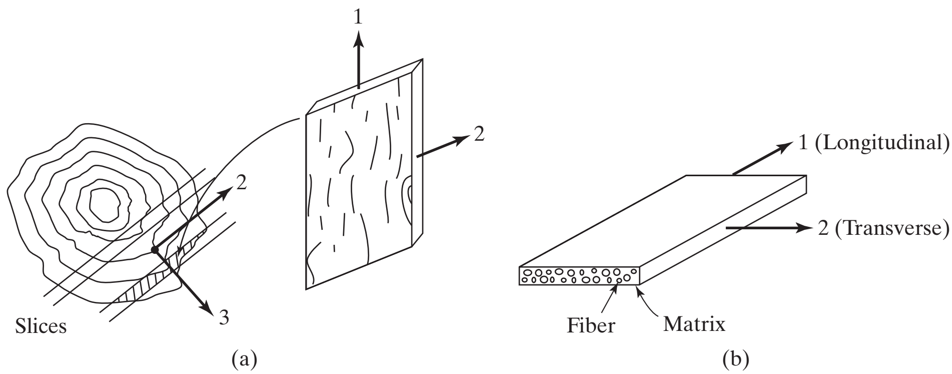


FIGURE 6.14 Plane stress orthotropic bodies: (a) wood planks and (b) unidirectional composites.

than that across. That is, $E_1 > E_2$. The axis 1 is often referred to as the *longitudinal axis* and 2 is referred to as the *transverse axis*. In plane stress, all stresses and displacements are assumed to be averaged across the thickness and are consequently only functions of 1, 2. The loading is confined to be in the 1, 2 plane.

Neglecting the z-component stresses, we have, from Eq. 6.74,

$$\epsilon_1 = \frac{1}{E_1}\sigma_1 - \frac{\nu_{21}}{E_2}\sigma_2, \qquad \epsilon_2 = -\frac{\nu_{12}}{E_1}\sigma_1 + \frac{1}{E_2}\sigma_2, \qquad \gamma_{12} = \frac{1}{G_{12}}\tau_{12} \tag{6.76}$$

These equations can be inverted to express stress in terms of strain as

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} \dfrac{E_1}{1 - \nu_{12}\nu_{21}} & \dfrac{E_1\nu_{21}}{1 - \nu_{12}\nu_{21}} & 0 \\ \dfrac{E_2\nu_{12}}{1 - \nu_{12}\nu_{21}} & \dfrac{E_2}{1 - \nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{6.77}$$

The 3×3 coefficient matrix in Eq. 6.77 will be denoted by $\mathbf{D}^m$, the superscript m denoting the material axes. Thus, $\mathbf{D}_{11}^m = E_1/(1 - \nu_{12}\nu_{21})$, $D_{33}^m = G_{12}$ and so on. $\mathbf{D}^m$ is symmetric since $E_1\nu_{21} = E_2\nu_{12}$. Four independent constants are involved here.

When an orthotropic plate is loaded parallel to its material axes, it results only in normal strains and not in shear strains. When the load is not parallel to any of its material axes, it results in both normal and shear strains. To analyze general problems of this sort, we will consider an orthotropic material with its material axes oriented at an angle θ with the *global* x-, y-axes as shown in Fig. 6.15. Note that θ is the angle measured counterclockwise from the x-axis to the 1-axis. A transformation matrix $\mathbf{T}$ is introduced as

$$\mathbf{T} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -2\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{6.78}$$

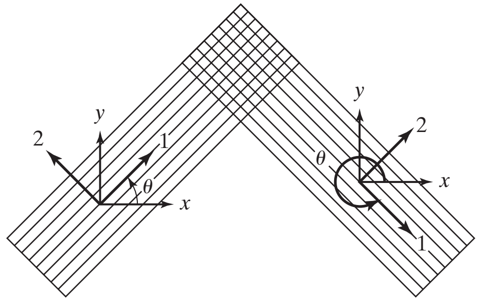


FIGURE 6.15 Orientation of material axes with respect to global axes; θ is the counterclockwise angle from x-axis to 1-axis. Note: $\theta = 330°$ is equivalent to $\theta = -30°$.

The relations between the stresses (or strains) in the material coordinate system and the global coordinate system are

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}, \qquad \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \dfrac{1}{2}\gamma_{12} \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \dfrac{1}{2}\gamma_{xy} \end{Bmatrix} \tag{6.79}$$

The important relation we need is the **D** matrix, which relates stress and strain in the global system as

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{6.80}$$

It can be shown[3] that the **D** matrix is related to the $\mathbf{D}^m$ matrix as

$$\begin{aligned}
D_{11} &= D_{11}^m \cos^4\theta + 2(D_{12}^m + 2D_{33}^m)\sin^2\theta\cos^2\theta + D_{22}^m \sin^4\theta \\
D_{12} &= (D_{11}^m + D_{22}^m - 4D_{33}^m)\sin^2\theta\cos^2\theta + D_{12}^m(\sin^4\theta + \cos^4\theta) \\
D_{13} &= (D_{11}^m - D_{12}^m - 2D_{33}^m)\sin\theta\cos^3\theta + (D_{12}^m - D_{22}^m + 2D_{33}^m)\sin^3\theta\cos\theta \\
D_{22} &= D_{11}^m \sin^4\theta + 2(D_{12}^m + 2D_{33}^m)\sin^2\theta\cos^2\theta + D_{22}^m \cos^4\theta \\
D_{23} &= (D_{11}^m - D_{12}^m - 2D_{33}^m)\sin^3\theta\cos\theta + (D_{12}^m - D_{22}^m + 2D_{33}^m)\sin\theta\cos^3\theta \\
D_{33} &= (D_{11}^m + D_{22}^m - 2D_{12}^m - 2D_{33}^m)\sin^2\theta\cos^2\theta + D_{33}^m(\sin^4\theta + \cos^4\theta)
\end{aligned} \tag{6.81}$$

Implementation of Eq. 6.81 into the finite element program CST2 is straightforward. The existing isotropic **D** matrix is replaced by that given in Eq. 6.81. The angle θ will be assumed to be constant within each finite element, although the angle can vary from one element to another. This variation in θ makes it possible to tailor the material so as to be most effective in resisting the loads. After solving the equations and obtaining the stresses in the global coordinate system, the stresses in the material coordinate system can be obtained using Eq. 6.79 and then inserted into an appropriate failure theory to determine the factor of safety.

[3] Agarwal B. D. and L. J. Broutman, *Analysis and Performance of Fiber Composites*, New York: John Wiley & Sons, Inc. (1980).

TABLE 6.2 Typical Properties for Some Orthotropic Materials

Material	E_1, 10^6 psi	E_1/E_2	ν_{12}	E_1/G_{12}	α_1, $10^{-6}/°\mathrm{F}$	α_2, $10^{-6}/°\mathrm{F}$
Balsa wood	0.125	20.0	0.30	29.0	—	—
Pine wood	1.423	23.8	0.24	13.3	—	—
Plywood	1.707	2.0	0.07	17.1	—	—
Boron epoxy	33.00	1.571	0.23	4.714	3.20	11.0
S-glass epoxy	7.50	4.412	0.25	9.375	3.50	11.0
Graphite (Thornel 300)	23.06	14.587	0.38	24.844	0.025	11.2
Kevlar-49	12.04	14.820	0.34	39.500	–1.22 to –1.28	19.4

Temperature Effects

We have studied how temperature strains are handled for isotropic materials. The stress–strain law is of the form $\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^0)$. This same relation also holds for orthotropic materials. In material coordinates, an increase in temperature ΔT will cause normal strains, but no shearing strain. Thus, $\epsilon_1^0 = \alpha_1 \Delta T$ and $\epsilon_2^0 = \alpha_2 \Delta T$. The $\mathbf{T}$-matrix in Eq. 6.78 can be used to transform the coefficients of thermal expansion as

$$\begin{Bmatrix} \alpha_x \\ \alpha_y \\ \dfrac{1}{2}\alpha_{xy} \end{Bmatrix} = \mathbf{T}\begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ 0 \end{Bmatrix} \tag{6.82}$$

The initial strain vector $\boldsymbol{\epsilon}^0$ is now given by

$$\begin{Bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} = \begin{Bmatrix} \alpha_x & \Delta T \\ \alpha_y & \Delta T \\ \alpha_{xy} & \Delta T \end{Bmatrix} \tag{6.83}$$

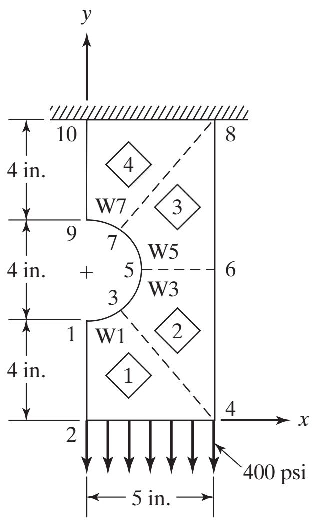
Typical values of the elastic constants for some orthotropic materials, such as wood materials and unidirectional composites, are given in Table 6.2. The unidirectional composites are made from embedding fibers in a matrix. In the table, the matrix is an epoxy resin with $E \approx 0.5 \times 10^6$ psi, $\nu = 0.3$.

Example 6.9

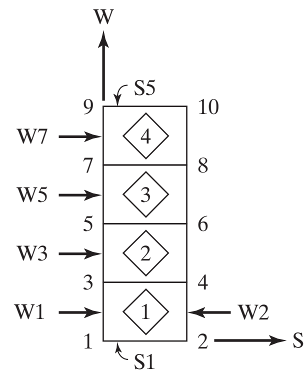
This example shows how more detailed models can be analyzed using program CST with pre- and postprocessing programs of Chapter 12.

Consider the problem shown in Fig. E6.9a. It is necessary to determine the location and magnitude of the maximum y stress in the plate.

Use of the mesh generation program MESHGEN requires mapping the region into a checkerboard and specifying the number of subdivisions for discretization. The detailed explanation of using MESHGEN is given in Chapter 12. Here, the emphasis is only on using the program to generate input data for CST. Thus, using the checkerboard in Fig. E6.9b, a 36-node, 48-element mesh is created as shown in Fig. E6.9c. Program PLOT2D has been used to generate the plot after executing MESHGEN. A text editor is then used to define



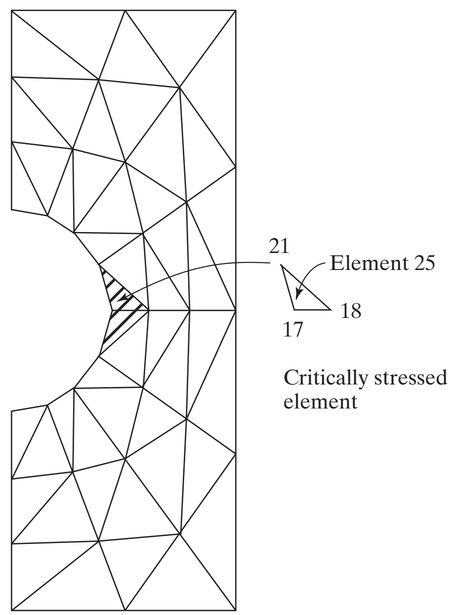
(a) Region



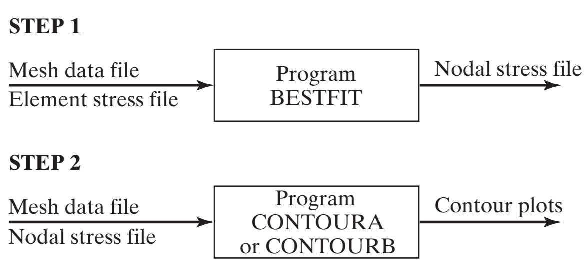
(b) Block diagram

STEPS

1. Divide the region into four-sided subregions
2. Create a block diagram
3. Number the blocks, corner nodes and sides on the block diagram
4. Transfer these numbers onto the region
5. Create input file and run MESHGEN.BAS
6. Run PLOT2D.BAS
7. Use text editor to prepare cst. inp see front pages of book for the structure of input file
8. Run CST
9. Run BESTFIT followed by CONTOURA. BAS and CONTOURB.BAS



(c) Finite element mesh viewed using PLOT2D.BAS



(d) Contour plotting using programs BESTFIT and CONTOUR

FIGURE E6.9

the boundary conditions, loads, and material properties. The MESHGEN input file is listed subsequently. MESHGEN is run using this input file. The output of MESHGEN is then edited using any text editor. The changes and additions are shown in bold face in the CST input file, which follows the MESHGEN input file listing. Note that the structure of input files is shown in the inside front cover of the book. The resulting data file is input into CST. In summary, the order in which programs are executed is MESHGEN, PLOT2D, text editor, and CST.

From the output, we note the maximum y stress to be 1768.0 psi occurring in the hatched region in Fig. E6.9c.

The reader is urged to follow these steps, which will help in the solution of complex problems with less effort. Programs BESTFIT and CONTOUR A or CONTOUR B can be used at this stage for obtaining nodal stresses and contour plots, as discussed in Chapter 12. Contour plotting with programs BESTFIT and CONTOUR is shown schematically in Fig. E6.9d. *Also, the stresses in the elements may be considered to be accurate at the centroids of the elements and can be extrapolated to obtain the maximum stresses.* (*See Fig. E7.4c or Fig. E8.3b for such extrapolation.*) ■

Input Data/Output

```
INPUT TO MESHGEN FOR EXAMPLE 6.9
MESH GENERATION
EXAMPLE 6.9
Number of Nodes per Element <3 or 4>
   3
BLOCK DATA                NS=#S-Spans
NS   NW  NSJ              NW=#W-Spans
1     4   0               NSJ=#PairsOfEdgesMerged
SPAN DATA
S-Span#   #Div (for each S-Span/ Single division = 1)
1         3
W-Span#   #Div (for each W-Span/ Single division = 1)
1         2
2         2
3         2
4         2
BLOCK MATERIAL DATA
Block#   Material  (Void => 0 Block# = 0 completes this data)
0
BLOCK CORNER DATA
Corner#   X-Coord   Y-Coord   (Corner# = 0 completes this data)
1         0         4
2         0         0
3         1.4142    4.5858
4         5         0
5         2         6
6         5         6
7         1.4142    7.4142
```

```
8          5         12
9          0         8
10         0         12
0
MID POINT DATA FOR CURVED OR GRADED SIDES
S-Side#    X-Coord   Y-Coord   (Sider# = 0 completes this data)
0
W-Side#    X-Coord   Y-Coord   (Sider# = 0 completes this data)
1          0.7654    4.1522
3          1.8478    5.2346
5          1.8478    4.7654
7          0.7654    7.8478
0
MERGING SIDES (Node1 is the lower number)
Pair#   Sid1Nod1   Sid1Nod2   Sid2Nod1   Sid2Nod2
```

```
OUTPUT FROM MESHGEN FOR EXAMPLE 6.9 (Edit as indicated)
Program MESHGEN - CHANDRUPATLA & BELEGUNDU
EXAMPLE 6.9
NN   NE   NM   NDIM   NEN   NDN
36   48   1    2      3     2
ND  NL   NMPC
0    0    0     <= Edit in the correct values
Nod#    X-Coord   Y-Coord
1       0         4
2       0         2.6666667
3       0         1.3333333
...     ...       ...
35      0         10.666667
36      0         12
Elem#  Node1   Node2   Node3   Mat#   Th    TempRise
1       1       2       5       1     0.4    0 <= Edit Mat#, Thickness,
2       6       5       2       1     0.4    0    TempRise as applicable
3       2       3       6       1     0.4    0
...    ...     ...     ...     ...    ...   ...
...    ...     ...     ...     ...    ...   ...
46      35      34      30      1     0.4    0
47      31      32      36      1     0.4    0
48      36      35      31      1     0.4    0
DOF#    Displ.
55      0       <= Add specified displacements
56      0
63      0
64      0
71      0
72      0
```

continued

```
DOF#   Load
8       -200     <= Add applied component loads
16      -400
24      -200
MAT#     E      Nu    Alpha
1        30e6   0.3   0         <= Add material# and properties
B1       i      B2    j         B3 <== Multipoint constr. <B1*Qi+B2*Qj=B3>
```

```
INPUT TO CST
2D STRESS ANALYSIS USING CST
EXAMPLE 6.7
NN   NE   NM   NDIM   NEN   NDN
4     2   1    2       3     2
ND   NL   NMPC
5     1    0
Node#    X    Y
1        3    0
2        3    2
3        0    2
4        0    0
Elem#    N1   N2  N3  Mat#   Thickness   TempRise
1         4    1   2   1      0.5          0
2         3    4   2   1      0.5          0
DOF#   Displacement
2       0
5       0
6       0
7       0
8       0
DOF#   Load
4      -1000
MAT#   E        Nu   Alpha
1      3.00E+07  0.25  1.20E-05
B1     i      B2  j     B3  (Multi-point constr.B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM CST
Program CST - Plane Stress Analysis
EXAMPLE 6.7
Node#   X-Displ          Y-Displ
1       1.90756E-05     -5.86181E-09
2       8.73255E-06     -7.416E-05
3       1.92157E-09     -1.18396E-09
4       -1.92157E-09    -9.70898E-11
Elem#   SX          SY          Txy         S1          S2           Angle X to S1
1       -93.12      -1135.59    -62.08      -89.44      -1139.28     -3.40
2       93.12       23.26       -296.61     356.85      -240.47      -41.64
```

```
DOF#    Reaction
2       820.6532
5       -269.0202
6       165.7542
7       269.0202
8       13.5926
```

PROBLEMS

6.1. The nodal coordinates of the triangular element are shown in Fig. P6.1. At the interior point P, the x-coordinate is 3.5 and $N_1 = 0.25$. Determine N_2, N_3, and the y-coordinate at point P.

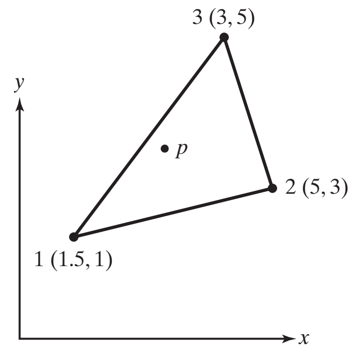


FIGURE P6.1

6.2. Determine the Jacobian for the $(x, y) - (\xi, \eta)$ transformation for the element shown in Fig. P6.2. Also, find the area of the triangle.

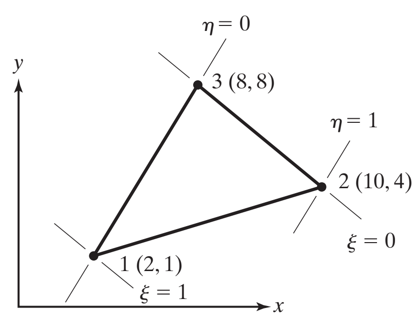


FIGURE P6.2

6.3. For point P located inside the triangle shown in Fig. P6.3, the shape functions N_1 and N_2 are 0.2 and 0.3, respectively. Determine the x- and y-coordinates of point P.

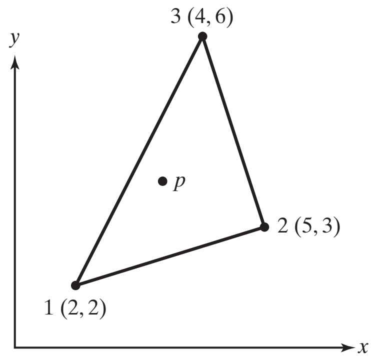


FIGURE P6.3

6.4. In Problem 6.1, determine the shape functions using the area coordinate approach. [*Hint:* Use area $= 0.5(x_{13}y_{23} - x_{23}y_{13})$ for triangle 123.]

6.5. For the triangular element shown in Fig. P6.5, obtain the strain–displacement relation matrix **B** and determine the strains ϵ_x, ϵ_y, and γ_{xy}.

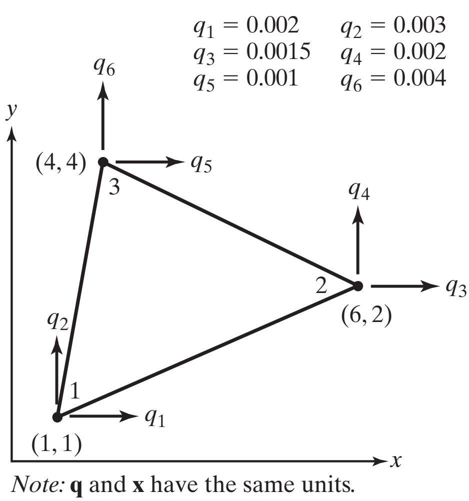


FIGURE P6.5

6.6. Figure P6.6 shows a two-dimensional region modeled with 10 CST elements.

(a) Determine the bandwidth NBW (also referred to as the "half-bandwidth").

(b) If the multipoint constraint $Q_2 = Q_{20}$ is imposed (2 and 20 are dof numbers corresponding to x displacement of node 2 and y displacement of node 10, respectively), what is the new value of NBW?

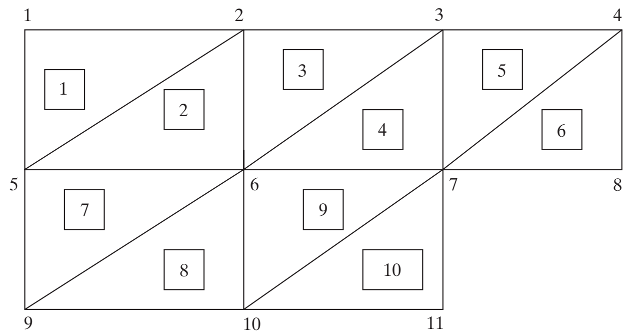


FIGURE P6.6

6.7. Indicate all the mistakes in the following finite element models with CST elements (Fig. P6.7):

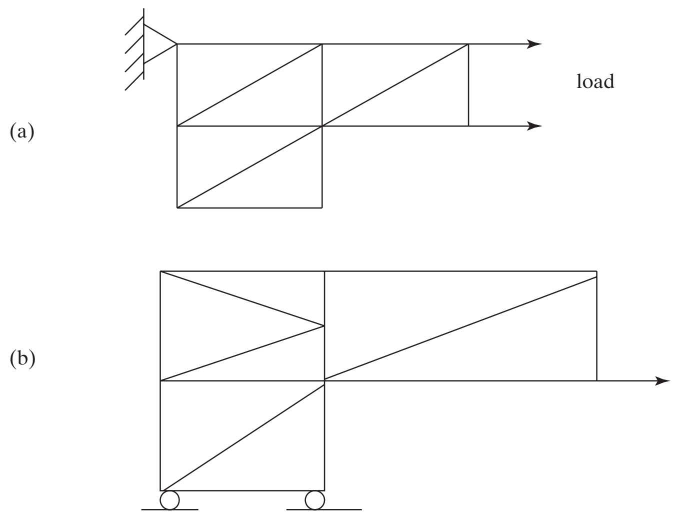


FIGURE P6.7

6.8. Consider the plane stress/plane strain finite element model shown in Fig. P6.8. Answer the following: (a) Is the model valid? Justify. (b) Does the answer depend on the type of elements used to mesh the body?

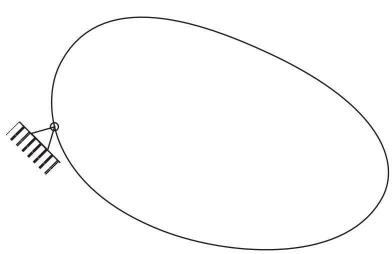


FIGURE P6.8

6.9. For a two-dimensional triangular element, the stress–displacement matrix **DB** appearing in $\boldsymbol{\sigma} = \mathbf{DBq}$ is given by

$$\mathbf{DB} = \begin{bmatrix} 2700 & 2100 & -1400 & 1100 & -4300 & 900 \\ 5600 & 3900 & 4000 & 2500 & -1400 & 1000 \\ 1900 & 2300 & -3800 & 1600 & 2100 & 4300 \end{bmatrix} \text{N/mm}^3$$

If the coefficient of linear expansion is $9.8 \times 10^{-6}/°\text{C}$, the temperature rise of the element is 150°C, and the volume of the element is 30 mm^3, determine the equivalent temperature load θ for the element.

6.10. The two-element configuration shown in Fig. P6.10 is subjected to body force $\mathbf{f} = [f_x \quad f_y]^T = [x \quad 0]^T$. Determine the 8×1 global load vector **F** using Eq. 6.33. Also find the centroidal values of f_x and f_y using Eq. 6.36 and compare.

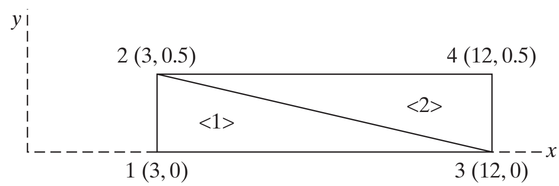


FIGURE P6.10

6.11. For the configuration shown in Fig. P6.11, determine the deflection at the point of load application using a one-element model. If a mesh of several triangular elements is used, comment on the stress values in the elements close to the tip.

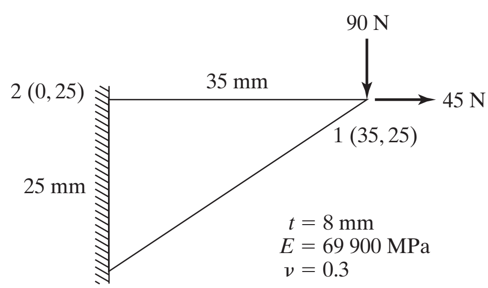


FIGURE P6.11

6.12. Determine the bandwidth for the two-dimensional region for the triangular element division with the node numbering shown in Fig. P6.12. How do you proceed to decrease the bandwidth?

6.13. Consider the four-element CST model in Fig. P6.13 subjected to a body force $f = y^2\text{N/m}^3$ in the y direction. Assemble the global load vector $\mathbf{F}_{18\times1}$ for the model.

6.14. Assemble the load vector $\mathbf{F}_{6\times1}$ at the three nodes on the inner boundary, which is subjected to a pressure $p = 1.0$ MPa. (See Fig. P6.14.)

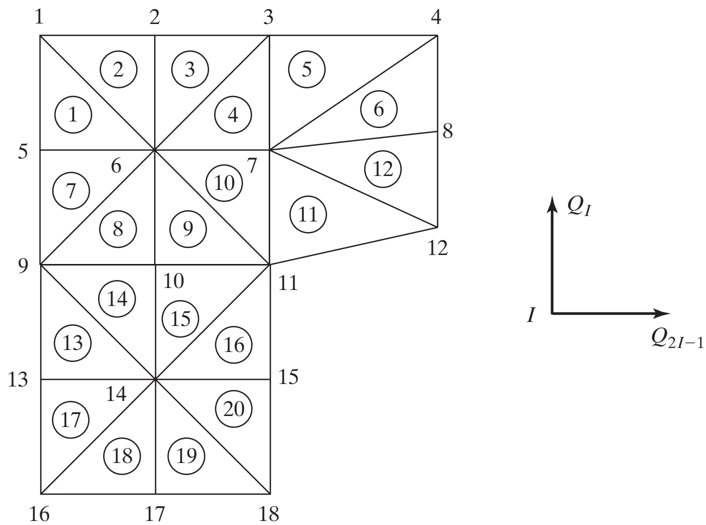
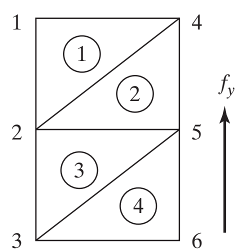


FIGURE P6.12



Coordinates, m

Node	x	y
1	0	3
2	0	2
3	0	1
4	2.25	3
5	2.25	2
6	2.25	1

FIGURE P6.13

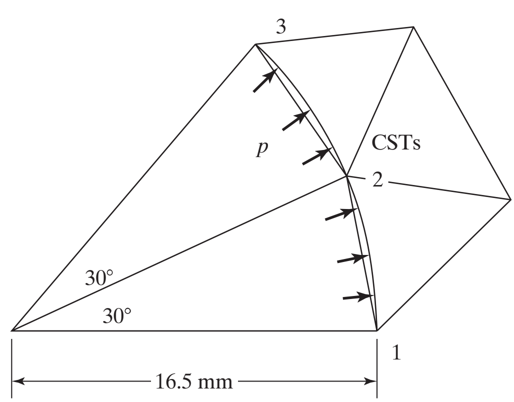


FIGURE P6.14

6.15. Consider the three-node triangular element shown in Fig. P6.15. Express the integral for area moment of inertia $I = \int_e y^2 dA$ as

$$I = \mathbf{y}_e^{\mathrm{T}}[\mathbf{R}]\mathbf{y}_e$$

where $\mathbf{y}_e = [y_1, y_2, y_3]^{\mathrm{T}} =$ a vector of y-coordinates of the three nodes, and $\mathbf{R}$ is a 3×3 matrix. (*Hint*: Interpolate y using shape functions N_i.)

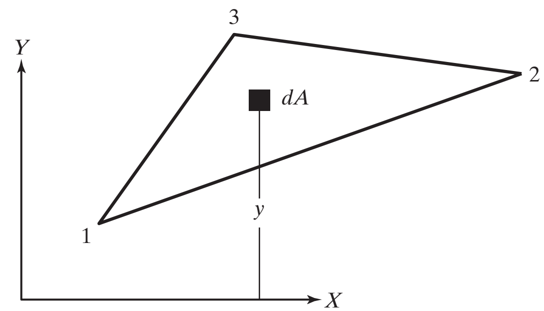


FIGURE P6.15

6.16. Compute the integral $I = \int_e N_1 N_2 N_3 \, dA$, where $N_i, i = 1, 2, 3$, are the linear shape functions for a three-node CST element.

6.17. Solve the plane stress problem in Fig. P6.17 using three different mesh divisions. Compare your deformation and stress results with values obtained from elementary beam theory. Try 2×2, 3×3, and 4×8 configurations with each rectangle divided into two triangles. Use MESHGEN for higher divisions.

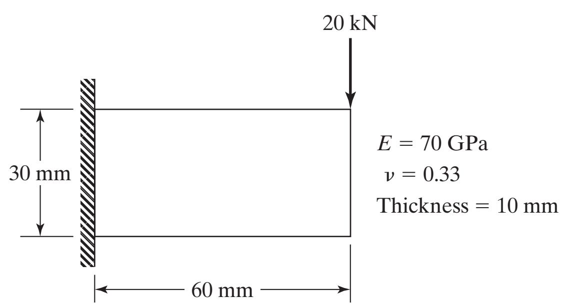


FIGURE P6.17

6.18. For the plate with a hole under plane stress (Fig. P6.18), find the deformed shape of the hole and determine the maximum stress distribution along AB by using stresses in elements adjacent to the line. (*Note*: The result in this problem is the same for any thickness. You may use $t = 1$ in.)

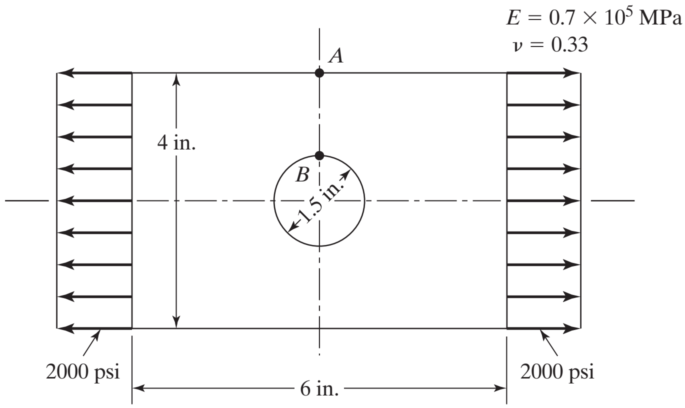


FIGURE P6.18

6.19. Model a half-disk with a hole (Fig. P6.19), and find the major and minor dimensions after compression. Also, plot the distribution of maximum stress along AB.

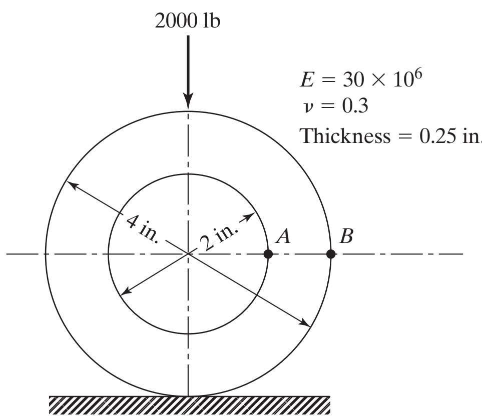


FIGURE P6.19

6.20. Consider the multipoint constraint

$$3Q_5 - 2Q_9 = 0.1$$

where Q_5 is the displacement along dof 5, and Q_9 is the displacement along dof 9. Write the penalty term

$$\frac{1}{2}C(3Q_5 - 2Q_9 - 0.1)^2 \text{ as } \frac{1}{2}(Q_5, Q_9)\mathbf{k}\begin{pmatrix} Q_5 \\ Q_9 \end{pmatrix} - (Q_5, Q_9)\mathbf{f}$$

and, hence, determine the stiffness additions **k** and force additions **f**. Then, fill in the following blanks to show how these additions are made in the computer program that uses a banded stiffness matrix **S**:

$$S(5,1) = S(5,1) + \_\_\_\_$$
$$S(9,1) = S(9,1) + \_\_\_\_$$
$$S(5,\_\_) = S(5,\_\_) + \_\_\_\_$$
$$F(5) = F(5) + \_\_\_\_$$
$$F(9) = F(9) + \_\_\_\_$$

6.21. Model the 22.5° segment of the octagonal pipe shown in Fig. P6.21. Show the deformed configuration of the segment and the distribution of maximum in-plane shear stress. [*Hint*: For all points along CD, use stiffness modification suggested in Eq. 6.73. Also, maximum in-plane shear stress $= (\sigma_1 - \sigma_2)/2$, where σ_1 and σ_2 are the principal stresses. Assume plane strain.]

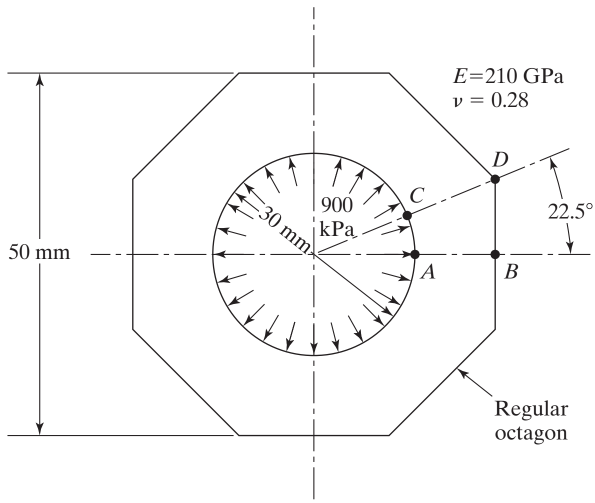


FIGURE P6.21

6.22. Determine the location and magnitude of maximum principal stress and maximum shearing stress in the fillet shown in Fig. P6.22.

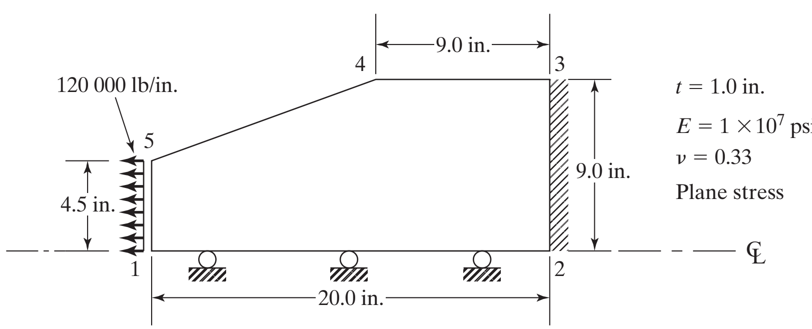


FIGURE P6.22

6.23. The torque arm shown in Fig. P6.23 is an automotive component, fixed at left bolt hole. Determine the location and magnitude of maximum von Mises stress, σ_{VM}, given by

$$\sigma_{VM} = \sqrt{\sigma_x^2 - \sigma_x\sigma_y + \sigma_y^2 + 3\tau_{xy}^2}$$

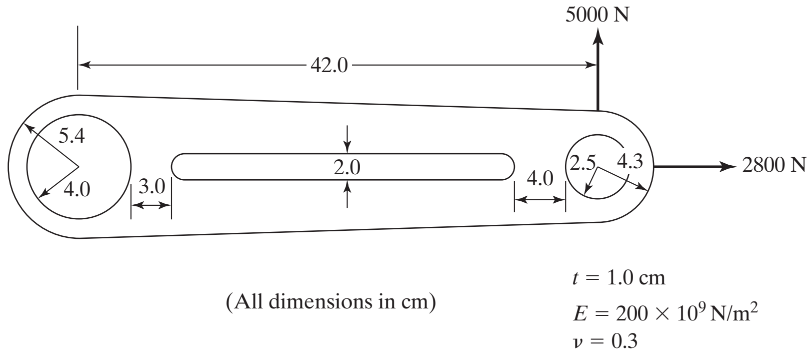


FIGURE P6.23

6.24. A large, flat surface of an aluminium body is subjected to a line load of 100 lb/in. Assuming plane strain, consider an enclosure as shown in Fig. P6.24 and determine the deformation of the surface and stress distribution in the body. (*Note* : Choose small elements close to the load and assume that deflection at 10 in. away is negligible.)

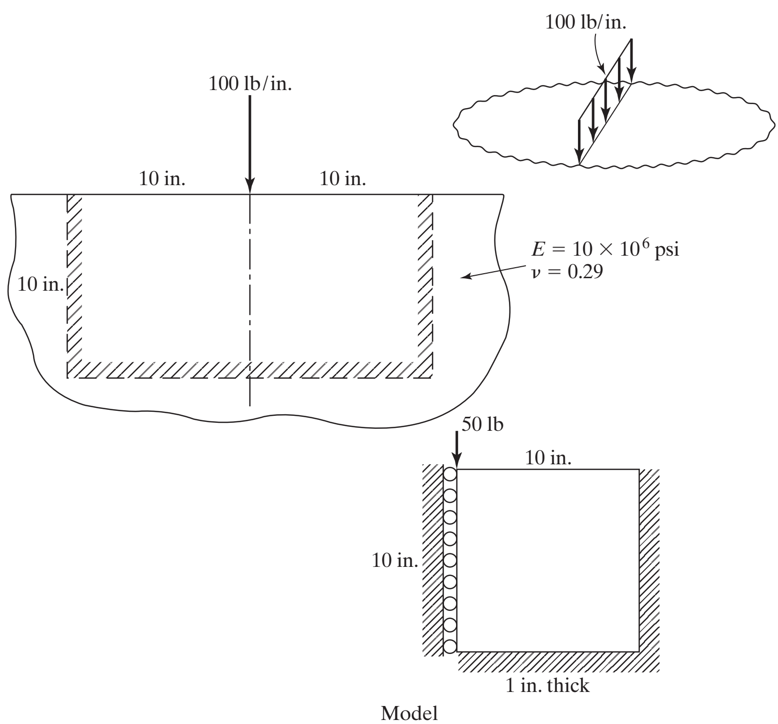


FIGURE P6.24

6.25. In Problem 6.24, the load is changed to a distributed load 400 lb/in.^{2} on a 1/4-in.-wide long region, as in Fig. P6.25. Model the problem as above with this loading and find deformation of the surface and stress distribution in the body. (*Note*: Assume that deflection at 10 in. away is negligible.)

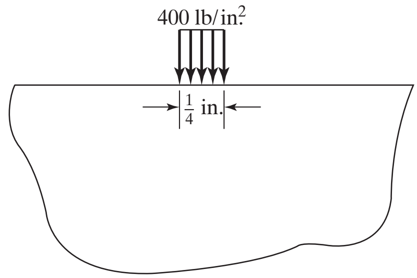


FIGURE P6.25

6.26. A (1/2) $\times$ 5-in. copper piece fits snugly into a short channel-shaped steel piece at room temperature, as shown in Fig. P6.26. The assembly is subjected to a uniform temperature increase of 100°F. Assuming that the properties are constant within this change and that the surfaces are bonded together, find the deformed shape and the stress distribution.

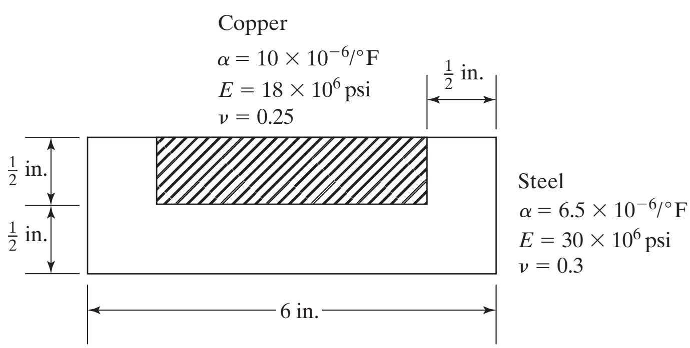


FIGURE P6.26

6.27. In the slotted copper ring shown in Fig. P6.27, two loads of magnitude P and load R are applied such that the 3-mm gap closes. Determine the magnitude of P and show the deformed shape of the part. (*Hint*: Find the deflection of gap for $P = 100$ and multiply the deflections proportionately. R is reaction at support.)

6.28. A titanium piece (A) is pressfitted into a titanium workpiece (B) as shown in Fig. P6.28. Determine the location (show on a sketch) and magnitude of maximum von Mises stress in both the parts (from your CST output file). *Then*, provide contour plots of the von Mises stress in each part. Data are as follows: $E = 101$ GPa, $\nu = 0.34$.

The guidelines are (a) use less than 100 elements in all, (b) mesh each part independently, but without duplicating node or element numbers, (c) choose a value for $L_{\text{interface}}$ and then enforce MPCs between the coincident nodes on this interface—the choice of $L_{\text{interface}}$ will involve trial-and-check as nodes that want to separate should not be forced together through the MPC, and (d) use symmetry. Assume a no-slip interface, a fixed base, and plane strain.

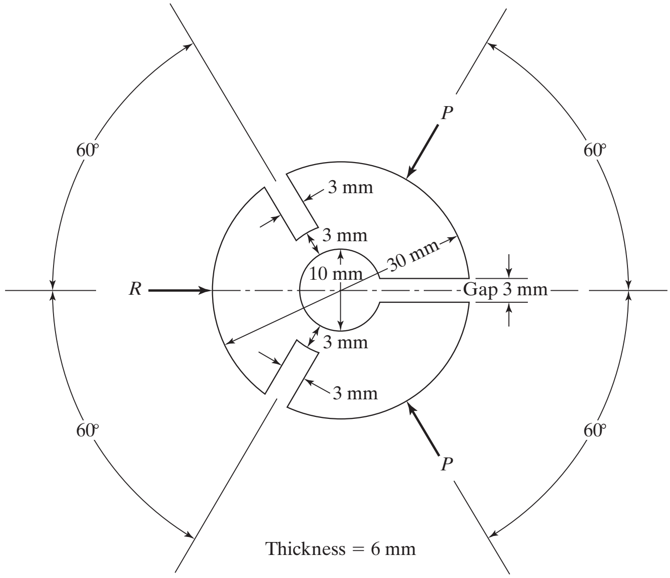


FIGURE P6.27

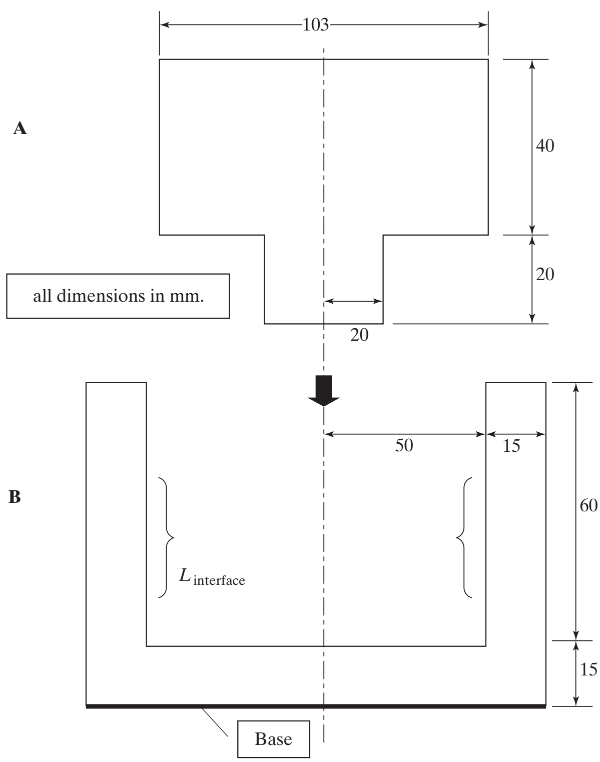


FIGURE P6.28

6.29. An edge crack of length a in a rectangular plate is subjected to a tensile stress σ_0 as shown in Fig. P6.29. Using a half-symmetry model, complete the following:

(a) Determine the crack opening angle, θ ($\theta = 0$ before the load is applied).

(b) Plot the y stress σ_y versus x, along the line A–O. Assuming that $\sigma_y = K_I/(\sqrt{2\pi x})$, use regression to estimate K_I. Compare your result for infinitely long plates, for which $K_I = 1.2\sigma_0\sqrt{\pi a}$ is used.

(c) Repeat (b) for increasing fine meshes near the crack tip.

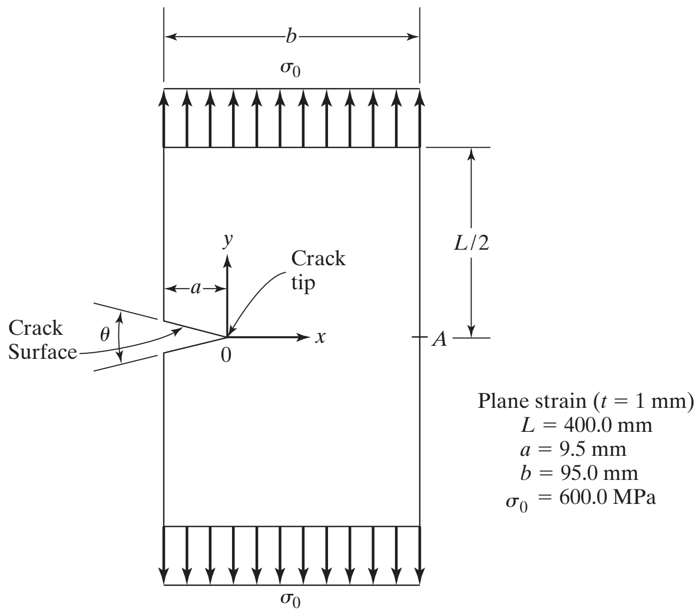


FIGURE P6.29

6.30. Use the geometry of the plate for the plane stress problem in P6.17. If the material of the plate is graphite–epoxy resin with fiber orientation at an angle θ to the horizontal, determine the deformation and stress values σ_x, σ_y, and σ_1, σ_2 for $\theta = 0°, 30°, 45°, 60°$, and $90°$. Properties of graphite in epoxy resin are given in Table 6.1. (*Hint*: The solution requires modification of program CST to incorporate the **D** matrix defined in Eq. 6.81.)

6.31. The plate with a hole in Problem 6.18 is made of pine wood. For $\theta = 0°, 30°, 45°, 60°$, and $90°$, complete the following:

(a) Determine the deformed shape of the hole.

(b) Find the stress distribution along AB and, hence, the stress concentration factor K_t. Plot K_t versus θ.

6.32. A structure in plane strain is shown in Fig. P6.32. Also shown is a simple finite element mesh. The structure and loading is symmetric with respect to the y-axis. Give all the boundary conditions and equivalent nodal loads for the mesh. Prepare data set for input to program CST.

6.33. Redo Example 6.7 using MATLAB. Show all matrices such as element **J**, **B**, **k** and global **K**. Show **K** with BC modifications and displacement and stress calculations. (*Hint*: *MATLAB operations*: **K** ([1:4][7:8], [1:4][7:8]) = **K** ([1:4][7:8], [1:4][7:8]) + **k** will introduce the 6 × 6 [**k**] of element 124 into rows and columns 1, 2, 3, 4, 7, 8 of global **K**. Also $\mathbf{K}_1$ = **K** ([1 4:5], [1 4:5]) will create a new matrix [$\mathbf{K}_1$] from rows/columns 1, 4, 5 of [**K**].)

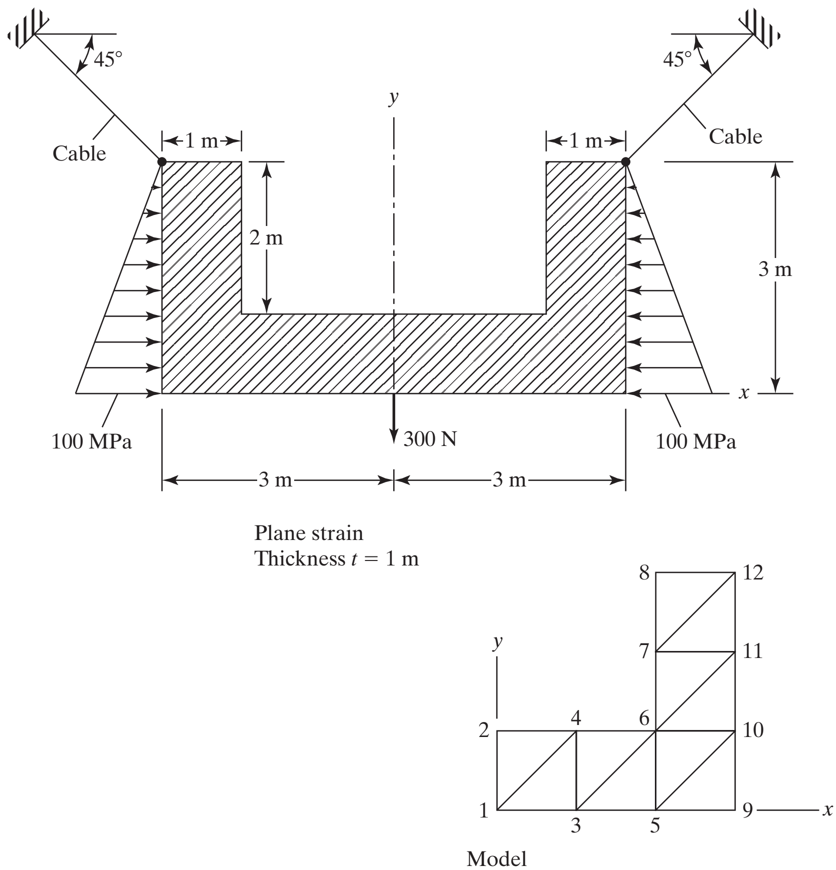


FIGURE P6.32

6.34. Figure P6.34 shows a four-element six-node mesh modeled using CST elements. Solve the problem using plane stress assumption. Use thickness of 1, $E = 1$, $\nu = 0.3$, coefficient of thermal expansion $\alpha = 0.1$ per degree. Draw your half-symmetry model showing the boundary conditions. Write your own MATLAB code for the problem and compare the displacement results and von Mises stress with the one obtained from CST. Plot the deformed shape using an appropriate scale factor.

6.35. For the CST element shown in Fig. P6.35, derive an expression for the row vector [**S**] where $\epsilon_x = \partial u/\partial x = [\mathbf{S}]\mathbf{q}$. That is, determine the matrix **S**, which relates the x normal strain with the nodal displacement vector **q**.

6.36. Express the integral $\int_e u^2\, dA$ in the form $\mathbf{q}^T\mathbf{W}\mathbf{q}$, and thus, give an expression for the matrix **W**. (*Hint*: Substitute $u = \mathbf{N}\mathbf{q}$ and write $dA = \det \mathbf{J}\, d\xi\, d\eta$, $\det \mathbf{J} = 2A_e$.)

6.37. Prepare data and solve the displacement patch test problem shown in Fig. 6.12a.

6.38. Prepare data and solve the load patch test problem shown in Fig. 6.12b.

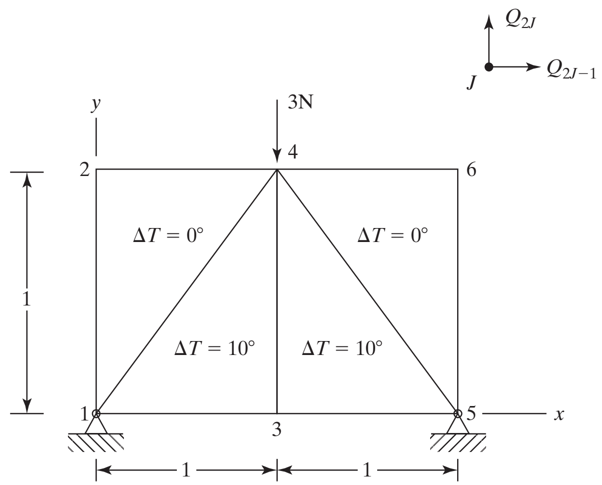


FIGURE P6.34

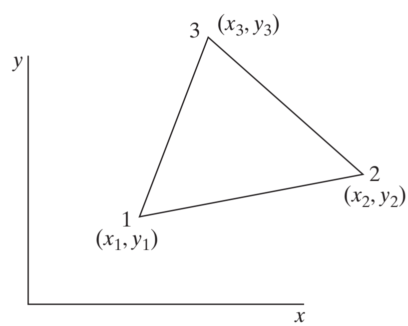


FIGURE P6.35

Program Listing

```
MAIN PROGRAM
'****************************************
'*            PROGRAM CST               *
'*      CONSTANT STRAIN TRIANGLE        *
'*  T.R.Chandrupatla and A.D.Belegundu  *
'****************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call Stiffness
     Call ModifyForBC
     Call BandSolver
     Call StressCalc
     Call ReactionCalc
     Call Output
End Sub
```

```
ELEMENT STIFFNESS AND GLOBAL STIFFNESS
Private Sub Stiffness()
     ReDim S(NQ, NBW)
     '----- Global Stiffness Matrix -----
     For N = 1 To NE
     Call DbMat(N, 1)
     '--- Element Stiffness
        For I = 1 To 6
           For J = 1 To 6
               C = 0
               For K = 1 To 3
                   C = C + 0.5 * Abs(DJ) * B(K, I) * DB(K, J) * TH(N)
               Next K
               SE(I, J) = C
           Next J
        Next I
     '--- Temperature Load Vector
        AL = PM(MAT(N), 3)
        C = AL * DT(N): If LC = 2 Then C = C * (1 + PNU)
        For I = 1 To 6
           TL(I) = 0.5 * C * TH(N) * Abs(DJ) * (DB(1, I) + DB(2, I))
        Next I
        For II = 1 To NEN
           NRT = NDN * (NOC(N, II) - 1)
           For IT = 1 To NDN
               NR = NRT + IT
               I = NDN * (II - 1) + IT
               For JJ = 1 To NEN
                   NCT = NDN * (NOC(N, JJ) - 1)
                   For JT = 1 To NDN
                      J = NDN * (JJ - 1) + JT
                      NC = NCT + JT - NR + 1
                      If NC > 0 Then
                          S(NR, NC) = S(NR, NC) + SE(I, J)
                      End If
                   Next JT
               Next JJ
               F(NR) = F(NR) + TL(I)
           Next IT
        Next II
     Next N
End Sub
```

```
D MATRIX, B MATRIX, AND DB MATRIX
Private Sub DbMat(N, ISTR)
     '----- D(), B() and DB() matrices
     '--- First the D-Matrix
     M = MAT(N): E = PM(M, 1): PNU = PM(M, 2): AL = PM(M, 3)
     '--- D() Matrix
     If LC = 1 Then
        '--- Plane Stress
        C1 = E / (1 - PNU ^ 2): C2 = C1 * PNU
```

continued

```
      Else
         '--- Plane Strain
         C = E / ((1 + PNU) * (1 - 2 * PNU))
         C1 = C * (1 - PNU): C2 = C * PNU
      End If
      C3 = 0.5 * E / (1 + PNU)
      D(1, 1) = C1: D(1, 2) = C2: D(1, 3) = 0
      D(2, 1) = C2: D(2, 2) = C1: D(2, 3) = 0
      D(3, 1) = 0: D(3, 2) = 0: D(3, 3) = C3
      '--- Strain-Displacement Matrix B()
      I1 = NOC(N, 1): I2 = NOC(N, 2): I3 = NOC(N, 3)
      X1 = X(I1, 1): Y1 = X(I1, 2)
      X2 = X(I2, 1): Y2 = X(I2, 2)
      X3 = X(I3, 1): Y3 = X(I3, 2)
      X21 = X2 - X1: X32 = X3 - X2: X13 = X1 - X3
      Y12 = Y1 - Y2: Y23 = Y2 - Y3: Y31 = Y3 - Y1
      DJ = X13 * Y23 - X32 * Y31 'DJ is determinant of Jacobian
      '--- Definition of B() Matrix
      B(1, 1) = Y23 / DJ: B(2, 1) = 0: B(3, 1) = X32 / DJ
      B(1, 2) = 0: B(2, 2) = X32 / DJ: B(3, 2) = Y23 / DJ
      B(1, 3) = Y31 / DJ: B(2, 3) = 0: B(3, 3) = X13 / DJ
      B(1, 4) = 0: B(2, 4) = X13 / DJ: B(3, 4) = Y31 / DJ
      B(1, 5) = Y12 / DJ: B(2, 5) = 0: B(3, 5) = X21 / DJ
      B(1, 6) = 0: B(2, 6) = X21 / DJ: B(3, 6) = Y12 / DJ
      '--- DB Matrix DB = D*B
      For I = 1 To 3
         For J = 1 To 6
           C = 0
           For K = 1 To 3
               C = C + D(I, K) * B(K, J)
             Next K
             DB(I, J) = C
          Next J
      Next I
      If ISTR = 2 Then
      '----- Stress Evaluation
      Q(1) = F(2 * I1 - 1): Q(2) = F(2 * I1)
      Q(3) = F(2 * I2 - 1): Q(4) = F(2 * I2)
      Q(5) = F(2 * I3 - 1): Q(6) = F(2 * I3)
      C1 = AL * DT(N): If LC = 2 Then C1 = C1 * (1 + PNU)
      For I = 1 To 3
             C = 0
             For K = 1 To 6
                    C = C + DB(I, K) * Q(K)
             Next K
             STR(I) = C - C1 * (D(I, 1) + D(I, 2))
      Next I
      End If
  End Sub
```

```
STRESS CALCULATIONS
Private Sub StressCalc()
     ReDim Stress(NE, 3), PrinStress(NE, 3), PltStress(NE)
     '----- Stress Calculations
     For N = 1 To NE
         Call DbMat(N, 2)
     '--- Principal Stress Calculations
         If STR(3) = 0 Then
            S1 = STR(1): S2 = STR(2): ANG = 0
            If S2 > S1 Then
               S1 = STR(2): S2 = STR(1): ANG = 90
            End If
         Else
            C = 0.5 * (STR(1) + STR(2))
            R = Sqr(0.25 * (STR(1) - STR(2)) ^ 2 + (STR(3)) ^ 2)
            S1 = C + R: S2 = C - R
            If C > STR(1) Then
               ANG = 57.2957795 * Atn(STR(3) / (S1 - STR(1)))
               If STR(3) > 0 Then ANG = 90 - ANG
               If STR(3) < 0 Then ANG = -90 - ANG
            Else
               ANG = 57.29577951 * Atn(STR(3) / (STR(1) - S2))
            End If
         End If
         Stress(N, 1) = STR(1)
         Stress(N, 2) = STR(2)
         Stress(N, 3) = STR(3)
         PrinStress(N, 1) = S1
         PrinStress(N, 2) = S2
         PrinStress(N, 3) = ANG
         '--- ANG is angle in degrees from X to S1
         If IPL = 2 Then PltStress(N) = 0.5 * (S1 - S2)
         If IPL = 3 Then
            S3 = 0: If LC = 2 Then S3 = PNU * (S1 + S2)
            C = (S1 - S2) ^ 2 + (S2 - S3) ^ 2 + (S3 - S1) ^ 2
            PltStress(N) = Sqr(0.5 * C)
         End If
     Next N
End Sub
```

CHAPTER 7

Axisymmetric Solids Subjected to Axisymmetric Loading

7.1 INTRODUCTION

Problems involving three-dimensional axisymmetric solids or solids of revolution, subjected to axisymmetric loading, reduce to simple two-dimensional problems. Because of total symmetry about the z-axis, as seen in Fig. 7.1, all deformations and stresses are independent of the rotational angle θ. Thus, the problem needs to be looked at as a two-dimensional problem in rz, defined on the revolving area (Fig. 7.1b). Gravity forces

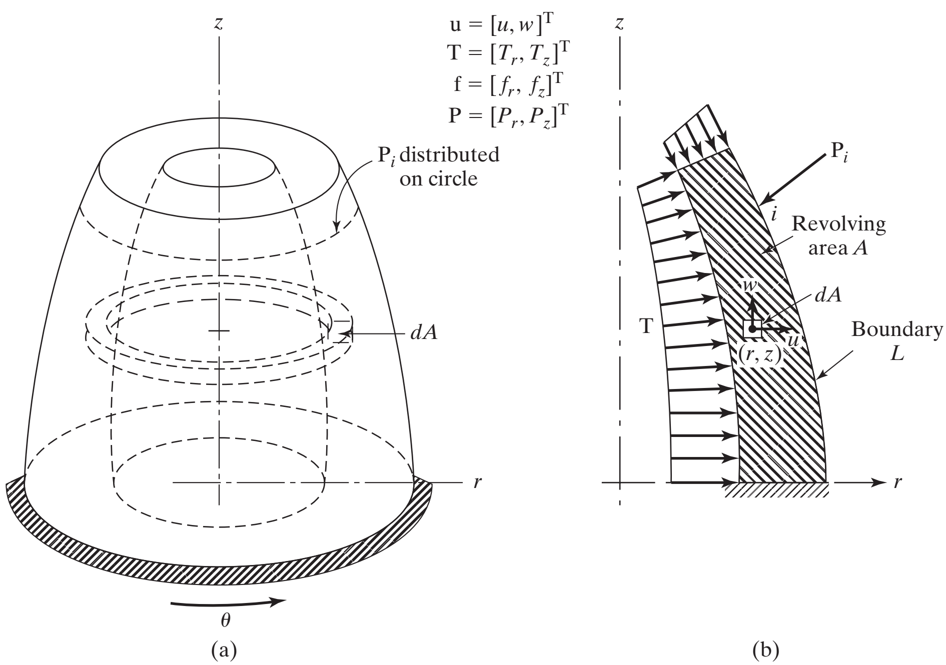


FIGURE 7.1 Axisymmetric problem.

can be considered if acting in the z direction. Revolving bodies like flywheels can be analyzed by introducing centrifugal forces in the body force term. We will now discuss the axisymmetric problem formulation.

7.2 AXISYMMETRIC FORMULATION

Considering the elemental volume shown in Fig. 7.2, the potential energy can be written in the form

$$\Pi = \frac{1}{2}\int_0^{2\pi}\int_A \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}\, r\, dA\, d\theta - \int_0^{2\pi}\int_A \mathbf{u}^{\mathrm{T}}\mathbf{f}\, r\, dA\, d\theta - \int_0^{2\pi}\int_L \mathbf{u}^{\mathrm{T}}\mathbf{T}\, r\, d\ell\, d\theta - \sum_i \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i \tag{7.1}$$

where $r\,d\ell\,d\theta$ is the elemental surface area and the point load $\mathbf{P}_i$ represents a line load distributed around a circle, as shown in Fig. 7.1.

All variables in the integrals are independent of θ. Thus, Eq. 7.1 can be written as

$$\Pi = 2\pi\left(\frac{1}{2}\int_A \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}\, r\, dA - \int_A \mathbf{u}^{\mathrm{T}}\mathbf{f}\, r\, dA - \int_L \mathrm{u}^{\mathrm{T}}\mathbf{T}\, r\, d\ell\right) - \sum_i \mathbf{u}_i^{\mathrm{T}}\mathbf{P}_i \tag{7.2}$$

where

$$\mathbf{u} = [u \quad w]^{\mathrm{T}} \tag{7.3}$$

$$\mathbf{f} = [f_r \quad f_z]^{\mathrm{T}} \tag{7.4}$$

$$\mathbf{T} = [T_r \quad T_z]^{\mathrm{T}} \tag{7.5}$$

From Fig. 7.3, we can write the relationship between strains $\boldsymbol{\epsilon}$ and displacements $\mathbf{u}$ as

$$\begin{aligned}\boldsymbol{\epsilon} &= [\epsilon_r \quad \epsilon_z \quad \gamma_{rz} \quad \epsilon_\theta]^{\mathrm{T}} \\ &= \left[\frac{\partial u}{\partial r} \quad \frac{\partial w}{\partial z} \quad \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \quad \frac{u}{r}\right]^{\mathrm{T}}\end{aligned} \tag{7.6}$$

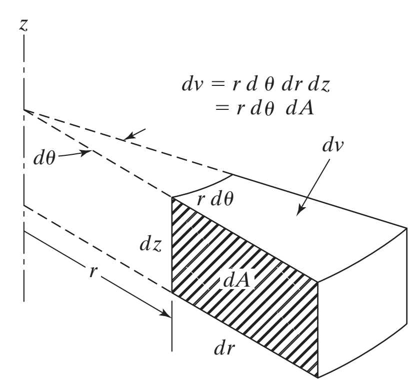


FIGURE 7.2 Elemental volume.

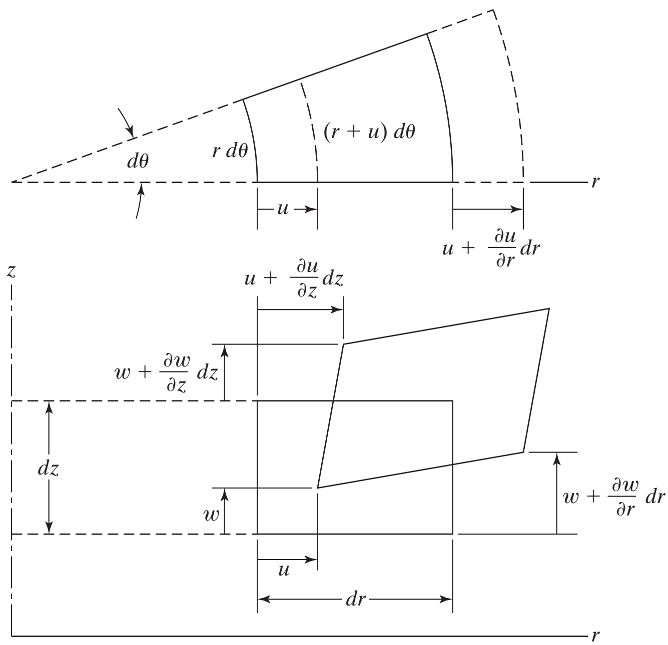


FIGURE 7.3 Deformation of elemental volume.

The stress vector is correspondingly defined as

$$\boldsymbol{\sigma} = [\sigma_r \quad \sigma_z \quad \tau_{rz} \quad \sigma_\theta]^{\mathrm{T}} \tag{7.7}$$

The stress–strain relation is given in the usual form

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} \tag{7.8}$$

where the (4×4) matrix $\mathbf{D}$ can be written by neglecting the appropriate terms from the three-dimensional matrix in Chapter 1, as

$$\mathbf{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 & \frac{\nu}{1-\nu} \\ \frac{\nu}{1-\nu} & 1 & 0 & \frac{\nu}{1-\nu} \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 1 \end{bmatrix} \tag{7.9}$$

In the Galerkin formulation, we require

$$2\pi \int_A \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\epsilon}(\boldsymbol{\varphi}) r\, dA - \left(2\pi \int_A \boldsymbol{\varphi}^{\mathrm{T}} \mathbf{f} r\, dA + 2\pi \int_L \boldsymbol{\varphi}^{\mathrm{T}} \mathbf{T} r\, d\ell + \sum \boldsymbol{\varphi}_i^{\mathrm{T}} \mathbf{P}_i \right) = 0 \tag{7.10}$$

where

$$\boldsymbol{\varphi} = [\phi_r, \phi_z]^{\mathrm{T}} \tag{7.11}$$

$$\boldsymbol{\epsilon}(\boldsymbol{\varphi}) = \left[\frac{\partial \phi_r}{\partial r} \quad \frac{\partial \phi_z}{\partial z} \quad \frac{\partial \phi_r}{\partial z} + \frac{\partial \phi_z}{\partial r} \quad \frac{\phi_r}{r} \right]^{\mathrm{T}} \tag{7.12}$$

7.3 FINITE ELEMENT MODELING: TRIANGULAR ELEMENT

The two-dimensional region defined by the revolving area is divided into triangular elements, as shown in Fig. 7.4. Though each element is completely represented by the area in the *rz* plane, in reality, it is a ring-shaped solid of revolution obtained by revolving the triangle about the *z*-axis. A typical element is shown in Fig. 7.5.

The definition of connectivity of elements and the nodal coordinates follows the steps involved in the constant-strain triangle (CST) element discussed in Section 6.3. We note here that the *r*- and *z*-coordinates, respectively, replace *x* and *y*.

Using the three shape functions N_1, N_2, and N_3, we define

$$\mathbf{u} = \mathbf{N}\mathbf{q} \tag{7.13}$$

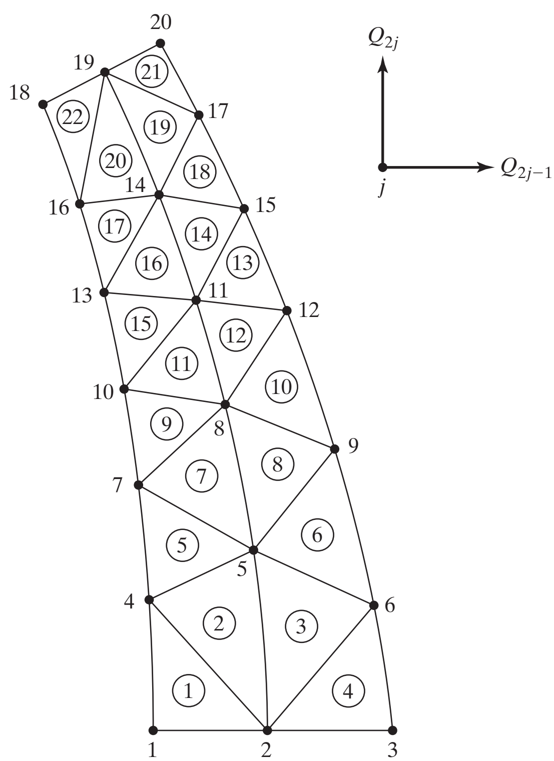


FIGURE 7.4 Triangulation.

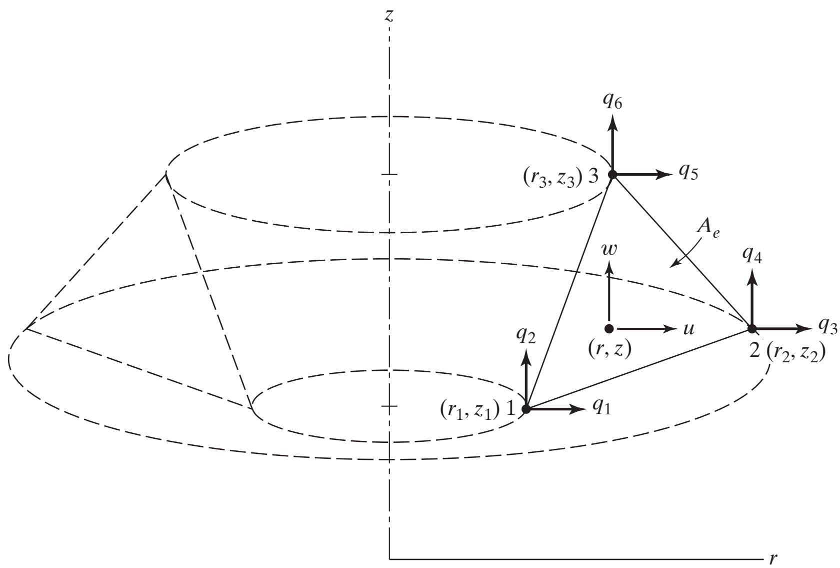


FIGURE 7.5 Axisymmetric triangular element.

where **u** is defined in Eq. 7.3 and

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \tag{7.14}$$

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^{\mathrm{T}} \tag{7.15}$$

If we denote $N_1 = \xi$ and $N_2 = \eta$, and note that $N_3 = 1 - \xi - \eta$, then Eq. 7.13 gives

$$\begin{aligned} u &= \xi q_1 + \eta q_3 + (1 - \xi - \eta) q_5 \\ w &= \xi q_2 + \eta q_4 + (1 - \xi - \eta) q_6 \end{aligned} \tag{7.16}$$

By using the isoparametric representation, we find

$$\begin{aligned} r &= \xi r_1 + \eta r_2 + (1 - \xi - \eta) r_3 \\ z &= \xi z_1 + \eta z_2 + (1 - \xi - \eta) z_3 \end{aligned} \tag{7.17}$$

The chain rule of differentiation gives

$$\begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \dfrac{\partial u}{\partial r} \\ \dfrac{\partial u}{\partial z} \end{Bmatrix} \tag{7.18}$$

and

$$\begin{Bmatrix} \dfrac{\partial w}{\partial \xi} \\ \dfrac{\partial w}{\partial \eta} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \dfrac{\partial w}{\partial r} \\ \dfrac{\partial w}{\partial z} \end{Bmatrix} \tag{7.19}$$

where the Jacobian is given by

$$\mathbf{J} = \begin{bmatrix} r_{13} & z_{13} \\ r_{23} & z_{23} \end{bmatrix} \tag{7.20}$$

In the definition of **J** earlier, we have used the notation $r_{ij} = r_i - r_j$ and $z_{ij} = z_i - z_j$. The determinant of **J** is

$$\det \mathbf{J} = r_{13}z_{23} - r_{23}z_{13} \tag{7.21}$$

Recall that $|\det \mathbf{J}| = 2A_e$. That is, the absolute value of the determinant of **J** equals twice the area of the element. The inverse relations for Eq. 7.18 and 7.19 are given by

$$\begin{Bmatrix} \dfrac{\partial u}{\partial r} \\ \dfrac{\partial u}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} \quad \text{and} \quad \begin{Bmatrix} \dfrac{\partial w}{\partial r} \\ \dfrac{\partial w}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \dfrac{\partial w}{\partial \xi} \\ \dfrac{\partial w}{\partial \eta} \end{Bmatrix} \tag{7.22}$$

where

$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} z_{23} & -z_{13} \\ -r_{23} & r_{13} \end{bmatrix} \tag{7.23}$$

Introducing this transformation relationship into the strain–displacement relation in Eq. 7.6 and by using Eq. 7.16, we get

$$\boldsymbol{\epsilon} = \begin{Bmatrix} \dfrac{z_{23}(q_1 - q_5) - z_{13}(q_3 - q_5)}{\det \mathbf{J}} \\ \dfrac{-r_{23}(q_2 - q_6) + r_{13}(q_4 - q_6)}{\det \mathbf{J}} \\ \dfrac{-r_{23}(q_1 - q_5) + r_{13}(q_3 - q_5) + z_{23}(q_2 - q_6) - z_{13}(q_4 - q_6)}{\det \mathbf{J}} \\ \dfrac{N_1 q_1 + N_2 q_3 + N_3 q_5}{r} \end{Bmatrix}$$

This can be written in the matrix form as

$$\boldsymbol{\epsilon} = \mathbf{Bq} \tag{7.24}$$

where the element strain–displacement matrix of dimension (4×6) is given by

$$\mathbf{B} = \begin{bmatrix} \frac{z_{23}}{\det \mathbf{J}} & 0 & \frac{z_{31}}{\det \mathbf{J}} & 0 & \frac{z_{12}}{\det \mathbf{J}} & 0 \\ 0 & \frac{r_{32}}{\det \mathbf{J}} & 0 & \frac{r_{13}}{\det \mathbf{J}} & 0 & \frac{r_{21}}{\det \mathbf{J}} \\ \frac{r_{32}}{\det \mathbf{J}} & \frac{z_{23}}{\det \mathbf{J}} & \frac{r_{13}}{\det \mathbf{J}} & \frac{z_{31}}{\det \mathbf{J}} & \frac{r_{21}}{\det \mathbf{J}} & \frac{z_{12}}{\det \mathbf{J}} \\ \frac{N_1}{r} & 0 & \frac{N_2}{r} & 0 & \frac{N_3}{r} & 0 \end{bmatrix} \tag{7.25}$$

Potential Energy Approach

The potential energy Π on the discretized region is given by

$$\Pi = \sum_e \left[\frac{1}{2}\left(2\pi \int_e \boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{D} \boldsymbol{\epsilon}\, r\, dA \right) - 2\pi \int_e \mathbf{u}^{\mathrm{T}} \mathbf{f} r\, dA - 2\pi \int_e \mathbf{u}^{\mathrm{T}} \mathbf{T} r\, d\ell \right] - \sum \mathbf{u}_i^{\mathrm{T}} \mathbf{P}_i \tag{7.26}$$

The element strain energy U_e given by the first term can be written as

$$U_e = \frac{1}{2} \mathbf{q}^{\mathrm{T}} \left(2\pi \int_e \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} r\, dA \right) \mathbf{q} \tag{7.27}$$

The term inside the parentheses is the element stiffness matrix:

$$\mathbf{k}^e = 2\pi \int_e \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} r\, dA \tag{7.28}$$

The fourth row in $\mathbf{B}$ has terms of the type N_i/r. Further, this integral also has an additional r in it. As a simple approximation, $\mathbf{B}$ and r can be evaluated at the centroid of the triangle and used as representative values for the triangle. At the centroid of the triangle,

$$N_1 = N_2 = N_3 = \frac{1}{3} \tag{7.29}$$

and

$$\bar{r} = \frac{r_1 + r_2 + r_3}{3}$$

where $\bar{r}$ is the radius of the centroid. Denoting $\overline{\mathbf{B}}$ as the element strain–displacement matrix $\mathbf{B}$ evaluated at the centroid, we get

$$\mathbf{k}^e = 2\pi \bar{r} \overline{\mathbf{B}}^{\mathrm{T}} \mathbf{D} \overline{\mathbf{B}} \int_e dA$$

or

$$\mathbf{k}^e = 2\pi \bar{r} A_e \overline{\mathbf{B}}^{\mathrm{T}} \mathbf{D} \overline{\mathbf{B}} \tag{7.30}$$

We note here that $2\pi \bar{r} A_e$ is the volume of the ring-shaped element shown in Fig. 7.5. Also, A_e is given by

$$A_e = \frac{1}{2}|\det \mathbf{J}| \tag{7.31}$$

We also use this centroid or midpoint rule[1] for body forces and surface tractions as discussed in the following section. Caution must be exerted for elements close to the axis of symmetry. For better results, smaller elements need to be chosen close to the axis of symmetry. Another approach is to introduce $r = N_1 r_1 + N_2 r_2 + N_3 r_3$ in the following equations and perform elaborate integration. More elaborate methods of numerical integration are discussed in Chapter 7.

Body Force Term

We first consider the body force term $2\pi \int_e \mathbf{u}^\mathrm{T}\mathbf{f} r\, dA$. We have

$$\begin{aligned} 2\pi \int_e \mathbf{u}^\mathrm{T}\mathbf{f} r\, dA &= 2\pi \int_e (uf_r + wf_z) r\, dA \\ &= 2\pi \int_e [(N_1 q_1 + N_2 q_3 + N_3 q_5) f_r + (N_1 q_2 + N_2 q_4 + N_3 q_6) f_z] r\, dA \end{aligned}$$

Once again, approximating the variable quantities by their values at the centroid of the triangle, we get

$$2\pi \int_e \mathbf{u}^\mathrm{T}\mathbf{f} r\, dA = \mathbf{q}^\mathrm{T}\mathbf{f}^e \tag{7.32}$$

where the element body force vector $\mathbf{f}^e$ is given by

$$\mathbf{f}^e = \frac{2\pi \bar{r} A_e}{3}[\bar{f}_r \quad \bar{f}_z \quad \bar{f}_r \quad \bar{f}_z \quad \bar{f}_r \quad \bar{f}_z]^\mathrm{T} \tag{7.33}$$

The bar on the $\mathbf{f}$ terms indicates that they are evaluated at the centroid. Where body force is the primary load, greater accuracy may be obtained by substituting $r = N_1 r_1 + N_2 r_2 + N_3 r_3$ into Eq. 7.32 and integrating to get nodal loads.

Rotating Flywheel

As an example, let us consider a rotating flywheel with its axis in the z direction. We consider the flywheel to be stationary and apply the equivalent radial centrifugal (inertial) force per unit volume of $\rho r \omega^2$ where ρ is the density (mass per unit volume), and ω is the angular velocity in rad/s. In addition, if gravity acts along the negative z-axis, then

$$\mathbf{f} = [f_r, f_z]^\mathrm{T} = [\rho r \omega^2 \ \ -\rho g]^\mathrm{T} \tag{7.34}$$

and

$$\bar{f}_r = \rho \bar{r} \omega^2, \bar{f}_z = -\rho g \tag{7.35}$$

[1] Suggested by Zienkiewicz, O. C., *The Finite Element Method*, 3rd edn. New York: McGraw-Hill (1983).

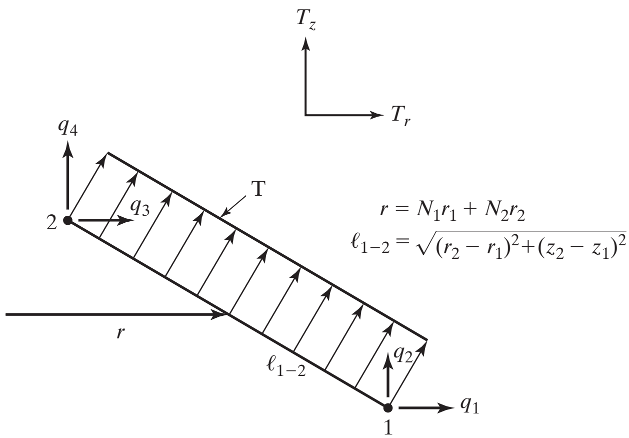


FIGURE 7.6 Surface traction.

For more precise results with coarse meshes, we need to use $r = N_1r_1 + N_2r_2 + N_3r_3$ and integrate.

Surface Traction

For a uniformly distributed load with components T_r and T_z, shown in Fig. 7.6, on the edge connecting nodes 1 and 2, we get

$$2\pi \int_e \mathbf{u}^{\mathrm{T}}\mathbf{T}r\,d\ell = \mathbf{q}^{\mathrm{T}}\mathbf{T}^e \tag{7.36}$$

where

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]^{\mathrm{T}} \tag{7.37}$$

$$\mathbf{T}^e = 2\pi\ell_{1-2}[aT_r \quad aT_z \quad bT_r \quad bT_z]^{\mathrm{T}} \tag{7.38}$$

$$a = \frac{2r_1 + r_2}{6} \qquad b = \frac{r_1 + 2r_2}{6} \tag{7.39}$$

$$\ell_{1-2} = \sqrt{(r_2 - r_1)^2 + (z_2 - z_1)^2} \tag{7.40}$$

In this derivation, r is expressed as $N_1r_1 + N_2r_2$ and then integrated. When the line 1–2 is parallel to the z-axis, we have $r_1 = r_2$, which gives $a = b = 0.5r_1$. For a linearly distributed traction, see Problem 7.13.

Example 7.1

An axisymmetric body with a linearly distributed load on the conical surface is shown in Fig. E7.1. Determine the equivalent point loads at nodes 2, 4, and 6.

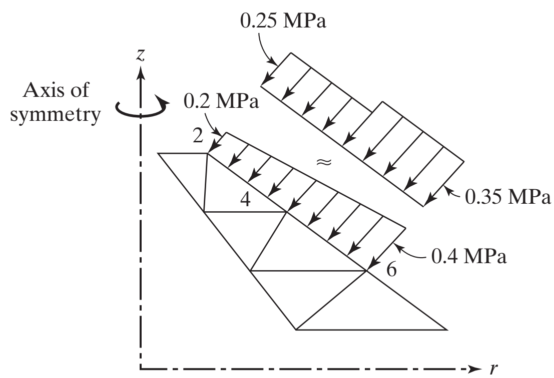


FIGURE E7.1

Solution We approximate the linearly distributed load by the average uniformly distributed loads on the edges 6–4 and 4–2 as shown in Fig. E7.1. Relationships for more precise modeling of a linearly distributed load are provided in Problem 7.13. We now consider the two edges 6–4 and 4–2 separately and then merge them.

For edge 6–4

$$p = 0.35\,\text{MPa}, \quad r_1 = 60\,\text{mm}, \quad z_1 = 40\,\text{mm}, \quad r_2 = 40\,\text{mm}, \quad z_2 = 55\,\text{mm}$$

$$\ell_{1-2} = \sqrt{(r_1 - r_2)^2 + (z_1 - z_2)^2} = 25\,\text{mm}$$

$$c = \frac{z_2 - z_1}{\ell_{1-2}} = 0.6, \qquad s = \frac{r_1 - r_2}{\ell_{1-2}} = 0.8$$

$$T_r = -pc = -0.21, \qquad T_z = -ps = -0.28$$

$$a = \frac{2r_1 + r_2}{6} = 26.67, \qquad b = \frac{r_1 + 2r_2}{6} = 23.33$$

$$\begin{aligned}\mathbf{T}^1 &= 2\pi\ell_{1-2}[aT_r \quad aT_z \quad bT_r \quad bT_z]^{\mathrm{T}} \\ &= [-879.65 \quad -1172.9 \quad -769.69 \quad -1026.25]^{\mathrm{T}}\ \text{N}\end{aligned}$$

These loads add to F_{11}, F_{12}, F_7, and F_8, respectively.

For edge 4–2

$$p = 0.25\,\text{MPa}, \quad r_1 = 40\,\text{mm}, \quad z_1 = 55\,\text{mm}, \quad r_2 = 20\,\text{mm}, \quad z_2 = 70\,\text{mm}$$

$$\ell_{1-2} = \sqrt{(r_1 - r_2)^2 + (z_1 - z_2)^2} = 25\,\text{mm}$$

$$c = \frac{z_2 - z_1}{\ell_{1-2}} = 0.6, \qquad s = \frac{r_1 - r_2}{\ell_{1-2}} = 0.8$$

$$T_r = -pc = -0.15, \qquad T_z = -ps = -0.2$$

$$a = \frac{2r_1 + r_2}{6} = 16.67, \qquad b = \frac{r_1 + 2r_2}{6} = 13.33$$

$$\mathbf{T}^1 = 2\pi\ell_{1-2}[aT_r \quad aT_z \quad bT_r \quad bT_z]^{\mathrm{T}}$$

$$= [-392.7 \quad -523.6 \quad -314.16 \quad -418.88]^{\mathrm{T}} \text{ N}$$

These loads add to F_7, F_8, F_3, and F_4, respectively. Thus,

$$[F_3 \quad F_4 \quad F_7 \quad F_8 \quad F_{11} \quad F_{12}] = [-314.2 \quad -418.9 \quad -1162.4 \quad -1696.5 \quad -879.7 \quad -1172.9]\text{N}$$

■

The load distributed along a circumference of a circle on the surface has to be applied at a point on the revolving area. We may conveniently locate a node here and add the load components.

On summing up the strain energy and force terms over all the elements and modifying for the boundary conditions while minimizing the total potential energy, we get

$$\mathbf{KQ} = \mathbf{F} \tag{7.41}$$

We note here that axisymmetric boundary conditions need to be applied only on the revolving area shown in Fig. 7.1.

Galerkin Approach

In the Galerkin formulation, the consistent variation $\boldsymbol{\phi}$ in an element is expressed as

$$\boldsymbol{\phi} = \mathrm{N}\psi \tag{7.42}$$

where

$$\boldsymbol{\psi} = [\psi_1 \quad \psi_2 \quad \ldots \quad \psi_6]^{\mathrm{T}} \tag{7.43}$$

The corresponding strain $\boldsymbol{\epsilon}(\boldsymbol{\phi})$ is given by

$$\boldsymbol{\epsilon}(\boldsymbol{\phi}) = \mathbf{B}\boldsymbol{\psi} \tag{7.44}$$

The global vector of variations $\boldsymbol{\Psi}$ is represented by

$$\boldsymbol{\Psi} = [\Psi_1 \quad \Psi_2 \quad \Psi_3 \quad \ldots \quad \Psi_N]^{\mathrm{T}} \tag{7.45}$$

We now introduce the interpolated displacements into the Galerkin variational form (Eq. 7.10). The first term representing the internal virtual work gives

$$\begin{aligned}\text{Internal virtual work} &= 2\pi \int_A \sigma^{\mathrm{T}}\boldsymbol{\epsilon}(\boldsymbol{\phi})r\,dA \\ &= \sum_e 2\pi \int_e \mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\psi r\,dA \\ &= \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{k}^e\psi \end{aligned} \tag{7.46}$$

where the element stiffness $\mathbf{k}^e$ is given by

$$\mathbf{k}^e = 2\pi \bar{r} A_e \overline{\mathbf{B}}^{\mathrm{T}} \mathbf{D} \overline{\mathbf{B}} \tag{7.47}$$

We note that $\mathbf{k}^e$ is symmetric. Using the connectivity of the elements, the internal virtual work can be expressed in the form

$$\begin{aligned} \text{Internal virtual work} &= \sum_e \mathbf{q}^{\mathrm{T}} \mathbf{k}^e \psi = \sum \psi^{\mathrm{T}} \mathbf{k}^e \mathbf{q} \\ &= \mathbf{\Psi}^{\mathrm{T}} \mathbf{K} \mathbf{Q} \end{aligned} \tag{7.48}$$

where $\mathbf{K}$ is the global stiffness matrix. The external virtual work terms in Eq. 7.10 involving body forces, surface tractions, and point loads can be treated in the same way as in the potential energy approach by replacing $\mathbf{q}$ with Ψ. The summation of all the force terms over the elements then yields

$$\text{External virtual work} = \mathbf{\Psi}^{\mathrm{T}} \mathbf{F} \tag{7.49}$$

The boundary conditions are considered using the ideas discussed in Chapter 3. The stiffness matrix $\mathbf{K}$ and the force $\mathbf{F}$ are modified, resulting in the same set of equations as Eq. 7.41.

Detailed calculations in the example that follows are provided for illustrating the steps involved. However, it is expected that the exercise problems at the end of the chapter will be solved using the program AXISYM, which is provided.

Example 7.2

In Fig. E7.2, a long cylinder of inside diameter 80 mm and outside diameter 120 mm snugly fits in a hole over its full length. The cylinder is then subjected to an internal pressure of 2 MPa. Using two elements on the 10 mm length shown, find the displacements at the inner radius.

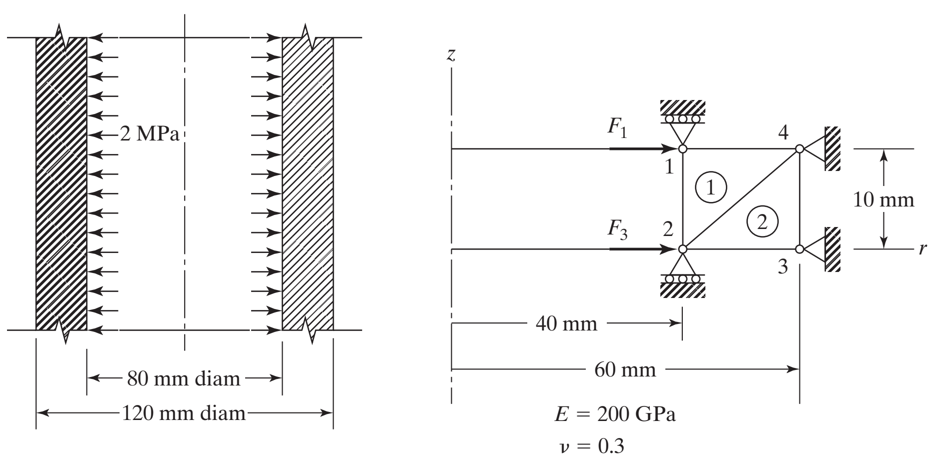


FIGURE E7.2

Solution Consider the following table:

	Connectivity		
Element	1	2	3
1	1	2	4
2	2	3	4

	Coordinates	
Node	r	z
1	40	10
2	40	0
3	60	0
4	60	10

We will use the units of millimeters for length, newtons for force, and megapascals for stress and E. These units are consistent. On substituting $E = 200{,}000$ MPa and $\nu = 0.3$, we have

$$\mathbf{D} = \begin{bmatrix} 2.69 \times 10^5 & 1.15 \times 10^5 & 0 & 1.15 \times 10^5 \\ 1.15 \times 10^5 & 2.69 \times 10^5 & 0 & 1.15 \times 10^5 \\ 0 & 0 & 0.77 \times 10^5 & 0 \\ 1.15 \times 10^5 & 1.15 \times 10^5 & 0 & 2.69 \times 10^5 \end{bmatrix}$$

for both elements, det $\mathbf{J} = 200\,\text{mm}^2$ and $A_e = 100\,\text{mm}^2$. From Eq. 7.38, forces F_1 and F_3 are given by

$$F_1 = F_3 = \frac{2\pi r_1 \ell_e p_i}{2} = \frac{2\pi(40)(10)(2)}{2} = 2514\,\text{N}$$

The $\mathbf{B}$ matrices relating element strains to nodal displacements are obtained first. For element 1, $\bar{r} = (\frac{1}{3})(40 + 40 + 60) = 46.67$ mm and

$$\overline{\mathbf{B}}^1 = \begin{bmatrix} -0.05 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0.1 & 0 & -0.1 & 0 & 0 \\ 0.1 & -0.05 & -0.1 & 0 & 0 & 0.05 \\ 0.0071 & 0 & 0.0071 & 0 & 0.0071 & 0 \end{bmatrix}$$

For element 2, $\bar{r} = (\frac{1}{3})(40 + 60 + 60) = 53.33$ mm and

$$\overline{\mathbf{B}}^2 = \begin{bmatrix} -0.05 & 0 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.1 & 0 & 0.1 \\ 0 & -0.05 & -0.1 & 0.05 & 0.1 & 0 \\ 0.00625 & 0 & 0.00625 & 0 & 0.00625 & 0 \end{bmatrix}$$

The element stress–displacement matrices are obtained by multiplying $\mathbf{DB}$:

$$\mathbf{D}\overline{\mathbf{B}}^1 = 10^4 \begin{bmatrix} -1.26 & 1.15 & 0.082 & -1.15 & 1.43 & 0 \\ -0.49 & 2.69 & 0.082 & -2.69 & 0.657 & 0.1 \\ 0.77 & -0.385 & -0.77 & 0 & 0 & 0.385 \\ -0.384 & 1.15 & 0.191 & -1.15 & 0.766 & 0 \end{bmatrix}$$

$$\mathbf{D}\overline{\mathbf{B}}^2 = 10^4 \begin{bmatrix} -1.27 & 0 & 1.42 & -1.15 & 0.072 & 1.15 \\ -0.503 & 0 & 0.647 & -2.69 & 0.072 & 2.69 \\ 0 & -0.385 & -0.77 & 0.385 & 0.77 & 0 \\ -0.407 & 0 & 0.743 & -1.15 & 0.168 & 1.15 \end{bmatrix}$$

The stiffness matrices are obtained by finding $2\pi\bar{r}A_e\overline{\mathbf{B}}^{\mathrm{T}}\mathbf{D}\overline{\mathbf{B}}$ for each element:

$$\begin{array}{rc} \text{Global dof} \rightarrow & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 7 & 8 \end{array} \\ \mathbf{k}^1 = 10^7 & \begin{bmatrix} 4.03 & -2.58 & -2.34 & 1.45 & -1.932 & 1.13 \\ & 8.45 & 1.37 & -7.89 & 1.93 & -0.565 \\ & & 2.30 & -0.24 & 0.16 & -1.13 \\ & & & 7.89 & -1.93 & 0 \\ \text{Symmetric} & & & & 2.25 & 0 \\ & & & & & 0.565 \end{bmatrix} \end{array}$$

$$\begin{array}{rc} \text{Global dof} \rightarrow & \begin{array}{cccccc} 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \mathbf{k}^2 = 10^7 & \begin{bmatrix} 2.05 & 0 & -2.22 & 1.69 & -0.085 & -1.69 \\ & 0.645 & 1.29 & -0.645 & -1.29 & 0 \\ & & 5.11 & -3.46 & -2.42 & 2.17 \\ & & & 9.66 & 1.05 & -9.01 \\ \text{Symmetric} & & & & 2.62 & 0.241 \\ & & & & & 9.01 \end{bmatrix} \end{array}$$

Using the elimination approach, on assembling the matrices with reference to the degrees of freedom (dof) 1 and 3, we get

$$10^7 = \begin{bmatrix} 4.03 & -2.34 \\ -2.34 & 4.35 \end{bmatrix} \begin{Bmatrix} Q_1 \\ Q_3 \end{Bmatrix} = \begin{Bmatrix} 2514 \\ 2514 \end{Bmatrix}$$

so that

$$Q_1 = 0.014 \times 10^{-2}\ \text{mm}$$

$$Q_3 = 0.0133 \times 10^{-2}\ \text{mm}$$ ■

Stress Calculations

From the set of nodal displacements obtained above, the element nodal displacements $\mathbf{q}$ can be found using the connectivity. Then, using stress–strain relation in Eq. 7.8 and strain–displacement relation in Eq. 7.24, we have

$$\boldsymbol{\sigma} = \mathbf{D}\overline{\mathbf{B}}\mathbf{q} \tag{7.50}$$

where $\overline{\mathbf{B}}$ is $\mathbf{B}$, given in Eq. 7.25, evaluated at the centroid of the element. We also note that σ_θ is a principal stress. The two principal stresses σ_1 and σ_2 corresponding to σ_r, σ_z, and τ_{rz} can be calculated using Mohr's circle.

Example 7.3

Calculate the element stresses in the problem discussed in Example 7.2.

Solution We need to find $\sigma^{e^{\mathrm{T}}} = [\sigma_r \quad \sigma_z \quad \tau_{rz} \quad \sigma_\theta]^e$ for each element. From the connectivity established in Example 7.2,

$$\mathbf{q}^1 = [0.0140 \quad 0 \quad 0.0133 \quad 0 \quad 0 \quad 0]^{\mathrm{T}} \times 10^{-2}$$

$$\mathbf{q}^2 = [0.0133 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}} \times 10^{-2}$$

Using the product matrices $\mathbf{DB}^e$ and $\mathbf{q}$ in the formula

$$\boldsymbol{\sigma}^e = \mathbf{D}\overline{\mathbf{B}}^e\mathbf{q}$$

we get

$$\boldsymbol{\sigma}^1 = [-166 \quad -58.2 \quad 5.4 \quad -28.4]^{\mathrm{T}} \times 10^{-2}\,\mathrm{MPa}$$

$$\boldsymbol{\sigma}^2 = [-169.3 \quad -66.9 \quad 0 \quad -54.1]^{\mathrm{T}} \times 10^{-2}\,\mathrm{MPa}$$

■

Temperature Effects

Uniform increase in temperature of ΔT introduces initial normal strains $\boldsymbol{\epsilon}_0$ given as

$$\boldsymbol{\epsilon}_0 = [\alpha\Delta T \quad \alpha\Delta T \quad 0 \quad \alpha\Delta T]^{\mathrm{T}} \tag{7.51}$$

The stresses are given by

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) \tag{7.52}$$

where $\boldsymbol{\epsilon}$ is the total strain.

Substituting into the strain energy yields an additional term of $-\boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}_0$ in the potential energy Π. Using the element strain–displacement relation in Eq. 7.24, we get

$$2\pi\int_A \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}_0 r\,dA = \sum_e \mathbf{q}^{\mathrm{T}}(2\pi\bar{r}A_e\overline{\mathbf{B}}^{\mathrm{T}}\mathbf{D}\overline{\boldsymbol{\epsilon}}_0) \tag{7.53}$$

Considering the temperature effect in the Galerkin approach is rather simple. The term $\boldsymbol{\epsilon}^{\mathrm{T}}$ in Eq. 7.53 is replaced by $\boldsymbol{\epsilon}^{\mathrm{T}}(\boldsymbol{\phi})$.

The expression in parentheses gives element nodal load contributions. The vector $\overline{\boldsymbol{\epsilon}}_0$ is the initial strain evaluated at the centroid, representing the average temperature rise of the element. We have

$$\boldsymbol{\Theta}^e = 2\pi\bar{r}A_e\overline{\mathbf{B}}^{\mathrm{T}}\mathbf{D}\overline{\boldsymbol{\epsilon}}_0 \tag{7.54}$$

where

$$\boldsymbol{\Theta}^e = [\Theta_1 \quad \Theta_2 \quad \Theta_3 \quad \Theta_4 \quad \Theta_5 \quad \Theta_6]^{\mathrm{T}} \tag{7.55}$$

7.4 PROBLEM MODELING AND BOUNDARY CONDITIONS

We have seen that the axisymmetric problem simply reduces to consideration of the revolving area. The boundary conditions need to be enforced on this area. θ independence arrests the rotation. Axisymmetry also implies that points lying on the z-axis remain radially fixed. Let us now consider some typical problems with a view to modeling them.

Cylinder Subjected to Internal Pressure

Figure 7.7 shows a hollow cylinder of length L subjected to an internal pressure. One end of the cylindrical pipe is attached to a rigid wall. In this, we need to model only the rectangular region of the length L bound between r_i and r_0. Nodes at the fixed end are constrained in the z and r directions. Stiffness and force modifications will be made for these nodes.

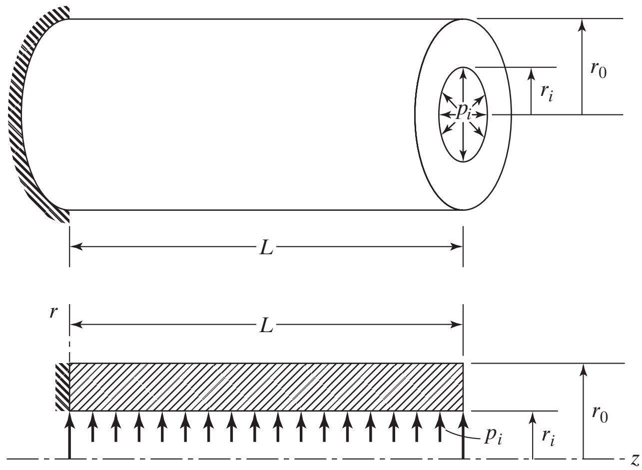


FIGURE 7.7 Hollow cylinder under internal pressure.

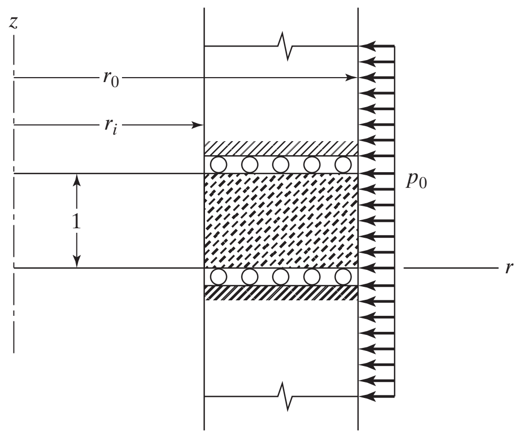


FIGURE 7.8 Cylinder of infinite length under external pressure.

Infinite Cylinder

Modeling of a cylinder of infinite length subjected to external pressure is shown in Fig. 7.8. The length dimensions are assumed to remain constant. This plane strain condition is modeled by considering a unit length and restraining the end surfaces in the z direction.

Press Fit on a Rigid Shaft

Press fit of a ring of length L and internal radius r_i onto a rigid shaft of radius $r_i + \delta$ is shown in Fig. 7.9. When symmetry is assumed about the midplane, this plane is restrained in the z direction. When we impose the condition that nodes at the internal radius have to displace radially by δ, a large stiffness C is added to the diagonal locations for the radially constrained dof and a force $C\delta$ is added to the corresponding force components. Solution of the equations gives displacements at nodes; stresses can then be evaluated.

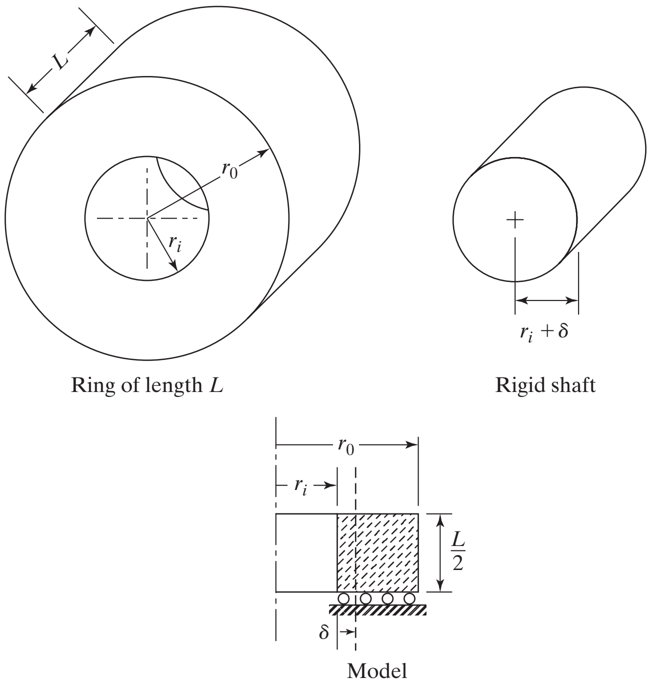


FIGURE 7.9 Press fit on a rigid shaft.

Press Fit on an Elastic Shaft

The condition at the contacting boundary leads to an interesting problem when an elastic sleeve is press fitted onto an elastic shaft. Consider the problem of Fig. 7.9 with the shaft also treated as elastic. A method to handle this is considered by referring to Fig. 7.10. We may define pairs of nodes on the contacting boundary, each pair consisting of one node on the sleeve and one on the shaft. If Q_i and Q_j are displacements of a typical pair along the radial dof, we need to satisfy the multipoint constraint

$$Q_j - Q_i = \delta \tag{7.56}$$

When the term $(\frac{1}{2})C(Q_j - Q_i - \delta)^2$ is added to the potential energy, the constraint is approximately enforced. The penalty approach for handling multipoint constraints is discussed in Chapter 3. Note that C is a large number. We have

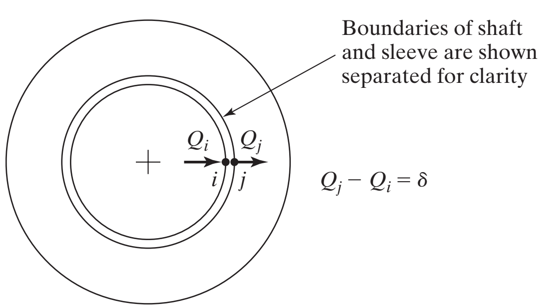


FIGURE 7.10 Elastic sleeve on an elastic shaft.

$$\frac{1}{2}C(Q_j - Q_i - \delta)^2 = \frac{1}{2}CQ_i^2 + \frac{1}{2}CQ_j^2 - \frac{1}{2}C(Q_iQ_j + Q_jQ_i)$$
$$+ CQ_i\delta - CQ_j\delta + \frac{1}{2}C\delta^2 \tag{7.57}$$

This implies the following modifications:

$$\begin{bmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{bmatrix} \rightarrow \begin{bmatrix} K_{ii} + C & K_{ij} - C \\ K_{ji} - C & K_{jj} + C \end{bmatrix} \tag{7.58}$$

and

$$\begin{bmatrix} F_i \\ F_j \end{bmatrix} \rightarrow \begin{bmatrix} F_i - C\delta \\ F_j + C\delta \end{bmatrix} \tag{7.59}$$

Belleville Spring

The Belleville spring, also called the Belleville washer, is a conical disk spring. The load is applied on the periphery of the circle and supported at the bottom as shown in Fig. 7.11a. As the load is applied in the axial direction, the supporting edge moves out. Only the shaded rectangular area shown in Fig. 7.11c needs to be modeled. An axisymmetric load P is placed at the top corner, and the bottom supporting corner is constrained in the z direction. Load–deflection characteristics and stress distribution can be obtained by dividing the area into elements and using a computer program. In the Belleville spring, the load–deflection curve is *nonlinear* (Fig. 7.11b). The stiffness depends on the geometry. We can find a good approximate solution by an incremental

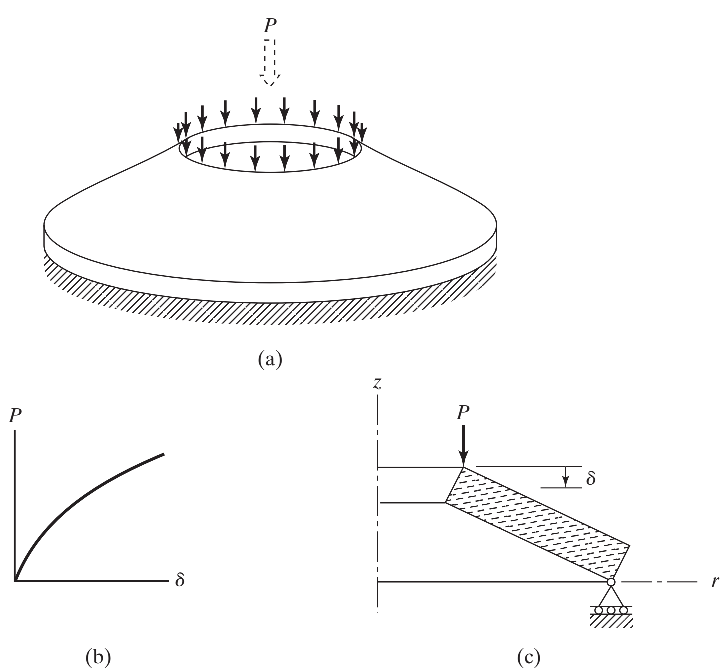


FIGURE 7.11 Belleville spring.

approach. We find the stiffness matrix $\mathbf{K}(\mathbf{x})$ for the given coordinate geometry. We obtain the displacements $\Delta\mathbf{Q}$ for an incremental loading of $\Delta\mathbf{F}$ from

$$\mathbf{K}(\mathbf{x})\,\Delta\mathbf{Q} = \Delta\mathbf{F} \tag{7.60}$$

The displacements $\Delta\mathbf{Q}$ are converted to the components Δu and Δw and are added to $\mathbf{x}$ to update the new geometry:

$$\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{u} \tag{7.61}$$

$\mathbf{K}$ is recalculated for the new geometry, and the new set of equations (Eq. 7.60) is solved. The process is continued until the full applied load is reached.

This example illustrates the incremental approach for geometric nonlinearity.

Thermal Stress Problem

A steel sleeve inserted into a rigid insulated wall is shown in Fig. 7.12a. The sleeve fits snugly, and then the temperature is raised by ΔT. The stresses in the sleeve increase because of the constraint. The rectangular area of length $L/2$, bounded by r_i and r_0 is considered (Fig. 7.12b), with points on the outer radius constrained radially and points on r constrained axially. The load vector is modified using Eq. 7.55, and the finite element equations are solved.

Modeling of simple to complex problems of engineering importance has been discussed. In real life, each problem poses its own challenge. With a clear understanding of the loading, boundary conditions, and the material behavior, the modeling of a problem can be broken down into simple and easy steps.

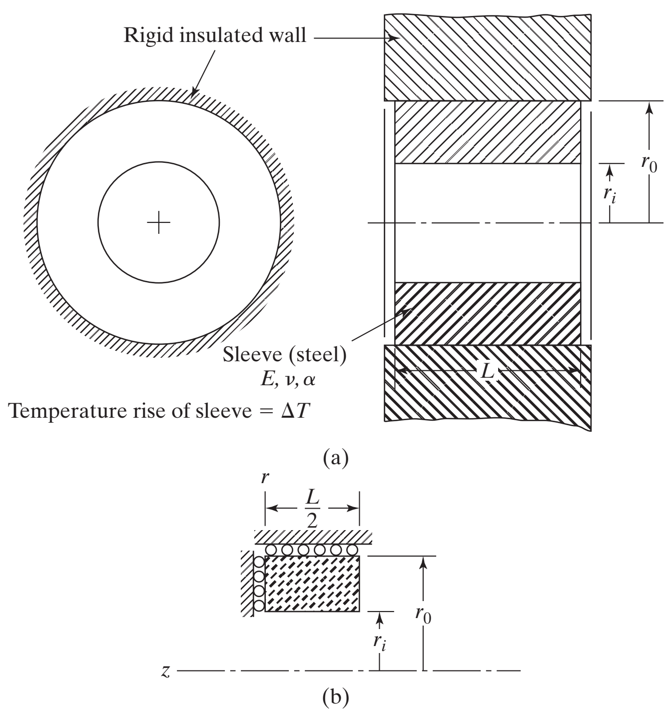


FIGURE 7.12 Thermal stress problem.

Example 7.4

A steel disk flywheel rotates at 3000 rpm. The outer diameter is 24 in., and the hole diameter is 6 in. (Fig. E7.4a). Find the value of the maximum tangential stress under the following conditions: thickness = 1 in., $E = 30 \times 10^6$ psi, Poisson's ratio = 0.3, and weight density = 0.283 lb/in.^{3}.

A four-element finite element model is shown in Fig. E7.4b. The load vector is calculated from Eq. 7.34, neglecting gravity load. The result is

$$\mathbf{F} = [3449, \quad 0, \quad 9580, \quad 0, \quad 23\,380, \quad 0, \quad 38\,711, \quad 0, \quad 32\,580, \quad 0, \quad 18\,780, \quad 0]^{\mathrm{T}}\ \text{lb}$$

The input data for program AXISYM and output are given subsequently.

The computer output gives us the tangential stresses in each of the four elements. Treating these values as centroidal values and *extrapolating* as shown in Fig. E7.4c, the maximum tangential stress occurring at the inner boundary is obtained as $\sigma_{t\,\max} = 8700$ psi. ■

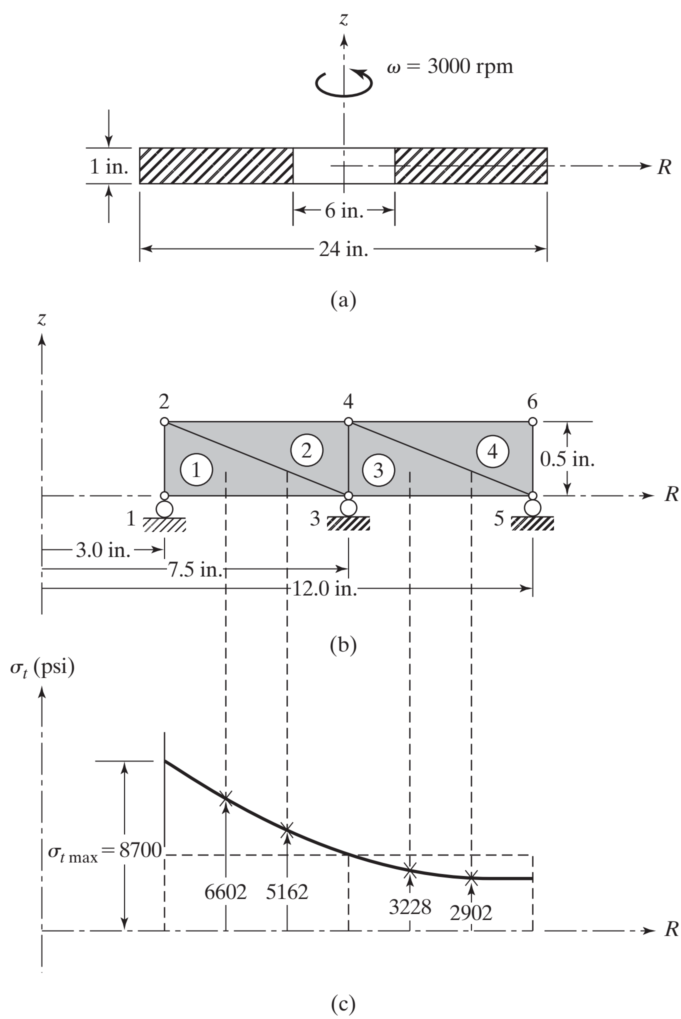


FIGURE E7.4

Input Data/Output

```
INPUT TO AXISYM
<< AXISYMMETRIC STRESS ANALYSIS USING TRIANGULAR ELEMENT >>
EXAMPLE 7.4
NN     NE     NM     NDIM   NEN    NDN
6      4      1      2      3      2
ND     NL     NMPC
3      6      0
Node#  X      Y
1      3      0
2      3      0.5
3      7.5    0
4      7.5    0.5
5      12     0
6      12     0.5
Elem#  N1     N2     N3     Mat#   TempRise
1      1      3      2      1      0
2      2      3      4      1      0
3      4      3      5      1      0
4      4      5      6      1      0
DOF#   Displacement
2      0
6      0
10     0
DOF#   Load
1      3449
3      9580
5      23380
7      38711
9      32580
11     18780
MAT#   E             Nu     Alpha
1      3.00E+07      0.3    1.20E-05
B1     i      B2     j      B3     (Multi-point constr. B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM AXISYM
Program AXISYM - Triangular Element
EXAMPLE 7.4
Node#  R-Displ        Z-Displ
1      0.000900314     3.18925E-12
2      0.000898978    -4.27574E-05
3      0.00090119     -2.55875E-12
4      0.000902908    -2.65201E-05
5      0.000919789    -6.30495E-13
6      0.0009178      -1.93142E-05
Elem#  SR         SZ        Trz       ST         S1         S2        Ang R@s1
1      1989.953    12.044   -30.814   6601.670   1990.433    11.564   -0.892
2      1716.377   472.221    81.294   5161.707   1721.666   466.932    3.723
3       994.991  -324.390    39.660   3227.721    996.182  -325.582    1.720
4       970.838     3.047   -27.421   2902.162    971.615     2.270   -1.622
DOF#   Reaction
2      -548.3645
6       439.9560
10      108.4085
```

PROBLEMS

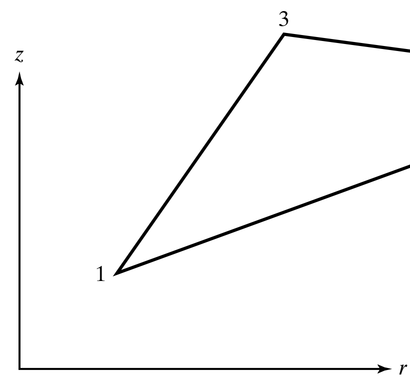
7.1. In an axisymmetric problem, the element coordinates and displacements are as shown in the Fig. P7.1.

(a) What will be the value of the tangential (hoop) stress printed out by program AXISYM?

(b) What are the three principal stresses, σ_1, σ_2, and σ_3?

(c) What is the von Mises stress in the element?

Take $E = 20\,\text{E}6\,\text{psi}$ and $\nu = 0.3$. Coordinates and displacements are in inches.



Point	(R, Z-coord)	(R, Z-disp.)
1	(1,1)	(0,0)
2	(10,4)	(−0.2, −0.1)
3	(6,7)	(0.6, 0.8)

FIGURE P7.1

7.2. The open-ended steel cylinder shown in Fig. P7.2 is subjected to an internal pressure of 1 MPa. Find the deformed shape and distribution of principal stresses.

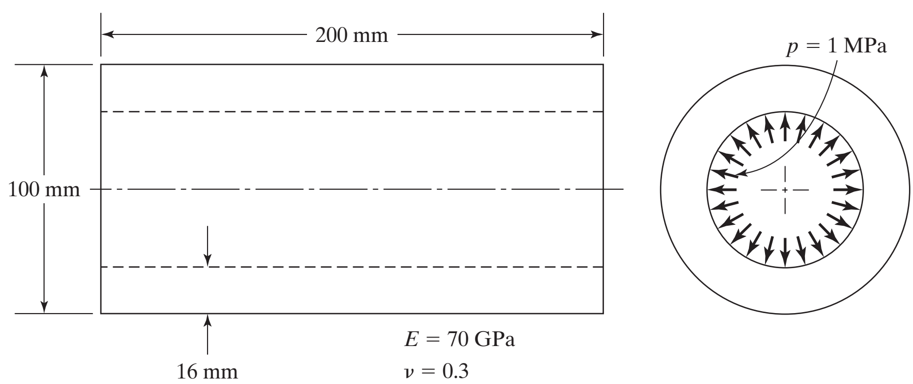


FIGURE P7.2

7.3. Find the deformed configuration and stress distribution at the walls of the closed cylinder shown in Fig. P7.3.

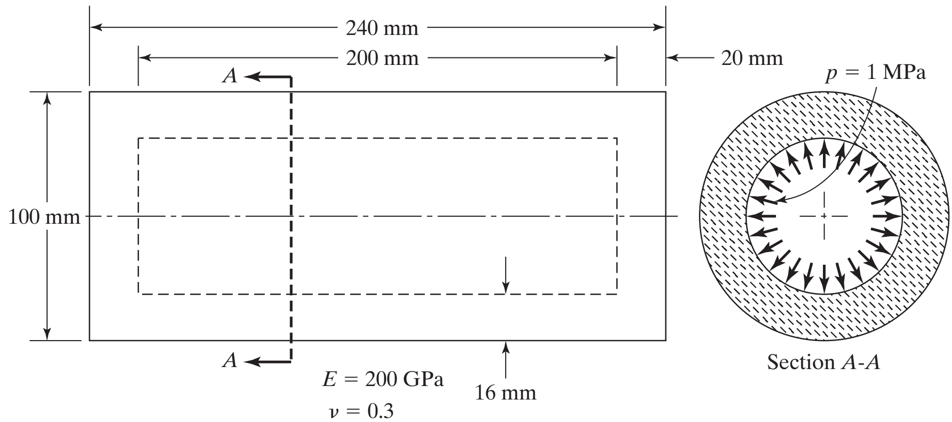


FIGURE P7.3

7.4. Determine the diameters after deformation and distribution of principal stresses along the radius of the infinite cylinder subjected to internal pressure as shown in Fig. P7.4.

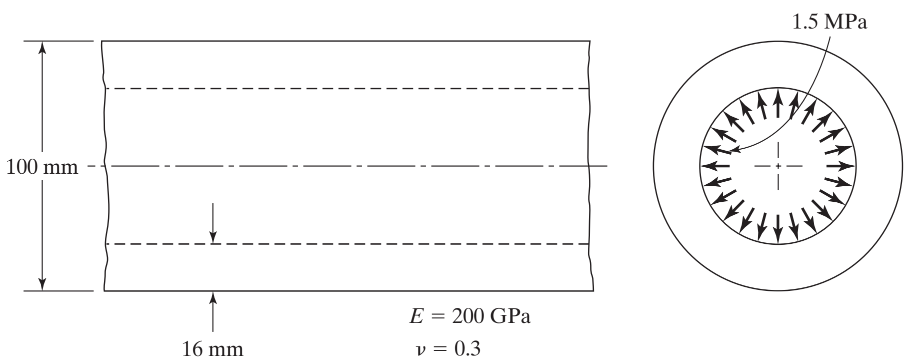


FIGURE P7.4

7.5. The steel sleeve of internal diameter 3 in. is press fitted onto a rigid shaft of diameter 3.01 in., as shown in Fig. P7.5. Determine (a) the outer diameter of the sleeve after fitting and (b) the stress distribution. Estimate the contact pressure by interpolating the radial stress in the neighboring elements.

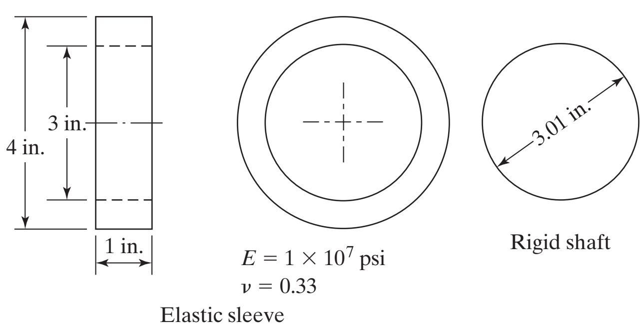


FIGURE P7.5

7.6. Solve Problem 7.5 if the shaft is also made out of steel.

7.7. The steel flywheel shown in Fig. P7.7 rotates at 3000 rpm. Find the deformed shape of the flywheel and give the stress distribution.

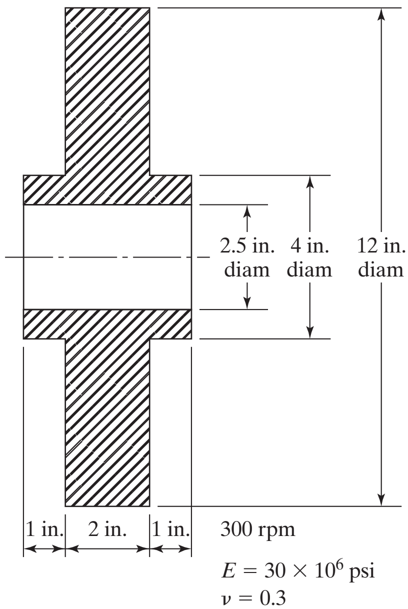


FIGURE P7.7 Flywheel.

7.8. The circular pad hydrostatic bearing shown in Fig. P7.8 is used for supporting slides subjected to large forces. Oil under pressure is supplied through a small hole at the center and flows out through the gap. The pressure distribution in the pocket area and the gap is shown in the figure. Find the deformed configuration of the pad and determine the stress distribution. (*Note:* Neglect the dimension of the oil supply hole.)

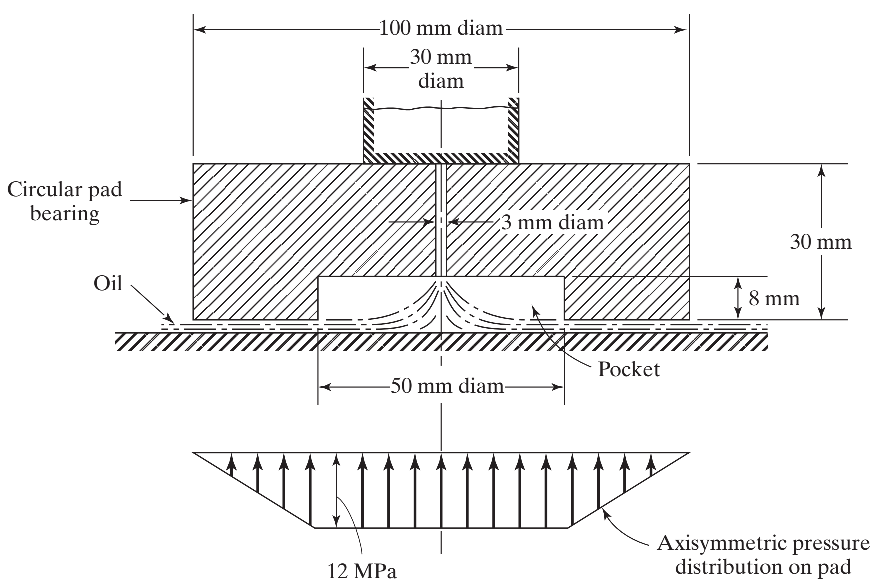


FIGURE P7.8 Hydrostatic bearing.

7.9. A Belleville spring is a conical disk spring. Determine the axial load required to flatten the spring shown in Fig. P7.9. Solve the problem using the incremental approach discussed in this chapter and plot the load–deflection curve as the spring flattens.

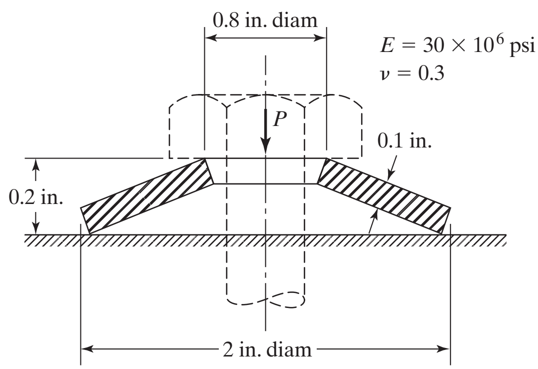


FIGURE P7.9 Belleville spring.

7.10. The aluminum tube shown in Fig. P7.10 fits snugly into a rigid hole at room temperature. If the temperature of the aluminum tube is increased by 40°C, find the deformed configuration and stress distribution.

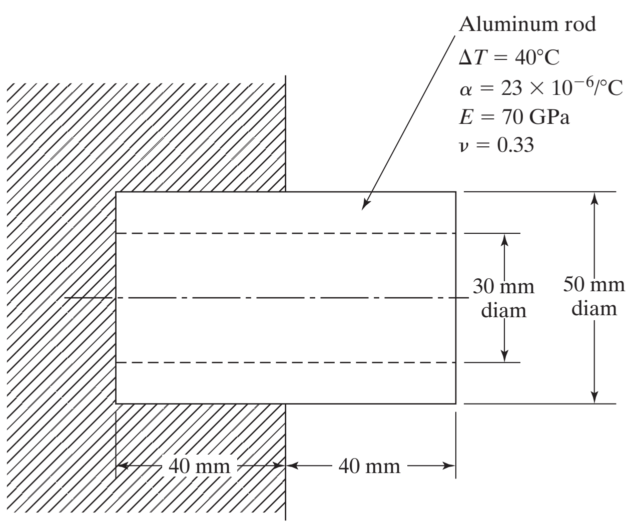


FIGURE P7.10

7.11. The steel water tank shown in Fig. P7.11 is bolted to a 5-m circular support. If the water is at a height of 3 m as shown, find the deformed shape and stress distribution. (*Note:* Pressure $= \rho g h$, water density $\rho = 1\,\text{Mg/m}^3$, and $g = 9.8\,\text{m/s}^2$.)

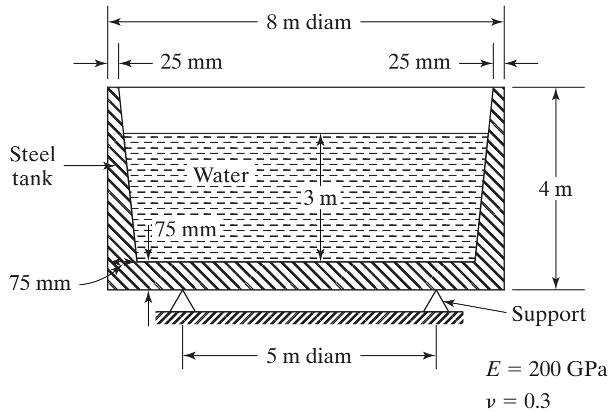


FIGURE P7.11 Water tank.

7.12. For the axisymmetric pressure loading shown in Fig. P7.12, determine the equivalent point loads F_1, F_2, F_3, F_4, F_7, and F_8.

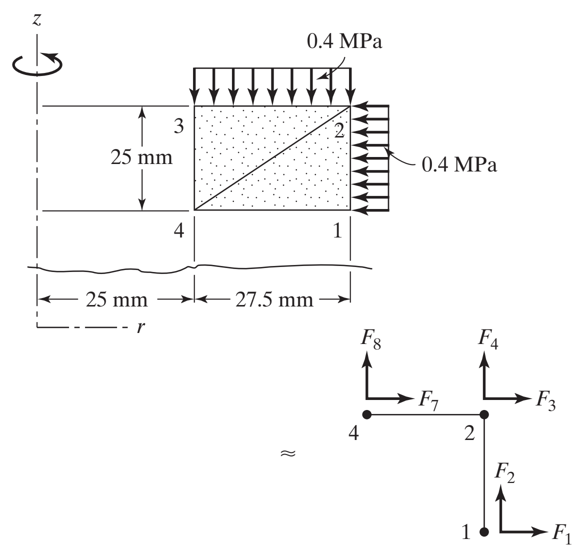


FIGURE P7.12

7.13. For the linearly varying distributed load on the axisymmetric conical surface shown in Fig. P7.13, complete the following:

(a) Prove that the equivalent point load vector $\mathbf{T}$ is given by

$$\mathbf{T} = [aT_{r1} + bT_{r2} \quad aT_{z1} + bT_{z2} \quad bT_{r1} + cT_{r2} \quad bT_{z1} + cT_{z2}]^{\mathrm{T}}$$

where

$$a = \frac{2\pi\ell}{12}(3r_1 + r_2), \quad b = \frac{2\pi\ell}{12}(r_1 + r_2), \quad c = \frac{2\pi\ell}{12}(r_1 + 3r_2)$$

(b) Solve the problem discussed in Example 7.1 (Fig. E7.1), and check by how much the more precise calculations from (a) differ from the approximation of piecewise uniform loads.

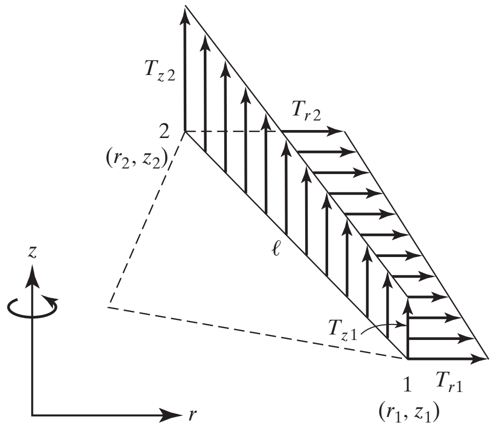


FIGURE P7.13

7.14. A cup-shaped steel die block snugly fitted in a shrink ring is shown in Fig. P7.14a. The punch applies force on a slug placed in the die block to produce a cup-shaped part. If the process is modeled by linearly varying pressures (*use the results from Problem 7.13 to compute nodal loads*) on the die block as shown in Fig. P7.14b, determine the location and magnitude of maximum principal stresses in the die block for the following cases:

(a) The die block modeled without the shrink ring

(b) The die block modeled with the shrink ring with no slip between the shrink ring and the die block

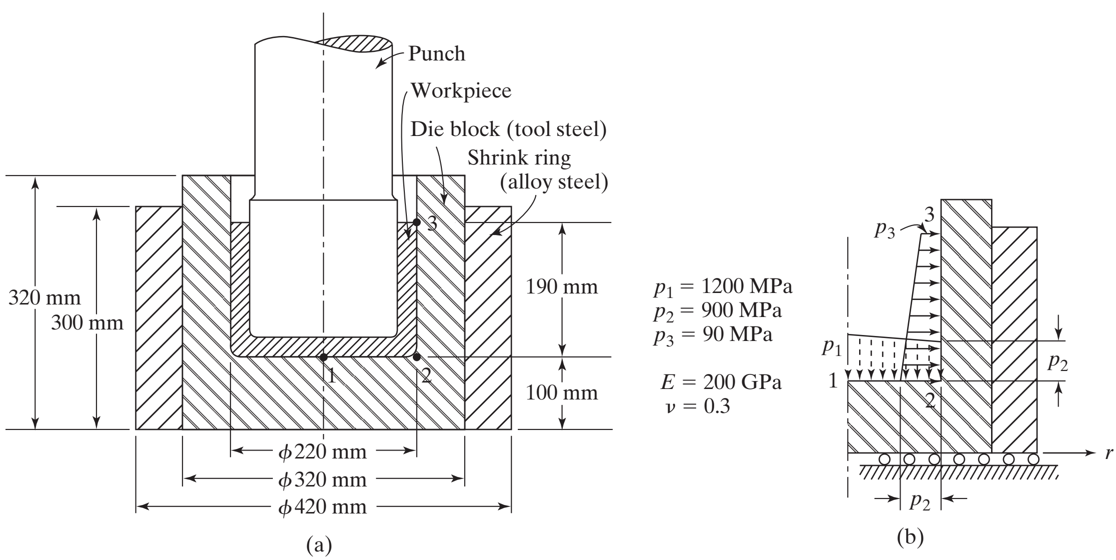


FIGURE P7.14

(c) The die block modeled with the shrink ring with the assumption of frictionless axial slip. (*Hint:* You need duplicate nodes on the interface between the die block and the shrink ring. If I and J are a pair of nodes on the interface, then the multipoint constraint is $Q_{2I-1} - Q_{2J-1} = 0$. Use MESHGEN and DATAFEM programs followed by AXISYM).

7.15. A 90-mm outside diameter steel disk held at 200°C above the room temperature fits snugly onto a steel shaft of 40 mm diameter at room temperature, as shown in Fig. P7.15. Determine the maximum stresses in the disk and the shaft when the assembly reaches room temperature.

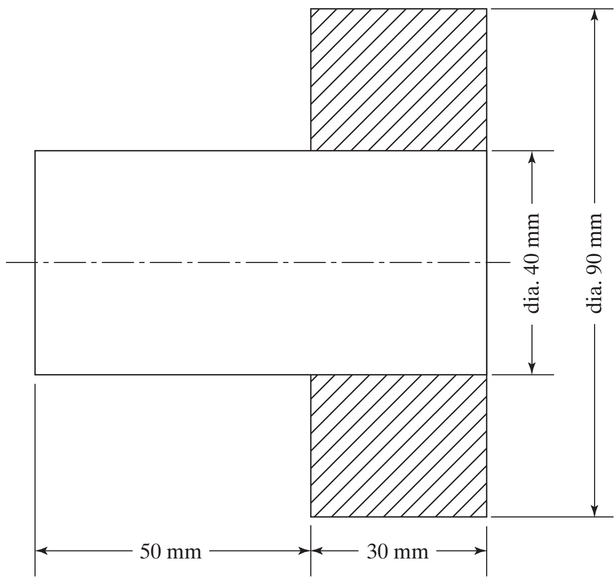


FIGURE P7.15

7.16. A syringe–plunger assembly is shown in Fig. P6.16. Model the glass syringe assuming that the 4-mm hole end is closed under test conditions. Determine the deformation and stresses and compare the maximum principal stress with the ultimate tensile strength of glass.

7.17. A steel sleeve of length 30 mm is snugly fitted (i.e., clearance = 0) over a solid copper cylinder of diameter 50 mm as shown in Fig. P7.17. The outside diameter of the sleeve is 80 mm. The temperature of the shaft is then raised by 30°C. Treat this as an axisymmetric analysis problem. Find:

(a) The area of revolution with a simple mesh with just a few elements.

(b) Number all the nodes, using different node numbers for common nodes on the sleeve–shaft interface.

(c) Write down all boundary conditions for analysis.

Solve the problem using AXISYM.

7.18. The three-node triangular axisymmetric element shown is rotating with constant angular velocity ω rads/s about the z-axis. Determine the 6×1 element load vector $\mathbf{f}$ due to the resulting body force. Give the answer in terms of ω and material density ρ.

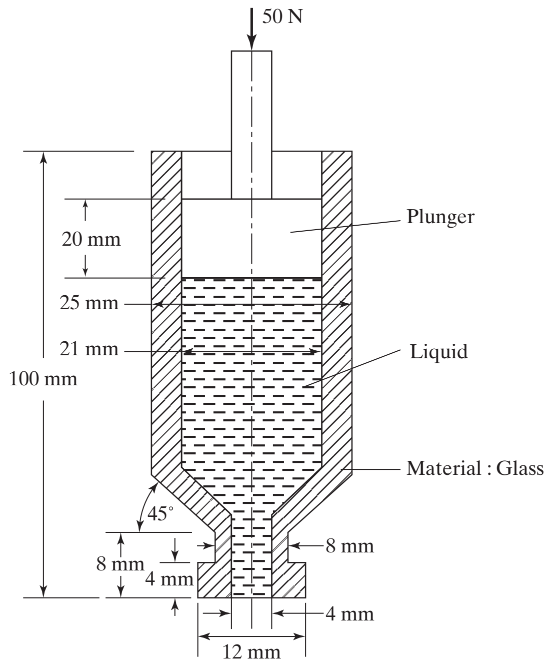


FIGURE P7.16

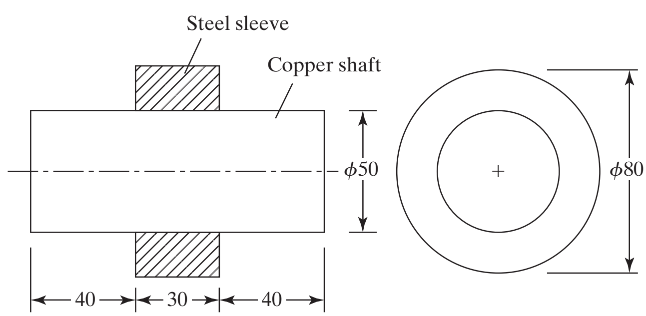


FIGURE P7.17

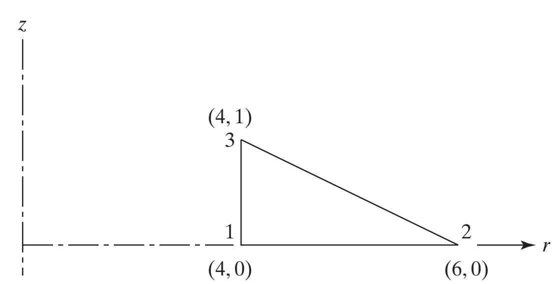


FIGURE P7.18

Program Listing

```
MAIN PROGRAM
'****************************************
'*          PROGRAM  AXISYM             *
'*    AXISYMMETRIC STRESS ANALYSIS      *
'*          WITH TEMPERATURE            *
'*  T.R.Chandrupatla and A.D.Belegundu  *
'****************************************
Const PI = 3.14159265358979
Private Sub CommandButton1_Click()
    Call InputData
    Call Bandwidth
    Call Stiffness
    Call ModifyForBC
    Call BandSolver
    Call StressCalc
    Call ReactionCalc
    Call Output
End Sub
```

```
STIFFNESS
Private Sub Stiffness()
     ReDim S(NQ, NBW)
     '----- Global Stiffness Matrix -----
     For N = 1 To NE
     Call DbMat(N, 1, RBAR)
         '--- Element Stiffness
         For I = 1 To 6
            For J = 1 To 6
                C = 0
                For K = 1 To 4
                    C = C + Abs(DJ) * B(K, I) * DB(K, J) * PI * RBAR
                Next K
                SE(I, J) = C
            Next J
         Next I
     '--- Temperature Load Vector
         AL = PM(MAT(N), 3)
         C = AL * DT(N) * PI * RBAR * Abs(DJ)
         For I = 1 To 6
             TL(I) = C * (DB(1, I) + DB(2, I) + DB(4, I))
         Next I
      '--- <<<Stiffness assembly same as other programs>>>
End Sub
```

```
D MATRIX, B MATRIX, DB MATRIX
Private Sub DbMat(N, ISTR, RBAR)
      '----- D(), B() AND DB() matrices
      '--- First the D-Matrix
```

continued

```
      M = MAT(N): E = PM(M, 1): PNU = PM(M, 2): AL = PM(M, 3)
      C1 = E * (1 - PNU) / ((1 + PNU) * (1 - 2 * PNU))
      C2 = PNU / (1 - PNU)
      For I = 1 To 4: For J = 1 To 4: D(I, J) = 0: Next J: Next I
      D(1, 1) = C1: D(1, 2) = C1 * C2: D(1, 4) = C1 * C2
      D(2, 1) = D(1, 2): D(2, 2) = C1: D(2, 4) = C1 * C2
      D(3, 3) = 0.5 * E / (1 + PNU)
      D(4, 1) = D(1, 4): D(4, 2) = D(2, 4): D(4, 4) = C1
      '--- Strain-Displacement Matrix B()
      I1 = NOC(N, 1): I2 = NOC(N, 2): I3 = NOC(N, 3)
      R1 = X(I1, 1): Z1 = X(I1, 2)
      R2 = X(I2, 1): Z2 = X(I2, 2)
      R3 = X(I3, 1): Z3 = X(I3, 2)
      R21 = R2 - R1: R32 = R3 - R2: R13 = R1 - R3
      Z12 = Z1 - Z2: Z23 = Z2 - Z3: Z31 = Z3 - Z1
      DJ = R13 * Z23 - R32 * Z31   'Determinant of Jacobian
      RBAR = (R1 + R2 + R3) / 3
      '--- Definition of B() Matrix
      B(1, 1) = Z23 / DJ: B(2, 1) = 0
      B(3, 1) = R32 / DJ: B(4, 1) = 1 / (3 * RBAR)
      B(1, 2) = 0: B(2, 2) = R32 / DJ: B(3, 2) = Z23 / DJ: B(4, 2) = 0
      B(1, 3) = Z31 / DJ: B(2, 3) = 0: B(3, 3) = R13 / DJ
      B(4, 3) = 1 / (3 * RBAR)
      B(1, 4) = 0: B(2, 4) = R13 / DJ: B(3, 4) = Z31 / DJ: B(4, 4) = 0
      B(1, 5) = Z12 / DJ: B(2, 5) = 0: B(3, 5) = R21 / DJ
      B(4, 5) = 1 / (3 * RBAR)
      B(1, 6) = 0: B(2, 6) = R21 / DJ: B(3, 6) = Z12 / DJ: B(4, 6) = 0
      '--- DB Matrix DB = D*B
      For I = 1 To 4
          For J = 1 To 6
              DB(I, J) = 0
              For K = 1 To 4
                  DB(I, J) = DB(I, J) + D(I, K) * B(K, J)
              Next K
          Next J
     Next I
     If ISTR = 2 Then
        '----- Stress Evaluation -----
        Q(1) = F(2 * I1 - 1): Q(2) = F(2 * I1)
        Q(3) = F(2 * I2 - 1): Q(4) = F(2 * I2)
        Q(5) = F(2 * I3 - 1): Q(6) = F(2 * I3)
        C1 = AL * DT(N)
        For I = 1 To 4
           C = 0
           For K = 1 To 6
              C = C + DB(I, K) * Q(K)
           Next K
           STR(I) = C - C1 * (D(I, 1) + D(I, 2) + D(I, 4))
        Next I
      End If
End Sub
```

CHAPTER 8

Two-Dimensional Isoparametric Elements and Numerical Integration

8.1 INTRODUCTION

In Chapters 6 and 7, we have developed the constant-strain triangular (CST) element for stress analysis. In this chapter, we develop four-node and higher order *isoparametric* elements and apply them to stress analysis. These elements have proved effective on a wide variety of two- and three-dimensional problems in engineering. We discuss the two-dimensional four-node quadrilateral in detail. Development of higher order elements follows the same basic steps used in the four-node quadrilateral. The higher order elements can capture variations in stress near fillets, holes, etc. We can view the isoparametric family of elements in a unified manner due to the simple and versatile manner in which shape functions can be derived, followed by the generation of the element stiffness matrix using numerical integration.

8.2 THE FOUR-NODE QUADRILATERAL

Consider the general quadrilateral element shown in Fig. 8.1. The local nodes are numbered as 1, 2, 3, and 4 in a *counterclockwise* fashion as shown, and (x_i, y_i) are the coordinates of node i. The vector $\mathbf{q} = [q_1, q_2, \ldots, q_8]^T$ denotes the element displacement vector. The displacement of an interior point P located at (x, y) is represented as $\mathbf{u} = [u(x, y), v(x, y)]^T$.

Shape Functions

Following the steps in earlier chapters, we first develop the shape functions on a master element, shown in Fig. 8.2. The master element is defined in ξ-, η-coordinates (or *natural* coordinates) and is square-shaped. The Lagrange shape functions where $i = 1, 2, 3$, and 4, are defined such that N_i is equal to unity at node i and is zero at other nodes. In particular, consider the definition of N_i:

$$\begin{aligned} N_1 &= 1 \quad \text{at node 1} \\ &= 0 \quad \text{at nodes 2, 3, and 4} \end{aligned} \tag{8.1}$$

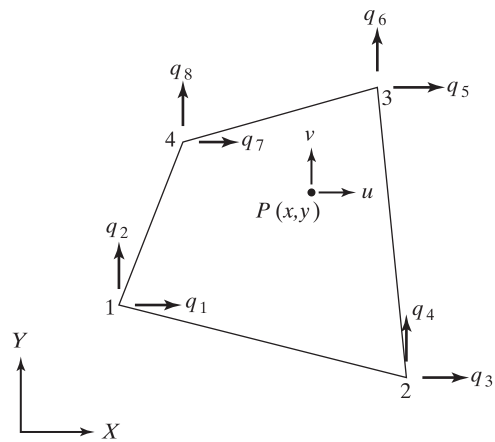


FIGURE 8.1 Four-node quadrilateral element.

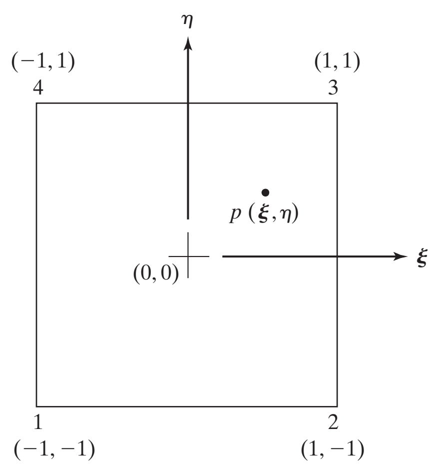


FIGURE 8.2 The quadrilateral element in ξ, η space (the master element).

Now, the requirement that $N_1 = 0$ at nodes 2, 3, and 4 is equivalent to requiring that $N_1 = 0$ along the edges $\xi = +1$ and $\eta = +1$ (Fig. 8.2). Thus, N_1 has to be of the form

$$N_1 = c(1 - \xi)\ (1 - \eta) \tag{8.2}$$

where c is some constant. The constant is determined from the condition $N_1 = 1$ at node 1. Since $\xi = -1,\ \ \eta = -1$ at node 1, we have

$$1 = c(2)(2) \tag{8.3}$$

which yields $c = 1/4$. Thus,

$$N_1 = \tfrac{1}{4}(1 - \xi)(1 - \eta) \tag{8.4}$$

All the four shape functions can be written as

$$
\begin{aligned}
N_1 &= \tfrac{1}{4}(1 - \xi)(1 - \eta) \\
N_2 &= \tfrac{1}{4}(1 + \xi)(1 - \eta) \\
N_3 &= \tfrac{1}{4}(1 + \xi)(1 + \eta) \\
N_4 &= \tfrac{1}{4}(1 - \xi)(1 + \eta)
\end{aligned}
\tag{8.5}
$$

While implementing in a computer program, the compact representation of Eq. 8.5 is useful

$$N_i = \tfrac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) \tag{8.6}$$

where (ξ_i, η_i) are the coordinates of node i.

We now express the displacement field within the element in terms of the nodal values. Thus, if $\mathbf{u} = [u, v]^{\mathrm{T}}$ represents the displacement components of a point located at (ξ, η), and $\mathbf{q}$, of dimension (8×1), is the element displacement vector, then

$$
\begin{aligned}
u &= N_1 q_1 + N_2 q_3 + N_3 q_5 + N_4 q_7 \\
v &= N_1 q_2 + N_2 q_4 + N_3 q_6 + N_4 q_8
\end{aligned}
\tag{8.7a}
$$

which can be written in matrix form as

$$\mathbf{u} = \mathbf{N}\mathbf{q} \tag{8.7b}$$

where

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \tag{8.8}$$

In the isoparametric formulation, we use the *same* shape functions N_i to also express the coordinates of a point within the element in terms of nodal coordinates. Thus,

$$
\begin{aligned}
x &= N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4 \\
y &= N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4
\end{aligned}
\tag{8.9}
$$

Subsequently, we will need to express the derivatives of a function in x-, y-coordinates in terms of its derivatives in ξ-, η-coordinates. This is done as follows: a function $f = f(x, y)$, according to Eq. 8.9, can be considered to be an implicit function of ξ and η as $f = f[x(\xi, \eta), y(\xi, \eta)]$. Using the chain rule of differentiation, we have

$$
\begin{aligned}
\frac{\partial f}{\partial \xi} &= \frac{\partial f}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial \xi} \\
\frac{\partial f}{\partial \eta} &= \frac{\partial f}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial \eta}
\end{aligned}
\tag{8.10}
$$

or

$$\begin{Bmatrix} \dfrac{\partial f}{\partial \xi} \\ \dfrac{\partial f}{\partial \eta} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{Bmatrix} \tag{8.11}$$

where $\mathbf{J}$ is the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \tag{8.12}$$

According to Eqs. 8.5 and 8.9, we have

$$\mathbf{J} = \frac{1}{4}\left[\begin{array}{c|c} -(1-\eta)x_1+(1-\eta)x_2+(1+\eta)x_3-(1+\eta)x_4 & -(1-\eta)y_1+(1-\eta)y_2+(1+\eta)y_3-(1+\eta)y_4 \\ -(1-\xi)x_1-(1+\xi)x_2+(1+\xi)x_3+(1-\xi)x_4 & -(1-\xi)y_1-(1+\xi)y_2+(1+\xi)y_3+(1-\xi)y_4 \end{array}\right] \tag{8.13a}$$

$$= \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{8.13b}$$

Equation 8.11 can be inverted as

$$\begin{Bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \dfrac{\partial f}{\partial \xi} \\ \dfrac{\partial f}{\partial \eta} \end{Bmatrix} \tag{8.14a}$$

or

$$\begin{Bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial f}{\partial \xi} \\ \dfrac{\partial f}{\partial \eta} \end{Bmatrix} \tag{8.14b}$$

These expressions will be used in the derivation of the element stiffness matrix.

An additional result that will be needed is the relation

$$dx\,dy = \det \mathbf{J}\, d\xi\, d\eta \tag{8.15}$$

The proof of this result, found in many textbooks on calculus, is given in the appendix.

Element Stiffness Matrix

The stiffness matrix for the quadrilateral element can be derived from the strain energy in the body, given by

$$U = \int_V \frac{1}{2} \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\epsilon} \, dV \tag{8.16}$$

or

$$U = \sum_e t_e \int_e \frac{1}{2} \boldsymbol{\sigma}^{\mathrm{T}} \boldsymbol{\epsilon} \, dA \tag{8.17}$$

where t_e is the thickness of element e.

The strain–displacement relation is

$$\boldsymbol{\epsilon} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} \tag{8.18}$$

By considering $f \equiv u$ in Eq. 8.14b, we have

$$\begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} \tag{8.19a}$$

Similarly,

$$\begin{Bmatrix} \dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} \tag{8.19b}$$

Equations 8.18 and 8.19a, b yield

$$\boldsymbol{\epsilon} = \mathbf{A} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} \tag{8.20}$$

where $\mathbf{A}$ is given by

$$\mathbf{A} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \tag{8.21}$$

Now, using Eq. 8.7a, we get

$$\begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} = \mathbf{Gq} \tag{8.22}$$

where

$$\mathbf{G} = \frac{1}{4}\begin{bmatrix} -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) & 0 \\ -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) & 0 \\ 0 & -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) \\ 0 & -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) \end{bmatrix} \tag{8.23}$$

Equations 8.20 and 8.22 now yield

$$\boxed{\boldsymbol{\epsilon} = \mathbf{Bq}} \tag{8.24}$$

where

$$\mathbf{B} = \mathbf{AG} \tag{8.25}$$

The relation $\boldsymbol{\epsilon} = \mathbf{Bq}$ is the desired result. The strain in the element is expressed in terms of its nodal displacement. The stress is now given by

$$\boxed{\boldsymbol{\sigma} = \mathbf{DBq}} \tag{8.26}$$

where $\mathbf{D}$ is a (3×3) material matrix. The strain energy in Eq. 8.17 becomes

$$U = \sum_e \frac{1}{2}\mathbf{q}^{\mathrm{T}}\left[t_e \int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\mathbf{DB} \det \mathbf{J}\, d\xi\, d\eta\right]\mathbf{q} \tag{8.27a}$$

$$= \sum_e \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} \tag{8.27b}$$

where

$$\boxed{\mathbf{k}^e = t_e \int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\mathbf{DB} \det \mathbf{J}\, d\xi\, d\eta} \tag{8.28}$$

is the element stiffness matrix of dimension (8×8).

We note here that quantities $\mathbf{B}$ and $\det \mathbf{J}$ in the integral in Eq. 8.28 are involved functions of ξ and η, and so the integration has to be performed numerically. Methods of numerical integration are discussed subsequently.

Element Force Vectors

Body Force. A body force is the distributed force per unit volume that contributes to the global load vector $\mathbf{F}$. This contribution can be determined by considering the body force term in the potential energy expression

$$\int_V \mathbf{u}^{\mathrm{T}}\mathbf{f}\,dV \tag{8.29}$$

Using $\mathbf{u} = \mathbf{Nq}$, and treating the body force $\mathbf{f} = [f_x \ \ f_y]^{\mathrm{T}}$ as constant within each element, we get

$$\int_V \mathbf{u}^{\mathrm{T}}\mathbf{f}\,dV = \sum_e \mathbf{q}^{\mathrm{T}}\mathbf{f}^e \tag{8.30}$$

where the (8×1) element body force vector is given by

$$\mathbf{f}^e = t_e\left[\int_{-1}^{1}\int_{-1}^{1} \mathbf{N}^{\mathrm{T}} \det \mathbf{J}\, d\xi\, d\eta\right]\begin{Bmatrix} f_x \\ f_y \end{Bmatrix} \tag{8.31}$$

As with the stiffness matrix derived earlier, this body force vector has to be evaluated by numerical integration.

Traction Force. Assume that a constant traction force $\mathbf{T} = [T_x \ \ T_y]^{\mathrm{T}}$—a force per unit area—is applied on edge 2–3 of the quadrilateral element. Along this edge, we have $\xi = 1$. If we use the shape functions given in Eq. 8.5, this becomes $N_1 = N_4 = 0$, $N_2 = (1 - \eta)/2$, and $N_3 = (1 + \eta)/2$. Note that the shape functions are linear functions along the edges. Consequently, from the potential, the element traction load vector is readily given by

$$\mathbf{T}^e = \frac{t_e \ell_{2-3}}{2}[0 \quad 0 \quad T_x \quad T_y \quad T_x \quad T_y \quad 0 \quad 0]^{\mathrm{T}} \tag{8.32}$$

where $\ell_{2-3} =$ length of edge 2–3. For varying distributed loads, we may express T_x and T_y in terms of values of nodes 2 and 3 using shape functions. Numerical integration can be used in this case.

Finally, **point** loads are considered in the usual manner by having a structural node at that point and simply adding to the global load vector $\mathbf{F}$.

8.3 NUMERICAL INTEGRATION

Consider the problem of numerically evaluating a one-dimensional integral of the form

$$I = \int_{-1}^{1} f(\xi)\, d\xi \tag{8.33}$$

The *Gauss–Legendre quadrature* approach for evaluating I is given subsequently. This method has proved most useful in finite element work. Extension to integrals in two and three dimensions follows readily.

Consider the n-point approximation

$$I = \int_{-1}^{1} f(\xi)\,d\xi \approx w_1 f(\xi_1) + w_2 f(\xi_2) + \cdots + w_n f(\xi_n) \tag{8.34}$$

where $w_1, w_2, \ldots,$ and w_n are the **weights** and $\xi_1, \xi_2, \ldots,$ and ξ_n are the sampling points or **Gauss–Legendre points**. The idea behind Gauss–Legendre quadrature is to select the n Gauss points and n weights such that Eq. 8.34 provides an exact answer for polynomials $f(\xi)$ of as large a degree as possible. In other words, the idea is that if the n-point integration formula is exact for all polynomials up to as high a degree as possible, then the formula will work well even if f is not a polynomial. To get some intuition for the method, the one-point and two-point approximations are discussed in the sections that follow.

One-Point Formula. Consider the formula with $n = 1$ as

$$\int_{-1}^{1} f(\xi)\,d\xi \approx w_1 f(\xi_1) \tag{8.35}$$

Since there are two parameters, w_1 and ξ_1, we consider requiring the formula in Eq. 8.35 to be exact when $f(\xi)$ is a polynomial of order 1. Thus, if $f(\xi) = a_0 + a_1\xi$, then we require

$$\text{Error} = \int_{-1}^{1} (a_0 + a_1\xi)\,d\xi - w_1 f(\xi_1) = 0 \tag{8.36a}$$

$$\text{Error} = 2a_0 - w_1(a_0 + a_1\xi_1) = 0 \tag{8.36b}$$

or

$$\text{Error} = a_0(2 - w_1) - w_1 a_1 \xi_1 = 0 \tag{8.36c}$$

From Eq. 8.36c, we see that the error is zeroed if

$$w_1 = 2 \quad \xi_1 = 0 \tag{8.37}$$

For any general f, then, we have

$$I = \int_{-1}^{1} f(\xi)\,d\xi \approx 2f(0) \tag{8.38}$$

which is seen to be the familiar *midpoint rule* (Fig. 8.3).

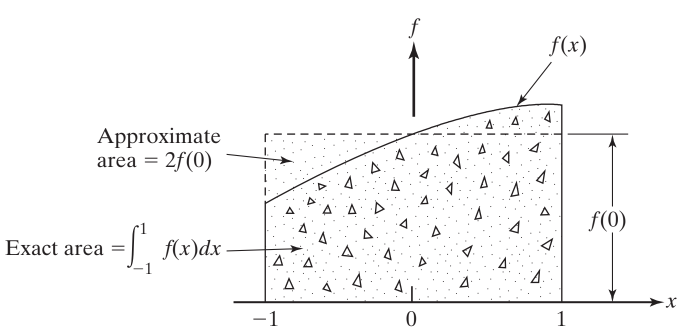


FIGURE 8.3 One-point Gauss quadrature.

Two-Point Formula. Consider the formula with $n = 2$ as

$$\int_{-1}^{1} f(\xi)\, d\xi \approx w_1 f(\xi_1) + w_2 f(\xi_2) \tag{8.39}$$

We have four parameters to choose: w_1, w_2, ξ_1, and ξ_2. We can therefore expect Eq. 8.39 to be exact for a cubic polynomial. Thus, choosing $f(\xi) = a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3$ yields

$$\text{Error} = \left[\int_{-1}^{1} (a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3)\, d\xi\right] - [w_1 f(\xi_1) + w_2 f(\xi_2)] \tag{8.40}$$

Requiring zero error yields

$$\begin{aligned} w_1 + w_2 &= 2 \\ w_1\xi_1 + w_2\xi_2 &= 0 \\ w_1\xi_1^2 + w_2\xi_2^2 &= \tfrac{2}{3} \\ w_1\xi_1^3 + w_2\xi_2^3 &= 0 \end{aligned} \tag{8.41}$$

These nonlinear equations have the unique solution:

$$w_1 = w_2 = 1 \quad -\xi_1 = \xi_2 = 1/\sqrt{3} = 0.5773502691\ldots \tag{8.42}$$

From this solution, we can conclude that n-point Gaussian quadrature will provide an exact answer if f is a polynomial of order $(2n - 1)$ or less. The Gauss–Legendre points are more easily obtained by recognizing that they are zeros of Legendre polynomials $P_k(x)$. Some Legendre polynomials and the zeros are given below.

$$P_1(x) = x \quad \text{zero at } x = 0$$

$$P_2(x) = \frac{1}{2}(3x^2 - 1) \quad \text{zeros at } x = \pm\frac{1}{\sqrt{3}}$$

$$P_2(x) = \frac{1}{2}(5x^3 - 3x) \quad \text{zeros at } x = 0, x = \pm\sqrt{\frac{3}{5}}$$

$$\ldots$$

The Legendre polynomials may be generated using

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ (j + 1)P_{j+1}(x) &= (2j + 1)xP_j(x) - jP_{j-1}(x) \quad j = 1 \text{ to } n - 1 \end{aligned} \tag{8.43}$$

Also the derivative $P_n'(x) = [dP_n(x)/dx]$ is given by the relation

$$(x^2 - 1)P_n'(x) = n[xP_n(x) - P_{n-1}(x)] \tag{8.44}$$

The kth zero can be obtained using the starting value $x_0 = \cos(\pi[k - (1/4)]/[n + (1/2)])$ and Newton's iteration

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{8.45}$$

For $i = 0, 1, 2, \ldots$ until convergence is reached.

The kth weight factor is then obtained using

$$w_k = \frac{2}{(1 - x_k^2)[P_n'(x_k)]^2} \tag{8.46}$$

These steps have been implemented in the javascript program *GaussLegendre.html* and the Excel program *GaussLegendre.xls*. These programs can be used to find zeros and weight factors for various values of n.

Another javascript program *GLInteg.html*, which can be used to calculate the integral of a one-variable function on an interval, has been included in the downloadable programs.

Table 8.1 gives the values of w_i and ξ_i for Gauss quadrature formulas of orders $n = 1$ through $n = 6$. Note that the Gauss points are located symmetrically with respect to the origin and that symmetrically placed points have the same weights. Moreover, the large number of digits given in Table 8.1 should be used in the calculations for accuracy (i.e., use double precision on the computer).

TABLE 8.1 Gauss Points and Weights for Gaussian Quadrature

$$\int_{-1}^{1} f(\xi)\, d\xi \approx \sum_{i=1}^{n} w_i f(\xi_i)$$

No. of Points, n	Location, ξ_i	Weights, w_i
1	0.0	2.0
2	$\pm 1/\sqrt{3} = \pm 0.5773502692$	1.0
3	±0.7745966692	0.5555555556
	0.0	0.8888888889
4	±0.8611363116	0.3478548451
	±0.3399810436	0.6521451549
5	±0.9061798459	0.2369268851
	±0.5384693101	0.4786286705
	0.0	0.5688888889
6	±0.9324695142	0.1713244924
	±0.6612093865	0.3607615730
	±0.2386191861	0.4679139346

Example 8.1

Evaluate

$$I = \int_{-1}^{1} \left[3e^x + x^2 + \frac{1}{(x + 2)} \right] dx$$

using one-point and two-point Gauss quadratures.

Solution For $n = 1$, we have $w_1 = 2$, $x_1 = 0$, and

$$I \approx 2f(0)$$
$$= 7.0$$

For $n = 2$, we have $w_1 = w_2 = 1$, $x_1 = -0.57735\ldots$, $x_2 = +0.57735\ldots$, and $I \approx 8.7857$. This may be compared with the exact solution

$$I_{\text{exact}} = 8.8165$$ ■

The integral may be easily checked using the javascript program *GLInteg.html*.

Two-Dimensional Integrals

The extension of Gaussian quadrature to two-dimensional integrals of the form

$$I = \int_{-1}^{1} \int_{-1}^{1} f(\xi, \eta)\, d\xi\, d\eta \tag{8.47}$$

follows readily, since

$$I \approx \int_{-1}^{1} \left[\sum_{i=1}^{n} w_i f(\xi_i, \eta) \right] d\eta$$
$$\approx \sum_{j=1}^{n} w_j \left[\sum_{i=1}^{n} w_i f(\xi_i, \eta_j) \right]$$

or

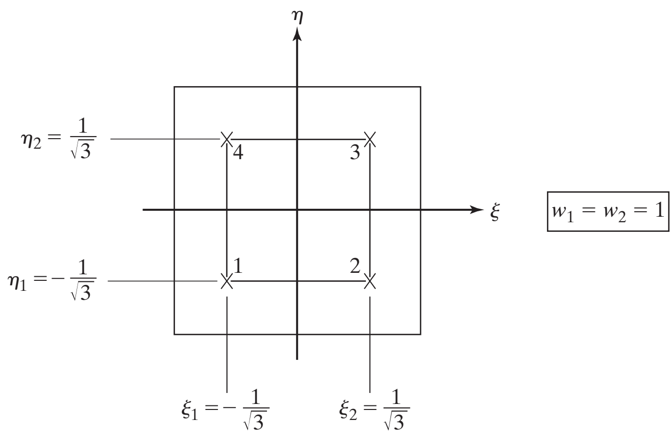
$$I \approx \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j f(\xi_i, \eta_j) \tag{8.48}$$

Stiffness Integration

To illustrate the use of Eq. 8.48, consider the element stiffness for a quadrilateral element

$$\mathbf{k}^e = t_e \int_{-1}^{1} \int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\ \det \mathbf{J}\, d\xi\, d\eta$$

where $\mathbf{B}$ and $\det \mathbf{J}$ are functions of ξ and η. Note that this integral actually consists of the integral of each element in an (8×8) matrix. However, using the fact that $\mathbf{k}^e$ is symmetric, we do not need to integrate elements below the main diagonal.



$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\, d\xi d\eta \approx w_1^2 f(\xi_1,\eta_1) + w_2 w_1 f(\xi_2,\eta_1) + w_2^2 f(\xi_2,\eta_2) + w_1 w_2 f(\xi_1,\eta_2)$$

FIGURE 8.4 Gaussian quadrature in two dimensions using the 2 × 2 rule.

Let ϕ represent the ijth element in the integrand. That is, let

$$\phi(\xi, \eta) = t_e(\mathbf{B}^{\mathrm{T}}\mathbf{DB} \det \mathbf{J})_{ij} \tag{8.49}$$

Then, if we use a 2 × 2 rule, we get

$$k_{ij} \approx w_1^2\phi(\xi_1, \eta_1) + w_1 w_2\phi(\xi_1, \eta_2) + w_2 w_1\phi(\xi_2, \eta_1) + w_2^2\phi(\eta_2, \eta_2) \tag{8.50a}$$

where $w_1 = w_2 = 1.0$, $\xi_1 = \eta_1 = -0.57735\ldots$, and $\xi_2 = \eta_2 = +0.57735\ldots$. The Gauss points for the two-point rule used above are shown in Fig. 8.4. Alternatively, if we label the Gauss points as 1, 2, 3, and 4, then k_{ij} in Eq. 8.50a can also be written as

$$k_{ij} = \sum_{\mathrm{IP}=1}^{4} W_{\mathrm{IP}}\phi_{\mathrm{IP}} \tag{8.50b}$$

where ϕ_{IP} is the value of ϕ and W_{IP} is the weight factor at integration point (IP). We note that $W_{\mathrm{IP}} = (1)(1) = 1$. Computer implementation is sometimes easier using Eq. 8.50b. We may readily follow the implementation of the previous integration procedure in program QUAD provided at the end of this chapter.

The evaluation of three-dimensional integrals is similar. For triangles, however, the weights and Gauss points are different, as discussed later in this chapter.

Stress Calculations

Unlike the CST element (Chapters 6 and 7), the stresses $\boldsymbol{\sigma} = \mathbf{DBq}$ in the quadrilateral element are not constant within the element; they are functions of ξ and η and consequently vary within the element. In practice, the stresses are evaluated at the Gauss points, which are also the points used for numerical evaluation of $\mathbf{k}^e$, where they are found to be accurate. For a quadrilateral with 2×2 integration, this gives four sets of stress values. For generating less data, one may evaluate stresses at one point per element, say, at $\xi = 0$ and $\eta = 0$. The latter approach is used in program QUAD.

Example 8.2

Consider a rectangular element as shown in Fig. E8.2. Assume plane stress condition, $E = 30 \times 10^6$ psi, $\nu = 0.3$, and, $\mathbf{q} = [0, 0, 0.002, 0.003, 0.006, 0.0032, 0, 0]^T$ in. Evaluate $\mathbf{J}$, $\mathbf{B}$, and $\boldsymbol{\sigma}$ at $\xi = 0$ and $\eta = 0$.

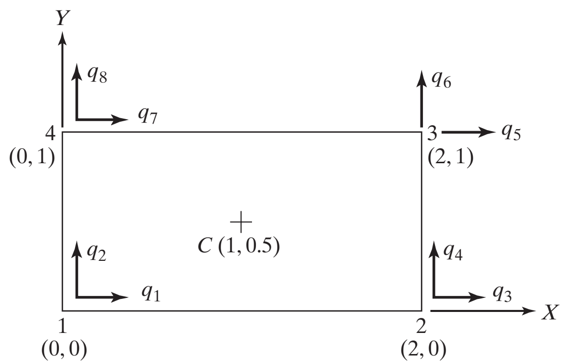


FIGURE E8.2

Solution Referring to Eq. 8.13a, we have

$$\mathbf{J} = \frac{1}{4}\left[\begin{array}{c|c} 2(1-\eta) + 2(1+\eta) & (1+\eta) - (1+\eta) \\ -2(1+\xi) + 2(1+\xi) & (1+\xi) + (1-\xi) \end{array}\right]$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

For this rectangular element, we find that $\mathbf{J}$ is a constant matrix. From Eq. 8.21,

$$\mathbf{A} = \frac{1}{(1/2)}\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & \frac{1}{2} & 0 \end{bmatrix}$$

Evaluating $\mathbf{G}$ in Eq. 8.23 at $\xi = \eta = 0$, and using $\mathbf{B} = \mathbf{QG}$, we get

$$\mathbf{B}^0 = \begin{bmatrix} -\frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{4} & -\frac{1}{2} & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix}$$

The stresses at $\xi = \eta = 0$ are now given by the product

$$\boldsymbol{\sigma}^0 = \mathbf{DB}^0\mathbf{q}$$

For the given data, we have

$$\mathbf{D} = \frac{30 \times 10^6}{(1 - 0.09)}\begin{bmatrix} 1 & 0.3 & 0 \\ 0.03 & 1 & 0 \\ 0 & 0 & 0.35 \end{bmatrix}$$

Thus,

$$\boldsymbol{\sigma}^0 = [66920 \quad 23080 \quad 40960]^T \text{ psi}$$ ■

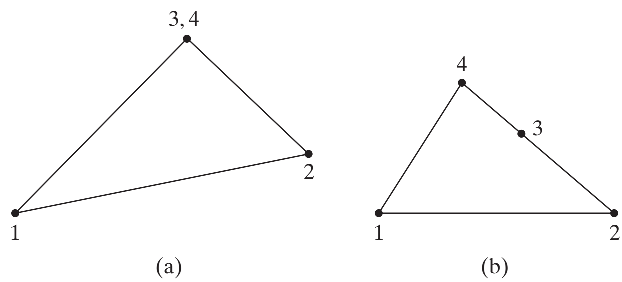


FIGURE 8.5 Degenerate four-node quadrilateral elements.

Comment on Degenerate Quadrilaterals. In some situations, we cannot avoid using degenerated quadrilaterals of the type shown in Fig. 8.5, where quadrilaterals degenerate into triangles. Numerical integration will permit the use of such elements, but the errors are higher than for regular elements. Standard codes normally permit the use of such elements.

8.4 HIGHER ORDER ELEMENTS

The concepts presented earlier for the four-node quadrilateral element can be readily extended to other, higher order, isoparametric elements. In the four-node quadrilateral element, the shape functions contain terms 1, ξ, η, and $\xi\eta$. In addition, the elements to be discussed here also contain terms such as $\xi^2\eta$ and $\xi\eta^2$, which generally provide greater accuracy. Only the shape functions **N** are given in Eq. 8.51. The generation of element stiffness follows the routine steps:

$$\mathbf{u} = \mathbf{Nq} \tag{8.51a}$$

$$\boldsymbol{\epsilon} = \mathbf{Bq} \tag{8.51b}$$

$$\mathbf{k}^e = t_e \int_{-1}^{1} \int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\mathbf{DB} \det \mathbf{J} \, d\xi \, d\eta \tag{8.51c}$$

where $\mathbf{k}^e$ is evaluated using Gaussian quadrature.

Nine-Node Quadrilateral

The nine-node quadrilateral has been found to be very effective in finite element practice. The local node numbers for this element are shown in Fig. 8.6a. The square master element is shown in Fig. 8.6b. The shape functions are defined as follows:

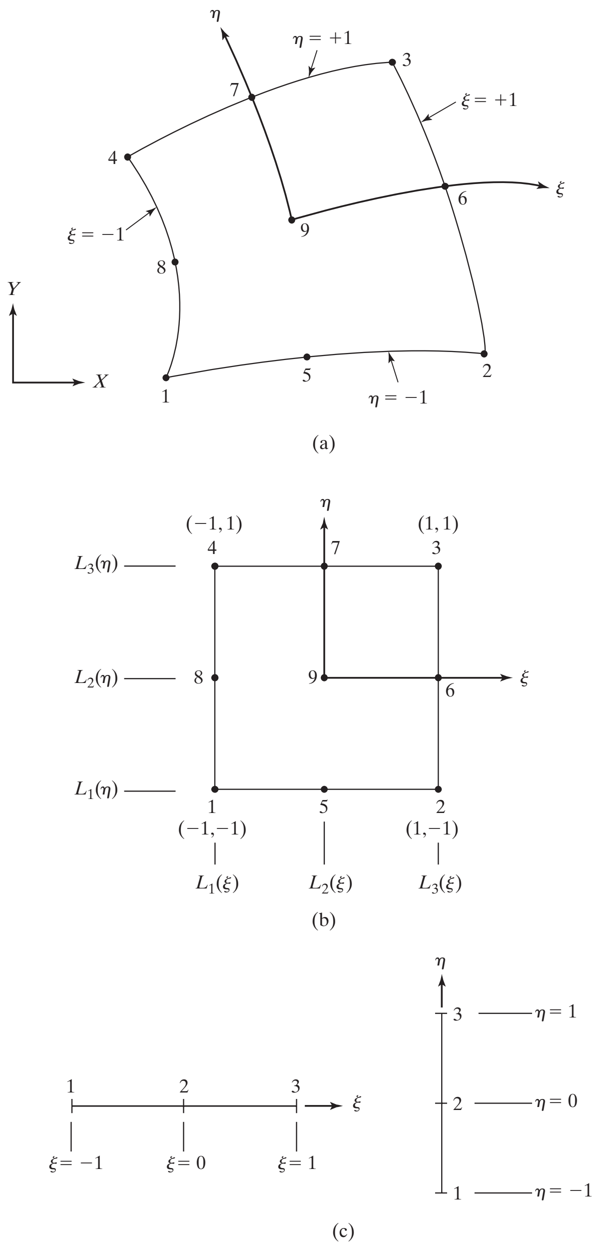


FIGURE 8.6 Nine-node quadrilateral in (a) x, y space and (b) ξ, η space. (c) Definition of general shape functions.

Consider, first, the ξ-axis alone as shown in Fig. 8.6c. The local node numbers 1, 2, and 3 on this axis correspond to locations $\xi = -1, 0$, and $+1$, respectively. At these nodes, we define the generic shape functions L_1, L_2, and L_3 as

$$\begin{aligned} L_i(\xi) &= 1 \quad \text{at node } i \\ &= 0 \quad \text{at other nodes} \end{aligned} \tag{8.52}$$

Now, consider L_1. Since $L_1 = 0$ at $\xi = 0$ and $\xi = +1$, we know that L_1 is of the form $L_1 = c\xi(1 - \xi)$. The constant c is obtained from $L_1 = 1$ at $\xi = -1$ as $c = -(1/2)$. Thus, $L_1 = (\xi) = -\xi(1 - \xi)/2$. L_2 and L_3 can be obtained by using similar arguments. We have

$$\begin{aligned} L_1(\xi) &= -\frac{\xi(1 - \xi)}{2} \\ L_2(\xi) &= (1 + \xi)(1 - \xi) \\ L_3(\xi) &= \frac{\xi(1 + \xi)}{2} \end{aligned} \tag{8.53a}$$

Similarly, generic shape functions can be defined along the η-axis (Fig. 8.6c) as

$$\begin{aligned} L_1(\eta) &= -\frac{\eta(1 - \eta)}{2} \\ L_2(\eta) &= (1 + \eta)(1 - \eta) \\ L_3(\eta) &= \frac{\eta(1 + \eta)}{2} \end{aligned} \tag{8.53b}$$

Referring back to the master element in Fig. 8.6b, we observe that every node has the coordinates $\xi = -1, 0$, or $+1$ and $\eta = -1, 0$, or $+1$. Thus, the *product rule* that follows yields the shape functions $N_1, N_2, \ldots, N_9$, as

$$\begin{aligned} N_1 &= L_1(\xi)L_1(\eta) \quad & N_5 &= L_2(\xi)L_1(\eta) \quad & N_2 &= L_3(\xi)L_1(\eta) \\ N_8 &= L_1(\xi)L_2(\eta) \quad & N_9 &= L_2(\xi)L_2(\eta) \quad & N_6 &= L_3(\xi)L_2(\eta) \\ N_4 &= L_1(\xi)L_3(\eta) \quad & N_7 &= L_2(\xi)L_3(\eta) \quad & N_3 &= L_3(\xi)L_3(\eta) \end{aligned} \tag{8.54}$$

By the manner in which L_i are constructed, it can be readily verified that $N_i = 1$ at node i and $N_i = 0$ at other nodes, as desired.

As noted in the beginning of this section, the use of higher order terms in $\mathbf{N}$ leads to a higher order interpolation of the displacement field as given by $\mathbf{u} = \mathbf{Nq}$. In addition, since $x = \sum_i N_i x_i$ and $y = \sum_i N_i y_i$, it means that higher order terms can also be used to define geometry. Thus, the elements can have curved edges if desired. However, it is possible to define a *subparametric* element by using nine-node shape functions to interpolate displacement and using only four-node quadrilateral shape functions to define geometry.

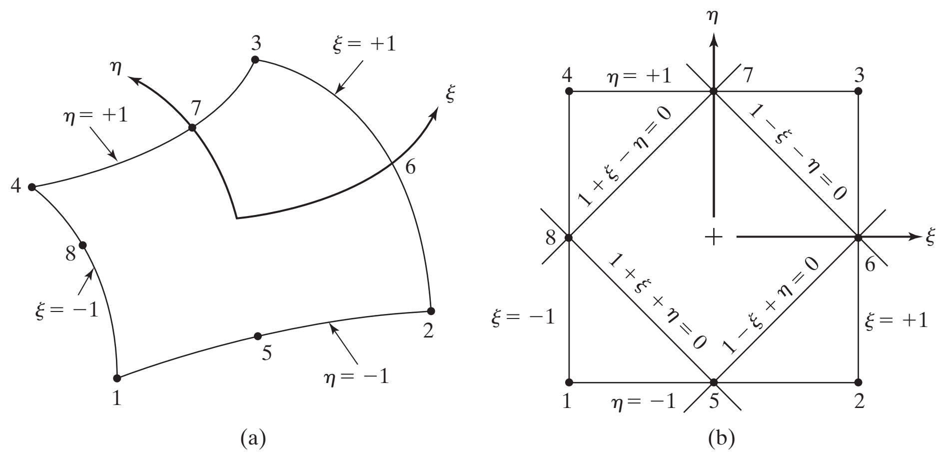


FIGURE 8.7 Eight-node quadrilateral in (a) x, y space and (b) ξ, η space.

Eight-Node Quadrilateral

This element belongs to the **serendipity** family of elements. The element consists of eight nodes (Fig. 8.7a), all of which are located on the boundary. Our task is to define shape functions N_i such that $N_i = 1$ at node i and 0 at all other nodes. In defining N_i, we refer to the master element shown in Fig. 8.7b. First, we define $N_1 - N_4$. For N_1, we note that $N_1 = 1$ at node 1 and 0 at other nodes. Thus, N_1 has to vanish along the lines $\xi = +1, \eta = +1$, and $\xi + \eta = -1$ (Fig. 8.7a). Consequently, N_1 is of the form

$$N_1 = c(1 - \xi)(1 - \eta)(1 + \xi + \eta) \tag{8.55}$$

At node 1, $N_1 = 1$ and $\xi = \eta = -1$. Thus, $c = -(1/4)$. We thus get

$$\begin{aligned}
N_1 &= -\frac{(1 - \xi)(1 - \eta)(1 + \xi + \eta)}{4} \\
N_2 &= -\frac{(1 + \xi)(1 - \eta)(1 - \xi + \eta)}{4} \\
N_3 &= -\frac{(1 + \xi)(1 + \eta)(1 - \xi - \eta)}{4} \\
N_4 &= -\frac{(1 - \xi)(1 + \eta)(1 + \xi - \eta)}{4}
\end{aligned} \tag{8.56}$$

Now, we define N_5, N_6, N_7, and N_8 at the midpoints. For N_5, we know that it vanishes along edges $\xi = +1, \eta = +1$, and $\xi = -1$. Consequently, it has to be of the form

$$N_5 = c(1 - \xi)(1 - \eta)(1 + \xi) \tag{8.57a}$$

$$= c(1 - \xi^2)(1 - \eta) \tag{8.57b}$$

The constant c in Eq. 8.57 is determined from the condition $N_5 = 1$ at node 5, or $N_5 = 1$ at $\xi = 0,\ \eta = -1$. Thus, $c = 1/2$ and

$$N_5 = \frac{(1-\xi^2)(1-\eta)}{2} \tag{8.57c}$$

We have

$$\begin{aligned}
N_5 &= \frac{(1-\xi^2)(1-\eta)}{2} \\
N_6 &= \frac{(1+\xi)(1-\eta^2)}{2} \\
N_7 &= \frac{(1-\xi^2)(1+\eta)}{2} \\
N_8 &= \frac{(1-\xi)(1-\eta^2)}{2}
\end{aligned} \tag{8.58}$$

Six-Node Triangle

The six-node triangle is shown in Figs. 8.8a and b. By referring to the master element in Fig. 8.8b, we can write the shape functions as

$$\begin{aligned}
N_1 &= \xi(2\xi - 1) \qquad & N_4 &= 4\xi\eta \\
N_2 &= \eta(2\eta - 1) \qquad & N_5 &= 4\zeta\eta \\
N_3 &= \zeta(2\zeta - 1) \qquad & N_6 &= 4\xi\zeta
\end{aligned} \tag{8.59}$$

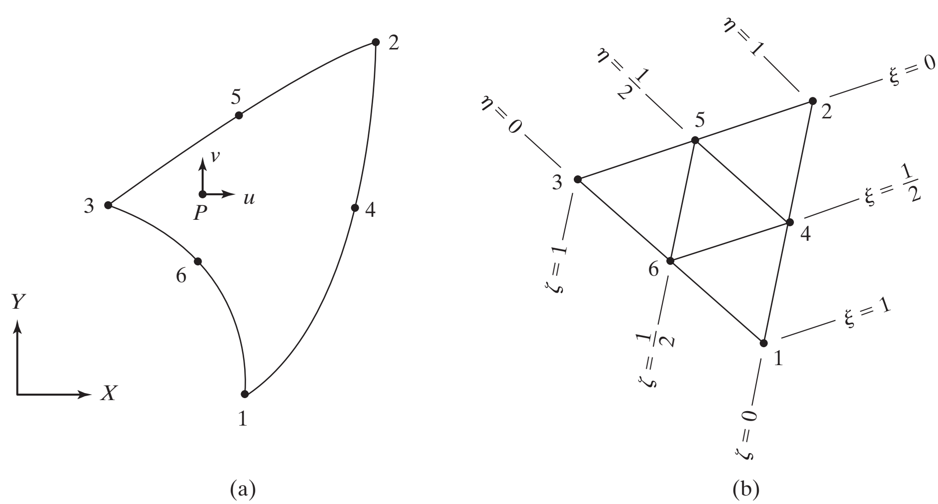


FIGURE 8.8 Six-node triangular element.

where $\zeta = 1 - \xi - \eta$. Because of terms ξ^2, η^2, etc., in the shape functions, this element is also called a *quadratic* triangle. The isoparametric representation is

$$\mathbf{u} = \mathbf{N}\mathbf{q}$$
$$x = \sum N_i x_i \qquad y = \sum N_i y_i \tag{8.60}$$

The element stiffness, which has to be integrated numerically, is given by

$$\mathbf{k}^e = t_e \int_A \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B} \det \mathbf{J}\, d\xi\, d\eta \tag{8.61}$$

Integration on a Triangle—Symmetric Points

The Gauss points for a triangular region differ from the square region considered earlier. The simplest is the one-point rule at the centroid with weight $w_1 = 1/2$ and $\xi_1 = \eta_1 = \zeta_1 = 1/3$. Equation 8.61 then yields

$$\mathbf{k}^e \approx \tfrac{1}{2} t_e \overline{\mathbf{B}}^{\mathrm{T}} \overline{\mathbf{D}}\ \overline{\mathbf{B}} \det \overline{\mathbf{J}} \tag{8.62}$$

where $\overline{\mathbf{B}}$ and $\overline{\mathbf{J}}$ are evaluated at the Gauss point. Other choices of weights and Gauss points are given in Table 8.2. The Gauss points given in Table 8.2 are arranged symmetrically within the triangle. Because of triangular symmetry, the Gauss points occur in groups or *multiplicity* of one, three, or six. For multiplicity of three, if ξ-, η-, and ζ-coordinates of a Gauss point are, for example, [(2/3),(1/6),(1/6)], then the other two Gauss points are located at [(1/6),(2/3),(1/6)], and [(1/6),(1/6),(2/3)]. Note that $\zeta = 1 - \xi - \eta$, as discussed in Chapter 6. For multiplicity of six, (ξ, η, ζ) in the table implies points given by all six permutations (ξ, η, ζ), (ξ, ζ, η), (η, ξ, ζ), (η, ζ, ξ), (ζ, ξ, η), and (ζ, η, ξ). When two coordinates are equal, it represents three points, and when all coordinates are equal it is one point (the centroid).

TABLE 8.2 Gauss Quadrature Formulas for a Triangle

$$\int_0^1 \int_0^{1-\xi} f(\xi,\eta)\, d\eta\, d\xi \approx \sum_{i=1}^{n} w_i f(\xi_i, \eta_i)$$

No. of Points, n	Weight, w_i	Multiplicity	ξ_i	η_i	ζ_i
One	1/2	1	1/3	1/3	1/3
Three	1/6	3	2/3	1/6	1/6
Three	1/6	3	1/2	1/2	0
Four	–(9/32)	1	1/3	1/3	1/3
	25/96	3	3/5	1/5	1/5
Six	1/12	6	0.6590276223	0.2319333685	0.1090390090

Integration on a Triangle—Degenerate Quadrilateral

We consider here the degenerate quadrilateral shown in Fig. 8.5a. A triangle 1–2–3 is defined as a quadrilateral 1–2–3–1, with two merging nodes. Four-point integration follows naturally. We provide here the proof of the general polynomial integration formula (Eq. 6.46) given by

$$\int_0^1 \int_0^{1-\xi} \xi^a \eta^b (1 - \xi - \eta)^c \, d\xi \, d\eta = \frac{a!b!c!}{(a + b + c + 2)}$$

Unit square mapped on to a master triangle is shown in Fig. 8.9. The shape functions are given by $N_1 = (1 - u)(1 - v)$, $N_2 = u(1 - v)$, $N_3 = uv$, and $N_4 = (1 - u)v$. The mapping is given by defining the triangle as a quadrilateral as stated above.

Node j	ξ_j	η_j
1	0	0
2	1	0
3	0	1
4	0	1

Now

$$\xi = \sum_{j=1}^{4} N_j \, \xi_j = u(1 - v)$$

$$\eta = \sum_{j=1}^{4} N_j \, \eta_j = v$$

$$1 - \xi - \eta = 1 - u(1 - v) - v = (1 - u)(1 - v)$$

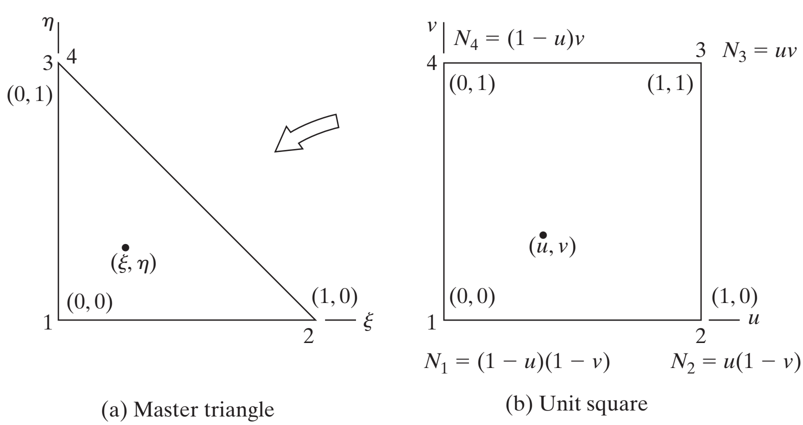


FIGURE 8.9 Triangle mapping.

The Jacobian of the transformation is

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \xi}{\partial u} & \dfrac{\partial \eta}{\partial u} \\ \dfrac{\partial \xi}{\partial v} & \dfrac{\partial \eta}{\partial v} \end{bmatrix} = \begin{bmatrix} 1 - v & 0 \\ -u & 1 \end{bmatrix}$$

$$\det \mathbf{J} = 1 - v.$$

$$I = \int_0^1 \int_0^{1-\xi} \xi^a \eta^b (1 - \xi - \eta)^c \, d\xi \, d\eta = \int_0^1 \int_0^1 \xi^a \eta^b (1 - \xi - \eta)^c \det \mathbf{J} \, du \, dv$$

Introducing ξ, η, and det $\mathbf{J}$ expressions from above, the integral becomes

$$I = \int_0^1 \int_0^1 u^a (1 - v)^a \, v^b (1 - u)^c \, (1 - v)^c \, (1 - v) \, du \, dv$$

$$= \int_0^1 u^a (1 - u)^c \, du \int_0^1 v^b (1 - v)^{a+c+1} \, dv$$

These are complete beta integrals, giving

$$I = \frac{\Gamma(a + 1)\Gamma(c + 1)}{\Gamma(a + c + 2)} \frac{\Gamma(b + 1)\Gamma(a + c + 2)}{\Gamma(a + b + c + 3)}$$

Canceling the common terms, and noting that $\Gamma(x + 1) = x!$ for integer x, we get the right-hand side of Eq. 6.46. Gamma function and the factorial are defined in Chapter 6 following Eq. 6.46.

This derivation shows that triangle defined as a degenerate quadrilateral is valid for a general polynomial expression. It then ensures integration of other functions to within computational error.

Comment on Midside Node. In the higher order isoparametric elements discussed previously, we note the presence of midside nodes. The midside node should be as near as possible to the center of the side. The node should not be outside of $(1/4) < s/\ell < (3/4)$, as shown in Fig. 8.10. This condition ensures that det $\mathbf{J}$ does not attain a value of zero in the element.

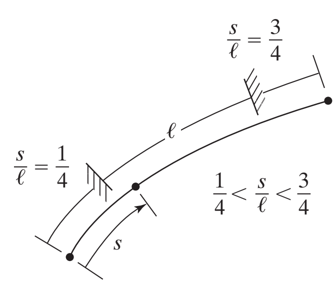


FIGURE 8.10 Restrictions on the location of a midside node.

Comment on Temperature Effect. Using the temperature strain defined in Eqs. 6.63 and 6.64 and following the derivation in Chapter 6, the nodal temperature load can be evaluated as

$$\boldsymbol{\theta}^e = t_e \int_A \int \mathbf{B}^{\mathrm{T}} \mathbf{D} \boldsymbol{\epsilon}_0 \, dA = t_e \int_{-1}^{1} \int_{-1}^{1} \mathbf{B}^{\mathrm{T}} \mathbf{D} \boldsymbol{\epsilon}_0 |\det \mathbf{J}| \, d\xi \, d\eta \tag{8.63}$$

This integral is performed using numerical integration.

8.5 FOUR-NODE QUADRILATERAL FOR AXISYMMETRIC PROBLEMS

The stiffness development for the four-node quadrilateral for axisymmetric problems follows steps similar to the quadrilateral element presented earlier. The x-, y-coordinates are replaced by r, z. The main difference occurs in the development of the $\mathbf{B}$ matrix, which relates the four strains to element nodal displacements. We partition the strain vector as

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_r \\ \epsilon_z \\ \gamma_{rz} \\ \epsilon_\theta \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{\epsilon}} \\ \epsilon_\theta \end{bmatrix} \tag{8.64}$$

where $\overline{\boldsymbol{\epsilon}} = [\epsilon_r \epsilon_z \gamma_{rz}]^{\mathrm{T}}$.

Now in the relation $\boldsymbol{\epsilon} = \mathbf{Bq}$, we partition $\mathbf{B}$ as $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$ such that $\mathbf{B}_1$ is a 3×8 matrix relating $\overline{\boldsymbol{\epsilon}}$ and $\mathbf{q}$ by

$$\overline{\boldsymbol{\epsilon}} = \mathbf{B}_1 \mathbf{q} \tag{8.65}$$

and $\mathbf{B}_2$ is a row vector 1×8 relating ϵ_θ and $\mathbf{q}$ by

$$\epsilon_\theta = \mathbf{B}_2 \mathbf{q} \tag{8.66}$$

Noting that r, z replace x, y, it is clear that $\mathbf{B}_1$ is same as the 3×8 matrix given in Eq. 8.24 for the four-node quadrilateral. Since $\epsilon_\theta = u/r$ and $u = N_1 q_1 + N_1 q + N_3 q_3 + N_4 q_4$, $\mathbf{B}_2$ can be written as

$$\mathbf{B}_2 = \begin{bmatrix} \dfrac{N_1}{r} & 0 & \dfrac{N_2}{r} & 0 & \dfrac{N_3}{r} & 0 & \dfrac{N_4}{r} & 0 \end{bmatrix} \tag{8.67}$$

On introducing these changes, the element stiffness is then obtained by performing numerical integration on

$$\mathbf{k}^e = 2\pi \int_{-1}^{1} \int_{-1}^{1} r \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \det \mathbf{J} \, d\xi \, d\eta \tag{8.68}$$

The force terms (in Eqs. 8.31 and 8.32) are to be multiplied by the factor of 2π as in the axisymmetric triangle.

The axisymmetric quadrilateral element has been implemented in program AXIQUAD.

8.6 CONJUGATE GRADIENT IMPLEMENTATION OF THE QUADRILATERAL ELEMENT

The ideas of the conjugate gradient method have been presented in Chapter 2. The equations are reproduced here using the notation for displacements, force, and stiffness:

$$\mathbf{g}_0 = \mathbf{K}\mathbf{Q}_0 - \mathbf{F}, \quad \mathbf{d}_0 = -\mathbf{g}_0$$

$$\begin{aligned}
\alpha_k &= \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k}{\mathbf{d}_k^{\mathrm{T}}\mathbf{K}\mathbf{d}_k} \\
\mathbf{Q}_{k+1} &= \mathbf{Q}_k + \alpha_k\mathbf{d}_k \\
\mathbf{g}_{k+1} &= \mathbf{g}_k + \alpha_k\mathbf{K}\mathbf{d}_k \\
\beta_k &= \frac{\mathbf{g}_{k+1}^{\mathrm{T}}\mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k} \\
\mathbf{d}_{k+1} &= -\mathbf{g}_{k+1} + \beta_k\mathbf{d}_k
\end{aligned} \tag{8.69}$$

Here, $k = 0, 1, 2, \ldots$. The iterations are carried out until $\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k$ reaches a small value.

We state here the steps in its implementation in finite element analysis (FEA). The main difference in this implementation is that the stiffness of each element is first generated and stored in a three-dimensional array. The stiffness of an element can be recalled from this array without recalculating for the iterations carried out. We start with the initial displacements at $\mathbf{Q}_0 = \mathbf{0}$. In the evaluation of $\mathbf{g}_0$, the force modifications for the boundary conditions are implemented. The term $\mathbf{K}\mathbf{d}_k$ is evaluated directly using element stiffness values by using $\sum_e \mathbf{k}^e\mathbf{d}_k^e$. The conjugate gradient approach is implemented in QUADCG.

8.7 CONCLUDING REMARKS AND CONVERGENCE

The concept of a master element defined in ξ-, η-coordinates, the definition of shape functions for interpolating displacement and geometry, and use of numerical integration are all key ingredients of the isoparametric formulation. A wide variety of elements can be formulated in a unified manner. Though only stress analysis has been considered in this chapter, the elements can be applied to nonstructural problems quite readily.

As discussed earlier, smaller-sized elements are required in regions of stress concentration, where derivatives of **u** with respect to x and/or y are high. In practice, the "stress jumps" between adjacent elements is a measure of correctness of the solution. A question that naturally arises is whether decreasing the element size h to zero gives the mathematically exact solution. The *patch test* presented in Chapter 6 is also applicable to higher order elements. Some basic criteria for h-convergence, which are implied in the patch test, are summarized below.

(1) *Admissibility*: Boundary conditions must prevent any rigid body motion of the structure; in two dimensions, x-, y-translations and in-plane rotation must not be allowed. Further, since the energy functional involves first derivatives, continuity of displacement (C^0 continuity) must exist. Specifically, the displacements must be continuous across the element boundaries. In a four-node quadrilateral element, the shape functions are linear on any edge. For example, along edge 1–2 in Fig. 8.2, we have $\eta = -1$ which makes N_i in Eq. 8.5 all linear in ξ, making u and v from Eq. 8.7 linear in ξ too. Thus, the displacement u along edge 1–2 will be in linear form as $u = a + b\xi$, which involves two coefficients that are uniquely determined from q_1 and q_3 and satisfies the interelement boundary compatibility criterion. Generalizing, a quadratic shape function along the edge will be continuous provided there are three nodes on the edge. On the other hand, a quadratic function along an edge will not be compatible if there are four nodes on the common boundary.

(2) *Completeness*: The shape functions must satisfy completeness requirements in the coordinate system used. Specifically, $u = a + bx$ is complete for one-dimensional problems. In two dimensions, the linear polynomial $u = a + bx + cy$ and the quadratic $u = a + bx + cy + dx^2 + exy + fy^2$ are complete. However, $u = a + bx + cy + exy$ is incomplete in a quadratic sense.

The reader should note that the above discussion pertaining to mesh discretization error and its reduction to zero, in theory, is independent of modeling errors which have to do with choice of boundary conditions and loads. For example, a structural support may be chosen as pinned, fixed, or attached to a spring. Experiments may be necessary to make a decision, and lower/upper bounds can be generated to guide the engineering analysis. In industrial projects, making entries in a laboratory notebook regarding modeling and other aspects is very important for scientific and legal reasons.

Example 8.3

The problem in Example 6.9 (Fig. E6.9) is now solved using four-node quadrilateral elements using program QUAD. The loads, boundary conditions, and node locations are the same as in Fig. E6.9. The only difference is the modeling with 24 quadrilateral elements, as against 48 CST elements in Fig. E6.9. Again, MESHGEN has been used to create the mesh (Fig. E8.3a), and a text editor has been used to define the loads, boundary conditions, and material properties.

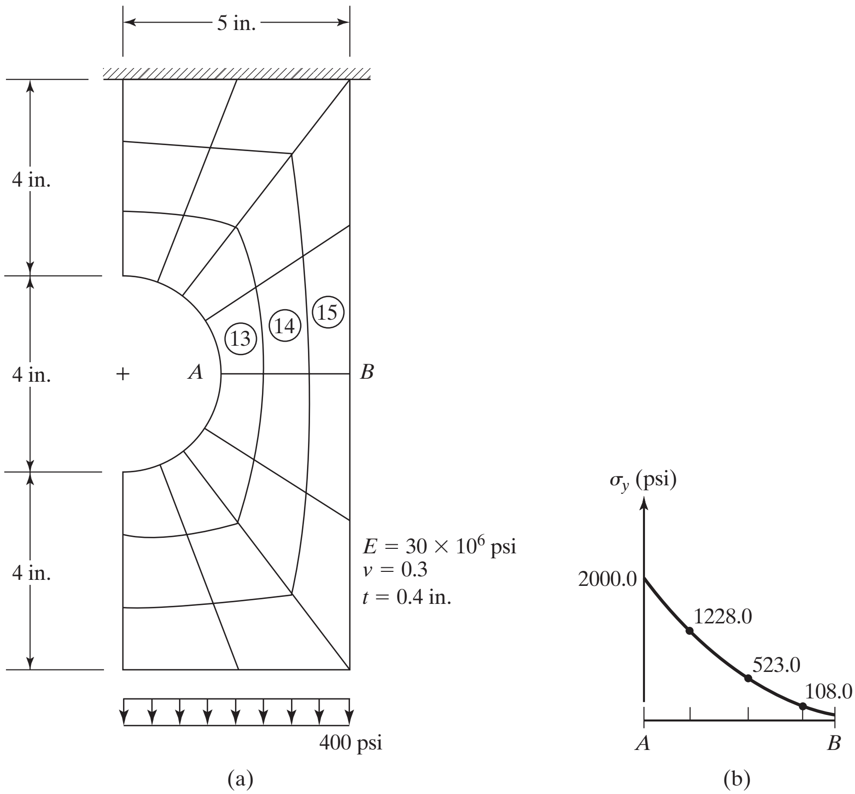


FIGURE E8.3

The stresses output by program QUAD correspond to the (0, 0) location in the natural coordinate system (master element). Using this fact, we *extrapolate* the y-stresses in elements 13, 14, and 15 to obtain the maximum y-stress near the semicircular edge of the plate, as shown in Fig. E8.3b. ■

REFERENCES FOR CONVERGENCE

1. Irons, B. M., "Engineering application of numerical integration in stiffness method." *AIAA Journal* 14: 2035–2037 (1966).
2. Oden, J. T., "A general theory of finite elements." *International Journal for Numerical Methods in Engineering* 1: 205–221 and 247–259 (1969).
3. Strang, W.G. and G. Fix, *An analysis of the finite element method,* Prentice Hall, Englewood Cliffs, NJ (1973).
4. Tong, P. and T. H. H. Pian, "On the convergence of a finite element method in solving linear elastic problems." *International Journal of Solids and Structures* 3: 865–879 (1967).

Input Data/Output

```
INPUT TO QUAD
<< 2D STRESS ANALYSIS USING QUAD >>
EXAMPLE 8.4
NN     NE     NM       NDIM     NEN      NDN
9      4      1        2        4        2
ND     NL     NMPC
6      1      0
Node#  X      Y
1      0      0
2      0      15
3      0      30
4      30     0
5      30     15
6      30     30
7      60     0
8      60     15
9      60     30
Elem#  N1     N2       N3       N4       Material#     Thickness     TempRise
1      1      4        5        2        1             10            0
2      2      5        6        3        1             10            0
3      4      7        8        5        1             10            0
4      5      8        9        6        1             10            0
DOF#   Displ.
1      0
2      0
3      0
4      0
5      0
6      0
DOF#   Load
18     -10000
MAT#   E      Nu       Alpha
1      7.00E+04        0.33     1.20E-05
B1     i      B2       j        B3       (Multi-point constr. B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM QUAD
Program Quad - Plane Stress Analysis
EXAMPLE 8.4
Node#  X-Displ         Y-Displ
1      -8.89837E-07    -2.83351E-07
2      1.77363E-08     1.50706E-07
3      8.72101E-07     -3.07839E-07
4      -0.088095167    -0.131050922
5      -0.001282636    -0.123052776
6      0.087963341     -0.126964356
7      -0.116924591    -0.365192248
8      0.000352218     -0.370143531
9      0.125124584     -0.386856887
```

```
Elem# Iteg1 Iteg2 Iteg3 Iteg4 <== vonMises Stresses
1      213.3629   160.2804    53.7790    141.1354
2      136.9611    48.5291   159.9454    208.3194
3      93.7355     58.8159    38.02357    91.4752
4      92.3071     69.3212    94.1831    120.1013
```

INPUT TO AXIQUAD

```
<< AXISYMMETRIC STRESS ANALYSIS USING AXIQUAD ELEMENT >>
EXAMPLE 6.4
NN     NE     NM      NDIM    NEN     NDN
6      2      1       2       4       2
ND     NL     NMPC
3      6      0
Node#  X      Y
1      3      0
2      3      0.5
3      7.5    0
4      7.5    0.5
5      12     0
6      12     0.5
Elem#  N1     N2      N3      N4      Material#      TempRise
1      1      3       4       2       1       0
2      3      5       6       4       1       0
DOF#   Displ.
2      0
6      0
10     0
DOF#   Load
1      3449
3      9580
5      23380
7      38711
9      32580
11     18780
MAT#   E              Nu      Alpha
1      3.00E+07       0.3     1.20E-05
B1     i      B2      j       B3   (Multi-point constr. B1*Qi+B2*Qj=B3)
```

OUTPUT FROM AXIQUAD

```
Program AxiQuad - Stress Analysis
EXAMPLE 6.4
Node#  R-Displ         Z-Displ
1      0.000829703     9.02758E-12
2      0.000828915     -5.42961E-05
3      0.000885462     -1.43252E-11
4      0.000887988     -2.52897E-05
5      0.000903563     5.29761E-12
6      0.000898857     -1.61269E-05
Elem#   Iteg1          Iteg2      Iteg3      Iteg4 <== vonMises Stresses
1       6271.290       3959.454   3964.174   6270.092
2       3094.231       2391.461   2390.800   3100.338
```

PROBLEMS

8.1. Figure P8.1 shows a four-node quadrilateral. The (x, y) coordinates of each node are given in the figure. The element displacement vector $\mathbf{q}$ is given as

$$\mathbf{q} = [1, \quad 0, \quad 0.15, \quad 0, \quad 0.2, \quad 0.35, \quad 0, \quad 0.08]^T$$

Find the following:

(a) The x-, y-coordinates of a point P whose location in the master element is given by $\xi = 1$ and $\eta = 1$

(b) The u, v displacements of the point P.

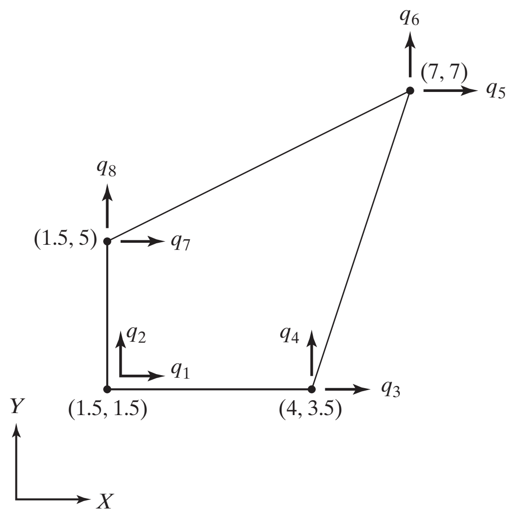


FIGURE P8.1

8.2. Using a 2 × 2 rule, evaluate the integral

$$\iint_A (x^2y + y^2)\, dx\, dy$$

by Gaussian quadrature, where A denotes the region shown in Fig. P8.1.

8.3. State whether the following statements are true or false:

(a) The shape functions are linear along an edge of a four-node quadrilateral element.

(b) For isoparametric elements, such as four-, eight-, and nine-node quadrilaterals, the point $\xi = 0, \eta = 0$ in the master element corresponds to the centroid of the element in x- and y-coordinates.

(c) The maximum stresses within an element occur at the Gauss points.

(d) The integral of a cubic polynomial can be performed exactly using two-point Gauss quadrature.

8.4. Solve Problem P6.17 with four-node quadrilaterals. Use program QUAD. (*Note*: Example input data set is given for 2 × 2 mesh division.)

8.5. A half-symmetry model of a culvert is shown in Fig. P8.5. The pavement load is a uniformly distributed load of 5000 N/m^2. Using program MESHGEN (discussed in Chapter 12), develop a finite element mesh with four-node quadrilateral elements. Using program QUAD determine the location and magnitude of maximum principal stress. First, try a mesh with about six elements and then compare results using about 18 elements.

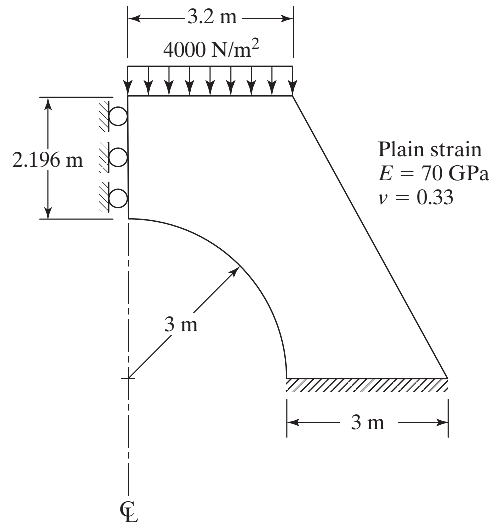


FIGURE P8.5

8.6. Solve Problem P6.18 using four-node quadrilateral elements (program QUAD). Compare the results with the solution obtained with CST elements. Use comparable-size meshes.

8.7. Solve Problem P6.19 using four-node quadrilaterals (program QUAD).

8.8. Solve Problem P6.22 using four-node quadrilaterals (program QUAD).

8.9. Program AXIQUAD is for axisymmetric stress analysis with four-node quadrilateral elements. Use that program to solve Example 7.2. Compare the results. [*Hint:* The first three rows of the **B** matrix are the same as for the plane stress problem in Eq. 8.25, and the last row can be obtained from $\epsilon_\theta = u/r$.]

8.10. This problem focuses on a concept used in program MESHGEN discussed in Chapter 12. An eight-node region is shown in Fig. P8.10a. The corresponding master element or *block* is shown in Fig. P8.10b. The block is divided into a grid of $3 \times 3 = 9$ smaller blocks of equal size, as indicated by dotted lines. Determine the corresponding x- and y-coordinates of all the 16 nodal points, and plot the nine subregions in Fig. P8.10a. Use the shape functions given in Eqs. 8.56 and 8.58.

8.11. Develop a computer program for the eight-node quadrilateral. Analyze the γ_{xy} cantilever beam shown in Fig. P8.11 with three finite elements. Compare results of x stress and centerline deflections with

(a) The six-element CST model

(b) Elementary beam theory.

8.12. Solve Problem 7.16 using axisymmetric quadrilateral elements (program AXIQUAD).

8.13. Answer the questions in brief:

(a) What is meant by the term "higher order elements?" ("Higher" than what?)

(b) How many independent material properties exist for an isotropic material?

(c) If a two-dimensional element for plane stress/plane strain is shaped as a six-node hexagon, what are the dimensions of **k** and **B**?

(d) Comment on what is meant by convergence in FEA. Search internet sources to learn h-convergence and p-convergence.

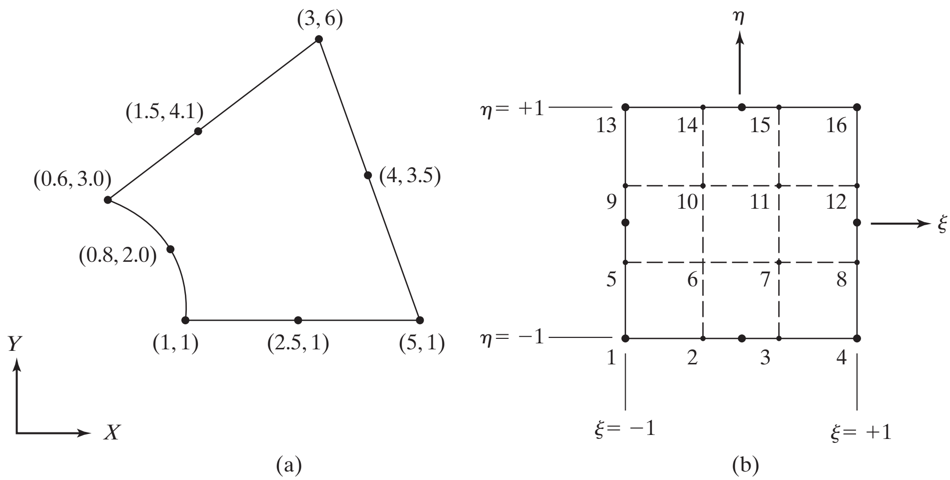


FIGURE P8.10

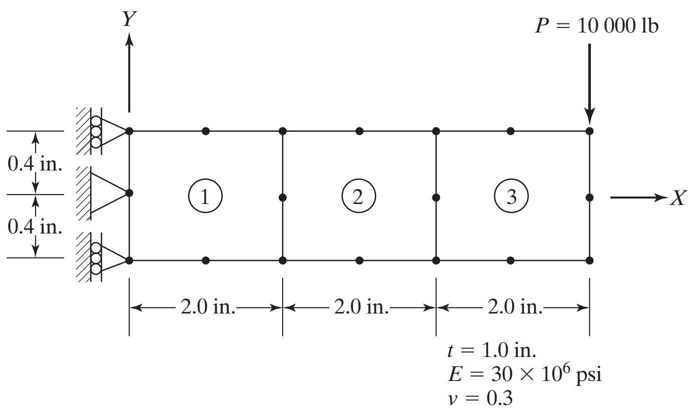


FIGURE P8.11

8.14. Indicate what is wrong with the model shown in Fig. P8.14 consisting of four quadrilateral elements. What do you suggest to set the model right?

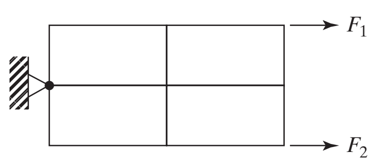


FIGURE P8.14

8.15. Answer the questions concisely and with justification:

(a) Are stresses constant within a four-node quadrilateral element?

(b) Are shape functions linear on the edge of a four-node quadrilateral element?

(c) Why do you need numerical integration?

(d) How many integration points are commonly used in computing matrices for the four-node quadrilateral element?

(e) A node k on a structure is connected by a *rigid link* to a fixed node j, as shown in Fig. P8.15. Assuming small deformations, write the corresponding boundary conditions (constraint equation) in the form of $\beta_1 Q_{kx} + \beta_2 Q_{ky} = \beta_0$. (*Hint*: Refer to discussion in Chapter 3.)

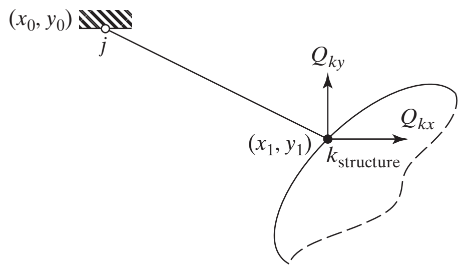


FIGURE P8.15

8.16. A four-node element shown in Fig. P8.16 undergoes deformation as shown in the figure. Determine expressions for ϵ_x, ϵ_y, and γ_{xy} in terms of x and y.

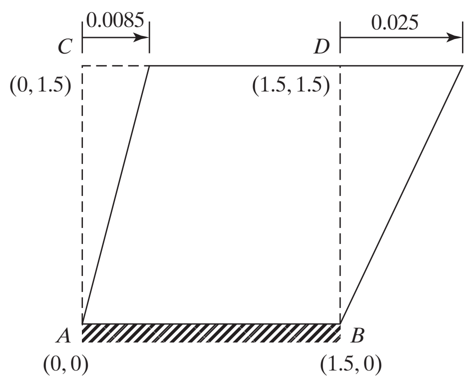


FIGURE P8.16

8.17. The potential energy of an element in terms of local displacements q_1 and q_2 is given by the expression

$$\Pi_e = 4q_1^2 + 8q_2^2 - 5q_1q_2 + 7q_1$$

Write down the expressions for element stiffness matrix **k** and element force **f**.

8.18. Answer the following questions concisely and with justification:

(a) What are the independent material constants for an *orthotropic* material in two dimensions?

(b) Comment on "modeling error" and "mesh-dependent error."

(c) The main advantage of an eight-node quadrilateral element over a four-node quadrilateral for plane elasticity problem is that the sides of the element can be curved. Comment.

(d) An element is subjected to hydrostatic stress, $\sigma_x = \sigma_y = \sigma_z = 0$, and all shear stresses are zero. What is the von Mises stress in the element?

(e) Does the structure shown in Fig. P8.18 exhibit rigid body motion?

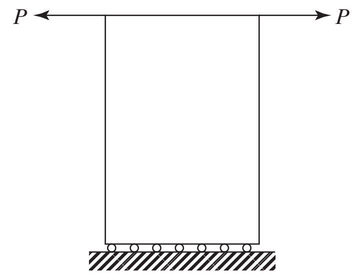


FIGURE P8.18

8.19. For the plane stress problem shown in Fig. P8.19, a coarse finite element mesh is shown. Give all the boundary conditions and loads for the mesh (similar to the BC and loads section of the data set).

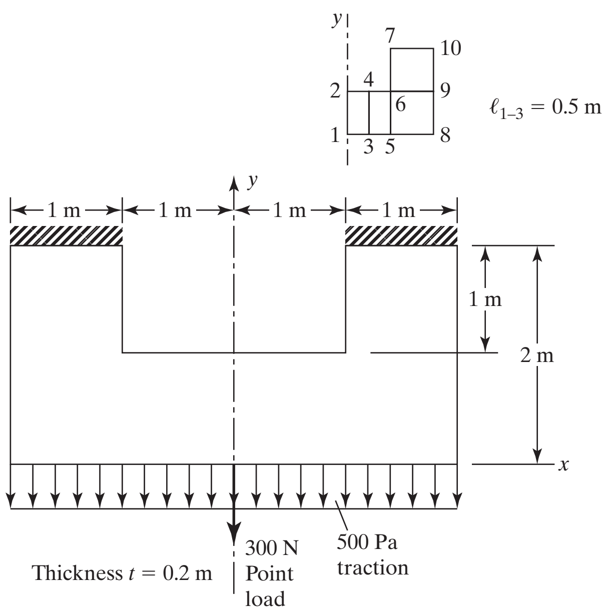


FIGURE P8.19

8.20. Which mesh, A or B, shown in Fig. P8.20, will give a better solution to the beam problem? Justify.

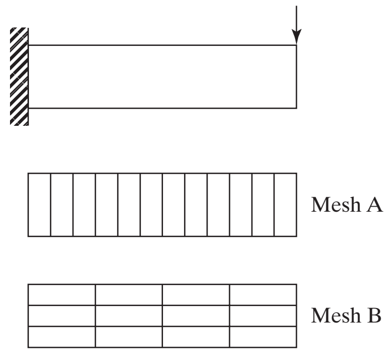
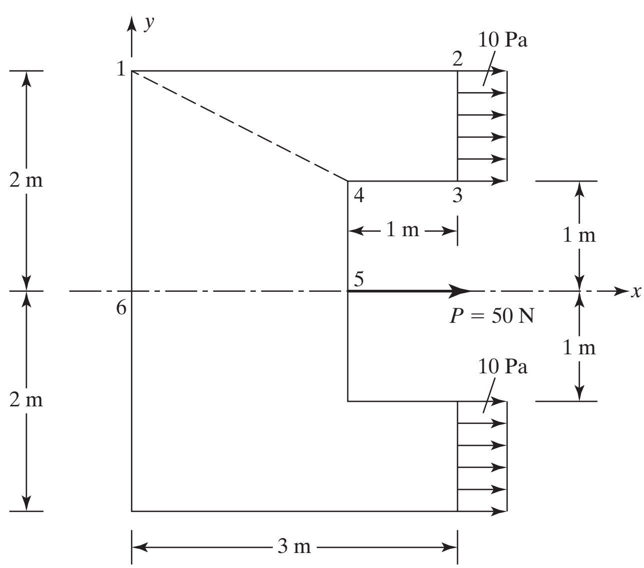


FIGURE P8.20

8.21. For the plane stress problem shown in Fig. P8.21, a simple model consisting of two elements 1-4-3-2 and 6-5-4-1 is shown. Loading consists of point load P along the axis of symmetry and symmetric-distributed load on the right edge. Give all the boundary conditions and equivalent point loads for the half-symmetry model using the dataset format for the computer program QUAD.



Thickness 1 m

FIGURE P8.21

8.22. For the plane stress problem shown in Fig. P8.22, divide the symmetric part into four rectangular elements. Number the nodes and elements, and give all the boundary conditions and component loads in the dataset format for the computer program QUAD.

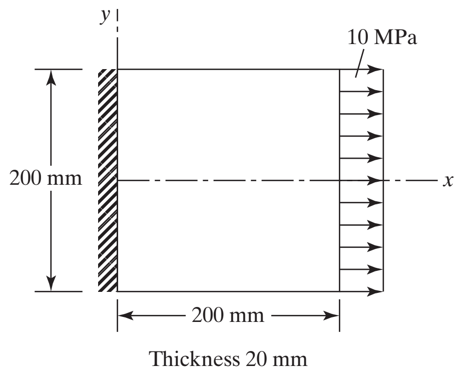


FIGURE P8.22

8.23. Consider a quadrilateral element whose nodes 1, 2, 3, and 4 in the counter clockwise (CCW) order are at coordinates (0, 0), (3, 0), (3, 1), and (0, 1), respectively. Thickness of the element t_e is 1. Evaluate the integral

$$I = \int_{\text{node 1}}^{\text{node 2}} N_1\, dS + \int_{\text{node 2}}^{\text{node 3}} N_1\, dS$$

using one-point Gauss–Legendre quadrature (i.e., one-point numerical integration). Note that $dS = t_e\, d\ell$.

8.24. Redo Example 8.3 with body force $\mathbf{f} = [f_x, f_y]^{\mathrm{T}}$ lb/in.^{3}.

8.25. Consider a four-node quadrilateral element whose edge 1–2 is loaded by triangularly varying traction as shown in Fig. P8.25. Derive the equivalent nodal forces acting at the nodes 1 and 2 using

(a) One-point numerical integration

(b) Two-point numerical integration.

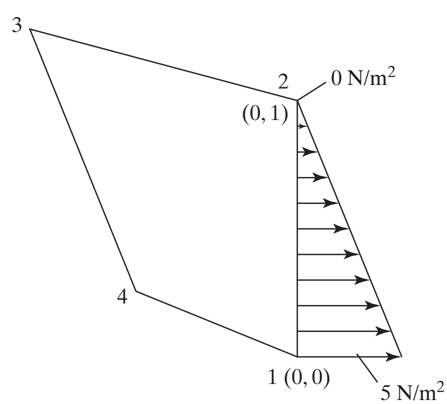


FIGURE P8.25

8.26. Evaluate $\int(\xi^2 + \xi\eta)\,dA$ over the CST (three-node triangular element) with node 1 at $(0, 0)$, node 2 at $(1, 0)$, node 3 at $(0, 1)$ using one-point and three-point integration. Compare your values with the exact integration using the triangle integration formula for polynomials.

8.27. Perform the load patch test for the plane stress problem shown in Fig. P8.27.

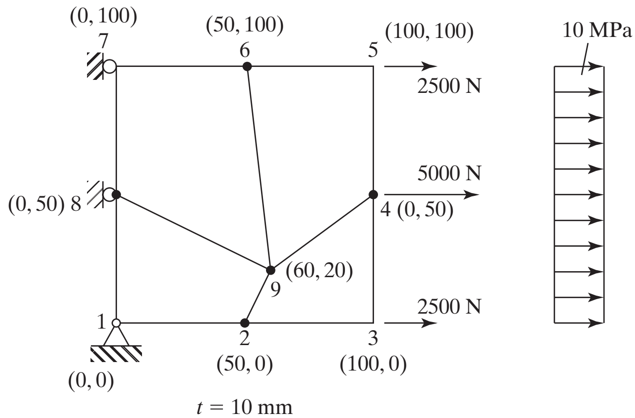


FIGURE P8.27

8.28. Perform the displacement patch test for the problem defined in Fig. 6.12a of Chapter 6 using degenerate quadrilateral elements 5-1-2-5, 5-2-3-5, 5-3-4-5, and 5-4-1-5.

8.29. Perform the load patch test for the problem defined in Fig. 6.12b of Chapter 6 using degenerate quadrilateral elements as defined in Problem 8.28.

8.30. The element in Fig. E8.2 is subjected to a body force $\mathbf{f} = [f_x \quad f_y]^T = [x \quad 0]^T$. Using thickness of the element as t, determine the equivalent point loads at the four nodes in the x directions using 2×2 Gauss–Legendre integration. Implement this in program QUAD or write a dedicated code in a computer language of your choice.

8.31. Figure P8.31 shows a traction load $T_x = 1\ \text{N/m}^2$, $T_y = 0$ applied on the edge of an eight-node quadrilateral. Determine the equivalent nodal forces. Use thickness of the element of 1 m. [*Hint:* On the edge 2-6-3, $\xi = 1$ ($d\xi = 0$), and $dS = \sqrt{(dx)^2 + (dy)^2} = \left[\sqrt{(dx/d\eta)^2 + (dx/d\eta)^2}\right] d\eta$. Use two-point numerical integration.]

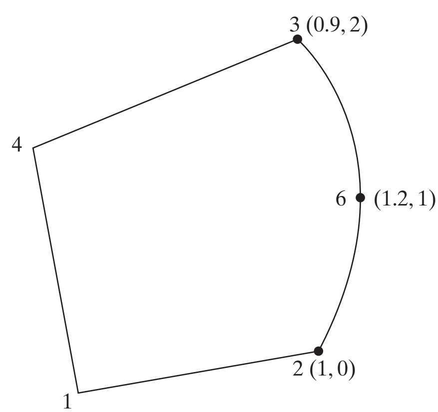


FIGURE P8.31

Program Listings

```
MAIN PROGRAM
'*************** PROGRAM QUAD **************
'*      2-D STRESS ANALYSIS USING 4-NODE    *
'* QUADRILATERAL ELEMENTS WITH TEMPERATURE *
'*     T.R.Chandrupatla and A.D.Belegundu   *
'*******************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call Stiffness
     Call ModifyForBC
     Call BandSolver
     Call StressCalc
     Call ReactionCalc
     Call Output
End Sub
```

```
GLOBAL STIFFNESS
Private Sub Stiffness()
     ReDim S(NQ, NBW)
     '----- Global Stiffness Matrix -----
     For N = 1 To NE
     Call IntegPoints
     Call DMatrix(N)
     Call ElemStiffness(N)
     '----- <<< Stiffness Assembly same as other programs >>>
End Sub
```

```
INTEGRATION POINTS
Private Sub IntegPoints()
'------- Integration Points XNI() --------
     C = 0.57735026919
     XNI(1, 1) = -C: XNI(1, 2) = -C
     XNI(2, 1) = C: XNI(2, 2) = -C
     XNI(3, 1) = C: XNI(3, 2) = C
     XNI(4, 1) = -C: XNI(4, 2) = C
End Sub
```

```
D MATRIX
Private Sub DMatrix(N)
'-----       D() Matrix     -----
     '--- Material Properties
     MATN = MAT(N)
     E = PM(MATN, 1): PNU = PM(MATN, 2)
     AL = PM(MATN, 3)
     '--- D() Matrix
```

```
      If LC = 1 Then
         '--- Plane Stress
         C1 = E / (1 - PNU ^ 2): C2 = C1 * PNU
      Else
         '--- Plane Strain
         C = E / ((1 + PNU) * (1 - 2 * PNU))
         C1 = C * (1 - PNU): C2 = C * PNU
      End If
      C3 = 0.5 * E / (1 + PNU)
      D(1, 1) = C1: D(1, 2) = C2: D(1, 3) = 0
      D(2, 1) = C2: D(2, 2) = C1: D(2, 3) = 0
      D(3, 1) = 0: D(3, 2) = 0: D(3, 3) = C3
End Sub
```

ELEMENT STIFFNESS

```
Private Sub ElemStiffness(N)
'--------    Element Stiffness and Temperature Load       -----
      For I = 1 To 8
      For J = 1 To 8: SE(I, J) = 0: Next J: TL(I) = 0: Next I
      DTE = DT(N)
      '--- Weight Factor is ONE
      '--- Loop on Integration Points
      For IP = 1 To 4
         '---Get DB Matrix at Integration Point IP
         Call DbMat(N, 1, IP)
         '--- Element Stiffness Matrix      SE
         For I = 1 To 8
            For J = 1 To 8
               C = 0
               For K = 1 To 3
                  C = C + B(K, I) * DB(K, J) * DJ * TH(N)
                  Next K
                  SE(I, J) = SE(I, J) + C
            Next J
         Next I
         '--- Determine Temperature Load TL
         AL = PM(MAT(N), 3)
         C = AL * DTE: If LC = 2 Then C = (1 + PNU) * C
         For I = 1 To 8
            TL(I) = TL(I) + TH(N) * DJ * C * (DB(1, I) + DB(2, I))
         Next I
      Next IP
End Sub
```

STRESS CALCULATIONS

```
Private Sub StressCalc()
      ReDim vonMisesStress(NE, 4), maxShearStress(NE, 4)
      '-----  Stress Calculations
      For N = 1 To NE
         Call DMatrix(N)
         For IP = 1 To 4
           '--- Get DB Matrix with Stress calculation
```

```
          '--- Von Mises Stress at Integration Point

          Call DbMat(N, 2, IP)
          C = 0: If LC = 2 Then C = PNU * (STR(1) + STR(2))
          C1 = (STR(1) - STR(2))^2 + (STR(2) - C)^2 + (C - STR(1))^2
          SV = Sqr(0.5 * C1 + 3 * STR(3) ^ 2)
          '--- Maximum Shear Stress R
          R = Sqr(0.25 * (STR(1) - STR(2))^2 + (STR(3))^2)
          maxShearStress(N, IP) = R
          vonMisesStress(N, IP) = SV
        Next IP
     Next N
End Sub
Private Sub DbMat(N, ISTR, IP)
```

```
'-------    DB()    MATRIX  ------
     XI = XNI(IP, 1): ETA = XNI(IP, 2)
     '--- Nodal Coordinates
     THICK = TH(N)
     N1 = NOC(N, 1): N2 = NOC(N, 2)
     N3 = NOC(N, 3): N4 = NOC(N, 4)
     X1 = X(N1, 1): Y1 = X(N1, 2)
     X2 = X(N2, 1): Y2 = X(N2, 2)
     X3 = X(N3, 1): Y3 = X(N3, 2)
     X4 = X(N4, 1): Y4 = X(N4, 2)
     '--- Formation of Jacobian   TJ
     TJ11 = ((1 - ETA) * (X2 - X1) + (1 + ETA) * (X3 - X4)) / 4
     TJ12 = ((1 - ETA) * (Y2 - Y1) + (1 + ETA) * (Y3 - Y4)) / 4
     TJ21 = ((1 - XI) * (X4 - X1) + (1 + XI) * (X3 - X2)) / 4
     TJ22 = ((1 - XI) * (Y4 - Y1) + (1 + XI) * (Y3 - Y2)) / 4
     '--- Determinant of the JACOBIAN
     DJ = TJ11 * TJ22 - TJ12 * TJ21
     '--- A(3,4) Matrix relates Strains to
     '--- Local Derivatives of u
     A(1, 1) = TJ22 / DJ: A(2, 1) = 0: A(3, 1) = -TJ21 / DJ
     A(1, 2) = -TJ12 / DJ: A(2, 2) = 0: A(3, 2) = TJ11 / DJ
     A(1, 3) = 0: A(2, 3) = -TJ21 / DJ: A(3, 3) = TJ22 / DJ
     A(1, 4) = 0: A(2, 4) = TJ11 / DJ: A(3, 4) = -TJ12 / DJ
     '--- G(4,8) Matrix relates Local Derivatives of u
     '--- to Local Nodal Displacements q(8)
     For I = 1 To 4: For J = 1 To 8
     G(I, J) = 0: Next J: Next I
     G(1, 1) = -(1 - ETA) / 4: G(2, 1) = -(1 - XI) / 4
     G(3, 2) = -(1 - ETA) / 4: G(4, 2) = -(1 - XI) / 4
     G(1, 3) = (1 - ETA) / 4: G(2, 3) = -(1 + XI) / 4
     G(3, 4) = (1 - ETA) / 4: G(4, 4) = -(1 + XI) / 4
     G(1, 5) = (1 + ETA) / 4: G(2, 5) = (1 + XI) / 4
     G(3, 6) = (1 + ETA) / 4: G(4, 6) = (1 + XI) / 4
     G(1, 7) = -(1 + ETA) / 4: G(2, 7) = (1 - XI) / 4
```

```
      G(3, 8) = -(1 + ETA) / 4: G(4, 8) = (1 - XI) / 4
      '--- B(3,8) Matrix Relates Strains to q
      For I = 1 To 3
         For J = 1 To 8
            C = 0
            For K = 1 To 4
               C = C + A(I, K) * G(K, J)
            Next K
            B(I, J) = C
         Next J
      Next I
      '--- DB(3,8) Matrix relates Stresses to q(8)
      For I = 1 To 3
         For J = 1 To 8
            C = 0
            For K = 1 To 3
               C = C + D(I, K) * B(K, J)
            Next K:
            DB(I, J) = C
         Next J
      Next I
      If ISTR = 2 Then
            '--- Stress Evaluation
            For I = 1 To NEN
               IIN = NDN * (NOC(N, I) - 1)
               II = NDN * (I - 1)
               For J = 1 To NDN
                  Q(II + J) = F(IIN + J)
               Next J
            Next I
            AL = PM(MAT(N), 3)
            C1 = AL * DT(N): If LC = 2 Then C1 = C1 * (1 + PNU)
            For I = 1 To 3
               C = 0
               For K = 1 To 8
                  C = C + DB(I, K) * Q(K)
               Next K
               STR(I) = C - C1 * (D(I, 1) + D(I, 2))
            Next I
      End If
End Sub
```

CHAPTER 9

Three-Dimensional Problems in Stress Analysis

9.1 INTRODUCTION

Most engineering problems are three dimensional. So far, we have studied the possibilities of finite element analysis of simplified models, where rod elements, constant-strain triangles, axisymmetric elements, beams, and so on give reasonable results. In this chapter, we deal with the formulation of three-dimensional stress-analysis problems. The four-node tetrahedral element is discussed in detail. Problem modeling and brick elements are also discussed. In addition, frontal solution method is introduced.

We recall from the formulation given in Chapter 1 that

$$\mathbf{u} = [u \quad v \quad w]^{\mathrm{T}} \tag{9.1}$$

where u, v, and w are displacements in the x, y, and z directions, respectively. The stresses and strains are given by

$$\boldsymbol{\sigma} = [\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{yz} \quad \tau_{xz} \quad \tau_{xy}]^{\mathrm{T}} \tag{9.2}$$

$$\boldsymbol{\epsilon} = [\epsilon_x \quad \epsilon_y \quad \epsilon_z \quad \gamma_{yz} \quad \gamma_{xz} \quad \gamma_{xy}]^{\mathrm{T}} \tag{9.3}$$

The stress–strain relations are given by

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} \tag{9.4}$$

where $\mathbf{D}$ is a (6×6) symmetric matrix. For isotropic materials, $\mathbf{D}$ is given by Eq. 1.15.

The strain–displacement relations are given by

$$\boldsymbol{\epsilon} = \left[\frac{\partial u}{\partial x} \quad \frac{\partial v}{\partial y} \quad \frac{\partial w}{\partial z} \quad \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \quad \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \quad \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right]^{\mathrm{T}} \tag{9.5}$$

The body force and traction vectors are given by

$$\mathbf{f} = [f_x \quad f_y \quad f_z]^{\mathrm{T}} \tag{9.6}$$

$$\mathbf{T} = [T_x \quad T_y \quad T_z]^{\mathrm{T}} \tag{9.7}$$

The total potential and the Galerkin/virtual work form for three dimensions are given in Chapter 1.

9.2 FINITE ELEMENT FORMULATION

We divide the volume into four-node tetrahedra. Each node is assigned a number, and the x-, y-, and z-coordinates are read in. A typical element e is shown in Fig. 9.1. The connectivity may be defined as shown in Table 9.1.

For each local node i, we assign three degrees of freedom (dof) q_{3i-2}, q_{3i-1}, and q_{3i}, and for the corresponding global node I, we assign Q_{3I-2}, Q_{3I-1}, and Q_{3I}. Thus, the element and global displacement vectors are

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad \ldots \quad q_{12}]^{\mathrm{T}} \tag{9.8}$$

$$\mathbf{Q} = [Q_1 \quad Q_2 \quad Q_3 \quad \ldots \quad Q_N]^{\mathrm{T}} \tag{9.9}$$

where N is the total number of dof for the structure, three per node. We define four Lagrange-type shape functions N_1, N_2, N_3, and N_4, where shape function N_i has a value of 1 at node i and zero at the other three nodes. Specifically, N_1 is 0 at nodes 2, 3, and 4 and linearly increases to 1 at node 1. Using the master element shown in Fig. 9.2, we can

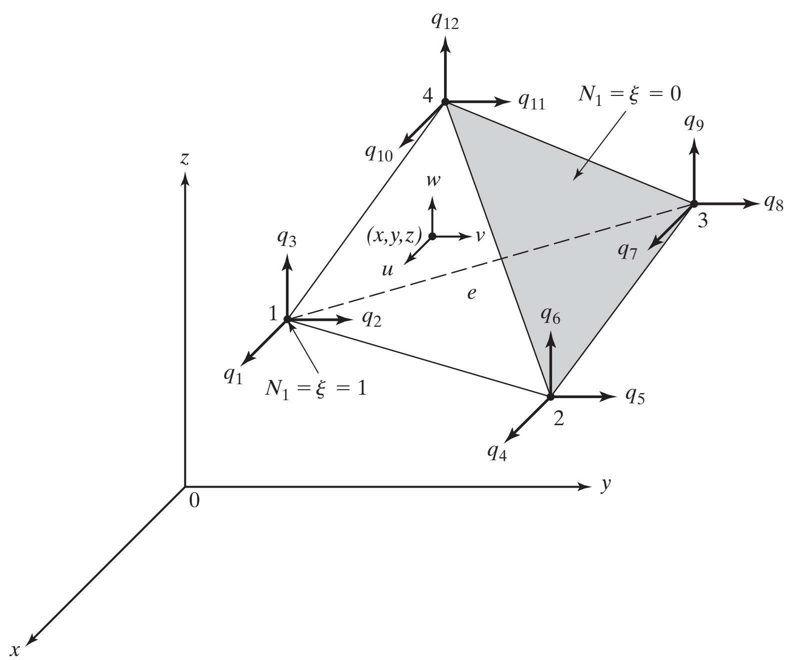


FIGURE 9.1 Tetrahedral element.

TABLE 9.1 Connectivity

	Nodes			
Element No.	1	2	3	4
e	I	J	K	L

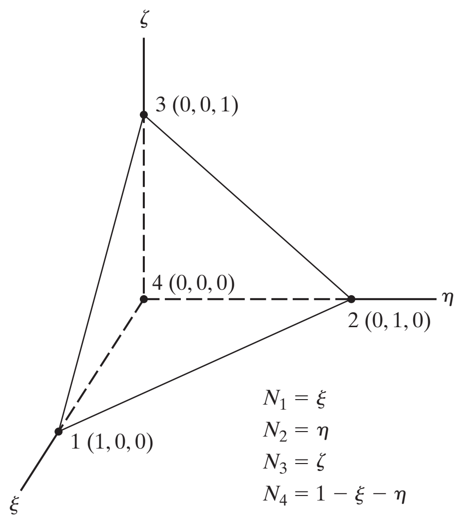


FIGURE 9.2 Master element for shape functions.

define the shape functions as

$$N_1 = \xi \quad N_2 = \eta \quad N_3 = \zeta \quad N_4 = 1 - \xi - \eta - \zeta \tag{9.10}$$

The displacements u, v, and w at $\mathbf{x}$ can be written in terms of the unknown nodal values as

$$\mathbf{u} = \mathbf{Nq} \tag{9.11}$$

where

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \tag{9.12}$$

It is easy to see that the shape functions given by Eq. 9.10 can be used to define the coordinates x, y, and z of the point at which the displacements u, v, and w are interpolated. The isoparametric transformation is given by

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 \\ y &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 \\ z &= N_1z_1 + N_2z_2 + N_3z_3 + N_4z_4 \end{aligned} \tag{9.13}$$

which, on substituting for N_i from Eq. 9.10 and using the notation $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$, and $z_{ij} = z_i - z_j$, yields

$$\begin{aligned} x &= x_4 + x_{14}\xi + x_{24}\eta + x_{34}\zeta \\ y &= y_4 + y_{14}\xi + y_{24}\eta + y_{34}\zeta \\ z &= z_4 + z_{14}\xi + z_{24}\eta + z_{34}\zeta \end{aligned} \tag{9.14}$$

Using the chain rule for partial derivatives, say, of u, we have

$$\begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial u}{\partial \zeta} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \\ \dfrac{\partial u}{\partial z} \end{Bmatrix} \tag{9.15}$$

Thus, the partial derivatives with respect to ξ, η, and ζ are related to x, y, and z derivatives by the foregoing relationship. The Jacobian transformation is given by

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} & \dfrac{\partial z}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial z}{\partial \eta} \\ \dfrac{\partial x}{\partial \zeta} & \dfrac{\partial y}{\partial \zeta} & \dfrac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} x_{14} & y_{14} & z_{14} \\ x_{24} & y_{24} & z_{24} \\ x_{34} & y_{34} & z_{34} \end{bmatrix} \tag{9.16}$$

We note here that

$$\det \mathbf{J} = x_{14}(y_{24}z_{34} - y_{34}z_{24}) + y_{14}(z_{24}x_{34} - z_{34}x_{24}) + z_{14}(x_{24}y_{34} - x_{34}y_{24}) \tag{9.17}$$

The volume of the element is given by

$$V_e = \left| \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} \det \mathbf{J}\, d\xi\, d\eta\, d\zeta \right| \tag{9.18}$$

Since $\det \mathbf{J}$ is constant,

$$V_e = |\det \mathbf{J}| \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} d\xi\, d\eta\, d\zeta \tag{9.19}$$

Using the polynomial integral formula

$$\int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} \xi^m \eta^n \zeta^p (1 - \xi - \eta - \xi)^q\, d\xi\, d\eta\, d\zeta = \frac{m!\, n!\, p!\, q!}{(m + n + p + q + 3)!} \tag{9.20}$$

we get

$$V_e = \tfrac{1}{6}|\det \mathbf{J}| \tag{9.21}$$

The inverse relation corresponding to Eq. 9.15 is given by

$$\begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \\ \dfrac{\partial u}{\partial z} \end{Bmatrix} = \mathbf{A} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial u}{\partial \zeta} \end{Bmatrix} \tag{9.22}$$

where **A** is the inverse of the Jacobian matrix **J** given in Eq. 9.16:

$$\mathbf{A} = \mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} y_{24}z_{34} - y_{34}z_{24} & y_{34}z_{14} - y_{14}z_{34} & y_{14}z_{24} - y_{24}z_{14} \\ z_{24}x_{34} - z_{34}x_{24} & z_{34}x_{14} - z_{14}x_{34} & z_{14}x_{24} - z_{24}x_{14} \\ x_{24}y_{34} - x_{34}y_{24} & x_{34}y_{14} - x_{14}y_{34} & x_{14}y_{24} - x_{24}y_{14} \end{bmatrix} \tag{9.23}$$

Using the strain–displacement relation in Eq. 9.5, the relation between derivatives in x, y, and z and ξ, η, and ζ in Eq. 9.22, and the assumed displacement field $\mathbf{u} = \mathbf{N}\mathbf{q}$ in Eq. 9.11, we get

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{q} \tag{9.24}$$

where **B** is a (6×12) matrix given by

$$\mathbf{B} = \begin{bmatrix} A_{11} & 0 & 0 & A_{12} & 0 & 0 & A_{13} & 0 & 0 & -\tilde{A}_1 & 0 & 0 \\ 0 & A_{21} & 0 & 0 & A_{22} & 0 & 0 & A_{23} & 0 & 0 & -\tilde{A}_2 & 0 \\ 0 & 0 & A_{31} & 0 & 0 & A_{32} & 0 & 0 & A_{33} & 0 & 0 & -\tilde{A}_3 \\ 0 & A_{31} & A_{21} & 0 & A_{32} & A_{22} & 0 & A_{33} & A_{23} & 0 & -\tilde{A}_3 & -\tilde{A}_2 \\ A_{31} & 0 & A_{11} & A_{32} & 0 & A_{12} & A_{33} & 0 & A_{13} & -\tilde{A}_3 & 0 & -\tilde{A}_1 \\ A_{21} & A_{11} & 0 & A_{22} & A_{12} & 0 & A_{23} & A_{13} & 0 & -\tilde{A}_2 & -\tilde{A}_1 & 0 \end{bmatrix} \tag{9.25}$$

where $\tilde{A}_1 = A_{11} + A_{12} + A_{13}$, $\tilde{A}_2 = A_{21} + A_{22} + A_{23}$, and $\tilde{A}_3 = A_{31} + A_{32} + \tilde{A}_{33}$. All the terms of **B** are constants. Thus, Eq. 9.24 gives constant strains after the nodal displacements are calculated.

Element Stiffness

The element strain energy in the total potential is given by

$$\begin{aligned} U_e &= \frac{1}{2}\int_e \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon}\, dV \\ &= \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\mathbf{q}\int_e dV \\ &= \frac{1}{2}\mathbf{q}^{\mathrm{T}}V_e\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\mathbf{q} \\ &= \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{k}^e\mathbf{q} \end{aligned} \tag{9.26}$$

where the element stiffness matrix $\mathbf{k}^e$ is given by

$$\mathbf{k}^e = V_e\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B} \tag{9.27}$$

in which V_e is the volume of the element given by $(1/6)|\det \mathbf{J}|$. In Galerkin approach, the internal virtual work of the element comes out to be

$$\int_e \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\epsilon}(\phi)\, dV = \boldsymbol{\psi}^{\mathrm{T}}V_e\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\mathbf{q} \tag{9.28}$$

which gives the element stiffness in Eq. 9.27.

Force Terms

The potential term associated with body force is

$$\begin{aligned}\int_e \mathbf{u}^{\mathrm{T}}\mathbf{f}\,dV &= \iiint \mathbf{N}^{\mathrm{T}}\mathbf{f}\det\mathbf{J}\,d\xi\,d\eta\,d\zeta \\ &= \mathbf{q}^{\mathrm{T}}\mathbf{f}^e\end{aligned} \tag{9.29}$$

Using the integration formula in Eq. 9.20, we get

$$\mathbf{f}^e = \frac{V_e}{4}[f_x, f_y, f_z, f_x, f_y, f_z, \ldots, f_z]^{\mathrm{T}} \tag{9.30}$$

For Eq. 9.30, the element body force vector $\mathbf{f}^e$ is of dimension 12×1. Note that $V_e f_x$ is the x component of the body force, which is distributed to the dof $q_1, q_4, q_7,$ and q_{10}.

Let us now consider uniformly distributed traction on the boundary surface. The boundary surface of a tetrahedron is a triangle. Without loss of generality, if A_e is the boundary surface on which traction is applied, formed by local nodes 1, 2, and 3, then

$$\int_{A_e} \mathbf{u}^{\mathrm{T}}\mathbf{T}\,dA = \mathbf{q}^{\mathrm{T}}\int_{A_e} \mathbf{N}^{\mathrm{T}}\mathbf{T}\,dA = \mathbf{q}^{\mathrm{T}}\mathbf{T}^e \tag{9.31}$$

The element traction load vector is given by

$$\mathbf{T}^e = \frac{A_e}{3}[T_x \quad T_y \quad T_z \quad T_x \quad T_y \quad T_z \quad T_x \quad T_y \quad T_z \quad 0 \quad 0 \quad 0] \tag{9.32}$$

The stiffnesses and forces are gathered into global locations using element connectivity. Point loads are added into proper locations of the force vector. Boundary conditions are considered using penalty or other approaches. The energy and Galerkin approaches yield the set of equations

$$\mathbf{KQ} = \mathbf{F} \tag{9.33}$$

9.3 STRESS CALCULATIONS

After these equations are solved, the element nodal displacements $\mathbf{q}$ can be obtained. Since $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon} = \mathbf{Bq}$, the element stresses are given by

$$\boldsymbol{\sigma} = \mathbf{DBq} \tag{9.34}$$

The three principal stresses can be calculated by using the relationships in Eq. 9.35. The three invariants of the (3×3) stress tensor are

$$\begin{aligned}I_1 &= \sigma_x + \sigma_y + \sigma_z \\ I_2 &= \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_x\sigma_z - \tau_{yz}^2 - \tau_{xz}^2 - \tau_{xy}^2 \\ I_3 &= \sigma_x\sigma_y\sigma_z + 2\tau_{yz}\tau_{xz}\tau_{xy} - \sigma_x\tau_{yz}^2 - \sigma_y\tau_{xz}^2 - \sigma_z\tau_{xy}^2\end{aligned} \tag{9.35}$$

We define

$$
\begin{aligned}
a &= \frac{I_1^2}{3} - I_2 \\
b &= -2\left(\frac{I_1}{3}\right)^3 + \frac{I_1 I_2}{3} - I_3 \\
c &= 2\sqrt{\frac{a}{3}} \\
\theta &= \frac{1}{3}\cos^{-1}\left(-\frac{3b}{ac}\right)
\end{aligned}
\tag{9.36}
$$

The principal stresses are given by

$$
\begin{aligned}
\sigma_1 &= \frac{I_1}{3} + c\cos\theta \\
\sigma_2 &= \frac{I_1}{3} + c\cos\left(\theta + \frac{2\pi}{3}\right) \\
\sigma_3 &= \frac{I_1}{3} + c\cos\left(\theta + \frac{4\pi}{3}\right)
\end{aligned}
\tag{9.37}
$$

The orientation $[\nu_{kx} \;\; \nu_{ky} \;\; \nu_{kz}]^{\mathrm{T}}$ of the principal stress σ_k is obtained by solving the set of equations

$$
\begin{bmatrix} \sigma_x - \sigma_k & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y - \sigma_k & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z - \sigma_k \end{bmatrix} \begin{bmatrix} \nu_{kx} \\ \nu_{ky} \\ \nu_{kz} \end{bmatrix} = 0 \tag{9.38}
$$

Note that the 3×3 matrix in Eq. 9.38 is singular. Thus, not all three equations are independent. If all three principal stresses are distinct, the first two equations may be solved for ν_{kx} and ν_{ky} by setting $\nu_{kz} = 1$ and then the vector $\mathbf{v}_k$ is normalized such that $\mathbf{v}_k^T\mathbf{v}_k = 1$. Care must be taken when two eigen values are equal.

9.4 MESH PREPARATION

While complex three-dimensional regions can be effectively filled by tetrahedral elements, similar to triangular elements filling a two-dimensional region, it is a tedious affair to carry out manual data preparation. To overcome this, for simple regions, it is easier to divide the regions into eight-node blocks. Consider the master cube shown in Fig. 9.3. The cube can be divided into five tetrahedra, as shown in Fig. 9.4, with the connectivity as given in Table 9.2.

In this division, the first four elements are of equal volume and element 5 has twice the volume of other elements. In this case, care must be taken to match element edges on adjacent blocks.

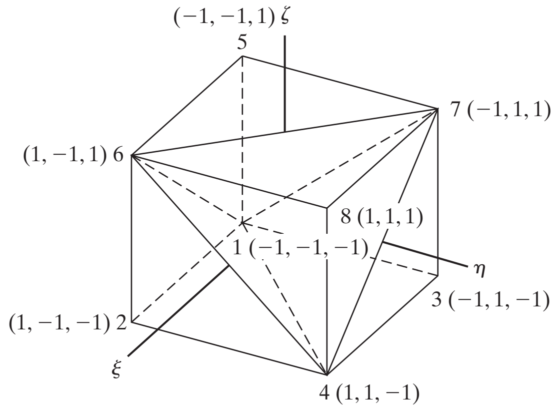


FIGURE 9.3 Cube for tetrahedral division.

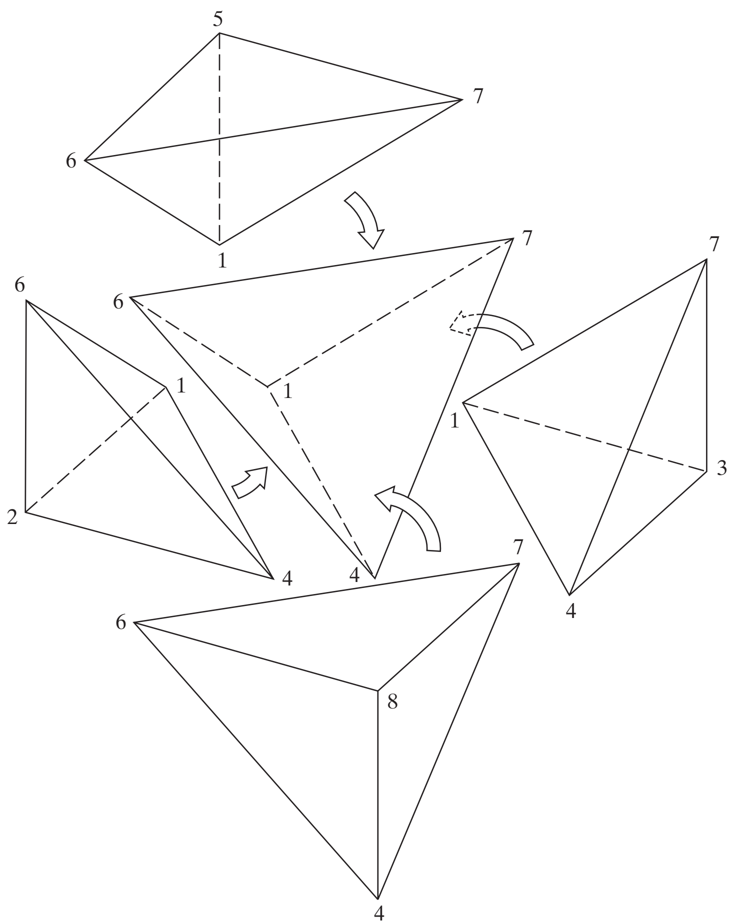


FIGURE 9.4 Division of a cube into five tetrahedra.

TABLE 9.2 Five Tetrahedra

	Nodes			
Element No.	1	2	3	4
1	1	4	2	6
2	1	4	3	7
3	6	7	5	1
4	6	7	8	4
5	1	4	6	7

The master cube can also be divided into six elements with equal volume. A typical division is given in Table 9.3. The element division of one-half of the cube is shown in Fig. 9.5. For the division shown in Table 9.3, the same division pattern repeats for adjacent elements.

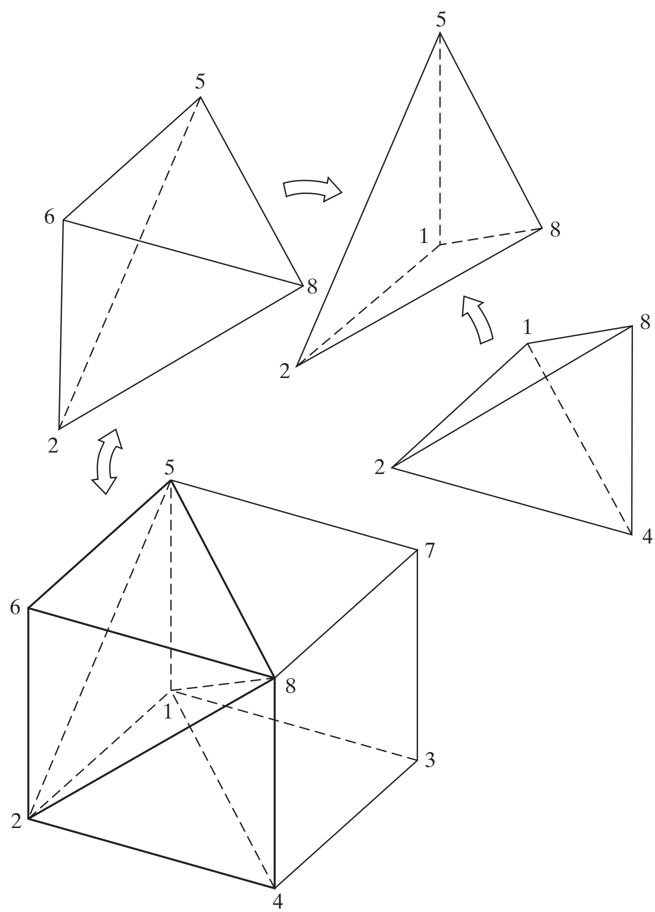


FIGURE 9.5 Division of a cube into six tetrahedra.

TABLE 9.3 Six Tetrahedra

	Nodes			
Element No.	1	2	3	4
1	1	2	4	8
2	1	2	8	5
3	2	8	5	6
4	1	3	4	7
5	1	7	8	5
6	1	8	4	7

Use of det **J** in the calculation of **B** in Eq. 9.24 and use of |det **J**| in the estimation of element volume V_e enables us to use element node numbers in any order. Among solid elements, this holds for four-node tetrahedra, since every node is connected to the other three. Some codes may still require consistent numbering schemes.

Program TETRA is included in the downloadable programs.

Example 9.1

Figure E9.1 shows a four-node tetrahedral object. The coordinate dimensions shown are in inches. The material used is steel with $E = 30 \times 10^6$ psi and $\nu = 0.3$. Nodes 2, 3, and 4 are fixed, and a 1000-lb load is applied at node 1 as shown. Determine the displacement of node 1 using a single element.

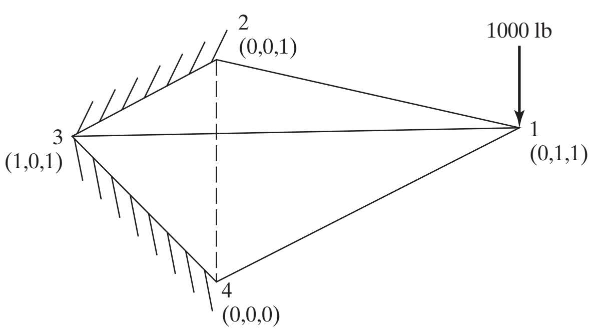


FIGURE E9.1

Solution The Jacobian **J** given in Eq. 9.16 is easily calculated using the nodal coordinates:

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

and det **J** = 1. The inverse of the Jacobian, **A**, is calculated using Eq. 9.23:

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Using the elements of $\mathbf{A}$, the strain–displacement matrix $\mathbf{B}$ can be evaluated using Eq. 9.25. In the product, $\mathbf{Bq}$, only the first three columns multiply the first three components of $\mathbf{q}$. The last six components of $\mathbf{q}$ are zero. When we use the strike-off approach in assembling the stiffness matrix $\mathbf{k} = V_e\mathbf{B}^{\mathrm{T}}\mathbf{DB}$, we need to deal with the first three columns of $\mathbf{B}$. Partitioning $\mathbf{B} = [\mathbf{B}_1\mathbf{B}_2]$ with $\mathbf{B}_1$ representing the first three columns, the modified 3×3 stiffness matrix $\mathbf{K}$ is given by $\mathbf{B}_1^{\mathrm{T}}\mathbf{DB}_1$. The volume of the element V_e is given by (1/6).

$\mathbf{B}_1$ is calculated using the first three columns of $\mathbf{B}$ defined in Eq. 9.25 as

$$\mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The stress–strain relation matrix $\mathbf{D}$ is evaluated using Eq. 1.15 from Chapter 1:

$$\mathbf{D} = 10^7 \begin{bmatrix} 4.038 & 1.731 & 1.731 & 0 & 0 & 0 \\ 1.731 & 4.038 & 1.731 & 0 & 0 & 0 \\ 1.731 & 1.731 & 4.038 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.154 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.154 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.154 \end{bmatrix}$$

The modified stiffness matrix is given by

$$\mathbf{K} = V_e\mathbf{B}_1^{\mathrm{T}}\mathbf{DB}_1 = 10^6 \begin{bmatrix} 1.923 & 0 & 0 \\ 0 & 6.731 & 0 \\ 0 & 0 & 1.923 \end{bmatrix}$$

The force vector is $\mathbf{F} = [0 \quad 0 \quad -1000]^{\mathrm{T}}$. Solving for $\mathbf{KQ} = \mathbf{F}$, we get

$$\mathbf{Q} = [0 \quad 0 \quad -.00052]^{\mathrm{T}}$$

We note that for this one-element case, the modified stiffness is a diagonal matrix for the geometry of the problem chosen. ■

9.5 HEXAHEDRAL ELEMENTS AND HIGHER ORDER ELEMENTS

In the hexahedral elements, a consistent node-numbering scheme must be followed for defining the connectivity. For an eight-node hexahedral or brick element, we consider the mapping onto a cube of 2-unit sides placed symmetrically with ξ-, η-, and ζ-coordinates as shown in Fig. 9.6. The corresponding element in two dimensions is the four-node quadrilateral discussed in Chapter 8.

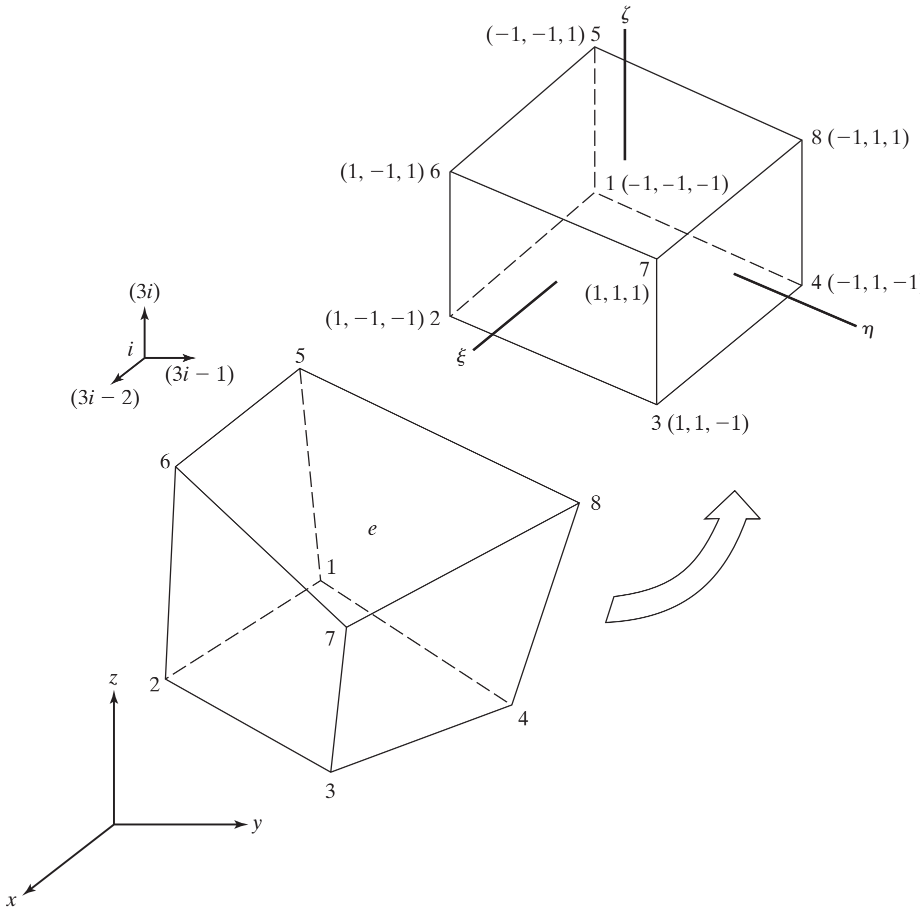


FIGURE 9.6 Hexahedral element.

On the master cube, the Lagrange shape functions can be written as

$$N_i = \tfrac{1}{8}(1 + \xi_i\xi)(1 + \eta_i\eta)(1 + \zeta_i\zeta) \quad i = 1 \text{ to } 8 \tag{9.39}$$

where (ξ_i, η_i, ζ_i) represents the coordinates of node i of the element in the (ξ, η, ζ) system. The element nodal displacements are represented by the vector

$$\mathbf{q} = [q_1 \quad q_2 \quad \cdots \quad q_{24}]^T \tag{9.40}$$

We use the shape functions N_i to define displacements at any point inside the element in terms of its nodal values:

$$\begin{aligned} u &= N_1q_1 + N_2q_4 + \cdots + N_8q_{22} \\ v &= N_1q_2 + N_2q_5 + \cdots + N_8q_{23} \\ w &= N_1q_3 + N_2q_6 + \cdots + N_8q_{24} \end{aligned} \tag{9.41}$$

Also,

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + \cdots + N_8x_8 \\ y &= N_1y_1 + N_2y_2 + \cdots + N_8y_8 \\ z &= N_1z_1 + N_2z_2 + \cdots + N_8z_8 \end{aligned} \tag{9.42}$$

Following the steps used in the development of the quadrilateral element in Chapter 8, we can get the strains in the form

$$\boldsymbol{\epsilon} = \mathbf{Bq} \tag{9.43}$$

The element stiffness matrix is given by

$$\mathbf{k}^e = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}^{\mathrm{T}}\mathbf{DB}\left|\det \mathbf{J}\right| d\xi\, d\eta\, d\zeta \tag{9.44}$$

where we have used $dV = \left|\det \mathbf{J}\right| d\xi\, d\eta\, d\zeta$ and $\mathbf{J}$ is the (3×3) Jacobian matrix. The integration in Eq. 9.44 is performed numerically using Gauss quadrature.

Higher order elements, for example, 10-node tetrahedral elements or 20-node or 27-node hexahedral elements, can be developed using the ideas discussed in Chapter 8. Temperature effect is treated in a very similar manner, as in the case of the quadrilateral in Chapter 8. Program HEXAFRON is included in the downloadable program set.

9.6 PROBLEM MODELING

In solving a problem, the first step is to start with a coarse model. The data needed will be nodal coordinates, element nodal connectivity, material properties, constraint conditions, and nodal loads. In the three-dimensional cantilever shown in Fig. 9.7, the geometry and loading conditions demand a three-dimensional model. Element and connectivities can easily be established by defining the four 8-cornered blocks. We can model the first block, near the base of the cantilever, as a hexahedral element with connectivity 2-1-5-6-3-4-8-7. For each subsequent block, the connectivity can be generated by increasing each number in the present set by four. Coordinates of nodes can be generated using the shape functions of Eq. 9.38 for geometry definition. These aspects will be discussed in Chapter 12. Alternatively, each block in the three-dimensional cantilever can be modeled using tetrahedral elements. For the repeating block pattern shown in Fig. 9.6, the six-element division given in Table 9.3 may be used.

The consideration of boundary conditions follows those presented for one- and two-dimensional problems. However, to give a general idea of constraints and their consideration in finite element analysis, we refer to Fig. 9.8. A point fully restrained is a point constraint. This is considered by adding a large stiffness C to the diagonal locations corresponding to dof of node I. When the node is constrained to move along a line, say, $\mathbf{t}$, with direction cosines (ℓ, m, n), the penalty term comes from setting $\mathbf{u} \times \mathbf{t} = \mathbf{0}$. This results in the addition of following stiffness terms when the node is constrained along a line:

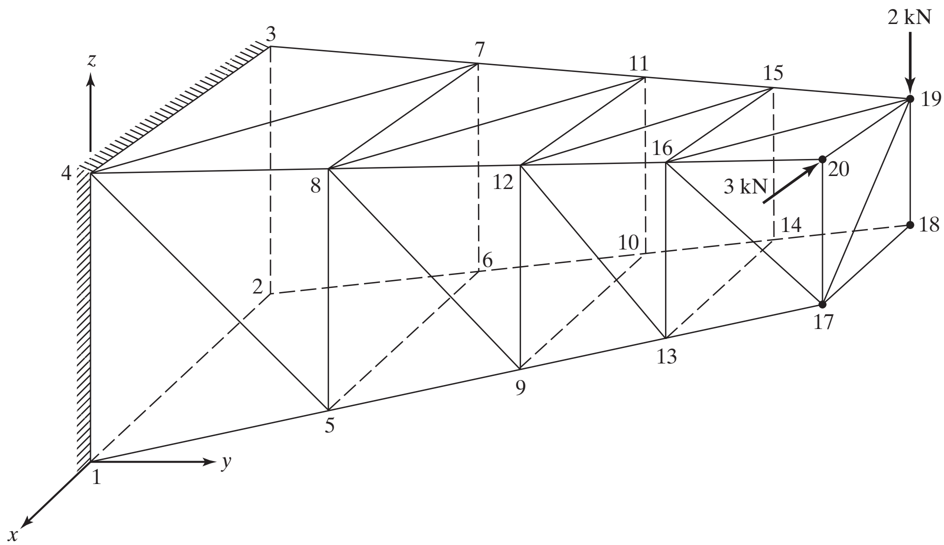


FIGURE 9.7 Three-dimensional elastic body.

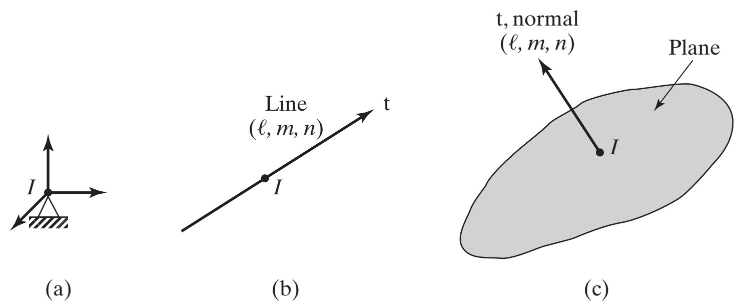


FIGURE 9.8 Nodal constraints: (a) point constraint, (b) line constraint, and (c) plane constraint.

$$
\begin{array}{c c}
 & \begin{array}{ccc} 3I-2 & \quad 3I-1 & \quad 3I \end{array} \\
\begin{array}{c} 3I-2 \\ 3I-1 \\ 3I \end{array} &
\begin{bmatrix} C(1-\ell^2) & -C\ell m & -C\ell n \\ & C(1-m^2) & -Cmn \\ \text{Symmetric} & & C(1-n^2) \end{bmatrix}
\end{array}
$$

When a node is forced to lie on a plane with normal direction $\mathbf{t}$, shown in Fig. 9.8c, the penalty terms come from $\mathbf{u} \cdot \mathbf{t} = 0$. This requires that the following terms be added to the stiffness matrix:

$$
\begin{array}{c c}
 & \begin{array}{ccc} 3I-2 & \quad 3I-1 & \quad 3I \end{array} \\
\begin{array}{c} 3I-2 \\ 3I-1 \\ 3I \end{array} &
\begin{bmatrix} C\ell^2 & C\ell m & C\ell n \\ & Cm^2 & Cmn \\ \text{Symmetric} & & Cn^2 \end{bmatrix}
\end{array}
$$

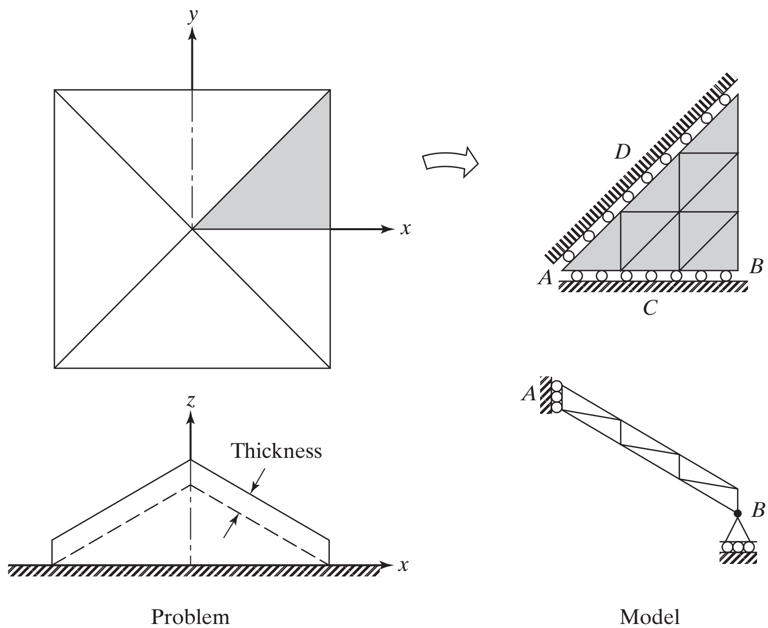


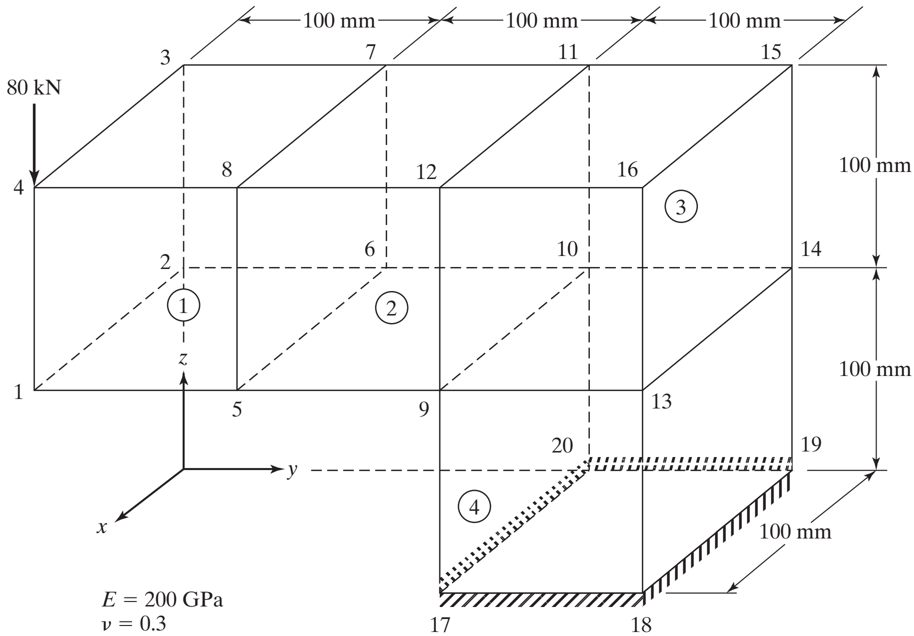
FIGURE 9.9 Metal part with a pyramid surface.

Figure 9.9 shows a pyramid-shaped metal part and its finite element model. We observe here that nodes at A and B are line-constrained and nodes along C and D are plane-constrained. This discussion helps one to handle the modeling of three-dimensional problems with relative ease.

9.7 FRONTAL METHOD FOR FINITE ELEMENT MATRICES

In three-dimensional problems, the size of the stiffness matrix increases rapidly even with the banded method of handling. An alternative direct method that results in considerable saving in the use of computer memory is called the *frontal method*. In this method, the order of element numbering plays a more important role than the order of node numbering. The frontal method relies on the fact that a dof can be eliminated as soon as all stiffness values in rows and columns for that dof are complete. Irons[1] observed that all of the dof for a node can be eliminated when it appears for the last time as we assemble the elements in the ascending order. In the example illustrated in Fig. 9.10, nodes 1, 2, 3, and 4 appear for the last time in element 1. The dof corresponding to all these nodes can be eliminated as soon as element 1 is assembled. Once a dof is eliminated, the corresponding equation is no more necessary until backsubstitution. This equation

[1]Bruce M. Irons, "A frontal solution program for finite element analysis," *International Journal for Numerical Methods in Engineering* 2: 5–32 (1970).



Element No.	Nodes								Front Size	
	1	2	3	4	5	6	7	8	Assembled	Eliminated
1	−1	−2	−3	−4	5	6	7	8	8	4
2	−5	−6	−7	−8	9	10	11	12	8	4
3	9	10	−11	−12	13	14	−15	−16	8	4
4	−9	−10	−14	−13	−17	−20	−19	−18	8	0

Block size = 8 × dof per node

FIGURE 9.10 Example for frontal method.

can be written to an external device such as a tape or hard disk for backsubstitution in the reverse order. As we assemble an element, the active matrix size grows, and when some dof are eliminated, the matrix size shrinks. The active matrix size can be compared to the action of an accordion. The largest block size needed can be determined using a prefront routine that employs a modified element connectivity matrix.

Connectivity and Prefront Routine

The first step is to determine the last appearance of a node. This is simplified by looking for the first appearance of the node as we proceed in the descending order of elements. In the element where this occurs, a negative sign is added to the node number in the connectivity array. The node-number modification is shown in the table in Fig. 9.10 for the hexahedral element. After this operation is carried out for all the node numbers, we are ready to determine the block size. Let us first evaluate the front size in terms of the number of nodes. At the assembly of the first element, the nodal front size is 8. The dof corresponding to four nodes are now eliminated. The nodal front size shrinks to 4. The assembly of element 2 adds four new nodes; thus, the front grows to size 8. As seen from the table in Fig. 9.10, the maximum nodal front size is 8, which corresponds to 24 dof. The block size IBL needed is 24×24. By the banded storage method, the maximum matrix size for the problem is 60×36 (verify this using bandwidth evaluation). The prefront routine in the program uses a simple algorithm to evaluate this block size. In the actual program, a small modification is introduced to handle multipoint constraints. This aspect is discussed later. First, the stiffness matrix S(IBL, IBL) is defined. An index array INDX(IBL), initialized as INDX(I) = I for I = 1 to IBL is defined. We also define the global dof array ISBL(IBL) initialized to all zeros and IEBL() of size equal to the number of dof per element. The front size NFRON and number of variables ready for elimination NTOGO are initialized to zero. The element assembly starts at this initial setting.

Element Assembly and Consideration of Specified dof

Now consider the assembly of a new element when NFRON is at some level and all the variables ready for elimination have been eliminated; that is, NTOGO is zero. We consider the dof of each node of an element using connectivity. Consider the jth dof, say, IDF, of an element and, say, the corresponding node is i. The first search is made in the NFRON locations ISBL(INDX(L)), L = 1 to NFRON if the dof IDF is already in the set. If IDF is already in the set at L = K, then we set IEBL(j) = INDX(K). If IDF is not in the set, then we find the next open location as follows: we set K = NFRON + 1 and ISBL(INDX(K)) = IDF and IEBL(j) = INDX(K). NFRON is incremented by 1 (i.e., set NFRON = NFRON + 1). If node i is negative in the element connectivity, then this dof will be ready for elimination and must be floated into NTOGO. If IDF is a dof that has a specified value, a large penalty number CNST is added into the location S(INDX(K), INDX(K)), and CNST times the specified value is added into the global force location F(IDF). If K > NTOGO, the floating operation is carried out as follows: the number in INDX(NTOGO + 1) is exchanged with the number in INDX(K), and NTOGO is incremented by 1. When all the element dof are completed, IEBL() will have the locations in S() where the element stiffness is to be assembled. The element stiffness matrix SE() is added into the S() locations using IEBL(). The variables in INDX(I) from I = 1 to NTOGO are now ready for elimination. The relationships of various arrays used in the assembly process are shown in Fig. 9.11.

Elimination of Completed dof

We eliminate the variable in the location INDX(1) by reducing the active equations INDX(2) through INDX(NFRON). The equation INDX(1) is written to the disk by writing the stiffness values and the corresponding dof numbers. In the BASIC program, the data are written to a random-access file. Now INDX(1) is open. A few integer exchange operations are done to simplify the elimination process. First, the number at INDX(NTOGO) is exchanged with the number at INDX(1); then the number at INDX(NTOGO) is exchanged with the number at INDX(NFRON). NTOGO and

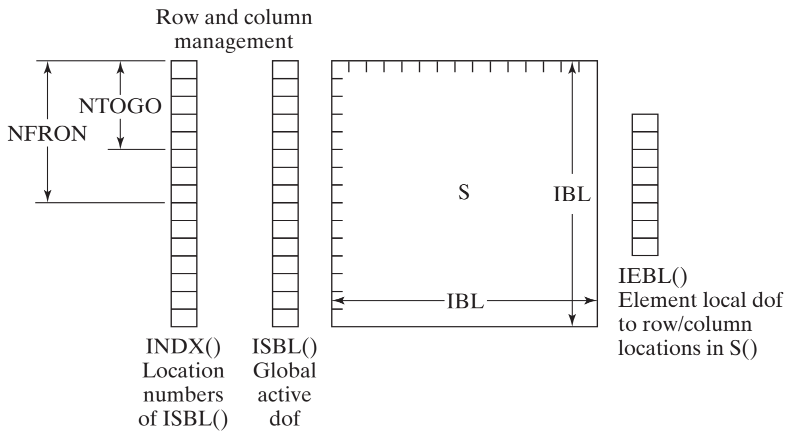


FIGURE 9.11 Stiffness assembly for frontal method.

NFRON are each decremented by 1. Once again, the reduction is carried out from INDX(2) through INDX(NFRON). The process continues for each element until NTOGO is zero or NFRON is 1.

Backsubstitution

Backsubstitution is a straightforward process. In the last equation, there is a stiffness value, a variable number, and the right-hand side. This variable is easily determined. The last but one equation will have two stiffnesses and two variable numbers and the right-hand side. Since one of the variables has already been determined, the other one can be calculated and so on. The backsubstitution can even be carried out independently, if needed.

Consideration of Multipoint Constraints

Multipoint constraints of the type $\beta_1 Q_i + \beta_2 Q_j = \beta_0$ are easily considered by treating each constraint as an element of two dof. The penalty parameter CNST is determined using the first element diagonal stiffness values. The equivalent element stiffness and right-hand side for the multipoint constraint are, respectively,

$$\text{CNST}\begin{bmatrix} \beta_1^2 & \beta_1\beta_2 \\ \beta_1\beta_2 & \beta_2^2 \end{bmatrix} \quad \text{and} \quad \text{CNST}\begin{bmatrix} \beta_1\beta_0 \\ \beta_2\beta_0 \end{bmatrix}$$

In the implementation of this boundary condition, these stiffnesses are first introduced into S() and then the regular element stiffnesses are introduced. The same procedure introduced into the PREFRONT with dof used instead of node numbers gives the needed block size. Assembly and elimination are then similar to the procedure discussed previously.

Example 9.2

The L-shaped beam shown in Fig. 9.10 is analyzed using program HEXAFRON. The input and output data and the program listing are given below. ■

Input Data/Output

```
INPUT TO HEXA
3-D ANALYSIS USING HEXAHEDRAL ELEMENT
EXAMPLE STRUCTURE IN FIG. 9.10
NN    NE    NM    NDIM  NEN   NDN
20    4     1     3     8     3
ND    NL    NMPC
12    1     0
Node# X     Y     Z
1     100   0     100
2     0     0     100
3     0     0     200
4     100   0     200
5     100   100   100
6     0     100   100
7     0     100   200
8     100   100   200
9     100   200   100
10    0     200   100
11    0     200   200
12    100   200   200
13    100   300   100
14    0     300   100
15    0     300   200
16    100   300   200
17    100   200   0
18    100   300   0
19    0     300   0
20    0     200   0
Elem# N1    N2    N3    N4    N5    N6    N7    N8    MAT#  Temp_Ch
1     1     2     3     4     5     6     7     8     1     0
2     5     6     7     8     9     10    11    12    1     0
3     9     10    11    12    13    14    15    16    1     0
4     9     10    14    13    17    20    19    18    1     0
DOF#  Displ.
49    0
50    0
51    0
52    0
53    0
54    0
55    0
56    0
57    0
58    0
59    0
60    0
```

```
DOF#  Load
12    -80000
MAT#  E        Nu       Alpha
1     2.00E+05          0.3      0.00E+00
B1    i        B2       j        B3  (Multi-point constr. B1*Qi+B2*Qj=B3)
```

```
OUTPUT FROM HEXA
Program HexaFront - 3D Stress Analysis
EXAMPLE STRUCTURE IN FIG. 9.10
Node#       X-Displ           Y-Displ           Z-Displ
1        -0.021568703      -0.003789445      -0.409828806
2        -0.025306207      -0.003307915      -0.33229407
3         0.057350356      -0.178955652      -0.326759294
4         0.057756218      -0.184485091      -0.427797766
5        -0.006825272      -0.010486851      -0.223092587
6        -0.010749555      -0.01168289       -0.167106482
7         0.049096429      -0.172496827      -0.167070616
8         0.042790517      -0.173762353      -0.21737743
9         0.013642693      -0.033666087      -0.047867013
10       -6.02951E-05      -0.032557668      -0.029338382
11        0.032541239      -0.14954021       -0.037397392
12        0.027861532      -0.148035354      -0.063003081
13        0.003657825      -0.038341794       0.029087101
14        0.011885798      -0.041411438       0.039684492
15        0.026633192      -0.138416599       0.055798691
16        0.02060568       -0.134943716       0.03958274
17        2.89951E-15      -1.8921E-15       -1.23346E-14
18       -1.60149E-15       1.95027E-15       5.24369E-15
19        1.65966E-15       9.67455E-16       8.93813E-15
20       -2.95768E-15      -1.02562E-15      -8.93813E-15
vonMises Stresses in Elements
Elem# 1      vonMises Stresses at 8 Integration Points
  23.359      15.984      18.545           40.582
  29.929      17.910      14.849           43.396
Elem# 2      vonMises Stresses at 8 Integration Points
  31.416      26.193      35.722           28.272
  61.174      38.615      35.948           51.608
Elem# 3      vonMises Stresses at 8 Integration Points
  45.462      52.393      34.486           32.530
  30.872      41.090      26.155           21.852
Elem# 4      vonMises Stresses at 8 Integration Points
  58.590      46.398      48.482           41.407
  51.148      38.853      49.391           38.936
```

PROBLEMS

9.1. Determine the deflections at the corner points of the steel cantilever beam shown in Fig. P9.1.

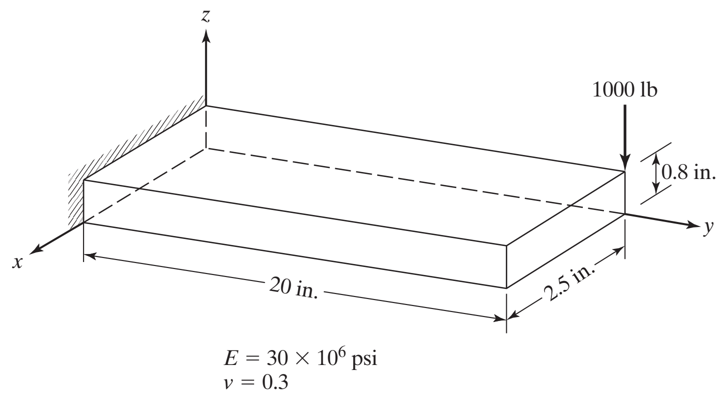


FIGURE P9.1

9.2. A cast iron hollow member used in a machine tool structure is fixed at one end and loaded at the other, as shown in Fig. P9.2. Find the deflection at the load and maximum principal stresses. Compare the values with the structure without an opening.

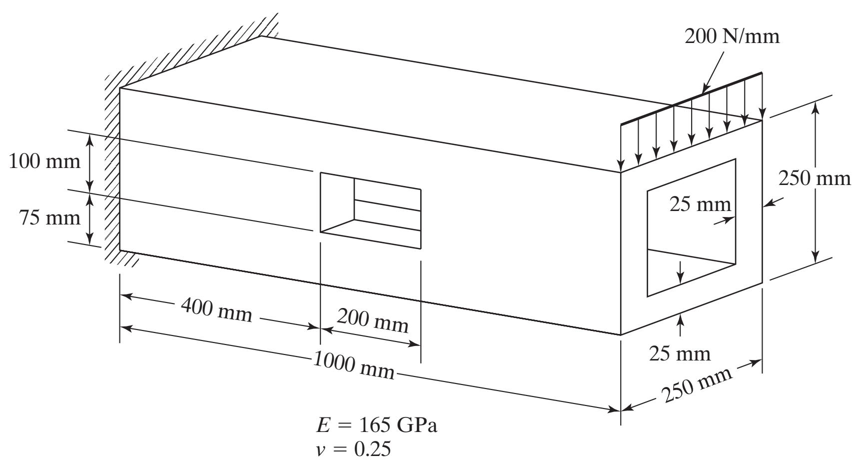


FIGURE P9.2

9.3. An S-shaped block used in force measurement is subjected to a load as shown in Fig. P9.3. Determine the amount by which the block is compressed. Take $E = 70{,}000\ \text{N/mm}^2$ and $\nu = 0.3$.

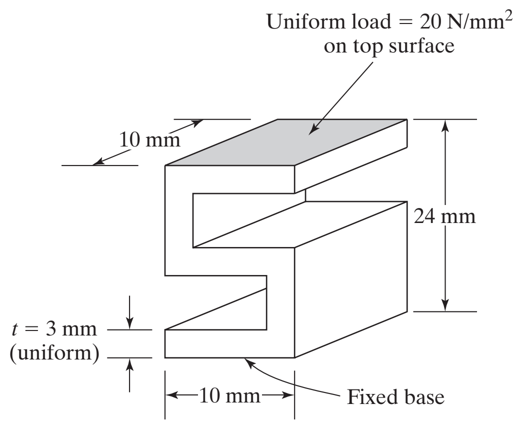


FIGURE P9.3

9.4. A device is hydraulically loaded as shown in Fig. P9.4. Plot the deformed configuration, and determine the magnitude and location of the maximum principal stresses.

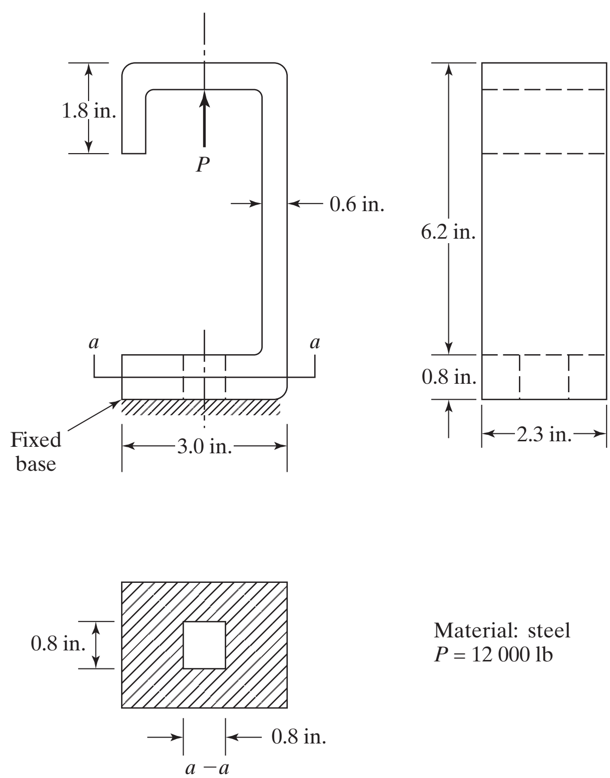


FIGURE P9.4

9.5. A portion of the brake pedal in automobile is modeled as shown in Fig. P9.5. Determine the deflection at the pedal for a 500-N load.

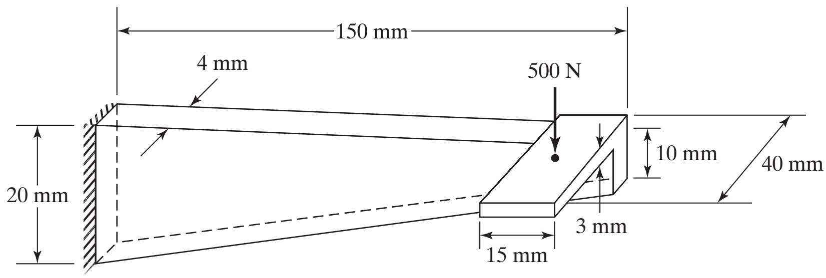
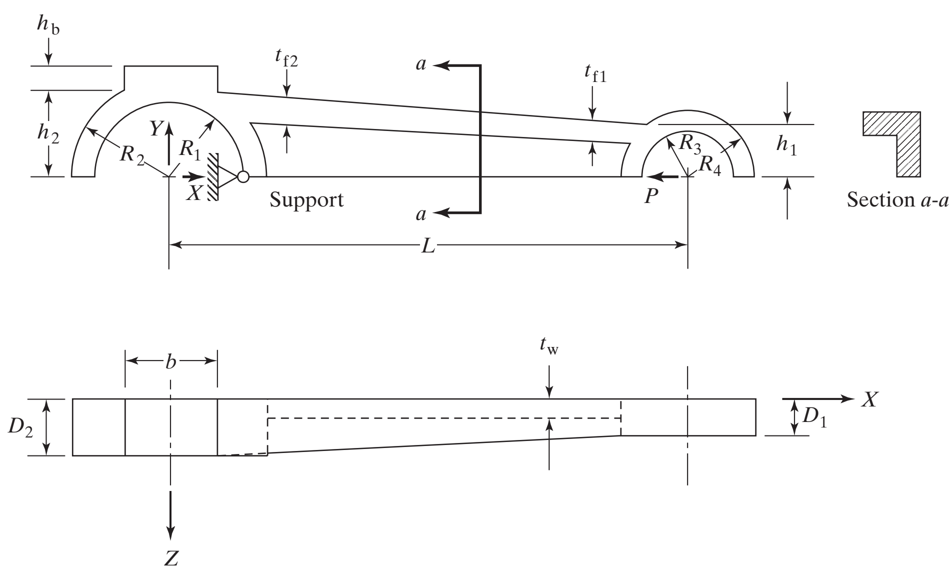


FIGURE P9.5

9.6. Determine the axial elongation and location and magnitude of maximum von Mises stress in the connecting rod shown in Fig. P9.6.



$\frac{1}{4}$ Symmetry model: x, z and x, y are planes of symmetry.
Dimensions (mm)

$R_1 = 20$	$R_2 = 26$	$R_3 = 12$	$R_4 = 18$	$L = 140$
$t_{f1} = 4$	$t_{f2} = 6$	$t_w = 5$	$h_b = 7$	
$h_1 = 14$	$h_2 = 23$	$D_1 = 10$	$D_2 = 15$	$b = 12.1$

$P = 30000$-N compression
Material: steel

FIGURE P9.6

9.7. An overhanging beam made of rigidly bonded steel and aluminum plates is shown in Fig. P9.7. The aluminum plate has a constant thickness of 10 mm. Due to a manufacturing defect, the steel plate has straight edges, a thickness of 9 mm at one of the free ends, and 10 mm at other corners. From the position shown, if the temperature is raised 60°C, determine the following:

(a) The deformed shape
(b) The maximum vertical deflection and its location
(c) The maximum von Mises stress and its location.

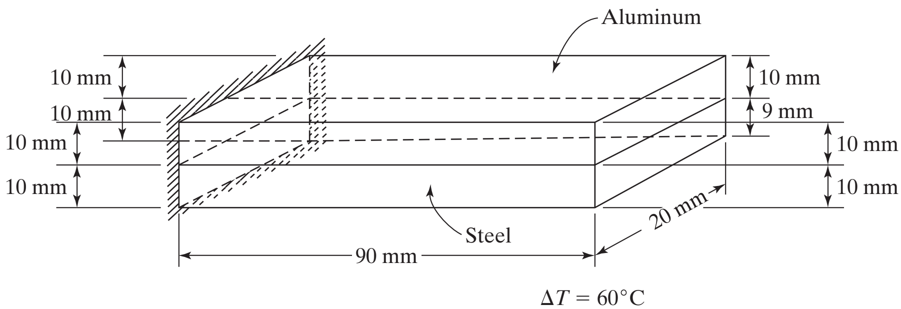


FIGURE P9.7

9.8. Determine the nodal deformations and element stresses for the beam shown in Fig. P9.8 using hexahedral elements. Plot the deformed surface shown shaded in the figure (use MATLAB for your plotting).

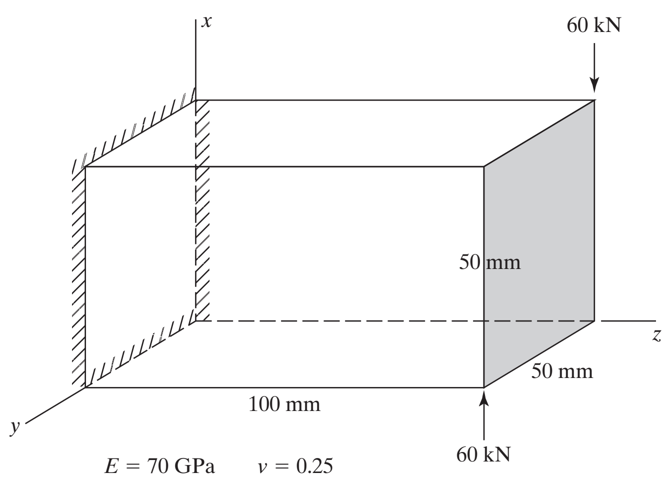


FIGURE P9.8

9.9. Measure the dimensions of a commonly used wrench for tightening 25-mm across-the-flat hexagonal head bolts of height 10 mm. Apply the boundary conditions and load for tightening, and determine the deformation and stresses using three-dimensional formulation.

Program Listings

MAIN PROGRAM

```
'*****     PROGRAM HEXAFNT              *****
'*   3-D STRESS ANALYSIS USING 8-NODE       *
'*   ISOPARAMETRIC HEXAHEDRAL ELEMENT       *
'*      USING FRONTAL SOLVER                *
'*   T.R.Chandrupatla and A.D.Belegundu     *
'*******************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call PreFront
     RecordLen = Len(Adat)
     '--- Scratch file for writing
     Open "SCRATCH.DAT" For Random As #3 Len = RecordLen
     Call Stiffness
     Call BackSub
     Close #3
     Kill "SCRATCH.DAT"
     Call StressCalc
     Call ReactionCalc
     Call Output
End Sub
```

PREPARATION FOR FRONTAL METHOD

```
Private Sub PreFront()
        '----- Mark Last Appearance of Node / Make it negative in NOC()
        ' Last appearance is first appearance for reverse element order
        NEDF = NEN * NDN
        For I = 1 To NN
           II = 0
           For J = NE To 1 Step -1
              For K = 1 To NEN
                 If I = NOC(J, K) Then
                     II = 1
                     Exit For
                 End If
              Next K
              If II = 1 Then Exit For
           Next J
           NOC(J, K) = -I
        Next I
        '===== Block Size Determination
        NQ = NN * NDN
        ReDim IDE(NQ)
        For I = 1 To NQ: IDE(I) = 0: Next I
        For I = 1 To NMPC: For J = 1 To 2: IDE(MPC(I, J)) = 1
        Next J: Next I
        IFRON = 0: For I = 1 To NQ: IFRON = IFRON + IDE(I): Next I
        IBL = IFRON
```

```
        For N = 1 To NE
           INEG = 0
           For I = 1 To NEN
              I1 = NOC(N, I): IA = NDN * (Abs(I1) - 1)
              For J = 1 To NDN
                 IA = IA + 1
                 If IDE(IA) = 0 Then
                     IFRON = IFRON + 1: IDE(IA) = 1
                 End If
              Next J
              If I1 < 0 Then INEG = INEG + 1
           Next I
           If IBL < IFRON Then IBL = IFRON
           IFRON = IFRON - NDN * INEG
        Next N
        Erase IDE
        ReDim ISBL(IBL), S(IBL, IBL), IEBL(NEDF), INDX(IBL)
        NFRON = 0: NTOGO = 0: NDCNT = 0
        For I = 1 To IBL: INDX(I) = I: Next I
End Sub
```

```
STIFFNESS MATRIX
Private Sub Stiffness()
     '----- Global Stiffness Matrix -----
     Call IntegPoints
     For N = 1 To NE
        Call DMatrix(N)
        Call ElemStiffness(N)
        If N = 1 Then
           CNST = 0
           For I = 1 To NEDF: CNST = CNST + SE(I, I): Next I
           CNST = 100000000000# * CNST
           Call MpcFron
        End If
        '----- Account for temperature loads QT()
        For I = 1 To NEN
           IL = 3 * (I - 1): IG = 3 * (Abs(NOC(N, I)) - 1)
           For J = 1 To 3
              IL = IL + 1: IG = IG + 1
              F(IG) = F(IG) + QT(IL)
           Next J
        Next I
        '----- Frontal assembly and Forward Elimination
        Call Front(N)
     Next N
End Sub
```

```
STRESS CALCULATIONS
Private Sub StressCalc()
     ReDim vonMisesStress(NE, 8)
     '----- Stress Calculations
     For N = 1 To NE
        Call DMatrix(N)
        For IP = 1 To 8
        '--- Von Mises Stress at Integration Points
        '--- Get DB Matrix with Stress calculation
           Call DbMat(N, 2, IP)
           '--- Calculation of Von Mises Stress at IP
           SIV1 = STR(1) + STR(2) + STR(3)
           SIV2 = STR(1) * STR(2) + STR(2) * STR(3) + STR(3) * STR(1)
           SIV2 = SIV2 - STR(4) ^ 2 - STR(5) ^ 2 - STR(6) ^ 2
           vonMisesStress(N, IP) = Sqr(SIV1 * SIV1 - 3 * SIV2)
        Next IP
     Next N
End Sub
```

```
INTEGRATION POINTS
Private Sub IntegPoints()
'------- Integration Points XNI() --------
     C = 0.57735026919
     XI(1, 1) = -1: XI(2, 1) = -1: XI(3, 1) = -1
     XI(1, 2) = 1: XI(2, 2) = -1: XI(3, 2) = -1
     XI(1, 3) = 1: XI(2, 3) = 1: XI(3, 3) = -1
     XI(1, 4) = -1: XI(2, 4) = 1: XI(3, 4) = -1
     XI(1, 5) = -1: XI(2, 5) = -1: XI(3, 5) = 1
     XI(1, 6) = 1: XI(2, 6) = -1: XI(3, 6) = 1
     XI(1, 7) = 1: XI(2, 7) = 1: XI(3, 7) = 1
     XI(1, 8) = -1: XI(2, 8) = 1: XI(3, 8) = 1
     For I = 1 To 8
        XNI(1, I) = C * XI(1, I)
        XNI(2, I) = C * XI(2, I)
        XNI(3, I) = C * XI(3, I)
     Next I
End Sub
```

```
DB MATRIX
Private Sub DMatrix(N)
     '--- D() Matrix relating Stresses to Strains
     M = MAT(N)
     E = PM(M, 1): PNU = PM(M, 2): AL = PM(M, 3)
     C1 = E / ((1 + PNU) * (1 - 2 * PNU))
     C2 = 0.5 * E / (1 + PNU)
     For I = 1 To 6: For J = 1 To 6: D(I, J) = 0: Next J: Next I
     D(1, 1) = C1 * (1 - PNU): D(1, 2) = C1 * PNU: D(1, 3) = D(1, 2)
     D(2, 1) = D(1, 2): D(2, 2) = D(1, 1): D(2, 3) = D(1, 2)
     D(3, 1) = D(1, 3): D(3, 2) = D(2, 3): D(3, 3) = D(1, 1)
     D(4, 4) = C2: D(5, 5) = C2: D(6, 6) = C2
End Sub
```

```
ELEMENT STIFFNESS
Private Sub ElemStiffness(N)
'-------- Element Stiffness -----
     For I = 1 To 24: For J = 1 To 24
     SE(I, J) = 0: Next J: QT(I) = 0: Next I
     DTE = DT(N)
     '--- Weight Factor is ONE
     '--- Loop on Integration Points
     For IP = 1 To 8
        '--- Get DB Matrix at Integration Point IP
        Call DbMat(N, 1, IP)
        '--- Element Stiffness Matrix SE
        For I = 1 To 24
           For J = 1 To 24
              For K = 1 To 6
                 SE(I, J) = SE(I, J) + B(K, I) * DB(K, J) * DJ
              Next K
           Next J
        Next I
        '--- Determine Temperature Load QT()
        C = PM(MAT(N), 3) * DTE
        For I = 1 To 24
           DSum = DB(1, I) + DB(2, I) + DB(3, I)
           QT(I) = QT(I) + C * Abs(DJ) * DSum / 6
        Next I
     Next IP
End Sub
```

```
DB MATRIX
Private Sub DbMat(N, ISTR, IP)
'------- DB() MATRIX ------
     '--- Gradient of Shape Functions - The GN() Matrix
     For I = 1 To 3
        For J = 1 To 8
           C = 1
           For K = 1 To 3
              If K <> I Then
                 C = C * (1 + XI(K, J) * XNI(K, IP))
              End If
           Next K
           GN(I, J) = 0.125 * XI(I, J) * C
        Next J
     Next I
     '--- Formation of Jacobian TJ
     For I = 1 To 3
        For J = 1 To 3
           TJ(I, J) = 0
           For K = 1 To 8
              KN = Abs(NOC(N, K))
              TJ(I, J) = TJ(I, J) + GN(I, K) * X(KN, J)
```

continued

```
            Next K
         Next J
      Next I
      '--- Determinant of the JACOBIAN
      DJ1 = TJ(1, 1) * (TJ(2, 2) * TJ(3, 3) - TJ(3, 2) * TJ(2, 3))
      DJ2 = TJ(1, 2) * (TJ(2, 3) * TJ(3, 1) - TJ(3, 3) * TJ(2, 1))
      DJ3 = TJ(1, 3) * (TJ(2, 1) * TJ(3, 2) - TJ(3, 1) * TJ(2, 2))
      DJ = DJ1 + DJ2 + DJ3
      '--- Inverse of the Jacobian AJ()
      AJ(1, 1) = (TJ(2, 2) * TJ(3, 3) - TJ(2, 3) * TJ(3, 2)) / DJ
      AJ(1, 2) = (TJ(3, 2) * TJ(1, 3) - TJ(3, 3) * TJ(1, 2)) / DJ
      AJ(1, 3) = (TJ(1, 2) * TJ(2, 3) - TJ(1, 3) * TJ(2, 2)) / DJ
      AJ(2, 1) = (TJ(2, 3) * TJ(3, 1) - TJ(2, 1) * TJ(3, 3)) / DJ
      AJ(2, 2) = (TJ(1, 1) * TJ(3, 3) - TJ(1, 3) * TJ(3, 1)) / DJ
      AJ(2, 3) = (TJ(1, 3) * TJ(2, 1) - TJ(1, 1) * TJ(2, 3)) / DJ
      AJ(3, 1) = (TJ(2, 1) * TJ(3, 2) - TJ(2, 2) * TJ(3, 1)) / DJ
      AJ(3, 2) = (TJ(1, 2) * TJ(3, 1) - TJ(1, 1) * TJ(3, 2)) / DJ
      AJ(3, 3) = (TJ(1, 1) * TJ(2, 2) - TJ(1, 2) * TJ(2, 1)) / DJ
      '--- H() Matrix relates local derivatives of u to local
      '  displacements q
      For I = 1 To 9
         For J = 1 To 24
            H(I, J) = 0
         Next J
      Next I
      For I = 1 To 3
         For J = 1 To 3
            IR = 3 * (I - 1) + J
            For K = 1 To 8
               IC = 3 * (K - 1) + I
               H(IR, IC) = GN(J, K)
            Next K
         Next J
      Next I
      '--- G() Matrix relates strains to local derivatives of u
      For I = 1 To 6
         For J = 1 To 9
            G(I, J) = 0
         Next J
      Next I
      G(1, 1) = AJ(1, 1): G(1, 2) = AJ(1, 2): G(1, 3) = AJ(1, 3)
      G(2, 4) = AJ(2, 1): G(2, 5) = AJ(2, 2): G(2, 6) = AJ(2, 3)
      G(3, 7) = AJ(3, 1): G(3, 8) = AJ(3, 2): G(3, 9) = AJ(3, 3)
      G(4, 4) = AJ(3, 1): G(4, 5) = AJ(3, 2): G(4, 6) = AJ(3, 3)
      G(4, 7) = AJ(2, 1): G(4, 8) = AJ(2, 2): G(4, 9) = AJ(2, 3)
      G(5, 1) = AJ(3, 1): G(5, 2) = AJ(3, 2): G(5, 3) = AJ(3, 3)
      G(5, 7) = AJ(1, 1): G(5, 8) = AJ(1, 2): G(5, 9) = AJ(1, 3)
      G(6, 1) = AJ(2, 1): G(6, 2) = AJ(2, 2): G(6, 3) = AJ(2, 3)
      G(6, 4) = AJ(1, 1): G(6, 5) = AJ(1, 2): G(6, 6) = AJ(1, 3)
      '--- B() Matrix relates strains to q
      For I = 1 To 6
```

```
        For J = 1 To 24
           B(I, J) = 0
           For K = 1 To 9
              B(I, J) = B(I, J) + G(I, K) * H(K, J)
           Next K
        Next J
     Next I
     '--- DB() Matrix relates stresses to q
     For I = 1 To 6
        For J = 1 To 24
           DB(I, J) = 0
           For K = 1 To 6
              DB(I, J) = DB(I, J) + D(I, K) * B(K, J)
           Next K
        Next J
     Next
     If ISTR = 1 Then Exit Sub
           '--- Element Nodal Displacements stored in QT()
           For I = 1 To 8
              IIN = 3 * (Abs(NOC(N, I)) - 1)
              II = 3 * (I - 1)
              For J = 1 To 3
                 QT(II + J) = F(IIN + J)
              Next J
           Next I
           '--- Stress Calculation STR = DB * Q
           CAL = PM(MAT(N), 3) * DT(N)
           For I = 1 To 6
              STR(I) = 0
              For J = 1 To 24
                 STR(I) = STR(I) + DB(I, J) * QT(J)
              Next J
              STR(I) = STR(I) - CAL * (D(I, 1) + D(I, 2) + D(I, 3))
           Next I
End Sub
```

```
MULTIPOINT CONSTRAINT HANDLING
Private Sub MpcFron()
    '----- Modifications for Multipoint Constraints by Penalty Method
       For I = 1 To NMPC
          I1 = MPC(I, 1)
          IFL = 0
          For J = 1 To NFRON
             J1 = INDX(J)
             If I1 = ISBL(J1) Then
                 IFL = 1: Exit For
             End If
          Next J
          If IFL = 0 Then
             NFRON = NFRON + 1: J1 = INDX(NFRON): ISBL(J1) = I1
```

continued

```
            End If
            I2 = MPC(I, 2)
            IFL = 0
            For K = 1 To NFRON
               K1 = INDX(K)
               If K1 = ISBL(K1) Then
                      IFL = 1: Exit For
               End If
            Next K
            If IFL = 0 Then
               NFRON = NFRON + 1: K1 = INDX(NFRON): ISBL(K1) = I2
            End If
            '----- Stiffness Modification
            S(J1, J1) = S(J1, J1) + CNST * BT(I, 1) ^ 2
            S(K1, K1) = S(K1, K1) + CNST * BT(I, 2) ^ 2
            S(J1, K1) = S(J1, K1) + CNST * BT(I, 1) * BT(I, 2)
            S(K1, J1) = S(J1, K1)
            '----- Force Modification
            F(I1) = F(I1) + CNST * BT(I, 3) * BT(I, 1)
            F(I2) = F(I2) + CNST * BT(I, 3) * BT(I, 2)
         Next I
End Sub
```

```
FRONTAL REDUCTION - WRITE TO DISK
Private Sub Front(N)
'----- Frontal Method Assembly and Elimination -----
'---------------- Assembly of Element N --------------------
        For I = 1 To NEN
           I1 = NOC(N, I): IA = Abs(I1): IS1 = Sgn(I1)
           IDF = NDN * (IA - 1): IE1 = NDN * (I - 1)
           For J = 1 To NDN
              IDF = IDF + 1: IE1 = IE1 + 1: IFL = 0
              If NFRON > NTOGO Then
                 For II = NTOGO + 1 To NFRON
                    IX = INDX(II)
                    If IDF = ISBL(IX) Then
                       IFL = 1: Exit For
                     End If
                 Next II
              End If
              If IFL = 0 Then
                 NFRON = NFRON + 1: II = NFRON: IX = INDX(II)
              End If
              ISBL(IX) = IDF: IEBL(IE1) = IX
              If IS1 = -1 Then
                 NTOGO = NTOGO + 1
                 ITEMP = INDX(NTOGO)
                 INDX(NTOGO) = INDX(II)
                 INDX(II) = ITEMP
              End If
```

```
            Next J
         Next I
         For I = 1 To NEDF
            I1 = IEBL(I)
            For J = 1 To NEDF
               J1 = IEBL(J)
               S(I1, J1) = S(I1, J1) + SE(I, J)
            Next J
         Next I
'-------------------------------------------------------------------
   If NDCNT < ND Then
'----- Modification for displacement BCs / Penalty Approach -----
         For I = 1 To NTOGO
            I1 = INDX(I)
            IG = ISBL(I1)
               For J = 1 To ND
                  If IG = NU(J) Then
                     S(I1, I1) = S(I1, I1) + CNST
                     F(IG) = F(IG) + CNST * U(J)
                     NDCNT = NDCNT + 1       'Counter for check
                     Exit For
                  End If
               Next J
            Next I
         End If
'------------  Elimination of completed variables  ---------------
         NTG1 = NTOGO
         For II = 1 To NTG1
            IPV = INDX(1): IPG = ISBL(IPV)
            Pivot = S(IPV, IPV)
         '----- Write separator "0" and PIVOT value to disk -----
            Adat.VarNum = 0
            Adat.Coeff = Pivot
            ICOUNT = ICOUNT + 1
            Put #3, ICOUNT, Adat
            S(IPV, IPV) = 0
            For I = 2 To NFRON
               I1 = INDX(I): IG = ISBL(I1)
               If S(I1, IPV) <> 0 Then
                  C = S(I1, IPV) / Pivot: S(I1, IPV) = 0
                  For J = 2 To NFRON
                     J1 = INDX(J)
                     If S(IPV, J1) <> 0 Then
                        S(I1, J1) = S(I1, J1) - C * S(IPV, J1)
                     End If
                  Next J
                  F(IG) = F(IG) - C * F(IPG)
               End If
            Next I
            For J = 2 To NFRON
```

continued

```
        '----- Write Variable# and Reduced Coeff/PIVOT to disk -----
           J1 = INDX(J)
           If S(IPV, J1) <> 0 Then
              ICOUNT = ICOUNT + 1: IBA = ISBL(J1)
              Adat.VarNum = IBA
              Adat.Coeff = S(IPV, J1) / Pivot
              Put #3, ICOUNT, Adat
              S(IPV, J1) = 0
           End If
        Next J
        ICOUNT = ICOUNT + 1
    '----- Write Eliminated Variable# and RHS/PIVOT to disk -----
        Adat.VarNum = IPG
        Adat.Coeff = F(IPG) / Pivot
        F(IPG) = 0
        Put #3, ICOUNT, Adat
    '----- (NTOGO) into (1); (NFRON) into (NTOGO)
    '----- IPV into (NFRON) and reduce front & NTOGO sizes by 1
        If NTOGO > 1 Then
           INDX(1) = INDX(NTOGO)
        End If
        INDX(NTOGO) = INDX(NFRON): INDX(NFRON) = IPV
        NFRON = NFRON - 1: NTOGO = NTOGO - 1
    Next II
End Sub
```

```
BACKSUBSTITUTION
Private Sub BackSub()
        '===== Backsubstitution
        Do While ICOUNT > 0
           Get #3, ICOUNT, Adat
           ICOUNT = ICOUNT - 1
           N1 = Adat.VarNum
           F(N1) = Adat.Coeff
           Do
              Get #3, ICOUNT, Adat
              ICOUNT = ICOUNT - 1
              N2 = Adat.VarNum
              If N2 = 0 Then Exit Do
              F(N1) = F(N1) - Adat.Coeff * F(N2)
           Loop
         Loop
End Sub
```

C H A P T E R 10

Scalar Field Problems

10.1 INTRODUCTION

In previous chapters, the unknowns in the problem represented components of a *vector* field. In a two-dimensional plate, for example, the unknown quantity is the vector field $\mathbf{u}(x, y)$, where $\mathbf{u}$ is a (2×1) displacement vector. On the other hand, quantities such as temperature, pressure, and stream potentials are ***scalar*** in nature. In two-dimensional steady-state heat conduction, for example, the temperature field $\mathbf{T}(x, y)$ is the unknown to be determined.

In this chapter, the finite element method for solving such problems is discussed. In Section 10.2, one-dimensional and two-dimensional steady-state heat conduction are considered, as well as temperature distribution in fins. Section 10.3 deals with torsion of solid shafts. Scalar field problems related to fluid flow, seepage, electric/magnetic fields, and flow in ducts are defined in Section 10.4.

The striking feature of scalar field problems is that they are to be found in almost all branches of engineering and physics. Most of them can be regarded as special forms of the general Helmholtz equation, given by

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial \phi}{\partial z}\right) + \lambda\phi + Q = 0 \qquad (10.1)$$

together with boundary conditions on ϕ and its derivatives. In Eq. 10.1, $\phi = \phi(x, y, z)$ is the field variable that has to be determined. Table 10.1 lists some of the engineering problems described by Eq. 10.1. For example, if we set $\phi = T$, $k_x = k_y = k$, and $\lambda = 0$ and consider only x and y, we get $\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2 + Q = 0$, which describes the heat-conduction problem for temperature T, where k is the thermal conductivity and Q is the heat source/sink. Mathematically, we can develop the finite element method for various field problems in a general manner by considering Eq. 10.1. The solution to specific problems can then be obtained by suitable definition of variables. We discuss here the heat-transfer and torsion problems in some detail. These are important in themselves, because they provide us an opportunity to understand the physical problem and how to handle different boundary conditions needed for modeling. Once the steps are

TABLE 10.1 Examples of Scalar Field Problems on Engineering

Problem	Equation	Field Variable	Parameter	Boundary Conditions
Heat conduction	$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$	Temperature, T	Thermal conductivity, k	$T=T_0,\ -k\frac{\partial T}{\partial n}=q_0$ $-k\frac{\partial T}{\partial n}=h(T-T_\infty)$
Torsion	$\left(\frac{\partial^2 \theta}{\partial x^2}+\frac{\partial^2 \theta}{\partial y^2}\right)+2=0$	Stress function, θ		$\theta=0$
Potential flow	$\left(\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial y^2}\right)=0$	Stream function, ψ		$\psi=\psi_0$
Seepage and groundwater flow	$k\left(\frac{\partial^2 \phi}{\partial x^2}+\frac{\partial^2 \phi}{\partial y^2}\right)+Q=0$	Hydraulic potential, ϕ	Hydraulic conductivity, k	$\phi=\phi_0$ $\frac{\partial \phi}{\partial n}=0$ $\phi=y$
Electric potential	$\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=-\rho$	Electric potential, u	Permittivity, ϵ	$u=u_0,\ \frac{\partial u}{\partial n}=0$
Fluid flow in ducts	$\left(\frac{\partial^2 W}{\partial X^2}+\frac{\partial^2 W}{\partial Y^2}\right)+1=0$	Nondimensional velocity, W		$W=0$
Acoustics	$\left(\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}\right)+k^2 p=0$	Pressure, p (complex)	Wave number, $k^2=\omega^2/c^2$	$p=p_0$, $\frac{1}{ik\rho c}\frac{\partial p}{\partial n}=v_0$

understood, extension to other areas in engineering should present no difficulty. While in other chapters, both energy and Galerkin approaches were used to derive element matrices, derivation by Galerkin's approach is used here owing to its greater generality for field problems.

10.2 STEADY-STATE HEAT TRANSFER

We now discuss the finite element formulation for the solution of steady-state heat-transfer problems. Heat transfer occurs when there is a temperature difference within a body or between a body and its surrounding medium. Heat is transferred in the form of conduction, convection, and thermal radiation. Only conduction and convection modes are treated here.

The heat flow through the wall of a heated room on a winter day is an example of conduction. The conduction process is quantified by Fourier's law. In a thermally isotropic medium, Fourier's law for two-dimensional heat flow is given by

$$q_x=-k\frac{\partial T}{\partial x}\qquad q_y=-k\frac{\partial T}{\partial y}\tag{10.2}$$

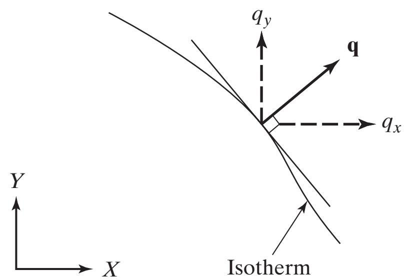


FIGURE 10.1 Heat flux in two dimensions.

where $T = T(x, y)$ is a temperature field in the medium, q_x and q_y are the components of the heat flux (W/m^2), k is the thermal conductivity (W/m· °C), and $\partial T/\partial x$, $\partial T/\partial y$ are the temperature gradients along x and y, respectively. The resultant heat flux $\mathbf{q} = q_x\mathbf{i} + q_y\mathbf{j}$ is at right angles to an isotherm or a line of constant temperature (Fig. 10.1). Note that 1 W = 1 J/s = 1 N·m/s. The minus sign in Eq. 10.2 reflects the fact that heat is transferred in the direction of decreasing temperature. Thermal conductivity k is a material property.

In convection heat transfer, there is transfer of energy between a fluid and a solid surface as a result of a temperature difference. There can be free or natural convection, such as the circulation pattern set up while boiling water in a kettle due to hot water rising and cooler water moving down, or there can be forced convection, such as when the fluid flow is caused by a fan. The governing equation is of the form

$$q = h(T_s - T_\infty) \tag{10.3}$$

where q is the convective heat flux (W/m^2), h is the convection heat-transfer coefficient or film coefficient (W/m^2 · °C), and T_s and T_∞ are the surface and fluid temperatures, respectively. The film coefficient h is a property of the flow and depends on various factors, such as whether convection is natural or forced, whether the flow is laminar or turbulent, the type of fluid, and the geometry of the body.

In addition to conduction and convection, heat transfer can also occur in the form of thermal radiation. The radiation heat flux is proportional to the fourth power of the absolute temperature, which causes the problem to be nonlinear. This mode of heat transfer is not considered here.

One-Dimensional Heat Conduction

We now turn our attention to the steady-state heat-conduction problem in one dimension. Our objective is to determine the temperature distribution. In one-dimensional steady-state problems, a temperature gradient exists along only one coordinate axis, and the temperature at each point is independent of time. Many engineering systems fall into this category.

Governing Equation Consider heat conduction in a plane wall with uniform heat generation (Fig. 10.2). Let A be the area normal to the direction of heat flow, and let Q (W/m^3) be the internal heat generated per unit volume. A common example of

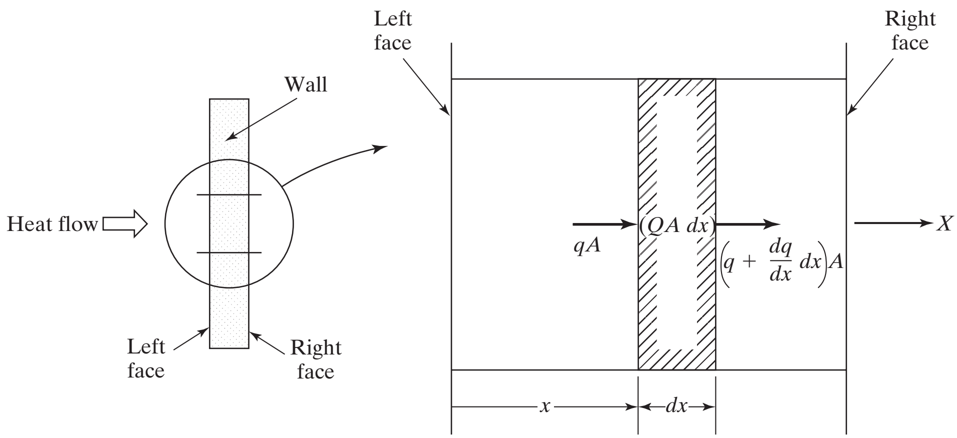


FIGURE 10.2 One-dimensional heat conduction.

heat generation is the heat produced in a wire carrying a current I and having a resistance R through a volume V, which results in $Q = I^2R/V$. A control volume is shown in Fig. 10.2. Since the heat rate (heat flux × area) that is entering the control volume plus the heat rate generated equals the heat rate leaving the control volume, we have

$$qA + QA\,dx = \left(q + \frac{dq}{dx}dx\right)A \tag{10.4}$$

Canceling qA from both sides yields

$$Q = \frac{dq}{dx} \tag{10.5}$$

Substituting Fourier's law

$$q = -k\frac{dT}{dx} \tag{10.6}$$

into Eq. 10.5 results in

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) + Q = 0 \tag{10.7}$$

Usually, Q is called a *source* when positive (heat is generated) and is called a *sink* when negative (heat is consumed). Here, Q will simply be referred to as a source. In general, k in Eq. 10.7 is a function of x. Equation 10.7 has to be solved with appropriate boundary conditions.

Boundary Conditions. The boundary conditions are mainly of three kinds: specified temperature, specified heat flux (or insulated), and convection. For example, consider the wall of a tank containing a hot liquid at a temperature T_0, with an airstream

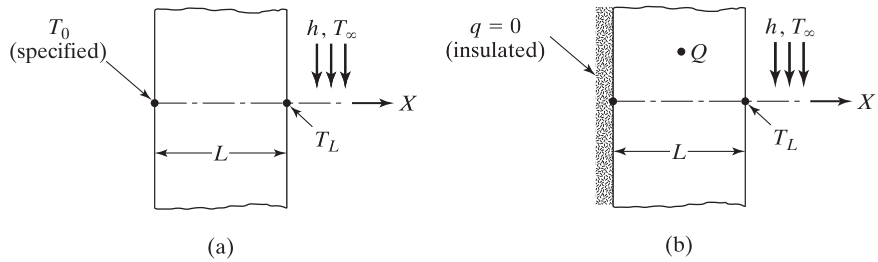


FIGURE 10.3 Examples of boundary conditions in one-dimensional heat conduction.

of temperature T_∞ passed on the outside, maintaining a wall temperature of T_L at the boundary (Fig. 10.3a). The boundary conditions for this problem are

$$T|_{x=0} = T_0 \tag{10.8}$$

$$q|_{x=L} = h(T_L - T_\infty) \tag{10.9}$$

As another example, consider a wall, as shown in Fig. 10.3b, where the inside surface is insulated and the outside is a convection surface. Then, the boundary conditions are

$$q|_{x=0} = 0 \quad q|_{x=L} = h(T_L - T_\infty) \tag{10.10}$$

We will consider the various types of boundary conditions in the sections that follow.

The One-Dimensional Element. The two-node element with linear shape functions is considered subsequently. Extension to three-node quadratic elements follows the same procedures as discussed in Chapter 3. Now, to apply the finite element method, the problem is discretized in the x dimension, as shown in Fig. 10.4a. The temperatures at the various nodal points, denoted by $\mathbf{T}$, are the unknowns (except at node 1, where $T_1 = T_0$). Within a typical element e (Fig. 10.4b), whose local node numbers are 1 and 2, the temperature field is approximated using shape functions N_1 and N_2 as

$$\begin{aligned} T(\xi) &= N_1 T_1 + N_2 T_2 \\ &= \mathbf{N}\mathbf{T}^e \end{aligned} \tag{10.11}$$

where $N_1 = (1 - \xi)/2$, $N_2 = (1 + \xi)/2$, ξ varies from -1 to $+1$, $\mathbf{N} = [N_1 \;\; N_2]$ and $\mathbf{T}^e = [T_1 \;\; T_2]^{\mathrm{T}}$. Noting the relations

$$\begin{aligned} \xi &= \frac{2}{x_2 - x_1}(x - x_1) - 1 \\ d\xi &= \frac{2}{x_2 - x_1}\,dx \end{aligned} \tag{10.12}$$

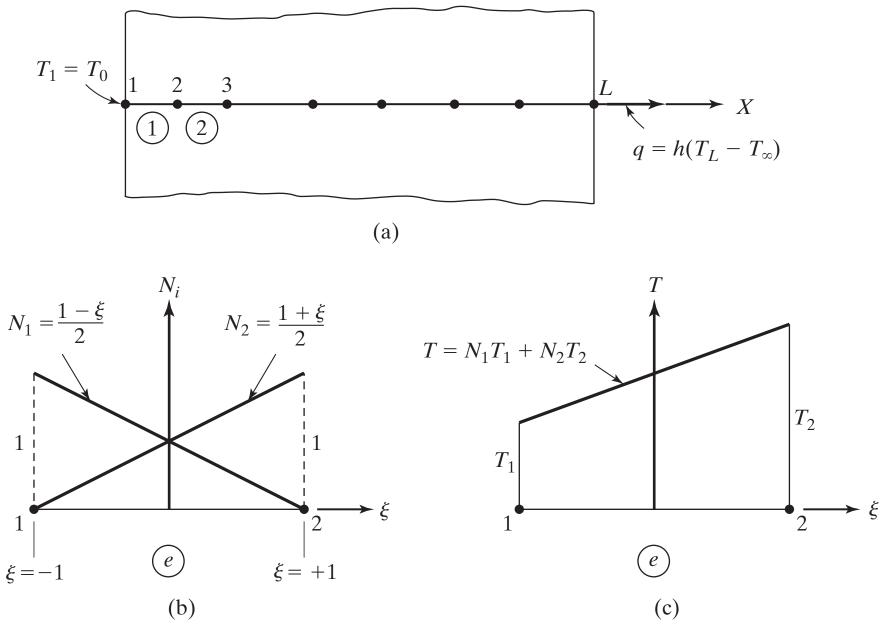


FIGURE 10.4 Finite element modeling and shape functions for linear interpolation of the temperature field.

we get

$$
\begin{aligned}
\frac{dT}{dx} &= \frac{dT}{d\xi}\frac{d\xi}{dx} \\
&= \frac{2}{x_2 - x_1}\frac{d\mathbf{N}}{d\xi}\mathbf{T}^e \qquad (10.13a) \\
&= \frac{1}{x_2 - x_1}[-1 \quad 1]\mathbf{T}^e
\end{aligned}
$$

or

$$\frac{dT}{dx} = \mathbf{B}_T\mathbf{T}^e \qquad (10.13b)$$

where

$$\mathbf{B}_T = \frac{1}{x_2 - x_1}[-1 \quad 1] \qquad (10.14)$$

Galerkin's Approach for Heat Conduction. The element matrices will now be derived using Galerkin's approach. The problem is

$$
\begin{gathered}
\frac{d}{dx}\left(k\frac{dT}{dx}\right) + Q = 0 \\
T\big|_{x=0} = T_0 \quad q\big|_{x=L} = h(T_L - T_\infty) \qquad (10.15)
\end{gathered}
$$

If an approximate solution T is desired, Galerkin's approach is used to solve

$$\int_0^L \phi\left[\frac{d}{dx}\left(k\frac{dT}{dx}\right) + Q\right]dx = 0 \tag{10.16}$$

for every ϕ constructed from the same basis functions as those of T, with $\phi(0) = 0$. ϕ can be thought of as a virtual temperature change that is consistent with boundary conditions. Thus, $\phi = 0$, where T is specified. Integrating the first term by parts, we get

$$\phi k\frac{dT}{dx}\bigg|_0^L - \int_0^L k\frac{d\phi}{dx}\frac{dT}{dx}dx + \int_0^L \phi Q\,dx = 0 \tag{10.17}$$

Now,

$$\phi k\frac{dT}{dx}\bigg|_0^L = \phi(L)k(L)\frac{dT}{dx}(L) - \phi(0)k(0)\frac{dT}{dx}(0) \tag{10.18a}$$

Since $\phi(0) = 0$ and $q = -k(L)[dT(L)/dx] = h(T_L - T_\infty)$, we get

$$\phi k\frac{dT}{dx}\bigg|_0^L = -\phi(L)h(T_L - T_\infty) \tag{10.18b}$$

Thus, Eq. 10.17 becomes

$$-\phi(L)h(T_L - T_\infty) - \int_0^L k\frac{d\phi}{dx}\frac{dT}{dx}dx + \int_0^L \phi Q\,dx = 0 \tag{10.19}$$

We now use the isoparametric relations $T = \mathbf{N}\mathbf{T}^e$, etc., defined in Eqs. 10.11–10.14. Further, a global virtual-temperature vector is denoted as $\boldsymbol{\psi} = [\psi_1 \quad \psi_2 \quad \cdots \quad \psi_L]^{\mathrm{T}}$, and the test function within each element is interpolated as

$$\phi = \mathbf{N}\boldsymbol{\psi} \tag{10.20}$$

Analogous to $dT/dx = \mathbf{B}_T\mathbf{T}^e$ in Eq. 10.13b, we have

$$\frac{d\phi}{dx} = \mathbf{B}_T\boldsymbol{\psi} \tag{10.21}$$

Thus, Eq. 10.19 becomes

$$-\psi_L h(T_L - T_\infty) - \sum_e \psi^{\mathrm{T}}\left(\frac{k_e\ell_e}{2}\int_{-1}^{1}\mathbf{B}_T^{\mathrm{T}}\mathbf{B}_T\,d\xi\right)\mathbf{T}^e + \sum_e \psi^{\mathrm{T}}\frac{Q_e\ell_e}{2}\int_{-1}^{1}\mathbf{N}^{\mathrm{T}}d\xi = 0 \tag{10.22}$$

$$-\psi_L hT_L + \psi_L hT_\infty - \boldsymbol{\psi}^{\mathrm{T}}\mathbf{K}_T\mathbf{T} + \boldsymbol{\psi}^{\mathrm{T}}\mathbf{R} = 0 \tag{10.23}$$

which should be satisfied for all $\boldsymbol{\Psi}$ with $\Psi_1 = 0$. The global matrices $\mathbf{K}_T$ and $\mathbf{R}$ are assembled from element matrices $\mathbf{k}_T$ and $\mathbf{r}_Q$ as given in

$$\mathbf{k}_T = \frac{k_e}{\ell_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{10.24}$$

$$\mathbf{r}_Q = \frac{Q_e \ell_e}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{10.25}$$

When each $\boldsymbol{\Psi}$ is chosen in turn as $[0 \quad 1 \quad 0 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}}$, $[0 \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}}$, $\ldots$, $[0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad 1]^{\mathrm{T}}$ and since $T_1 = T_0$, then Eq. 10.23 yields

$$\begin{bmatrix} K_{22} & K_{23} & \cdots & K_{2L} \\ K_{32} & K_{33} & \cdots & K_{3L} \\ \vdots & & & \\ K_{L2} & K_{L3} & \cdots & (K_{LL} + h) \end{bmatrix} \begin{Bmatrix} T_2 \\ T_3 \\ \vdots \\ T_L \end{Bmatrix} = \begin{Bmatrix} R_2 \\ R_3 \\ \vdots \\ (R_L + hT_\infty) \end{Bmatrix} - \begin{Bmatrix} K_{21}T_0 \\ K_{31}T_0 \\ \vdots \\ K_{L1}T_0 \end{Bmatrix} \tag{10.26}$$

We observe that Eq. 10.26 can be solved for $T_2, T_3, \ldots, T_L$. We thus note that the Galerkin approach naturally leads to the elimination approach for handling nonzero specified temperature $T = T_0$ at node 1. However, it is also possible to develop Galerkin's method with a penalty approach to handle $T_1 = T_0$. In this case, the equations are as given by

$$\begin{bmatrix} (K_{11} + C) & K_{12} & \cdots & K_{1L} \\ K_{21} & K_{22} & \cdots & K_{2L} \\ \vdots & & & \vdots \\ K_{L1} & K_{L2} & \cdots & (K_{LL} + h) \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ \vdots \\ T_L \end{Bmatrix} = \begin{Bmatrix} (R_1 + CT_0) \\ R_2 \\ \vdots \\ (R_L + hT_\infty) \end{Bmatrix} \tag{10.27}$$

Example 10.1

A composite wall consists of three materials, as shown in Fig. E10.1a. The outer temperature is $T_0 = 20$°C. Convection heat transfer takes place on the inner surface of the wall with $T_\infty = 800$°C and $h = 25\ \mathrm{W/m^2 \cdot {}^\circ C}$. Determine the temperature distribution in the wall.

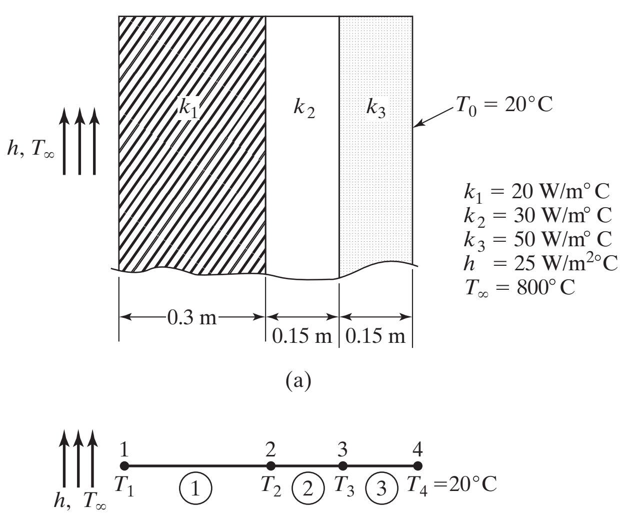


FIGURE E10.1

Solution A three-element finite element model of the wall is shown in Fig. E10.1b. The element conductivity matrices are

$$\mathbf{k}_T^{(1)} = \frac{20}{0.3}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \mathbf{k}_T^{(2)} = \frac{30}{0.15}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\mathbf{k}_T^{(3)} = \frac{50}{0.15}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

The global $\mathbf{K} = \Sigma \mathbf{k}_T$ is obtained from these matrices as

$$\mathbf{K} = 66.7\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 4 & -3 & 0 \\ 0 & -3 & 8 & -5 \\ 0 & 0 & -5 & 5 \end{bmatrix}$$

Now, since convection occurs at node 1, the constant $h = 25$ is added to the $(1, 1)$ location of $\mathbf{K}$. This results in

$$\mathbf{K} = 66.7\begin{bmatrix} 1.375 & -1 & 0 & 0 \\ -1 & 4 & -3 & 0 \\ 0 & -3 & 8 & -5 \\ 0 & 0 & -5 & 5 \end{bmatrix}$$

Since no heat generation Q occurs in this problem, the heat rate vector $\mathbf{R}$ consists only of hT_∞ in the first row. That is,

$$\mathbf{R} = [25 \times 800 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$

The specified temperature boundary condition $T_4 = 20$°C, will now be handled by the penalty approach. We choose C based on

$$\begin{aligned} C &= \max |\mathbf{K}_{ij}| \times 10^4 \\ &= 66.7 \times 8 \times 10^4 \end{aligned}$$

Now, C gets added to $(4, 4)$ location of $\mathbf{K}$, while CT_4 is added to the fourth row of $\mathbf{R}$. The resulting equations are

$$66.7\begin{bmatrix} 1.375 & -1 & 0 & 0 \\ -1 & 4 & -3 & 0 \\ 0 & -3 & 8 & -5 \\ 0 & 0 & -5 & 80\,005 \end{bmatrix}\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 25 \times 800 \\ 0 \\ 0 \\ 10\,672 \times 10^4 \end{Bmatrix}$$

The solution is

$$\mathbf{T} = [304.6 \quad 119.0 \quad 57.1 \quad 20.0]^{\mathrm{T}}\ ^\circ\mathrm{C}$$

Comment: The boundary condition $T_4 = 20°C$ can also be handled by the elimination approach. The fourth row and column of **K** is deleted, and **R** is modified according to Eq. 3.70. The resulting equations are

$$66.7\begin{bmatrix} 1.375 & -1 & 0 \\ -1 & 4 & -3 \\ 0 & -3 & 8 \end{bmatrix}\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} 25 \times 800 \\ 0 \\ 0 + 6670 \end{bmatrix}$$

which yields

$$[T_1 \quad T_2 \quad T_3]^{\mathrm{T}} = [304.6_1 \quad 119.0 \quad 57.1]^{\mathrm{T}}\,°\mathrm{C}$$ ■

Heat Flux Boundary Condition. Certain physical situations are modeled using the boundary condition

$$q = q_0 \qquad \text{at } x = 0 \tag{10.28}$$

where q_0 is a specified heat flux on the boundary. If $q = 0$, then the surface is perfectly insulated. A nonzero value of q_0 occurs, for example, due to an electrical heater or pad where one face is in contact with the wall and the other face is insulated. It is important to note that the input heat flux q_0 has a sign convention associated with it: q_0 is input as a positive value if heat is flowing out of the body and as a negative value if heat is flowing into the body. The boundary condition in Eq. 10.28 is handled by adding $(-q_0)$ to the heat rate vector. The resulting equations are

$$\mathbf{KT} = \mathbf{R} + \begin{Bmatrix} -q_0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \tag{10.29}$$

The sign convention for specified heat flux given in Eq. 10.29 is clear if we consider the heat transfer occurring at a boundary. Let n be the outward normal (in one-dimensional problems, $n = +x$ or $-x$). The heat flow in the body towards the $+n$ direction is $q = -k\,\partial T/\partial n$, where $\partial T/\partial n < 0$. Thus, $q > 0$ and since this heat flows out of the body, we have the boundary condition $q = q_0$ with the stated sign convention.

Comment on Forced and Natural Boundary Conditions. In this problem, boundary conditions of the type $T = T_0$, which is on the field variable itself, are called *forced* boundary conditions. On the other hand, the boundary condition $q|_{x=0} = q_0$, or equivalently, $-k\,dT/dx_{x=0} = q_0$ is called a *natural* boundary condition involving the derivative of the field variable. Further, it is evident from Eq. 10.29 that the homogeneous natural boundary condition $q = q_0 = 0$ does not require any modifications in the element matrices. These are automatically satisfied at the boundary, in an average sense.

Example 10.2

Heat is generated in a large plate ($k = 0.8$ W/m · °C) at the rate of 4000 W/m^3. The plate is 25 cm thick. The outside surfaces of the plate are exposed to ambient air at 30°C with a convective heat-transfer coefficient of 20 $\mathrm{W/m^2 \cdot {}^\circ C}$. Determine the temperature distribution in the wall.

Solution The problem is symmetric about the centerline of the plate. A two-element finite element model is shown in Fig. E10.2. The left end is insulated ($q = 0$) because no heat can flow across a line of symmetry. Noting that $k/\ell = 0.8/.0625 = 12.8$, we get

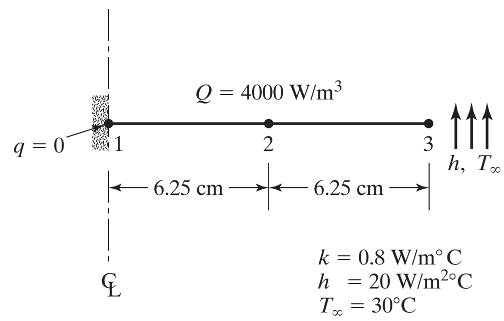


FIGURE E10.2

$$\mathbf{K} = \begin{bmatrix} 12.8 & -12.8 & 0 \\ -12.8 & 25.6 & -12.8 \\ 0 & -12.8 & (12.8 + 20) \end{bmatrix}$$

The heat rate vector is assembled from the heat source (Eq. 10.25) as well as due to convection as

$$\mathbf{R} = [125 \quad 250 \quad (125 + 20 \times 30)]^{\mathrm{T}}$$

Solution of $\mathbf{KT} = \mathbf{R}$ yields

$$[T_1 \quad T_2 \quad T_3] = [94.0 \quad 84.3 \quad 55.0]^\circ\mathrm{C}$$

In concluding one-dimensional heat conduction, we note that all element matrices described earlier were derived using Galerkin's approach. It is also possible to derive these matrices using an energy approach based on minimizing the functional

$$\Pi_T = \int_0^L \frac{1}{2}k\left(\frac{dT}{dx}\right)^2 dx - \int_0^L QT\,dx + \frac{1}{2}h(T_L - T_\infty)^2 \qquad (10.30) \quad \blacksquare$$

One-Dimensional Heat Transfer in Thin Fins

A fin is an extended surface that is added onto a structure to increase the rate of heat removal. A familiar example is in the motorcycle where fins extend from the cylinder head to quickly dissipate heat through convection. We present here the finite element method for analyzing heat transfer in thin rectangular fins (Fig. 10.5). This problem differs from the conduction problem discussed previously in which both conduction and convection occur within the body.

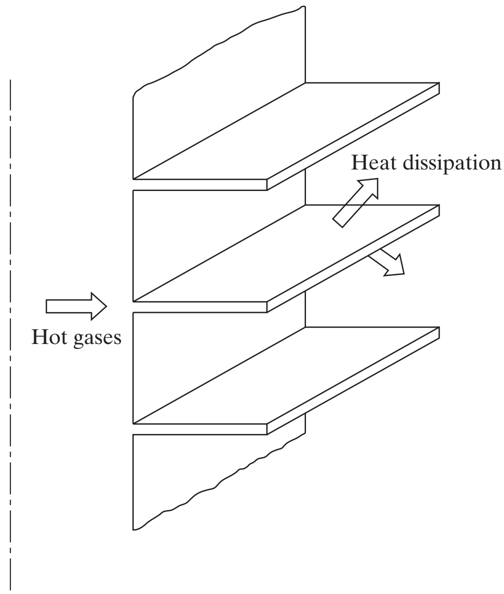


FIGURE 10.5 An array of thin rectangular fins.

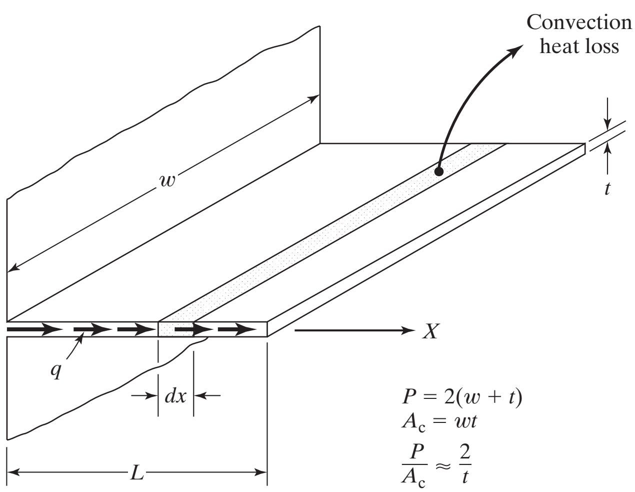


FIGURE 10.6 Heat flow in a thin rectangular fin.

Consider a thin rectangular fin as shown in Fig. 10.6. The problem can be treated as one-dimensional because the temperature gradients along the width and across the thickness are negligible. The governing equation may be derived from the conduction equation with heat source, given by

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) + Q = 0$$

The convection heat loss in the fin can be considered as a negative heat source

$$
\begin{aligned}
Q &= -\frac{(P\,dx)h(T - T_\infty)}{A_c\,dx} \\
&= -\frac{Ph}{A_c}(T - T_\infty)
\end{aligned}
\tag{10.31}
$$

where P is the perimeter of fin, and A_c is the area of cross section. Thus, the governing equation is

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) - \frac{Ph}{A_c}(T - T_\infty) = 0 \tag{10.32}$$

We present our analysis for the case when the base of the fin is held at T_0 and the tip of the fin is insulated (heat going out of the tip is negligible). The boundary conditions are then given by

$$T = T_0 \qquad \text{at } x = 0 \tag{10.33a}$$

$$q = 0 \qquad \text{at } x = L \tag{10.33b}$$

The Finite Element Method: Galerkin Approach. The element matrices and heat-rate vectors for solving Eq. 10.32 with the boundary conditions in Eq. 10.33 will now be developed. Galerkin's approach is attractive since we do not have to set up the functional that is to be minimized. Element matrices can be derived directly from the differential equation. Let $\phi(x)$ be any function satisfying $\phi(0) = 0$ using the same basis as T. We require that

$$\int_0^L \phi\left[\frac{d}{dx}\left(k\frac{dT}{dx}\right) - \frac{Ph}{A_c}(T - T_\infty)\right]dx = 0 \tag{10.34}$$

Integrating the first term by parts, we get

$$\phi k\frac{dT}{dx}\bigg|_0^L - \int_0^L k\frac{d\phi}{dx}\frac{dT}{dx}\,dx - \frac{Ph}{A_c}\int_0^L \phi T\,dx + \frac{Ph}{A_c}T_\infty\int_0^L \phi\,dx = 0 \tag{10.35}$$

Using $\phi(0) = 0$, $k(L)[dT(L)/dx] = 0$, and the isoparametric relations

$$dx = \frac{\ell_e}{2}d\xi \qquad T = \mathbf{N}\mathbf{T}^e \qquad \phi = \mathbf{N}\boldsymbol{\psi} \qquad \frac{dT}{dx} = \mathbf{B}_T\mathbf{T}^e \qquad \frac{d\phi}{dx} = \mathbf{B}_T\boldsymbol{\psi}$$

we get

$$
\begin{aligned}
&-\sum_e \boldsymbol{\psi}^{\mathrm{T}}\left[\frac{k_e\ell_e}{2}\int_{-1}^{1}\mathbf{B}_T^{\mathrm{T}}\mathbf{B}_T\,d\xi\right]\mathbf{T}^e - \frac{Ph}{A_c}\sum_e \boldsymbol{\psi}^{\mathrm{T}}\frac{\ell_e}{2}\int_{-1}^{1}\mathbf{N}^T\mathbf{N}\,d\xi\,\mathbf{T}^e \\
&+ \frac{PhT_\infty}{A_c}\sum_e \psi^{\mathrm{T}}\frac{\ell_e}{2}\int_{-1}^{1}\mathbf{N}^{\mathrm{T}}d\xi = 0
\end{aligned}
\tag{10.36}
$$

We define

$$\mathbf{h}_T = \frac{Ph}{A_c}\frac{\ell_e}{2}\int_{-1}^{1}\mathbf{N}^{\mathrm{T}}\mathbf{N}\,d\xi = \frac{Ph}{A_c}\frac{\ell_e}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} \tag{10.37a}$$

or, since $P/A_c \approx 2/t$ (Fig. 10.6),

$$\mathbf{h}_T \approx \frac{h\ell_e}{3t}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} \tag{10.37b}$$

and

$$\mathbf{r}_\infty = \frac{Ph}{A_c}T_\infty\frac{\ell_e}{2}\int_{-1}^{1}\mathbf{N}^{\mathrm{T}}d\xi = \frac{PhT_\infty}{A_c}\frac{\ell_e}{2}\begin{Bmatrix}1\\ 1\end{Bmatrix} \tag{10.38a}$$

or

$$\mathbf{r}_\infty \approx \frac{hT_\infty\ell_e}{t}\begin{Bmatrix}1\\ 1\end{Bmatrix} \tag{10.38b}$$

Equation 10.36 reduces to

$$-\sum_e \boldsymbol{\psi}^T(\mathbf{k}_T + \mathbf{h}_T)\mathbf{T}^e + \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{r}_\infty = 0 \tag{10.39}$$

or

$$-\boldsymbol{\psi}^T(\mathbf{K}_T + \mathbf{H}_T)\mathbf{T} + \boldsymbol{\psi}^{\mathrm{T}}\mathbf{R}_\infty = 0$$

which should hold for all $\boldsymbol{\Psi}$ satisfying $\Psi_1 = 0$.

Denoting $K_{ij} = (K_T + H_T)_{ij}$, we obtain

$$\begin{bmatrix}K_{22} & K_{23} & \cdots & K_{2L}\\ K_{32} & K_{33} & \cdots & K_{3L}\\ \vdots & \vdots & & \vdots\\ K_{L2} & K_{L3} & \cdots & K_{LL}\end{bmatrix}\begin{Bmatrix}T_2\\ T_3\\ \vdots\\ T_L\end{Bmatrix} = \begin{Bmatrix}\\ \mathbf{R}_\infty\\ \\ \end{Bmatrix} - \begin{Bmatrix}K_{21}T_0\\ K_{31}T_0\\ \vdots\\ K_{L1}T_0\end{Bmatrix} \tag{10.40}$$

which can be solved for $\mathbf{T}$. These equations incorporate the elimination approach for handling the boundary condition $T = T_0$. Other types of boundary conditions as discussed for heat conduction can also be considered for fin problems.

Example 10.3

A metallic fin, with thermal conductivity $k = 360\ \mathrm{W/m\cdot{}^\circ C}$, 0.1 cm thick, and 10 cm long, extends from a plane wall whose temperature is 235°C. Determine the temperature distribution and amount of heat transferred from the fin to the air at 20°C with $h = 9\ \mathrm{W/m^2\cdot{}^\circ C}$. Take the width of fin to be 1 m.

Solution Assume that the tip of the fin is insulated. Using a three-element finite element model (Fig. E10.3) and assembling $\mathbf{K}_T, \mathbf{H}_T, \mathbf{R}_\infty$ as given previously, Eq. 10.40 yields

$$\left[\frac{360}{3.33\times10^{-2}}\begin{bmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 1\end{bmatrix} + \frac{9\times3.33\times10^{-2}}{3\times10^{-3}}\begin{bmatrix}4 & 1 & 0\\ 1 & 4 & 1\\ 0 & 1 & 2\end{bmatrix}\right]\begin{Bmatrix}T_2\\ T_3\\ T_4\end{Bmatrix}$$

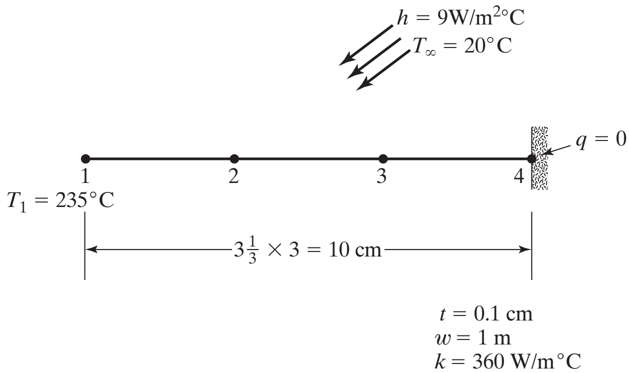


FIGURE E10.3

$$= \frac{9 \times 20 \times 3.33 \times 10^{-2}}{10^{-3}} \begin{Bmatrix} 2 \\ 2 \\ 1 \end{Bmatrix} - \begin{Bmatrix} -10711 \times 235 \\ 0 \\ 0 \end{Bmatrix}$$

The solution is

$$[T_2 \quad T_3 \quad T_4] = [209.8 \quad 195.2 \quad 190.5]°\text{C}$$

The total heat loss in the fin can now be computed as

$$H = \sum_e H_e$$

The loss H_e in each element is

$$H_e = h(T_{\text{av}} - T_\infty)A_S$$

where $A_s = 2 \times (1 \times 0.0333)\ \text{m}^2$, and T_{av} is the average temperature within the element. We obtain

$$H_{\text{loss}} = 334.3\ \text{W/m}$$ ■

Two-Dimensional Steady-State Heat Conduction

Our objective here is to determine the temperature distribution $T(x, y)$ in a long, prismatic solid in which two-dimensional conduction effects are important. An example is a chimney of rectangular cross section, as shown in Fig. 10.7. Once the temperature distribution is known, the heat flux can be determined from Fourier's law.

Differential Equation. Consider a differential control volume in the body, as shown in Fig. 10.8. The control volume has a constant thickness τ in the z direction. The heat generation Q is denoted by Q (W/m^3). Since the heat rate (= heat flux × area) entering the control volume plus the heat rate generated equals the heat rate coming out, we have (Fig. 10.8)

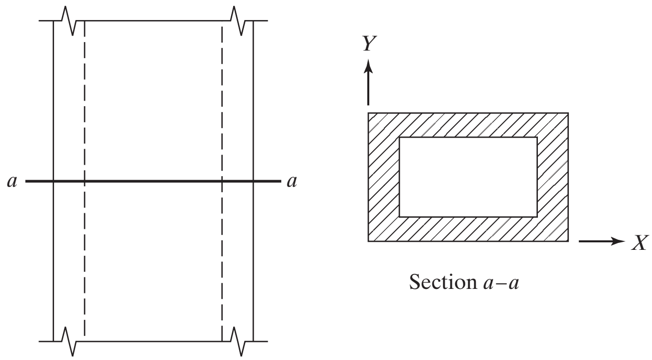


FIGURE 10.7 Two-dimensional model for heat conduction in a chimney.

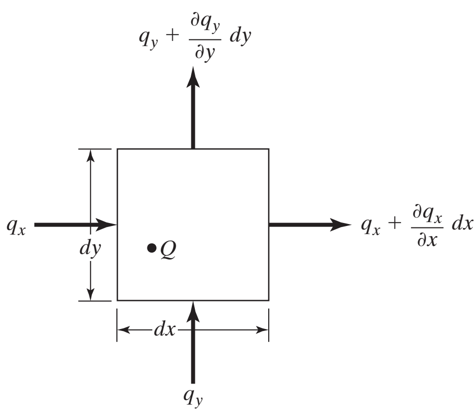


FIGURE 10.8 A differential control volume for heat transfer.

$$q_x\,dy\,\tau + q_y\,dx\,\tau + Q\,dx\,dy\,\tau = \left(q_x + \frac{\partial q_x}{\partial x}dx\right)dy\,\tau + \left(q_y + \frac{\partial q_y}{\partial y}dy\right)dx\,\tau \qquad (10.41)$$

or, upon canceling terms,

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} - Q = 0 \qquad (10.42)$$

Substituting $q_x = -k\partial T/\partial x$ and $q_y = -k\partial T/\partial y$ into Eq. 10.42, we get the **heat-diffusion equation**:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + Q = 0 \qquad (10.43)$$

We note that this partial differential equation is a special case of the Helmholtz equation given in Eq. 10.1.

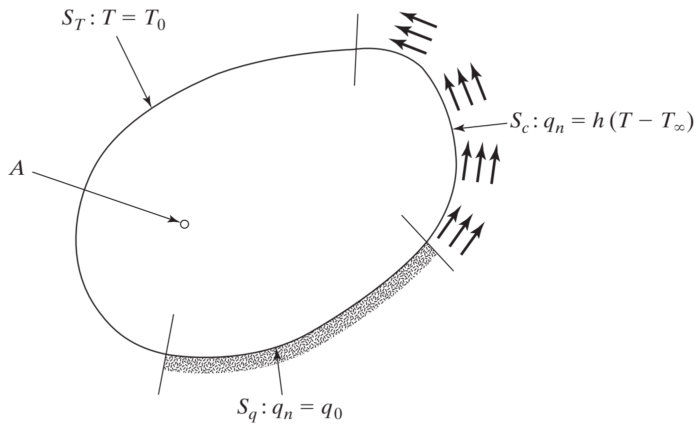


FIGURE 10.9 Boundary conditions for two-dimensional heat conduction.

Boundary Conditions. The governing equation, Eq. 10.43, has to be solved together with certain boundary conditions. These boundary conditions are of three types, as shown in Fig. 10.9: (1) specified temperature $T = T_0$ on S_T, (2) specified heat flux $q_n = q_0$ on S_q, and (3) convection $q_n = h(T - T_\infty)$ on S_c. The interior of the body is denoted by A, and the boundary is denoted as $S = (S_T + S_q + S_c)$. Further, q_n is the heat flux normal to the boundary. The sign convention adopted here for specifying q_0 is that $q_0 > 0$ if heat is flowing out of the body, while $q_0 < 0$ if heat is flowing into the body.

The Triangular Element. The triangular element (Fig. 10.10) will be used to solve the heat-conduction problem. Extension to quadrilateral or other isoparametric elements follows in a similar manner as discussed earlier for stress analysis.

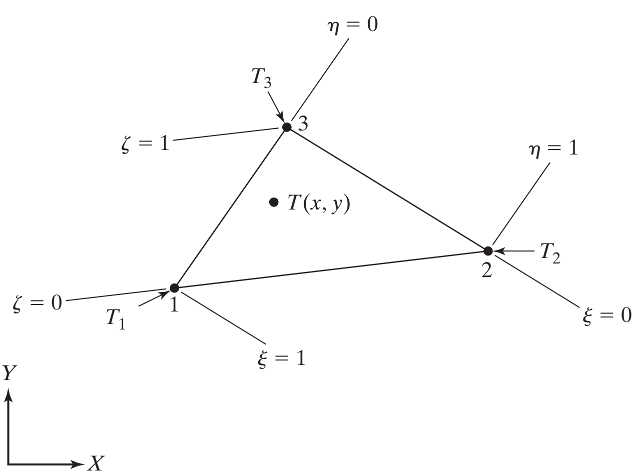


FIGURE 10.10 The linear triangular element for scalar field problems.

Consider a constant length of the body perpendicular to the x, y plane. The temperature field within an element is given by

$$T = N_1T_1 + N_2T_2 + N_3T_3$$

or

$$T = \mathbf{NT}^e \tag{10.44}$$

where $\mathbf{N} = [\xi \quad \eta \quad 1 - \xi - \eta]$ are the element shape functions and $\mathbf{T}^e = [T_1 \quad T_2 \quad T_3]^{\mathrm{T}}$. Referring to Chapter 5, we also have

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + N_3x_3 \\ y &= N_1y_1 + N_2y_2 + N_3y_3 \end{aligned} \tag{10.45}$$

Further, the chain rule of differentiation yields

$$\begin{aligned} \frac{\partial T}{\partial \xi} &= \frac{\partial T}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \xi} \\ \frac{\partial T}{\partial \eta} &= \frac{\partial T}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \eta} \end{aligned} \tag{10.46}$$

or

$$\begin{Bmatrix} \dfrac{\partial T}{\partial \xi} \\ \dfrac{\partial T}{\partial \eta} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix} \tag{10.47}$$

In Eq. 10.47, $\mathbf{J}$ is the Jacobian matrix given by

$$\mathbf{J} = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix} \tag{10.48}$$

where $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$, and $|\det \mathbf{J}| = 2A_e$, where A_e is the area of the triangle. Equation 10.47 yields

$$\begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \dfrac{\partial T}{\partial \xi} \\ \dfrac{\partial T}{\partial \eta} \end{Bmatrix} \tag{10.49a}$$

$$= \frac{1}{\det \mathbf{J}} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \mathbf{T}^e \tag{10.49b}$$

which can be written as

$$\begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix} = \mathbf{B}_T\mathbf{T}^e \tag{10.50}$$

where

$$\mathbf{B}_T = \frac{1}{\det \mathbf{J}}\begin{bmatrix} y_{23} & -y_{13} & (y_{13} - y_{23}) \\ -x_{23} & x_{13} & (x_{23} - x_{13}) \end{bmatrix} \tag{10.51a}$$

$$= \frac{1}{\det \mathbf{J}}\begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \tag{10.51b}$$

Galerkin Approach.[1] Consider the heat-conduction problem

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + Q = 0 \tag{10.52}$$

with the boundary conditions

$$T = T_0 \quad \text{on} \quad S_T \qquad q_n = q_0 \quad \text{on} \quad S_q \qquad q_n = h(T - T_\infty) \quad \text{on} \quad S_c \tag{10.53}$$

In Galerkin's approach, we seek an approximate solution T such that

$$\int_A\int \phi\left[\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)\right]dA + \int_A\int \phi Q\,dA = 0 \tag{10.54}$$

for every $\phi(x, y)$ constructed from the same basis functions as those used for T and satisfying $\phi = 0$ on S_T. Noting that

$$\phi\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) = \frac{\partial}{\partial x}\left(\phi k\frac{\partial T}{\partial x}\right) - k\frac{\partial \phi}{\partial x}\frac{\partial T}{\partial x}$$

we find that Eq. 10.54 gives

$$\int_A\int \left\{\left[\frac{\partial}{\partial x}\left(\phi k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\phi k\frac{\partial T}{\partial y}\right)\right] - \left[k\frac{\partial \phi}{\partial x}\frac{\partial T}{\partial x} + k\frac{\partial \phi}{\partial y}\frac{\partial T}{\partial y}\right]\right\}dA$$
$$+ \int_A\int \phi Q\,dA = 0 \tag{10.55}$$

From the notation $q_x = -k(\partial T/\partial x)$ and $q_y = -k(\partial T/\partial y)$, and the divergence theorem, the first term in Eq. 10.55 is

$$-\int_A\int\left[\frac{\partial}{\partial x}(\phi q_x) + \frac{\partial}{\partial y}(\phi q_y)\right]dA = -\int_S \phi[q_x n_x + q_y n_y]\,dS$$
$$= -\int_S \phi q_n\,dS \tag{10.56}$$

where n_x and n_y are the direction cosines of the unit normal $\mathbf{n}$ to the boundary, and $q_n = q_x n_x + q_y n_y = \mathbf{q} \cdot \mathbf{n}$ is the normal heat flow along the unit outward normal, which

[1] The functional approach would be based on minimizing

$$\pi_T = \frac{1}{2}\int_A\left[k\left(\frac{\partial T}{\partial x}\right)^2 + k\left(\frac{\partial T}{\partial y}\right)^2 - 2QT\right]dA + \int_{S_q} q_o T\,dS + \int_{S_c} h(T - T_\infty)^2\,dS$$

is specified by boundary conditions. Since $S = S_T + S_q + S_c$, $\phi = 0$ on S_T, $q_n = q_0$ on S_q, and $q_n = h(T - T_\infty)$ on S_c, Eq. 10.55 reduces to

$$-\int_{S_q} \phi q_0 \, dS - \int_{S_c} \phi h(T - T_\infty) \, dS - \int_A \int \left(k \frac{\partial \phi}{\partial x} \frac{\partial T}{\partial x} + k \frac{\partial \phi}{\partial y} \frac{\partial T}{\partial y} \right) dA + \int_A \int \phi Q \, dA = 0 \tag{10.57}$$

Now, we introduce the isoparametric relations for the triangular element such as $T = \mathbf{N}\mathbf{T}^e$ given in Eqs. 10.47 through 10.55. Further, we denote the global virtual-temperature vector as $\mathbf{\Psi}$ whose dimension equals number of nodes in the finite element model. The virtual temperature distribution within each element is interpolated as

$$\phi = \mathbf{N}\boldsymbol{\psi} \tag{10.58a}$$

Moreover, just as $[\partial T/\partial x \quad \partial T/\partial y]^T = \mathbf{B}_T \mathbf{T}^e$, we have

$$\left[\frac{\partial \phi}{\partial x} \; \frac{\partial \phi}{\partial y} \right]^{\mathrm{T}} = \mathbf{B}_T \boldsymbol{\psi} \tag{10.58b}$$

Now, consider the first term in Eq. 10.57:

$$\int_{S_q} \phi q_0 \, dS = \sum_e \boldsymbol{\psi}^{\mathrm{T}} q_0 \int_{S_q^e} \mathbf{N}^{\mathrm{T}} dS \tag{10.59}$$

If edge 2–3 is on the boundary (Fig. 10.11), we get $\mathbf{N} = [0 \quad \eta \quad 1 - \eta]$ $dS = \ell_{2-3}\, d\eta$, and it follows that

$$\int_{S_q} \phi q_0 \, dS = \sum_e \boldsymbol{\psi}^{\mathrm{T}} q_0 \ell_{2-3} \int_0^1 \mathbf{N}^{\mathrm{T}} d\eta \tag{10.60a}$$

$$= \sum_e \boldsymbol{\psi}^{\mathrm{T}} \mathbf{r}_q \tag{10.60b}$$

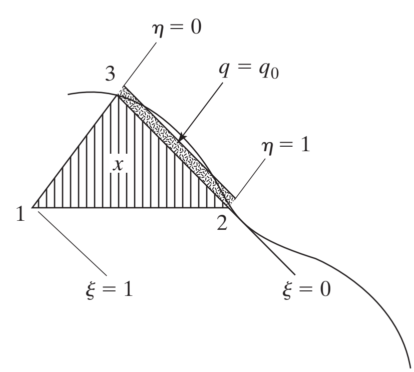


FIGURE 10.11 Specified heat flux boundary conduction on edge 2–3 of a triangular element.

where

$$\mathbf{r}_q = \frac{q_0\ell_{2-3}}{2}[0 \quad 1 \quad 1]^{\mathrm{T}} \tag{10.61}$$

Next, consider

$$\int_{S_c} \phi h(T - T_\infty)dS = \int_{S_c} \phi hT\,dS - \int_{S_c} \phi hT_\infty\,dS \tag{10.62a}$$

If edge 2–3 is the convection edge of the element, then

$$\begin{aligned}\int_{S_c} \phi h(T - T_\infty)dS &= \sum_e \psi^{\mathrm{T}}\left[h\ell_{2-3}\int_0^1 \mathbf{N}^{\mathrm{T}}\mathbf{N}\,d\eta\right]\mathbf{T}^e - \sum_e \psi^{\mathrm{T}} hT_\infty \ell_{2-3}\int_0^1 \mathbf{N}^{\mathrm{T}}\,d\eta \\ &= \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{h}_T\mathbf{T}^e - \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{r}_\infty \end{aligned} \tag{10.62b}$$

Substituting for $\mathbf{N} = [0, \quad \eta, \quad 1 - \eta]$, we get

$$\mathbf{h}_T = \frac{h\ell_{2-3}}{6}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \tag{10.63}$$

$$\mathbf{r}_\infty = \frac{hT_\infty\ell_{2-3}}{2}[0 \quad 1 \quad 1]^{\mathrm{T}} \tag{10.64}$$

Next,

$$\int_A \int k\left(\frac{\partial\phi}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial\phi}{\partial y}\frac{\partial T}{\partial y}\right)dA = \int_A \int k\left[\frac{\partial\phi}{\partial x} \; \frac{\partial\phi}{\partial y}\right]\begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix} dA \tag{10.65a}$$

$$= \sum_e \boldsymbol{\psi}^{\mathrm{T}}\left[k_e\int_e \mathbf{B}_T^{\mathrm{T}}\mathbf{B}_T\,dA\right]\mathbf{T}^e \tag{10.65b}$$

$$= \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{k}_T\mathbf{T}^e \tag{10.65c}$$

where

$$\mathbf{k}_T = k_e A_e \mathbf{B}_T^{\mathrm{T}}\mathbf{B}_T \tag{10.66}$$

Finally, if $Q = Q_e$ is constant within the element,

$$\int_A \int \phi Q\,dA = \sum_e \boldsymbol{\psi}^{\mathrm{T}} Q_e \int_e \mathbf{N}\,dA = \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{r}_Q$$

where

$$\mathbf{r}_Q = \frac{Q_e A_e}{3}[1 \quad 1 \quad 1]^{\mathrm{T}} \tag{10.67}$$

Other distributions of Q within the element are considered in the exercises at the end of this chapter. Thus, Eq. 10.57 is of the form

$$-\sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{r}_q - \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{h}_T\mathbf{T}^e + \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{r}_\infty - \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{k}_T\mathbf{T}^e + \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{r}_Q = 0 \tag{10.68}$$

or

$$\boldsymbol{\psi}^{\mathrm{T}}(\mathbf{R}_\infty - \mathbf{R}_q + \mathbf{R}_Q) - \boldsymbol{\psi}^{\mathrm{T}}(\mathbf{H}_T + \mathbf{K}_T)\mathbf{T} = 0 \tag{10.69}$$

which is to hold for all $\boldsymbol{\Psi}$ satisfying $\boldsymbol{\Psi} = \mathbf{0}$ at nodes on S_T. We thus obtain

$$\mathbf{K}^E\,\mathbf{T}^E = \mathbf{R}^E \tag{10.70}$$

where $\mathbf{K} = \sum_e(\mathbf{k}_T + \mathbf{h}_T)$, $\mathbf{R} = \sum_e(\mathbf{r}_\infty - \mathbf{r}_q + \mathbf{r}_Q)$, and superscript E represents the familiar modifications made to $\mathbf{K}$ and $\mathbf{R}$ to handle $T = T_0$ on S_T by the elimination approach. Alternatively, the Penalty approach can also be used to handle $T = T_0$.

Example 10.4

A long bar of rectangular cross section, having thermal conductivity of 1.5 W/m·°C, is subjected to the boundary conditions shown in Fig. E10.4a. Two opposite sides are maintained at a uniform temperature of 180°C; one side is insulated, and the remaining side is subjected to a convection process with $T_\infty = 25$°C and $h = 50$ W/m^2·°C. Determine the temperature distribution in the bar.

Solution A five-node, three-element finite element model of the problem is shown in Fig. E10.4b, where symmetry about the horizontal axis is used. Note that the line of symmetry is shown as insulated, since no heat can flow across it.

The element matrices are developed as follows. The element connectivity is defined as in the following table:

Element	1	2	3	← local
1	1	2	3	↑
2	5	1	3	Global
3	5	4	3	↓

We have

$$\mathbf{B}_T = \frac{1}{\det \mathbf{J}}\begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix}$$

For each element,

$$\mathbf{B}_T^{(1)} = \frac{1}{0.06}\begin{bmatrix} -0.15 & 0.15 & 0 \\ 0 & -0.4 & 0.4 \end{bmatrix}$$

$$\mathbf{B}_T^{(2)} = \frac{1}{0.12}\begin{bmatrix} -0.15 & -0.15 & 0.3 \\ 0.4 & -0.4 & 0 \end{bmatrix}$$

$$\mathbf{B}_T^{(3)} = \frac{-1}{0.06}\begin{bmatrix} 0.15 & -0.15 & 0 \\ 0 & -0.4 & 0.4 \end{bmatrix}$$

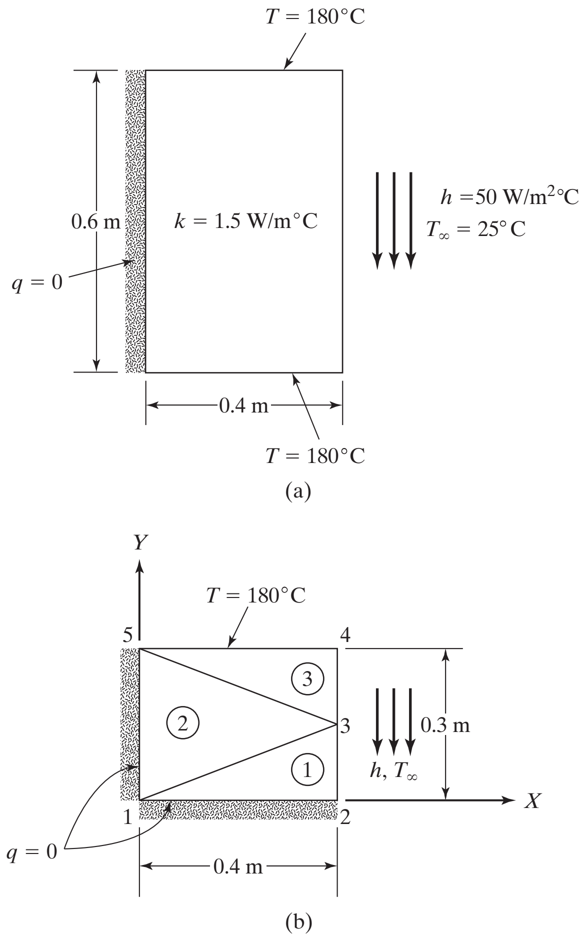


FIGURE E10.4

Then, $\mathbf{k}_T = kA_e\mathbf{B}_T^{\mathrm{T}}\mathbf{B}_T$ yields

$$\mathbf{k}_T^{(1)} = (1.5)(0.03)\mathbf{B}_T^{(1)\mathrm{T}}\mathbf{B}_T^{(1)}$$

$$= \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0.28125 & -0.28125 & 0 \\ -0.28125 & 2.28125 & -2.0 \\ 0 & -2.0 & 2.0 \end{bmatrix} \end{array}$$

$$\mathbf{k}_T^{(2)} = \begin{array}{c} \begin{array}{ccc} 5 & 1 & 3 \end{array} \\ \begin{bmatrix} 1.14 & -0.86 & -0.28125 \\ -0.86 & 1.14 & -0.28125 \\ -0.28125 & -0.28125 & 0.5625 \end{bmatrix} \end{array}$$

$$\mathbf{k}_T^{(3)} = \begin{array}{c} \begin{array}{ccc} 5 & 4 & 3 \end{array} \\ \begin{bmatrix} 0.28125 & -0.28125 & 0 \\ -0.28125 & 2.28125 & -2.0 \\ 0 & -2.0 & 2.0 \end{bmatrix} \end{array}$$

Now the matrices $\mathbf{h}_T$ for elements with convection edges are developed. Since both elements 1 and 3 have edges 2–3 (in local node numbers) as convection edges, the formula

$$\mathbf{h}_T = \frac{h\ell_{2-3}}{6}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

can be used, resulting in

$$\mathbf{h}_T^{(1)} = \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.5 & 1.25 \\ 0 & 1.25 & 2.5 \end{bmatrix} \end{array} \qquad \mathbf{h}_T^{(2)} = \begin{array}{c} \begin{array}{ccc} 5 & 4 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.5 & 1.25 \\ 0 & 1.25 & 2.5 \end{bmatrix} \end{array}$$

The matrix $\mathbf{K} = \sum(\mathbf{k}_T + \mathbf{h}_T)$ is now assembled. The elimination approach for handling the boundary conditions $T = 180°\text{C}$ at nodes 4 and 5 results in the elimination of these rows and columns. However, these fourth and fifth rows are used subsequently for modifying the $\mathbf{R}$ vector. The result is

$$\mathbf{K} = \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1.42125 & -0.28125 & -0.28125 \\ -0.28125 & 4.78125 & -0.75 \\ -0.28125 & -0.75 & 9.5625 \end{bmatrix} \end{array}$$

Now the heat-rate vector $\mathbf{R}$ is assembled from element convection contributions. The formula

$$\mathbf{r}_\infty = \frac{hT_\infty \ell_{2-3}}{2}[0 \quad 1 \quad 1]$$

results in

$$\mathbf{r}_\infty^{(1)} = \frac{(50)(25)(0.15)}{2} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ [0 \quad 1 \quad 1] \end{array}$$

and

$$\mathbf{r}_\infty^{(3)} = \frac{(50)(25)(0.15)}{2} \begin{array}{c} \begin{array}{ccc} 5 & 4 & 3 \end{array} \\ [0 \quad 1 \quad 1] \end{array}$$

Thus,

$$\mathbf{R} = 93.75 \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ [0 \quad 1 \quad 2]^{\mathrm{T}} \end{array}$$

In the elimination approach, $\mathbf{R}$ gets modified according to Eq. 3.70. Solution of $\mathbf{KT} = \mathbf{R}$ then yields

$$[T_1 \quad T_2 \quad T_3] = [124.5 \quad 34.0 \quad 45.4]°\text{C}$$

Note: A large temperature gradient exists along the line connecting nodes 2 and 4. This is because node 4 is maintained at 180°C, while node 2 has a temperature close to the ambient temperature of $T_\infty = 25°C$ because of the relatively large value of h. This fact implies that our finite element model should capture this large temperature gradient by having sufficient number of nodes along line 2–4. In fact, a model with only two nodes (as opposed to three as used here) will lead to an incorrect solution for the temperatures. Also, with the three-element model considered here, heat-flow values (see computer output) are not accurate. A more detailed model is necessary. ■

It is also noted that a thermal-stress analysis can now be performed once the temperature distribution is known, as discussed in Chapter 5.

Two-Dimensional Fins

In Fig. 10.12a, a thin plate is receiving heat from a pipe and then dissipating it to the surrounding media (air) by convection. We may assume that the temperature gradients are negligible in the z direction. Thus, the problem is two-dimensional. Our interest is to determine the temperature distribution $T(x, y)$ in the plate. The plate is the fin here.

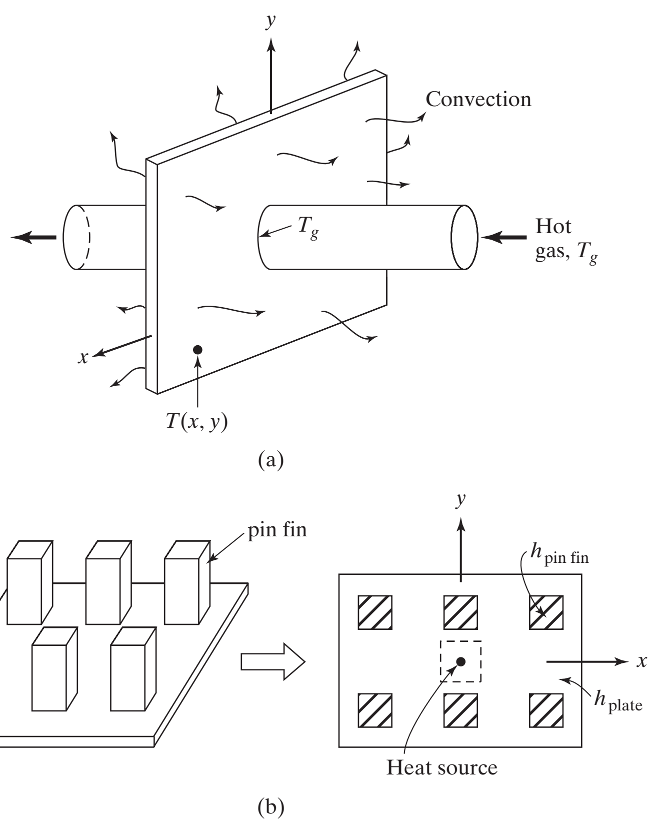


FIGURE 10.12 Two-dimensional fins.

Considering a differential area dA, the convection heat loss from both lateral surfaces of the fin is $2h(T - T_\infty)\,dA$. Treating this heat loss as a negative heat source per unit volume, $Q = -2h(T - T_\infty)/t$, where t is the thickness of the plate. Equation 10.43 yields the differential equation for two-dimensional fins, namely,

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) - C(T - T_\infty) + Q = 0 \qquad (10.71)$$

where $C = -2h/t$. Another example of a two-dimensional fin may be found in electronic packaging. The thin plate shown in Fig. 10.12b is subjected to a heat source from the surface underneath generated from electronic chips or other circuitry. Pin fins are attached to the top surface to dissipate the heat. As shown in the figure, the plate may be considered as a two-dimensional fin with higher convective heat-transfer coefficients where the pin fins are attached. In fact, these coefficients may be related to the fin size and material. Maximum temperature at the surface of the chip will be of importance in this analysis. The conductivity matrix $\mathbf{k}$ in Eq. 10.66 and the right-hand side heat-rate vector $\mathbf{r}_Q$ in Eq. 10.67 get augmented by the matrices

$$+\frac{CA_e}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad \text{and} \quad +CT_\infty\frac{A_e}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad (10.72)$$

respectively.

Preprocessing for Program HEAT2D

The input data file for program HEAT2D can be created, in most part, using the MESHGEN program. Mesh generation is as usual. Treat specified temperatures as "constrained degrees of freedom," nodal heat sources as "loads," element heat sources as "element characteristics" (enter zero if there are none), and thermal conductivity as "material property." The only thing that remains is the heat flux and convection boundary conditions along the edges; for this, simply edit the data file that you have created and enter this information as per the format of Example 10.4. Note that in heat conduction each node has only one degree of freedom.

10.3 TORSION

Consider a prismatic rod of arbitrary cross-sectional shape, which is subjected to a twisting moment M as shown in Fig. 10.13. The problem is to determine shearing stresses τ_{xz} and τ_{yz} (Fig. 10.14) and the angle of twist per unit length, α. It can be shown that the solution of such problems, with simply connected cross sections, reduces to solving the two-dimensional equation

$$\frac{\partial^2\theta}{\partial x^2} + \frac{\partial^2\theta}{\partial y^2} + 2 = 0 \qquad \text{in } A \qquad (10.73)$$

$$\theta = 0 \qquad \text{on } S \qquad (10.74)$$

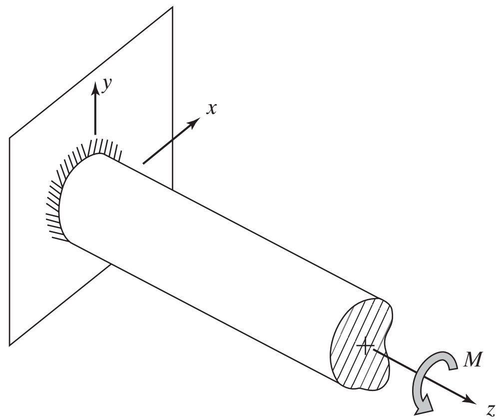


FIGURE 10.13 A rod of arbitrary cross section subjected to a torque.

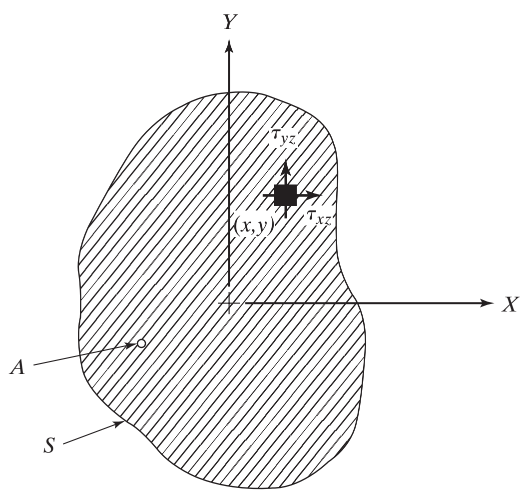


FIGURE 10.14 Shearing stresses in torsion.

where A is interior and S is the boundary of the cross section. Again, we note that Eq. 10.73 is a special case of Helmholtz's equations given in Eq. 10.1. In Eq. 10.74, θ is called the **stress function**, since once θ is known, then shearing stresses are obtained as

$$\tau_{xz} = G\alpha \frac{\partial \theta}{\partial y} \qquad \tau_{yz} = -G\alpha \frac{\partial \theta}{\partial x} \tag{10.75}$$

with α determined from

$$M = 2G\alpha \int_A \int \theta \, dA \tag{10.76}$$

where G is the shear modulus of the material. The finite element method for solving Eqs. 10.73 and 10.74 will now be given.

Triangular Element

The stress function θ within a triangular element is interpolated as

$$\theta = \mathbf{N}\theta^e \tag{10.77}$$

where $\mathbf{N} = [\xi \quad \eta \quad 1 - \xi - \eta]$ are the usual shape functions, and $\boldsymbol{\theta}^e = [\theta_1 \quad \theta_2 \quad \theta_3]^{\mathrm{T}}$ are the nodal values of θ. Furthermore, we have the isoparametric relations (Chapter 6)

$$x = N_1x_1 + N_2x_2 + N_3x_3$$
$$y = N_1y_1 + N_2y_2 + N_3y_3$$

$$\begin{Bmatrix} \dfrac{\partial\theta}{\partial\xi} \\ \dfrac{\partial\theta}{\partial\eta} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi} \\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial\theta}{\partial x} \\ \dfrac{\partial\theta}{\partial y} \end{Bmatrix} \tag{10.78}$$

or

$$\left[\frac{\partial\theta}{\partial\xi} \quad \frac{\partial\theta}{\partial\eta}\right]^{\mathrm{T}} = \mathbf{J}\left[\frac{\partial\theta}{\partial x} \quad \frac{\partial\theta}{\partial y}\right]^{\mathrm{T}}$$

where the Jacobian matrix is given by

$$\mathbf{J} = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix} \tag{10.79}$$

with $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$, and $|\det \mathbf{J}| = 2A_e$. The preceding equations yield

$$\left[\frac{\partial\theta}{\partial x} \quad \frac{\partial\theta}{\partial y}\right]^{\mathrm{T}} = \mathbf{B}\boldsymbol{\theta}^e \tag{10.80a}$$

or

$$[-\tau_{yz} \quad \tau_{xz}]^{\mathrm{T}} = G\alpha\,\mathbf{B}\boldsymbol{\theta}^e \tag{10.80b}$$

where

$$\mathbf{B} = \frac{1}{\det \mathbf{J}}\begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \tag{10.81}$$

The fact that identical relations also apply to the heat-conduction problem in the previous section shows the similarity of treating all field problems by the finite element method.

Galerkin Approach[2]

The problem in Eqs. 10.73 and 10.74 will now be solved using Galerkin's approach. The problem is to find the approximate solution θ such that

$$\int_A\int \phi\left(\frac{\partial^2\theta}{\partial x^2}+\frac{\partial^2\theta}{\partial y^2}+2\right)dA = 0 \tag{10.82}$$

for every $\phi(x, y)$ constructed from the same basis as θ and satisfying $\phi = 0$ on S. Since

$$\phi\,\frac{\partial^2\theta}{\partial x^2} = \frac{\partial}{\partial x}\left(\phi\,\frac{\partial\theta}{\partial x}\right) - \frac{\partial\phi}{\partial x}\frac{\partial\theta}{\partial x}$$

we have

$$\int_A\int\left[\frac{\partial}{\partial x}\left(\phi\,\frac{\partial\theta}{\partial x}\right)+\frac{\partial}{\partial y}\left(\phi\,\frac{\partial\theta}{\partial y}\right)\right]dA - \int_A\int\left(\frac{\partial\phi}{\partial x}\frac{\partial\theta}{\partial x}+\frac{\partial\phi}{\partial y}\frac{\partial\theta}{\partial y}\right)dA + \int_A\int 2\phi\,dA = 0 \tag{10.83}$$

Using the divergence theorem, the first term in the previous expression reduces to

$$\int_A\int\left[\frac{\partial}{\partial x}\left(\phi\,\frac{\partial\theta}{\partial x}\right)+\frac{\partial}{\partial y}\left(\phi\,\frac{\partial\theta}{\partial y}\right)\right]dA = \int_S \phi\left(\frac{\partial\theta}{\partial x}n_x + \frac{\partial\phi}{\partial y}n_y\right)dS = 0 \tag{10.84}$$

where the right-hand side is equated to zero owing to the boundary condition $\phi = 0$ on S. Equation 10.83 becomes

$$\int_A\int\left[\frac{\partial\phi}{\partial x}\frac{\partial\theta}{\partial x}+\frac{\partial\phi}{\partial y}\frac{\partial\theta}{\partial y}\right]dA - \int_A 2\phi\,dA = 0 \tag{10.85}$$

Now, we introduce the isoparametric relations $\theta = \mathbf{N}\boldsymbol{\theta}^e$, etc., as given in Eqs. 10.77–10.81. Further, we denote the global virtual-stress function vector as $\boldsymbol{\Psi}$ whose dimension equals number of nodes in the finite element model. The virtual-stress function within each element is interpolated as

$$\phi = \mathbf{N}\boldsymbol{\psi} \tag{10.86}$$

Moreover, we have

$$\left[\frac{\partial\phi}{\partial x}\quad\frac{\partial\phi}{\partial y}\right]^{\mathrm{T}} = \mathbf{B}\boldsymbol{\psi} \tag{10.87}$$

Substituting these into Eq. 10.85 and noting that

$$\left(\frac{\partial\phi}{\partial x}\frac{\partial\theta}{\partial x}+\frac{\partial\phi}{\partial y}\frac{\partial\theta}{\partial y}\right) = \left(\frac{\partial\phi}{\partial x}\quad\frac{\partial\phi}{\partial y}\right)\begin{Bmatrix}\dfrac{\partial\theta}{\partial x}\\[2mm]\dfrac{\partial\theta}{\partial y}\end{Bmatrix}$$

[2]The functional approach would be based on minimizing

$$\pi = G\alpha^2\int_A\left\{\frac{1}{2}\left[\left(\frac{\partial\theta}{\partial x}\right)^2+\left(\frac{\partial\theta}{\partial y}\right)^2\right]-2\theta\right\}dA$$

we get

$$\sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{k}\boldsymbol{\theta}^e - \sum_e \boldsymbol{\psi}^{\mathrm{T}}\mathbf{f} = 0 \tag{10.88}$$

where

$$\mathbf{k} = A_e\mathbf{B}^{\mathrm{T}}\mathbf{B} \tag{10.89}$$

$$\mathbf{f} = \frac{2A_e}{3}[1 \quad 1 \quad 1]^{\mathrm{T}} \tag{10.90}$$

Equation 10.88 can be written as

$$\boldsymbol{\psi}^{\mathrm{T}}(\mathbf{K}\boldsymbol{\Theta} - \mathbf{F}) = 0 \tag{10.91}$$

which should hold for all $\boldsymbol{\Psi}$ satisfying $\Psi_i = 0$ at nodes i on the boundary. We thus get

$$\mathbf{K}\boldsymbol{\Theta} = \mathbf{F} \tag{10.92}$$

where rows and columns of **K** and **F** that correspond to boundary nodes have been deleted.

Example 10.5

Consider the shaft with a rectangular cross section as shown in Fig. E10.5a. Determine, in terms of M and G, the angle of twist per unit length.

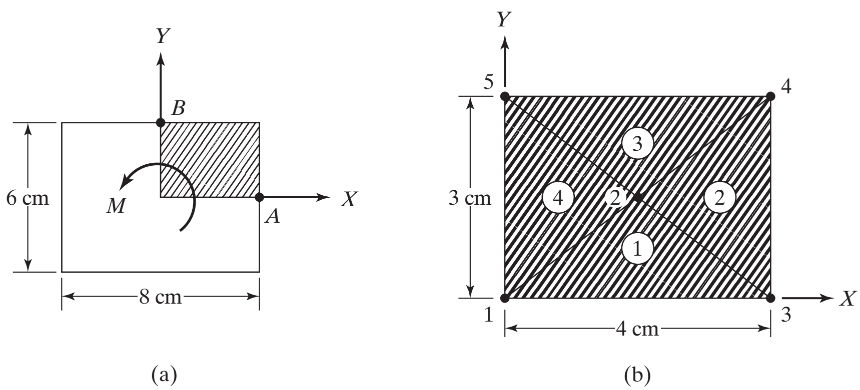


FIGURE E10.5

Solution A finite element model of a quadrant of this cross section is shown in Fig. E10.5b. We define the element connectivity as in the following table:

Element	1	2	3
1	1	3	2
2	3	4	2
3	4	5	2
4	5	1	2

Using the relations

$$\mathbf{B} = \frac{1}{\det \mathbf{J}}\begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix}$$

and

$$\mathbf{k} = A_e \mathbf{B}^{\mathrm{T}}\mathbf{B}$$

we get

$$\mathbf{B}^{(1)} = \frac{1}{6}\begin{bmatrix} -1.5 & 1.5 & 0 \\ -2 & -2 & 4 \end{bmatrix} \qquad \mathbf{k}^{(1)} = \frac{1}{2}\begin{array}{c} \begin{array}{ccc} \quad 1 \quad & \quad 2 \quad & \quad 3 \quad \end{array} \\ \begin{bmatrix} 1.042 & 0.292 & -1.333 \\ & 1.042 & -1.333 \\ \text{Symmetric} & & 2.667 \end{bmatrix} \end{array}$$

Similarly,

$$\mathbf{k}^{(2)} = \frac{1}{2}\begin{array}{c} \begin{array}{ccc} \quad 3 \quad & \quad 4 \quad & \quad 2 \quad \end{array} \\ \begin{bmatrix} 1.042 & -0.292 & -0.75 \\ & 1.042 & -0.75 \\ \text{Symmetric} & & 1.5 \end{bmatrix} \end{array}$$

$$\mathbf{k}^{(3)} = \frac{1}{2}\begin{array}{c} \begin{array}{ccc} \quad 4 \quad & \quad 5 \quad & \quad 2 \quad \end{array} \\ \begin{bmatrix} 1.042 & 0.292 & -1.333 \\ & 1.042 & -1.333 \\ \text{Symmetric} & & 2.667 \end{bmatrix} \end{array}$$

$$\mathbf{k}^{(4)} = \frac{1}{2}\begin{array}{c} \begin{array}{ccc} \quad 5 \quad & \quad 1 \quad & \quad 2 \quad \end{array} \\ \begin{bmatrix} 1.042 & -0.292 & -0.75 \\ & 1.042 & -0.75 \\ \text{Symmetric} & & 1.5 \end{bmatrix} \end{array}$$

Similarly, the element load vector $\mathbf{f} = (2A_e/3)[1, \quad 1, \quad 1,]^{\mathrm{T}}$ for each element is

$$\mathbf{f}^{(i)} = \begin{Bmatrix} 2 \\ 2 \\ 2 \end{Bmatrix} \qquad i = 1, 2, 3, 4$$

We can now assemble **K** and **F**. Since the boundary conditions are

$$\Theta_3 = \Theta_4 = \Theta_5 = 0$$

we are interested only in degrees of freedom (dof) 1 and 2. Thus, the finite element equations are

$$\frac{1}{2}\begin{bmatrix} 2.084 & -2.083 \\ -2.083 & 8.334 \end{bmatrix}\begin{Bmatrix} \Theta_1 \\ \Theta_2 \end{Bmatrix} = \begin{Bmatrix} 4 \\ 8 \end{Bmatrix}$$

The solution is

$$[\Theta_1 \quad \Theta_2] = [7.676 \quad 3.838]$$

Consider the equation

$$M = 2G\alpha \int_A \int \theta \, dA$$

Using $\theta = \mathbf{N}\boldsymbol{\theta}^e$, and noting that $\int_e \mathbf{N}\,dA = (A_e/3)[1 \quad 1 \quad 1]$, we get

$$M = 2G\alpha\left[\sum_e \frac{A_e}{3}(\theta_1^e + \theta_2^e + \theta_3^e)\right] \times 4$$

This multiplication by 4 is because the finite element model represents only one-quarter of the rectangular cross section. Thus, we get the angle of twist per unit length to be

$$\alpha = 0.004\,\frac{M}{G}$$

For given values of M and G, we can thus determine the value of α. Further, the shearing stresses in each element can be calculated from Eq. 10.80b. ■

10.4 POTENTIAL FLOW, SEEPAGE, ELECTRIC AND MAGNETIC FIELDS, AND FLUID FLOW IN DUCTS

We have discussed steady-state heat conduction and torsion problems in some detail. Other examples of field problems occurring in engineering are briefly discussed subsequently. Their solution follows the same procedure as for heat conduction and torsion problems, since the governing equations are special cases of the general Helmholtz equation, as discussed in the introduction of this chapter. In fact, the computer program HEAT2D can be used to solve the problems given in this section.

Potential Flow

Consider steady-state irrotational flow of an incompressible, nonviscous fluid around a cylinder, as shown in Fig. 10.15a. The velocity of the incoming flow is u_0. We need to determine the flow velocities near the cylinder. The solution of this problem is given by

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = 0 \tag{10.93}$$

where ψ is a stream function (m^3/s) per meter in the z direction. The value of ψ is constant along a stream line. A stream line is a line that is tangent to the velocity vector. By definition, there is no flow crossing a stream line. The flow between two adjacent stream lines can be thought of as the flow through a tube. Once the stream function $\psi = \psi(x, y)$ is known, the velocity components u and v along x and y, respectively, are obtained as

$$u = \frac{\partial\psi}{\partial y} \qquad v = \frac{-\partial\psi}{\partial x} \tag{10.94}$$

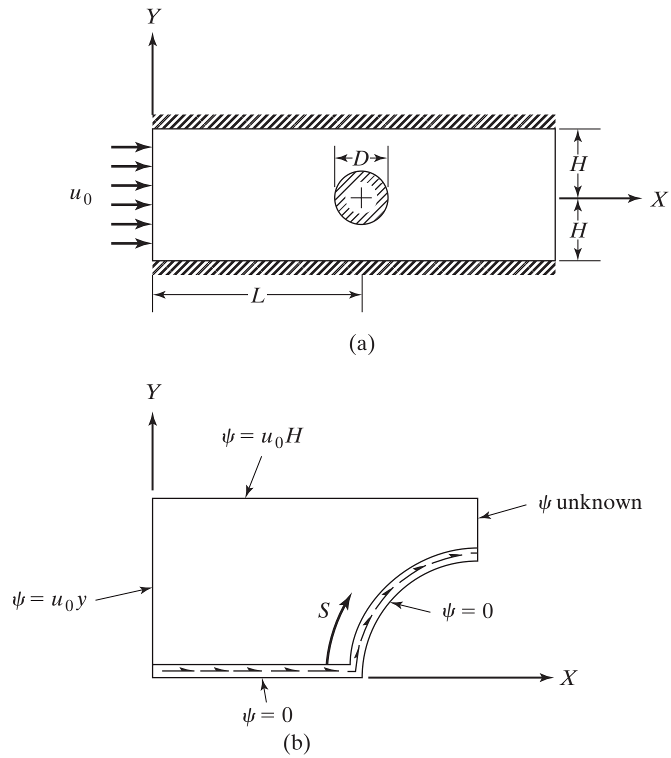


FIGURE 10.15 (a) Flow of an ideal fluid around a cylinder, and (b) boundary conditions for the finite element model.

Thus, the stream function ψ is analogous to the stress function in the torsion problem. Further, the rate of flow Q through a region bounded by two stream lines A and B is

$$Q = \psi_B - \psi_A \tag{10.95}$$

To illustrate the boundary conditions and use of symmetry, we consider one quadrant of Fig. 10.15a as shown in Fig. 10.15. First, note that velocities depend only on derivatives of ψ. Thus, we may choose the reference or base value of ψ; in Fig. 10.15, we have chosen $\psi = 0$ at all nodes on the x-axis. Then, along the y-axis, we have $u = u_0$ or $\partial\psi/\partial y = u_0$. This is integrated to give the boundary condition $\psi = u_0 y$. That is, for each node i along the y-axis, we have $\psi = u_0 y_i$. Along all nodes on $y = H$, we therefore have $\psi = u_0 H$. We now know that the velocity of the flow into the cylinder is zero, that is, $\partial\psi/\partial s = 0$ (Fig. 10.15b). Integrating this with the fact that $\psi = 0$ at the bottom of the cylinder results in $\psi = 0$ at all nodes along the cylinder. Thus, the fixed boundary is a stream line, as is to be expected.

Seepage

Flow of water that occurs in land drainage or seepage under dams can, under certain conditions, be described by Laplace's equation

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial \phi}{\partial y}\right) = 0 \tag{10.96}$$

where $\phi = \phi(x, y)$ is the hydraulic potential (or hydraulic head) and k_x and k_y are the hydraulic conductivities in the x and y directions, respectively. The fluid velocity components are obtained from Darcy's law as $v_x = -k_x(\partial\phi/\partial x), v_y = -k_y(\partial\phi/\partial y)$. Equation 10.96 is similar to the heat-conduction equation. Lines of $\phi =$ constant are called equipotential surfaces, across which flow occurs. Equation 10.96 can include a source or sink Q (see Table 10.1), representing discharge per unit volume, to solve problems where pumps remove water from an aquifer.

The appropriate boundary conditions associated with Eq. 10.96 are illustrated in the problem of water seepage through an earth dam (Fig. 10.16). The region to be modeled is shown shaded in the figure. Along the left and right surfaces, we have the boundary condition

$$\phi = \text{constant} \tag{10.97}$$

The impermeable bottom surface corresponds to the natural boundary condition, $\partial\phi/\partial n = 0$, where n is the normal, and does not affect the element matrices; the values of ϕ are unknowns. The top of the region is a *line of seepage* (free surface) where $\partial\phi/\partial n = 0$ and ϕ has its value equal to the y-coordinate:

$$\phi = y \tag{10.98a}$$

This boundary condition requires iterative solution of the finite element analysis since the location of the boundary is unknown. We first assume a location for the line of seepage and impose the boundary condition $\phi = y_i$ at nodes i on the surface. Then, we solve for $\phi = \tilde{\phi}$ and check the error $(\tilde{\phi} - y_i)$. Based on this error, we update the locations of the nodes and obtain a new line of seepage. This process is repeated until the error is

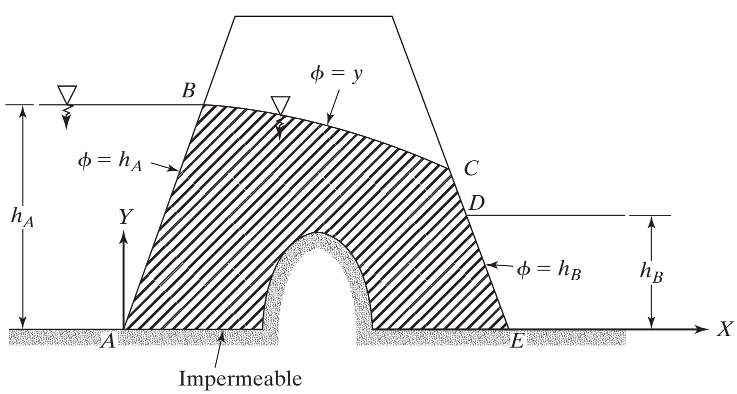


FIGURE 10.16 Seepage through an earth dam.

sufficiently small. Finally, portion CD in Fig. 10.16 is a *surface of seepage*. If no evaporation is taking place in this surface, then we have the boundary condition

$$\phi = \bar{y} \tag{10.98b}$$

where $\bar{y}$ is the coordinate of the surface.

Electrical and Magnetic Field Problems

In the area of electrical engineering, there are several interesting problems involving scalar and vector fields in two and three dimensions. We consider here some of the typical two-dimensional scalar field problems. In an isotropic dielectric medium with a permittivity of ϵ (F/m) and a volume charge density ρ (C/m^3), the electric potential u(V) must satisfy (Fig. 10.17)

$$\epsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = -\rho \tag{10.99}$$

where

$$u = a \quad \text{on } S_1 \qquad u = b \quad \text{on } S_2$$

Unit thickness may be assumed without loss of generality.

Finite element formulation may proceed from the minimization of the stored field energy

$$\Pi = \frac{1}{2}\int_A \int \epsilon\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right] dx\, dy - \int_A \rho u\, dA \tag{10.100}$$

In Galerkin's formulation, we seek the approximate solution u such that

$$\int_A \int \epsilon\left(\frac{\partial u}{\partial x}\frac{\partial \phi}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial \phi}{\partial y}\right) dx\, dy - \int_A \rho\phi\, dA = 0 \tag{10.101}$$

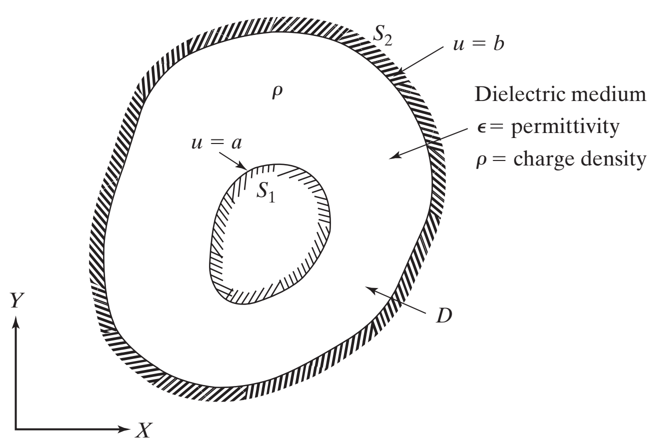


FIGURE 10.17 Electric potential problem.

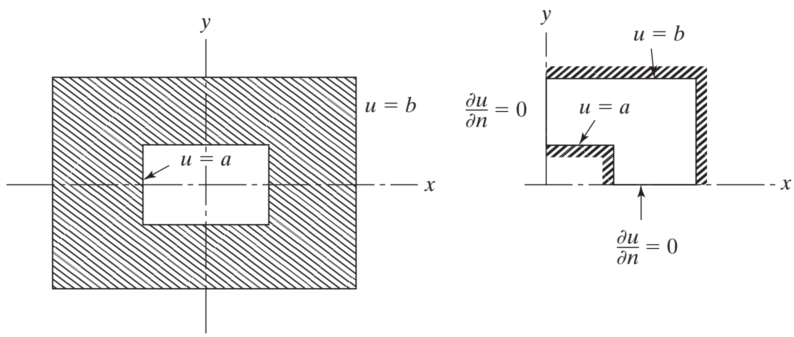


FIGURE 10.18 Rectangular coaxial cable.

for every ϕ constructed from the basis functions of u, satisfying $\phi = 0$ on S_1 and S_2. In Eq. 10.101, integration by parts has been carried out.

Permittivity ϵ for various materials is defined in terms of relative permittivity ϵ_R and permittivity of free space $\epsilon_0(=8.854 \times 10^{-12}\ \text{F/m})$ as $\epsilon = \epsilon_R \epsilon_0$. Relative permittivity of rubber is in the range of 2.5–3. The coaxial cable problem is a typical example of Eq. 10.99 with $\rho = 0$. Figure 10.18 shows the section of a coaxial cable of rectangular cross section. By symmetry, only a quarter of the section needs to be considered. On the separated boundary, $\partial u/\partial n = 0$ is a natural boundary condition, which is satisfied automatically in the potential and Galerkin formulations. Another example is the determination of the electrical field distribution between two parallel plates (Fig. 10.19). Here, the field extends to infinity. Since the field drops as we move away from the plates, an arbitrary large domain D is defined, enclosing the plates symmetrically. The dimensions of this enclosure may be 5–10 times the plate dimensions. However, we may use larger elements away from the plates. On the boundary S, we may typically set $u = 0$.

If u is the magnetic field potential, and μ is the permeability (H/m), the field equation is

$$\mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{10.102}$$

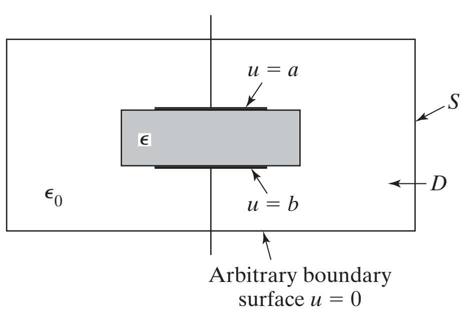


FIGURE 10.19 Parallel strips separated by dielectric medium.

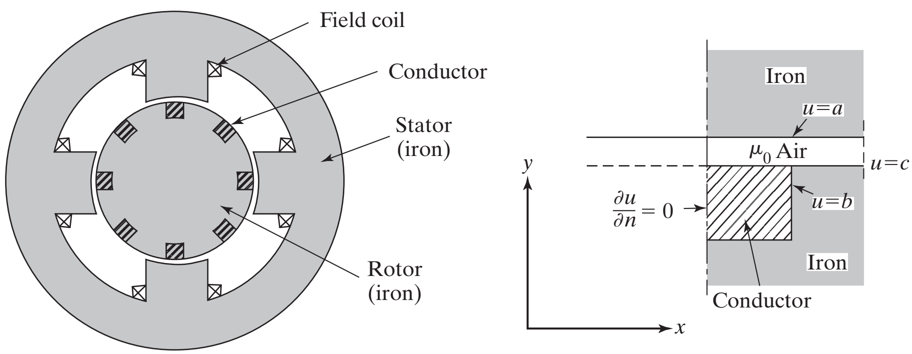


FIGURE 10.20 Model of a simple electric motor.

where u is the scalar magnetic potential (A). Permeability μ is defined in terms of relative permeability μ_R and permeability of free space $\mu_0(-4\pi \times 10^{-7}\,\text{H/m})$ as $\mu = \mu_R\mu_0 \cdot \mu_R$ for pure iron is about 4000, and for aluminum or copper it is about 1. Consider a typical application in an electric motor with no current flowing through the conductor, as shown in Fig. 10.20. We have $u = a$ and $u = b$ on the iron surface; $u = c$ is used on an arbitrarily defined boundary. ($u = 0$ may be used if the boundary is set at a large distance relative to the gap.)

The ideas may be easily extended to axisymmetric coaxial cable problems. Problems in three dimensions can be considered using the steps developed in Chapter 9.

Fluid Flow in Ducts

The pressure drop occurring in the flow of a fluid in long, straight, uniform pipes, and ducts is given by the equation

$$\Delta p = 2f\rho v_m^2 \frac{L}{D_h} \tag{10.103}$$

where f is the Fanning friction factor, ρ is the density, v_m is the mean velocity of fluid, L is the length of duct, and $D_h = (4 \times \text{area})/\text{perimeter}$ is the hydraulic diameter. The finite element method for determining the Fanning friction factor f for fully developed laminar flow in ducts of general cross-sectional shape will now be discussed.

Let fluid flow be in the z direction, with x, y being the plane of the cross section. A force balance (Fig. 10.21) yields

$$0 = pA - \left(p + \frac{dp}{dz}\Delta z\right)A - \tau_w P \Delta z \tag{10.104a}$$

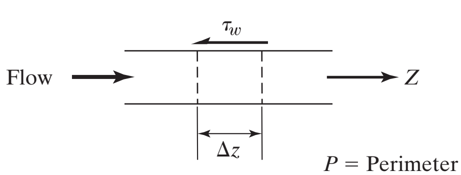


FIGURE 10.21 Force balance for fluid flow in a duct.

or

$$-\frac{dp}{dz} = \frac{4\tau}{D_h} \tag{10.104b}$$

where τ_w is the shear stress at the wall. The friction factor is defined as the ratio $f = \tau_w/(\rho v_m^2/2)$. The Reynolds number is defined as $R_e = v_m D_h/v$, where $v = \mu/\rho$ is the kinematic viscosity, with μ representing absolute viscosity. Thus, from previous equations, we get

$$-\frac{dp}{dz} = \frac{2\mu v_m f R_e}{D_h^2} \tag{10.105}$$

The momentum equation is given by

$$\mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right) - \frac{dp}{dz} = 0 \tag{10.106}$$

where $w = w(x, y)$ is the velocity of the fluid in the z direction. We introduce the nondimensional quantities

$$X = \frac{x}{D_h} \qquad Y = \frac{y}{D_h} \qquad W = \frac{w}{2v_m f R_e} \tag{10.107}$$

Equations 10.105–10.107 result in

$$\frac{\partial^2 W}{\partial X^2} + \frac{\partial^2 W}{\partial Y^2} + 1 = 0 \tag{10.108}$$

Since the velocity of the fluid in contact with the wall of the duct is zero, we have $w = 0$ on the boundary, and hence,

$$W = 0 \qquad \text{on boundary} \tag{10.109}$$

The solution of Eqs. 10.108 and 10.109 by the finite element method follows the same steps as for the heat conduction or torsion problems. Once W is known, then its average value may be determined from

$$W_m = \frac{\int_A W\,dA}{\int_A dA} \tag{10.110}$$

The integral $\int_A W\,dA$ may be readily evaluated using the element shape functions. For example, with constant-strain triangle (CST) elements, we get $\int_A W\,dA = \sum_e [A_e(w_1 + w_2 + w_3)/3]$. Once W_m is obtained, Eq. 10.107 is used to get

$$W_m = \frac{w_m}{2v_m f R_e} = \frac{v_m}{2v_m f R_e} \tag{10.111}$$

which yields

$$f = \frac{1/(2W_m)}{R_e} \tag{10.112}$$

Our aim is to determine the constant $1/(2W_m)$, which depends only on the cross-sectional shape. In preparing input data to solve Eqs. 10.108 and 10.109, we should remember that the nodal coordinates are in nondimensional form, as given in Eq. 10.107.

Acoustics

A very interesting physical phenomenon that can be modeled using Helmholtz's equation in Eq. 10.1 occurs in acoustics. Consider the wave equation in linear acoustics, given by

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{10.113}$$

where p is a scalar quantity, a function of position and time, representing the change in pressure from some ambient value, and c is the speed of the sound wave in the media. In many situations, the acoustical disturbance, and hence the response, is sinusoidal (harmonic) in time. That is, we may represent p as

$$p(\mathbf{x}, t) = p_{\text{amp}}(\mathbf{x}) \cos(\omega t - \phi) \tag{10.114}$$

where p_{amp} is the amplitude or peak pressure, ω is the angular frequency in radians/s, and ϕ is the phase. Substituting Eq. 10.114 into Eq. 10.113, we obtain the Helmholtz equation

$$\nabla^2 p_{\text{amp}} + k^2 p_{\text{amp}} = 0 \quad \text{in } V \tag{10.115}$$

where $k = \omega/c$ is called the wave number, and V represents the acoustic space. Upon solving Eq. 10.115 for the pressure amplitude, we may obtain the pressure function from Eq. 10.114.

Use of complex arithmetic greatly simplifies the handling of amplitude and phase in acoustics. Note the following complex arithmetic concepts: First, a complex number is represented as $a + ib$, where a is the real part and b is the imaginary part, with $i = \sqrt{-1}$ being the imaginary unit. Second, $\mathbf{e}^{-i\phi} = \cos\phi - i\sin\phi$. We may now write p in Eq. 10.114 as

$$p = R_e\{P_{\text{amp}}\mathrm{e}^{-i(\omega t - \phi)}\} = R_e\{p_{\text{amp}}\mathrm{e}^{i\phi}\mathrm{e}^{-i\omega t}\} = R_e\{\hat{p}\mathrm{e}^{-i\omega t}\} \tag{10.116}$$

where R_e denotes the real part of the complex number. In Eq. 10.116, $\hat{p} = p_{\text{amp}}(\cos\phi + i\sin\phi)$. For example, assume that

$$\hat{p} = 3 - 4i$$

Then we have $p_{\text{amp}} = \sqrt{(3^2 + 4^2)} = 5$ and $\phi = \tan^{-1}(-4/3) = -53.1° = -0.927$ rad, resulting in the pressure $p = 5\cos(\omega t + 0.927)$.

If we substitute $p = R_e\{\hat{p}\mathrm{e}^{-i\omega t}\}$ into the wave equation, we see that the complex pressure term $\hat{p}$ also satisfies Helmholtz equation

$$\nabla^2\hat{p} + k^2\hat{p} = 0 \quad \text{in } V \tag{10.117}$$

Boundary Conditions

A vibrating or stationary surface S adjacent to the fluid imposes boundary conditions, which must be accounted for while solving Eq. 10.117. Common types of boundary conditions are as follows:

(i) Specified pressure: $\hat{p} = \hat{p}_0$ on S_1. For example, $p = 0$ is a pressure release condition that occurs when a sound wave encounters the atmosphere (ambient surrounding).

(ii) Specified normal velocity: $v_n = vn_0$ on S_2, where $v_n = \boldsymbol{v} \cdot \mathbf{n}$. This states that the normal component of the wave velocity at the solid (impenetrable) surface must be the same as that of the surface itself. Noting that velocity at a point may be specified as a complex quantity, just as in Eq. 10.116, with $v = R_e\{\hat{v}e^{-i\omega t}\}$ the boundary condition can be written as $\hat{v}_n = \hat{v}_{n0}$ on S_2. Equivalently, this condition can be written as

$$\frac{1}{ik\rho c}\nabla\hat{p} \cdot \mathbf{n} = \hat{v}_{n0} \tag{10.118a}$$

If the surface is stationary, then the condition takes the form

$$\frac{\partial\hat{p}}{\partial n} = \nabla\hat{p} \cdot \mathbf{n} = 0 \tag{10.118b}$$

(iii) There is also the “mixed” boundary condition involving both p and $\partial p/\partial n$, which occurs when the surface is porous. The impedance Z is specified, where $\hat{p} = Z(\omega)\hat{v}_n$, where $\hat{v}_n$ is the inward normal velocity.

Finally, acoustics in an open region (with no enclosed surface) requires the pressure field to satisfy the Sommerfeld conditions at a distance far from the sound source. However, boundary element methods are more popular in such situations. Later, we focus on interior acoustic cavities (closed regions) with impenetrable surfaces. Thus, we consider only solution of Eq. 10.117 with the boundary conditions in (i) and (ii).

One-Dimensional Acoustics

In one dimension, Eq. 10.117 reduces to

$$\frac{d^2\hat{p}}{dx^2} + k^2\hat{p} = 0 \tag{10.119}$$

Assume that the problem is a duct or tube, with a piston vibrating the air at the left end (at $x = 0$) and a rigid wall at the right end at $x = L$. Thus, the boundary conditions are

$$\left.\frac{d\hat{p}}{dx}\right|_{x=L} = 0 \quad \text{and} \quad \left.\frac{d\hat{p}}{dx}\right|_{x=0} = ik\rho c v_0 \tag{10.120}$$

Galerkin's approach requires the equation

$$\int_0^L \phi\left[\frac{d^2\hat{p}}{dx^2} + k^2\hat{p}\right]dx = 0$$

to be satisfied for every choice of an arbitrary pressure field $\phi(x)$. If the pressure $\hat{p}$ was specified at a point, then ϕ must equal zero at that point. Here, however, pressure boundary conditions are absent. Now, following the Galerkin procedure, identical to one-dimensional heat transfer in thin fins (see Eqs. 10.33 and 10.121), we obtain

$$\phi(L)\frac{d\hat{p}}{dx}(L) - \phi(0)\frac{d\hat{p}}{dx}(0) - \int_0^L \frac{d\hat{p}}{dx}\frac{d\phi}{dx}\,dx + k^2\int_0^L \phi\hat{p}\,dx = 0 \qquad (10.121)$$

Using two-node elements with the usual linear shape functions, we get

$$\phi = \mathbf{N}\boldsymbol{\psi},\ \hat{\mathbf{p}} = \mathbf{N}\hat{\mathbf{p}},\ \frac{d\hat{p}}{dx} = \mathbf{B}\hat{\mathbf{p}},\ \frac{d\phi}{dx} = \mathbf{B}\boldsymbol{\psi}$$

where $\boldsymbol{\psi} = [\psi_1, \psi_2]^{\mathrm{T}}$ is the arbitrary pressure field at the ends of an element and, as before, $\mathbf{N} = [N_1 \;\; N_2]$, $\mathbf{B} = [1/(x_2 - x_1)][-1 \;\; 1]$, and nodal pressure vector $\hat{\mathbf{p}} = [\hat{p}_1 \;\; \hat{p}_2]^{\mathrm{T}}$. Using these, we get

$$\int_{\ell_e} \frac{d\hat{p}}{dx}\frac{d\phi}{dx}\,dx = \hat{\mathbf{p}}^{\mathrm{T}}\mathbf{k}\boldsymbol{\psi},\ \int_{\ell_e} \phi\hat{p}\,dx = \hat{\mathbf{p}}^{\mathrm{T}}\mathbf{m}\boldsymbol{\psi}$$

where $\mathbf{k}$ and $\mathbf{m}$ are the acoustic stiffness and mass matrices, respectively, given by

$$\mathbf{k} = \frac{1}{\ell_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \mathbf{m} = \frac{\ell_e}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad (10.122)$$

The integral over the entire length of the tube leads to the usual assembly of element matrices as

$$\int_0^L \frac{d\hat{p}}{dx}\frac{d\phi}{dx}\,dx = \boldsymbol{\psi}^{\mathrm{T}}\mathbf{K}\hat{\mathbf{P}} \quad \text{and} \quad \int_0^L \hat{p}\phi\,dx = \boldsymbol{\psi}^{\mathrm{T}}\mathbf{M}\hat{\mathbf{P}} \qquad (10.123)$$

where $\hat{\mathbf{P}}$ and $\boldsymbol{\Psi}$ are global nodal vectors of dimension $(N \times 1)$, with N = number of nodes in the model. Referring to Eq. 10.121, and using Eq. 10.120, we get

$$\phi(L)\frac{d\hat{p}}{dx}(L) - \phi(0)\frac{d\hat{p}}{dx}(0) = -\phi(0)ik\rho cv_0$$

Denoting $\mathbf{F} = -ik\rho cv_0[1 \quad 0 \quad 0 \quad 0 \cdots 0]^{\mathrm{T}}$ and noting that $\phi(0) \equiv \boldsymbol{\psi}_1$, we can write $-\phi(0)ik\rho cv_0 = \boldsymbol{\psi}^{\mathrm{T}}\mathbf{F}$. Substituting this and Eq. 10.120 into Eq. 10.121, and noting that $\boldsymbol{\Psi}$ is arbitrary, we get

$$\mathbf{K}\hat{\mathbf{P}} - k^2\mathbf{M}\hat{\mathbf{P}} = \mathbf{F} \qquad (10.124)$$

Equation 10.124 may be solved as $\hat{\mathbf{P}} = [\mathbf{K} - k^2\mathbf{M}]^{-1}\mathbf{F}$. However, determining modes of the system (as explained subsequently) and then using these to solve for $\hat{\mathbf{P}}$ is more efficient and gives better understanding.

One-Dimensional Axial Vibrations

As is well known in dynamic systems, if the forcing function $\mathbf{F}$ coincides with the natural frequency of the system, resonance occurs. In the present context of acoustics in a tube, if the piston vibrates the air in the tube at certain frequencies, the air reflected from the fixed end will arrive at the piston face just as the piston begins its next stroke. That is, successive echoes reinforce the pressure on the piston face. The shapes of these waves are called eigenvectors or mode shapes, and the corresponding values $k_n^2 = \omega_n^2/c^2$ are eigenvalues; $\omega_n/2\pi$ is the nth resonance frequency in cycles per second (cps) or hertz (Hz). Determination of the mode shapes and frequencies is of interest in itself and also useful to efficiently solve Eq. 10.121 using the "method of modal superposition," especially for large finite element models. The eigenvalue problem is obtained by setting $\mathbf{F} = 0$, which in effect is a tube with both ends rigid. The resulting free-vibration problem is similar to perturbing a spring-mass system and observing its natural vibrations. We obtain the eigenvalue problem

$$\mathbf{K}\hat{\mathbf{P}}^n = k_n^2\mathbf{M}\hat{\mathbf{P}}^n \tag{10.125}$$

In Eq. 10.122, $\lambda_n = k_n^2$ is the nth eigenvalue. The solution $\hat{\mathbf{P}} = 0$ is called the "trivial" solution and is of no interest. Our interest is in nonzero pressures that satisfy Eq. 10.125, which implies that det $\lfloor \mathbf{K} - k_n^2\mathbf{M} \rfloor = 0$. Techniques for solving eigenvalue problems are given in Chapter 11. Here, in Example 10.6, we simply use the Jacobi solver from that chapter and present the solution.

Example 10.6

Consider a tube with both ends rigid, of length 6 m as shown in Fig. E10.6a. Fluid in the tube is air, thus $c = 343$ m/s. Determine the mode shapes and natural frequencies and compare them with analytical solution.

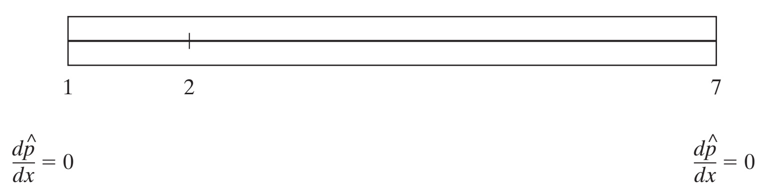


FIGURE E10.6A

Adopting a six-element model, we get

$$\mathbf{K} = \begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{bmatrix}, \mathbf{M} = \frac{1}{6}\begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 2 \end{bmatrix}$$

The banded versions of these matrices are

$$\mathbf{K}_{\text{banded}} = \begin{bmatrix} 1 & -1 \\ 2 & -1 \\ 2 & -1 \\ 2 & -1 \\ 2 & -1 \\ 2 & -1 \\ 1 & 0 \end{bmatrix}, \mathbf{M}_{\text{banded}} = \frac{1}{6}\begin{bmatrix} 2 & 1 \\ 4 & 1 \\ 4 & 1 \\ 4 & 1 \\ 4 & 1 \\ 4 & 1 \\ 2 & 0 \end{bmatrix}$$

The following input data file was created for use with program Jacobi:

```
Banded Stiffness and Mass for one-dimensional Acoustic Vibrations
Number of dof    Bandwidth
 7                2

Banded Stiffness Matrix
 1      -1
 2      -1
 2      -1
 2      -1
 2      -1
 2       1
 1       0


Banded Mass Matrix
 0.333333    0.166667
 0.666667    0.166667
 0.666667    0.166667
 0.666667    0.166667
 0.666667    0.166667
 0.666667    0.166667
 0.333333    0.166667
```

Solution The solution for the eigenvalues and eigenvectors are plotted in Fig. E10.6b. Note that frequency in cps is obtained from the eigenvalue as $f, cps = \dfrac{c \cdot \sqrt{\lambda_n}}{2\pi}$.

We note that the first few frequencies match the theoretical values, $f_n = mc/2L$, $m = 1, 2, \ldots$ quite well. Higher order elements that better maintain the boundary conditions are indicated for accurate prediction of higher frequencies. Comparison between finite element solution and theory for the various natural frequencies in cps is tabulated as follows:

Finite Element	28.9	59.8	94.6	133.7
Theory	28.6	57.2	85.8	114.3

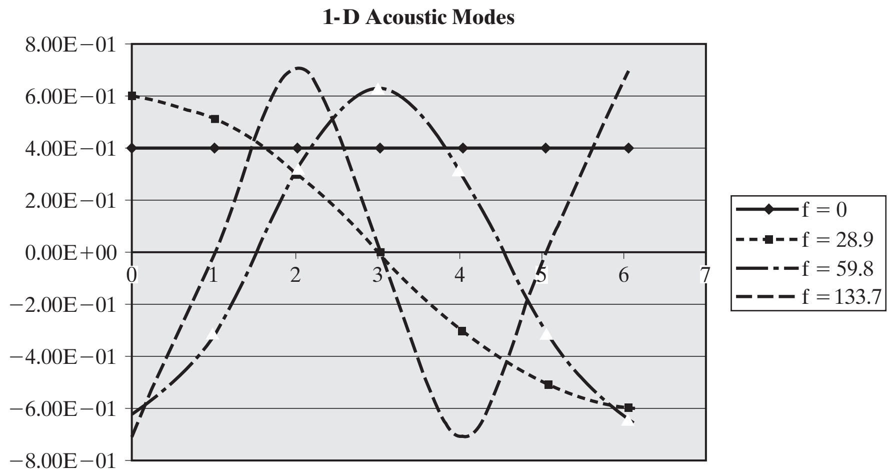


FIGURE E10.6B

■

Two-Dimensional Acoustics

The two-dimensional problem considered here is

$$\frac{\partial^2 \hat{p}}{\partial x^2} + \frac{\partial^2 \hat{p}}{\partial y^2} + k^2 \hat{p} = 0 \qquad \text{in } A \tag{10.126}$$

together with the boundary conditions

$$\hat{p} = \hat{p}_0 \qquad \text{on } S_1$$

and

$$\frac{1}{ik\rho c} \nabla \hat{p} \cdot \mathbf{n} = \hat{v}_{n0} \qquad \text{on } S_2 \tag{10.127}$$

Galerkin's variational principle requires that the equation

$$\int_A \phi \left(\frac{\partial^2 \hat{p}}{\partial x^2} + \frac{\partial^2 \hat{p}}{\partial y^2} + k^2 \hat{p} \right) dA = 0 \tag{10.128}$$

must be satisfied for every ϕ, $\phi = 0$ on S_1. Following the procedure used in the heat-conduction problem earlier, with three-node triangles, the student should be able to arrive at the equations

$$[\mathbf{K} - k^2 \mathbf{M}] \hat{\mathbf{P}} = \mathbf{F} \tag{10.129}$$

where **K** and **M** are assembled from element matrices

$$\mathbf{k} = A_e \mathbf{B}^{\mathrm{T}} \mathbf{B}$$

where

$$\mathbf{B} = \frac{1}{2A_e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix}, \quad \mathbf{m} = \frac{A_e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$\mathbf{F} = -ik\rho c \int_{S_2} \hat{v}_{n0} \mathbf{N} \, dS \tag{10.130}$$

Computation of **F** is similar to the computation of force vector from surface tractions in Chapter 5. As in the one-dimensional case, acoustic modes can be obtained by setting $\mathbf{F} = 0$ and solving the resulting eigenvalue problem.

10.5 CONCLUSION

We have seen that all the field equations stem from the Helmholtz equations. Our presentation stressed the physical problems rather than considering one general equation with different variables and constants. This approach should help us identify the proper boundary conditions for modeling a variety of problems in engineering.

Input Data/Output

```
INPUT TO HEAT1D
PROGRAM HEAT1D << 1D HEAT ANALYSIS
Example 10.1
NE         #B.C.s     #Nodal Heat Sources
3          2          0
ELEM#      Thermal Conductivity
1          20
2          30
3          50
NODE#      Coordinate
1          0
2          0.3
3          0.45
4          0.6
NODE#      BC-TYP followed by T0(if TEMP) or q0(if FLUX) or H and Tinf(if CONV)
1          CONV
25         800
4          TEMP
20
NODE#      HEAT SOURCE
```

```
OUTPUT FROM HEAT1D
Results from Program HEAT1D
Example 10.1
Node#   Temperature
1       304.7634
2       119.0496
3       57.1451
4       20.0023
```

```
INPUT TO HEAT2D
PROGRAM HEAT2D << 2D Heat Analysis
Example 10.4
NN     NE    NM     NDIM  NEN   NDN
5      3     1      2     3     1
ND     NL    NMPC
2      0     0
Node#  X     Y
1      0     0
2      0.4   0
3      0.4   0.15
4      0.4   0.3
5      0     0.3
Elem#  N1    N2     N3    Mat#  Elem_Heat_source
1      1     2      3     1     0
2      1     3      5     1     0
3      4     3      5     1     0
DOF#  Displacement (Specified Temperature)
4    180
5    180
DOF# Load (Nodal Heat Source)
MAT# Thermal Conductivity
1    1.5
No. of edges with Specified Heat flux FOLLOWED BY two edges & q0 (positive if out)
0
No. of Edges with Convection FOLLOWED BY edge(2 nodes) & h & Tinf
2
2    3       50      25
3    4       50      25
```

```
OUTPUT FROM HEAT2D
Program Heat2D - Heat Flow Analysis
Example 10.4
Node#   Temperature
1       124.4960
2        34.0451
3        45.3514
4       179.9998
5       179.0000
Conduction Heat Flow per Unit Area in Each Element
Elem#   Qx= -K*DT/Dx    Qy= -K*DT/Dy
1       339.1909        -113.0636
2       400.8621        -277.5199
3       0.00051         -1346.4840
```

PROBLEMS

10.1. Consider a brick wall (Fig. P10.1) of thickness $L = 30$ cm, $K = 0.7\ \text{W/m} \cdot °\text{C}$. The inner surface is at 40°C, and the outer surface is exposed to cold air at −10 °C. The heat-transfer coefficient associated with the outside surface is $h = 50\ \text{W/m}^2 \cdot °\text{C}$. Determine the steady-state temperature distribution within the wall and also the heat flux through the wall. Use a two-element model, and obtain the solution by hand calculations. Assume one-dimensional flow. Then prepare input data and run program HEAT1D.

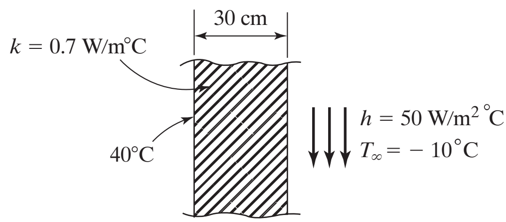


FIGURE P10.1

10.2. Heat is entering into a large plate at the rate of $q_0 = -450\ \text{W/m}^2$ as in Fig. P10.2. The plate is 100-mm thick. The outside surface of the plate is maintained at a temperature of 15°C. Using two finite elements, solve for the vector of nodal temperatures **T**. Thermal conductivity $k = 1.0\ \text{W/m} \cdot °\text{C}$.

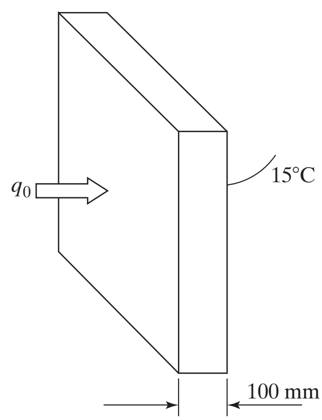


FIGURE P10.2

10.3. Refer to Fig. P10.3. The outside of a heating tape is insulated, while the inside is attached to one face of a 4-cm thick stainless steel plate ($k = 16.6\ \text{W/m} \cdot °\text{C}$). The other face of the plate is exposed to the surroundings, which are at a temperature of 20°C. Heat is supplied at a rate of 300 W/m². Determine the temperature of the face to which the heating tape is attached.

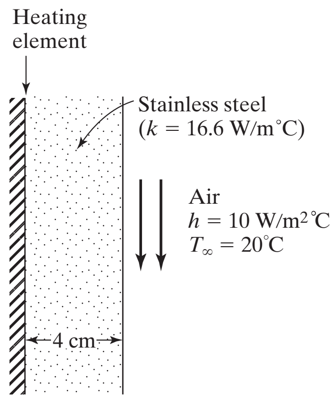


FIGURE P10.3

10.4. Consider a pin fin (Fig. P10.4) having a diameter of 0.3 in. and length of 6 in. At the root, the temperature is 147°F. The ambient temperature is 80°F and $h = 5$ BTU/(h · ft^2 · °F). Take $k = 25.5$ BTU/(h · ft · °F). Assume that the tip of the fin is insulated. Using a two-element model, determine the temperature distribution and heat loss in the fin (by hand calculations).

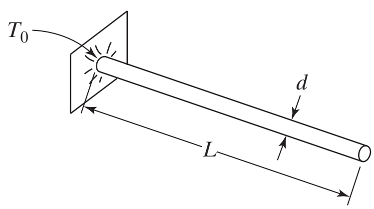


FIGURE P10.4

10.5. In our derivation using Galerkin's approach for straight rectangular fins, we assume that the fin tip is insulated. Modify the derivation to account for the case when convection occurs from the tip of the fin as well. Repeat Problem 10.4 with this type of boundary condition.

10.6. A point P is located inside the triangle as shown in Fig. P10.6. Assuming a linear distribution, determine the temperature at P. Coordinates of the various points are given in the following table:

Point	X-coordinate	Y-coordinate
1	1	1
2	8	3
3	14	12
P	10	06

Temperatures at nodes 1, 2, 3 are 120, 150, 90°C respectively.

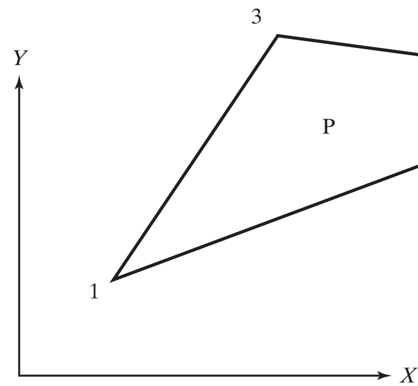


FIGURE P10.6

10.7. Consider a mesh for a heat-conduction problem shown in Fig. P10.7. Determine the (half) bandwidth, *NBW*.

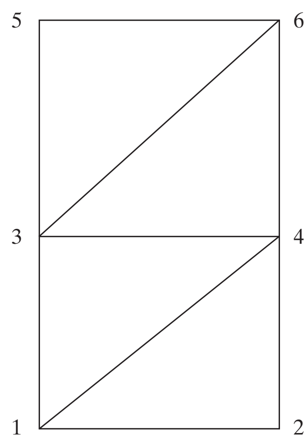


FIGURE P10.7

10.8. Using Galerkin's approach on a heat-conduction problem has resulted in the equations

$$(\psi_1, \psi_2)\left\{\begin{bmatrix} 5 & -1 \\ -1 & 6 \end{bmatrix}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} - \begin{pmatrix} 15 \\ 25 \end{pmatrix}\right\} = 0$$

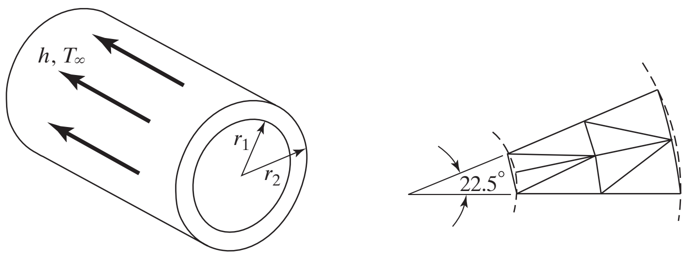
(a) Determine the temperature T_2 if $T_1 = 35°$.
(b) Determine the temperatures (T_1, T_2) if $T_1 - T_2 = 18°$.

10.9. Assume that the heat source vector is linearly distributed within a three-node triangular element with $\mathbf{Q}^e = [Q_1\ \ Q_2\ \ Q_3]^{\mathrm{T}}$ being the nodal values.
(a) Derive an expression for the heat-rate vector $\mathbf{r}_Q$. Show whether your expression reduces to Eq. 10.67 when Q is a constant or $Q_1 = Q_2 = Q_3$.
(b) Derive the element heat-rate vector $\mathbf{r}_Q$ due to a point heat source of magnitude Q_0 located at (ξ_0, η_0) within the element.

10.10. A long steel tube (Fig. P10.10a) with inner radius $r_1 = 3$ cm and outer radius $r_2 = 5$ cm and $k = 15$ W/m · °C has its inner surface heated at a rate $q_0 = -150\,000$ W/m^2. (The minus sign indicates that heat flows into the body.) Heat is dissipated by convection from the outer surface into a fluid at temperature $T_\infty = 150$°C and $h = 450$ W/m^2 · °C. Considering the eight-element, nine-node finite element model shown in Fig. P10.10b, determine the following:

(a) The boundary conditions for the model.

(b) The temperatures T_1, T_2 at the inner and outer surfaces, respectively. Use program HEAT2D.



(a) (b)

FIGURE P10.10

10.11. Solve Example 10.4 with a fine mesh consisting of about 100 elements. View the isotherms using CONTOUR. Plot temperature as a function of x, y. Also, calculate the total heat flow into the plate and the total heat leaving the plate. Is the difference zero? Explain.

10.12. In P10.10, assume that the steel tube is free of stress at a room temperature $T = 25$°C. Determine the thermal stresses in the tube using program AXISYM. Take $E = 200{,}000$ MPa and $\nu = 0.3$.

10.13. The brick chimney shown in Fig. P10.13 is 6 m high. The inside surfaces are at a uniform temperature of 150°C and the outside surfaces are held at a uniform temperature of 25°C. Using a quarter-symmetry model and preprocessing program MESHGEN (plus a little editing as discussed in the text), determine the total rate of heat transfer through the chimney wall. Thermal conductivity of the brick used is 0.72 W/m · °C. [For thermal conductivities of various materials, see F. W. Schmidt et al., *Introduction to Thermal Sciences*, 2nd ed., John Wiley & Sons, Inc: New York, (1993).]

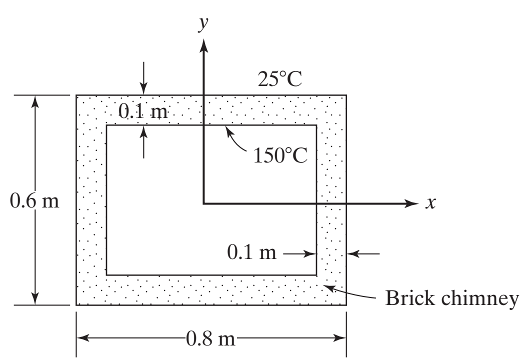


FIGURE P10.13

10.14. A large industrial furnace is supported on a long column of fireclay brick, which is 1×1 m on a side (Fig. P10.14). During steady-state operation, installation is such that three surfaces of the column are maintained at 400 K while the remaining surface is exposed to an airstream for which $T_\infty = 200$ K and $h = 20$ W/m$^2 \cdot$K. Determine, using program HEAT2D, the temperature distribution in the column and the heat rate to the airstream per unit length of column. Take $k = 1$ W/m $\cdot$ K.

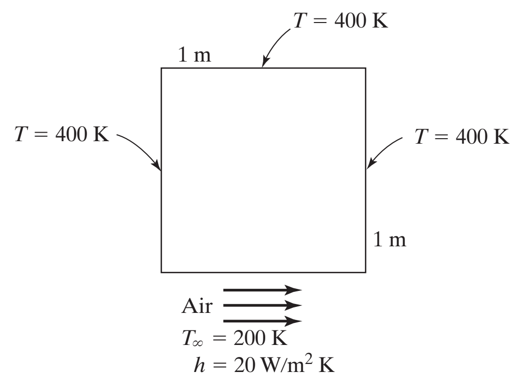


FIGURE P10.14

10.15. Figure P10.15 shows a two-dimensional fin. A hot pipe running through the thin plate results in the inner surface being maintained at a specified temperature of 80°C. Thickness of plate = 0.2 cm, $k = 100$ W/m $\cdot$ °C, and $T_\infty = 20$°C. Determine the temperature distribution in the plate. (You will need to modify program HEAT2D to account for the matrices in Eq. 10.72.)

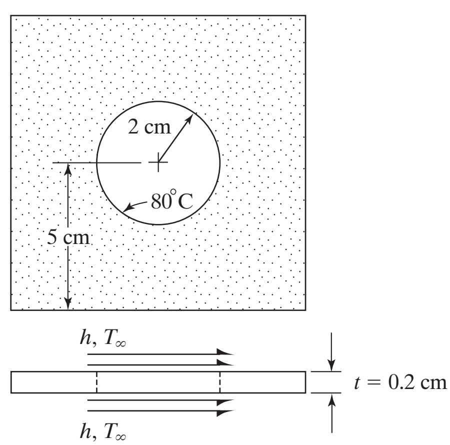


FIGURE P10.15

10.16. A thermal diffuser of axisymmetric shape is shown in Fig. P10.16. The thermal diffuser receives a constant thermal flux of magnitude $q_1 = 400{,}000$ W/m$^2 \cdot$ °C from a solid-state device on which the diffuser is mounted. At the opposite end, the diffuser is kept at a uniform value of $T = 0°C$ by isothermalizing heat pipes. The lateral surface of the diffuser is insulated, and thermal conductivity $k = 200$ W/m·°C. The differential equation is

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{\partial^2 T}{\partial z^2} = 0$$

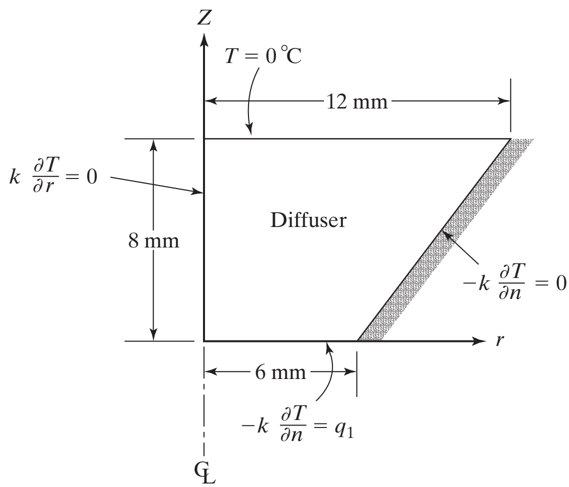


FIGURE P10.16

Develop an axisymmetric element to determine the temperature distribution and the outward heat flux at the heat pipes. Refer to Chapter 6 for details on the axisymmetric element.

10.17. Develop a four-node quadrilateral for heat conduction and solve problem 10.11. Refer to Chapter 7 for details pertaining to the quadrilateral element. Compare your solution with use of three-node triangles.

10.18. The L-shaped beam in Fig. P10.18, which supports a floor slab in a building, is subjected to a twisting moment T in./lb. Determine the following, using program TORSION:

(a) The angle of twist per unit length, α.

(b) The contribution of each finite element to the total twisting moment.

Give the answers in terms of torque T and shear modulus G. Verify the answers by refining the finite element grid.

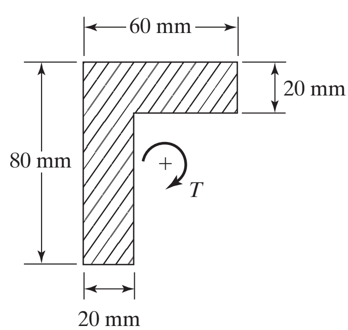


FIGURE P10.18

10.19. The cross section of the steel beam in Fig. P10.19 is subjected to a torque $T = 5000$ in./lb. Determine, using program TORSION, the angle of twist and the location and magnitude of the maximum shearing stresses.

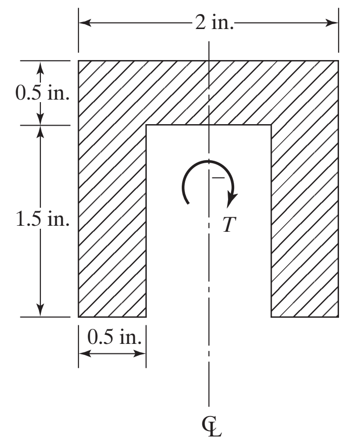


FIGURE P10.19

10.20. For Fig. 10.14a in the text, let $u^0 = 1$ m/s, $L = 5$ m, and $H = 2.0$ m. Determine the velocity field using a coarse grid and a fine grid (with smaller elements nearer the cylinder). In particular, determine the maximum velocity in the flow. Comment on the relation of this problem to a stress-concentration problem.

10.21. Determine and plot the stream lines for the flow in the venturi meter shown in Fig. P10.21. The incoming flow has a velocity of 100 cm/s. Also plot the velocity distribution at the waist *a–a*.

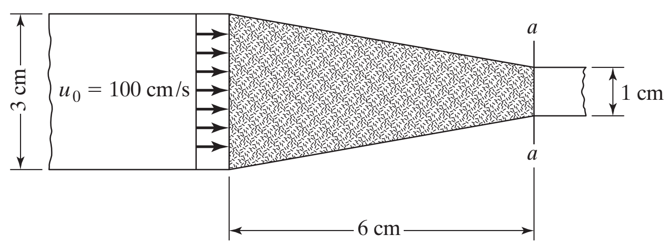


FIGURE P10.21

10.22. The dam shown in Fig. P10.22 rests on a homogeneous isotropic soil, which has confined impermeable boundaries as shown. The walls and base of the dam are impervious. The dam retains water at a constant height of 5 m, the downstream level being zero. Determine and plot the equipotential lines, and find the quantity of water seeping underneath the dam per unit width of the dam. Use hydraulic conductivity $k = 30$ m/day.

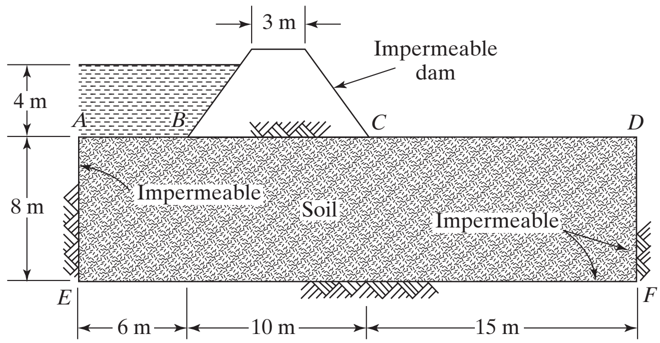


FIGURE P10.22

10.23. For the dam section shown in Fig. P10.23, $k = 0.003$ ft/min. Determine the following:
(a) The line of seepage.
(b) The quantity of seepage per 100 ft length of the dam.
(c) The length of the surface of seepage a.

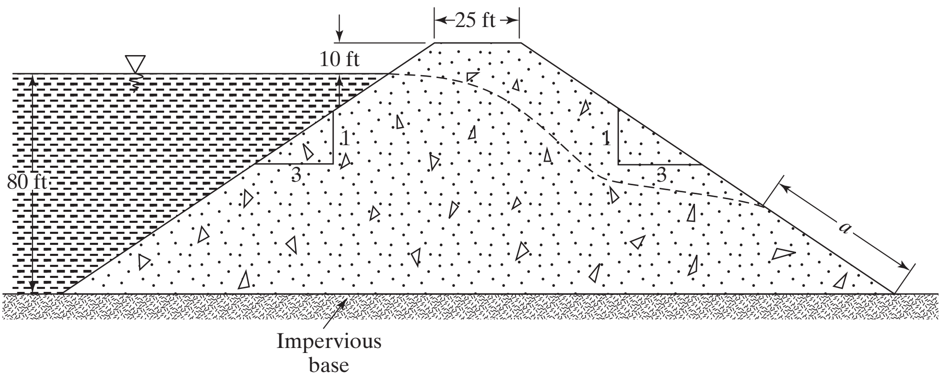


FIGURE P10.23

10.24. For the triangular duct shown in Fig. P10.24, obtain the constant C, which relates the Fanning friction factor f to the Reynolds number R_e as $f = C/R_e$. Use triangular finite elements. Verify the answer by refining the finite element model. Compare the results for C with that to a square duct having the same perimeter.

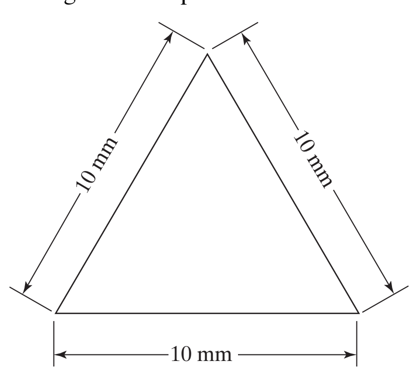


FIGURE P10.24

10.25. Figure P10.25 shows the cross section of a rectangular coaxial cable. At the inner surface of the dielectric insulator ($\epsilon_R = 3$), a voltage of 100 V is applied. If the voltage at the outer surface is zero, determine the voltage field distribution in the annular space.

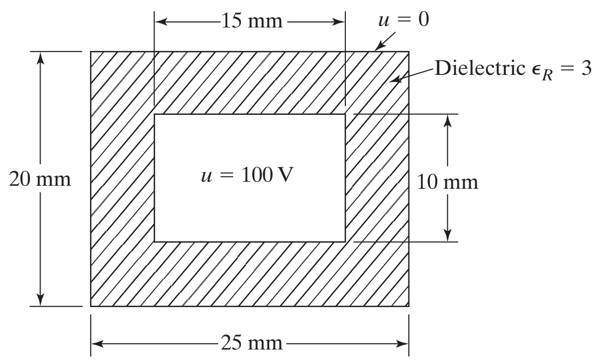


FIGURE P10.25

10.26. A pair of strip lines, shown in Fig. P10.26, is separated by a dielectric medium $\epsilon_R = 5.4$. The strips are enclosed by a fictitious box 2×1 m with enclosed space having $\epsilon_R = 1$. Assuming $u = 0$ on the boundary of this box, find the voltage field distribution. (Use large elements away from the strip plates.)

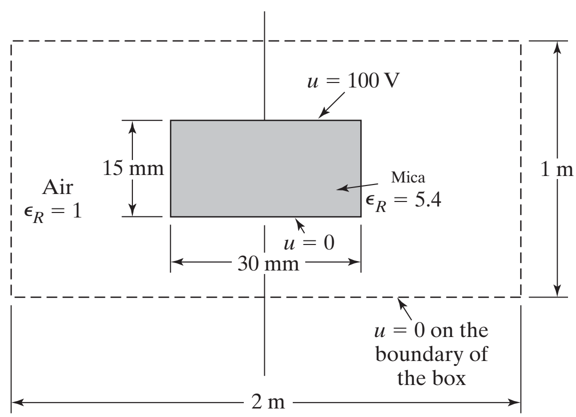


FIGURE P10.26

10.27. Determine the scalar magnetic potential u for the simplified model of the slot in an electric motor armature shown in Fig. P10.27.

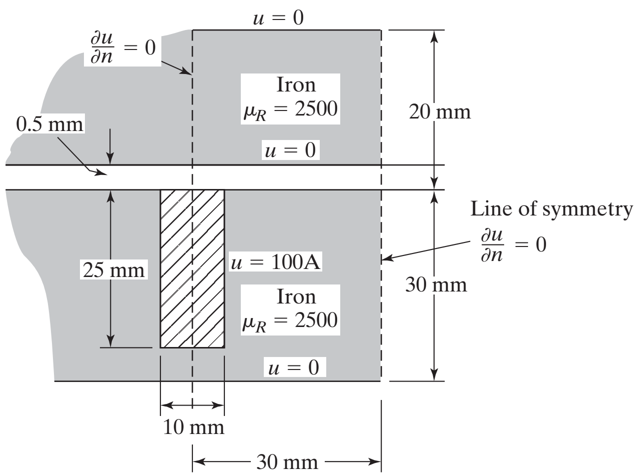


FIGURE P10.27

10.28. Repeat Example 10.6 with
(a) 12 elements
(b) 24 elements
(c) 48 elements.
Plot convergence curves of frequency, cps vs number of elements.

10.29. Consider a tube of length 6 m as in Fig. P10.29. Fluid in the tube is air; thus, $c = 343$ m/s. One end is rigid while the other end is a pressure-release condition, as shown. Determine the mode shapes and natural frequencies, and compare them with the analytical solution $f_n = [m + (1/2)]c/(2L)$. Adopt a six-element model.

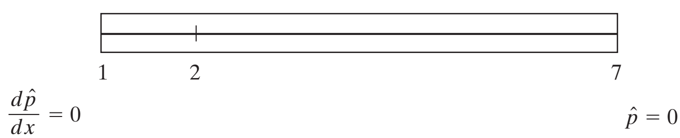


FIGURE P10.29

10.30. Derive the element matrices in Eq. 10.130 from Galerkin's variational statement in Eq. 10.128.

10.31. Solve the eigenvalue problem resulting from the two-dimensional problem in Eq. 10.129 and solve for the first four modes of the rectangular room/cavity shown in Fig. P10.31. Dimensions of the cavity are $L_x, L_y, L_z = (20\text{ m}, 10\text{ m}, 0.1\text{ m})$. Provide plots of the mode shapes, and compare the natural frequencies with the analytical solution $c = 343$ m/s. Try a coarse mesh and a fine mesh. Use the equation

$$f_{\ell,m,n} = \frac{c}{2}\sqrt{\left(\frac{\ell}{L_x}\right)^2 + \left(\frac{m}{L_y}\right)^2 + \left(\frac{n}{L_z}\right)^2} \quad \text{cps}, \qquad \ell, m, n = 0, 1, 2, \ldots.$$

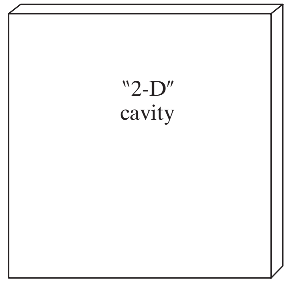


FIGURE P10.31

10.32. A single fin is attached to a base plate to remove heat generated by a computer chip underneath. The power generated by the chip is 8 W. Treating this as a uniform heat flux into the base, determine the maximum temperature at the base (Fig. P10.32). Use the following data:

Fin volume (w^2L) $= 125\ \text{mm}^3$

Try with $w = 2$ and $w = 5$ and compare

$h_{\text{base}} = 100\ \text{W/m}^2\cdot{}^\circ\text{C}$, $h_{\text{fin}} = 200\ \text{W/m}^2\cdot{}^\circ\text{C}$, $T_\infty = 25^\circ\text{C}$

Material: aluminum, thermal conductivity $k = 210\ \text{W/m}\cdot{}^\circ\text{C}$

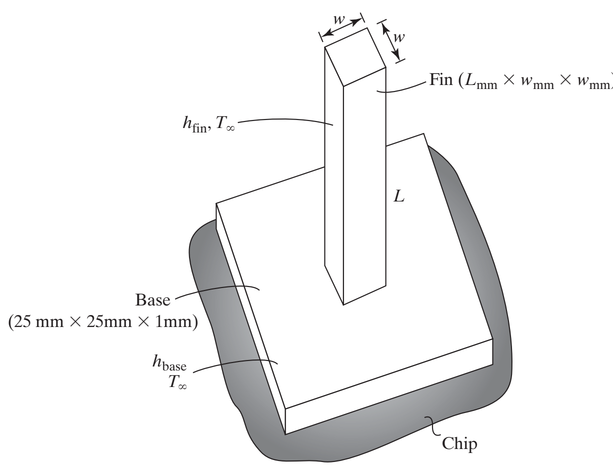


FIGURE P10.32

10.33. Repeat problem 10.32 with a five-pin configuration keeping the fin volume constant at $125\ \text{mm}^3$. Use $w = 2$ mm.

Program Listing

```
MAIN PROGRAM HEAT1D
'****************************************
'*           PROGRAM HEAT1D             *
'* T.R.Chandrupatla and A.D.Belegundu   *
'****************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call Stiffness
     Call ModifyForBC
     Call BandSolver
     Call Output
End Sub
```

```
HEAT1D - INPUT FROM SHEET1 FOR EXCEL (from file for C, FORTRAN, MATLAB)
Private Sub InputData()
    NE = Val(Worksheets(1).Range("A4"))
    NBC = Val(Worksheets(1).Range("B4"))
    NQ = Val(Worksheets(1).Range("C4"))
    NN = NE + 1
    '--- NBW is half the bandwidth Elements 1-2,2-3,3-4,...
    NBW = 2
    ReDim X(NN), S(NN, NBW), TC(NE), F(NN), V(NBC), H(NBC), NB(NBC)
    ReDim BC(NBC)
    LI = 0
    '----- Elem# Thermal Conductivity -----
     For I = 1 To NE
        LI = LI + 1
        N = Val(Worksheets(1).Range("A5").Offset(LI, 0))
        TC(N) = Val(Worksheets(1).Range("A5").Offset(LI, 1))
     Next I
    '----- Coordinates -----
    LI = LI + 1
    For I = 1 To NN
        LI = LI + 1
        N = Val(Worksheets(1).Range("A5").Offset(LI, 0))
        X(N) = Val(Worksheets(1).Range("A5").Offset(LI, 1))
    Next I
    '----- Boundary Conditions -----
    LI = LI + 1
    For I = 1 To NBC
        LI = LI + 1
        NB(I) = Val(Worksheets(1).Range("A5").Offset(LI, 0))
        BC(I) = Worksheets(1).Range("A5").Offset(LI, 1)
        LI = LI + 1
        If BC(I) = "TEMP" Or BC(I) = "temp" Then
           V(I) = Val(Worksheets(1).Range("A5").Offset(LI, 0))
        End If
        If BC(I) = "FLUX" Or BC(I) = "flux" Then
           V(I) = Val(Worksheets(1).Range("A5").Offset(LI, 0))
        End If
```

```
        If BC(I) = "CONV" Or BC(I) = "conv" Then
           H(I) = Val(Worksheets(1).Range("A5").Offset(LI, 0))
           V(I) = Val(Worksheets(1).Range("A5").Offset(LI, 1))
        End If
    Next I
    '----- Calculate and Input Nodal Heat Source Vector -----
    LI = LI + 1
    For I = 1 To NN: F(I) = 0: Next I
    If NQ > 0 Then
        For I = 1 To NQ
           LI = LI + 1
           N = Val(Worksheets(1).Range("A7").Offset(LI, 0))
           F(N) = Val(Worksheets(1).Range("A7").Offset(LI, 1))
        Next I
    End If
End Sub
```

```
STIFFNESS – HEAT1D
Private Sub Stiffness()
     ReDim S(NN, NBW)
     '----- Stiffness Matrix -----
     For J = 1 To NBW
     For I = 1 To NN: S(I, J) = 0: Next I: Next J
     For I = 1 To NE
     I1 = I: I2 = I + 1
     ELL = Abs(X(I2) - X(I1))
     EKL = TC(I) / ELL
     S(I1, 1) = S(I1, 1) + EKL
     S(I2, 1) = S(I2, 1) + EKL
     S(I1, 2) = S(I1, 2) - EKL: Next I
End Sub
```

```
HEAT1D-MODIFICATION FOR BC
Private Sub ModifyForBC()
     '----- Decide Penalty Parameter CNST -----
     AMAX = 0
     For I = 1 To NN
        If S(I, 1) > AMAX Then AMAX = S(I, 1)
     Next I
     CNST = AMAX * 10000
     For I = 1 To NBC
        N = NB(I)
        If BC(I) = "CONV" Or BC(I) = "conv" Then
           S(N, 1) = S(N, 1) + H(I)
           F(N) = F(N) + H(I) * V(I)
        ElseIf BC(I) = "HFLUX" Or BC(I) = "hflux" Then
           F(N) = F(N) - V(I)
        Else
           S(N, 1) = S(N, 1) + CNST
           F(N) = F(N) + CNST * V(I)
        End If
     Next I
End Sub
```

MAIN PROGRAM HEAT2D

```
'****************************************
'*            PROGRAM HEAT2D            *
'*   HEAT 2-D WITH 3-NODED TRIANGLES    *
'* T.R.Chandrupatla and A.D.Belegundu   *
'****************************************
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call Stiffness
     Call ModifyForBC
     Call BandSolver
     Call HeatFlowCalc
     Call Output
End Sub
```

HEAT2D – INPUT FROM SHEET1 EXCEL (from file for C, FORTRAN, MATLAB)

```
Private Sub InputData()
     Dim msg As String
     msg = " 1) No Plot Data" & Chr(13)
     msg = msg + " 2) Create Data File Containing Nodal TempS" & Chr(13)
     msg = msg + "  Choose 1 or 2"
     IPL = InputBox(msg, "Plot Choice", 1)        '--- default is no data
     NN = Val(Worksheets(1).Range("A4"))
     NE = Val(Worksheets(1).Range("B4"))
     NM = Val(Worksheets(1).Range("C4"))
     NDIM = Val(Worksheets(1).Range("D4"))
     NEN = Val(Worksheets(1).Range("E4"))
     NDN = Val(Worksheets(1).Range("F4"))
     ND = Val(Worksheets(1).Range("A6"))
     NL = Val(Worksheets(1).Range("B6"))
     NMPC = Val(Worksheets(1).Range("C6"))
     NPR = 1 'One Material Property Thermal Conductivity
     NMPC = 0
     '--- ND = NO. OF SPECIFIED TEMPERATURES
     '--- NL = NO. OF NODAL HEAT SOURCES
     'NOTE!! NPR =1 (THERMAL CONDUCTIVITY) AND NMPC = 0 FOR THIS PROGRAM
     '--- EHS(I) = ELEMENT HEAT SOURCE, I = 1,...,NE
     ReDim X(NN, 2), NOC(NE, 3), MAT(NE), PM(NM, NPR), F(NN)
     ReDim NU(ND), U(ND), EHS(NE)
     '============= READ DATA      ===============
      LI = 0
     '----- Coordinates -----
     For I = 1 To NN
        LI = LI + 1
        N = Val(Worksheets(1).Range("A7").Offset(LI, 0))
        For J = 1 To NDIM
          X(N, J) = Val(Worksheets(1).Range("A7").Offset(LI, J))
        Next J
     Next I
      LI = LI + 1
```

```
'----- Connectivity, Material#, Element Heat Source
For I = 1 To NE
     LI = LI + 1
     N = Val(Worksheets(1).Range("A7").Offset(LI, 0))
   For J = 1 To NEN
     NOC(N, J) = Val(Worksheets(1).Range("A7").Offset(LI, J))
   Next J
     MAT(N) = Val(Worksheets(1).Range("A7").Offset(LI, NEN + 1))
     EHS(N) = Val(Worksheets(1).Range("A7").Offset(LI, NEN + 2))
 Next I
 '----- Temperature BC -----
 LI = LI + 1
 For I = 1 To ND
    LI = LI + 1
    NU(I) = Val(Worksheets(1).Range("A7").Offset(LI, 0))
    U(I) = Val(Worksheets(1).Range("A7").Offset(LI, 1))
 Next I
 '----- Nodal Heat Sources -----
 LI = LI + 1
 For I = 1 To NL
    LI = LI + 1
    N = Val(Worksheets(1).Range("A7").Offset(LI, 0))
    F(N) = Val(Worksheets(1).Range("A7").Offset(LI, 1))
 Next I
 LI = LI + 1
 '----- Thermal Conductivity of Material -----
 For I = 1 To NM
    LI = LI + 1
    N = Val(Worksheets(1).Range("A7").Offset(LI, 0))
    For J = 1 To NPR
        PM(N, J) = Val(Worksheets(1).Range("A7").Offset(LI, J))
    Next J
 Next I
 'No. of edges with Specified Heat flux
 'FOLLOWED BY two edges & q0 (positive if out)
 LI = LI + 1
 LI = LI + 1
 NHF = Val(Worksheets(1).Range("A7").Offset(LI, 0))
 If NHF > 0 Then
    ReDim NFLUX(NHF, 2), FLUX(NHF)
    For I = 1 To NHF
        LI = LI + 1
        NFLUX(I, 1) = Val(Worksheets(1).Range("A7").Offset(LI, 0))
        NFLUX(I, 2) = Val(Worksheets(1).Range("A7").Offset(LI, 1))
        FLUX(I) = Val(Worksheets(1).Range("A7").Offset(LI, 2))
    Next I
 End If
 'No. of Edges with Convection FOLLOWED BY edge(2 nodes) &
 'h & Tinf
 LI = LI + 1
 LI = LI + 1
 NCV = Val(Worksheets(1).Range("A7").Offset(LI, 0))
 If NCV > 0 Then
    ReDim NCONV(NCV, 2), H(NCV), TINF(NCV)
```

```
      For I = 1 To NCV
         LI = LI + 1
         NCONV(I, 1) = Val(Worksheets(1).Range("A7").Offset(LI, 0))
         NCONV(I, 2) = Val(Worksheets(1).Range("A7").Offset(LI, 1))
         H(I) = Val(Worksheets(1).Range("A7").Offset(LI, 2))
         TINF(I) = Val(Worksheets(1).Range("A7").Offset(LI, 3))
      Next I
   End If
End Sub
```

```
STIFFNESS - HEAT2D
Private Sub Stiffness()
     '----- Initialization of Conductivity Matrix and Heat Rate Vector
     ReDim S(NN, NBW)
     For I = 1 To NN
        For J = 1 To NBW
           S(I, J) = 0
        Next J
     Next I
     If NHF > 0 Then
        For I = 1 To NHF
           N1 = NFLUX(I, 1): N2 = NFLUX(I, 2)
           V = FLUX(I)
           ELEN = Sqr((X(N1, 1)-X(N2, 1))^2 + (X(N1, 2)-X(N2, 2))^2)
           F(N1) = F(N1) - ELEN * V /2
           F(N2) = F(N2) - ELEN * V / 2
        Next I
     End If
     If NCV > 0 Then
        For I = 1 To NCV
           N1 = NCONV(I, 1): N2 = NCONV(I, 2)
           ELEN = Sqr((X(N1, 1)-X(N2, 1))^2 + (X(N1, 2)-X(N2, 2))2)
           F(N1) = F(N1) + ELEN * H(I) * TINF(I) / 2
           F(N2) = F(N2) + ELEN * H(I) * TINF(I) / 2
           S(N1, 1) = S(N1, 1) + H(I) * ELEN / 3
           S(N2, 1) = S(N2, 1) + H(I) * ELEN / 3
           If N1 >= N2 Then
            N3 = N1: N1 = N2: N2 = N3
           End If
           S(N1, N2 - N1 + 1) = S(N1, N2 - N1 + 1) + H(I) * ELEN / 6
        Next I
     End If
     '----- Conductivity Matrix
     ReDim BT(2, 3)
        For I = 1 To NE
           I1 = NOC(I, 1): I2 = NOC(I, 2): I3 = NOC(I, 3)
           X32 = X(I3, 1) - X(I2, 1): X13 = X(I1, 1) - X(I3, 1)
           X21 = X(I2, 1) - X(I1, 1)
           Y23 = X(I2, 2) - X(I3, 2): Y31 = X(I3, 2) - X(I1, 2)
           Y12 = X(I1, 2) - X(I2, 2)
           DETJ = X13 * Y23 - X32 * Y31
           AREA = 0.5 * Abs(DETJ)
```

```
         '--- Element Heat Sources
         If EHS(I) <> 0 Then
            C = EHS(I) * AREA / 3
            F(I1) = F(I1) + C: F(I2) = F(I2) + C: F(I3) = F(I3) + C
         End If
         BT(1, 1) = Y23: BT(1, 2) = Y31: BT(1, 3) = Y12
         BT(2, 1) = X32: BT(2, 2) = X13: BT(2, 3) = X21
         For II = 1 To 3
            For JJ = 1 To 2
               BT(JJ, II) = BT(JJ, II) / DETJ
            Next JJ
         Next II
         For II = 1 To 3
            For JJ = 1 To 3
               II1 = NOC(I, II): II2 = NOC(I, JJ)
               If II1 <= II2 Then
                  Sum = 0
                  For J = 1 To 2
                     Sum = Sum + BT(J, II) * BT(J, JJ)
                  Next J
                  IC = II2 - II1 + 1
                  S(II1, IC) = S(II1, IC) + Sum * AREA * PM(MAT(I), 1)
               End If
            Next JJ
         Next II
     Next I
End Sub
```

```
HEAT FLOW CALCULATIONS
Private Sub HeatFlowCalc()
     ReDim Q(NE, 2)
     For I = 1 To NE
        I1 = NOC(I, 1): I2 = NOC(I, 2): I3 = NOC(I, 3)
        X32 = X(I3, 1) - X(I2, 1): X13 = X(I1, 1) - X(I3, 1)
        X21 = X(I2, 1) - X(I1, 1)
        Y23 = X(I2, 2) - X(I3, 2): Y31 = X(I3, 2) - X(I1, 2)
        Y12 = X(I1, 2) - X(I2, 2)
        DETJ = X13 * Y23 - X32 * Y31
        BT(1, 1) = Y23: BT(1, 2) = Y31: BT(1, 3) = Y12
        BT(2, 1) = X32: BT(2, 2) = X13: BT(2, 3) = X21
        For II = 1 To 3
           For JJ = 1 To 2
              BT(JJ, II) = BT(JJ, II) / DETJ
           Next JJ
        Next II
        QX = BT(1, 1) * F(I1) + BT(1, 2) * F(I2) + BT(1, 3) * F(I3)
        QX = -QX * PM(MAT(I), 1)
        QY = BT(2, 1) * F(I1) + BT(2, 2) * F(I2) + BT(2, 3) * F(I3)
        QY = -QY * PM(MAT(I), 1)
        Q(I, 1) = QX
        Q(I, 2) = QY
     Next I
End Sub
```

CHAPTER 11

Dynamic Considerations

11.1 INTRODUCTION

In Chapters 3–9, we have discussed the static analysis of structures. Static analysis holds when the loads are slowly applied. When the loads are suddenly applied, or when the loads are of a variable nature, the mass and acceleration effects come into the picture. If a solid body, such as an engineering structure, is deformed elastically and suddenly released, it tends to vibrate about its equilibrium position. This periodic motion due to the restoring strain energy is called **free vibration**. The number of cycles per unit time is called the **frequency**. The maximum displacement from the equilibrium position is the **amplitude**. In the real world, the vibrations subside with time due to damping action. In the simplest vibration model, the damping effects are neglected. The undamped free-vibration model of a structure gives significant information about its dynamic behavior. We present here the considerations needed to apply finite elements to the analysis of undamped free vibrations of structures.

11.2 FORMULATION

We define the Lagrangean by

$$L = T - \Pi \tag{11.1}$$

where T is the kinetic energy and Π is the potential energy.

Hamilton's Principle For an arbitrary time interval from t_1 to t_2, the state of motion of a body extremizes the functional

$$I = \int_{t_1}^{t_2} L\,dt \tag{11.2}$$

If L can be expressed in terms of the generalized variables $(q_1, q_2, \ldots, q_n, \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n)$ where $\dot{q}_i = dq_i/dt$, then the equations of motion are given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0 \qquad i = 1 \text{ to } n \tag{11.3}$$

To illustrate the principle, let us consider two point masses connected by springs. Consideration of distributed masses follows the example.

Example 11.1

Consider the spring–mass system in Fig. E11.1. The kinetic and potential energies are given by

$$T = \tfrac{1}{2}m_1\dot{x}_1^2 + \tfrac{1}{2}m_2\dot{x}_2^2$$
$$\Pi = \tfrac{1}{2}k_1x_1^2 + \tfrac{1}{2}k_2(x_2 - x_1)^2$$

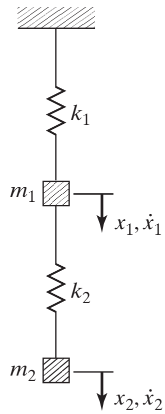


FIGURE E11.1

Using $L = T - \Pi$, we obtain the equations of motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_1}\right) - \frac{\partial L}{\partial x_1} = m_1\ddot{x}_1 + k_1x_1 - k_2(x_2 - x_1) = 0$$
$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_2}\right) - \frac{\partial L}{\partial x_2} = m_2\ddot{x}_2 + k_2(x_2 - x_1) = 0$$

which can be written in the form

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} (k_1 + k_2) & -k_2 \\ -k_2 & k_2 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \mathbf{0}$$

which is of the form

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{0} \tag{11.4}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, and $\ddot{\mathbf{x}}$ and $\mathbf{x}$ are vectors representing accelerations and displacements. ■

Solid Body with Distributed Mass

We now consider a solid body with distributed mass (Fig. 11.1). The potential energy term has already been considered in Chapter 1. The kinetic energy is given by

$$T = \frac{1}{2}\int_V \dot{\mathbf{u}}^{\mathrm{T}}\dot{\mathbf{u}}\rho\, dV \tag{11.5}$$

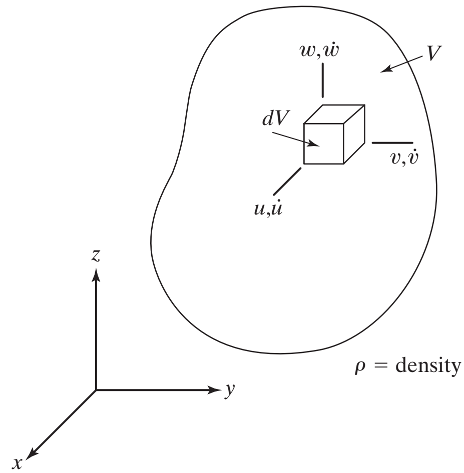


FIGURE 11.1 Body with distributed mass.

where ρ is the density (mass per unit volume) of the material and

$$\dot{\mathbf{u}} = [\dot{u} \quad \dot{v} \quad \dot{w}]^{\mathrm{T}} \tag{11.6}$$

is the velocity vector of the point at $\mathbf{x}$, with components $\dot{u}$, $\dot{v}$, and $\dot{w}$. In the finite element method, we divide the body into elements, and in each element, we express $\mathbf{u}$ in terms of the nodal displacements $\mathbf{q}$, using shape functions $\mathbf{N}$. Thus,

$$\mathbf{u} = \mathbf{N}\mathbf{q} \tag{11.7}$$

In dynamic analysis, the elements of $\mathbf{q}$ are dependent on time, while $\mathbf{N}$ represents (spatial) shape functions defined on a master element. The velocity vector is then given by

$$\dot{\mathbf{u}} = \mathbf{N}\dot{\mathbf{q}} \tag{11.8}$$

When Eq. 11.8 is substituted into Eq. 11.5, the kinetic energy T_e in element e is

$$T_e = \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\left[\int_e \rho\mathbf{N}^{\mathrm{T}}\mathbf{N}\,dV\right]\dot{\mathbf{q}} \tag{11.9}$$

where the bracketed expression is the element mass matrix

$$\mathbf{m}^e = \int_e \rho\mathbf{N}^{\mathrm{T}}\mathbf{N}\,dV \tag{11.10}$$

This mass matrix is consistent with the shape functions chosen and is called the *consistent mass matrix*. Mass matrices for various elements are given in the next section. On summing over all the elements, we get

$$T = \sum_e T_e = \sum_e \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{m}^e\dot{\mathbf{q}} = \frac{1}{2}\dot{\mathbf{Q}}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{Q}} \tag{11.11}$$

The potential energy is given by

$$\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} - \mathbf{Q}^{\mathrm{T}}\mathbf{F} \tag{11.12}$$

Using the Lagrangean $L = T - \Pi$, we obtain the equation of motion:

$$\mathbf{M}\ddot{\mathbf{Q}} + \mathbf{K}\mathbf{Q} = \mathbf{F} \tag{11.13}$$

For free vibrations the force $\mathbf{F}$ is zero. Thus,

$$\mathbf{M}\ddot{\mathbf{Q}} + \mathbf{K}\mathbf{Q} = \mathbf{0} \tag{11.14}$$

For the steady-state condition, starting from the equilibrium state, we set

$$\mathbf{Q} = \mathbf{U}\sin\omega t \tag{11.15}$$

where $\mathbf{U}$ is the vector of nodal amplitudes of vibration and ω (rad/s) is the circular frequency ($=2\pi f$, f = cycles/s or Hz). Introducing Eq. 11.15 into Eq. 11.14, we have

$$\mathbf{K}\mathbf{U} = \omega^2\mathbf{M}\mathbf{U} \tag{11.16}$$

This is the generalized eigenvalue problem

$$\mathbf{K}\mathbf{U} = \lambda\mathbf{M}\mathbf{U} \tag{11.17}$$

where $\mathbf{U}$ is the eigenvector, representing the vibrating mode, corresponding to the eigenvalue λ. The eigenvalue λ is the square of the circular frequency ω. The frequency f in hertz (cycles per second) is obtained from $f = \omega/(2\pi)$.

The previous equations can also be obtained by using D'Alembert's principle and the principle of virtual work. Galerkin's approach applied to equations of motion of an elastic body also yields this set of equations.

11.3 ELEMENT MASS MATRICES

Since the shape functions for various elements have been discussed in detail in the earlier chapters, we now give the element mass matrices. Treating the material density ρ to be constant over the element, we have, from Eq. 11.10,

$$\mathbf{m}^e = \rho\int_e \mathbf{N}^{\mathrm{T}}\mathbf{N}\,dV \tag{11.18}$$

One-Dimensional Bar Element For the bar element shown in Fig. 11.2,

$$\begin{aligned}\mathbf{q}^{\mathrm{T}} &= [q_1 \;\; q_2]\\ \mathbf{N} &= [N_1 \;\; N_2]\end{aligned} \tag{11.19}$$

where

$$N_1 = \frac{1-\xi}{2} \qquad N_2 = \frac{1+\xi}{2}$$

$$\mathbf{m}^e = \rho\int_e \mathbf{N}^{\mathrm{T}}\mathbf{N}A\,dx = \frac{\rho A_e \ell_e}{2}\int_{-1}^{+1}\mathbf{N}^{\mathrm{T}}\mathbf{N}\,d\xi$$

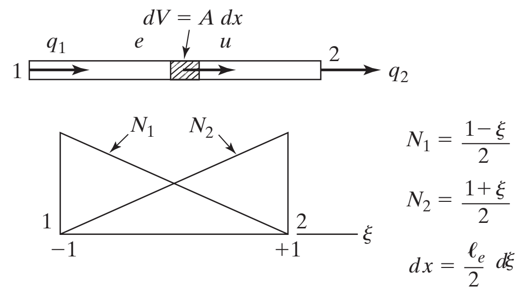


FIGURE 11.2 Bar element.

On carrying out the integration of each term in $\mathbf{N}^{\mathrm{T}}\mathbf{N}$, we find that

$$\mathbf{m}^e = \frac{\rho A_e \ell_e}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{11.20}$$

Truss Element For the truss element shown in Fig. 11.3,

$$\begin{aligned} \mathbf{u}^{\mathrm{T}} &= [u \;\; v] \\ q^{\mathrm{T}} &= [q_1 \;\; q_2 \;\; q_3 \;\; q_4] \\ \mathbf{N} &= \begin{bmatrix} N_1 & 0 & N_2 & 0 \\ 0 & N_1 & 0 & N_2 \end{bmatrix} \end{aligned} \tag{11.21}$$

where

$$N_1 = \frac{1-\xi}{2} \quad \text{and} \quad N_2 = \frac{1+\xi}{2}$$

in which ξ is defined from –1 to + 1. Then,

$$\mathbf{m}^e = \frac{\rho A_e \ell_e}{6}\begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \tag{11.22}$$

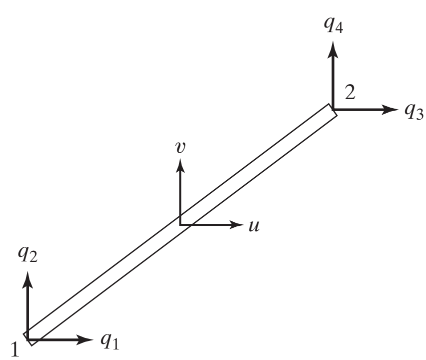


FIGURE 11.3 Truss element.

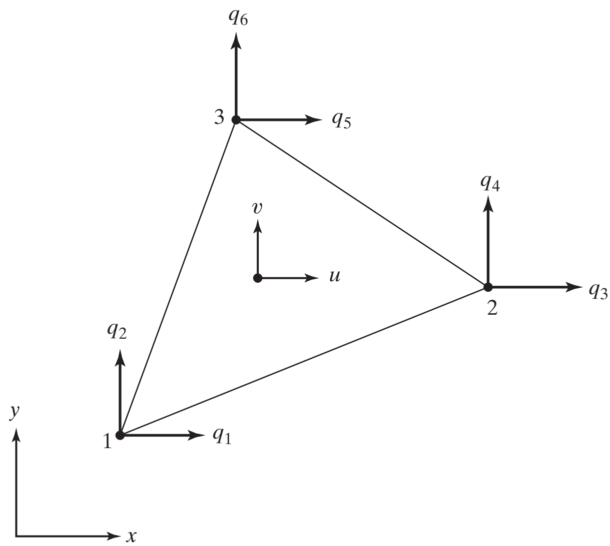


FIGURE 11.4 CST element.

CST Element For the plane stress and plane strain conditions for the CST element shown in Fig. 11.4, we have, from Chapter 5,

$$\mathbf{u}^{\mathrm{T}} = [u \quad v]$$
$$\mathbf{q}^{\mathrm{T}} = [q_1 \quad q_2 \quad \cdots \quad q_6] \tag{11.23}$$
$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}$$

The element mass matrix is then given by

$$\mathbf{m}^e = \rho t_e \int_e \mathbf{N}^{\mathrm{T}}\mathbf{N}\, dA$$

Noting that $\int_e N_1^2\, dA = \frac{1}{6}A_e, \int_e N_1 N_2\, dA = \frac{1}{12}A_e$, etc., we have

$$\mathbf{m}^e = \frac{\rho t_e A_e}{12}\begin{bmatrix} 2 & 0 & 1 & 0 & 1 & 0 \\ & 2 & 0 & 1 & 0 & 1 \\ & & 2 & 0 & 1 & 0 \\ & & & 2 & 0 & 1 \\ & & & & 2 & 0 \\ \text{Symmetric} & & & & & 2 \end{bmatrix} \tag{11.24}$$

Axisymmetric Triangular Element For the axisymmetric triangular element, we have

$$\mathbf{u}^{\mathrm{T}} = [u \quad w]$$

where u and w are the radial and axial displacements, respectively. The vectors $\mathbf{q}$ and $\mathbf{N}$ are similar to those for the triangular element given in Eq. 11.23. We have

$$\mathbf{m}^e = \int_e \rho \mathbf{N}^{\mathrm{T}}\mathbf{N}\, dV = \int_e \rho \mathbf{N}^{\mathrm{T}}\mathbf{N} 2\pi r\, dA \tag{11.25}$$

Since $r = N_1 r_1 + N_2 r_2 + N_3 r_3$, we have

$$\mathbf{m}^e = 2\pi\rho \int_e (N_1 r_1 + N_2 r_2 + N_3 r_3)\mathbf{N}^{\mathrm{T}}\mathbf{N}\, dA$$

Noting that

$$\int_e N_1^3\, dA = \frac{2A_e}{20}, \qquad \int_e N_1^2 N_2\, dA = \frac{2A_e}{60}, \qquad \int_e N_1 N_2 N_3\, dA = \frac{2A_e}{120}, \text{ etc.}$$

we get

$$\mathbf{m}_e = \frac{\pi\rho A_e}{10}\begin{bmatrix} \frac{4}{3}r_1 + 2\bar{r} & 0 & 2\bar{r} - \frac{r_3}{3} & 0 & 2\bar{r} - \frac{r_2}{3} & 0 \\ & \frac{4}{3}r_1 + 2\bar{r} & 0 & 2\bar{r} - \frac{r_3}{3} & 0 & 2\bar{r} - \frac{r_2}{3} \\ & & \frac{4}{3}r_2 + 2\bar{r} & 0 & 2\bar{r} - \frac{r_1}{3} & 0 \\ & & & \frac{4}{3}r_2 + 2\bar{r} & 0 & 2\bar{r} - \frac{r_1}{3} \\ \text{Symmetric} & & & & \frac{4}{3}r_3 + 2\bar{r} & 0 \\ & & & & & \frac{4}{3}r_3 + 2\bar{r} \end{bmatrix} \tag{11.26}$$

where

$$\bar{r} = \frac{r_1 + r_2 + r_3}{3} \tag{11.27}$$

Quadrilateral Element For the quadrilateral element for plane stress and plane strain,

$$\begin{aligned} \mathbf{u}^{\mathrm{T}} &= [u \;\; v] \\ \mathbf{q}^{\mathrm{T}} &= [q_1 \;\; q_2 \;\; \cdots \;\; q_8] \\ \mathbf{N} &= \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \end{aligned} \tag{11.28}$$

The mass matrix is then given by

$$\mathbf{m}^e = \rho t_e \int_{-1}^{1}\int_{-1}^{1} \mathbf{N}^{\mathrm{T}}\mathbf{N} \det \mathbf{J}\, d\xi\, d\eta \tag{11.29}$$

This integral needs to be evaluated by numerical integration.

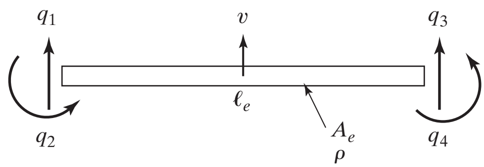


FIGURE 11.5 Beam element.

Beam Element For the beam element shown in Fig. 11.5, we use the Hermite shape functions given in Chapter 5. We have

$$v = \mathbf{Hq}$$

$$\mathbf{m}^e = \int_{-1}^{+1} \mathbf{H}^{\mathrm{T}}\mathbf{H}\rho A_e \frac{\ell_e}{2}\, d\xi \tag{11.30}$$

On integrating, we get

$$\mathbf{m}^e = \frac{\rho A_e \ell_e}{420}\begin{bmatrix} 156 & 22\ell_e & 54 & -13\ell_e \\ & 4\ell_e^2 & 13\ell_e & -3\ell_e^2 \\ \text{Symmetric} & & 156 & -22\ell_e \\ & & & 4\ell_e^2 \end{bmatrix} \tag{11.31}$$

Frame Element We refer to Fig. 5.11, showing the frame element. In the body coordinate system x', y', the mass matrix can be seen as a combination of bar element and beam element. Thus, the mass matrix in the prime system is given by

$$\mathbf{m}^{e'} = \begin{bmatrix} 2a & 0 & 0 & a & 0 & 0 \\ & 156b & 22\ell_e^2 b & 0 & 54b & -13\ell_e b \\ & & 4\ell_e^2 b & 0 & 13\ell_e b & -3\ell_e^2 b \\ & & & 2a & 0 & 0 \\ \text{Symmetric} & & & & 156b & -22\ell_e b \\ & & & & & 4\ell_e^2 b \end{bmatrix} \tag{11.32}$$

where

$$a = \frac{\rho A_e \ell_e}{6} \quad \text{and} \quad b = \frac{\rho A_e \ell_e}{420}$$

Using the transformation matrix $\mathbf{L}$ given by Eq. 5.48, we now obtain the mass matrix $\mathbf{m}^e$ in the global system:

$$\mathbf{m}^e = \mathbf{L}^{\mathrm{T}}\mathbf{m}^{e'}\mathbf{L} \tag{11.33}$$

Tetrahedral Element For the tetrahedral element presented in Chapter 9,

$$\mathbf{u}^{\mathrm{T}} = [u \;\; v \;\; w]$$

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \tag{11.34}$$

The mass matrix of the element is then given by

$$\mathbf{m}^e = \frac{\rho V_e}{20}\begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ & & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ & & & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & & & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ & \text{Symmetric} & & & & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ & & & & & & 2 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & & 2 & 0 & 0 & 1 & 0 \\ & & & & & & & & 2 & 0 & 0 & 1 \\ & & & & & & & & & 2 & 0 & 0 \\ & & & & & & & & & & 2 & 0 \\ & & & & & & & & & & & 2 \end{bmatrix} \tag{11.35}$$

Lumped Mass Matrices Consistent mass matrices have been presented. Practicing engineers also use lumped mass techniques, where the total element mass in each direction is distributed equally to the nodes of the element, and the masses are associated with translational degrees of freedom (dof) only. For the truss element, the lumped mass approach gives a mass matrix of

$$\mathbf{m}^e = \frac{\rho A_e \ell_e}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ \text{Symmetric} & & & 1 \end{bmatrix} \tag{11.36}$$

For the beam element, the lumped element mass matrix is

$$\mathbf{m}^e = \frac{\rho A_e \ell_e}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ & & 1 & 0 \\ \text{Symmetric} & & & 0 \end{bmatrix} \tag{11.37}$$

Consistent mass matrices yield more accurate results for flexural elements such as beams. The lumped mass technique is easier to handle since only diagonal elements are involved. The natural frequencies obtained from lumped mass techniques are lower than the exact values. In our presentation, we discuss techniques for determining the eigenvalues and eigenvectors with consistent mass formulations. The programs presented can be used for lumped mass cases also.

11.4 EVALUATION OF EIGENVALUES AND EIGENVECTORS

The generalized problem in free vibration is that of evaluating an eigenvalue $\lambda(=\omega^2)$, which is a measure of the frequency of vibration together with the corresponding eigenvector $\mathbf{U}$ indicating the mode shape, as in Eq. 11.17, restated here:

$$\mathbf{KU} = \lambda\mathbf{MU} \tag{11.38}$$

We observe here that $\mathbf{K}$ and $\mathbf{M}$ are symmetric matrices. Further, $\mathbf{K}$ is positive definite for properly constrained problems.

Properties of Eigenvectors

For a positive definite symmetric stiffness matrix of size n, there are n real eigenvalues and corresponding eigenvectors satisfying Eq. 11.38. The eigenvalues may be arranged in ascending order:

$$0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \tag{11.39}$$

If $\mathbf{U}_1, \mathbf{U}_2, \ldots \mathbf{U}_n$ are the corresponding eigenvectors, we have

$$\mathbf{KU}_i = \lambda_i \mathbf{MU}_i \tag{11.40}$$

The eigenvectors possess the property of being orthogonal with respect to both the stiffness and mass matrices:

$$\mathbf{U}_i^{\mathrm{T}} \mathbf{MU}_j = 0 \quad \text{if } i \neq j \tag{11.41a}$$

$$\mathbf{U}_i^{\mathrm{T}} \mathbf{KU}_j = 0 \quad \text{if } i \neq j \tag{11.41b}$$

The lengths of eigenvectors are generally normalized so that

$$\mathbf{U}_i^{\mathrm{T}} \mathbf{MU}_i = 1 \tag{11.42a}$$

The foregoing normalization of the eigenvectors leads to the relation

$$\mathbf{U}_i^{\mathrm{T}} \mathbf{KU}_i = \lambda_i \tag{11.42b}$$

In many codes, other normalization schemes are also used. The length of an eigenvector may be fixed by setting its largest component to a preset value, say, unity.

Eigenvalue–Eigenvector Evaluation

The eigenvalue–eigenvector evaluation procedures fall into the following basic categories:

1. Characteristic polynomial technique
2. Vector iteration methods
3. Transformation methods

Characteristic Polynomial. From Eq. 11.38, we have

$$(\mathbf{K} - \lambda \mathbf{M})\mathbf{U} = \mathbf{0} \tag{11.43}$$

If the eigenvector is to be nontrivial, the required condition is

$$\det(\mathbf{K} - \lambda \mathbf{M}) = \mathbf{0} \tag{11.44}$$

This represents the characteristic polynomial in λ.

Example 11.2

Determine the eigenvalues and eigenvectors for the stepped bar shown in Fig. E11.2a

Solution Gathering the stiffness and mass values corresponding to the dof Q_2 and Q_3, we get the eigenvalue problem

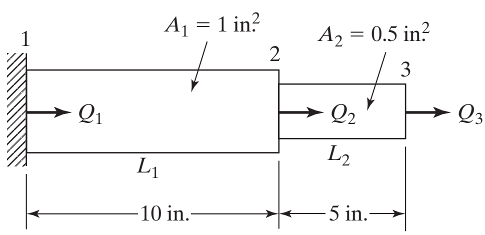
$$E\begin{bmatrix} \left(\frac{A_1}{L_1}+\frac{A_2}{L_2}\right) & -\frac{A_2}{L_2} \\ -\frac{A_2}{L_2} & \frac{A_2}{L_2} \end{bmatrix}\begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \lambda\frac{\rho}{6}\begin{bmatrix} 2(A_1L_1 + A_2L_2) & A_2L_2 \\ A_2L_2 & 2A_2L_2 \end{bmatrix}\begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix}$$

We note here that the density is

$$\rho = \frac{f}{g} = \frac{0.283}{32.2 \times 12} = 7.324 \times 10^{-4}\ \text{lbs}^2/\text{in.}^4$$

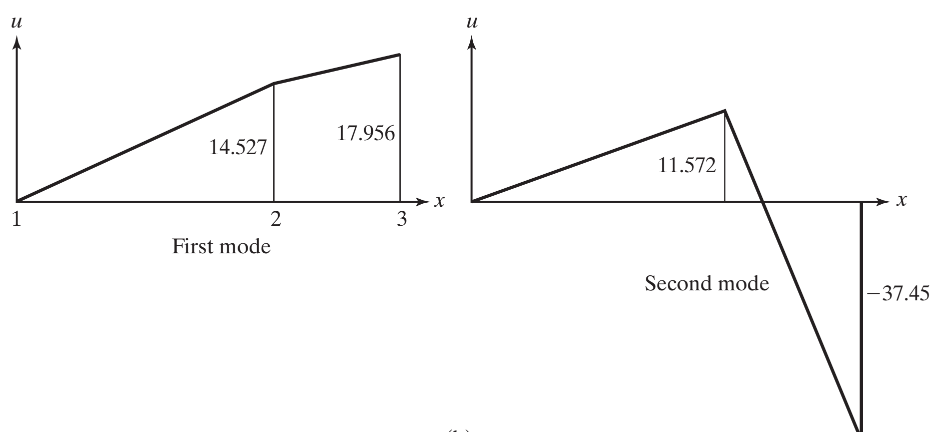
Substituting the values, we get

$$30 \times 10^6\begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.1 \end{bmatrix}\begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \lambda 1.22 \times 10^{-4}\begin{bmatrix} 25 & 2.5 \\ 2.5 & 5 \end{bmatrix}\begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix}$$



$E = 30 \times 10^6$ psi
Specific weight $f = 0.283$ lb/in.^{3}

(a)



(b)

FIGURE E11.2

The characteristic equation is

$$\det\begin{bmatrix} (6 \times 10^6 - 30.5 \times 10^{-4}\lambda) & (-3 \times 10^6 - 3.05 \times 10^{-4}\lambda) \\ (-3 \times 10^6 - 3.05 \times 10^{-4}\lambda) & (3 \times 10^6 - 6.1 \times 10^{-4}\lambda) \end{bmatrix} = 0$$

which simplifies to

$$1.77 \times 10^{-6}\lambda^2 - 1.465 \times 10^4\lambda + 9 \times 10^{12} = 0$$

The eigenvalues are

$$\lambda_1 = 6.684 \times 10^8$$

$$\lambda_2 = 7.61 \times 10^9$$

Note that $\lambda = \omega^2$, where ω is the circular frequency given by $2\pi f$ and f = frequency in hertz (cycles/s).

These frequencies are

$$f_1 = 4115\,\text{Hz}$$

$$f_2 = 13\,884\,\text{Hz}$$

The eigenvector for λ_1 is found from

$$(\mathbf{K} - \lambda_1\mathbf{M})\mathbf{U_1} = \mathbf{0}$$

which gives

$$10^6\begin{bmatrix} 3.96 & -3.204 \\ -3.204 & 2.592 \end{bmatrix}\begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix}_1 = \mathbf{0}$$

The two previous equations are not independent, since the determinant of the matrix is zero. This gives

$$3.96U_2 = 3.204U_3$$

Thus,

$$\mathbf{U}_1^{\mathrm{T}} = [U_2, 1.236U_2]$$

For normalization, we set

$$\mathbf{U}_1^{\mathrm{T}}\mathbf{M}\mathbf{U}_1 = 1$$

On substituting for $\mathbf{U}_1$, we get

$$\mathbf{U}_1^{\mathrm{T}} = [14.527 \quad 17.956]$$

The eigenvector corresponding to the second eigenvalue is similarly found to be

$$\mathbf{U}_2^{\mathrm{T}} = [11.572 \quad -37.45]$$

The mode shapes are shown in Fig. E11.2b. ■

Implementation of characteristic polynomial approach on computers is rather tedious and requires further mathematical considerations. We now discuss the other two categories.

Vector Iteration Methods. Various vector iteration methods use the properties of the *Rayleigh quotient*. For the generalized eigenvalue problem given by Eq. 11.38, we define the Rayleigh quotient

$$Q(\mathbf{v}) = \frac{\mathbf{v}^{\mathrm{T}}\mathbf{K}\mathbf{v}}{\mathbf{v}^{\mathrm{T}}\mathbf{M}\mathbf{v}} \tag{11.45}$$

where $\mathbf{v}$ is an arbitrary vector. A fundamental property of the Rayleigh quotient is that it lies between the smallest and the largest eigenvalue:

$$\lambda_1 \leq Q(\mathbf{v}) \leq \lambda_n \tag{11.46}$$

Power iteration, inverse iteration, and subspace iteration methods use this property. Power iteration leads to evaluation of the largest eigenvalue. Subspace iteration technique is suitable for large-scale problems and is used in several codes. The inverse iteration scheme can be used in evaluating the lowest eigenvalues. We present here the inverse iteration scheme and give a computer program for banded matrices.

Inverse Iteration Method

In the inverse iteration scheme, we start with a trial vector $\mathbf{u}^0$. The iterative solution proceeds as follows:

Step 0. Estimate an initial trial vector $\mathbf{u}^0$. Set iteration index $k = 0$.
Step 1. Set $k = k + 1$.
Step 2. Determine right side: $\mathbf{v}^{k-1} = \mathbf{M}\mathbf{u}^{k-1}$.
Step 3. Solve equations: $\mathbf{K}\hat{\mathbf{u}}^k = \mathbf{v}^{k-1}$.
Step 4. Denote $\hat{\mathbf{v}}^k = \mathbf{M}\hat{\mathbf{u}}^k$
Step 5. Estimate eigenvalue: $\lambda^k = \dfrac{\hat{\mathbf{u}}^{k^{\mathrm{T}}}\mathbf{v}^{k-1}}{(\hat{\mathbf{u}}^{k^{\mathrm{T}}}\hat{\mathbf{v}}^k)}$. (11.47)

Step 6. Normalize eigenvector: $u^k = \dfrac{\hat{\mathbf{u}}^k}{(\hat{\mathbf{u}}^{k^{\mathrm{T}}}\hat{\mathbf{v}}^k)^{1/2}}$.

Step 7. Check for tolerance: $\left|\dfrac{\lambda^k - \lambda^{k-1}}{\lambda^k}\right| \leq$ tolerance.

If satisfied, denote the eigenvector $\mathbf{u}^k$ as $\mathbf{U}$ and exit. Otherwise, go to step 1.

The algorithm just described converges to the lowest eigenvalue, provided the trial vector does not coincide with one of the eigenvectors. Other eigenvalues can be evaluated by shifting, or by taking the trial vector from a space that is M-orthogonal to the calculated eigenvectors.

Shifting We define a shifted stiffness matrix as

$$\mathbf{K}_s = \mathbf{K} + s\mathbf{M} \tag{11.48}$$

where $\mathbf{K}_s$ is the shifted matrix. Now the shifted eigenvalue problem is

$$\mathbf{K}_s\mathbf{U} = \lambda_s\mathbf{MU} \tag{11.49}$$

We state here without proof that the eigenvectors of the shifted problem are the same as those of the original problem. The eigenvalues get shifted by s:

$$\lambda_s = \lambda + s \tag{11.50}$$

Orthogonal Space Higher eigenvalues can be obtained by the inverse iteration method by choosing the trial vector from a space M-orthogonal to the calculated eigenvectors. This can be done effectively by the Gram–Schmidt process. Let $U_1, U_2, \ldots, U_m$ be the first m eigenvectors already determined. The trial vector for each iteration is then taken as

$$\mathbf{u}^{k-1} = \mathbf{u}^{k-1} - (\mathbf{u}^{k-1^{\mathrm{T}}}\mathbf{M}\mathbf{U}_1)\mathbf{U}_1 - (\mathbf{u}^{k-1^{\mathrm{T}}}\mathbf{M}\mathbf{U}_2)\mathbf{U}_2 - \cdots - (\mathbf{u}^{k-1^{\mathrm{T}}}\mathbf{M}\mathbf{U}_m)\mathbf{U}_m \tag{11.51}$$

This is the Gram–Schmidt process, which results in the evaluation of λ_{m+1} and $\mathbf{U}_{m+1}$. This is used in the program given in this chapter. Equation 11.51 is implemented after step 1 in the algorithm given in Eq. 11.51.

Example 11.3

Evaluate the lowest eigenvalue and the corresponding eigenmode for the beam shown in Fig. E11.3a.

Solution Using only the dof Q_3, Q_4, Q_5, and Q_6, we obtain the stiffness and mass matrices:

$$\mathbf{K} = 10^3\begin{bmatrix} 355.56 & 0 & -177.78 & 26.67 \\ & 10.67 & -26.67 & 2.667 \\ \text{Symmetric} & & 177.78 & -26.67 \\ & & & 5.33 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 0.4193 & 0 & 0.0726 & -.0052 \\ & .000967 & .0052 & -.00036 \\ \text{Symmetric} & & 0.2097 & -.0089 \\ & & & .00048 \end{bmatrix}$$

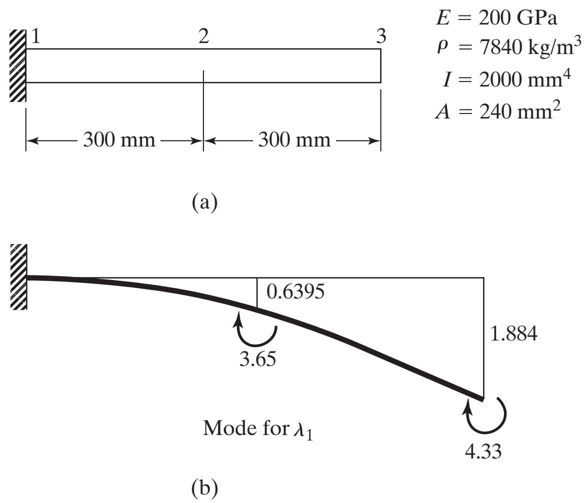


FIGURE E11.3

The inverse iteration program requires the creation of a data file. The format of the file for the preceding problem is as follows:

```
Data File:
    TITLE
      NDOF      NBW
       4         4
  Banded Stiffness Matrix
     3.556E5             0            -1.778E5     2.667E4
     1.067E4            -2.667E4       2.667E3     0
     1.778E5            -2.667E4       0           0
     5.333E3             0             0           0
  Banded Mass Matrix
     0.4193              0             0.0726     -0.0052
     0.000967            0.0052       -0.00036     0
     0.2097             -0.0089        0           0
     0.00048             0             0           0
```

The first line of data contains the values of n = dimension of the matrices and nbw = half-band width. This is followed by **K** and **M** matrices in banded form. (See Chapter 2.) The two titles are part of the data file. Though these data were created by hand calculations, it is possible to write a program, as discussed at the end of this chapter.

Feeding this data file into the inverse iteration program, INVITR, gives the lowest eigenvalue,

$$\lambda_1 = 2.03 \times 10^4$$

and the corresponding eigenvector or mode shape,

$$\mathbf{U}_1^{\mathrm{T}} = [0.64 \quad 3.65 \quad 1.88 \quad 4.32]$$

λ_1 corresponds to a circular frequency of 142.48 rad/s or 22.7 Hz (=142.48/2π). The mode shape is shown in Fig. E11.3b. ■

Transformation Methods. The basic approach here is to transform the matrices to a simpler form and then determine the eigenvalues and eigenvectors. The major methods in this category are the generalized Jacobi method and the QR method. These methods are suitable for large-scale problems. In the QR method, the matrices are first reduced to tridiagonal form using Householder matrices. The generalized Jacobi method uses the transformation to simultaneously diagonalize the stiffness and mass matrices. This method needs the full matrix locations and is quite efficient for calculating all eigenvalues and eigenvectors for small problems. We present here the generalized Jacobi method as an illustration of the transformation approach.

If all the eigenvectors are arranged as columns of a square matrix $\mathbb{U}$ and all eigenvalues as the diagonal elements of a square matrix $\mathbf{\Lambda}$, then the generalized eigenvalue problem can be written in the form

$$\mathbf{K}\mathbb{U} = \mathbf{M}\mathbb{U}\mathbf{\Lambda} \tag{11.52}$$

where

$$\mathbb{U} = [\mathbf{U}_1,\ \mathbf{U}_2, \ldots,\ \mathbf{U}_n] \tag{11.53}$$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & 0 \\ 0 & & \ddots & \\ & & & \lambda_n \end{bmatrix} \tag{11.54}$$

Using the M-orthonormality of the eigenvectors, we have

$$\mathbb{U}^{\mathrm{T}}\mathbf{M}\mathbb{U} = \Lambda \tag{11.55a}$$

and

$$\mathbb{U}^{\mathrm{T}}\mathbf{K}\mathbb{U} = \mathbf{I} \tag{11.55b}$$

where $\mathbf{I}$ is the identity matrix.

Generalized Jacobi Method

In the generalized Jacobi method, a series of transformations $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_\ell$ are used such that if $\mathbf{P}$ represents the product

$$\mathbf{P} = \mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_\ell \tag{11.56}$$

then the off-diagonal terms of $\mathbf{P}^{\mathrm{T}}\mathbf{KP}$ and $\mathbf{P}^{\mathrm{T}}\mathbf{MP}$ are zero. In practice, the off-diagonal terms are set to be less than a small tolerance. If we denote these diagonal matrices as

$$\hat{\mathbf{K}} = \mathbf{P}^{\mathrm{T}}\mathbf{KP} \tag{11.57a}$$

and

$$\hat{\mathbf{M}} = \mathbf{P}^{\mathrm{T}}\mathbf{MP} \tag{11.57b}$$

then the eigenvectors are given by

$$\mathbb{U} = \mathbf{P}\hat{\mathbf{M}}^{-1/2} \tag{11.58}$$

and

$$\mathbf{\Lambda} = \hat{\mathbf{M}}^{-1}\hat{\mathbf{K}} \tag{11.59}$$

where

$$\hat{\mathbf{M}}^{-1} = \begin{bmatrix} \hat{M}_{11}^{-1} & & & 0 \\ & \hat{M}_{22}^{-1} & & \\ & & \ddots & \\ 0 & & & \hat{M}_{nn}^{-1} \end{bmatrix} \tag{11.60}$$

$$\hat{\mathbf{M}}^{-1/2} = \begin{bmatrix} \hat{M}_{11}^{-1/2} & & & 0 \\ & \hat{M}_{22}^{-1/2} & & \\ & & \ddots & \\ 0 & & & \hat{M}_{nn}^{-1/2} \end{bmatrix} \tag{11.61}$$

Computationally, Eq. 11.58 indicates that each row of $\mathbf{P}$ is divided by the square root of the diagonal element of $\hat{\mathbf{M}}$, and Eq. 11.59 indicates that each diagonal element of $\hat{\mathbf{K}}$ is divided by the diagonal element of $\hat{\mathbf{M}}$.

We mentioned that the diagonalization follows in several steps. At step k, we choose a transformation matrix $\mathbf{P}_k$ given by

$$\mathbf{P}_k = \begin{array}{cc} & \begin{array}{cccccccccc} \text{Column} \rightarrow & & & i & & & & j & & \end{array} \\ \begin{array}{c} \text{Row} \downarrow \\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \end{array} & \begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & \alpha & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & \beta & & & & 1 & & \\ & & & & & & & & 1 & \\ & & & & & & & & & 1 \end{bmatrix} \end{array} \tag{11.62}$$

P_k has all diagonal elements equal to 1, has a value of α at row i and column j and β at row j and column i, and has all other elements equal to zero. The scalars α and β are chosen so that the ij locations of $\mathbf{P}_k^{\mathrm{T}}\mathbf{K}\mathbf{P}_k$ and $\mathbf{P}_k^{\mathrm{T}}\mathbf{M}\mathbf{P}_k$ are simultaneously zero. This is represented by

$$\alpha K_{ii} + (1 + \alpha\beta)K_{ij} + \beta K_{jj} = 0 \tag{11.63}$$

$$\alpha M_{ii} + (1 + \alpha\beta)M_{ij} + \beta M_{jj} = 0 \tag{11.64}$$

where $K_{ii}, K_{ij}, \ldots, M_{ii}, \ldots$ are elements of the stiffness and mass matrices. The solution of these equations is as follows:

Denote

$$\begin{aligned} A &= K_{ij}M_{ij} - M_{ii}K_{ij} \\ B &= K_{jj}M_{ij} - M_{jj}K_{ij} \\ C &= K_{ii}M_{jj} - M_{ii}K_{jj} \end{aligned}$$

Then α and β are given by

$$A \neq 0, B \neq 0: \quad \alpha = \frac{-0.5C + \operatorname{sgn}(C)\sqrt{0.25C^2 + AB}}{A}$$

$$\beta = -\frac{A\alpha}{B}$$

$$A = 0: \quad \beta = 0$$

$$\alpha = -\frac{K_{ij}}{K_{jj}}$$

$$B = 0: \quad \alpha = 0$$

$$\beta = -\frac{K_{ij}}{K_{jj}}$$

When both A and B are zero, any one of the two values listed can be chosen. (*Note:* There is no summation on repeated indices in these expressions.)

In the generalized Jacobi program given at the end of the chapter, the elements of **K** and **M** are zeroed out in the order indicated in Fig. 11.6. Once $\mathbf{P}_k$ is defined by determining α and β, $\mathbf{P}_k^{\mathrm{T}}[\,]\mathbf{P}_k$ can be performed on **K** and **M** as shown in Fig. 11.7. Also by starting with $\mathbf{P} = \mathbf{I}$, the product $\mathbf{P}\mathbf{P}_k$ is computed after each step. When all elements are covered as shown in Fig. 11.6, one pass is completed. After the operations at step k, some of the previously zeroed elements are altered. Another pass is conducted to check for the value of diagonal elements. The transformation is performed if the element at ij is larger than a tolerance value. A tolerance of $10^{-6} \times$ smallest K_{ii} is used for stiffness, and $10^{-6} \times$ largest M_{ii} is used for the mass. The tolerance can be redefined for higher accuracy. The process stops when all off-diagonal elements are less than the tolerance.

If the diagonal masses are less than the tolerance, the diagonal value is replaced by the tolerance value; thus, a large eigenvalue will be obtained. In this method, **K** need not be positive definite.

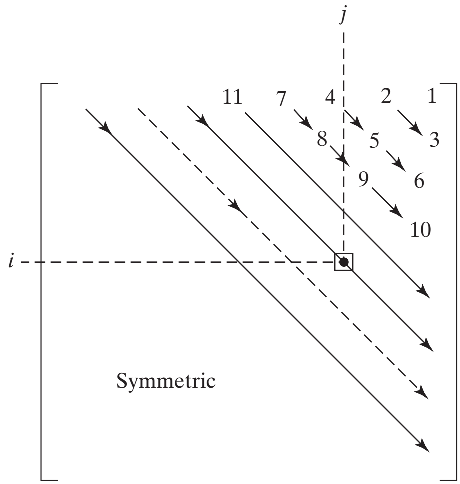


FIGURE 11.6 Diagonalization.

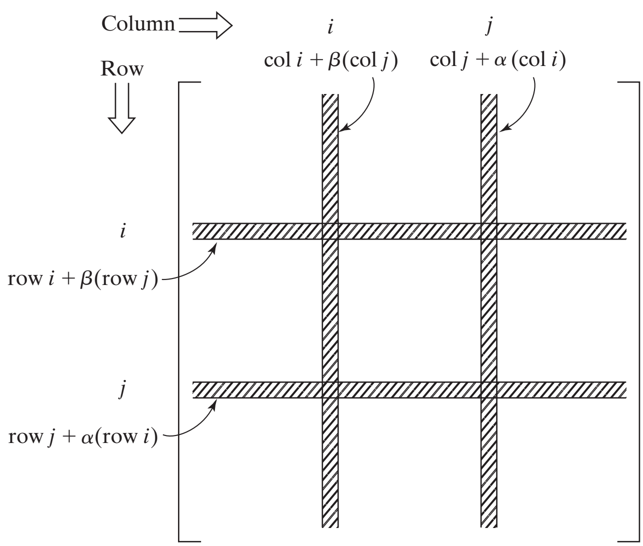


FIGURE 11.7 Multiplication $\mathbf{P}_k^{\mathrm{T}}[\,]\mathbf{P}$.

Example 11.4

Determine all the eigenvalues and eigenvectors for the beam discussed in Example 11.3 using program JACOBI.

Solution The input data for JACOBI is same as that for INVITR. However, the program converts to full matrices in calculations. Convergence occurs at the fourth sweep. The solution is as follows:

$$\lambda_1 = 2.0304 \times 10^4\,(22.7\,\text{Hz})$$
$$\mathbf{U}_1^{\mathrm{T}} = [0.64 \quad 3.65 \quad 1.88 \quad 4.32]$$

$$\lambda_2 = 8.0987 \times 10^5 (143.2 \text{ Hz})$$
$$\mathbf{U}_2^{\mathrm{T}} = [-1.37 \quad 1.39 \quad 1.901 \quad 15.27]$$
$$\lambda_3 = 9.2651 \times 10^6 (484.4 \text{ Hz})$$
$$\mathbf{U}_3^{\mathrm{T}} = [-0.20 \quad 27.16 \quad -2.12 \quad -33.84]$$
$$\lambda_4 = 7.7974 \times 10^7 (1405.4 \text{ Hz})$$
$$\mathbf{U}_4^{\mathrm{T}} = [0.8986 \quad 30.89 \quad 3.546 \quad 119.15]$$

Note that the eigenvalues are arranged in ascending order after they are evaluated. ■

Tridiagonalization and Implicit Shift Approach

We present here a powerful method for evaluating eigenvalues and eigenvectors using the implicit shift approach. We first bring the problem $\mathbf{Kx} = \lambda\mathbf{Mx}$ to the standard form $\mathbf{Ax} = \lambda\mathbf{x}$. *Householder reflection* steps are then applied to bring the matrix to a tridiagonal form. Implicit symmetric QR step is applied with Wilkinson shift[1] to diagonalize the matrix. These steps are now provided in detail.

Bringing Generalized Problem to Standard Form

We observe that the mass matrix $\mathbf{M}$ is positive semidefinite. Positive semidefinite matrices have the property that if the diagonal element is zero, all the nondiagonal elements in the row and column are zero. In such a case, a small mass equal to the tolerance value, say $10^{-6} \times$ largest M_{ii}, is added at the diagonal location. This makes the mass matrix positive definite. Correspondingly, this yields a higher eigenvalue. The first step in bringing the problem to the standard form is to perform the *Cholesky decomposition* of $\mathbf{M}$ using the calculations presented in Chapter 2:

$$\mathbf{M} = \mathbf{LL}^{\mathrm{T}} \tag{11.65}$$

Symmetric manipulation of the generalized eigenvalue problem yields the form

$$\mathbf{L}^{-1}\mathbf{K}(\mathbf{L}^{-1})^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{x} = \lambda\mathbf{L}^{\mathrm{T}}\mathbf{x} \tag{11.66}$$

Denoting $\mathbf{A} = \mathbf{L}^{-1}\mathbf{K}(\mathbf{L}^{-1})^{\mathrm{T}}$ and $\mathbf{y} = \mathbf{L}^{\mathrm{T}}\mathbf{x}$, we get the standard form

$$\mathbf{Ay} = \lambda\mathbf{y} \tag{11.67}$$

This problem has the same eigenvalues as the generalized problem. The eigenvectors $\mathbf{x}$ are evaluated by the forward substitution performed on

$$\mathbf{L}^{\mathrm{T}}\mathbf{x} = \mathbf{y} \tag{11.68}$$

In the computer implementation, $\mathbf{A}$ is obtained in two steps: $\mathbf{LB} = \mathbf{K}$ and $\mathbf{LA} = \mathbf{B}^{\mathrm{T}}$. These two forward-substitution steps are more efficient than finding the inverse and performing the multiplication steps.

[1]Golub, H.G., and C. F. Van Loan, *Matrix Computations*, 3rd Ed. Baltimore: The Johns Hopkins University Press, 1996.

Example 11.5

Bring the generalized problem $\mathbf{Kx} = \lambda\mathbf{Mx}$ to the standard form $\mathbf{Ay} = \lambda\mathbf{y}$, given that

$$\mathbf{K} = \begin{bmatrix} 4 & 1 & 2 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 1 & 4 & 2 \\ 1 & 2 & 2 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 5 \end{bmatrix}$$

Solution We make use of the Cholesky decomposition algorithm given in Chapter 2 to decompose $\mathbf{M} = \mathbf{LL}^{\mathrm{T}}$. $\mathbf{L}$ is obtained as

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{bmatrix}$$

Solving for $\mathbf{LK} = \mathbf{B}$, which is a forward-substitution operation, we get

$$\mathbf{B} = \begin{bmatrix} 4 & 1 & 2 & 1 \\ -3 & 2 & -1 & 1 \\ -2 & 0 & 2 & 1 \\ -7 & 0 & -2 & 2 \end{bmatrix}$$

Another forward-substitution step of solving $\mathbf{LA} = \mathbf{B}^{\mathrm{T}}$ gives

$$\mathbf{A} = \begin{bmatrix} 4 & -3 & -2 & -7 \\ -3 & 5 & 2 & 7 \\ -2 & 2 & 4 & 5 \\ -7 & 7 & 5 & 16 \end{bmatrix}$$

The standard form is now $\mathbf{Ay} = \lambda\mathbf{y}$, with $\mathbf{L}^{\mathrm{T}}\mathbf{x} = \mathbf{y}$. ■

Tridiagonalization

There are several different methods available to transform a symmetric matrix to tridiagonal form. We use the Householder reflection ideas in the tridiagonalization process. Given a unit vector $\mathbf{w}$ normal to a hyperplane, the reflected vector $\mathbf{b}$ of vector $\mathbf{a}$, shown in Fig. 11.8, is given by

$$\mathbf{b} = \mathbf{a} - 2(\mathbf{w}^{\mathrm{T}}\mathbf{a})\mathbf{w} \tag{11.69}$$

This can be put in the form

$$\mathbf{b} = \mathbf{Ha} \tag{11.70}$$

where

$$\mathbf{H} = \mathbf{I} - 2\mathbf{ww}^{\mathrm{T}} \quad \text{with} \quad \mathbf{w}^{\mathrm{T}}\mathbf{w} = 1 \tag{11.71}$$

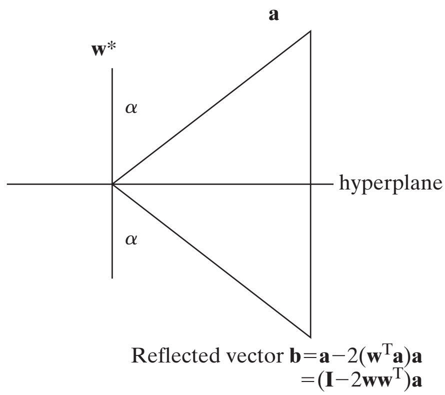


FIGURE 11.8 Householder reflection.

is the Householder transformation. This transformation, which reflects a vector about a plane for which **w** is the normal direction, has some interesting properties:

1. The Householder transformation is symmetric (i.e., $\mathbf{H}^T = \mathbf{H}$).
2. Its inverse is itself (i.e., $\mathbf{HH} = \mathbf{I}$).

It is thus an orthogonal transformation.

If a vector **a** has to be along the unit vector $\mathbf{e}_1$ after reflection, it is easy to see from Fig. 11.8 that **w** is the unit vector in the direction given by $\mathbf{a} \pm |\mathbf{a}|\mathbf{e}_1$. The reflected vector is along $-\mathbf{e}_1$, if **w** is along $\mathbf{a} + |\mathbf{a}|\mathbf{e}_1$. We choose vector $\mathbf{a} + |\mathbf{a}|\mathbf{e}_1$ or $\mathbf{a} - |\mathbf{a}|\mathbf{e}_1$, whichever has a larger magnitude. This reduces numerical errors in the calculation. We note that this is accomplished by taking **w** along $\mathbf{a} + \text{sign}(a_1)|\mathbf{a}|\mathbf{e}_1$, where a_1 is the component of **a** along the unit vector $\mathbf{e}_1$. The steps involved in the tridiagonalization process are illustrated by extending the Example 11.5. The symmetric matrix in $\mathbf{Ay} = \lambda\mathbf{y}$ is

$$\mathbf{A} = \begin{bmatrix} 4 & -3 & -2 & -7 \\ -3 & 5 & 2 & 7 \\ -2 & 2 & 4 & 5 \\ -7 & 7 & 5 & 16 \end{bmatrix}$$

To start the tridiagonalization, we make use of the vector $[-3\ -2\ -7]^T$, which is made up of the elements below the diagonal in column 1. Consider this as vector **a**, which we would like to bring to $\mathbf{e}_1 = [100]^T$. We have $|\mathbf{a}| = \sqrt{3^2 + 2^2 + 7^2} = 7.874$. $\mathbf{w}_1$ is then the unit vector along $\mathbf{a} - |\mathbf{a}|\mathbf{e}_1 = [-10.874 \quad -2 \quad -7]^T$. The length of this vector is $\sqrt{10.874^2 + 2^2 + 7^2} = 13.086$. The unit vector is $\mathbf{w}_1 = [-0.831 \quad -0.1528 \quad -0.5349]^T$. Denoting $\mathbf{H}_1 = [\mathbf{I} - 2\mathbf{w}_1\mathbf{w}_1]^T$, we have

$$\mathbf{H}_1 \begin{bmatrix} -3 \\ -2 \\ -7 \end{bmatrix} = \begin{bmatrix} 7.874 \\ 0 \\ 0 \end{bmatrix}$$

in the first column and $[-3 \quad -2 \quad -7]\mathbf{H}_1 = [7.874 \quad 0 \quad 0]$ in the first row. Thus the first row of the tridiagonal matrix $\mathbf{T}$ is $[4 \quad 7.874 \quad 0 \quad 0]^{\mathrm{T}}$ and this matrix is symmetric.

Multiplication from both sides on the 3×3 partition is performed as follows:

$$\mathbf{H}_1 \begin{bmatrix} 5 & 2 & 7 \\ 2 & 4 & 5 \\ 7 & 5 & 16 \end{bmatrix} \mathbf{H}_1 = \begin{bmatrix} 21.0161 & -0.7692 & 0.9272 \\ -0.7692 & 2.4395 & 0.2041 \\ 0.9272 & 0.2041 & 1.5443 \end{bmatrix}$$

$\mathbf{H}_1$ is not formed at this stage. If the partitioned matrix is designated as $\mathbf{B}$, the multiplication is easily implemented by using the formula

$$\mathbf{H}_1\mathbf{B}\mathbf{H}_1 = [\mathbf{I} - 2\mathbf{w}_1\mathbf{w}_1^{\mathrm{T}}]\mathbf{B}[\mathbf{I} - 2\mathbf{w}_1\mathbf{w}_1^{\mathrm{T}}] = \mathbf{B} - 2\mathbf{w}_1\mathbf{b}^{\mathrm{T}} - 2\mathbf{b}\mathbf{w}_1^{\mathrm{T}} + 4\beta\mathbf{w}_1\mathbf{w}_1^{\mathrm{T}} \tag{11.72}$$

where $\mathbf{b} = \mathbf{B}\mathbf{w}_1, \beta = \mathbf{w}_1^{\mathrm{T}}\mathbf{b}$.

At the next step, the vector $[-0.7692 \quad 0.9272]^{\mathrm{T}}$ is reflected to line up along $[1 \quad 0]^{\mathrm{T}}$. The magnitude of the vector is 1.2047. $\mathbf{w}_2$ is the vector along $[(-0.7692 - 1.2047)0.9272]^{\mathrm{T}}$. The unit vector $\mathbf{w}_2$ for this is $[-0.9051 \quad 0.4252]^{\mathrm{T}}$. On performing the multiplication with the partitioned 2×2 matrix and placing in the 4×4 matrix $\mathbf{T}$, we get the tridiagonal matrix

$$\mathbf{T} = \begin{bmatrix} 4 & 7.874 & 0 & 0 \\ 7.874 & 21.0161 & 1.2047 & \\ & 1.2047 & 1.7087 & -0.4022 \\ & & -0.4022 & 2.271 \end{bmatrix} = \begin{bmatrix} d_1 & b_1 & 0 & 0 \\ b_1 & d_2 & b_2 & 0 \\ 0 & b_2 & d_3 & b_3 \\ 0 & 0 & b_3 & d_4 \end{bmatrix} \tag{11.73}$$

In this development, the tridiagonal matrix is stored in two vectors $\mathbf{d}$ and $\mathbf{b}$. The original matrix $\mathbf{A}$ is used for storing the Householder vectors $\mathbf{w}_1$, $\mathbf{w}_2$, etc. as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.831 & 1 & 0 & 0 \\ -0.1528 & -0.9051 & 1 & 0 \\ -0.5349 & 0.4252 & 0 & 1 \end{bmatrix}$$

The product of $\mathbf{H}_1\mathbf{H}_2$ is easily performed in place inside $\mathbf{A}$. First, it is the product of $\mathbf{H}_2$ and the lower right 2×2 identity matrix, to obtain

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.831 & 1 & 0 & 0 \\ -0.1528 & 0 & -0.6385 & 0.7696 \\ -0.5349 & 0 & 0.7696 & 0.6385 \end{bmatrix}$$

Then the multiplication of $\mathbf{H}_2$ and the lower right 3×3 matrix gives

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.381 & -0.522 & -0.7631 \\ 0 & -0.254 & -0.7345 & 0.6292 \\ 0 & -0.889 & 0.4336 & 0.1473 \end{bmatrix}$$

This matrix represents the current contribution to the eigenvectors. We now discuss the steps to diagonalize the matrix and finding the eigenvectors.

Implicit Symmetric QR Step with Wilkinson Shift for Diagonalization[2]

The inverse iteration may be applied to the tridiagonal matrix to get desired eigenvalues. If all eigenvalues and eigenvectors are desired, the implicit shift ideas of Wilkinson provide a remarkably fast algorithm. The order of convergence with this method is cubic. The shift value μ for the *Wilkinson shift* is taken as the eigenvalue of the bottom 2×2 matrix of the tridiagonal matrix close to d_n, viz.,

$$\mu = d_n + t - \operatorname{sign}(t)\sqrt{b_{n-1}^2 + t^2} \tag{11.74}$$

where $t = 0.5(d_{n-1}-d_n)$. The implicit shift is carried out by performing a Givens rotation $\mathbf{G}_1$. $c\,(=\cos\theta)$ and $s\,(=\sin\theta)$ are chosen such that we get zero in the second position as follows:

$$\mathbf{G}_1\begin{bmatrix} d_1 - \mu \\ b_1 \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}\begin{bmatrix} d_1 - \mu \\ b_1 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix} \tag{11.75}$$

We note that if $r = \sqrt{b_1^2 + (d_1 - \mu)^2}$, $c = -(d_1 - \mu)/r$ and $s = b_1/r$. We perform the rotation $\mathbf{G}_1\mathbf{T}\mathbf{G}_1^{\mathrm{T}}$ on the tridiagonal matrix on the first two rows from the left and first two columns from the right. Note that the rotation is calculated based on shift μ, but the shift itself is not performed. This is implicit shift. We refer to the tridiagonal matrix in Eq. 11.73. With $d_3 = 1.7087$, $d_4 = 2.2751$, and $b_3 = -0.4022$, we have $t = -0.2832$ and $\mu = 2.4838$. With $d_1 - \mu = 1.5162$, $b_1 = 7.874$, and $r = 8.0186$, we get $c = -0.1891$.

and $s = 0.982$. We then get from Eq. 11.75, [$\mathbf{G}_1 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} -0.1891 & -0.982 \\ 0.982 & -0.1891 \end{bmatrix}$.

After this rotation, two additional elements, each with value $a = -1.18297$, are introduced at (3,1) and (1,3) and the matrix is no longer tridiagonal.

$$\mathbf{G}_1\mathbf{T}\mathbf{G}_1^{\mathrm{T}} = \begin{bmatrix} 23.3317 & -4.1515 & -1.18297 & 0 \\ -4.1515 & 1.6844 & -0.2278 & 0 \\ -1.18297 & -0.2278 & 1.7087 & -0.4022 \\ 0 & 0 & -0.4022 & 2.2751 \end{bmatrix}$$

Givens rotation $\mathbf{G}_2$ is then applied to rows 2, 3 and columns 2, 3 with reference to -4.1515 and -1.18297 such that the elements at (3,1) and (1,3) become zero. We get $\mathbf{G}_2 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} 0.9617 & 0.274 \\ -0.274 & 0.9617 \end{bmatrix}$. This leads to

[2]Wilkinson, J. H., "Global Convergence of Tridiagonal QR Algorithm with Origin Shifts," *Linear Algebra and Its Applications*, I: 409–420 (1968).

$$\mathbf{G}_2\mathbf{T}\mathbf{G}_2^{\mathrm{T}} = \begin{bmatrix} 23.3317 & -4.3168 & 0 & 0 \\ -4.3168 & 1.5662 & -0.1872 & -0.1102 \\ 0 & -0.1872 & 1.8269 & -0.3868 \\ 0 & -0.1102 & -0.3868 & 2.2751 \end{bmatrix}$$

Givens rotation $\mathbf{G}_3$ is then applied with respect to elements at (3,2) and (4,2) to make the locations (4,2) and (2,4) zero. We get $\mathbf{G}_3 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} 0.8617 & 0.5074 \\ -0.5074 & 0.8617 \end{bmatrix}$.

After this application, the resulting matrix is tridiagonal. The off-diagonal elements at the bottom become smaller:

$$\mathbf{G}_3\mathbf{T}\mathbf{G}_3^{\mathrm{T}} = \begin{bmatrix} 23.3317 & -4.3168 & 0 & 0 \\ -4.3168 & 1.5662 & -0.2172 & 0 \\ 0 & -0.2172 & 1.6041 & 0.00834 \\ 0 & 0 & 0.00834 & 2.498 \end{bmatrix}$$

The eigenvector matrix is updated by multiplying $\mathbf{A}$ with the Givens rotations $\mathbf{A}\mathbf{G}_1^{\mathrm{T}}\mathbf{G}_2^{\mathrm{T}}\mathbf{G}_3^{\mathrm{T}}$.

The iteration process is repeated until the off diagonal element $b_3(b_{n-1}$ for $n \times n$ matrix) becomes small in magnitude (say, less than 10^{-8}). d_4 is an eigenvalue. The process is now repeated on the 3×3 tridiagonal matrix obtained by excluding row 4 and column 4. This is continued till each of the off-diagonal terms approaches zero. All eigen values are thus evaluated. Wilkinson has shown that this procedure cubically converges to diagonal form. The eigenvalues are 24.1567, 0.6914, 1.6538, and 2.4981, and the corresponding eigenvectors, after multiplying by $\mathbf{L}^{-1}$ (a forward-substitution operation given by Eq. 11.68), are the columns of the matrix

$$\begin{bmatrix} -2.6278 & -0.1459 & 0.0934 & 0.2543 \\ 0.3798 & 0.1924 & -0.8112 & 0.4008 \\ 0.2736 & 0.1723 & -0.2780 & -0.9045 \\ 0.8055 & -0.5173 & 0.2831 & 0.0581 \end{bmatrix}$$

This algorithm for finding the eigenvalues and eigenvectors of the generalized eigenvalue problem is implemented in the program GENEIGEN.

11.5 INTERFACING WITH PREVIOUS FINITE ELEMENT PROGRAMS AND A PROGRAM FOR DETERMINING CRITICAL SPEEDS OF SHAFTS

Once the stiffness matrix $\mathbf{K}$ and mass matrix $\mathbf{M}$ for a structure are known, then the inverse iteration or Jacobi programs that are provided can be used to determine the natural frequencies and mode shapes. The finite element programs for rod, truss, beam, and elasticity problems that we used in previous chapters can be readily modified to output the banded $\mathbf{K}$ and $\mathbf{M}$ matrices onto a file. This file is then input into the inverse iteration program, which gives the natural frequencies and mode shapes.

We have provided the BEAMKM program, which outputs the banded **K** and **M** matrices for a beam. This output file is then provided to program INVITR, which calculates the eigenvalues and eigenvectors (mode shapes). Example 11.6 illustrates the use of these two programs. Program CSTKM, which outputs **K** and **M** matrices for the CST element, has also been provided.

Example 11.6

Determine the lowest critical speed (or transverse natural frequency) of the shaft shown in Fig. E11.6. The shaft has two lumped weights, W_1 and W_2, representing flywheels, as shown. Take $E = 30 \times 10^6$ psi and mass density of shaft $\rho = 0.0007324$ lb·s²/in.⁴ (=0.283 lb/in.³).

Solution The lumped weights W_1 and W_2 correspond to lumped masses W_1/g and W_2/g, respectively, where $g = 386$ in./s². Program BEAMKM is executed, followed by program INVITR. The input data and solution are given at the end of the next section.

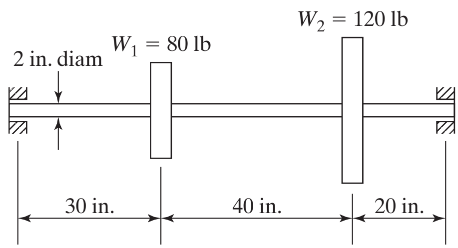


FIGURE E11.6

Now, we can obtain the critical speed in rpm from the eigenvalue 4042 as

$$\begin{aligned} n &= \sqrt{\lambda} \times \frac{60}{2\pi} \text{ rpm} \\ &= \sqrt{4042} \times \frac{60}{2\pi} \\ &= 607 \text{ rpm} \end{aligned}$$

This example illustrates how the inverse iteration and Jacobi programs given in this chapter can be interfaced with other programs for vibration analysis. ■

11.6 GUYAN REDUCTION

Often, large finite element models with thousands of dof are used for stress and deformation analysis of ships, aircraft, automobiles, nuclear reactors, and the like. It is clearly impractical and unnecessary to perform dynamic analyses using the detailed representation that is required for static analysis. Furthermore, design and control methods work best for systems with a small number of dof. To overcome this difficulty, dynamic reduction techniques have been developed to reduce the number of dof prior to performing dynamic analysis. Guyan reduction is one of the popular methods for dynamic reduction.[3] We have

[3]Guyan, R. J., "Reduction in stiffness and mass matrices," *AIAA Journal*, vol. 3, no. 2, p. 380, Feb. 1965.

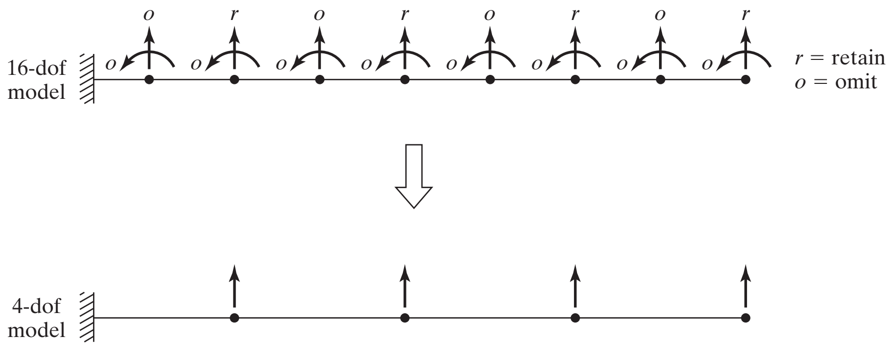


FIGURE 11.9 Guyan reduction.

to make the decision as to which dof are to be retained and which are to be omitted. For example, Fig. 11.9 shows how a reduced model is obtained by omitting certain dof. The omitted dof correspond to those at which the applied and inertial forces are negligible.

The reduced stiffness and mass matrices are obtained as follows: The equations of motion (see Eq. 11.13) are $\mathbf{M}\ddot{\mathbf{Q}} + \mathbf{KQ} = \mathbf{F}$. If we group the inertial force together with the applied force, we can write the equations as $\mathbf{KQ} = \mathbf{F}$. We will partition $\mathbf{Q}$ as

$$\mathbf{Q} = \begin{Bmatrix} \mathbf{Q}_r \\ \mathbf{Q}_o \end{Bmatrix} \tag{11.76}$$

where $\mathbf{Q}_r$ = retained set and $\mathbf{Q}_o$ = omitted set. Typically, the retained set is about 20% of the total dof. The equations of motion can now be written in partitioned form as

$$\begin{bmatrix} \mathbf{K}_{rr} & \mathbf{K}_{ro} \\ \mathbf{K}_{ro}^{\mathrm{T}} & \mathbf{K}_{oo} \end{bmatrix} \begin{Bmatrix} \mathbf{Q}_r \\ \mathbf{Q}_o \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_r \\ \mathbf{F}_o \end{Bmatrix} \tag{11.77}$$

The idea is to choose the omitted set such that the components of $\mathbf{F}_o$ are small. Thus, we should retain those dof in the r-set with large concentrated masses, which are loaded (in transient-response analysis) and which are needed to adequately describe the mode shape. Setting $\mathbf{F}_o = 0$, the lower part of Eq. 11.76 yields

$$\mathbf{Q}_o = -\mathbf{K}_{oo}^{-1}\mathbf{K}_{ro}^{T}\mathbf{Q}_r \tag{11.78}$$

The strain energy in the structure is $U = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{KQ}$. This can be written as

$$U = \frac{1}{2}\begin{bmatrix} \mathbf{Q}_r^{\mathrm{T}} & \mathbf{Q}_o^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{rr} & \mathbf{K}_{ro} \\ \mathbf{K}_{ro}^{\mathrm{T}} & \mathbf{K}_{oo} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_r \\ \mathbf{Q}_o \end{bmatrix}$$

Upon substituting Eq. 11.77 into the equation just described, we can write $U = \frac{1}{2}\mathbf{Q}_r^{\mathrm{T}}\mathbf{K}_r\mathbf{Q}_r$,

where

$$\mathbf{K}_r = \mathbf{K}_{rr} - \mathbf{K}_{ro}\mathbf{K}_{oo}^{-1}\mathbf{K}_{ro}^{\mathrm{T}} \tag{11.79}$$

is the reduced stiffness matrix. To obtain an expression for the reduced mass matrix, we consider the kinetic energy. Upon partitioning the mass matrix and using Eq. 11.78, we can write the kinetic energy as $V = \frac{1}{2}\dot{\mathbf{Q}}_r^{\mathrm{T}}\mathbf{M}_r\dot{\mathbf{Q}}_r$, where

$$\mathbf{M}_r = \mathbf{M}_{rr} - \mathbf{M}_{ro}\mathbf{K}_{oo}^{-1}\mathbf{K}_{ro}^{\mathbf{T}} - \mathbf{K}_{ro}\mathbf{K}_{oo}^{-1}\mathbf{M}_{ro}^{\mathrm{T}} + \mathbf{K}_{ro}\mathbf{K}_{oo}^{-1}\mathbf{M}_{oo}\mathbf{K}_{oo}^{-1}\mathbf{K}_{ro}^{\mathrm{T}} \tag{11.80}$$

is the reduced mass matrix. With the reduced stiffness and mass matrices, only a smaller eigenvalue problem needs to be solved:

$$\mathbf{K}_r\mathbf{U}_r = \lambda\mathbf{M}_r\mathbf{U}_r \tag{11.81}$$

Then we recover

$$\mathbf{U}_o = -\mathbf{K}_{oo}^{-1}\mathbf{K}_{ro}^{\mathbf{T}}\mathbf{U}_r \tag{11.82}$$

Example 11.7

In Example 11.3, the eigenvalues and mode shapes of a cantilever beam were determined based on a four-dof model. We will apply Guyan reduction to this problem based on omitting the rotational dofs and see how our results compare with the full model. Referring to Fig. E11.3, Q_3 and Q_5 refer to the translational dofs while Q_4 and Q_6 refer to the rotational dofs. Thus, the retained set is Q_3 and Q_5 and the omitted set is Q_4 and Q_6. Extracting the appropriate components from the full 4×4 **K** and **M** matrices, we obtain

$$\mathbf{k}_{rr} = 1000\begin{bmatrix} 355.6 & -177.78 \\ -177.78 & 177.78 \end{bmatrix} \qquad \mathbf{k}_{ro} = 1000\begin{bmatrix} 0 & 26.67 \\ -26.67 & -26.67 \end{bmatrix}$$

$$\mathbf{k}_{oo} = 1000\begin{bmatrix} 10.67 & 2.667 \\ 2.667 & 5.33 \end{bmatrix} \qquad \mathbf{m}_{rr} = \begin{bmatrix} 0.4193 & 0.0726 \\ 0.0726 & 0.2097 \end{bmatrix}$$

$$\mathbf{m}_{ro} = \begin{bmatrix} 0 & -0.0052 \\ 0.0052 & -0.0089 \end{bmatrix} \qquad \mathbf{m}_{oo} = \begin{bmatrix} 0.000967 & -0.00036 \\ -0.00036 & 0.00048 \end{bmatrix}$$

From Eqs. 11.68 and 11.69, we obtain the reduced matrices

$$\mathbf{K}_r = 10000\begin{bmatrix} 20.31 & -6.338 \\ -6.338 & 2.531 \end{bmatrix}, \qquad \mathbf{M}_{rr} = \begin{bmatrix} 0.502 & 0.1 \\ 0.1 & 0.155 \end{bmatrix}$$

An input data file is prepared and program JACOBI is used to solve the eigenvalue problem in Eq. 11.70. The solution is

$$\lambda_1 = 2.025 \times 10^4, \ \ U_r^1 = [0.6401 \ 1.888]^{\mathrm{T}}$$
$$\lambda_2 = 8.183 \times 10^5, \ \ U_r^2 = [1.370 \ -1.959]^{\mathrm{T}}$$

Using Eq. 11.71, we obtain the eigenvector components corresponding to the omitted dof as

$$U_o^1 = [3.61 \quad 4.438]^{\mathrm{T}} \quad \text{and} \quad U_o^2 = [-0.838 - 16.238]^{\mathrm{T}}$$

In this example, the results correlate quite well with the solution of the unreduced system. ■

11.7 RIGID BODY MODES

In certain situations (such as in helicopter frames, flexible spacecraft structures, or flat panels that are placed on soft supports), we are faced with the task of determining mode shapes of structures that are freely suspended in space. These structures have rigid body modes as well as deformation modes. The rigid body modes correspond to translations and rotations of the entire structure along the x, y, and z axes, respectively. Thus, there are six rigid body modes for a three-dimensional body in space. Modes 7, 8, . . . correspond to deformation modes, which are to be determined from an eigenvalue analysis. It should be recognized that the stiffness matrix $\mathbf{K}$ is singular when rigid body modes are present. This follows from the fact that a finite translation or rotation $\mathbf{U}^0$ does not create any internal forces or stresses in the structure. Thus, $\mathbf{KU}^0 = \mathbf{0}$. Since $\mathbf{U}^0 \neq \mathbf{0}$, $\mathbf{K}$ has to be a singular matrix. Further, we can write $\mathbf{KU}^0 = \mathbf{0}$ as $\mathbf{KU}^0 = (0)\,\mathbf{MU}^0$ from which we see that a rigid mode is associated with a zero eigenvalue. Specifically, the first six rigid body modes are associated with six zero eigenvalues.

Steps in many eigenvalue evaluation algorithms given require that $\mathbf{K}$ be nonsingular and also positive definite (i.e., that all eigenvalues be positive). This can be effected by **shifting** as given in Eqs. 11.48–11.50. Thus, with a shift factor $s > 0$, we work with a positive definite matrix $\mathbf{K}_s$ even though the original stiffness matrix is singular.

Using JACOBI or GENEIGEN These methods do not require that the stiffness matrix be positive definite. Thus, they may be directly used on the unconstrained structure. Note that the first six eigenvalues (for a three-dimensional structure) will be zero representing the rigid modes. If small negative values are output as a result of round-off, these may be ignored—avoid taking their square root in computing the frequency within the programs.

Using INVITR Using the inverse iteration program to handle rigid body modes is more involved. We need to define the rigid body modes in the program and then mass normalize them. Let $\mathbf{U}_1^0, \mathbf{U}_2^0, \ldots, \mathbf{U}_6^0$ represent six rigid body modes. After defining these, each is mass normalized as

$$\mathbf{U}_i^0 = \frac{\mathbf{U}_i^0}{(\mathbf{U}_i^0)^{\mathrm{T}}\mathbf{M}(\mathbf{U}_i^0)} \qquad i = 1, \ldots, 6 \tag{11.83}$$

Subsequently, each trial eigenvector is chosen as in Eq. 11.51 from a space M-orthogonal to all previously calculated eigenvectors *including* the six normalized rigid body modes. Rigid body modes may be readily defined as follows: Consider a general three-dimensional body as shown in Fig. 11.10. In general, a node I will have six dof, labeled $Q_{6*I-5}, Q_{6*I-4}, \ldots, Q_{6*I}$, corresponding x, y, and z translations and rotations about x, y, and z axes, respectively. Defining the first mode to be a translation along x-axis, we have $Q(6*I-5, 1) = 1$ and $Q(6*I-4, 1) = Q(6*I-3, 1) = Q(6*I-2, 1) = Q(6*I-1, 1) = Q(6*I, 1) = 0$, where the first subscript is the dof number and the second is the mode number. Similarly, translations along y and z axes define modes $\mathbf{Q}(., 2)$ and $\mathbf{Q}(., 3)$, respectively. Now, consider the sixth rigid body mode corresponding to a rotation of the body about the z-axis, by an angle θ.

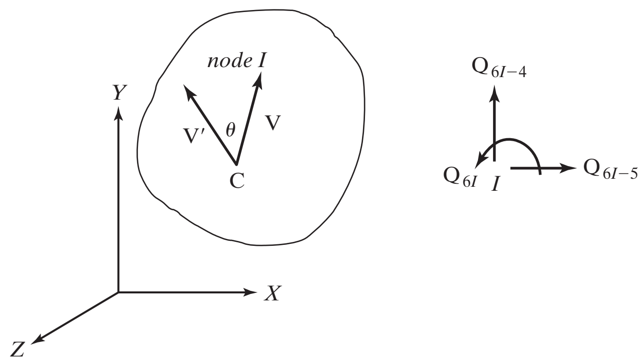


FIGURE 11.10 Rigid body rotation about z-axis.

That is, a rotation in the x–y plane. We can choose an arbitrary value for θ. Choosing the centroid as a reference point about which the body rotates, we can write the translational displacement vector θ of any node I in the body as

$$\delta = \mathbf{V}' - \mathbf{V}, \text{ where } \mathbf{V}' = [\mathbf{R}]\mathbf{V}$$

For this equation, $\mathbf{V} = \mathbf{x}_I - \mathbf{x}_c$ and $\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is a rotation matrix. From δ, we obtain Q(6*I – 5, 6) = δ_x and Q(6*I – 4, 6) = δ_y. Remaining components are Q(6*I – 3, 6) = 0, Q(6*I – 2, 6) = 0, Q(6*I – 1, 6) = 0 and, Q(6*I, 6) = θ (in radians). Rotations about x and y axes can similarly be considered.

An example problem involving rigid body modes is presented in Example 11.8.

Example 11.8

Consider a two-dimensional steel beam, modeled using four elements as shown in Fig. E11.8a. In this beam model, each node has a vertical translational and a counterclockwise rotational dof. No axial dofs are included. Taking the beam length to be 60 mm, E as 200 Gpa, and ρ as 7850 kg/m^3 rectangular cross section of width 6 mm and depth 1 mm (thus, inertia I = 0.5 mm^4), and the shift factor $s = -10^8$, we obtain the first three natural frequencies to be 1440 Hz, 3997 Hz, and 7850 Hz, respectively. The corresponding mode shapes are shown in Fig. E11.8b. Both programs JACOBI and INVITR give similar results. Program JACOBI is easier to implement. In program INVITR, two rigid body modes corresponding to vertical translation and rotation were introduced. Mass normalization of these involved Eq. 11.83, which needed special care as **M** is in banded form.

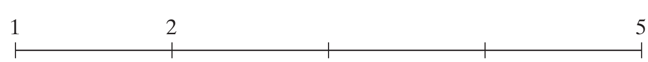


FIGURE E11.8A Unconstrained beam.

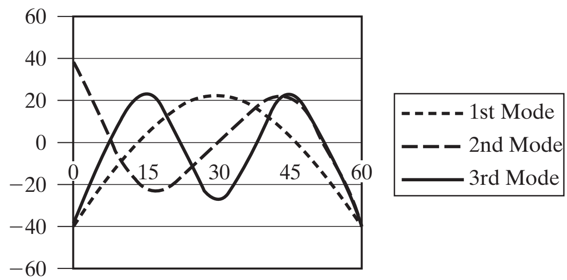


FIGURE E11.8B Mode shapes for unconstrained beam.

■

11.8 CONCLUSION

In this chapter, the application of finite elements for free vibrations is discussed in a general setting using consistent mass matrices. Solution techniques and computer programs are given. These programs can be integrated with static analysis programs to get the dynamic behavior of structures. Natural frequencies and mode shapes of structures give us the needed data concerning what excitation frequencies should be avoided.

Input Data/Output

```
INPUT TO BEAMKM
PROGRAM BeamKM << for Stiffness and Mass Matrices
Example 11.5
NN     NE       NM       NDIM     NEN      NDN
4      3        1        1        2        2
ND     NL       NMPC
2      0        0
NODE#  X-COORD
1      0
2      30
3      70
4      90
ELEM#  N1       N2       MAT#     Mom_In            Area
1      1        2        1        7.85E-01          3.1416
2      2        3        1        7.85E-01          3.1416
3      3        4        1        7.85E-01          3.1416
DOF#   Displ.
1      0
7      0
MAT#   E        Rho << Properties E and MassDensity
1      3.00E+07          0.0007324
SPRING SUPPORTS          <DOF#=0 Exits this
DOF#   Spring Const
0
LUMPED MASSES <DOF#=0 Exits this
DOF#   Lumped Mass
3      0.2072538
5      0.310881
0
```

```
OUTPUT FROM BEAMKM/INPUT TO INVITR, JACOBI, GENEIGEN
Example 11.5
NumDOF        BandWidth
8     4
Banded Stiffness Matrix
70686010472    157080    -10472      157080
3141600       -157080    1570800     0
14889.875     -68722.5   -4417.875   88357.5
5497800       -88357.5   1178100     0
39760.875      265072.5  -35343      353430
7068600       -353430    2356200     0
70686035343   -353430    0           0
4712400        0         0           0
Banded Mass Matrix
0.025638687   0.10847137      0.00887493      -0.064096718
0.591662016   0.064096718     -0.443746512    0
0.267077404   0.084366621     0.01183324      -0.113949722
1.994120128   0.113949722     -1.051843584    0
0.362158375   -0.144628493    0.00591662      -0.02848743
1.577765376   0.02848743      -0.131480448    0
0.017092458   -0.048209498    0               0
0.175307264   0               0               0
Starting Vector for Inverse Iteration
1     1       1       1       1       1       1       1
```

```
OUTPUT - TWO LOWEST EIGENVALUES FROM INVITR
Example 11.5
Eigenvalue Number 1 teration Number 5
Eigenvalue =       4041.9687
Omega =        63.5765
Freq Hz =      10.1185
Eigenvector
2.1E-08 0.05527 1.37830 0.0276 1.0496 -0.0422 2.6E-08 -0.0576
Eigenvalue Number 2   Iteration Number 4
Eigenvalue =  43183.9296
Omega =             207.8074
Freq Hz =            33.0736
Eigenvector
8.97E-08    0.0816    1.3013    -0.0290    -1.2372    0.0051    -1.47E-07    0.0907
```

```
OUTPUT- ALL EIGENVALUES FROM GENEIGEN OR JACOBI
Example 11.5
Eigenval#     1
Eigenval =    4041.9686
Omega =         63.5765
Freq Hz =       10.1185
Eigenvector
-2.1E-08 -0.05527 -1.3783 -0.0276 -1.0496 0.0422 -2.6E-08 0.0576
Eigenval#     2
Eigenval =     43183.9285
Omega =          207.8074
Freq Hz =         33.0736
Eigenvector
-8.96E-08 -0.0816 -1.3015 0.0290 1.2370 -0.0051 1.47E-07 -0.0907
Eigenval#              3
Eigenval =      1207349.4
```

```
Omega =            1098.79
Freq Hz =           174.88
Eigenvector
1.37E-06 0.3977 0.1141 -0.4456 0.3641 0.2618 9.13E-07 -0.1776
Eigenval#                4
Eigenval =       4503773.7
Omega =            2122.21
Freq Hz =           337.76
Eigenvector
4.34E-06 0.7778 -0.5369 -0.171 -0.1284 -0.6238 -3.43E-06 0.4846
Eigenval#                5
Eigenval =        14836541
Omega =             3851.8
Freq Hz =           613.04
Eigenvector
1.14E-05 1.1700 -0.3604 0.7597 -0.1588 0.4768 9.64E-06 -0.9141
Eigenval#                6
Eigenval =        43448547
Omega =            6591.55
Freq Hz =          1049.08
Eigenvector
9.81E-06 0.5056 0.0112 0.5488 0.3804 0.7667 -3.88E-05 2.3385
Eigenval#                7
Eigenval =       1.316E+13
Omega =            3627211
Freq Hz =           577289
Eigenvector
13.6178 -2.7177 0.2118 -0.2643 0.0042 -0.1668 0.8269 0.1029
Eigenval#               8
Eigenval =      1.894E+13
Omega =           4351512
Freq Hz =          692565
Eigenvector
0.992 -0.2690 -0.0037 -0.1167 -0.1353 -0.1832 -16.3370 -4.6521
```

PROBLEMS

11.1. Consider *axial* vibration of the copper bar shown in Fig. P11.1.

(a) Develop the global stiffness and mass matrices.

(b) By hand calculations, determine the lowest natural frequency and mode shape using the inverse iteration algorithm.

(c) Verify your results in (b) using programs INVITR and JACOBI.

(d) Verify the properties in Eqs. 11.41a and 11.41b.

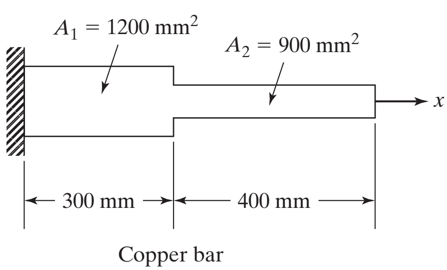


FIGURE P11.1

11.2. By hand calculations, determine the natural frequencies and mode shapes for the rod in P11.1 using the characteristic polynomial technique.

11.3. Use a lumped mass model for the rod in P11.1, and compare the results obtained with the consistent mass model. Use program INVITR or JACOBI.

11.4. Determine all natural frequencies of the simply supported beam shown in Fig. P11.4. Compare the results obtained using the following:

(a) A one-element model.

(b) A two-element model.

(c) Use either program INVITR or JACOBI.

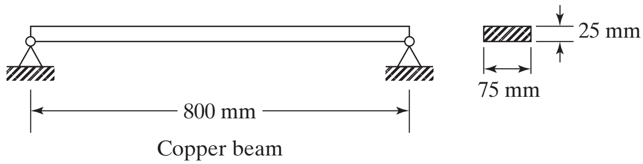


FIGURE P11.4

11.5. Determine, with the help of program BEAMKM, the two lowest natural frequencies (critical speeds) of the steel shaft shown in Fig. P11.5, considering the following cases:

(a) The three journals are like simple supports.

(b) Each journal bearing is like a spring of stiffness equal to 25,000 lb/in.

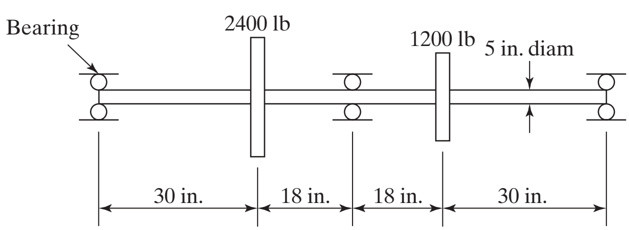


FIGURE P11.5

11.6. The existence of a crack renders an overall reduction in the stiffness of a structure. A crack in a bending member, such as a beam, suggests a slope discontinuity at the section containing the crack, even though the displacement is still continuous there. Thus, the effect of a fracture at a section may be represented by a torsional spring connecting two elements, whose torsional stiffness k may be determined analytically or experimentally.
Consider the cracked cantilever beam shown in Fig. P11.6.

(a) Discuss how you will model this using beam elements. Write down the boundary conditions at the cracked section, and the resulting modifications to the stiffness matrix.

(b) Determine the first three natural frequencies and mode shapes and compare these with those of an uncracked beam of same dimensions. Take $k = 8 \times 10^6$ in.-lb and $E = 30 \times 10^6$ psi.

11.7. A simplified model of an aluminium turbine blade is shown in Fig. P11.7. We want to determine the lowest resonant frequency, with the motion in x direction, and the corresponding mode shape. It is important that we do not excite this resonant frequency to avoid contact of the blades with the casing. The outer ring connecting all the blades is represented as a lumped mass. Use programs CSTKM and INVITR.

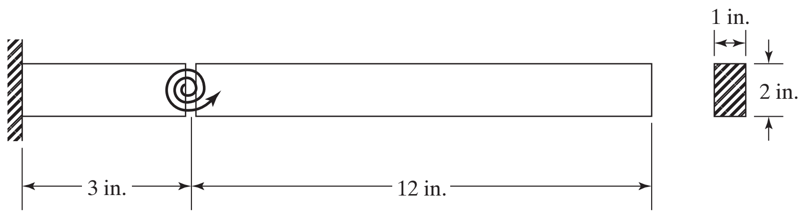


FIGURE P11.6

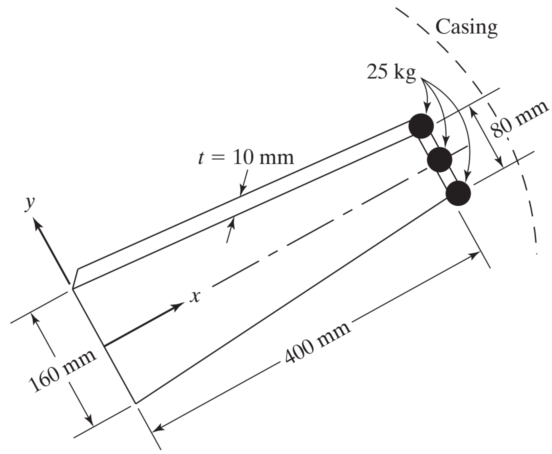


FIGURE P11.7

11.8. Figure P11.8 shows a beam modeled using four-node quadrilateral elements. Develop a program that will generate the banded **K** and **M** matrices. Then use program INVITR to determine the two lowest natural frequencies and mode shapes. Compare your results with those obtained using beam elements.

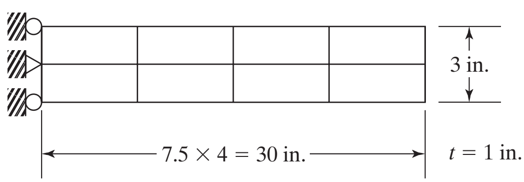


FIGURE P11.8 Steel beam.

11.9. Determine the two lowest natural frequencies and mode shapes for the one-bay, two-story planar steel frame shown in Fig. P11.9. You need to develop a program, analogous to BEAMKM, that will generate the banded **K** and **M** matrices and then use program INVITR.

11.10. For the signal pole arrangement shown in Fig. P11.10, a two-dimensional frame, determine the natural frequencies and mode shapes. (*Note:* Develop a program in line with BEAMKM to write stiffness and mass matrices to a file. Then an eigenvalue routine like INVITR can be run.)

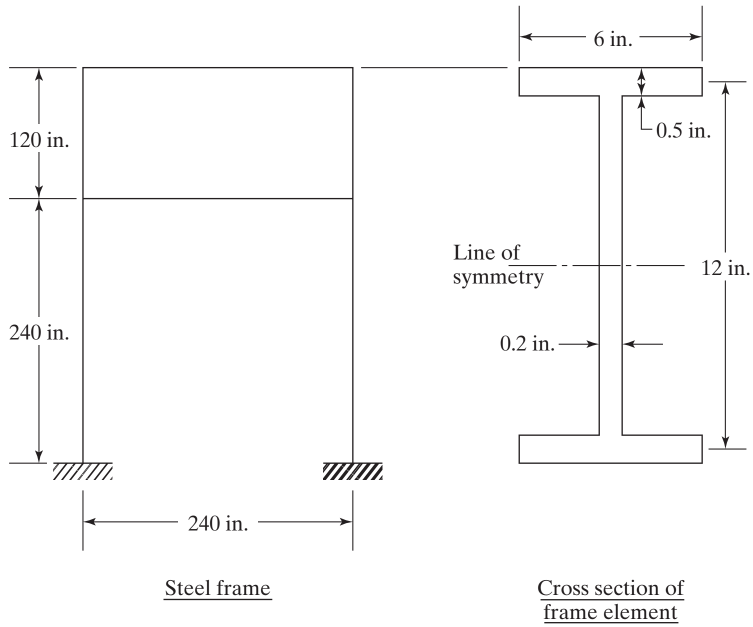


FIGURE P11.9

11.11. Consider the shaft in Example 11.6. Using Guyan reduction, reduce the eight-dof beam model to a two-dof model retaining the translational dof at the flywheels. Compare these frequencies and mode shapes with those obtained from the eight-dof model. Use BEAMKM and JACOBI programs. Also state which modes are missing in the reduced model.

11.12. Reduce the following symmetric matrix to tridiagonal form:

$$\begin{bmatrix} 5 & 2 & 0 & 0 & 1 \\ 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 0 & 3 & 1 & 2 \\ 1 & 0 & 0 & 2 & 1 \end{bmatrix}$$

11.13. Reduce the following two matrices simultaneously to diagonal form using Jacobi's approach:

$$\mathbf{K} = \begin{bmatrix} 4 & 1 & 2 & 0 \\ 1 & 2 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 3 & 3 & 2 & 0 \\ 3 & 1 & 1 & 0 \\ 2 & 1 & 3 & 1 \\ 10 & 0 & 1 & 2 \end{bmatrix}$$

11.14. Consider the beam model shown in Fig. P11.14. Each beam node has a vertical translational dof and a counter-clockwise rotational dof. No axial dofs are included. Take the beam length to be 60 mm, rectangular cross section of width 6 mm and depth 1 mm (thus, $I = 0.5\ \text{mm}^4$), and obtain the first three natural frequencies for the different cases that follow. Plot the mode shapes. (After obtaining the output from eigensolver, which contains nodal displacements of the mode shapes, you can interpolate using Hermite cubic shape

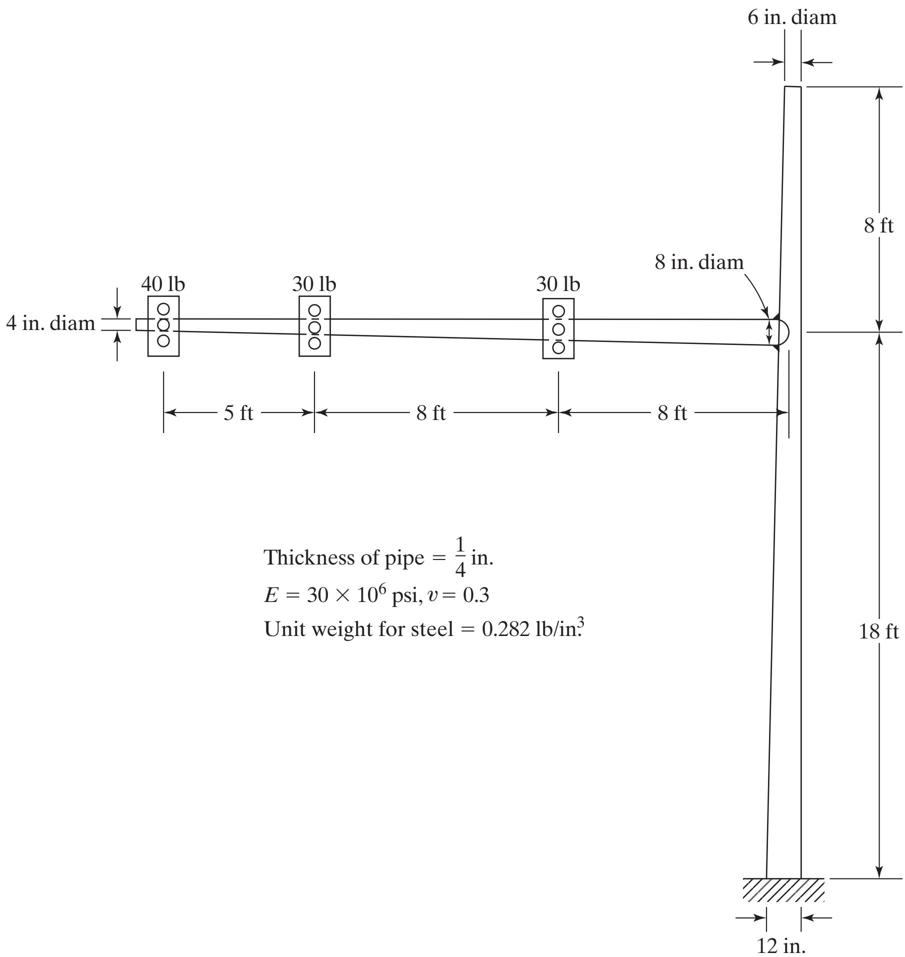


FIGURE P11.10

functions and then use MATLAB or other programs to plot the discretized curve.) Take material to be steel, with E = 200 GPa and ρ= 7850 kg/m^3.

(a) Left end is fixed.

(b) Left end is fixed and a concentrated mass M is attached to the right end (node 5). Take M = 5% of the beam mass.

(c) Beam is unconstrained, and a mass M is attached to the right end as in (b).

For cases (a) and (b), you can use program INVITR, JACOBI, or GENEIGEN. For (c), use program JACOBI or GENEIGEN.

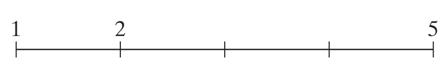


FIGURE P11.14

11.15. A rigid body with mass M and inertia I_c about its center of gravity is welded on to the end of a planar beam element as shown in Fig. P11.15. By writing the kinetic energy of the mass as $\frac{1}{2}M\,\mathrm{v}^2 + \frac{1}{2}I_C\omega^2$, and relating v and ω to $\dot{Q}_1$ and $\dot{Q}_2$, determine the 2×2 mass matrix contribution to the beam node.

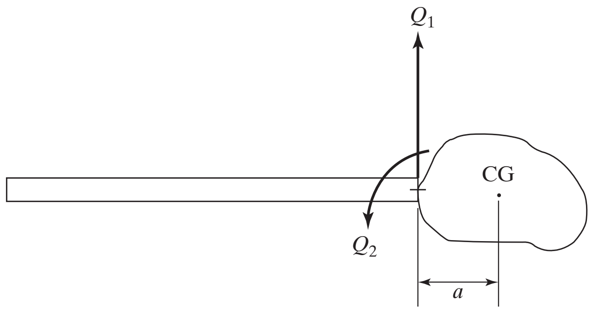


FIGURE P11.15

11.16. Determine the natural frequencies and mode shapes for a coarse axisymmetric triangular meshing of the cylindrical pipe shown in Fig. P11.16. (Modify the AXISYM program to enable it to give an output of stiffness and mass matrices in the format needed for the eigenvalue programs for your solution.)

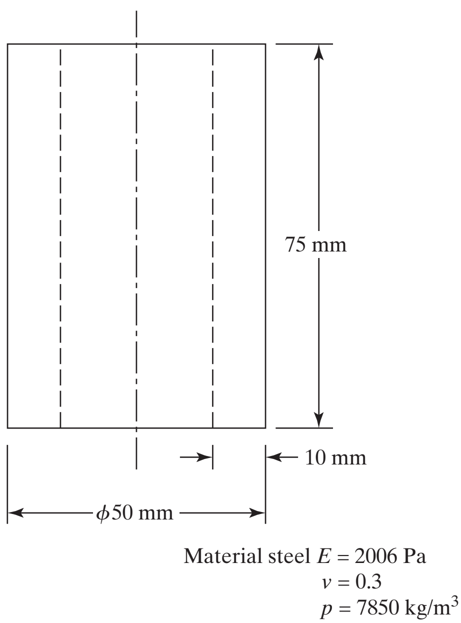


FIGURE P11.16

Program Listings

```
MAIN PROGRAM BEAMKM
'**************************************
'*           PROGRAM BEAMKM           *
'*   STIFFNESS AND MASS GENERATION    *
'* T.R.Chandrupatla and A.D.Belegundu *
Private Sub CommandButton1_Click()
     Call InputData
     Call Bandwidth
     Call StiffnMass
     Call ModifyForBC
     Call AddSprMass
     Call Output
End Sub
```

```
BEAMKM - STIFFNESS AND MASS MATRICES
Private Sub StiffnMass()
     ReDim S(NQ, NBW), GM(NQ, NBW)
     '----- Global Stiffness and Mass Matrices -----
     For N = 1 To NE
         Call ElemStiffMass(N)
         For II = 1 To NEN
             NRT = NDN * (NOC(N, II) - 1)
             For IT = 1 To NDN
                 NR = NRT + IT
                 I = NDN * (II - 1) + IT
                 For JJ = 1 To NEN
                     NCT = NDN * (NOC(N, JJ) - 1)
                     For JT = 1 To NDN
                         J = NDN * (JJ - 1) + JT
                         NC = NCT + JT - NR + 1
                         If NC > 0 Then
                             S(NR, NC) = S(NR, NC) + SE(I, J)
                             GM(NR, NC) = GM(NR, NC) + EM(I, J)
                         End If
                     Next JT
                 Next JJ
             Next IT
         Next II
     Next N
End Sub
```

```
BEAMKM - ELEMENT STIFFNESS AND MASS MATRICES
Private Sub ElemStiffMass(N)
'-------- Element Stiffness and Mass Matrices --------
     N1 = NOC(N, 1)
     N2 = NOC(N, 2)
     M = MAT(NE)
```

```
      EL = Abs(X(N1) - X(N2))
      '--- Element Stiffness
      EIL = PM(M, 1) * SMI(N) / EL ^ 3
         SE(1, 1) = 12 * EIL
         SE(1, 2) = EIL * 6 * EL
         SE(1, 3) = -12 * EIL
         SE(1, 4) = EIL * 6 * EL
            SE(2, 1) = SE(1, 2)
            SE(2, 2) = EIL * 4 * EL * EL
            SE(2, 3) = -EIL * 6 * EL
            SE(2, 4) = EIL * 2 * EL * EL
         SE(3, 1) = SE(1, 3)
         SE(3, 2) = SE(2, 3)
         SE(3, 3) = EIL * 12
         SE(3, 4) = -EIL * 6 * EL
            SE(4, 1) = SE(1, 4)
            SE(4, 2) = SE(2, 4)
            SE(4, 3) = SE(3, 4)
            SE(4, 4) = EIL * 4 * EL * EL
      '--- Element Mass
      RHO = PM(M, 2)
      C1 = RHO * AREA(N) * EL / 420
      EM(1, 1) = 156 * C1
      EM(1, 2) = 22 * EL * C1
      EM(1, 3) = 54 * C1
      EM(1, 4) = -13 * EL * C1
         EM(2, 1) = EM(1, 2)
         EM(2, 2) = 4 * EL * EL * C1
         EM(2, 3) = 13 * EL * C1
         EM(2, 4) = -3 * EL * EL * C1
       EM(3, 1) = EM(1, 3)
       EM(3, 2) = EM(2, 3)
       EM(3, 3) = 156 * C1
       EM(3, 4) = -22 * EL * C1
         EM(4, 1) = EM(1, 4)
         EM(4, 2) = EM(2, 4)
         EM(4, 3) = EM(3, 4)
         EM(4, 4) = 4 * EL * EL * C1
End Sub
```

ADDITION OF SPRING STIFFNESS AND POINT MASSES

```
Private Sub AddSprMass()
'----- Additional Springs and Lumped Masses -----
     LI = LI + 1
     Do
        LI = LI + 1
        N = Val(Worksheets(1).Range("A1").Offset(LI, 0))
        If N = 0 Then Exit Do
        C = Val(Worksheets(1).Range("A1").Offset(LI, 1))
        S(N, 1) = S(N, 1) + C
     Loop
```

continued

```
      LI = LI + 2
      Do
         LI = LI + 1
         N = Val(Worksheets(1).Range("A1").Offset(LI, 0))
         If N = 0 Then Exit Do
         C = Val(Worksheets(1).Range("A1").Offset(LI, 1))
         GM(N, 1) = GM(N, 1) + C
      Loop
End Sub
```

MAIN PROGRAM INVITR

```
'*****      PROGRAM INVITR          *****
'*      Inverse Iteration Method        *
'*  for Eigenvalues and Eigenvectors    *
'*        Searching in Subspace         *
'*         for Banded Matrices          *
'* T.R.Chandrupatla and A.D.Belegundu   *
'****************************************
Private Sub CommandButton1_Click()
      Call InputData
      Call BanSolve1                         '<----Stiffness to Upper Triangle
      Call InverseIter
      Call Output
End Sub
```

INVITR – INPUT OF DATA

```
Private Sub InputData()
      TOL = InputBox("Enter Value", "Tolerance", 0.000001)
      NEV = InputBox("Enter Number", "Number of Eigenvalues Desired", 1)
      SH = 0
      NQ = Val(Worksheets(1).Range("A4"))
      NBW = Val(Worksheets(1).Range("B4"))
      ReDim S(NQ, NBW), GM(NQ, NBW), EV1(NQ), EV2(NQ), NITER(NEV)
      ReDim EVT(NQ), EVS(NQ), ST(NQ), EVC(NQ, NEV), EVL(NEV)
      '============= READ DATA ===============
      '----- Read in Stiffness Matrix -----
      LI = 0
      For I = 1 To NQ
              LI = LI + 1
              For J = 1 To NBW
                  S(I, J) = Val(Worksheets(1).Range("A5").Offset(LI, J - 1))
              Next J
      Next I
      '----- Read in Mass Matrix
      LI = LI + 1
      For I = 1 To NQ
          LI = LI + 1
          For J = 1 To NBW
```

```
            GM(I, J) = Val(Worksheets(1).Range("A5").Offset(LI, J - 1))
        Next J
    Next I
    '----- Starting Vector for Inverse Iteration
    LI = LI + 1
    For I = 1 To NQ
        LI = LI + 1
        ST(I) = Val(Worksheets(1).Range("A5").Offset(LI, 0))
    Next I
    SH = InputBox("SHIFT", "Shift Value for Eigenvalue", 0)
    If SH <> 0 Then
        For I = 1 To NQ: For J = 1 To NBW
            S(I, J) = S(I, J) - SH * GM(I, J)
        Next J: Next I
    End If
End Sub
Private Sub Input2()
    '----- Read in Stiffness Matrix -----
    LI = 0
    LI = LI + 1
    For I = 1 To NQ
        LI = LI + 1
        For J = 1 To NBW
            S(I, J) = Val(Worksheets(1).Range("A5").Offset(LI, J - 1))
        Next J
    Next I
End Sub
```

INVERSE ITERATION ROUTINE

```
Private Sub InverseIter()
    ITMAX = 50: NEV1 = NEV
    PI = 3.14159
    For NV = 1 To NEV
        '--- Starting Value for Eigenvector
        For I = 1 To NQ
            EV1(I) = ST(I)
        Next I
        EL2 = 0: ITER = 0
        Do
            EL1 = EL2
            ITER = ITER + 1
            If ITER > ITMAX Then
                'picBox.Print "No Convergence for Eigenvalue# "; NV
                NEV1 = NV - 1
                Exit Sub
            End If
            If NV > 1 Then
                '----Starting Vector Orthogonal to
                '----Evaluated Vectors
                For I = 1 To NV - 1
                    CV = 0
```

continued

```
                For K = 1 To NQ
                    KA = K - NBW + 1: KZ = K + NBW - 1
                    If KA < 1 Then KA = 1
                    If KZ > NQ Then KZ = NQ
                    For L = KA To KZ
                        If L < K Then
                            K1 = L: L1 = K - L + 1
                        Else
                            K1 = K: L1 = L - K + 1
                        End If
                        CV = CV + EVS(K) * GM(K1, L1) * EVC(L, I)
                    Next L
                Next K
                For K = 1 To NQ
                    EV1(K) = EV1(K) - CV * EVC(K, I)
                Next K
            Next I
        End If
        For I = 1 To NQ
            IA = I - NBW + 1: IZ = I + NBW - 1: EVT(I) = 0
            If IA < 1 Then IA = 1
            If IZ > NQ Then IZ = NQ
            For K = IA To IZ
                If K < I Then
                    I1 = K: K1 = I - K + 1
                Else
                    I1 = I: K1 = K - I + 1
                End If
                EVT(I) = EVT(I) + GM(I1, K1) * EV1(K)
    Next K
    EV2(I) = EVT(I)
Next I
Call BanSolve2   '<--- Reduce Right Side and Solve
C1 = 0: C2 = 0
For I = 1 To NQ
    C1 = C1 + EV2(I) * EVT(I)
Next I
For I = 1 To NQ
    IA = I - NBW + 1: IZ = I + NBW - 1: EVT(I) = 0
    If IA < 1 Then IA = 1
    If IZ > NQ Then IZ = NQ
    For K = IA To IZ
        If K < I Then
            I1 = K: K1 = I - K + 1
        Else
            I1 = I: K1 = K - I + 1
        End If
        EVT(I) = EVT(I) + GM(I1, K1) * EV2(K)
    Next K
Next I
```

```
      For I = 1 To NQ
         C2 = C2 + EV2(I) * EVT(I)
      Next I
      EL2 = C1 / C2
      C2 = Sqr(C2)
      For I = 1 To NQ
         EV1(I) = EV2(I) / C2
         EVS(I) = EV1(I)
      Next I
    Loop While Abs(EL2 - EL1) / Abs(EL2) > TOL
    For I = 1 To NQ
         EVC(I, NV) = EV1(I)
    Next I
      NITER(NV) = ITER
      EL2 = EL2 + SH
      EVL(NV) = EL2
    Next NV
End Sub
```

```
BANDSOLVER FOR MULTIPLE RIGHTHAND SIDES
Private Sub BanSolve1()
'----- Gauss Elimination LDU Approach (for Symmetric Banded Matrices)
     '----- Reduction to Upper Triangular Form
     For K = 1 To NQ - 1
        NK = NQ - K + 1
        If NK > NBW Then NK = NBW
        For I = 2 To NK
           C1 = S(K, I) / S(K, 1)
           I1 = K + I - 1
           For J = I To NK
              J1 = J - I + 1
              S(I1, J1) = S(I1, J1) - C1 * S(K, J)
           Next J
        Next I
     Next K
End Sub
Private Sub BanSolve2()
     '----- Reduction of the right hand side
     For K = 1 To NQ - 1
        NK = NQ - K + 1
        If NK > NBW Then NK = NBW
        For I = 2 To NK: I1 = K + I - 1
           C1 = 1 / S(K, 1)
           EV2(I1) = EV2(I1) - C1 * S(K, I) * EV2(K)
        Next I
     Next K
     '----- Back Substitution
     EV2(NQ) = EV2(NQ) / S(NQ, 1)
     For II = 1 To NQ - 1
        I = NQ - II: C1 = 1 / S(I, 1)
        NI = NQ - I + 1
```

continued

```
        If NI > NBW Then NI = NBW
        EV2(I) = C1 * EV2(I)
        For K = 2 To NI
           EV2(I) = EV2(I) - C1 * S(I, K) * EV2(I + K - 1)
        Next K
     Next II
End Sub
```

```
INVITR- OUTPUT (EXCEL)
Private Sub Output()
' Now, writing out the results in a different worksheet
Worksheets(2).Cells.ClearContents
Worksheets(2).Range("A1").Offset(0, 0) = "Results from Program INVITR"
Worksheets(2).Range("A1").Offset(0, 0).Font.Bold = True
Worksheets(2).Range("A1").Offset(1, 0) = Worksheets(1).Range("A2")
Worksheets(2).Range("A1").Offset(1, 0).Font.Bold = True
      LI = 1
      If NEV1 < NEV Then
              LI = LI + 1
              Worksheets(2).Range("A1").Offset(LI, 0) = "Convergence of "
              Worksheets(2).Range("A1").Offset(LI, 1) = NEV1
              Worksheets(2).Range("A1").Offset(LI, 2) = " Eigenvalues Only."
              NEV = NEV1
      End If
      For NV = 1 To NEV
         LI = LI + 1
         Worksheets(2).Range("A1").Offset(LI, 0) = "Eigenvalue Number "
         Worksheets(2).Range("A1").Offset(LI, 1) = NV
         Worksheets(2).Range("A1").Offset(LI, 2) = "Iteration Number "
         Worksheets(2).Range("A1").Offset(LI, 3) = NITER(NV)
         LI = LI + 1
         Worksheets(2).Range("A1").Offset(LI, 0) = "Eigenvalue = "
         Worksheets(2).Range("A1").Offset(LI, 1) = EVL(NV)
         OMEGA = Sqr(EVL(NV)): FREQ = 0.5 * OMEGA / PI
         LI = LI + 1
         Worksheets(2).Range("A1").Offset(LI, 0) = "Omega = "
         Worksheets(2).Range("A1").Offset(LI, 1) = OMEGA
         LI = LI + 1
         Worksheets(2).Range("A1").Offset(LI, 0) = "Freq Hz ="
         Worksheets(2).Range("A1").Offset(LI, 1) = FREQ
         LI = LI + 1
         Worksheets(2).Range("A1").Offset(LI, 0) = "Eigenvector"
         LI = LI + 1
         For I = 1 To NQ
             Worksheets(2).Range("A1").Offset(LI, I - 1) = EVC(I, NV)
         Next I
      Next NV
End Sub
```

CHAPTER 12

Preprocessing and Postprocessing

12.1 INTRODUCTION

Finite element analysis involves three stages of activity: preprocessing, processing, and postprocessing. Preprocessing involves the preparation of data, such as nodal coordinates, connectivity, boundary conditions, and loading and material information. The processing stage involves stiffness generation, stiffness modification, and solution of equations, resulting in the evaluation of nodal variables. Other derived quantities, such as gradients or stresses, may be evaluated at this stage. The processing stage is presented in detail in earlier chapters, where the data were prepared in a formatted input file. The postprocessing stage deals with the presentation of results. Typically, the deformed configuration, mode shapes, temperature, and stress distribution are computed and displayed at this stage. A complete finite element analysis is a logical interaction of the three stages. The preparation of data and postprocessing require considerable effort if all data are to be handled manually. The tedium of handling the data and the possibility of errors creeping in as the number of elements increase are discouraging factors for the finite element analyst. In the following sections, we present a systematic development of preprocessing and postprocessing considerations. This should make finite element analysis a more interesting computational tool. We first present a general-purpose mesh generation scheme for two-dimensional plane problems.

12.2 MESH GENERATION

Region and Block Representation

The basic idea of a mesh-generation scheme is to generate element connectivity data and nodal-coordinate data by reading in input data for a few key points. We present here the theory and computer implementation of a mesh-generation scheme suggested by Zienkiewicz and Philips.[1] In this scheme, a complex region is divided into eight-noded

[1] Zienkiewicz, O.C., and D.V. Philips, "An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' coordinates." *International Journal for Numerical Methods in Engineering* 3: 519–528 (1971).

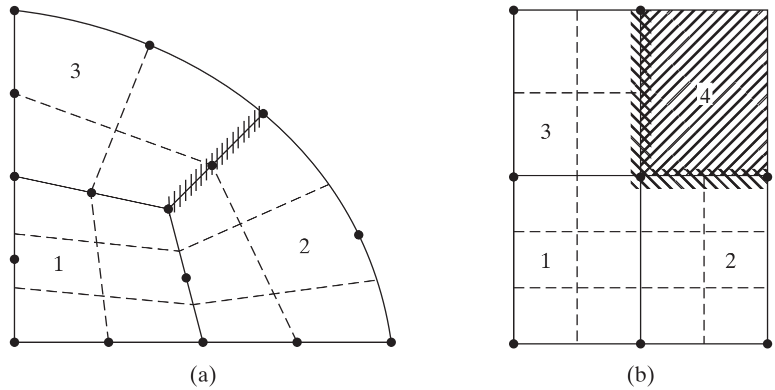


FIGURE 12.1 (a) Region and (b) block diagram.

quadrilaterals, which are then viewed in the form of a rectangular block pattern. Consider the region shown in Fig. 12.1. The full rectangular block pattern is convenient for node numbering. To match the pattern in the region, the block number 4 is to be treated as void and the two hatched edges need to be merged. In general, a complex region is viewed as a rectangle, composed of rectangular blocks, with some blocks left as void and some edges identified to be merged.

Block Corner Nodes, Sides, and Subdivisions

A general configuration of the full rectangle composed of blocks is shown in Fig. 12.2. We represent the sides of the rectangle as S and W, with respective numbers of spans of NS and NW. For consistent coordinate mapping, S, W, and the third coordinate direction Z must form a right-hand system. For mesh generation, each span is subdivided. Spans KS and KW are divided into $NSD(KS)$ and $NWD(KW)$ divisions, respectively. Since the node numbering will be carried out in the S direction first and incremented in the W direction next, the bandwidth of resulting matrices will be small if total number of divisions in the S direction is less than the total number in the W direction. S and W are chosen to represent short and wide directions in this sense. In this scheme, the bandwidth is a minimum when there are no void blocks and there is no side merging. We note here that the total number of nodes in the S and W directions are

$$NNS = 1 + \sum_{KS=1}^{NS} NSD(KS)$$

$$NNW = 1 + \sum_{KW=1}^{NW} NWD(KW) \tag{12.1}$$

The maximum possible nodes for quadrilateral or triangular division is taken as $NNT(=NNS \times NNW)$. We define an array $NNAR(NNT)$ to define the nodes in the problem. We also define a block identifier array $IDBLK(NSW)$, which stores the material number in the location representing the block. A zero is stored in the location corresponding to a void block. The x- and y-coordinates of all valid block corner nodes are read into $XB(NGN, 2)$. The program is given for planar regions. By introducing the

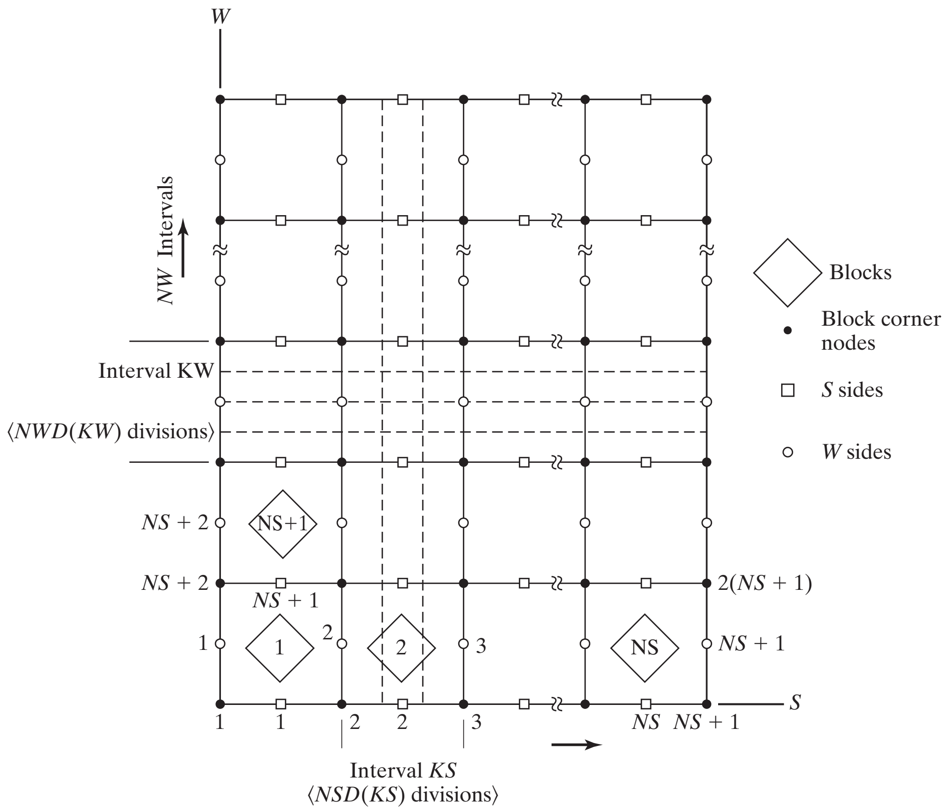


FIGURE 12.2 Numbering of corner nodes and sides.

z-coordinate, three-dimensional surfaces can be modeled. Two arrays, $SR(NSR, 2)$ and $WR(NWR, 2)$, are used for storing the coordinates of the nodes on the corresponding sides. First, we generate the nodes for all sides, assuming that the side is a straight line and the node is at the midpoint between the corner nodes. This represents the default configuration. Then, for sides that are curved and for those straight sides with nodes not located at physical midpoints, the x- and y-coordinates are read into $SR(.,2)$ and $WR(.,2)$, at appropriate locations. The sides to be merged are identified by the end node numbers of the sides. We now discuss the node numbering and coordinate-generation schemes.

Generation of Node Numbers. We present the node-numbering strategy by means of an example. Consider the region and block representation shown in Fig. 12.1. The node numbering scheme is shown in Fig. 12.3. We have two blocks in the S direction and two in the W direction. Block 4 is void. Array $NNAR(30)$ has all the locations defined. Edges 18–20 and 18–28 are to be merged. We first initialize the array $NNAR(30)$ by putting –1 at each of its locations. We then cover each of the void blocks and put *zero* where nodes do not exist. Existence of neighboring blocks is checked in implementing this process. For side merging, at the node locations of the side with higher node numbers, the location numbers of the corresponding nodes of the merging side are entered.

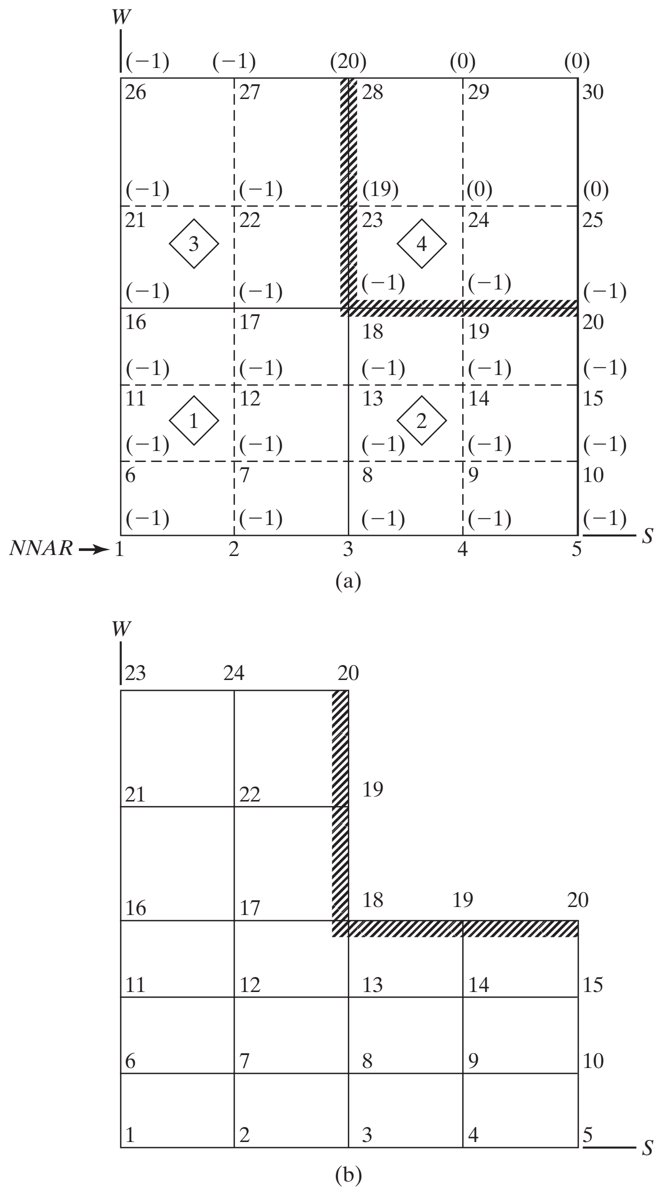


FIGURE 12.3 Node numbering.

The final node numbering is a simple process. We sweep along S and then increment along W. The node numbers are incremented by 1 whenever the location has a negative value. When the value is zero, it is skipped. If the location has a positive value, it indicates side merging and the corresponding node number from the location indicated by the value is inserted. The scheme is simple and nodal coordinate checking is not necessary in this process.

Generation of Coordinates and Connectivity. Here we use the shape functions for isoparametric mapping for an eight-noded quadrilateral developed in Chapter 8. We refer to Fig. 12.4, which establishes the relationships for the master block or $\xi-\eta$ block, the S–W block, and the region block or x–y block. The first step is one of extracting the global coordinates of corner and midside nodes of the block under consideration. For a general node $N1$, the ξ- and η-coordinates are obtained using the number of divisions.

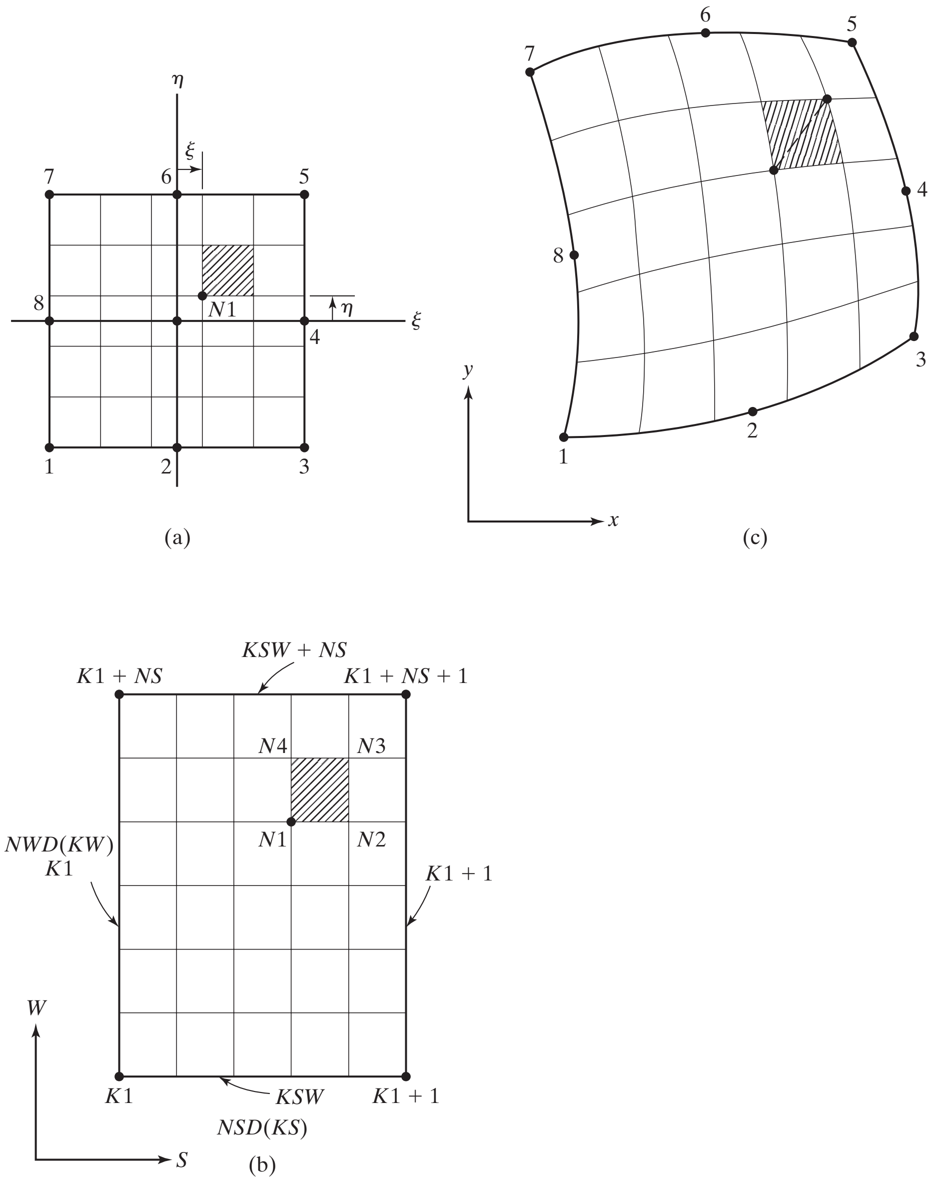


FIGURE 12.4 Coordinates and connectivity: (a) master block for shape functions, (b) block for node numbers, and (c) block in region.

The coordinates of $N1$ are given by

$$x = \sum_{I=1}^{8} SH(I) \cdot X(I)$$
$$y = \sum_{I=1}^{8} SH(I) \cdot Y(I) \tag{12.2}$$

where $SH(\ )$ are shape functions and $X(\ )$ and $Y(\ )$ are corner node coordinates. For the small rectangular shaded division with lower left corner $N1$, shown in Fig. 12.4, the other three nodes $N2$, $N3$, and $N4$ are computed. For quadrilateral elements, we use $N1$–$N2$–$N3$–$N4$ as the element, with the first element of the block starting at the lower left corner. The element numbers for the next block start after the last number of the previous block. For triangular element division, each rectangle is divided into two triangles, $N1$–$N2$–$N3$ and $N3$–$N4$–$N1$. The triangular division is readjusted to connect the diagonal of shorter length. The process of coordinate and connectivity generation is skipped for void blocks.

This is a general-purpose mesh-generation scheme with the capability to model complex problems. This scheme can be readily generalized to model three-dimensional surfaces by introducing the z-coordinate. To illustrate the use of the program, we consider a few examples.

Examples of Mesh Generation. In the first example shown in Fig. 12.5, there are four blocks. The default material number for all blocks is 1. Material number for

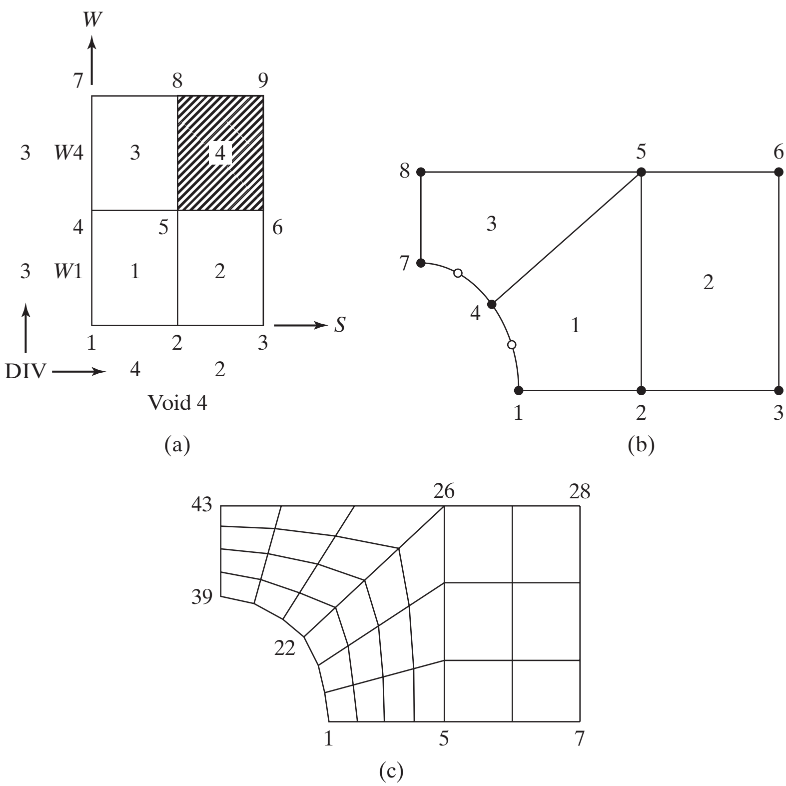


FIGURE 12.5 Example mesh 1: (a) block diagram, (b) region, and (c) mesh.

block 4 is read in as zero to represent void space. S spans 1 and 2 are divided into four and two divisions, respectively. W spans 1 and 2 are each divided into three divisions. The coordinates of corner nodes 1–8 and the coordinates of midpoints of curved sides $W1$ and $W4$ are read in. The generated mesh with node numbers is also shown in Fig. 12.5. If triangular mesh is desired, the shorter diagonal of each quadrilateral will be joined.

In the second example, shown in Fig. 12.6, we model a full annular region. To achieve a minimum bandwidth, the block diagram shown in Fig. 12.6a is suggested. Blocks 2 and 5 are void space. The side 1–2 merges with 4–3, and side 9–10 merges with 12–11. Coordinates of all corner nodes and the midpoints of $W1, W2, \ldots, W8$ of the block diagram need to be given. The resulting mesh for the span divisions shown in the block diagram is given in Fig. 12.6c.

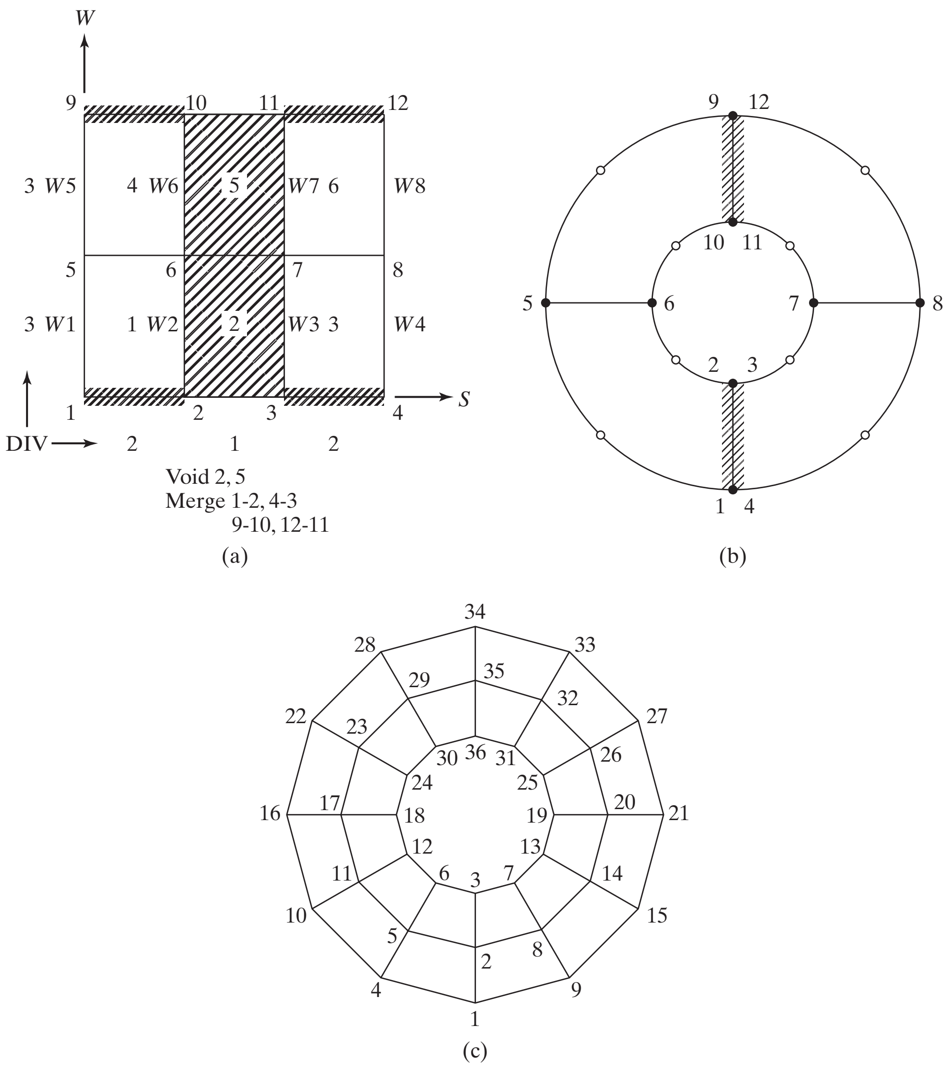


FIGURE 12.6 Example mesh 2: (a) block diagram, (b) region, and (c) mesh.

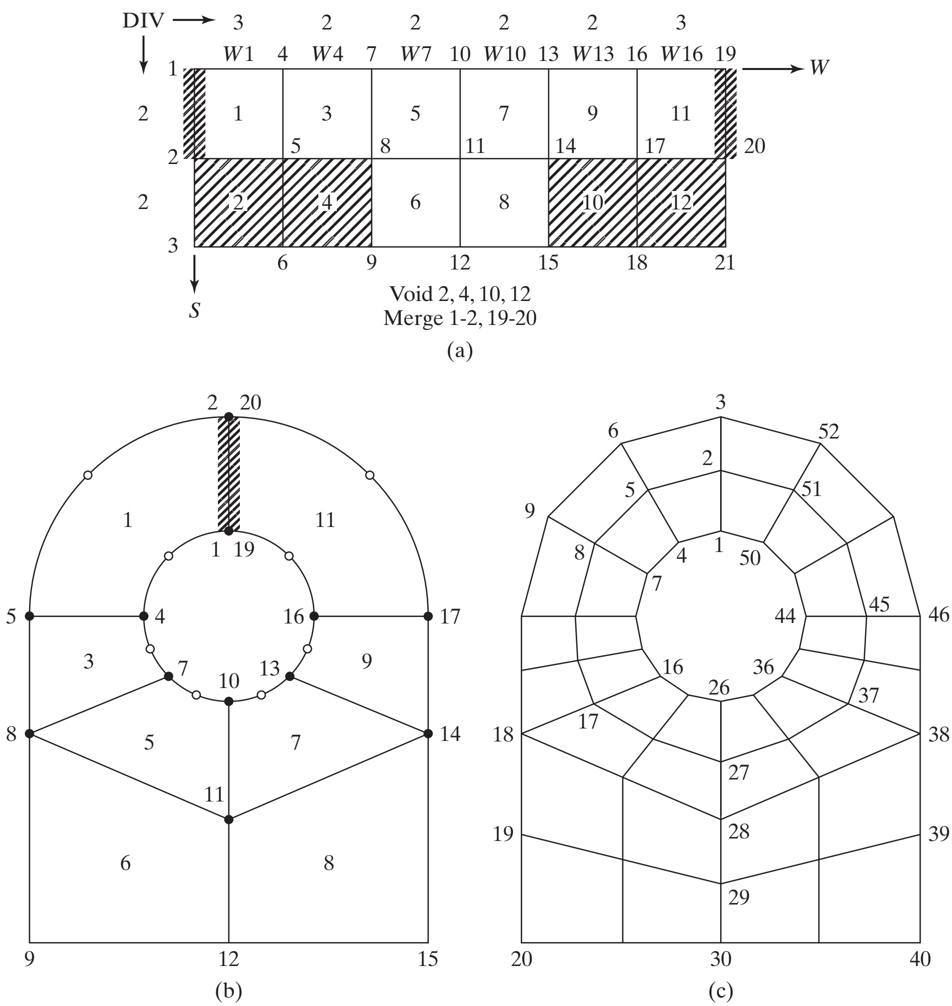


FIGURE 12.7 Example mesh 3: (a) block diagram, (b) region, and (c) mesh.

Figure 12.7 shows an eyelet. The full geometric shape is modeled. The block diagram shows void blocks and span divisions. Merging sides are indicated. Coordinates of all corner points of the block diagram are to be read in. The coordinates of midpoints of curved sides $W1$, $W2$, $W4$, $W7$, $W10$, $W13$, $W16$, and $W17$ have to be input. The mesh is shown for quadrilateral elements.

Division of a region and making a block diagram form the first step in the preparation of data for mesh generation.

Mesh Plotting. The generated data are saved in a file. The convenient way of reviewing the coordinate and connectivity data is by plotting it using the computer. The plots will quickly reveal if there are any errors. Points to be readjusted can easily be identified. The program PLOT2D can be used for plotting two-dimensional meshes on the screen. In mesh plotting, we scan each element and draw the element boundaries using the connectivity information. The coordinate bounds must first be adjusted for the screen resolution and size.

Data Handling and Editing. In simple problems with small number of elements and nodes, it is convenient to prepare data directly using a text editor. For larger problems, the user may generate the data files by using the MESHGEN program. Output of the MESHGEN program essentially consists of nodal coordinates and element connectivity. A text editor is then used to add loading, boundary conditions, material properties, and some other information to the mesh data file. The format for the data file is common for all problems and is given on the inside front cover of this book. Importantly, an example input file is provided at the end of every chapter. For two-dimensional problems, the program PLOT2D can be used to read the data and plot meshes on the screen.

The data thus created can be processed by the finite element programs presented in earlier chapters. The finite element program processes the data and calculates nodal variable quantities, such as displacements and temperatures, and element quantities, such as stresses and gradients. The stage is now set for postprocessing.

12.3 POSTPROCESSING

We discuss here the aspects of plotting a displaced configuration, plotting nodal data in the form of contour plots, such as isotherms and isobars, and conversion of element-oriented data into best fitting nodal values. We restrict our discussion here to two-dimensional problems; however, the ideas can be extended to three-dimensional problems with some additional effort.

Deformed Configuration and Mode Shape

Plotting a deformed or displaced shape is a simple extension of PLOT2D. If the displacements or components of the eigenvector are read into the matrix $U(NN, 2)$ and the coordinates are stored in $X(NN, 2)$, we can define the displaced position matrix $XP(NN, 2)$ so that

$$XP(I, J) = X(I, J) + \alpha U(I, J) \qquad \begin{array}{l} J = 1, 2 \\ I = 1, \ldots, NN \end{array} \tag{12.3}$$

where α is a magnification factor so chosen that the largest component of $\alpha U(I, J)$ is of reasonable proportion in relation to the body size and NN represents the number of nodes. One may try this largest component to be about 10% of the body-size parameter. In the program PLOT2D, we need to make changes to read displacements $U(NN, 2)$ decide the value of α, and replace $\mathbf{X}$ by $\mathbf{X} + \alpha\mathbf{U}$.

Contour Plotting

Contour plotting of a scalar nodal variable such as temperature is straightforward for three-noded triangular elements. We consider the variable f on one triangular element shown in Fig. 12.8. The nodal values are f_1, f_2, and f_3 at the three nodes 1, 2, and 3, respectively. The function f is interpolated using the linear shape functions used for the constant strain triangle. f represents a plane surface with values f_1, f_2, and f_3 at the three nodes. We check for each desired level. Say $\hat{f}$ represents a typical level for contour map. If $\hat{f}$ lies in the interval $f_2 - f_3$, it also lies in one of the intervals $f_1 - f_2$ or $f_1 - f_3$. Say it lies in the interval $f_2 - f_3$, as shown in Fig. 12.8. Then f has the value of $\hat{f}$ at points A and B and is constant along the line AB. Determination of the coordinates of points A and B will give us the contour line AB. The coordinates of point A can be obtained from

$$\begin{aligned} \xi &= \frac{\hat{f} - f_2}{f_3 - f_2} \\ x_A &= \xi x_3 + (1 - \xi)x_2 \\ y_A &= \xi y_3 + (1 - \xi)y_2 \end{aligned} \tag{12.4}$$

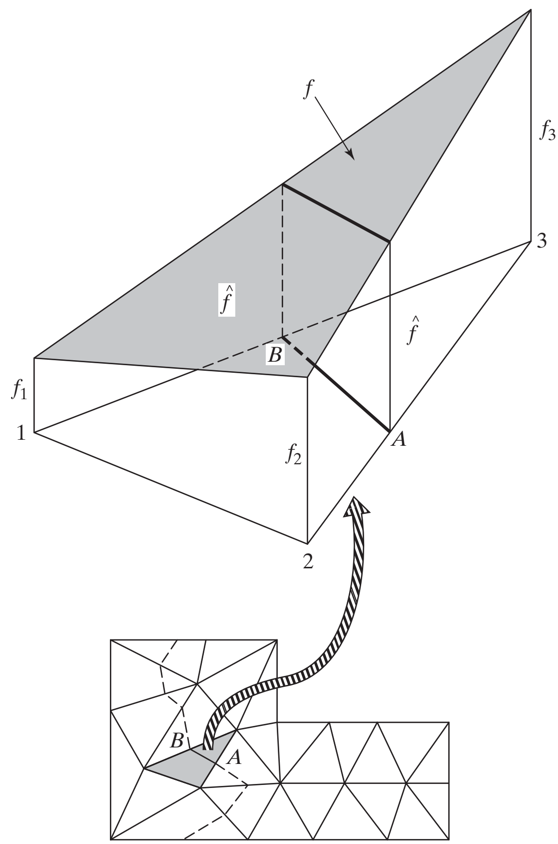


FIGURE 12.8 Constant level of variation f.

The coordinates of point B can be obtained by replacing the indices 2 and 3 by 1 and 3, respectively.

The program CONTOURA plots the variable FF represented by its nodal values. The coordinate, connectivity, and function data are read in from data files. In the first part of the program, the boundary limits are set on the screen. The function limits are found and the number of contour levels is read in. The boundary of the region is plotted on the screen. Each element is then scanned for the function levels and the constant value lines are drawn. The result is a contour map.

In addition, in CONTOURA, the number of levels is fixed at 10 and each level is associated with a distinct color. Violet is the lowest level, and red is set as the highest level, with intermediate colors roughly in the order of the rainbow spectrum. CONTOURB uses the idea of filling the color in a closed subregion of an element. Thus, for the same data used for CONTOURA, CONTOURB plots color bands. Both CONTOURA and CONTOURB also work for four-noded quadrilateral elements. The contour plot idea presented for the triangle is also used for the quadrilateral by introducing an interior point and considering the four triangles. There are other contour algorithms specifically for quadrilaterals, and interested readers are encouraged to search the literature in this area.

There are also some quantities, such as stresses, temperature, and velocity gradients, which are constant over triangular elements. For these, the contour mapping requires the evaluation of nodal values. We present here the procedure for evaluating the nodal values for least-squares fit. The procedure discussed is useful in diverse situations, such as smoothing data obtained in image processing. The least-squares fit for a four-noded quadrilateral is also presented following the best fit for the triangle.

Nodal Values from Known Constant Element Values for a Triangle

We evaluate the nodal values that minimize the least-squares error. We consider here triangular elements with constant function values. A triangular element having function value f_e is shown in Fig. 12.9. Let f_1, f_2, and f_3 be the local nodal values. The interpolated function is given by

$$f = \mathbf{NF} \tag{12.5}$$

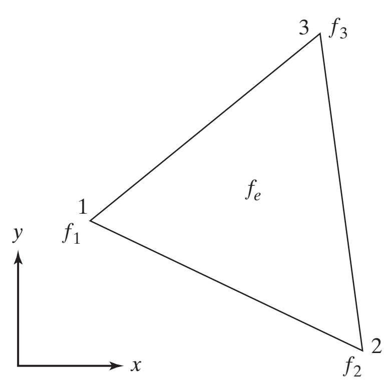


FIGURE 12.9 Triangular element for least-squares fit study.

where

$$\mathbf{N} = [N_1 \quad N_2 \quad N_3] \tag{12.6}$$

is the vector of shape functions and

$$\mathbf{f} = [f_1 \quad f_2 \quad f_3]^{\mathrm{T}} \tag{12.7}$$

The squared error may be represented by

$$E = \sum_e \frac{1}{2} \int_e (f - f_e)^2 \, dA \tag{12.8}$$

On expanding and substituting from Eq. 12.5, we get

$$E = \sum_e \left[\frac{1}{2} \mathbf{f}^{\mathrm{T}} \left(\int_e \mathbf{N}^{\mathrm{T}} \mathbf{N} \, dA \right) \mathbf{f} - \mathbf{f}^{\mathrm{T}} \left(f_e \int_A \mathbf{N}^{\mathrm{T}} dA \right) + \frac{1}{2} f_e^2 A \right] \tag{12.9}$$

Noting that the last term is a constant, we write the equation in the form

$$\mathbf{E} = \sum_e \left[\frac{1}{2} \mathbf{f}^{\mathrm{T}} \mathbf{W}^e \mathbf{f} - \mathbf{f}^{\mathrm{T}} \mathbf{R}^e \right] + \text{constant} \tag{12.10}$$

where

$$\mathbf{W}^e = \int_e \mathbf{N}^{\mathrm{T}} \mathbf{N} \, dA = \frac{A_e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{12.11}$$

$$\mathbf{R}^e = f_e \int_e \mathbf{N}^{\mathrm{T}} \, dA = \frac{f_e A_e}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \tag{12.12}$$

$\int_e \mathbf{N}^{\mathrm{T}} \mathbf{N} \, dA$ is similar to the evaluation of mass matrix for a triangle in Chapter 11. On assembling the stiffness $\mathbf{W}^e$ and load vector from $\mathbf{R}^e$ we get

$$E = \frac{1}{2} \mathbf{F}^{\mathrm{T}} \mathbf{W} \mathbf{F} - \mathbf{F}^{\mathrm{T}} \mathbf{R} + \text{constant} \tag{12.13}$$

where $\mathbf{F}$ is the global nodal-value vector given by

$$\mathbf{F} = [F_1 \quad F_2 \quad \cdots \quad F_{NN}]^{\mathrm{T}} \tag{12.14}$$

For least-squares error, setting the derivatives of E with respect to each F_i to be zero, we get

$$\mathbf{WF} = \mathbf{R} \tag{12.15}$$

Here $\mathbf{W}$ is a banded symmetric matrix. The set of equations is solved using the equation solving techniques used in other finite element programs. The program BESTFIT takes the mesh data and element value data $FS(NE)$ and evaluates the nodal data $F(NN)$ for a three-noded triangle. Only the nodal coordinate and element connectivity data are needed from the original program input file.

Least-Squares Fit for a Four-Noded Quadrilateral

Let $\mathbf{q} = [q_1\, q_2\, q_3\, q_4]^T$ represent the element nodal values to be determined for least-squares fit defined using error at four interior points. If $\mathbf{s} = [s_1\, s_2\, s_3\, s_4]^T$ represents the vector of interpolated values at the four interior points, and $\mathbf{a} = [a_1\, a_2\, a_3\, a_4]^T$ represents the actual values of the variable (see Fig. 12.10), the error may be defined as

$$\begin{aligned} \epsilon &= \sum_{\mathbf{e}} (\mathbf{s} - \mathbf{a})^T(\mathbf{s} - \mathbf{a}) \\ &= \sum_{\mathbf{e}} (\mathbf{s}^T\mathbf{s} - 2\mathbf{s}^T\mathbf{a} - \mathbf{a}^T\mathbf{a}) \end{aligned} \tag{12.16}$$

The four interior points are generally taken as the Gaussian integration points. The stress values match well at these points. If

$$\mathbf{N} = \begin{bmatrix} N_1^1 & N_2^1 & N_3^1 & N_4^1 \\ N_1^2 & N_2^2 & N_3^2 & N_4^2 \\ N_1^3 & N_2^3 & N_3^3 & N_4^3 \\ N_1^4 & N_2^4 & N_3^4 & N_4^4 \end{bmatrix} \tag{12.17}$$

where N_j^i represents the shape function N_j evaluated at interior point I, then $\mathbf{s}$ can be written as

$$\mathbf{s} = \mathbf{Nq} \tag{12.18}$$

Inserting this into Eq. 12.16, we find that the error becomes

$$\epsilon = \sum_{\mathbf{e}} \mathbf{q}^T\mathbf{N}^T\mathbf{N}\mathbf{q} - 2\mathbf{q}^T\mathbf{N}^T\mathbf{a} + \mathbf{a}^T\mathbf{a} \tag{12.19}$$

Noting that $\mathbf{N}^T\mathbf{N}$ is similar to element stiffness $\mathbf{k}^e$, and $\mathbf{N}^T\mathbf{a}$ is similar to the element force vector, the stiffness and force-vector assembly can be made. The assembled matrix equations can be put in the form

$$\mathbf{KQ} = \mathbf{F} \tag{12.20}$$

The solution of this set of equations gives $\mathbf{Q}$, which is the vector of nodal values of the variable considered for least-squares fit of the element values. This least-squares fit is implemented in the program BESTFITQ.

Element quantities such as maximum shear stress, von Mises stress, and temperature gradient can be converted to nodal values and then contour plotting can be done.

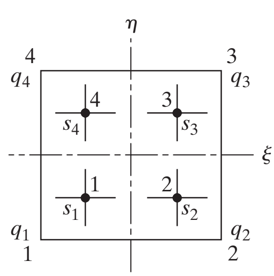


FIGURE 12.10 Least-squares fit for a quadrilateral.

The use of computer programs BESTFIT and CONTOUR has already been discussed in Chapter 6 (Example 6.9).

12.4 CONCLUSION

Preprocessing and postprocessing are integral parts of finite element analysis. The general-purpose mesh-generation scheme can model a variety of complex regions. One needs to use some imagination in preparing the block representation of the region. Definition of void blocks and merging of sides enables one to model multiple connected regions. The node numbering gives sparse matrices and in many cases should give minimum bandwidth by proper block representation. Mesh plotting shows the element layout. The data handling program is a dedicated routine for finite element data preparation and data editing. Ideas for the plotting of deformed configuration and mode shapes can be readily implemented into the programs included here. Contour plottings for triangular and quadrilateral elements have been presented, and programs are included. The computation of nodal values that best fit the element values takes some of the very same steps used in the development of finite elements in earlier chapters.

Finite element analysis involves solution of a wide variety of problems in solid mechanics, fluid mechanics, heat transfer, electrical and magnetic fields, and other areas. Problem solving involves large amounts of data that must be systematically handled and clearly presented. The ideas developed in this chapter should make preparation and handling of input and output data an interesting endeavor rather than a tedious task.

Example 12.1

The quadrant shown in Fig. E12.1 is meshed using program MESHGEN. The input data given are constructed from the display in Fig. E12.1. Connectivity and nodal coordinate data are contained in the output file, and a plot of the mesh can be obtained by running program PLOT2D.

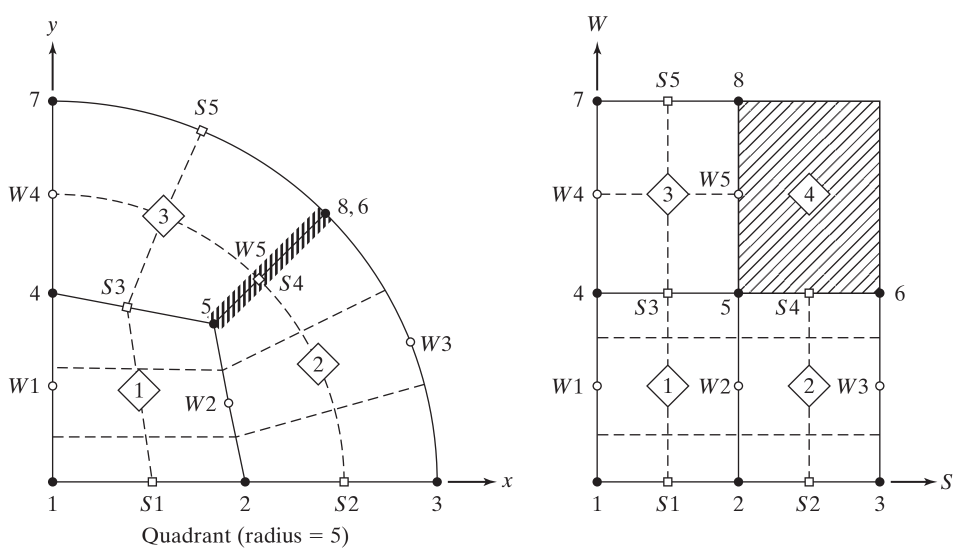


FIGURE E12.1

■

Input Data/Output

```
INPUT TO MESHGEN
MESH GENERATION
Example 12.1
Number of Nodes per Element <3 or 4>
3
BLOCK DATA                  NS=#S-Spans
NS    NW     NSJ            NW=#W-Spans
2     2      1              NSJ=#PairsOfEdgesMerged
SPAN DATA
S-Span#      #Div  (for each S-Span/Single division = 1)
1     2
2     2
W-Span#      #Div (for each W-Span/Single division = 1) TempRise(NCH=2 El
Char: Th Temp)
1     3
2     2
BLOCK MATERIAL DATA
Block#       Material  (Void => 0   Block# = 0 completes this data)
4     0
0
BLOCK CORNER DATA
Corner#      X-Coord     Y-Coord  (Corner# = 0 completes this data)
1     0      0
2     2.5    0
3     5      0
4     0      2.5
5     1.8    1.8
6     3.536  3.536
7     0      5
8     3.536  3.536
0
MID POINT DATA FOR CURVED OR GRADED SIDES
S-Side#      X-Coord     Y-Coord (Sider# = 0 completes this data)
5     1.913  4.619
0
W-Side#      X-Coord     Y-Coord (Sider# = 0 completes this data)
3     4.619  1.913
0
MERGING SIDES (Node1 is the lower number)
Pair#  Sid1Nod1  Sid1Nod2  Sid2Nod1  Sid2Nod2
1      5         6         5         8
```

```
FOR MESHGEN OUTPUT EXAMPLE SEE Chapter 6.
PLOT2D.XLS plots mesh without node numbers
Executable programs are given for PLOT2D, CONTOURA, and CONTOURB
```

```
INPUT TO BESTFIT (File input for C, FORTRAN, and MATLAB Programs)
Geometry Data for BESTFIT same as for CST or AXISYM
Example 6.7
NN     NE     NM     NDIM   NEN    NDN
4      2      1      2      3      2
ND     NL     NMPC
5      1      0
Node#  X      Y
1      3      0
2      3      2
3      0      2
4      0      0
Elem#  N1     N2     N3     Mat#     Thickness     TempRise
1      4      1      2      1        0.5           0
2      3      4      2      1        0.5           0
Element Data - <1> for Input from Sheet1  <2> for File Input
1
Element Values (If above number is 1 give element values here)
524.9
298.7
```

```
OUTPUT FROM BESTFIT
NODAL VALUES FROM SHEET2
Nodal Values
638
411.8
185.6
411.8

Executable programs for BESTFIT and BESTFITQ are given for file input
```

INPUT FOR PLOT2D IS THE INPUT FOR CST, AXISYM, QUAD, OR AXIQUAD

PROBLEMS

12.1. Use program MESHGEN to generate finite element meshes for the regions in Figs. P12.1a and P12.1b. Generate meshes using both triangular and quadrilateral elements. For the fillet in P12.1a, use $y = 42.5 - 0.5x + x^2/360$.

12.2. Generate a “graded” mesh for the region in Fig. P12.1a so that there are more elements near the left edge of the region. That is, the mesh density reduces along the $+x$ direction. Use MESHGEN with displaced midside nodes.

12.3. Use program CONTOUR to draw isotherms for the temperature distribution obtained in Example 10.4.

12.4. After solving Problem 6.17 using program CST, complete the following:

(a) Use program PLOT2D to plot the original and deformed shape. The deformed shape requires selecting a scaling factor and using Eq. 12.3.

(b) Use programs BESTFIT and CONTOUR and plot contours of maximum principal stress.

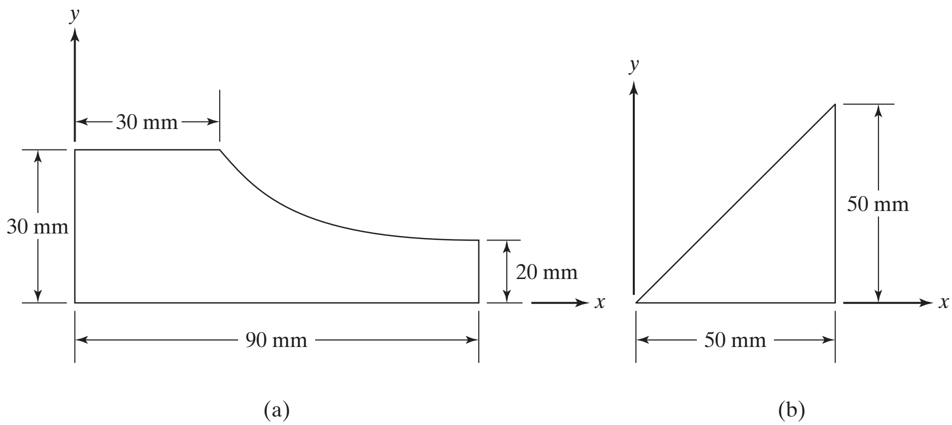


FIGURE P12.1

12.5. Plot the mode shapes of the beam in Problem 11.4. For this, you will need to modify PLOT2D and interface with INVITR.

12.6. This problem illustrates the concept of a **dedicated** finite element program. Only design related parameters are input to the program, while mesh generation, boundary conditions and loading definition, finite element analysis, and postprocessing are automatically performed.

Consider the flywheel in Fig. P12.6. By modifying and interfacing programs MESHGEN, PLOT2D, AXISYM, BESTFIT, and CONTOUR, develop a dedicated program that requires the user to input only the overall dimensions r_h, r_i, r_o, t_h, and t_f and the values of E, ν, ρ, and ω. Your program may consist of independent programs executed through a batch or command file or can consist of one single program. Include the following features:

(a) a printout of all input data and output displacements and stresses and

(b) a plot of original and deformed shapes

Solve Problem 7.7. Provide contour plots of stress components.

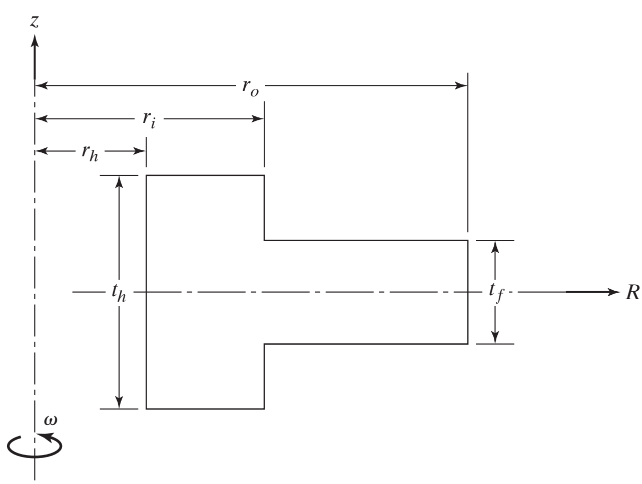


FIGURE P12.6

12.7. Plot shearing-stress contours for the torsion Problem 10.18.

12.8. Use MESHGEN for creating a coarse mesh for the plate with two holes shown in Fig. P12.8. Plot the mesh using PLOT2D.

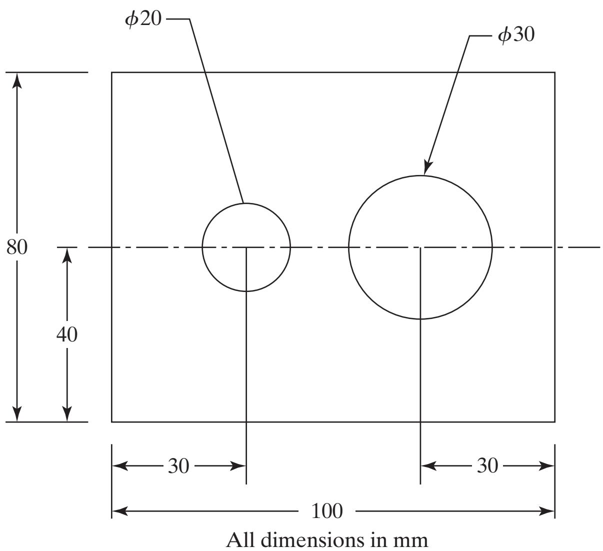


FIGURE P12.8

Program Listing

```
'*************************************************
'*              PROGRAM MESHGEN                  *
'*  MESH GENERATOR FOR TWO DIMENSIONAL REGIONS   *
'*     (c) T.R.CHANDRUPATLA & A.D.BELEGUNDU      *
'*************************************************
Private Sub cmdEnd_Click()
   End
End Sub
'=====  MAIN PROGRAM ======
Private Sub cmdStart_Click()
     Call InputData
     Call GlobalNode
     Call CoordConnect
     Call Output
     cmdView.Enabled = True
     cmdStart.Enabled = False
End Sub
'===============================
```

```
'=====  INPUT DATA FROM FILE ======
Private Sub InputData()
     File1 = InputBox("Input File d:\dir\fileName.ext", "Name of File")
     Open File1 For Input As #1
```

```
'=============  READ DATA  ===============
Line Input #1, Dummy: Line Input #1, Title
Line Input #1, Dummy
Input #1, NEN
' NEN = 3 for Triangle  4 for Quad
If NEN < 3 Then NEN = 3
If NEN > 4 Then NEN = 4
'Hints:  A region is divided into 4-cornered blocks viewed as a
'    mapping from a Checkerboard pattern of S- and W- Sides
'    S- Side is one with lower number of final divisions
'    Blocks, Corners, S- and W- Sides are labeled as shown in Fig. 12.2
'    Make a sketch and identify void blocks and merging sides
'----- Block Data -----
'#S-Spans(NS)  #W-Spans(NW)  #PairsOfEdgesMerged(NSJ)
Line Input #1, Dummy: Line Input #1, Dummy
Input #1, NS, NW, NSJ
NSW = NS * NW: NGN = (NS + 1) * (NW + 1): NM = 1
ReDim IDBLK(NSW), NSD(NS), NWD(NW), NGCN(NGN), SH(8)
'------------- Span Divisions ---------------
Line Input #1, Dummy
NNS = 1: NNW = 1
'--- Number of divisions for each S-Span
Line Input #1, Dummy
For KS = 1 To NS
   Input #1, N
   Input #1, NSD(N)
   NNS = NNS + NSD(N)
Next KS
'--- Number of divisions for each W-Span
Line Input #1, Dummy
For KW = 1 To NW
   Input #1, N
   Input #1, NWD(N)
   NNW = NNW + NWD(N)
Next KW

'--- Block Material Data
Input #1, Dummy: Input #1, Dummy
'-------- Block Identifier / Material# (Default# is 1) --------
For I = 1 To NSW: IDBLK(I) = 1: Next I
Do
   Input #1, NTMP
   If NTMP = 0 Then Exit Do
   Input #1, IDBLK(NTMP)
   If NM < IDBLK(NTMP) Then NM = IDBLK(NTMP)
Loop

'-----------------   Block Corner Data  ---------------
NSR = NS * (NW + 1): NWR = NW * (NS + 1)
ReDim XB(NGN, 2), SR(NSR, 2), WR(NWR, 2)
Input #1, Dummy: Input #1, Dummy
Do
   Input #1, NTMP
   If NTMP = 0 Then Exit Do
   Input #1, XB(NTMP, 1)
   Input #1, XB(NTMP, 2)
Loop
```

continued

```
    '---------- Evaluate Mid-points of S-Sides -------------
    For I = 1 To NW + 1
       For J = 1 To NS
           IJ = (I - 1) * NS + J
           SR(IJ, 1) = 0.5 * (XB(IJ + I - 1, 1) + XB(IJ + I, 1))
           SR(IJ, 2) = 0.5 * (XB(IJ + I - 1, 2) + XB(IJ + I, 2))
       Next J
    Next I
    '---------- Evaluate Mid-points of W-Sides -------------
    For I = 1 To NW
       For J = 1 To NS + 1
           IJ = (I - 1) * (NS + 1) + J
           WR(IJ, 1) = 0.5 * (XB(IJ, 1) + XB(IJ + NS + 1, 1))
           WR(IJ, 2) = 0.5 * (XB(IJ, 2) + XB(IJ + NS + 1, 2))
       Next J
    Next I
    '------ Mid Points for Sides that are curved or graded ------
    Line Input #1, Dummy: Line Input #1, Dummy
    '--- S-Sides
    Do
       Input #1, NTMP
       If NTMP = 0 Then Exit Do
       Input #1, SR(NTMP, 1)
       Input #1, SR(NTMP, 2)
    Loop
    Line Input #1, Dummy
    '--- W-Sides
    Do
       Input #1, NTMP
       If NTMP = 0 Then Exit Do
       Input #1, WR(NTMP, 1)
       Input #1, WR(NTMP, 2)
    Loop
    '--------- Merging Sides ----------
    If NSJ > 0 Then
       Input #1, Dummy: Input #1, Dummy
       ReDim MERG(NSJ, 4)
       For I = 1 To NSJ
           Input #1, N
           Input #1, L1
           Input #1, L2
           Call SideDiv(L1, L2, IDIV1)
           Input #1, L3
           Input #1, L4
           Call SideDiv(L3, L4, IDIV2)
           If IDIV1 <> IDIV2 Then
              picBox.Print "#Div don't match. Check merge data."
              End
           End If
           MERG(I, 1) = L1: MERG(I, 2) = L2
           MERG(I, 3) = L3: MERG(I, 4) = L4
       Next I
    End If
    Close #1
End Sub
'========================================================
```

```
'=====  GLOBAL NODE NUMBERS FOR THE MESH ======
Private Sub GlobalNode()
      '------- Global Node Locations of Corner Nodes ---------
      NTMPI = 1
      For I = 1 To NW + 1
         If I = 1 Then IINC = 0 Else IINC = NNS * NWD(I - 1)
         NTMPI = NTMPI + IINC: NTMPJ = 0
         For J = 1 To NS + 1
            IJ = (NS + 1) * (I - 1) + J
            If J = 1 Then JINC = 0 Else JINC = NSD(J - 1)
            NTMPJ = NTMPJ + JINC: NGCN(IJ) = NTMPI + NTMPJ
         Next J
      Next I
      '---------------- Node Point Array --------------------
      NNT = NNS * NNW
      ReDim NNAR(NNT)
      For I = 1 To NNT: NNAR(I) = -1: Next I
      '--------- Zero Non-Existing Node Locations ---------
      For KW = 1 To NW
         For KS = 1 To NS
            KSW = NS * (KW - 1) + KS
            If IDBLK(KSW) <= 0 Then
               '-------- Operation within an Empty Block --------
               K1 = (KW - 1) * (NS + 1) + KS: N1 = NGCN(K1)
               NS1 = 2: If KS = 1 Then NS1 = 1
               NW1 = 2: If KW = 1 Then NW1 = 1
               NS2 = NSD(KS) + 1
               If KS < NS Then
                     If IDBLK(KSW + 1) > 0 Then NS2 = NSD(KS)
                  End If
                  NW2 = NWD(KW) + 1
                  If KW < NW Then
                    If IDBLK(KSW + NS) > 0 Then NW2 = NWD(KW)
                  End If
                  For I = NW1 To NW2
                    IN1 = N1 + (I - 1) * NNS
                    For J = NS1 To NS2
                            IJ = IN1 + J - 1: NNAR(IJ) = 0
                    Next J
          Next I
          ICT = 0
          If NS2 = NSD(KS) Or NW2 = NWD(KW) Then ICT = 1
          If KS = NS Or KW = NW Then ICT = 1
          If ICT = 0 Then
             If IDBLK(KSW + NS + 1) > 0 Then NNAR(IJ) = -1
          End If
         End If
       Next KS
      Next KW
      '--------  Node Identification for Side Merging ------
      If NSJ > 0 Then
         For I = 1 To NSJ
            I1 = MERG(I, 1): I2 = MERG(I, 2)
            Call SideDiv(I1, I2, IDIV)
            IA1 = NGCN(I1): IA2 = NGCN(I2)
            IASTP = (IA2 - IA1) / IDIV
```

continued

```
            I1 = MERG(I, 3): I2 = MERG(I, 4)
            Call SideDiv(I1, I2, IDIV)
            IB1 = NGCN(I1): IB2 = NGCN(I2)
            IBSTP = (IB2 - IB1) / IDIV
            IAA = IA1 - IASTP
            For IBB = IB1 To IB2 Step IBSTP
               IAA = IAA + IASTP
               If IBB = IAA Then NNAR(IAA) = -1 Else NNAR(IBB) = IAA
            Next IBB
         Next I
      End If
      '----------  Final Node Numbers in the Array  --------
      NODE = 0
      For I = 1 To NNT
         If NNAR(I) > 0 Then
            II = NNAR(I): NNAR(I) = NNAR(II)
         ElseIf NNAR(I) < 0 Then
            NODE = NODE + 1: NNAR(I) = NODE
         End If
      Next I
End Sub
Private Sub SideDiv(I1, I2, IDIV)
      '===========  Number of Divisions  for Side I1,I2  ===========
      IMIN = I1: IMAX = I2
      If IMIN > I2 Then
         IMIN = I2
         IMAX = I1
      End If
      If (IMAX - IMIN) = 1 Then
         IDIV = NGCN(IMAX) - NGCN(IMIN)
      Else
         IDIV = (NGCN(IMAX) - NGCN(IMIN)) / NNS
      End If
End Sub
'===================================================================
```

```
'=====  COORDINATES AND CONNECTIVITY ======
Private Sub CoordConnect()
      '------------  Nodal Coordinates  ---------------
      NN = NODE: NELM = 0
      ReDim X(NN, 2), XP(8, 2), NOC(2 * NNT, NEN), MAT(2 * NNT)
      For KW = 1 To NW
         For KS = 1 To NS
         KSW = NS * (KW - 1) + KS
         If IDBLK(KSW) <> 0 Then
             '---------  Extraction of Block Data  ----------
             NODW = NGCN(KSW + KW - 1) - NNS - 1
             For JW = 1 To NWD(KW) + 1
                ETA = -1 + 2 * (JW - 1) / NWD(KW)
                NODW = NODW + NNS: NODS = NODW
                For JS = 1 To NSD(KS) + 1
                     XI = -1 + 2 * (JS - 1) / NSD(KS)
```

```
                    NODS = NODS + 1: NODE = NNAR(NODS)
                    Call BlockXY(KW, KSW)
                    Call Shape(XI, ETA)
                    For J = 1 To 2
                       C1 = 0
                       For I = 1 To 8
                          C1 = C1 + SH(I) * XP(I, J)
                       Next I
                       X(NODE, J) = C1
                    Next J
                    '----------------- Connectivity ----------------
                    If JS <> NSD(KS) + 1 And JW <> NWD(KW) + 1 Then
                       N1 = NODE: N2 = NNAR(NODS + 1)
                       N4 = NNAR(NODS + NNS): N3 = NNAR(NODS + NNS + 1)
                       NELM = NELM + 1
                       If NEN = 3 Then
                          '------------- Triangular Elements ------------
                          NOC(NELM, 1) = N1: NOC(NELM, 2) = N2
                          NOC(NELM, 3) = N3: MAT(NELM) = IDBLK(KSW)
                          NELM = NELM + 1: NOC(NELM, 1) = N3: NOC(NELM, 2) = N4
                          NOC(NELM, 3) = N1: MAT(NELM) = IDBLK(KSW)
                       Else
                          '------------- Quadrilateral Elements ----------
                          NOC(NELM, 1) = N1: NOC(NELM, 2) = N2
                          MAT(NELM) = IDBLK(KSW)
                          NOC(NELM, 3) = N3: NOC(NELM, 4) = N4
                       End If
                    End If
               Next JS
            Next JW
         End If
      Next KS
   Next KW
      NE = NELM
      If NEN = 3 Then
      '--------- Readjustment for Triangle Connectivity ----------
         NE2 = NE / 2
         For I = 1 To NE2
            I1 = 2 * I - 1: N1 = NOC(I1, 1): N2 = NOC(I1, 2)
            N3 = NOC(I1, 3): N4 = NOC(2 * I, 2)
            X13 = X(N1, 1) - X(N3, 1): Y13 = X(N1, 2) - X(N3, 2)
            X24 = X(N2, 1) - X(N4, 1): Y24 = X(N2, 2) - X(N4, 2)
            If (X13 * X13 + Y13 * Y13) > 1.1 * (X24 * X24 + Y24 * Y24) Then
               NOC(I1, 3) = N4: NOC(2 * I, 3) = N2
            End If
         Next I
      End If
End Sub

Private Sub BlockXY(KW, KSW)
      '====== Coordinates of 8-Nodes of the Block ======
      N1 = KSW + KW - 1
      XP(1, 1) = XB(N1, 1): XP(1, 2) = XB(N1, 2)
      XP(3, 1) = XB(N1 + 1, 1): XP(3, 2) = XB(N1 + 1, 2)
      XP(5, 1) = XB(N1 + NS + 2, 1): XP(5, 2) = XB(N1 + NS + 2, 2)
      XP(7, 1) = XB(N1 + NS + 1, 1): XP(7, 2) = XB(N1 + NS + 1, 2)
```

```
      XP(2, 1) = SR(KSW, 1): XP(2, 2) = SR(KSW, 2)
      XP(6, 1) = SR(KSW + NS, 1): XP(6, 2) = SR(KSW + NS, 2)
      XP(8, 1) = WR(N1, 1): XP(8, 2) = WR(N1, 2)
      XP(4, 1) = WR(N1 + 1, 1): XP(4, 2) = WR(N1 + 1, 2)
End Sub
Private Sub Shape(XI, ETA)
      '==============  Shape Functions  ================
      SH(1) = -(1 - XI) * (1 - ETA) * (1 + XI + ETA) / 4
      SH(2) = (1 - XI * XI) * (1 - ETA) / 2
      SH(3) = -(1 + XI) * (1 - ETA) * (1 - XI + ETA) / 4
      SH(4) = (1 - ETA * ETA) * (1 + XI) / 2
      SH(5) = -(1 + XI) * (1 + ETA) * (1 - XI - ETA) / 4
      SH(6) = (1 - XI * XI) * (1 + ETA) / 2
      SH(7) = -(1 - XI) * (1 + ETA) * (1 + XI - ETA) / 4
      SH(8) = (1 - ETA * ETA) * (1 - XI) / 2
End Sub
```

```
'=============   OUTPUT    ================
Private Sub Output()
      '===== Output from this program is input for FE programs after some changes
      File2 = InputBox("Output File d:\dir\fileName.ext", "Name of File")
      Open File2 For Output As #2
      Print #2, "Program MESHGEN - CHANDRUPATLA & BELEGUNDU"
      Print #2, Title
      NDIM = 2: NDN = 2
      Print #2, "NN  NE  NM  NDIM  NEN  NDN"
      Print #2, NN; NE; NM; NDIM; NEN; NDN
      Print #2, "ND   NL   NMPC"
      Print #2, ND; NL; NMPC
      Print #2, "Node#   X    Y"
      For I = 1 To NN
         Print #2, I;
         For J = 1 To NDIM
            Print #2, X(I, J);
         Next J
         Print #2,
      Next I
      Print #2, "Elem#  Node1  Node2  Node3";
      If NEN = 3 Then Print #2, "  Material#"
      If NEN = 4 Then Print #2, "  Node4  Material#"
      For I = 1 To NE
         Print #2, I;
         For J = 1 To NEN
            Print #2, NOC(I, J);
         Next J
         Print #2, MAT(I)
      Next I
      Close #2
      picBox.Print "Data has been stored in the file "; File2
End Sub
```

```
'*********************************************
'*              PROGRAM PLOT2D               *
'*    PLOTS 2D MESHES - TRIANGLES AND QUADS  *
'*   (c) T.R.CHANDRUPATLA & A.D.BELEGUNDU    *
'*********************************************
`========    PROGRAM MAIN    ========
Private Sub cmdPlot_Click()
      Call InputData
      Call DrawLimits(XMIN, YMIN, XMAX, YMAX)
      Call DrawElements
      cmdPlot.Enabled = False
      cmdULeft.Enabled = True
      cmdURight.Enabled = True
      cmdLLeft.Enabled = True
      cmdLRight.Enabled = True
End Sub
`===========================================
```

```
`=====    INPUT DATA FROM FE INPUT FILE    =====
Private Sub InputData()
      File1 = InputBox("Input File d:\dir\fileName", "Name of File")
      Open File1 For Input As #1
      Line Input #1, Dummy: Input #1, Title
      Line Input #1, Dummy: Input #1, NN, NE, NM, NDIM, NEN, NDN
      Line Input #1, Dummy: Input #1, ND, NL, NMPC
      If NDIM <> 2 Then
         picBox.Print "THE PROGRAM SUPPORTS TWO DIMENSIONAL PLOTS ONLY"
         picBox.Print "THE DIMENSION OF THE DATA IS  "; NDIM
         End
      End If
      ReDim X(NN, NDIM), NOC(NE, NEN)
      '=============  READ DATA  ===============
      Line Input #1, Dummy
      For I = 1 To NN: Input #1, N: For J = 1 To NDIM
      Input #1, X(N, J): Next J: Next I
      Line Input #1, Dummy
      For I = 1 To NE: Input #1, N: For J = 1 To NEN
      Input #1, NOC(N, J): Next J: Input #1, NTMP
            For J = 1 To 2: Input #1, C: Next J
      Next I
      Close #1
End Sub
`================================================
```

```
`========    DETERMINE DRAW LIMITS    ========
Private Sub DrawLimits(XMIN, YMIN, XMAX, YMAX)
      XMAX = X(1, 1): YMAX = X(1, 2): XMIN = X(1, 1): YMIN = X(1, 2)
      For I = 2 To NN
         If XMAX < X(I, 1) Then XMAX = X(I, 1)
         If YMAX < X(I, 2) Then YMAX = X(I, 2)
         If XMIN > X(I, 1) Then XMIN = X(I, 1)
         If YMIN > X(I, 2) Then YMIN = X(I, 2)
```

continued

```
      Next I
      XL = (XMAX - XMIN): YL = (YMAX - YMIN)
      A = XL: If A < YL Then A = YL
      XB = 0.5 * (XMIN + XMAX)
      YB = 0.5 * (YMIN + YMAX)
      XMIN = XB - 0.55 * A: XMAX = XB + 0.55 * A
      YMIN = YB - 0.55 * A: YMAX = YB + 0.55 * A
      XL = XMIN: YL = YMIN: XH = XMAX: YH = YMAX
      XOL = XL: YOL = YL: XOH = XH: YOH = YH
End Sub
'=========================================================
```

```
'========    DRAW ELEMENTS    ========
Private Sub DrawElements()
      '===========  Draw Elements  ================
      picBox.Scale (XL, YH)-(XH, YL)
      picBox.Cls
      For IE = 1 To NE
        For II = 1 To NEN
         I2 = II + 1
         If II = NEN Then I2 = 1
         X1 = X(NOC(IE, II), 1): Y1 = X(NOC(IE, II), 2)
         X2 = X(NOC(IE, I2), 1): Y2 = X(NOC(IE, I2), 2)
         picBox.Line (X1, Y1)-(X2, Y2), QBColor(1)
         If NEN = 2 Then Exit For
        Next II
      Next IE
      cmdNode.Enabled = True
End Sub
'===================================================
```

```
'*****        PROGRAM BESTFIT       *****
'*          BEST FIT PROGRAM            *
'*        FOR 3-NODED TRIANGLES         *
'* T.R.Chandrupatla and A.D.Belegundu   *
'***************************************
'========    PROGRAM MAIN    ========
Private Sub cmdStart_Click()
      Call InputData
      Call Bandwidth
      Call Stiffness
      Call BandSolver
      Call Output
      cmdView.Enabled = True
      cmdStart.Enabled = False
End Sub
'============================================
```

```
'=====    STIFFNESS FOR INTERPOLATION    =====
Private Sub Stiffness()
      ReDim S(NQ, NBW), F(NQ)
      '---  Global Stiffness Matrix
      For N = 1 To NE
         Call ElemStiff(N)
         For II = 1 To 3
            NR = NOC(N, II): F(NR) = F(NR) + FE(II)
            For JJ = 1 To 3
               NC = NOC(N, JJ) - NR + 1
               If NC > 0 Then
                  S(NR, NC) = S(NR, NC) + SE(II, JJ)
               End If
            Next JJ
         Next II
      Next N
      picBox.Print "Stiffness Formation completed..."
End Sub
Private Sub ElemStiff(N)
      '--- Element Stiffness Formation
      I1 = NOC(N, 1): I2 = NOC(N, 2): I3 = NOC(N, 3)
      X1 = X(I1, 1): Y1 = X(I1, 2)
      X2 = X(I2, 1): Y2 = X(I2, 2)
      X3 = X(I3, 1): Y3 = X(I3, 2)
      X21 = X2 - X1: X32 = X3 - X2: X13 = X1 - X3
      Y12 = Y1 - Y2: Y23 = Y2 - Y3: Y31 = Y3 - Y1
      DJ = X13 * Y23 - X32 * Y31     'DETERMINANT OF JACOBIAN
      AE = Abs(DJ) / 24
      SE(1, 1) = 2 * AE: SE(1, 2) = AE: SE(1, 3) = AE
      SE(2, 1) = AE: SE(2, 2) = 2 * AE: SE(2, 3) = AE
      SE(3, 1) = AE: SE(3, 2) = AE: SE(3, 3) = 2 * AE
      A1 = FS(N) * Abs(DJ) / 6
      FE(1) = A1: FE(2) = A1: FE(3) = A1
End Sub
'=======================================================
```

```
'********          CONTOURA          **********
'*    CONTOUR PLOTTING - CONTOUR LINES         *
'*    FOR 2D TRIANGLES AND QUADRILATERALS      *
'*    T.R.Chandrupatla and A.D.Belegundu       *
'**********************************************
'=======    PROGRAM MAIN    =======
Private Sub cmdPlot_Click()
      Call InputData
      Call FindBoundary
      Call DrawLimits(XMIN, YMIN, XMAX, YMAX)
      Call DrawBoundary
      Call DrawContours
End Sub
'===========================================
```

```
'=====              INPUT DATA FROM FILES          =====
Private Sub InputData()
      File1 = InputBox("FE Input File", "d:\dir\Name of File")
      File2 = InputBox("Contour Data File", "d:\dir\Name of File")
      Open File1 For Input As #1
      Line Input #1, D$: Input #1, Title$
      Line Input #1, D$: Input #1, NN, NE, NM, NDIM, NEN, NDN
      Line Input #1, D$: Input #1, ND, NL, NMPC
      If NDIM <> 2 Or NEN < 3 Or NEN > 4 Then
         picBox.Print "This program supports triangular and quadrilateral"
         picBox.Print "Elements only."
         End
      End If
      ReDim X(NN, NDIM), NOC(NE, NEN), FF(NN), NCON(NE, NEN)
      ReDim XX(3), YY(3), U(3), IC(10), ID(10)
      '=============  COLOR DATA  ===============
      IC(1) = 13: IC(2) = 5: IC(3) = 9: IC(4) = 1: IC(5) = 2
      IC(6) = 10: IC(7) = 14: IC(8) = 6: IC(9) = 4: IC(10) = 12
      For I = 1 To 10: ID(I) = 0: Next I
      '=============  READ DATA  ===============
      '----- Coordinates
      Line Input #1, D$
      For I = 1 To NN
         Input #1, n
         For J = 1 To NDIM:Input #1, X(n, J): Next J
      Next I
      '----- Connectivity
      Line Input #1, D$
      For I = 1 To NE
         Input #1, n: For J = 1 To NEN
      Input #1, NOC(n, J): Next J: Input #1, NTMP
      For J = 1 To 2: Input #1, C: Next J: Next I
      Close #1
      Open File2 For Input As #2
      '----- Nodal Values
      Line Input #2, D$
      For I = 1 To NN
         Input #2, FF(I)
      Next I
      Close #2
End Sub
```

```
'=====    FIND BOUNDARY LINES    =====
Private Sub FindBoundary()
'=============  Find Boundary Lines  ===============
      'Edges defined by nodes in NOC to nodes in NCON
      For IE = 1 To NE
        For I = 1 To NEN
          I1 = I + 1: If I1 > NEN Then I1 = 1
          NCON(IE, I) = NOC(IE, I1)
        Next I
      Next IE
      For IE = 1 To NE
```

```
        For I = 1 To NEN
          I1 = NCON(IE, I): I2 = NOC(IE, I)
          INDX = 0
          For JE = IE + 1 To NE
             For J = 1 To NEN
                If NCON(JE, J) <> 0 Then
                    If I1 = NCON(JE, J) Or I1 = NOC(JE, J) Then
                      If I2 = NCON(JE, J) Or I2 = NOC(JE, J) Then
                        NCON(JE, J) = 0: INDX = INDX + 1
                      End If
                    End If
                End If
              Next J
          Next JE
          If INDX > 0 Then NCON(IE, I) = 0
        Next I
      Next IE
End Sub
'=====================================================
```

```
'========    DRAW BOUNARY    ========
Private Sub DrawBoundary()
      picBox.Scale (XL, YH)-(XH, YL)
      picBox.Cls
      '============  Draw Boundary  ==============
      For IE = 1 To NE
        For I = 1 To NEN
          If NCON(IE, I) > 0 Then
              I1 = NCON(IE, I): I2 = NOC(IE, I)
              picBox.Line (X(I1, 1), X(I1, 2))-(X(I2, 1), X(I2, 2))
          End If
        Next I
      Next IE
End Sub
'========    DRAW CONTOUR LINES    ========
Private Sub DrawContours()
      '===========  Contour Plotting  ===========
      For IE = 1 To NE
         If NEN = 3 Then
            For IEN = 1 To NEN
               IEE = NOC(IE, IEN)
               U(IEN) = FF(IEE)
               XX(IEN) = X(IEE, 1)
               YY(IEN) = X(IEE, 2)
            Next IEN
            Call ElementPlot
         ElseIf NEN = 4 Then
            XB = 0: YB = 0: UB = 0
            For IT = 1 To NEN
               NIT = NOC(IE, IT)
               XB = XB + 0.25 * X(NIT, 1)
               YB = YB + 0.25 * X(NIT, 2)
               UB = UB + 0.25 * FF(NIT)
            Next IT
```

continued

```
          For IT = 1 To NEN
             IT1 = IT + 1: If IT1 > 4 Then IT1 = 1
             XX(1) = XB: YY(1) = YB: U(1) = UB
             NIE = NOC(IE, IT)
             XX(2) = X(NIE, 1): YY(2) = X(NIE, 2): U(2) = FF(NIE)
             NIE = NOC(IE, IT1)
             XX(3) = X(NIE, 1): YY(3) = X(NIE, 2): U(3) = FF(NIE)
             Call ElementPlot
        Next IT
      Else
        Print "NUMBER OF ELEMENT NODES > 4 IS NOT SUPPORTED"
        End
      End If
      Next IE
   For I = 1 To 10: ID(I) = 0: Next I
End SubPrivate Sub ElementPlot()
'THREE POINTS IN ASCENDING ORDER
   For I = 1 To 2
      C = U(I): II = I
      For J = I + 1 To 3
         If C > U(J) Then
            C = U(J): II = J
         End If
      Next J
      U(II) = U(I): U(I) = C
      C1 = XX(II): XX(II) = XX(I): XX(I) = C1
      C1 = YY(II): YY(II) = YY(I): YY(I) = C1
   Next I
   SU = (U(1) - FMIN) / STP
   II = Int(SU)
   If II <= SU Then II = II + 1
   UT = FMIN + II * STP
   Do While UT <= U(3)
      ICO = IC(II)
      X1 = ((U(3) - UT) * XX(1) + (UT - U(1)) * XX(3)) / (U(3) - U(1))
      Y1 = ((U(3) - UT) * YY(1) + (UT - U(1)) * YY(3)) / (U(3) - U(1))
      L = 1: If UT > U(2) Then L = 3
      X2 = ((U(L) - UT) * XX(2) + (UT - U(2)) * XX(L)) / (U(L) - U(2))
      Y2 = ((U(L) - UT) * YY(2) + (UT - U(2)) * YY(L)) / (U(L) - U(2))
      picBox.Line (X1, Y1)-(X2, Y2), QBColor(ICO)
      If ID(II) = 0 Then
         picBox.CurrentX = X1: picBox.CurrentY = Y1
         If (XL < X1 And X1 < XH) And (YL < Y1 And Y1 < YH) Then
            picBox.Print II
            ID(II) = 1
         End If
      End If
      UT = UT + STP: II = II + 1
   Loop
End Sub
```

A P P E N D I X

Proof of $dA = \det \mathbf{J}\, d\xi\, d\eta$

Consider a mapping of variables from x, y to u_1, u_2, given as

$$x = x(u_1, u_2) \quad y = y(u_1, u_2) \tag{A1.1}$$

We assume that these equations can be reversed to express u_1, u_2, in terms of x, y and that the correspondence is unique.

If a particle moves from a point P in such a way that u_2 is held constant and only u_1 varies, then a curve in the plane is generated. We call this the u_1 curve (Fig. A1.1). Similarly, the u_2 curve is generated by keeping u_1 constant and letting u_2 vary. Let

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} \tag{A1.2}$$

represent the vector of a point P where $\mathbf{i}$ and $\mathbf{j}$ are unit vectors along x and y, respectively. Consider the vectors

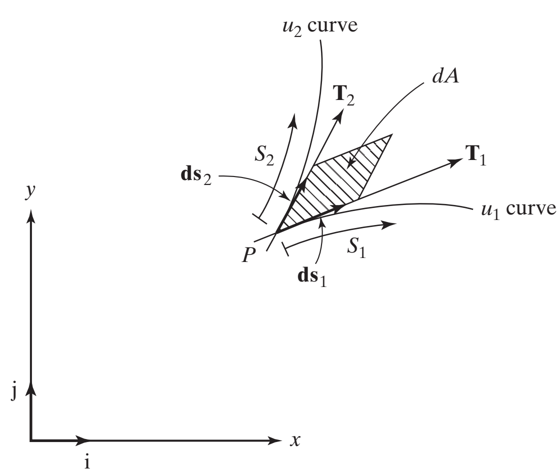


FIGURE A1.1

$$\mathbf{T}_1 = \frac{\partial \mathbf{r}}{\partial u_1} \qquad \mathbf{T}_2 = \frac{\partial \mathbf{r}}{\partial u_2} \tag{A1.3}$$

or, in view of Eq. A1.2,

$$\mathbf{T}_1 = \frac{\partial x}{\partial u_1}\mathbf{i} + \frac{\partial y}{\partial u_1}\mathbf{j} \qquad \mathbf{T}_2 = \frac{\partial x}{\partial u_2}\mathbf{i} + \frac{\partial y}{\partial u_2}\mathbf{j} \tag{A1.4}$$

We can show that $\mathbf{T}_1$ is a vector tangent to the u_1 curve and $\mathbf{T}_2$ is tangent to the u_2 curve (Fig. A1.1). To see this, we use the definition

$$\frac{\partial \mathbf{r}}{\partial u_1} = \lim_{\Delta u_1 \to 0} \frac{\Delta \mathbf{r}}{\Delta u_1} \tag{A1.5}$$

where $\Delta \mathbf{r} = \mathbf{r}(u_1 + \Delta u_1) - \mathbf{r}(u_1)$. In the limit, the chord $\Delta \mathbf{r}$ becomes the tangent to the u_1 curve (Fig. A1.2). However, $\partial \mathbf{r}/\partial u_1$ (or $\partial \mathbf{r}/\partial u_2$) is not a unit vector. To determine its magnitude (length), we write

$$\frac{\partial \mathbf{r}}{\partial u_1} = \frac{\partial \mathbf{r}}{\partial s_1}\frac{ds_1}{du_1} \tag{A1.6}$$

where s_1 is the arc length along the u_1 curve and ds_1 is the differential arc length. The magnitude of the vector

$$\frac{\partial \mathbf{r}}{\partial s_1} = \lim_{\Delta s_1 \to 0} \frac{\Delta \mathbf{r}}{\Delta s_1}$$

is the limiting ratio of the chord length to the arc length, which equals unity. Thus, we conclude that the magnitude of the vector $\partial \mathbf{r}/\partial u_1$ is ds_1/du_1. We have

$$\mathbf{T}_1 = \left(\frac{ds_1}{du_1}\right)\mathbf{t}_1 \tag{A1.7}$$

$$\mathbf{T}_2 = \left(\frac{ds_2}{du_2}\right)\mathbf{t}_2$$

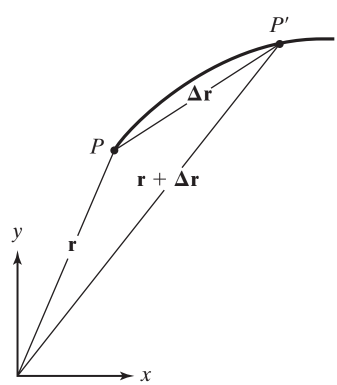


FIGURE A1.2

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are unit vectors tangent to the u_1 and u_2 curves, respectively. Using Eq. A1.7, we have the following representation of the vectors $\mathbf{ds}_1$ and $\mathbf{ds}_2$ whose lengths are ds_1 and ds_2 (Fig. A1.1):

$$\begin{aligned} \mathbf{ds}_1 &= \mathbf{t}_1\, ds_1 = \mathbf{T}_1\, du_1 \\ \mathbf{ds}_2 &= \mathbf{t}_2\, ds_2 = \mathbf{T}_2\, du_2 \end{aligned} \tag{A1.8}$$

The differential area $\mathbf{dA}$ is a vector with magnitude dA and direction normal to the element area, which in this case is $\mathbf{k}$. The vector $\mathbf{dA}$ in view of Eqs. A1.4 and A1.8 is given by the determinant rule

$$\begin{aligned} \mathbf{dA} &= \mathbf{ds}_1 \times \mathbf{ds}_2 \\ &= \mathbf{T}_1 \times \mathbf{T}_2\, du_1\, du_2 \\ &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dfrac{\partial x}{\partial u_1} & \dfrac{\partial y}{\partial u_1} & 0 \\ \dfrac{\partial x}{\partial u_2} & \dfrac{\partial y}{\partial u_2} & 0 \end{vmatrix} du_1\, du_2 \\ &= \left(\frac{\partial x}{\partial u_1}\frac{\partial y}{\partial u_2} - \frac{\partial x}{\partial u_2}\frac{\partial y}{\partial u_1} \right) du_1\, du_2\, \mathbf{k} \end{aligned} \tag{A1.9}$$

We denote the Jacobian matrix as

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial u_1} & \dfrac{\partial y}{\partial u_1} \\ \dfrac{\partial x}{\partial u_2} & \dfrac{\partial y}{\partial u_2} \end{bmatrix} \tag{A1.10}$$

The magnitude dA can now be written as

$$dA = \det \mathbf{J}\, du_1\, du_2 \tag{A1.11}$$

which is the desired result. Note that if we work with ξ- and η-coordinates instead of u_1- and u_2-coordinates, as in the text, then

$$dA = \det \mathbf{J}\, d\xi\, d\eta$$

This relation generalizes to three dimensions as

$$dV = \det \mathbf{J}\, d\xi\, d\eta\, d\zeta$$

where the Jacobian determinant $\det \mathbf{J}$ expresses the ratio of the volume element $dx\, dy\, dz$ to $d\xi\, d\eta\, d\zeta$.

Bibliography

There are many excellent published books and articles in various journals on the subject of the finite element method and its applications. A list of books published in the English language is given below. The list includes books dealing with both fundamental and advanced topics. Several of these have been used in the present text without explicit citation. On completing the material from various chapters from this book, the students and users should benefit from referring to other books and articles.

AINSWORTH, M., and J. T. ODEN, *A Posteriori Error Estimation in Finite Element Analysis*. Hoboken, NJ: Wiley, 2000.

AKIN, J. E., *Finite Elements for Analysis and Design*. San Diego, CA: Academic Press, 1994.

AKIN, J. E., *Finite Element Analysis with Error Estimators*. Burlington, MA: Elsevier, 2005.

ALLAIRE, P. E., *Basics of the Finite Element Method—Solid Mechanics, Heat Transfer, and Fluid Mechanics*. Dubuque, IA: W. C. Brown, 1985.

ASKENAZI, A., and V. ADAMS, *Building Better Products with Finite Element Analysis*. On World Press, 1998.

AXELSSON, O., and V. A. BARKER, *Finite Element Solution of Boundary Value Problems*. Orlando, FL: Academic, 1984.

BAKER, A. J., *Finite Element Computational Fluid Mechanics*. New York: McGraw-Hill, 1983.

BAKER, A. J., and D. W. PEPPER, *Finite Elements 1–2–3*. New York: McGraw-Hill, 1991.

BARAN, N. M., *Finite Element Analysis on Microcomputers*. New York: McGraw-Hill, 1991.

BATHE, K. J., *Finite Element Procedures*. Upper Saddle River, NJ: Prentice Hall, 1996.

BATHE, K. J., and E. L. WILSON, *Numerical Methods in Finite Element Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1976.

BECKER, A. A., *Introductory Guide to Finite Element Analysis*. ASME Press, 2003.

BECKER, E. B., G. F. CAREY, and J. T. ODEN, *Finite Elements—An Introduction*, Vol. 1. Englewood Cliffs, NJ: Prentice Hall, 1981.

BEYTSCHKO, T., B. MORAN, and W. K. LIU, *Nonlinear Finite Elements for Continua and Structures*. New York: Wiley, 2000.

BHATTI, M. A., *Fundamental Finite Element Analysis and Applications*. Hoboken, NJ: Wiley, 2005.

BICKFORD, W. M., *A First Course in the Finite Element Method*. Homewood, IL: Richard D. Irwin, 1990.

BONET, J., and R. D. WOOD, *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997.

BOWES, W. H., and L. T. RUSSEL, *Stress Analysis by the Finite Element Method for Practicing Engineers*. Lexington, MA: Lexington Books, 1975.

Brebbia, C. A., and J. J. Connor, *Fundamentals of Finite Element Techniques for Structural Engineers*. London: Butterworths, 1975.

Buchanan, G. R., *Finite Element Analysis*. New York: McGraw-Hill, 1994.

Burnett, D. S., *Finite Element Analysis from Concepts to Applications*. Reading, MA: Addison-Wesley, 1987.

Carey, G. F., and J. T. Oden, *Finite Elements—A Second Course*, Vol. 2. Englewood Cliffs, NJ: Prentice Hall, 1983.

Carey, G. F., and J. T. Oden, *Finite Elements—Computational Aspects*, Vol. 3. Englewood Cliffs, NJ: Prentice Hall, 1984.

Carroll, W. F., *A Primer for Finite Elements in Elastic Structures*. Wiley, 1999.

Chandrupatla, T. R., *Finite Element Analysis for Engineering and Technology*. Hyderabad, India: Universities Press, 2004.

Chari, M. V. K., and P. P. Silvester, *Finite Elements in Electrical and Magnetic Field Problems*. New York: Wiley, 1981.

Cheung, Y. K., and M. F. Yeo, *A Practical Introduction to Finite Element Analysis*. London: Pitman, 1979.

Chung, T. J., *Finite Element Analysis in Fluid Dynamics*. New York: McGraw-Hill, 1978.

Ciarlet, P. G., *The Finite Element Method for Elliptic Problems*. Amsterdam: North-Holland, 1978.

Connor, J. C., and C. A. Brebbia, *Finite Element Techniques for Fluid Flow*. London: Butterworths, 1976.

Cook, R. D., *Finite Element Modeling for Stress Analysis*. New York: Wiley, 1995.

Cook, R. D.,Cook , D.S. Malkus, M. E. Piesha, and R. J., Witt, *Concepts and Applications of Finite Element Analysis*, 4th Ed. Hoboken, NJ: Wiley, 2002.

Davies, A. J., *The Finite Element Method: A First Approach*. Oxford: Clarendon, 1980.

Desai, C. S., *Elementary Finite Element Method*. Englewood Cliffs, NJ: Prentice Hall, 1979.

Desai, C. S., and J. F. Abel, *Introduction to the Finite Element Method*. New York: Van Nostrand Reinhold, 1972.

Desai, C. S., and T. Kundu, *Introductory Finite Element Method*. CRC Press, 2001.

Fagan, M. J. J., *Finite Element Analysis: Theory and Practice*. Addison Wesley Longman, 1996.

Fairweather, G., *Finite Element Galerkin Methods for Differential Equations*. New York: Dekker, 1978.

Fenner, R. T., *Finite Element Methods for Engineers*. River Edge, NJ: World Scientific, 1996.

Ferreira, A. J. M., *MATLAB Codes for Finite Element Analysis*. Springer, 2009.

Fish, J., and T. Belytscho, *A First Course in Finite Elements*. Chichester, UK: Wiley, 2007.

Gallagher, R. H., *Finite Element Analysis—Fundamentals*. Englewood Cliffs, NJ: Prentice Hall, 1975.

Gockenback, M. S., *Understanding and Implementing the Finite Element Method*. SIAM, 2006.

Grandin, H., Jr., *Fundamentals of the Finite Element Method*. New York: Macmillan, 1986.

Heinrich, J. C., and D. W. Pepper, *Intermediate Finite Element Method: Fluid Flow and Heat Transfer Applications*. Taylor & Francis, 1997.

Hinton, E., and D. R. J. Owen, *Finite Element Programming*. London: Academic, 1977.

Hinton, E., and D. R. J. Owen, *An Introduction to Finite Element Computations*. Swansea, Great Britain: Pineridge Press, 1979.

Huebner, K. H., and E. A. Thornton, *The Finite Element Method for Engineers*, 2nd Ed. New York: Wiley-Interscience, 1982.

Hughes, T. J. R., *The Finite Element Method—Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.

Hutton, D., *Fundamentals of Finite Element Analysis*. New York: McGraw-Hill, 2004.

Irons, B., and S. Ahmad, *Techniques of Finite Elements*. New York: Wiley, 1980.

IRONS, B., and N. SHRIVE, *Finite Element Primer*. New York: Wiley, 1983.

JIN, J., *The Finite Element Method in Electromagnetics*. New York: Wiley, 1993.

KATTAN, P., *MATLAB Guide to Finite Elements*, 2nd Ed. Springer, 2008.

KIM, N-H., and B.V. SANKAR, *Introduction to Finite Element Analysis and Design*. Hoboken, NJ: Wiley, 2008.

KIKUCHI, N., *Finite Element Methods in Mechanics*. Cambridge, Great Britain: Cambridge University Press, 1986.

KUROWSKI, P. M., *Finite Element Analysis for Design Engineers*. SAE International, 2004.

KNIGHT, C. E., *The Finite Element Method in Machine Design*. Boston: PWS Kent, 1993.

KRISHNAMOORTY, C. S., *Finite Element Analysis—Theory and Programming*. New Delhi: Tata McGraw-Hill, 1987.

LEPI, S. M., *Practical Guide to Finite Elements: A Solid Mechanics Approach*. Marcel Dekker, 1998.

LIVESLEY, R. K., *Finite Elements: An Introduction for Engineers*. Cambridge, Great Britain: Cambridge University Press, 1983.

LOGAN, D. L., *A First Course in the Finite Element Method,* 5th Ed. Samford, CT: Cengage Learning, 2011.

MACDONALD, B. J., *Practical Stress Analysis with Finite Elements*. Dublin: Glasnevin Publishing, 2007.

MARTIN, H. C., and G. F. CAREY, *Introduction to Finite Element Analysis: Theory and Application*. New York: McGraw-Hill, 1972.

MELOSH, R. J., *Structural Engineering Analysis by Finite Elements*. Englewood Cliffs, NJ: Prentice Hall, 1990.

MITCHELL, A. R., and R. WAIT, *The Finite Element Method in Partial Differential Equations*. New York: Wiley, 1977.

MOAVENI, S., *Finite Element Analysis: Theory and Applications with Ansys,* 3rd Ed. Upper Saddle River: Prentice Hall, 2007.

MORRIS, A., *A Practical Guide to Finite Element Modelling.* Chichester, UK: Wiley, 2008.

NAKAZAWA, S., and D. W. KELLY, *Mathematics of Finite Elements—An Engineering Approach*. Swansea, Great Britain: Pineridge Press, 1983.

NATH, B., *Fundamentals of Finite Elements for Engineers*. London: Athlone, 1974.

NICHOLSON, D. W., *Finite Element Analysis Thermomechanics of Solids*. Boca Raton, FL: CRC Press, 2005.

NIKISHKOV, G., *Programming Finite Elements in Java*. Springer, 2010.

NORRIE, D. H., and G. DE VRIES, *The Finite Element Method: Fundamentals and Applications*. New York: Academic, 1973.

NORRIE, D. H., and G. DE VRIES, *An Introduction to Finite Element Analysis*. New York: Academic, 1978.

ODEN, J. T., *Finite Elements of Nonlinear Continua*. New York: McGraw-Hill, 1972.

ODEN, J. T., and G. F. CAREY, *Finite Elements: Mathematical Aspects*, Vol. 4. Englewood Cliffs, NJ: Prentice Hall, 1982.

ODEN, J. T., and J. N. REDDY, *An Introduction to the Mathematical Theory of Finite Elements*. New York: Wiley, 1976.

OWEN, D. R. J., and E. HINTON, *A Simple Guide to Finite Elements*. Swansea, Great Britain: Pineridge Press, 1980.

PAO, Y. C., *A First Course in Finite Element Analysis*. Newton, MA: Allyn & Bacon, 1986.

PINDER, G. F., and W. G. GRAY, *Finite Element Simulation in Surface and Subsurface Hydrology*. New York: Academic, 1977.

POTTS, J. F., and J. W. OLER, *Finite Element Applications with Microcomputers*. Englewood Cliffs, NJ: Prentice Hall, 1989.

PRZEMIENIECKI, J. S., *Theory of Matrix Structural Analysis*. New York: McGraw-Hill, 1968.

RAO, S. S., *The Finite Element Method in Engineering*, 5th Ed. Oxford, UK: Elsevier, 2011.

REDDY, J. N., *Energy and Variational Methods in Applied Mechanics,* 2nd Ed. Hoboken, NJ: Wiley, 2002.

REDDY, J. N., *An Introduction to the Finite Element Method,* 3rd Ed, New York: McGraw-Hill, 2005.

REDDY, J. N., and D. K. GARTLING, *The Finite Element Method in Heat Transfer and Fluid Dynamics*, 3rd Ed. CRC Press, 2010.

ROBINSON, J., *An Integrated Theory of Finite Element Methods*. New York: Wiley-Interscience, 1973.

ROBINSON, J., *Understanding Finite Element Stress Analysis*. Wimborne, Great Britain: Robinson and Associates, 1981.

ROCKEY, K. C., H. R. EVANS, D. W. GRIFFITHS, and D. A. NETHERCOT, *The Finite Element Method—A Basic Introduction*, 2nd Ed. New York: Halsted (Wiley), 1980.

ROSS, C. T. F., *Finite Element Programs for Axisymmetric Problems in Engineering*. Chichester, Great Britain: Ellis Horwood, 1984.

SEGERLIND, L. J., *Applied Finite Element Analysis*, 2nd Ed. New York: Wiley, 1984.

SHAMES, I. H., and C. L. DYM, *Energy and Finite Element Methods in Structural Mechanics*. New York: McGraw-Hill, 1985.

SILVESTER, P. P., and R. L. FERRARI, *Finite Elements for Electrical Engineers*. Cambridge, Great Britain: Cambridge University Press, 1996.

SMITH, I. M., and D. V. GRIFFITHS, *Programming the Finite Element Method,* 4th Ed. Chichester, UK: Wiley, 2004.

STASA, F. L., *Applied Finite Element Analysis for Engineers*. New York: Holt, Rinehart & Winston, 1985.

STRANG, G., and G. FIX, *An Analysis of the Finite Element Method*. Englewood Cliffs, NJ: Prentice Hall, 1973.

SZABÓ, B., and I. BABUŠKA, *Introduction to Finite Element Analysis, Verification and Validation*. Hoboken, NJ: Wiley, 2011.

TONG, P., and J. N. ROSSETOS, *Finite Element Method—Basic Techniques and Implementation*. Cambridge, MA: MIT Press, 1977.

URAL, O., *Finite Element Method: Basic Concepts and Applications*. New York: Intext Educational Publishers, 1973.

VOLAKIS, J. L., A. CHATTERJEE, and L. C. EMPEL, *Finite Element Method for Electromagnetics.* IEEE, 1998.

WACHSPRESS, E. L., *A Rational Finite Element Basis*. New York: Academic, 1975.

WAIT, R., and A. R. MITCHELL, *Finite Element Analysis and Applications*. New York: Wiley, 1985.

WHITE, R. E., *An Introduction to the Finite Element Method with Applications to Non-Linear Problems*. New York: Wiley-Interscience, 1985.

WILLIAMS, M. M. R., *Finite Element Methods in Radiation Physics*. Elmsford, NY: Pergamon, 1982.

WRIGGERS, P., *Nonlinear Finite Element Methods*. Springer, 2010.

YANG, T. Y., *Finite Element Structural Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1986.

ZAHAVI, E., *The Finite Element Method in Machine Design*. Englewood Cliffs, NJ: Prentice Hall, 1992.

ZIENKIEWICZ, O. C., and K. MORGAN, *Finite Elements and Approximation*. Dover Publications, 2006.

ZIENKIEWICZ, O. C., and R. L. TAYLOR, *The Finite Element Method*, 6th Ed. Burlington, MA: Butterworth-Heinemann, 2005.

Answers to Selected Problems

(1.3) 4500 psi.

(1.8) $\sigma_n = 50\,\text{MPa}$.

(1.12) $q_1 = 1.363$ mm and $q_2 = 1.963$ mm.

(1.15) $u_{x=1} = 4.90 \times 10^{-11}$

(2.1c) $\lambda_1 = 0.2325$, $\lambda_2 = 5.665$, and $\lambda_3 = 9.103$.

Matrix is positive definite.

$\mathbf{y}_1 = [0.172, 0.668, 0.724]^{\mathrm{T}}$,

$\mathbf{y}_2 = [0.495, 0.577, -0.65]^{\mathrm{T}}$, and

$\mathbf{y}_3 = [0.85, -0.47, 0.232]^{\mathrm{T}}$.

(2.2b) $\int_{-1}^{1} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,d\xi = \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & \frac{16}{15} \end{bmatrix}$.

(2.3) $\mathbf{Q} = \begin{bmatrix} 6 & 2 \\ 3 & -5 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 3 \\ -8 \end{bmatrix}$.

(2.8) $A_{11,14} \rightarrow B_{11,4}$ and $B_{6,1} \rightarrow A_{6,6}$.

(3.1) **(a)** $u = 0.05625$ in. **(b)** $\epsilon = 1.25 \times 10^{-3}$.

(d) $U_e = 143.7$ lb-in.

(3.9) $Q_2 = 0.436$ mm and $Q_3 = 0.211$ mm.

(3.12) Stress in element 1 = 2480.371 MPa.

(3.30) $\mathbf{T}^e = \frac{\ell_e}{30}[4T_1 - T_2 + 2T_3, -T_1 + 4T_2 + 2T_3, 2T_1 + 2T_2 + 16T_3]^{\mathrm{T}}$.

(4.1) $\ell = 0.8,\ m = 0.6,\ \mathbf{q}' = \begin{Bmatrix} 2.02 \\ 5 \end{Bmatrix} \times 10^{-2}$ in.

$\sigma = 17{,}880$ psi, and $U_e = 559.46$ lb.-in.

(4.3) $K_{1,1} = 4.42 \times 10^5$.

(4.4) $Q_3 = 1.3706 \times 10^{-3}$ in.

(4.6) Stress in element $1-3 = -110.8$ MPa.

(4.9) Point R moves horizontally by 3.39 mm.

(5.1) Deflection under the load point $= -0.05461$ mm.

(5.2) Deflection under load $= -0.0113$ in.

(5.8) Deflection at midpoint of $BC = -0.417$ in.

(5.12) Vertical deflection of point D without tie rod $= -19.338$ in. and with tie rod $= -1.4552$ in.

(6.1) $Q_1 = 0.000205$ mm (x-disp.) and $Q_2 = -0.00117$ mm (y-disp.)

(6.2) Area $= 19$.

(6.5) $\epsilon_x = [0.0005 \quad 0.00267 \quad -0.00583]^{\mathrm{T}}$

(6.11) x-disp. $= 0.000205$ mm.

(7.1) Use $\epsilon_\theta = 2.02 \times 10^6$ psi.

(7.4) Outer diameter after deformation $= 100.00117$ mm.

(7.5) Contact pressure is averaged to be -7022.15 psi.

(7.7) Peak radial stress $\approx 10{,}000$ psi and peak hoop stress $\approx 54{,}000$ psi.

(7.14) Hoop stress reduces from about 990 MPa without a shrink ring to 650 MPa with a ring.

(8.1) $x = 7$ and $y = 7$.

(8.2) Value of integral $= 368$.

(9.7) Max. vertical disp. $= -0.0148$ in., based on a four-element hexahedral mesh.

(10.1) $[T_1, T_2, T_3] = [40, 16.114, -7.7708]$°C. (More elements will give better answer.)

(10.3) Peak temperature $= 50.7229$°C.

(10.13) Heat flow out of chimney $= 1{,}883.8$ W/m.

(10.18) $\alpha = 5.263 \times 10^{-6} T/G$ rad/mm, where T is in N-mm and G is in MPa.

(10.21) Velocity at waist *a–a* varies from 345 cm/s to 281 cm/s.

(10.24) $C = 13.5$.

(11.1) Lowest natural frequency $= 1509$ Hz (cps).

(11.3) Lumped mass results are $\lambda_1 = 8E + 07$ and $\lambda_2 = 3.5E + 08$.

(11.7) Lowest resonant frequency $= 1392.02$ Hz.

Index

Programs Included in the CD-ROM

CH-1	CH-2	CH-3	CH-4	CH-5	CH-6
	GAUSS GSOL SKYLINE	FEM1D	TRUSS2D TRUSSKY	BEAM FRAME2D FRAME3D	CST
CH-7	**CH-8**	**CH-9**	**CH-10**	**CH-11**	**CH-12**
AXISYM	QUAD QUADCG AXIQUAD GaussLegendre GLInteg	TETRA3D HEXAFRON	HEAT1D HEAT2D TORSION	INVITR JACOBI BEAMKM CSTKM GENEIGEN	MESHGEN PLOT2D BESTFIT BESTFITQ CONTOURA CONTOURB

CD-ROM Contents

Directory	Description
\QBASIC	Programs in QBASIC
\FORTRAN	Programs in Fortran Language
\C	Programs in C
\VB 5	Programs in Visual Basic
\MATLAB	Programs in MATLAB
\EXAMPLES	Example Data Files (.inp extension)
\EXCELVB	Programs in Excel Visual Basic
\JavaScript	Programs in Javascript

Note: You may download all programs and ReadMe file from the book website www.pearsonhighered.com/chandrupatla

List of Key Symbols Used in the Text

Symbol	Description
$\mathbf{u}(x, y, z) = [u(x, y, z), v(x, y, z), w(x, y, z)]^{\mathrm{T}}$	displacements along coordinate directions at point (x, y, z)
$\mathbf{f} = [f_x, f_y, f_z]^{\mathrm{T}}$	components of body force per unit volume, at point (x, y, z)
$\mathbf{T} = [T_x, T_y, T_z]^{\mathrm{T}}$	components of traction force per unit area, at point (x, y, z) on the surface
$\boldsymbol{\epsilon} = [\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{yz}, \gamma_{zx}, \gamma_{xy}]^{\mathrm{T}}$	strain components; ϵ are normal strains and γ are engineering shear strains
$\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z, \tau_{yz}, \tau_{zx}, \tau_{xy}]^{\mathrm{T}}$	stress components; σ are normal stresses and τ are engineering shear stresses
Π	potential energy $= U + WP$, where $U =$ strain energy, $WP =$ work potential
$\mathbf{q}$	vector of displacements of the nodes (degrees of freedom or dof) of an element, dimension $(NDN*NEN, 1)$—see next Table for explanation of NDN and NEN
$\mathbf{Q}$	vector of displacements of ALL the nodes of an element, dimension $(NN*NDN, 1)$—see next Table for explanation of NN and NDN
$\mathbf{k}$	element stiffness matrix; strain energy in element, $U_e = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{kq}$
$\mathbf{K}$	global stiffness matrix for entire structure; $\Pi = \frac{1}{2}\mathbf{Q}^{\mathrm{T}}\mathbf{KQ} - \mathbf{Q}^{\mathrm{T}}\mathbf{F}$
$\mathbf{f}^e$	body force in element e distributed to the nodes of the element
$\mathbf{T}^e$	traction force in element e distributed to the nodes of the element
$\boldsymbol{\phi}(x, y, z)$	virtual displacement variable; counterpart of the real displacement $\mathbf{u}(x, y, z)$
$\boldsymbol{\psi}$	vector of virtual displacements of the nodes in an element; counterpart of $\mathbf{q}$
$\mathbf{N}$, $\mathbf{D}$, and $\mathbf{B}$	shape functions in $\xi\eta\zeta$ coordinates, material matrix, strain–displacement matrix, respectively. $\mathbf{u} = \mathbf{Nq}$, $\boldsymbol{\epsilon} = \mathbf{Bq}$ and $\boldsymbol{\sigma} = \mathbf{DBq}$

Structure of Input Files[†]

```
TITLE    (*)
PROBLEM DESCRIPTION    (*)
NN NE NM NDIM NEN NDN  (*)
 4  2  2   2   3   2            --- 1 Line of data, 6 entries per line
ND NL NMPC      (*)
 5  2  0                        --- 1 Line of data, 3 entries
Node#   Coordinate#1 ... Coordinate#NDIM       (*)
 1          3              0  }
 2          3              2  }
 3          0              2  }   ---NN Lines of data, (NDIM+1)entries
 4          0              0  }
Elem#   Node#1 ... Node#NEN    Mat#   Element  Characteristics†† (*)
 1       4     1      2         1       0.5              0. }---NE Lines of data,
 2       3     4      2         2       0.5              0. }  (NEN+2+ #of Char.)entries
DOF#    Specified Displacement (*)
 2          0  }
 5          0  }
 6          0  }  ---ND Lines of data, 2 entries
 7          0  }
 8          0  }
DOF#    Load  (*)
 4      -7500 } ---NL Lines of data, 2 entries
 3       3000 }

MAT#    Material    Properties††  (*)
 1       30e6         0.25   12e-6 }    ---NM Lines of data,(1+ # of prop.)entries
 2       20e6         0.3     0.   }
B1   i    B2     j    B3  (Multipoint constraint: B1*Qi+B2*Qj=B3)    (*)
                          }             ---NMPC Lines of data, 5 entries
```

[†]HEAT1D and HEAT2D Programs need extra boundary data about flux and convection. (See Chapter 10.)

(*) = DUMMY LINE – necessary

Note: No Blank Lines must be present in the input file

[††]See below for description of element characteristics and material properties

Main Program Variables

NN = Number of Nodes; NE = Number of Elements; NM = Number of Different Materials

NDIM = Number of Coordinates per Node (e.g., NDIM = 2 for 2-D, or = 3 for 3-D); NEN = Number of Nodes per Element (e.g., NEN = 3 for 3-noded triangular element, or = 4 for a 4-noded quadrilateral)

NDN = Number of Degrees of Freedom per Node (e.g., NDN = 2 for a CST element, or = 6 for 3-D beam element)

ND = Number of Degrees of Freedom along which Displacement is Specified (Boundary Conditions)

NL = Number of Applied Component Loads (along Degrees of Freedom)

NMPC = Number of Multipoint Constraints; NQ = Total Number of Degrees of Freedom = NN * NDN

(*continued inside back cover*)

(*continued*)

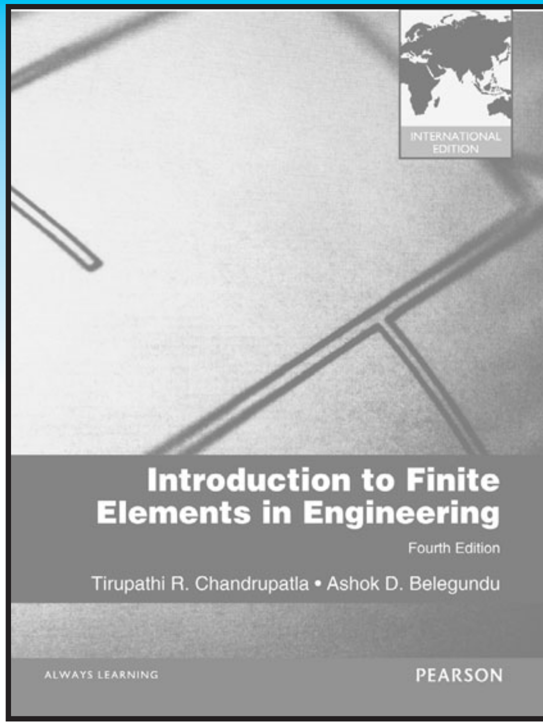
Program	**Element Characteristics**	**Material Properties**
FEM1D, TRUSS, TRUSSKY	Area, Temperature Rise	E
CST, QUAD	Thickness, Temperature Rise	$E,\ \nu,\ \alpha$
AXISYM	Temperature Rise	$E,\ \nu,\ \alpha$
FRAME2D	Area, Inertia, Distributed Load	E
FRAME3D	Area, 3-inertias, 2-Distributed Loads	E
TETRA, HEXAFNT	Temperature Rise	$E,\ \nu,\ \alpha$
HEAT2D	Element Heat Source	Thermal Conductivity, k
BEAMKM	Inertia, Area	$E,\ \rho$
CSTKM	Thickness	$E,\ \nu,\ \alpha,\ \rho$

TYPICAL PHYSICAL PROPERTIES OF SOME MATERIALS

Material	Density kg/m^3	Ultimate Strength Tension MPa	Ultimate Strength Comp. MPa	Yield strength MPa	Modulus of elasticity E GPa	Poisson's ratio	Coef. of thermal exp. $10^{-6}/°C$	Thermal conductivity W/m °C
Aluminum 2014-T6	2800	470		410	72	0.33	23	210
(alloy) 6061-T6	2800	228		131	70	0.33	23	210
Brass cold rolled	8470	540		420	105	0.35	19	105
annealed	8470	330		100	105	0.35	19	105
Bronze Manganese	8800	450		170	100	0.34	20	58
Cast Iron								
Gray	7200	170	650		95	0.25	12	45
Malleable	7200	370		250	170	0.25	12	45
Concrete								
Low strength	2400	2	20		22	0.15	11	1
Medium strength	2400	3	41		32	0.15	11	1
High strength	2400	4	62		40	0.15	11	1
Copper hard-drawn	8900	380		330	120	0.33	17	380
Glass Silicon	2400	80	400		70	0.17	8	0.8
Magnesium 8.5% Al	1800	350		250	45	0.35	26	160
Steel 0.2%C HR	7850	410		250	200	0.30	12	42
0.2%C CR	7850	550		350	200	0.30	12	42
0.6%C HR	7850	690		370	200	0.30	12	42
0.8%C HR quenched	7850	830		700	200	0.30	12	42
Stainless 302 CR	7920	860		600	194	0.30	17	18
Titanium 6% Al 4% V	4460	900		830	110	0.34	9	14

Properties vary widely depending on changes in composition, temperature, and treatment conditions.
CR = Cold rolled   HR = Hot rolled

Quantity	Units/Conversion
General	
Acceleration	1 in./s^2 = 0.0254 m/s^2
Area	1 in.^{2} = 645.16 mm^2
Density (i)	1 lbm/in.^{3} = 27 679.905 kg/m^3
(ii)	1 slug/ft^3 = 515.379 kg/m^3
Force	1 lb = 4.448 N (N = Newton)
Frequency	Hz (hertz = cycle/s)
Length	1 in. = 0.0254 m; 1 ft = 0.3048 m
Mass (i)	1 lbm = 0.45359 kg
(ii)	1 slug = 14.594 kg
Moment	1 in.-lb = 0.1130 N · m
Moment of inertia (area)	1 in.^{4} = 416 231.4 mm^4
Moment of inertia (mass) (i)	1 lbm-in.^{2} = 2.9264E-4 kg · m^2
(ii)	1 slug-in.^{2} = 0.009415 kg · m^2
Power (i)	1 in.-lb/s = 0.1130 W (watt = J/s)
(ii)	1 hp = 0.746 kW (1 hp = 550 ft-lb)
Pressure	1 psi = 6894.8 Pa (psi = pounds/in.^{2}; Pa = N/m^2)
Stiffness	1 lb/in. = 175.1 N/m
Stress (i)	1 psi = 6894.8 Pa
(ii)	1 ksi = 6.8948 MPa; 1 MPa = 145.04 psi (ksi = 1000 psi; MPa = 10^6 Pa)
Time	s (second)
Velocity	1 in./s = 0.0254 m/s
Volume	1 in.^{3} = 16.3871E-6 m^3
Work, energy	1 in.-lb = 0.1130 J (joule = N · m)
Heat Transfer	
Convection coefficient	1 Btu/h · ft^2 · °F = 5.6783 W/m^2 · °C
Heat	1 Btu = 1055.06 J (1 Btu = 778.17 ft-lb)
Heat flux	1 Btu/h · ft^2 = 3.1546 W/m^2
Specific heat	1 Btu/°F = 1899.108 J/°C
Temperature (i)	T °F = [(9/5)T + 32]°C
(ii)	T °K = T °C + 273.15 (K = kelvin)
Thermal conductivity	1 Btu/h · ft · °F = 1.7307 W/m · °C
Fluid Flow	
Absolute viscosity	1 lb · s/ft^2 = 478.803 P (poise = g/cm · s)
Kinematic viscosity	1 ft^2/s = 929.03 St (stoke = cm^2/s)
Electric and Magnetic Fields	
Capacitance	F (farad)
Charge	C (coulomb)
Electric charge density	C/m^3
Electric potential	V (volt)
Inductance	H (henry)
Permeability	H/m
Permittivity	F/m
Scalar magnetic potential	A (ampere)



Redeem access to self contained computer programs with source code at:

www.pearsoninternationaleditions.com/chandrupatla

Use a coin to scratch off the coating and reveal your student access code.
Do not use a knife or other sharp object as it may damage the code.

IMPORTANT: The access code on this page can only be used once to establish a subscription. If the access code has already been scratched off, it may no longer be valid.

Technical Support is available at www.247pearsoned.com.

PEARSON

Pearson Education Limited | Edinburgh Gate Harlow Essex CM20 2JE England