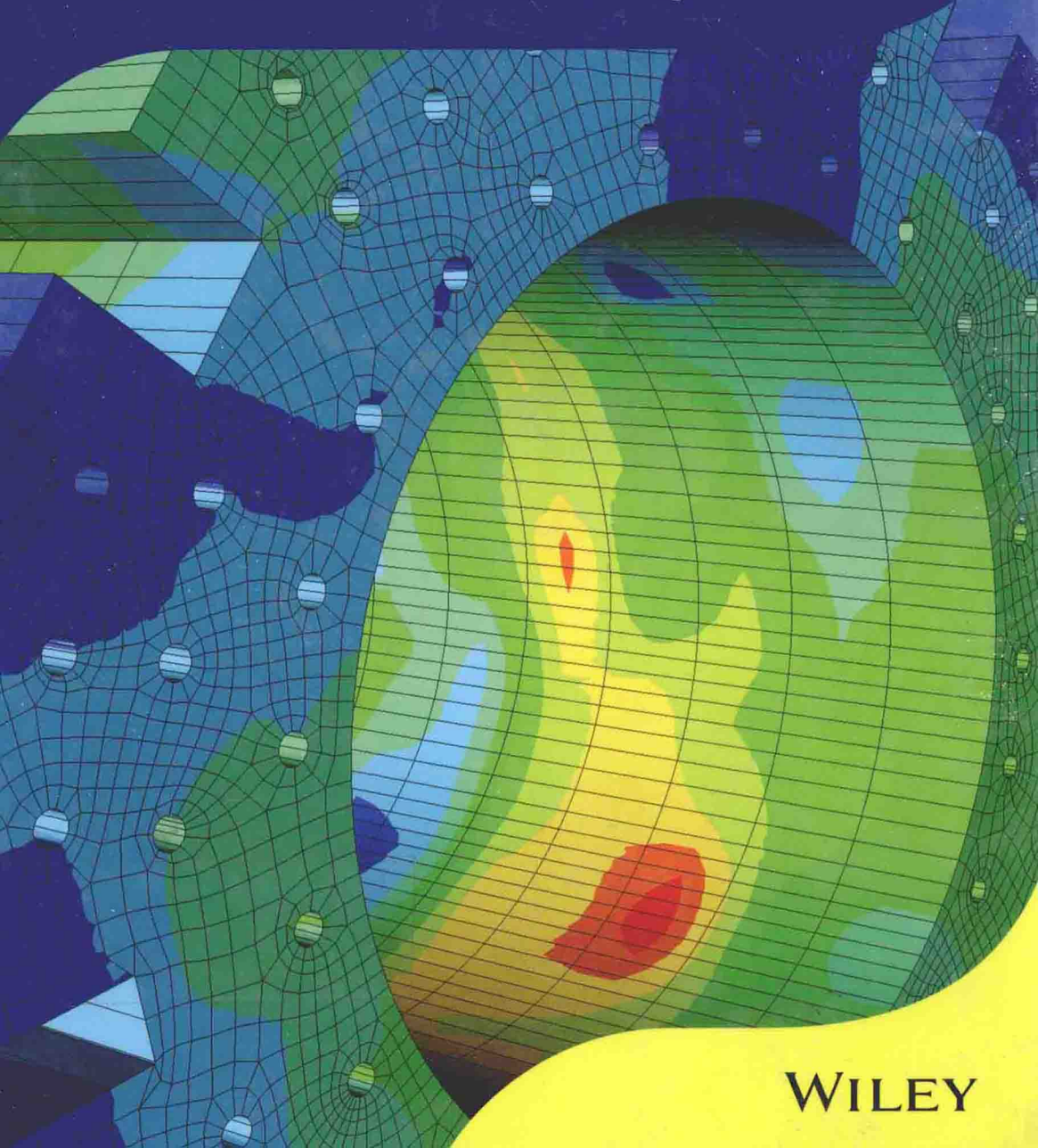


Fifth Edition

Programming the Finite Element Method

I. M. Smith, D. V. Griffiths and L. Margetts



WILEY

PROGRAMMING THE FINITE ELEMENT METHOD

Fifth Edition

I. M. Smith

University of Manchester, UK

D. V. Griffiths

Colorado School of Mines, USA

L. Margetts

University of Manchester, UK



WILEY

This edition first published 2014
© 2014 John Wiley & Sons Ltd

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. It is sold on the understanding that the publisher is not engaged in rendering professional services and neither the publisher nor the author shall be liable for damages arising herefrom. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Library of Congress Cataloguing-in-Publication Data

Smith, I. M. (Ian Moffat), 1940- author.

Programming the finite element method. – Fifth edition/Ian M. Smith, D. Vaughan Griffiths, Lee Margetts.
pages cm

Includes bibliographical references and index.

ISBN 978-1-119-97334-8 (hardback)

1. Finite element method—Data processing. 2. Engineering—Data processing. 3. FORTRAN 2003 (Computer program language) I. Griffiths, D. V., author. II. Margetts, Lee, author. III. Title.

TA347.F5S64 2014

620.001'51825—dc23

2013019445

A catalogue record for this book is available from the British Library.

ISBN: 978-1-119-97334-8

Set in 10/12pt Times by Laserwords Private Limited, Chennai, India.

Printed and bound in Malaysia by Vivar Printing Sdn Bhd

1 2014

PROGRAMMING THE FINITE ELEMENT METHOD

Preface to Fifth Edition

This edition maintains the successful theme of previous editions, namely a modular programming style which leads to concise, easy to read computer programs for the solution of a wide range of problems in engineering and science governed by partial differential equations.

The programming style has remained essentially the same despite huge advances in computer hardware. Readers will include beginners, making acquaintance with the finite element method for the first time, and specialists solving very large problems using the latest generation of parallel supercomputers.

In this edition special attention is paid to interfacing with other open access software, for example ParaView for results visualisation, ABAQUS user subroutines for a range of material constitutive models, ARPACK for large eigenvalue analyses, and METIS for mesh partitioning.

Chapter 1 has been extensively rewritten to take account of rapid developments in computer hardware, for example the availability of GPUs and cloud computing environments. In Chapters 2 to 11 numerous additions have been made to enhance analytical options, for example new return algorithms for elastoplastic analyses, more general boundary condition specification and a complex response option for dynamic analyses.

Chapter 12 has been updated to illustrate the rapidly advancing possibilities for finite element analyses in parallel computing environments. In the fourth edition the maximum number of parallel ‘processes’ used was 64 whereas in this edition the number has increased to 64,000. The use of GPUs to accelerate computations is illustrated.

Acknowledgements

The authors wish to acknowledge the contributions of a number of individuals and organizations. The support of the Australian Research Council Centre of Excellence for Geotechnical Science and Engineering (CGSE) at the University of Newcastle NSW is recognised, and particularly Jinsong Huang, who contributed to the development and validation of several of the new and modified programs in Chapters 6, 8 and 9. Louise Lever (University of Manchester), one of the principal ParaFEM developers, provided expertise in the use of ParaView for Chapters 1, 5, 6 and 12 and set up the community building website <http://parafem.org.uk>.

There were many contributions to Chapter 12. Llion Evans, Paul Mummery, Philip Manning, Graham Hall and Dimitris Christias (all University of Manchester) provided scientific case studies. Florent Lebeau and Francois Bodin (CAPS Entreprise) evaluated the use of GPUs and Philippe Young (Simpleware Ltd) provided support in image-based modelling.

Benchmarking of the programs in Chapter 12 was carried out using supercomputers hosted by the UK National High Performance Computing Service “HECToR” (e107, e254) and the UK Regional Service “N8 HPC” (EP/K000225/1). The EU FP7 project “Venus-C” and Barcelona Supercomputing Center (Spain) provided access, resources and training to use Microsoft Azure.

The authors would also like to thank family members for their support during preparation of the book, including Valerie Griffiths, Laura Sanchez and Nathan Margetts.

Contents

Preface to Fifth Edition

xv

Acknowledgements

xvii

1	Preliminaries: Computer Strategies	1
1.1	Introduction	1
1.2	Hardware	2
1.3	Memory Management	2
1.4	Vector Processors	3
1.5	Multi-core Processors	3
1.6	Co-processors	4
1.7	Parallel Processors	4
1.8	Applications Software	5
1.8.1	<i>Compilers</i>	5
1.8.2	<i>Arithmetic</i>	6
1.8.3	<i>Conditions</i>	7
1.8.4	<i>Loops</i>	8
1.9	Array Features	9
1.9.1	<i>Dynamic Arrays</i>	9
1.9.2	<i>Broadcasting</i>	9
1.9.3	<i>Constructors</i>	9
1.9.4	<i>Vector Subscripts</i>	10
1.9.5	<i>Array Sections</i>	11
1.9.6	<i>Whole-array Manipulations</i>	11
1.9.7	<i>Intrinsic Procedures for Arrays</i>	11
1.9.8	<i>Modules</i>	12
1.9.9	<i>Subprogram Libraries</i>	13
1.9.10	<i>Structured Programming</i>	15
1.10	Third-party Libraries	17
1.10.1	<i>BLAS Libraries</i>	17
1.10.2	<i>Maths Libraries</i>	17
1.10.3	<i>User Subroutines</i>	18
1.10.4	<i>MPI Libraries</i>	18
1.11	Visualisation	18
1.11.1	<i>Starting ParaView</i>	19
1.11.2	<i>Display Restrained Nodes</i>	20

1.11.3	<i>Display Applied Loads</i>	21
1.11.4	<i>Display Deformed Mesh</i>	21
1.12	Conclusions	23
	References	24
2	Spatial Discretisation by Finite Elements	25
2.1	Introduction	25
2.2	Rod Element	25
2.2.1	<i>Rod Stiffness Matrix</i>	25
2.2.2	<i>Rod Mass Element</i>	28
2.3	The Eigenvalue Equation	28
2.4	Beam Element	29
2.4.1	<i>Beam Element Stiffness Matrix</i>	29
2.4.2	<i>Beam Element Mass Matrix</i>	31
2.5	Beam with an Axial Force	31
2.6	Beam on an Elastic Foundation	32
2.7	General Remarks on the Discretisation Process	33
2.8	Alternative Derivation of Element Stiffness	33
2.9	Two-dimensional Elements: Plane Stress	35
2.10	Energy Approach and Plane Strain	38
2.10.1	<i>Thermoelasticity</i>	39
2.11	Plane Element Mass Matrix	40
2.12	Axisymmetric Stress and Strain	40
2.13	Three-dimensional Stress and Strain	42
2.14	Plate Bending Element	44
2.15	Summary of Element Equations for Solids	47
2.16	Flow of Fluids: Navier–Stokes Equations	47
2.17	Simplified Flow Equations	50
2.17.1	<i>Steady State</i>	51
2.17.2	<i>Transient State</i>	53
2.17.3	<i>Convection</i>	53
2.18	Further Coupled Equations: Biot Consolidation	54
2.19	Conclusions	56
	References	56
3	Programming Finite Element Computations	59
3.1	Introduction	59
3.2	Local Coordinates for Quadrilateral Elements	59
3.2.1	<i>Numerical Integration for Quadrilaterals</i>	61
3.2.2	<i>Analytical Integration for Quadrilaterals</i>	63
3.3	Local Coordinates for Triangular Elements	64
3.3.1	<i>Numerical Integration for Triangles</i>	65
3.3.2	<i>Analytical Integration for Triangles</i>	65
3.4	Multi-Element Assemblies	66
3.5	‘Element-by-Element’ Techniques	68
3.5.1	<i>Conjugate Gradient Method for Linear Equation Systems</i>	68
3.5.2	<i>Preconditioning</i>	69

3.5.3	<i>Unsymmetric Systems</i>	70
3.5.4	<i>Symmetric Non-Positive Definite Equations</i>	71
3.5.5	<i>Eigenvalue Systems</i>	71
3.6	<i>Incorporation of Boundary Conditions</i>	72
3.6.1	<i>Convection Boundary Conditions</i>	74
3.7	<i>Programming using Building Blocks</i>	75
3.7.1	<i>Black Box Routines</i>	76
3.7.2	<i>Special Purpose Routines</i>	77
3.7.3	<i>Plane Elastic Analysis using Quadrilateral Elements</i>	77
3.7.4	<i>Plane Elastic Analysis using Triangular Elements</i>	81
3.7.5	<i>Axisymmetric Strain of Elastic Solids</i>	82
3.7.6	<i>Plane Steady Laminar Fluid Flow</i>	83
3.7.7	<i>Mass Matrix Formation</i>	83
3.7.8	<i>Higher-Order 2D Elements</i>	84
3.7.9	<i>Three-Dimensional Elements</i>	86
3.7.10	<i>Assembly of Elements</i>	90
3.8	<i>Solution of Equilibrium Equations</i>	95
3.9	<i>Evaluation of Eigenvalues and Eigenvectors</i>	96
3.9.1	<i>Jacobi Algorithm</i>	96
3.9.2	<i>Lanczos and Arnoldi Algorithms</i>	98
3.10	<i>Solution of First-Order Time-Dependent Problems</i>	99
3.11	<i>Solution of Coupled Navier–Stokes Problems</i>	103
3.12	<i>Solution of Coupled Transient Problems</i>	104
3.12.1	<i>Absolute Load Version</i>	105
3.12.2	<i>Incremental Load Version</i>	106
3.13	<i>Solution of Second-Order Time-Dependent Problems</i>	106
3.13.1	<i>Modal Superposition</i>	107
3.13.2	<i>Newmark or Crank–Nicolson Method</i>	109
3.13.3	<i>Wilson’s Method</i>	110
3.13.4	<i>Complex Response</i>	111
3.13.5	<i>Explicit Methods and Other Storage-Saving Strategies</i>	112
	<i>References</i>	113
4	<i>Static Equilibrium of Structures</i>	115
4.1	<i>Introduction</i>	115
	Program 4.1 One-dimensional analysis of axially loaded elastic rods using 2-node rod elements	116
	Program 4.2 Analysis of elastic pin-jointed frames using 2-node rod elements in two or three dimensions	121
	Program 4.3 Analysis of elastic beams using 2-node beam elements (elastic foundation optional)	127
	Program 4.4 Analysis of elastic rigid-jointed frames using 2-node beam/rod elements in two or three dimensions	133
	Program 4.5 Analysis of elastic–plastic beams or frames using 2-node beam or beam/rod elements in one, two or three dimensions	141
	Program 4.6 Stability (buckling) analysis of elastic beams using 2-node beam elements (elastic foundation optional)	150

	Program 4.7 Analysis of plates using 4-node rectangular plate elements. Homogeneous material with identical elements. Mesh numbered in x - or y -direction	153
4.2	Conclusions	157
4.3	Glossary of Variable Names	157
4.4	Exercises	159
	References	168
5	Static Equilibrium of Linear Elastic Solids	169
5.1	Introduction	169
	Program 5.1 Plane or axisymmetric strain analysis of a rectangular elastic solid using 3-, 6-, 10- or 15-node right-angled triangles or 4-, 8- or 9-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction	170
	Program 5.2 Non-axisymmetric analysis of a rectangular axisymmetric elastic solid using 8-node rectangular quadrilaterals. Mesh numbered in r - or z -direction	184
	Program 5.3 Three-dimensional analysis of a cuboidal elastic solid using 8-, 14- or 20-node brick hexahedra. Mesh numbered in xz -planes then in the y -direction	191
	Program 5.4 General 2D (plane strain) or 3D analysis of elastic solids. Gravity loading option	196
	Program 5.5 Plane or axisymmetric thermoelastic analysis of an elastic solid using 3-, 6-, 10- or 15-node right-angled triangles or 4-, 8- or 9-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction	205
	Program 5.6 Three-dimensional strain of a cuboidal elastic solid using 8-, 14- or 20-node brick hexahedra. Mesh numbered in xz -planes then in the y -direction. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver	209
	Program 5.7 Three-dimensional strain of a cuboidal elastic solid using 8-, 14- or 20-node brick hexahedra. Mesh numbered in xz -planes then in the y -direction. No global stiffness matrix. Diagonally preconditioned conjugate gradient solver. Optimised maths library, ABAQUS UMAT version	213
5.2	Glossary of Variable Names	221
5.3	Exercises	224
	References	232
6	Material Non-linearity	233
6.1	Introduction	233
6.2	Stress–strain Behaviour	235
6.3	Stress Invariants	236
6.4	Failure Criteria	238
	6.4.1 <i>Von Mises</i>	238
	6.4.2 <i>Mohr–Coulomb and Tresca</i>	239
6.5	Generation of Body Loads	240
6.6	Viscoplasticity	240

6.7	Initial Stress	242
6.8	Corners on the Failure and Potential Surfaces	243
	Program 6.1 Plane-strain-bearing capacity analysis of an elastic–plastic (von Mises) material using 8-node rectangular quadrilaterals. Flexible smooth footing. Load control. Viscoplastic strain method	244
	Program 6.2 Plane-strain-bearing capacity analysis of an elastic–plastic (von Mises) material using 8-node rectangular quadrilaterals. Flexible smooth footing. Load control. Viscoplastic strain method. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver	250
	Program 6.3 Plane-strain-bearing capacity analysis of an elastic–plastic (Mohr–Coulomb) material using 8-node rectangular quadrilaterals. Rigid smooth footing. Displacement control. Viscoplastic strain method	254
	Program 6.4 Plane-strain slope stability analysis of an elastic–plastic (Mohr–Coulomb) material using 8-node rectangular quadrilaterals. Gravity loading. Viscoplastic strain method	260
	Program 6.5 Plane-strain earth pressure analysis of an elastic–plastic (Mohr–Coulomb) material using 8-node rectangular quadrilaterals. Rigid smooth wall. Initial stress method	265
6.9	Elastoplastic Rate Integration	270
	6.9.1 Forward Euler Method	272
	6.9.2 Backward Euler Method	274
6.10	Tangent Stiffness Approaches	275
	6.10.1 Inconsistent Tangent Matrix	275
	6.10.2 Consistent Tangent Matrix	275
	6.10.3 Convergence Criterion	276
	Program 6.6 Plane-strain-bearing capacity analysis of an elastic–plastic (von Mises) material using 8-node rectangular quadrilaterals. Flexible smooth footing. Load control. Consistent tangent stiffness. Closest point projection method (CPPM)	277
	Program 6.7 Plane-strain-bearing capacity analysis of an elastic–plastic (von Mises) material using 8-node rectangular quadrilaterals. Flexible smooth footing. Load control. Consistent tangent stiffness. CPPM. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver	282
	Program 6.8 Plane-strain-bearing capacity analysis of an elastic–plastic (von Mises) material using 8-node rectangular quadrilaterals. Flexible smooth footing. Load control. Consistent tangent stiffness. Radial return method (RR) with ‘line search’	286
6.11	The Geotechnical Processes of Embanking and Excavation	289
	6.11.1 Embanking	289
	Program 6.9 Plane-strain construction of an elastic–plastic (Mohr–Coulomb) embankment in layers on a foundation using 8-node quadrilaterals. Viscoplastic strain method	290
	6.11.2 Excavation	294

	Program 6.10 Plane-strain construction of an elastic–plastic (Mohr–Coulomb) excavation in layers using 8-node quadrilaterals. Viscoplastic strain method	300
6.12	Undrained Analysis	305
	Program 6.11 Axisymmetric ‘undrained’ strain of an elastic–plastic (Mohr–Coulomb) solid using 8-node rectangular quadrilaterals. Viscoplastic strain method	308
	Program 6.12 Three-dimensional strain analysis of an elastic–plastic (Mohr–Coulomb) slope using 20-node hexahedra. Gravity loading. Viscoplastic strain method	313
	Program 6.13 Three-dimensional strain analysis of an elastic–plastic (Mohr–Coulomb) slope using 20-node hexahedra. Gravity loading. Viscoplastic strain method. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver	319
6.13	Glossary of Variable Names	322
6.14	Exercises	327
	References	331
7	Steady State Flow	333
7.1	Introduction	333
	Program 7.1 One-dimensional analysis of steady seepage using 2-node line elements	334
	Program 7.2 Plane or axisymmetric analysis of steady seepage using 4-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction	337
	Program 7.3 Analysis of plane free surface flow using 4-node quadrilaterals. ‘Analytical’ form of element conductivity matrix	344
	Program 7.4 General two- (plane) or three-dimensional analysis of steady seepage	351
	Program 7.5 General two- (plane) or three-dimensional analysis of steady seepage. No global conductivity matrix assembly. Diagonally preconditioned conjugate gradient solver	355
7.2	Glossary of Variable Names	359
7.3	Exercises	361
	References	367
8	Transient Problems: First Order (Uncoupled)	369
8.1	Introduction	369
	Program 8.1 One-dimensional transient (consolidation) analysis using 2-node ‘line’ elements. Implicit time integration using the ‘theta’ method	370
	Program 8.2 One-dimensional transient (consolidation) analysis (settlement and excess pore pressure) using 2-node ‘line’ elements. Implicit time integration using the ‘theta’ method	373
	Program 8.3 One-dimensional consolidation analysis using 2-node ‘line’ elements. Explicit time integration. Element by element. Lumped mass	377

Program 8.4	Plane or axisymmetric transient (consolidation) analysis using 4-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction. Implicit time integration using the ‘theta’ method	380
Program 8.5	Plane or axisymmetric transient (consolidation) analysis using 4-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction. Implicit time integration using the ‘theta’ method. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver	388
Program 8.6	Plane or axisymmetric transient (consolidation) analysis using 4-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction. Explicit time integration using the ‘theta = 0’ method	390
Program 8.7	Plane or axisymmetric transient (consolidation) analysis using 4-node rectangular quadrilaterals. Mesh numbered in $x(r)$ - or $y(z)$ -direction. ‘theta’ method using an element-by-element product algorithm	394
8.2	Comparison of Programs 8.4, 8.5, 8.6 and 8.7	397
Program 8.8	General two- (plane) or three-dimensional transient (consolidation) analysis. Implicit time integration using the ‘theta’ method	397
Program 8.9	Plane analysis of the diffusion–convection equation using 4-node rectangular quadrilaterals. Implicit time integration using the ‘theta’ method. Self-adjoint transformation	401
Program 8.10	Plane analysis of the diffusion–convection equation using 4-node rectangular quadrilaterals. Implicit time integration using the ‘theta’ method. Untransformed solution	405
Program 8.11	Plane or axisymmetric transient thermal conduction analysis using 4-node rectangular quadrilaterals. Implicit time integration using the ‘theta’ method. Option of convection and flux boundary conditions	410
8.3	Glossary of Variable Names	416
8.4	Exercises	419
	References	422
9	Coupled Problems	423
9.1	Introduction	423
Program 9.1	Analysis of the plane steady-state Navier–Stokes equation using 8-node rectangular quadrilaterals for velocities coupled to 4-node rectangular quadrilaterals for pressures. Mesh numbered in x -direction. Freedoms numbered in the order $u - p - v$	424
Program 9.2	Analysis of the plane steady-state Navier–Stokes equation using 8-node rectangular quadrilaterals for velocities coupled to 4-node rectangular quadrilaterals for pressures. Mesh numbered in x -direction. Freedoms numbered in the order $u - p - v$. Element-by-element solution using BiCGStab(1) with no preconditioning. No global matrix assembly	429

Program 9.3	One-dimensional coupled consolidation analysis of a Biot poroelastic solid using 2-node 'line' elements. Freedoms numbered in the order $v - u_w$	433
Program 9.4	Plane strain consolidation analysis of a Biot elastic solid using 8-node rectangular quadrilaterals for displacements coupled to 4-node rectangular quadrilaterals for pressures. Freedoms numbered in order $u - v - u_w$. Incremental load version	438
Program 9.5	Plane strain consolidation analysis of a Biot elastic solid using 8-node rectangular quadrilaterals for displacements coupled to 4-node rectangular quadrilaterals for pressures. Freedoms numbered in order $u - v - u_w$. Incremental load version. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver	445
Program 9.6	Plane strain consolidation analysis of a Biot poroelastic-plastic (Mohr-Coulomb) material using 8-node rectangular quadrilaterals for displacements coupled to 4-node rectangular quadrilaterals for pressures. Freedoms numbered in the order $u - v - u_w$. Viscoplastic strain method	448
9.2	Glossary of Variable Names	454
9.3	Exercises	459
	References	460
10	Eigenvalue Problems	461
10.1	Introduction	461
Program 10.1	Eigenvalue analysis of elastic beams using 2-node beam elements. Lumped mass	462
Program 10.2	Eigenvalue analysis of an elastic solid in plane strain using 4- or 8-node rectangular quadrilaterals. Lumped mass. Mesh numbered in y-direction	465
Program 10.3	Eigenvalue analysis of an elastic solid in plane strain using 4-node rectangular quadrilaterals. Lanczos method. Consistent mass. Mesh numbered in y-direction	469
Program 10.4	Eigenvalue analysis of an elastic solid in plane strain using 4-node rectangular quadrilaterals with ARPACK. Lumped mass. Element-by-element formulation. Mesh numbered in y-direction	474
10.2	Glossary of Variable Names	477
10.3	Exercises	480
	References	482
11	Forced Vibrations	483
11.1	Introduction	483
Program 11.1	Forced vibration analysis of elastic beams using 2-node beam elements. Consistent mass. Newmark time stepping	483
Program 11.2	Forced vibration analysis of an elastic solid in plane strain using 4- or 8-node rectangular quadrilaterals. Lumped mass. Mesh numbered in the y-direction. Modal superposition	489

Program 11.3	Forced vibration analysis of an elastic solid in plane strain using rectangular 8-node quadrilaterals. Lumped or consistent mass. Mesh numbered in the y -direction. Implicit time integration using the ‘theta’ method	493
Program 11.4	Forced vibration analysis of an elastic solid in plane strain using rectangular 8-node quadrilaterals. Lumped or consistent mass. Mesh numbered in the y -direction. Implicit time integration using Wilson’s method	498
Program 11.5	Forced vibration of a rectangular elastic solid in plane strain using 8-node quadrilateral elements numbered in the y -direction. Lumped mass, complex response	501
Program 11.6	Forced vibration analysis of an elastic solid in plane strain using uniform size rectangular 4-node quadrilaterals. Mesh numbered in the y -direction. Lumped or consistent mass. Mixed explicit/ implicit time integration	504
Program 11.7	Forced vibration analysis of an elastic solid in plane strain using rectangular 8-node quadrilaterals. Lumped or consistent mass. Mesh numbered in the y -direction. Implicit time integration using the ‘theta’ method. No global matrix assembly. Diagonally preconditioned conjugate gradient solver	508
Program 11.8	Forced vibration analysis of an elastic–plastic (von Mises) solid in plane strain using rectangular 8-node quadrilateral elements. Lumped mass. Mesh numbered in the y -direction. Explicit time integration	512
11.2	Glossary of Variable Names	517
11.3	Exercises	521
	References	522
12	Parallel Processing of Finite Element Analyses	523
12.1	Introduction	523
12.2	Differences between Parallel and Serial Programs	525
12.2.1	<i>Parallel Libraries</i>	525
12.2.2	<i>Global Variables</i>	526
12.2.3	<i>MPI Library Routines</i>	526
12.2.4	<i>The _pp Appendage</i>	527
12.2.5	<i>Simple Test Problems</i>	527
12.2.6	<i>Reading and Writing</i>	530
12.2.7	<i>rest Instead of nf</i>	532
12.2.8	<i>Gathering and Scattering</i>	533
12.2.9	<i>Reindexing</i>	533
12.2.10	<i>Domain Composition</i>	533
12.2.11	<i>Third-party Mesh-partitioning Tools</i>	534
12.2.12	<i>Load Balancing</i>	535
Program 12.1	Three-dimensional analysis of an elastic solid. Compare Program 5.6	536
Program 12.2	Three-dimensional analysis of an elastoplastic (Mohr–Coulomb) solid. Compare Program 6.13	542
Program 12.3	Three-dimensional Laplacian flow. Compare Program 7.5	548

Program 12.4	Three-dimensional transient heat conduction—implicit analysis in time. Compare Program 8.5	553
Program 12.5	Three-dimensional transient flow—explicit analysis in time. Compare Program 8.6	562
Program 12.6	Three-dimensional steady-state Navier–Stokes analysis. Compare Program 9.2	565
Program 12.7	Three-dimensional analysis of Biot poro elastic solid. Incremental version. Compare Program 9.5	572
Program 12.8	Eigenvalue analysis of three-dimensional elastic solid. Compare Program 103	576
Program 12.9	Forced vibration analysis of a three-dimensional elastic solid. Implicit integration in time. Compare Program 11.7	581
Program 12.10	Forced vibration analysis of three-dimensional elastoplastic solid. Explicit integration in time. Compare Program 11.8	585
12.3	Graphics Processing Units	589
Program 12.11	Three-dimensional strain of an elastic solid using 8-, 14- or 20-node brick hexahedra. No global stiffness matrix assembly. Diagonally preconditioned conjugate gradient solver. GPU version. Compare Program 5.7	589
12.4	Cloud Computing	594
12.5	Conclusions	596
12.6	Glossary of Variable Names	597
	References	602
Appendix A	Equivalent Nodal Loads	605
Appendix B	Shape Functions and Element Node Numbering	611
Appendix C	Plastic Stress-Strain Matrices and Plastic Potential Derivatives	619
Appendix D	main Library Subprograms	623
Appendix E	geom Library Subroutines	635
Appendix F	Parallel Library Subroutines	639
Appendix G	External Subprograms	645
Author Index		649
Subject Index		653

1 Preliminaries: Computer Strategies

1.1 Introduction

Many textbooks exist which describe the principles of the finite element method of analysis and the wide scope of its applications to the solution of practical engineering and scientific problems. Usually, little attention is devoted to the construction of the computer programs by which the numerical results are actually produced. It is presumed that readers have access to pre-written programs (perhaps to rather complicated ‘packages’) or can write their own. However, the gulf between understanding in principle what to do, and actually doing it, can still be large for those without years of experience in this field.

The present book bridges this gulf. Its intention is to help readers assemble their own computer programs to solve particular engineering and scientific problems by using a ‘building block’ strategy specifically designed for computations via the finite element technique. At the heart of what will be described is not a ‘program’ or a set of programs but rather a collection (library) of procedures or subroutines which perform certain functions analogous to the standard functions (SIN, SQRT, ABS, etc.) provided in permanent library form in all useful scientific computer languages. Because of the matrix structure of finite element formulations, most of the building block routines are concerned with manipulation of matrices.

The building blocks are then assembled in different patterns to make test programs for solving a variety of problems in engineering and science. The intention is that one of these test programs then serves as a platform from which new applications programs are developed by interested users.

The aim of the present book is to teach the reader to write intelligible programs and to use them. Both serial and parallel computing environments are addressed and the building block routines (numbering over 100) and all test programs (numbering over 70) have been verified on a wide range of computers. Efficiency is considered.

The chosen programming language is FORTRAN which remains, overwhelmingly, the most popular language for writing large engineering and scientific programs. Later in this chapter a brief description of the features of FORTRAN which influence the programming of the finite element method will be given. The most recent update of the language was in 2008 (ISO/IEC 1539-1:2010). For parallel environments, MPI has been used, although the programming strategy has also been tested with OpenMP, or a combination of the two.