Preface

The finite element method can be applied to problems in various fields of science and engineering. It is well established and its algorithms are presented in numerous publications. Many books are devoted to different aspects of the finite element method. Still, algorithms of the finite element method are difficult to understand, and programming of the finite element techniques is complicated.

Objective

This book focuses on algorithms of the finite element method and their programming. First, general equations of the finite element method for solving solid mechanics and thermal conductivity problems are introduced. Then, algorithms of the finite element method and their programming in Java™ are considered. In addition to solution methods, the book presents algorithms and programming approaches for mesh generation and visualization.

Why Java?

The Java language is selected for its numerous advantages: an object-oriented paradigm, multiplatform support, ease of development, reliability and stability, the ability to use legacy C or C++ code, good documentation, development-tool availability, etc. The Java runtime environment always checks subscript legitimacy to ensure that each subscript is equal to or greater than zero and less than the number of elements in the array. Even this simple feature means a lot to developers. As a result, Java programs are less susceptible to bugs and security flaws. Java also provides application programming interfaces (APIs) for development of GUI, and three-dimensional graphics applications.

I started programming finite elements in Fortran. Later I used Pascal, C, and C++, before settling on Java. Comparing these languages I found that programming finite elements in Java is not just efficient because the productivity is higher and the code contains fewer errors, but is also more pleasant.

An opinion exists that Java is not suitable for computational modeling and finite element programming because of its slow execution speed. It is true that Java is slower than C in performing “multiply-add” arithmetic inside double and triple loops. However, tuning of important Java code fragments provides computational speed comparable to that of C.

The attractive features of Java prevail over some of its drawbacks. In my opinion, Java is good for both learning finite element programming and for finite element code development with easy debugging, modification, and support. Further, Java is easy to understand even for those who do not program in Java. In most cases, methods performing computations can be easily used with minimum modification to procedures written in other languages such as C or C++.

For Whom Is This Book Written?

This book is an introductory text about finite element algorithms and especially finite element programming using an object-oriented approach. All important aspects of finite element techniques are considered – finite element solution, generation of finite element meshes, and visualization of finite element models and results.

The book is useful for graduate and undergraduate students for self-study of finite element algorithms and programming techniques. It can be used as a textbook for introductory graduate courses or in advanced undergraduate courses. I hope that the book will be interesting to researchers and engineers who are already familiar with finite element algorithms and codes, since the programming approaches of this book differ from other publications.

Organization

The book is organized into four parts. Part I covers general formulation of the finite element method. Chapter 1 introduces the finite element formulation in the one-dimensional case. Both the Galerkin method and variational formulations are considered. Chapter 2 presents finite element equations for heat transfer problems derived with the use of the Galerkin approach. Chapter 3 contains variational formulation of general finite element equations for solid mechanics problems. An object-oriented approach to development of the finite element code is discussed in Chapter 4.

Part II is devoted to algorithms and programming of the finite element solution of solid mechanics problems. Chapter 5 considers the class structure of the finite element processor code. Data structures of the finite element model and corresponding Java class are presented in Chapter 6. Relations for elastic material and the corresponding class are given in Chapter 7. Chapters 8 and 9 describe an abstract class for a finite element and a class for numerical integration. Chapters 10–13 present algorithms and programming implementation for two- and three-dimensional isoparametric quadratic elements. Assembly and solution of the finite element equation system are discussed in Chapters 14–16. Chapter 17 is devoted to assembly of the global load vector. Computing stress increments is presented in Chapter 18. Solution and programming implementation of elastic–plastic problems is discussed in Chapter 19.

Part III focuses on mesh generation for solution of two- and three-dimensional finite element problems. The block decomposition method used for mesh generation and general organization of the mesh generator is given in Chapter 20. Chapter 21 presents two-dimensional mesh generators. Chapter 22 describes three-dimensional mesh generation by sweeping a two-dimensional mesh. Chapters 23–25 contain algorithms and classes for pasting mesh blocks and various operations on mesh blocks, including their pasting for creation of a complex mesh of simple blocks.

Part IV describes algorithms for visualization of finite element methods and results. Chapter 26 introduces the Java 3D™ API, which is used for rendering three-dimensional objects. Visualization algorithms for higher-order finite elements and visualization code structure are presented in Chapter 27. A scene graph for visualization of meshes and results is discussed in Chapter 28. Chapter 29 describes algorithms for creation of the model surface. Chapters 30 and 31 are devoted to subdivision of the model surface into polygons. Chapter 32 presents Java classes for results field, color-gradation strip, mouse interaction and lights.

Appendices A, B, and C contain brief instructions for preparing data for finite element programs that perform problem solution, mesh generation, and visualization of models and results. Appendix D provides examples of finite element analysis: mesh generation, problem solution, and visualization of results.

How This Book Differs from Others

There are many books about the finite element method. Some of them contain finite element program segments. Two qualities distinguish this book from other books on the finite element method.

First, programming in this book is based upon the Java programming language. In my opinion, Java is well suited for explaining programming of the finite element method. It allows compact and simple code to be written. This helps greatly because the reader has a chance to actually read finite element programs and to understand them.

Secondly, algorithms presented in this book are tightly connected to programming. The book is written around one finite element Java program, which includes solution of solid mechanics boundary value problems, mesh generation, and visualization of finite element models and results. Presentation of computational algorithms is followed by a Java class or group of Java methods and then accompanied by code explanation.

Web Resources

The Java programs and examples presented in this book are available on the Web: `http://www.springer.com/978-1-84882-971-8`. Comments, suggestions, and corrections are welcome by e-mail: `fem.java@gmail.com`.

About the Author

Gennadiy Nikishkov got his Ph.D. and D.Sc. degrees from the Moscow Engineering Physics Institute (Technical University) in computational mechanics. He held a Professor position at the Moscow Engineering Physics Institute. He also had visiting positions at Georgia Institute of Technology (USA), Karlsruhe Research Center (Germany), RIKEN Institute of Physical and Chemical Research (Japan), GKSS Research Center (Germany), and the University of California at Los Angeles (USA). Currently, he is a Professor at the University of Aizu (Japan). His research interests include computational mechanics, computational fracture mechanics, computational nanomechanics, development of finite element and boundary element codes, scientific visualization, and computer graphics.

Acknowledgments

The author is grateful to Prof. Michael Cohen, University of Aizu, Japan, for his attentive reading of the book manuscript and his valuable suggestions, and to Dr. Yuriy Nikishkov, Georgia Institute of Technology, USA, for his advice related to development of the Java program. I also acknowledge my colleagues who contributed to this book through many discussions on computational methods over the years.

I thank the anonymous reviewers for their precious comments. Special thanks go to my editor Oliver Jackson and to Ms. Aislinn Bunning and Ms. Sorina Moosdorf for their help, which greatly improved the quality of the book. Finally, I express my loving appreciation to my wife Valentina for her support and encouragement.

Aizu-Wakamatsu, Japan *Gennadiy Nikishkov*
July 2009

Springer
http://www.springer.com/978-1-84882-971-8

Programming Finite Elements in Java™
Nikishkov, G.P.
2010, XVI, 402 p. With online files/update., Hardcover
ISBN: 978-1-84882-971-8