PARALLEL-VECTOR EQUATION SOLVERS FOR FINITE ELEMENT ENGINEERING APPLICATIONS

Duc T. Nguyen

Parallel-Vector Equation Solvers for Finite Element Engineering Applications

Duc Thai Nguyen
Old Dominion University
Norfolk, Virginia

Kluwer Academic / Plenum Publishers
New York, Boston, Dordrecht, London, Moscow

Library of Congress Cataloging-in-Publication Data

Nguyen, Duc T.
Parallel-vector equation solvers for finite element engineering applications/Duc Thai Nguyen.
p. cm.
Includes bibliographical references and index.
ISBN 0-306-46640-6
1. Finite element method. 2. Parallel processing (Electronic computers) 3. Differential equations—Numerical solutions. I. Title.

TA347.F5 N48 2001
620'.001'51535—dc21

2001038333

ISBN 0-306-46640-6

Parallel-Vector Equation Solvers for Finite Element Engineering Applications

To Dac K. Nguyen
Thinh T. Thai
Hang N. Nguyen
Eric N. D. Nguyen and Don N. Nguyen

Preface

In spite of the fact that parallel-vector computational (equation solution) algorithms have been receiving a lot of attentions for over 20 years, and a large number of research articles have been published during this period, a limited number of texts and research books have been written on the subject. Most (if not all) existing texts on this subject have been written by computer scientists, and/or applied mathematicians. Majority of existing texts have been over-emphasizing on theoretical developments of new, and/or promising parallel (equation solution) algorithms (for varieties of applications in Engineering and Science disciplines). Materials presented in most existing texts are either too condense (without enough important detailed explanations), or too advance for the typical senior undergraduate and/or graduate engineering students. It should be emphasized here that while many important theoretical developments, which have significant impacts on highly efficient existing parallel-vector (equation solution) algorithms, have been carefully discussed and well-documented in current texts, important detailed computer implementations of the developed algorithms, however, have been usually omitted. Furthermore, it should be kept in minds that while few existing texts in this subject (direct equation solution algorithms for parallel and/or vector computers) have been written by computer scientists, and/or applied mathematicians, truly large-scale models (which require parallel and vector capabilities offered by modern high-performance computers) are often generated, solved, and interpreted by the engineering communities.

This book is written to address the concerns mentioned above and is intended to serve as a textbook for senior undergraduate, and graduate "engineering" students. A number of state-of-the-art FORTRAN codes, however, have been developed and discussed with great details in this textbook. Special efforts have been made by the author to present the materials in such a way to minimize the mathematical background requirements for typical senior undergraduate, and graduate engineering students. Thus, compromises between rigorous mathematics and practical simplicities are sometimes necessary.

This book has several unique features that distinguish it from other books:

1. Simplicity: The book has been written and explained in simple fashion, so that senior undergraduate and first year graduate students (in Civil, Mechanical, Aerospace, Electrical, Computer Science and Mathematic departments) can understand the presented materials with minimum background requirements. A working (undergraduate) knowledge in FORTRAN codings is helpful to understand the "detailed codings" of the presented materials. Some (undergraduate) linear algebra background should be useful, although it is NOT a requirement for reading and understanding the materials in the book. Undergraduate background in Matrix Structural Analysis and/or Finite Element Analysis should be useful, only for the materials presented in Chapter 3. Graph theories have not been traditionally introduced in the undergraduate/graduate engineering curriculums and therefore graph theories are not required to understand the materials presented in this book.
2. Algorithms are discussed for different parallel and/or vector computer platforms: Parallel and/or vectorized algorithms for various types of direct equation solvers are

presented and discussed for both "shared memory" (such as the Cray-2, Cray-YMP, Cray-C90, Convex) and "distributed memory" (such as the Intel i860, Intel Paragon, IBM-SP2, Meiko) computer platforms. The vectorized algorithms can also be "efficiently" executed on IBM-R6000/590 workstations. The vectorized algorithms and their associated FORTRAN codes can also be executed (with less efficiency) on other workstations and/or personal computers (P.C.) without having vectorized capabilities.

3. More emphasis on important detailed FORTRAN computer implementations: Efforts have been made to explain to the readers on important detailed FORTRAN computer implementations of various algorithms presented in the book. Thus, the readers should be able to incorporate the presented computer codes, subroutines into his/her application codes.
4. Several state-of-the-art FORTRAN equation solvers are discussed: While great amounts of effort have been spent to explain the detailed algorithms in a "simple fashion," many state-of-the-art equation solvers have been developed and presented in the book. Many of the presented solvers have been used by universities, large aerospace corporations and government research laboratories in the U.S., Europe and Asia.
5. Large-scale practical engineering finite element models are used: For derivations and explanations of various algorithms described in the book, small-scale examples are used to simplify and to facilitate the discussions. However, several medium to large-scale, practical engineering finite element models are used to demonstrate the efficiency and accuracy of the presented algorithms.
6. Algorithms are available for different types of linear equations: Different types of algorithms for the solutions of various types of system of simultaneous linear equations are presented in the book. Symmetrical/unsymmetrical, positive definite/negative definite/indefinite, incore/out-of-core, skyline/variable bandwidth/sparse/tridiagonal system of equations have all been treated in great detail by the author.

The book contains 11 chapters. Chapter 1 presents a brief review of some basic descriptions of shared and distributed parallel-vector computers. Measurements for algorithms' performance, and commonly "good practices" to achieve vector speed are also discussed in this chapter. Different storage schemes for the coefficient (stiffness) matrix (of system of linear equations) are discussed with great details in Chapter 2. Efficient parallel algorithms for generation and assembly of finite element coefficient (stiffness) matrices are explained in Chapter 3. Different parallel-vector "skyline" algorithms for shared memory computers (such as Cray-YMP, Cray-C90 etc...) are developed and evaluated in Chapter 4. These algorithms have been developed in conjunction with the skyline storage scheme, proposed earlier in Chapter 2. Parallel-vector "variable bandwidth" equation solution algorithms (for shared memory computers) are presented and explained in Chapter 5. These algorithms have been derived based upon the variable bandwidth storage scheme, proposed earlier in Chapter 2. Out-of-core equation solution algorithms on shared memory computers are considered in Chapter 6. These algorithms are useful for cases where very large-scale models need to be solved, and there are not enough core-memories to hold all arrays in the in-core memories. Parallel-vector equation solution strategies for "distributed-memory" computers are discussed in Chapter 7. These equation solution strategies are based upon the parallel generation and assembly of finite element (stiffness) matrices, suggested earlier in Chapter 3. Unsymmetrical banded system of equations are treated in Chapter 8, where both parallel and vector strategies are described. Parallel algorithms for tri-diagonal system of equations on distributed computers are explained in Chapter 9. Sparse equation solution algorithms are presented in Chapter 10. Unrolling techniques to enhance the vector performance of sparse algorithms are also explained in this chapter. Finally, system of sparse equations where the coefficient (stiffness) matrix is symmetrical/ unsymmetrical and/or indefinite (where special pivoting strategies are required) are considered with great details in Chapter 11.

The book also contains a limited number of exercises to further supplement and reinforce the concepts and ideas presented. The references are listed at the end of each chapter.

The author would like to invite the readers to point out any errors that come to their attention. The author also welcomes any comments and suggestions from the readers.

Duc Thai Nguyen

Norfolk, Virginia

Acknowledgments

During preparation of this book, I have received (directly and indirectly) help from many people. First, I would like to express my sincere gratitude to my colleagues at NASA Langley Research Center: Dr. Olaf O. Storaasli, Dr. Jerrold M. Housner, Dr. James Starnes, Dr. Jaroslaw S. Sobieski, Dr. Keith Belvin, Dr. Peigman M. Maghami, Dr. Tom Zang, Dr. John Barthelemy, Dr. Carl Gray Jr., Dr. Steve Scotti, Dr. Kim Bey, Dr. Willie R. Watson, and Dr. Andrea O. Salas for their encouragement and support on the subject of this book during the past 13 years.

The close collaborative works with Dr. Olaf O. Storaasli and Dr. Jiangning Qin, in particular, have direct impacts on the writing of several chapters in this textbook.

I am very grateful to Dr. Lon Water (Maui, Hawaii), Professors Pu Chen (China), S.D. Rajan (Arizona), B.D. Belegundu (Pennsylvania), J.S. Arora (Iowa), Dr. Brad Maker (California), Dr. Gerald Goudreau (California), Dr. Esmond Ng (Lawrence Berkeley Laboratory, California) and Mr. Maurice Sancer (California) for their enthusiasm and supports of many topics discussed in this book.

My appreciation also goes to several of my former doctoral students, such as Dr. T.K. Agarwal, Dr. John Zhang, Dr. Majdi Baddourah, Dr. Al-Nasra, and Dr. H. Runesha who have worked with me for several years. Some of their research contributions have been included in this book.

In addition, I would like to thank my colleagues at Old Dominion University (ODU) and Hong Kong University of Science and Technology (HKUST) for their support, collaborative works, and friendship. Among them, Professor A. Osman Akan, Professor Isao Ishibashi, Professor Chuh Mei, and Professor Zia Razzaq at ODU, Professor T.Y. Paul Chang, and Professor Pin Tong at HKUST. Substantial portions of this textbook have been completed during my sabbatical leave period (January 1 - December 30, 1996) from O.D.U. to work at HKUST (during February 22 - August 22, 1996).

The successful publication and smooth production of this book are due to ODU skillful office supported staff: Mrs. Sue Smith, Mrs. Mary Carmone, Mrs. Deborah L. Miller, and efficient management and careful supervision of Mr. Tom Cohn, Ms. Ana Bozicevic, and Mr. Felix Portnoy, Editors of Kluwer/Plenum Publishing Corporation.

Special thanks go to Ms. Catherine John, from Academic Press (AP) Ltd., London, UK for allowing us to reproduce some materials from the AP textbook "Sparse Matrix Technology," (by Sergio Pissanetzky) for discussions in Chapter 10 (Tables 10.2 and 10.5) of our book.

Last but not least, I would like to thank my parents (Mr. Dac K. Nguyen, and Mrs. Thinh T. Thai), my family (Hang N. Nguyen, Eric N. D. Nguyen and Don N. Nguyen), whose encouragement has been ever present.

Duc T. Nguyen
Norfolk, Virginia

Disclaimer of Warranty

Parallel-Vector Equation Solvers for Finite Element Engineering Applications

Contents

1 Introduction

1.1 Parallel Computers

Modern high performance computers have multiple processing capabilities. The Convex, Sequent, Alliant, Cray-2, Cray-YMP [1.1] and Cray-C90 [1.2], parallel computers, for example, belong to the broad class of "shared memory" computers. The nCUBE, Intel i860, Intel Paragon, Meiko, and IBM-SP2 [1.3, 1.4] parallel computers, however, belong to the broad class of "distributed memory", or "message passing" computers. Shared memory computers, in general, consist of few (say 20, or less) processors. Each processor has its own local memory. Different processors, however, can be communicated to each other through shared memory area, as shown in Figure 1.1.

Shared Memory Area			
Local Mem.	Local Mem.	Local Mem.	Local Mem.
Processor 1	Processor 2	Processor 3	Processor 4

Figure 1.1 Shared memory parallel computers

Distributed memory (or message passing) computers, in general, consist of many (say hundreds, or thousands) processors (or nodes). Each processor has its own local memory, but the processor itself usually is less powerful (in terms of computational speed and memories) than its counterpart shared memory processor.

Communication amongst different nodes can only be done by message passing, as shown in Figure 1.2. Designing efficient algorithms, which can fully exploit the parallel and vector capabilities offered by shared memory computers have already been challenging tasks. It is generally safe to state that it is even more difficult to develop and to implement efficient parallel-vector algorithms on distributed memory computers!