国际著名数学图书——影印版

Numerical Linear Algebra on High-Performance Computers

高性能计算机上的数值线性代数

Jack J. Dongarra
Iain S. Duff, Danny C. Sorensen
Henk A. van der Vorst
著

TSINGHUA UNIVERSITY PRESS
siam

Numerical Linear Algebra on High-Performance Computers

高性能计算机上的数值线性代数

Jack J. Dongarra, Iain S. Duff
Danny C. Sorensen 著
Henk A. van der Vorst

TSINGHUA UNIVERSITY PRESS
北京

siam

Jack J. Dongarra, Iain S. Duff, Danny C. Sorensen, Henk A. van der Vorst
Numerical Linear Algebra on High-Performance Computers
ISBN:0-89871-428-1

图书在版编目（CIP）数据

高性能计算机上的数值线性代数 = Numerical Linear Algebra on High-Performance Computers：英文 /（美）冬格拉（Dongarra, J. J.）等著 .-- 影印本 . -- 北京：清华大学出版社，2011.2
（国际著名数学图书）
ISBN 978-7-302-24499-8

Ⅰ. ①高… Ⅱ. ①冬… Ⅲ. ①线性代数计算法-英文 Ⅳ. ①O241.6

中国版本图书馆CIP数据核字（2010）第260805号

责任编辑：陈朝晖
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：175×245　**印　张**：22.75
版　　次：2011 年 2 月第 1 版　　**印　　次**：2011 年 2 月第1 次印刷
印　　数：1～3000
定　　价：49.00 元

产品编号：027798-01

About the Authors

Jack Dongarra holds a joint appointment as Distinguished Professor of Computer Science in the Computer Science Department at the University of Tennessee (UT) and as Distinguished Scientist in the Mathematical Sciences Section at Oak Ridge National Laboratory (ORNL) under the UT/ORNL Science Alliance Program. He specializes in numerical algorithms in linear algebra, parallel computing, use of advanced-computer architectures, programming methodology, and tools for parallel computers. Other current research involves the development, testing and documentation of high quality mathematical software. He was involved in the design and implementation of the software packages EISPACK, LINPACK, the BLAS, LAPACK, ScaLAPACK, the Templates project, *netlib*, PVM, MPI, the National High-Performance Software Exchange, and NetSolve and is currently involved in the design of algorithms and techniques for high performance computer architectures. Other experience includes work as a computer scientist and senior computer scientist in the Mathematics and Computer Science Division at Argonne National Laboratory from 1973 to 1989, as a visiting scientist with the Center for Supercomputing Research and Development at the University of Illinois at Urbana during 1987, as a visiting scientist at IBM's T. J. Watson Research Center in 1981, as a consultant to Los Alamos Scientific Laboratory in 1978, as a research assistant with the University of New Mexico in 1978, and as a visiting scientist at Los Alamos Scientific Laboratory in 1977. Dongarra received a Ph.D. in applied mathematics from the University of New Mexico in 1980, an M.S. in computer science from the Illinois Institute of Technology in 1973, and a B.S. in mathematics from Chicago State University in 1972. He has published numerous articles, papers, reports and technical memoranda, and has given many presentations on his research interests and is a Fellow of the American Association for the Advancement of Science (AAAS). `http://www.netlib.org/utk/people/JackDongarra/`

Iain S. Duff is Group Leader of Numerical Analysis in the Department for Computation and Information at the CCLRC Rutherford Appleton Laboratory in Oxfordshire, England. He is also the Project Leader for the Parallel Algorithms Group at CERFACS in Toulouse and is a Visiting Professor of mathematics at the University of Strathclyde. After completing his D.Phil. at Oxford, he was a Harkness Fellow in the United States, visiting Stony Brook and Stanford. He then spent two years lecturing at the University of Newcastle upon Tyne before joining the Harwell Laboratory, where he became Group Leader of

Numerical Analysis in 1986. In 1990, the Group moved to the Atlas Centre at the Rutherford Appleton Laboratory. He has had several extended visits to Argonne National Laboratory, the Australian National University, the University of Colorado at Boulder, Stanford University, and the University of Umeå.

He is a life Fellow and an Honorary Secretary of the Institute of Mathematics and its Applications and is chairman of the IMA Programme Committee. He is Editor of the IMA Journal of Numerical Analysis and of the IMANA Newsletter. He is a member of SIAM (USA), SMAI (France), and SBMAC (Brazil) and is an IMA representative on the Steering Committee of CICIAM. He is Editor or Associate Editor of thirteen journals in numerical analysis or scientific computation.

His main interests are in the solution of large sparse systems of equations, more recently on high performance computing platforms. His Group at RAL supports and develops the Harwell Subroutine Library. The Group at CERFACS is concerned with more general scientific computation and are involved in many European Projects.

`http://www.dci.clrc.ac.uk/Person.asp?I.S.Duff`

Danny C. Sorensen is a professor of the Mathematical Sciences Department of Rice University. His research interests are in numerical analysis and parallel computation, with specialties in numerical linear algebra, use of advanced-computer architectures, programming methodology and tools for parallel computers, and numerical methods for nonlinear optimization. Sorensen was a computer scientist and senior computer scientist in the Mathematics and Computer Science Division at Argonne National Laboratory from 1980 to 1989. He has also been a visiting professor at the Department of Operations Research at Stanford University and the Department of Mathematics, University of California at San Diego and a visiting scientist with the Center for Supercomputing Research and Development at the University of Illinois at Urbana.

Henk A. van der Vorst is a professor in numerical analysis in the Mathematical Department of Utrecht University in the Netherlands. His current research interests include iterative solvers for linear systems, large sparse eigenproblems, and the design of algorithms for parallel computers. Van der Vorst received a Ph.D. in applied mathematics from the University of Utrecht in 1982 for a thesis on the effect of preconditioning. Together with Meijerink, he proposed in the mid-1970s the so-called ICCG method, a combination of CG with Incomplete Cholesky decomposition and they proved the existence of incomplete LU decompositions for M-matrices. Furthermore, he introduced Bi-CGSTAB (1992), GMRESR (together with Vuik, 1994), and the Jacobi-Davidson methods (together with Sleijpen, 1996). For the paper on Jacobi-Davidson methods for standard eigenproblems, he and Sleijpen received a SIAG-LA Award in 1997. He authored and co-authored over a hundred publications on these and other subjects in numerical linear algebra and on parallel linear algebra. These publications include two joint papers with Gene Golub on the State of the Art in Iterative Methods. He also made a contribution to the SIAM book on templates for iterative methods for (sparse) linear systems. Van der Vorst was a senior consultant at the Academic Computing Centre Utrecht for more than 12

years, until fall 1984. From 1984 to 1990 he was a professor in numerical linear algebra and supercomputing in the Department of Technical Mathematics and Computer Science of Delft University.
`http://www.math.ruu.nl/people/vorst`

Preface

The purpose of this book is to unify and document in one place many of the techniques and much of the current understanding about solving systems of linear equations on vector and parallel computers. This book is not a textbook, but it is meant to provide a fast entrance to the world of vector and parallel processing for these linear algebra applications. We intend this book to be used by three groups of readers: graduate students, researchers working in computational science, and numerical analysts. As such, we hope this book can serve both as a reference and as a supplement to a teaching text on aspects of scientific computation.

The book is divided into five major parts: (1) introduction to terms and concepts, including an overview of the state of the art for high-performance computers and a discussion of performance evaluation (Chapters 1-4); (2) direct solution of dense matrix problems (Chapter 5); (3) direct solution of sparse systems of equations (Chapter 6); (4) iterative solution of sparse systems of equations (Chapters 7-9); and (5) iterative solution of sparse eigenvalue problems (Chapters 10-11). Any book that attempts to cover these topics must necessarily be somewhat out of date before it appears, because the area is in a state of flux. We have purposely avoided highly detailed descriptions of popular machines and have tried instead to focus on concepts as much as possible; nevertheless, to make the description more concrete, we do point to specific computers.

Rather than include a floppy disk containing the software described in the book, we have included a pointer to *netlib*. The problem with floppies in books is that they are never around when one needs them, and the software may undergo changes to correct problems or incorporate new ideas. The software included in *netlib* is in the public domain and can be used freely. With *netlib* we hope to have up-to-date software available at all times. A directory in *netlib* called *ddsv* contains the software (see `http://www.netlib.org/ddsv/`), and Appendix A of this book discusses what is available and how to make a request from *netlib*.

This book is a major revision to the book entitled *Solving Linear Systems on Vector and Shared Memory Computers*, published by SIAM in 1990. The book updates the material focusing on vector and parallel computing for linear algebra and includes new material on the eigenvalue problem.

We express appreciation to all those who helped in the preparation of this work, in particular to Gail Pieper for her tireless efforts in proofreading drafts

and improving the quality of the presentation and to Ed Anderson, Michael Botchev, Jeremy Du Croz, Victor Eijkhout, Robert Funderlic, Antoine Petitet, and Charlie Van Loan for their help in proofreading and their many suggestions to improve the readability. Much of the dense linear algebra parts would not be possible without the efforts and support of the developers of LAPACK, ScaLAPACK, and ARPACK.

Jack J. Dongarra
Iain S. Duff
Danny C. Sorensen
Henk A. ven der Vorst

Introduction

The recent availability of advanced-architecture computers has had a significant impact on all spheres of scientific computation including algorithm research and software development in numerical linear algebra. This book discusses some of the major elements of these new computers and indicates some recent developments in sparse and full linear algebra that are designed to exploit these elements.

The three main novel aspects of these advanced computers are the use of vectorization, parallelism, and super-scalar architectures, although how these are accommodated varies greatly between architectures. The first commercially available vector machine to have a significant impact on scientific computing was the CRAY-1, the first machine being delivered to Los Alamos in 1976. Thus, the use of vectorization is by now quite mature, and a good understanding of this architectural feature and general guidelines for its exploitation are now well established. However, the first commercially viable parallel machine was the Alliant in 1985, and more highly parallel machines did not appear on the marketplace until 1988. Thus, there remains a relative lack of definition and maturity in this area, although some guidelines and standards on the exploitation of parallelism are beginning to emerge.

Our background is in scientific computing rather than computer scientists; as such, one of our intentions in writing this book is to provide the computing infrastructure and necessary definitions to guide the computational scientist and, at the very least, to equip him or her with enough understanding to be able to read computer documentation and appreciate the influence of some of the major aspects of novel computer design. The majority of this basic material is covered in Chapter 1, although we address further aspects related to implementation and performance in Chapters 3 and 4. In such a volatile marketplace it is not sensible to concentrate too heavily on any specific architecture or any particular manufacturer, but we feel it is useful to illustrate our general remarks by reference to some currently existing machines. This we do in Chapter 2, as well as in Chapter 4 where we present some performance profiles for current machines.

Linear algebra, in particular the solution of linear systems of equations and eigenvalue problems, lies at the heart of most calculations in scientific computing. We thus concentrate on this area in this book, examining algorithms and software for dense coefficient matrices in Chapter 5 and for sparse direct sys-

tems in Chapter 6; iterative methods of solution are covered in Chapters 7-9, and Chapters 10 and 11 deal with large sparse eigenvlaue problems.

Within scientific computation, parallelism can be exploited at several levels. At the highest level a problem may be subdivided even before its discretization into a linear (or nonlinear) system. This technique, typified by domain decomposition, usually results in large parallel tasks ideal for mapping onto a distributed-memory architecture. In keeping with our decision to minimize machine description, we refer only briefly to this form of algorithmic parallelism in the following, concentrating instead on the solution of the discretized subproblems. Even here, more than one level of parallelism can exist—for example, if the discretized problem is sparse. We discuss sparsity exploitation in Chapters 6 and 7.

Our main algorithmic paradigm for exploiting both vectorization and parallelism in the sparse and the full case is the use of block algorithms, particularly in conjunction with highly tuned kernels for effecting matrix-vector and matrix-matrix operations. We discuss the design of these building blocks in Section 5.1 and their use in the solution of dense equations in the rest of Chapter 5. We discuss their use in the solution of sparse systems in Chapter 6, particularly Sections 6.4 and 6.5.

As we said in the preface, this book is intended to serve as a reference and as a supplementary teaching text for graduate students, researchers working in computational science, and numerical analysts. At the very least, the book should provide background, definitions, and basic techniques so that researchers can understand and exploit the new generation of computers with greater facility and efficiency.

Contents

Chapter 1

High-Performance Computing

In this chapter we review some of the basic features of traditional and advanced computers. The review is not intended to be a complete discussion of the architecture of any particular machine or a detailed analysis of computer architectures. Rather, our focus is on certain features that are especially relevant to the implementation of linear algebra algorithms.

1.1 Trends in Computer Design

In the past decade, the world has experienced one of the most exciting periods in computer development. Computer performance improvements have been dramatic—a trend that promises to continue for the next several years. One reason for the improved performance is the rapid advance in microprocessor technology. Microprocessors have become smaller, denser, and more powerful. Indeed, if cars had made equal progress, you could buy a car for a few dollars, drive it across the country in a few minutes, and "park" the car in your pocket! The result is that microprocessor-based supercomputing is rapidly becoming the technology of preference in attacking some of the most important problems of science and engineering. To exploit microprocessor technology, vendors have developed *highly parallel computers.*

Highly parallel systems offer the enormous computational power needed for solving some of our most challenging computational problems such as simulating the climate. Unfortunately, software development has not kept pace with hardware advances. New programming paradigms, languages, scheduling and partitioning techniques, and algorithms are needed to fully exploit the power of these highly parallel machines.

A major new trend for scientific problem solving is *distributed computing.* In distributed computing, computers connected by a network are used collectively to solve a single large problem. Many scientists are discovering that their