国外数学名著系列（续一）

（影印版）39

Yousef Saad

Iterative Methods for Sparse Linear Systems

Second Edition

稀疏线性系统的迭代方法

（第二版）

科学出版社
www.sciencep.com

国外数学名著系列（影印版） 39

Iterative Methods for Sparse Linear Systems

Second Edition

稀疏线性系统的迭代方法

（第二版）

Yousef Saad

科学出版社

北京

图字：01-2008-5107

Original American edition published by:
SIAM: Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania

图书在版编目 (CIP) 数据

稀疏线性系统的迭代方法：第2版＝Iterative Methods for Sparse Linear Systems (Second Edition) / (美)萨阿德(Saad, Y.)著. —影印版. —北京：科学出版社，2009

(国外数学名著系列；39)

ISBN 978-7-03-023483-4

I. 稀… II. 萨… III. 稀疏矩阵-迭代法-英文 IV. O241.6

中国版本图书馆 CIP 数据核字(2008) 第 186200 号

责任编辑：范庆奎/责任印刷：钱玉芬/封面设计：黄华斌

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

中国科学院印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2009 年 1 月第 一 版 开本：B5(720 × 1000)
2009 年 1 月第一次印刷 印张：34 1/2
印数：1—2 000 字数：665 000

定价：98.00 元

(如有印装质量问题，我社负责调换〈科印〉)

Preface to the Second Edition

In the six years that have passed since the publication of the first edition of this book, iterative methods for linear systems have made good progress in scientific and engineering disciplines. This is due in great part to the increased complexity and size of the new generation of linear and nonlinear systems that arise from typical applications. At the same time, parallel computing has penetrated the same application areas, as inexpensive computer power has become broadly available and standard communication languages such as MPI have provided a much needed standardization. This has created an incentive to utilize iterative rather than direct solvers, because the problems solved are typically from three-dimensional models for which direct solvers often become ineffective. Another incentive is that iterative methods are far easier to implement on parallel computers.

Although iterative methods for linear systems have seen a significant maturation, there are still many open problems. In particular, it still cannot be stated that an arbitrary sparse linear system can be solved iteratively in an efficient way. If physical information about the problem can be exploited, more effective and robust methods can be tailored to the solutions. This strategy is exploited by multigrid methods. In addition, parallel computers necessitate different ways of approaching the problem and solution algorithms that are radically different from classical ones.

Several new texts on the subject of this book have appeared since the first edition. Among these are the books by Greenbaum [154] and Meurant [208]. The exhaustive five-volume treatise by G. W. Stewart [273] is likely to become the de facto reference in numerical linear algebra in years to come. The related multigrid literature has also benefited from a few notable additions, including a new edition of the excellent *Multigrid Tutorial* [65] and a new title by Trottenberg et al. [285].

Most notable among the changes from the first edition is the addition of a sorely needed chapter on multigrid techniques. The chapters that have seen the biggest changes are Chapters 3, 6, 10, and 12. In most cases, the modifications were made to update the material by adding topics that have been developed recently or gained importance in the last few years. In some instances some of the older topics were removed or shortened. For example, the discussion on parallel architecture has been shortened. In the mid-1990s hypercubes and "fat-trees" were important topics to teach. This is no longer the case, since manufacturers have taken steps to hide the topology from the user, in the sense that communication has become much less sensitive to the underlying architecture.

The bibliography has been updated to include work that has appeared in the last few years, as well as to reflect the change of emphasis when new topics have gained importance. Similarly, keeping in mind the educational side of this book, many new exercises have been added. The first edition suffered from many typographical errors, which have been corrected. Many thanks to those readers who took the time to point out errors.

I would like to reiterate my thanks to all my colleagues who helped make the first edition a success (see the preface to the first edition). I received support and encouragement from many students and colleagues to put together this revised volume. I also wish to thank those who proofread this book. I found that one of the best ways to improve clarity is to solicit comments and questions from students in a course that teaches the material. Thanks to all students in CSci 8314 who helped in this regard. Special thanks to Bernie Sheeham, who pointed out quite a few typographical errors and made numerous helpful suggestions.

My sincere thanks to Michele Benzi, Howard Elman, and Steve McCormick for their reviews of this edition. Michele proofread a few chapters thoroughly and caught a few misstatements. Steve's review of Chapter 13 helped ensure that my slight bias for Krylov methods (versus multigrid) was not too obvious. His comments were the origin of the addition of Section 13.7 (Multigrid versus Krylov Methods).

Finally, I would also like to express my appreciation to all SIAM staff members who handled this book, especially Linda Thiel and Lisa Briggeman.

Suggestions for Teaching

This book can be used as a text to teach a graduate-level course on iterative methods for linear systems. Selecting topics to teach depends on whether the course is taught in a mathematics department or a computer science (or engineering) department, and whether the course is over a semester or a quarter. Here are a few comments on the relevance of the topics in each chapter.

For a graduate course in a mathematics department, much of the material in Chapter 1 should be known already. For nonmathematics majors, most of the chapter must be covered or reviewed to acquire a good background for later chapters. The important topics for the rest of the book are in Sections 1.8.1, 1.8.3, 1.8.4, 1.9, and 1.11. Section 1.12 is best treated at the beginning of Chapter 5. Chapter 2 is essentially independent of the rest and could be skipped altogether in a quarter session, unless multigrid methods are to be included in the course. One lecture on finite differences and the resulting matrices would be enough for a nonmath course. Chapter 3 aims at familiarizing the student with some implementation issues associated with iterative solution procedures for general sparse matrices. In a computer science or engineering department, this can be very relevant. For mathematicians, a mention of the graph theory aspects of sparse matrices and a few storage schemes may be sufficient. Most students at this level should be familiar with a few of the elementary relaxation techniques covered in Chapter 4. The convergence theory can be skipped for nonmath majors. These methods are now often used as preconditioners, which may be the only motive for covering them.

Chapter 5 introduces key concepts and presents projection techniques in general terms. Nonmathematicians may wish to skip Section 5.2.3. Otherwise, it is recommended to start the theory section by going back to Section 1.12 on general definitions of projectors. Chapters 6 and 7 represent the heart of the matter. It is recommended to describe the first algorithms carefully and emphasize the fact that they generalize the one-dimensional methods covered in Chapter 5. It is also important to stress the optimality properties of those methods in Chapter 6 and the fact that these follow immediately from the properties of projectors seen in Section 1.12. Chapter 6 is rather long and the instructor will need to select what to cover among the nonessential topics as well as choose topics for reading.

When covering the algorithms in Chapter 7, it is crucial to point out the main differences between them and those seen in Chapter 6. The variants such as conjugate gradient squared (CGS), biconjugate gradient stabilized (BICGSTAB), and transpose-free quasi-minimal residual (TFQMR) can be covered in a short time, omitting details of the algebraic derivations or covering only one of the three. The class of methods based on the normal equations approach, i.e., Chapter 8, can be skipped in a math-oriented course, especially in the case of a quarter session. For a semester course, selected topics may be Sections 8.1, 8.2, and 8.4.

Preconditioning is known as the determining ingredient in the success of iterative methods in solving real-life problems. Therefore, at least some parts of Chapters 9 and 10 should be covered. Sections 9.2 and (very briefly) 9.3 are recommended. From Chapter 10, discuss the basic ideas in Sections 10.1 through 10.3. The rest could be skipped in a quarter course.

Chapter 11 may be useful to present to computer science majors, but may be skimmed through or skipped in a mathematics or an engineering course. Parts of Chapter 12 could be taught primarily to make the students aware of the importance of alternative preconditioners. Suggested selections are Sections 12.2, 12.4, and 12.7.2 (for engineers).

Chapters 13 and 14 present important research areas and are primarily geared toward mathematics majors. Computer scientists or engineers may cover this material in less detail.

To make these suggestions more specific, the following two tables are offered as sample course outlines. Numbers refer to sections in the text. A semester course represents approximately 30 lectures of 75 minutes each whereas a quarter course is approximately 20 lectures of 75 minutes each. Different topics are selected for a mathematics course and a nonmathematics course.

Semester Course		
Weeks	Mathematics	Computer Science/Eng.
1–3	1.9–1.13 2.1–2.5 3.1–3.3	1.1–1.6 (Read); 1.7; 1.9; 1.11; 1.12; 2.1–2.2 3.1–3.6
4–6	4.1–4.2 5. 1–5.3; 6.1–6.4 6.5.1; 6.5.3–6.5.9	4.1–4.2.1; 4.2.3 5.1–5.2.1; 5.3 6.1–6.4; 6.5.1–6.5.5
7–9	6.6–6.8 6.9–6.11; 7.1–7.3 7.4.1; 7.4.2; 7.4.3 (Read)	6.7.1; 6.8–6.9 6.11.3; 7.1–7.3 7.4.1–7.4.2; 7.4.3 (Read)
10–12	8.1; 8.2; 9.1–9.4 10.1–10.3; 10.4.1 10.5.1–10.5.7	8.1–8.3; 9.1–9.3 10.1–10.3; 10.4.1–10.4.3 10.5.1–10.5.4; 10.5.7
13–15	12.2–12.4 13.1–13.5 14.1–14.6	11.1–11.4 (Read); 11.5–11.6 12.1–12.2; 12.4–12.7 14.1–14.3; 14.6

Quarter Course		
Weeks	Mathematics	Computer Science/Eng.
1–2	1.9–1.13 4.1–4.2; 5.1–5.4	1.1–1.6 (Read); 3.1–3.5 4.1; 1.12 (Read)
3–4	6.1–6.4 6.5.1; 6.5.3–6.5.5	5.1–5.2.1; 5.3 6.1–6.3
5–6	6.7.1; 6.11.3; 7.1–7.3 7.4.1–7.4.2; 7.4.3 (Read)	6.4; 6.5.1; 6.5.3–6.5.5 6.7.1; 6.11.3; 7.1–7.3
7–8	9.1–9.3 10.1–10.3; 10.5.1; 10.5.7	7.4.1–7.4.2 (Read); 9.1–9.3 10.1–10.3; 10.5.1; 10.5.7
9–10	13.1–13.5 14.1–14.4	11.1–11.4 (Read); 11.5; 11.6 12.1–12.2; 12.4–12.7

Preface to the First Edition

Iterative methods for solving general, large, sparse linear systems have been gaining popularity in many areas of scientific computing. Until recently, direct solution methods were often preferred over iterative methods in real applications because of their robustness and predictable behavior. However, a number of efficient iterative solvers were discovered and the increased need for solving very large linear systems triggered a noticeable and rapid shift toward iterative techniques in many applications.

This trend can be traced back to the 1960s and 1970s, when two important developments revolutionized solution methods for large linear systems. First was the realization that one can take advantage of *sparsity* to design special direct methods that can be quite economical. Initiated by electrical engineers, these *direct sparse solution methods* led to the development of reliable and efficient general-purpose direct solution software codes over the next three decades. Second was the emergence of preconditioned conjugate gradient–like methods for solving linear systems. It was found that the combination of preconditioning and Krylov subspace iterations could provide efficient and simple *general-purpose* procedures that could compete with direct solvers. Preconditioning involves exploiting ideas from sparse direct solvers. Gradually, iterative methods started to approach the quality of direct solvers. In earlier times, iterative methods were often special purpose in nature. They were developed with certain applications in mind and their efficiency relied on many problem-dependent parameters.

Now three-dimensional models are commonplace and iterative methods are almost mandatory. The memory and the computational requirements for solving three-dimensional partial differential equations, or two-dimensional ones involving many degrees of freedom per point, may seriously challenge the most efficient direct solvers available today. Also, iterative methods are gaining ground because they are easier to implement efficiently on high performance computers than direct methods.

My intention in writing this volume is to provide up-to-date coverage of iterative methods for solving large sparse linear systems. I focused the book on practical methods that work for general sparse matrices rather than for any specific class of problems. It is indeed becoming important to embrace applications not necessarily governed by partial differential equations, as these applications are on the rise. Apart from two recent volumes by Axelsson [14] and Hackbusch [163], few books on iterative methods have appeared since the excellent ones by Varga [292] and later Young [321]. Since then, researchers and practitioners have achieved remarkable progress in the development and use of effective iterative methods. Unfortunately, fewer elegant results have been discovered since the 1950s and 1960s. The field has moved in other directions. Methods have gained not only in efficiency but also in robustness and in generality. The traditional techniques, which required rather complicated procedures to determine optimal acceleration parameters, have yielded to the parameter-free conjugate gradient class of methods.

The primary aim of this book is to describe some of the best techniques available today, from both preconditioners and accelerators. One of the secondary aims of the book is to provide a good mix of theory and practice. It also addresses some of the current research issues, such as parallel implementations and robust preconditioners. The emphasis is on Krylov subspace methods, currently the most practical and common group of techniques used in applications. Although there is a tutorial chapter that covers the discretization of partial differential equations, the book is not biased toward any specific application area. Instead, the matrices are assumed to be general sparse and possibly irregularly structured.

The book has been structured in four distinct parts. The first part, Chapters 1 to 4, presents the basic tools. The second part, Chapters 5 to 8, presents projection methods and Krylov subspace techniques. The third part, Chapters 9 and 10, discusses preconditioning. The fourth part, Chapters 11 to 13, discusses parallel implementations and parallel algorithms.

Acknowledgments

I am grateful to a number of colleagues who proofread or reviewed different versions of the manuscript. Among them are Randy Bramley (University of Indiana at Bloomington), Xiao-Chuan Cai (University of Colorado at Boulder), Tony Chan (University of California at Los Angeles), Jane Cullum (IBM, Yorktown Heights), Alan Edelman (Massachusetts Institute of Technology), Paul Fischer (Brown University), David Keyes (Old Dominion University), Beresford Parlett (University of California at Berkeley), and Shang-Hua Teng (University of Minnesota). Their numerous comments and corrections and their encouragement were a highly appreciated contribution. In particular, they helped improve the presentation considerably and prompted the addition of a number of topics missing from earlier versions.

This book evolved from several successive improvements of a set of lecture notes for the course "Iterative Methods for Linear Systems," which I taught at the University of Minnesota in the last few years. I apologize to those students who used the earlier error-laden and incomplete manuscripts. Their input and criticism contributed significantly to improving the manuscript. I also wish to thank those students at MIT (with Alan Edelman) and UCLA (with Tony Chan) who used this book in manuscript form and provided helpful feedback. My colleagues at the University of Minnesota, staff and faculty members, have helped in different ways. I wish to thank in particular Ahmed Sameh for his encouragement and for fostering a productive environment in the department. Finally, I am grateful to the National Science Foundation for its continued financial support of my research, part of which is represented in this work.

Yousef Saad

Contents