

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)

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

Copyright © 2003. All rights reserved

This edition is authorized for sale only in: Mainland China excluding Hong
Kong, Macau and Taiwan

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

tioners 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.

Contents

Preface to the Second Edition	xiii
Preface to the First Edition	xvii
1 Background in Linear Algebra	1
1.1 Matrices	1
1.2 Square Matrices and Eigenvalues	2
1.3 Types of Matrices	4
1.4 Vector Inner Products and Norms	5
1.5 Matrix Norms	7
1.6 Subspaces, Range, and Kernel	9
1.7 Orthogonal Vectors and Subspaces	10
1.8 Canonical Forms of Matrices	14
1.8.1 Reduction to the Diagonal Form	15
1.8.2 The Jordan Canonical Form	15
1.8.3 The Schur Canonical Form	16
1.8.4 Application to Powers of Matrices	18
1.9 Normal and Hermitian Matrices	20
1.9.1 Normal Matrices	20
1.9.2 Hermitian Matrices	23
1.10 Nonnegative Matrices, M -Matrices	25
1.11 Positive Definite Matrices	29
1.12 Projection Operators	32
1.12.1 Range and Null Space of a Projector	32
1.12.2 Matrix Representations	34
1.12.3 Orthogonal and Oblique Projectors	34
1.12.4 Properties of Orthogonal Projectors	36
1.13 Basic Concepts in Linear Systems	37
1.13.1 Existence of a Solution	37
1.13.2 Perturbation Analysis	38
Exercises	39
Notes and References	43

2	Discretization of Partial Differential Equations	45
2.1	Partial Differential Equations	45
2.1.1	Elliptic Operators	46
2.1.2	The Convection Diffusion Equation	48
2.2	Finite Difference Methods	48
2.2.1	Basic Approximations	48
2.2.2	Difference Schemes for the Laplacian Operator	50
2.2.3	Finite Differences for One-Dimensional Problems	51
2.2.4	Upwind Schemes	52
2.2.5	Finite Differences for Two-Dimensional Problems	54
2.2.6	Fast Poisson Solvers	55
2.3	The Finite Element Method	60
2.4	Mesh Generation and Refinement	66
2.5	Finite Volume Method	68
	Exercises	71
	Notes and References	72
3	Sparse Matrices	73
3.1	Introduction	73
3.2	Graph Representations	75
3.2.1	Graphs and Adjacency Graphs	75
3.2.2	Graphs of PDE Matrices	76
3.3	Permutations and Reorderings	77
3.3.1	Basic Concepts	77
3.3.2	Relations with the Adjacency Graph	79
3.3.3	Common Reorderings	81
3.3.4	Irreducibility	89
3.4	Storage Schemes	89
3.5	Basic Sparse Matrix Operations	92
3.6	Sparse Direct Solution Methods	93
3.6.1	MD Ordering	93
3.6.2	ND Ordering	94
3.7	Test Problems	94
	Exercises	97
	Notes and References	101
4	Basic Iterative Methods	103
4.1	Jacobi, Gauss–Seidel, and Successive Overrelaxation	103
4.1.1	Block Relaxation Schemes	106
4.1.2	Iteration Matrices and Preconditioning	110
4.2	Convergence	111
4.2.1	General Convergence Result	112
4.2.2	Regular Splittings	115
4.2.3	Diagonally Dominant Matrices	116
4.2.4	Symmetric Positive Definite Matrices	119
4.2.5	Property A and Consistent Orderings	119

4.3	Alternating Direction Methods	124
	Exercises	126
	Notes and References	128
5	Projection Methods	129
5.1	Basic Definitions and Algorithms	129
5.1.1	General Projection Methods	130
5.1.2	Matrix Representation	131
5.2	General Theory	132
5.2.1	Two Optimality Results	133
5.2.2	Interpretation in Terms of Projectors	134
5.2.3	General Error Bound	135
5.3	One-Dimensional Projection Processes	137
5.3.1	Steepest Descent	138
5.3.2	MR Iteration	140
5.3.3	Residual Norm Steepest Descent	142
5.4	Additive and Multiplicative Processes	143
	Exercises	145
	Notes and References	149
6	Krylov Subspace Methods, Part I	151
6.1	Introduction	151
6.2	Krylov Subspaces	152
6.3	Arnoldi's Method	153
6.3.1	The Basic Algorithm	154
6.3.2	Practical Implementations	156
6.4	Arnoldi's Method for Linear Systems	159
6.4.1	Variation 1: Restarted FOM	160
6.4.2	Variation 2: IOM and DIOM	161
6.5	Generalized Minimal Residual Method	164
6.5.1	The Basic GMRES Algorithm	164
6.5.2	The Householder Version	165
6.5.3	Practical Implementation Issues	167
6.5.4	Breakdown of GMRES	171
6.5.5	Variation 1: Restarting	171
6.5.6	Variation 2: Truncated GMRES Versions	172
6.5.7	Relations Between FOM and GMRES	177
6.5.8	Residual Smoothing	181
6.5.9	GMRES for Complex Systems	184
6.6	The Symmetric Lanczos Algorithm	185
6.6.1	The Algorithm	185
6.6.2	Relation to Orthogonal Polynomials	186
6.7	The Conjugate Gradient Algorithm	187
6.7.1	Derivation and Theory	187
6.7.2	Alternative Formulations	191
6.7.3	Eigenvalue Estimates from the CG Coefficients	192

6.8	The Conjugate Residual Method	194
6.9	Generalized Conjugate Residual, ORTHOMIN, and ORTHODIR . . .	194
6.10	The Faber–Manteuffel Theorem	196
6.11	Convergence Analysis	198
6.11.1	Real Chebyshev Polynomials	199
6.11.2	Complex Chebyshev Polynomials	200
6.11.3	Convergence of the CG Algorithm	203
6.11.4	Convergence of GMRES	205
6.12	Block Krylov Methods	208
	Exercises	212
	Notes and References	215
7	Krylov Subspace Methods, Part II	217
7.1	Lanczos Biorthogonalization	217
7.1.1	The Algorithm	217
7.1.2	Practical Implementations	220
7.2	The Lanczos Algorithm for Linear Systems	221
7.3	The Biconjugate Gradient and Quasi-Minimal Residual Algorithms . .	222
7.3.1	The BCG Algorithm	222
7.3.2	QMR Algorithm	224
7.4	Transpose-Free Variants	228
7.4.1	CGS	229
7.4.2	BICGSTAB	231
7.4.3	TFQMR	234
	Exercises	241
	Notes and References	243
8	Methods Related to the Normal Equations	245
8.1	The Normal Equations	245
8.2	Row Projection Methods	247
8.2.1	Gauss–Seidel on the Normal Equations	247
8.2.2	Cimmino’s Method	249
8.3	Conjugate Gradient and Normal Equations	251
8.3.1	CGNR	252
8.3.2	CGNE	253
8.4	Saddle-Point Problems	254
	Exercises	257
	Notes and References	259
9	Preconditioned Iterations	261
9.1	Introduction	261
9.2	Preconditioned Conjugate Gradient	262
9.2.1	Preserving Symmetry	262
9.2.2	Efficient Implementations	265

9.3	Preconditioned Generalized Minimal Residual	267
9.3.1	Left-Preconditioned GMRES	268
9.3.2	Right-Preconditioned GMRES	269
9.3.3	Split Preconditioning	270
9.3.4	Comparison of Right and Left Preconditioning	271
9.4	Flexible Variants	272
9.4.1	FGMRES	273
9.4.2	DQGMRES	276
9.5	Preconditioned Conjugate Gradient for the Normal Equations	276
9.6	The Concus, Golub, and Widlund Algorithm	278
	Exercises	279
	Notes and References	280
10	Preconditioning Techniques	283
10.1	Introduction	283
10.2	Jacobi, Successive Overrelaxation, and Symmetric Successive Overrelaxation Preconditioners	284
10.3	Incomplete LU Factorization Preconditioners	287
10.3.1	ILU Factorizations	288
10.3.2	Zero Fill-in ILU (ILU(0))	293
10.3.3	Level of Fill and ILU(p)	296
10.3.4	Matrices with Regular Structure	300
10.3.5	MILU	305
10.4	Threshold Strategies and Incomplete LU with Threshold	306
10.4.1	The ILUT Approach	307
10.4.2	Analysis	308
10.4.3	Implementation Details	310
10.4.4	The ILUTP Approach	312
10.4.5	The ILUS Approach	314
10.4.6	The ILUC Approach	316
10.5	Approximate Inverse Preconditioners	320
10.5.1	Approximating the Inverse of a Sparse Matrix	321
10.5.2	Global Iteration	321
10.5.3	Column-Oriented Algorithms	323
10.5.4	Theoretical Considerations	324
10.5.5	Convergence of Self-Preconditioned MR	326
10.5.6	AINVs via Bordering	329
10.5.7	Factored Inverses via Orthogonalization: AINV	331
10.5.8	Improving a Preconditioner	333
10.6	Reordering for Incomplete LU	333
10.6.1	Symmetric Permutations	333
10.6.2	Nonsymmetric Reorderings	335
10.7	Block Preconditioners	337
10.7.1	Block Tridiagonal Matrices	337
10.7.2	General Matrices	339

10.8	Preconditioners for the Normal Equations	339
10.8.1	Jacobi, SOR, and Variants	340
10.8.2	IC(0) for the Normal Equations	340
10.8.3	Incomplete Gram–Schmidt and ILQ	342
	Exercises	345
	Notes and References	349
11	Parallel Implementations	353
11.1	Introduction	353
11.2	Forms of Parallelism	354
11.2.1	Multiple Functional Units	354
11.2.2	Pipelining	354
11.2.3	Vector Processors	355
11.2.4	Multiprocessing and Distributed Computing	355
11.3	Types of Parallel Architectures	355
11.3.1	Shared Memory Computers	356
11.3.2	Distributed Memory Architectures	357
11.4	Types of Operations	359
11.5	Matrix-by-Vector Products	361
11.5.1	The CSR and CSC Formats	362
11.5.2	Matvecs in the Diagonal Format	364
11.5.3	The Ellpack-Itpack Format	364
11.5.4	The JAD Format	365
11.5.5	The Case of Distributed Sparse Matrices	366
11.6	Standard Preconditioning Operations	369
11.6.1	Parallelism in Forward Sweeps	369
11.6.2	Level Scheduling: The Case of Five-Point Matrices	370
11.6.3	Level Scheduling for Irregular Graphs	370
	Exercises	373
	Notes and References	375
12	Parallel Preconditioners	377
12.1	Introduction	377
12.2	Block Jacobi Preconditioners	378
12.3	Polynomial Preconditioners	379
12.3.1	Neumann Polynomials	380
12.3.2	Chebyshev Polynomials	381
12.3.3	Least-Squares Polynomials	383
12.3.4	The Nonsymmetric Case	386
12.4	Multicoloring	389
12.4.1	Red-Black Ordering	389
12.4.2	Solution of Red-Black Systems	390
12.4.3	Multicoloring for General Sparse Matrices	391
12.5	Multi-Elimination Incomplete LU	392
12.5.1	Multi-Elimination	393
12.5.2	ILUM	394

12.6	Distributed Incomplete LU and Symmetric Successive Overrelaxation	396
12.7	Other Techniques	399
12.7.1	AINVs	399
12.7.2	EBE Techniques	399
12.7.3	Parallel Row Projection Preconditioners	401
	Exercises	402
	Notes and References	404
13	Multigrid Methods	407
13.1	Introduction	407
13.2	Matrices and Spectra of Model Problems	408
13.2.1	The Richardson Iteration	412
13.2.2	Weighted Jacobi Iteration	414
13.2.3	Gauss–Seidel Iteration	416
13.3	Intergrid Operations	419
13.3.1	Prolongation	419
13.3.2	Restriction	421
13.4	Standard Multigrid Techniques	422
13.4.1	Coarse Problems and Smoothers	423
13.4.2	Two-Grid Cycles	424
13.4.3	V-Cycles and W-Cycles	426
13.4.4	FMG	429
13.5	Analysis of the Two-Grid Cycle	433
13.5.1	Two Important Subspaces	433
13.5.2	Convergence Analysis	435
13.6	Algebraic Multigrid	437
13.6.1	Smoothness in AMG	438
13.6.2	Interpolation in AMG	439
13.6.3	Defining Coarse Spaces in AMG	442
13.6.4	AMG via Multilevel ILU	442
13.7	Multigrid versus Krylov Methods	445
	Exercises	446
	Notes and References	449
14	Domain Decomposition Methods	451
14.1	Introduction	451
14.1.1	Notation	452
14.1.2	Types of Partitionings	453
14.1.3	Types of Techniques	453
14.2	Direct Solution and the Schur Complement	456
14.2.1	Block Gaussian Elimination	456
14.2.2	Properties of the Schur Complement	457
14.2.3	Schur Complement for Vertex-Based Partitionings	458
14.2.4	Schur Complement for Finite Element Partitionings	460
14.2.5	Schur Complement for the Model Problem	463

14.3	Schwarz Alternating Procedures	465
14.3.1	Multiplicative Schwarz Procedure	465
14.3.2	Multiplicative Schwarz Preconditioning	470
14.3.3	Additive Schwarz Procedure	472
14.3.4	Convergence	473
14.4	Schur Complement Approaches	477
14.4.1	Induced Preconditioners	478
14.4.2	Probing	480
14.4.3	Preconditioning Vertex-Based Schur Complements	480
14.5	Full Matrix Methods	481
14.6	Graph Partitioning	483
14.6.1	Basic Definitions	484
14.6.2	Geometric Approach	484
14.6.3	Spectral Techniques	486
14.6.4	Graph Theory Techniques	487
	Exercises	491
	Notes and References	492
	Bibliography	495
	Index	517