CAMBRIDGE MONOGRAPHS ON
APPLIED AND COMPUTATIONAL
MATHEMATICS

19

Matrix Preconditioning Techniques and Applications

The *Cambridge Monographs on Applied and Computational Mathematics* reflects the crucial role of mathematical and computational techniques in contemporary science. The series publishes expositions on all aspects of applicable and numerical mathematics, with an emphasis on new developments in this fast-moving area of research.

State-of-the-art methods and algorithms as well as modern mathematical descriptions of physical and mechanical ideas are presented in a manner suited to graduate research students and professionals alike. Sound pedagogical presentation is a prerequisite. It is intended that books in the series will serve to inform a new generation of researchers.

Also in this series:

1. A Practical Guide to Pseudospectral Methods, *Bengt Fornberg*
2. Dynamical Systems and Numerical Analysis, *A. M. Stuart and A. R. Humphries*
3. Level Set Methods and Fast Marching Methods, *J. A. Sethian*
4. The Numerical Solution of Integral Equations of the Second Kind, *Kendall E. Atkinson*
5. Orthogonal Rational Functions, *Adhemar Bultheel, Pablo González-Vera, Erik Hendiksen, and Olav Njåstad*
6. The Theory of Composites, *Graeme W. Milton*
7. Geometry and Topology for Mesh Generation *Herbert Edelsbrunner*
8. Schwarz–Christoffel Mapping *Tofin A. Driscoll and Lloyd N. Trefethen*
9. High-Order Methods for Incompressible Fluid Flow, *M. O. Deville, P. F. Fischer and E. H. Mund*
10. Practical Extrapolation Methods, *Avram Sidi*
11. Generalized Riemann Problems in Computational Fluid Dynamics, *Matania Ben-Artzi and Joseph Falcovitz*
12. Radial Basis Functions: Theory and Implementations, *Martin Buhmann*
13. Iterative Krylov Methods for Large Linear Systems, *Henk A. van der Vorst*
14. Simulating Hamiltonian Dynamics, *Benedict Leimkuhler and Sebastian Reich*
15. Collocation Methods for Volterra Integral and Related Functional Equations, *Hermann Brunner*
16. Topology for Computing, *Afra J. Zomorodian*
17. Scattered Data Approximation, *Holger Wendland*

Matrix Preconditioning Techniques and Applications

KE CHEN
Reader in Mathematics
Department of Mathematical Sciences
The University of Liverpool

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 2RU, UK

www.cambridge.org
Information on this title: www.cambridge.org/9780521838283

First published 2005

Printed in the United Kingdom at the University Press, Cambridge

A catalogue record for this book is available from the British Library

ISBN-13 978-0-521-83828-3 hardback
ISBN-10 0-521-83828-2 hardback

Dedicated to
Zhuang and Leo Ling Yi
and the loving memories of my late parents Wan-Qing and Wen-Fang

In deciding what to investigate, how to formulate ideas and what problems to focus on, the individual mathematician has to be guided ultimately by their own sense of values. There are no clear rules, or rather if you only follow old rules you do not create anything worthwhile.

Sir Michael Atiyah (FRS, Fields Medallist 1966). What's it all about? *UK EPSRC Newsline Journal – Mathematics* (2001)

Contents

Preface

> The experiences of Fox, Huskey, and Wilkinson [from solving systems of orders up to 20] prompted Turing to write a remarkable paper [in 1948]... In this paper, Turing made several important contributions... He used the word "*preconditioning*" to mean improving the condition of a system of linear equations (a term that did not come into popular use until 1970s).
>
> NICHOLAS J. HIGHAM. *Accuracy and Stability of Numerical Algorithms.* SIAM Publications (1996)

Matrix computing arises in the solution of almost all linear and nonlinear systems of equations. As the computer power upsurges and high resolution simulations are attempted, a method can reach its applicability limits quickly and hence there is a constant demand for new and fast matrix solvers. Preconditioning is the key to a successful iterative solver. It is the intention of this book to present a comprehensive exposition of the many useful preconditioning techniques.

Preconditioning equations mainly serve for an iterative method and are often solved by a direct solver (occasionally by another iterative solver). Therefore it is inevitable to address direct solution techniques for both sparse and dense matrices. While fast solvers are frequently associated with iterative solvers, for special problems, a direct solver can be competitive. Moreover, there are situations where preconditioning is also needed for a direct solution method. This clearly demonstrates the close relationship between a direct and an iterative method.

This book is the first of its kind attempting to address an active research topic, covering these main types of preconditioners.

Type 1	Matrix splitting preconditioner	FEM setting
Type 2	Approximate inverse preconditioner	FEM setting
Type 3	Multilevel (approximate inverse) preconditioner	FEM setting
Type 4	Recursive Schur complements preconditioner	FEM setting
Type 5	Matrix splitting and Approximate inverses	Wavelet setting
Type 6	Recursive Schur complements preconditioner	Wavelet setting
Type 7	Implicit wavelet preconditioner	FEM setting

Here by 'FEM setting', we mean a usual matrix (as we found it) often formed from discretization by finite element methods (FEM) for partial differential equations with piecewise polynomial basis functions whilst the 'Wavelet setting' refers to wavelet discretizations. The iterative solvers, often called accelerators, are selected to assist and motivate preconditioning. As we believe that suitable preconditioners can work with most accelerators, many other variants of accelerators are only briefly mentioned to allow us a better focus on the main theme. However these accelerators are well documented in whole or in part in the more recent as well as the more classical survey books or monographs (to name only a few)

- Young, D. M. (1971). *Iterative Solution of Large Linear Systems*. Academic Press.
- Hageman A. L. and Young D. M. (1981). *Applied Iterative Methods*. Academic Press.
- McCormick S. F. (1992). *Multilevel Projection Methods for Partial Differential Equations*. SIAM Publications.
- Barrett R., *et al*. (1993). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM Publications.
- Axelsson O. (1994). *Iterative Solution Methods*. Cambridge University Press (reprinted by SIAM Publications in 2001)
- Hackbusch W. (1994). *Iterative Solution of Large Sparse Systems*. Springer-Verlag.
- Kelly C. T. (1995). *Iterative Methods for Solving Linear and Nonlinear Equations*. SIAM Publications.
- Smith B., *et al*. (1996). *Domain Decomposition Methods*. Cambridge University Press.
- Golub G. and van Loan C. (1996). *Matrix Computations*, 3rd edn. Johns Hopkins University Press.
- Brezinski C. (1997). *Projection Methods for Systems of Equations*. North-Holland.
- Demmel J. (1997). *Applied Numerical Linear Algebra*. SIAM Publications.

- Greenbaum A. (1997). *Iterative Methods for Solving Linear Systems*. SIAM Publications.
- Trefethen N. and Bau D. (1997). *Numerical Linear Algebra*. SIAM Publications.
- Dongarra J., *et al*. (1998). *Numerical Linear Algebra on High-Performance Computers*. SIAM Publications.
- Briggs W., *et al*. (2000). *A Multigrid Tutorial*, 2nd edn. SIAM Publications.
- Varga R. (2001). *Matrix Iteration Analysis*, 2nd edn. Springer.
- Higham N. J. (2002). *Accuracy and Stability of Numerical Algorithms*, 2nd edn. SIAM Publications.
- van der Vorst H. A. (2003). *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press
- Saad Y. (2003). *Iterative Methods for Sparse Linear Systems*. PWS.
- Duff I. S., *et al*. (2006). *Direct Methods for Sparse Matrices*, 2nd edn. Clarendon Press.
- Elman H., *et al*. (2005) *Finite Elements and Fast Solvers*. Oxford University Press.

Most generally applicable preconditioning techniques for unsymmetric matrices are covered in this book. More specialized preconditioners, designed for symmetric matrices, are only briefly mentioned; where possible we point to suitable references for details. Our emphasis is placed on a clear exposition of the motivations and techniques of preconditioning, backed up by MATLAB®[1] Mfiles, and theories are only presented or outlined if they help to achieve better understanding. Broadly speaking, the convergence of an iterative solver is dependent of the underlying problem class. The robustness can often be improved by suitably designed preconditioners. In this sense, one might stay with any preferred iterative solver and concentrate on preconditioner designs to achieve better convergence.

As is well known, the idea of providing and sharing software is to enable other colleagues and researchers to concentrate on solving newer and harder problems instead of repeating work already done. In the extreme case, there is nothing more frustrating than not being able to reproduce results that are claimed to have been achieved. The MATLAB Mfiles are designed in a friendly style to reflect the teaching of the author's friend and former MSc advisor Mr Will

[1] MATLAB is a registered trademark of MathWorks, Inc; see its home page `http://www.mathworks.com`. MATLAB is an easy-to-use script language, having almost the full capability as a *C* programming language without the somewhat complicated syntax of *C*. Beginners can consult a MATLAB text e.g. [135] from `http://www.liv.ac.uk/maths/ETC/matbook` or any tutorial document from the internet. Search `http://www.google.com` using the key words: **MATLAB tutorial**.

McLewin (University of Manchester) who rightly said: 'in Mathematics, never use the word 'obviously'.' A simple and useful feature of the supplied Mfiles is that *typing* in the file name invokes the help page, giving working examples. (The standard MATLAB reply to such a usage situation is *??? Error... Not enough input arguments*.)

The book was born mainly out of research work done in recent years and partly out of a need of helping out graduate students to implement a method following the not-always-easy-to-follow descriptions by some authors (who use the words 'trivial', 'standard', 'well-known', 'leave the reader to work it out as an exercise' in a casual way and in critical places). That is to say, we aspire to emphasize the practical implementation as well as the understanding rather than too much of the theory. In particular the book is to attempt a clear presentation and explanation, with the aid of illustrations and computer software, so that the reader can avoid the occasional frustration that *one must know the subject already before one can really understand and appreciate a beautiful mathematical idea or algorithm* presented in some (maybe a lot of) mathematical literature.

► **About the solvers and preconditioners.**

Chapter 1. (Introduction) defines the commonly used concepts; in particular the two most relevant terms in preconditioning: condition number and clustering. With non-mathematics majors' readers in mind, we give an introduction to several discretization and linearization methods which generate matrix equations – the idea of mesh ordering affecting the resulting matrix is first encountered. Examples of bounding conditioned numbers by considering norm equivalence (for symmetric systems) are given; these appealing theories are not a guarantee for fast convergence of iterative solvers. Both the fast Fourier transforms (FFT) and fast wavelet transforms (FWT) are introduced here (mainly discrete FWT and the continuous to come later); further discussions of FFT and FWT are in Chapters 2, 4 and 8.

Chapter 2. (Direct methods) discusses the direct Gaussian elimination method and the Gauss–Jordan and several variants. Direct methods are on one hand necessary for forward type preconditioning steps and on the other hand provide various motivations for designing an effective preconditioner. Likewise, for some ill-conditioned linear systems, there is a strong need for scaling and preconditioning to obtain accurate direct solutions – a much less addressed subject. Algorithms for inverting several useful special matrices are then given; for circulant matrices diagonalization by Fourier transforms is explained before considering Toeplitz matrices. Block Toeplitz matrices are considered later in Chapter 13. Algorithms for graph nodal or mesh (natural graph) orderings by reverse Cuthill–McKee method (RCM), spiral and domain decomposition methods (DDM) are given. The Schur complements and partitioned LU decompositions are presented together; for symmetric positive definite (SPD) matrices, some Schur properties are discussed. Overall, this chapter contains most of the ingredients for implementing a successful preconditioner.

Chapter 3. (Iterative methods) first discusses the classical iterative methods and highlights their use in multigrid methods (MGM, Chapter 6) and DDM. Then we introduce the topics most relevant to the book, conjugate gradient methods (CGM) of the Krylov subspace type (the complex variant algorithm does not appear in the literature as explicitly as presented in Section 3.7). We elaborate on the convergence with a view on preconditioners' design. Finally the popular fast multipole expansion method (along with preconditioning) is introduced. The mission of this chapter is to convey the message that preconditioning is relatively more important than modifying existing or inventing new CGM solvers.

Chapter 4. (Matrix splitting preconditioners: Type 1) presents a class of mainly sparse preconditioners and indicates their possible application areas, algorithms and limitations. All these preconditioners are of the forward type, i.e. $M \approx A$ in some way and efficiency in solving $Mx = b$ is assured. The most effective and general variant is the incomplete LU (ILU) preconditioner with suitable nodal ordering. The last two main sections (especially the last one) are mainly useful for dense matrix applications.

Chapter 5. (Approximate inverse preconditioners: Type 2) presents another large class of sparse approximate inverse preconditioners for a general sparse matrix problem, with band preconditioners suitable for diagonally dominant matrices and near neighbour preconditioners suitable for singular operator equations. All these preconditioners are of the backward type, i.e. $M \approx A^{-1}$ in some way and application of each sparse preconditioner M requires a simple multiplication.

Chapter 6. (Multilevel methods and preconditioners: Type 3) gives an introduction to geometric multigrid methods for partial differential equations (PDEs) and integral equations (IEs) and algebraic multigrid method for sparse linear systems, indicating that for PDEs, in general, smoothing is important but can be difficult while for IEs operator compactness is the key. Finally we discuss multilevel domain decomposition preconditioners for CG methods.

Chapter 7. (Multilevel recursive Schur preconditioners: Type 4) surveys the recent Schur complements based recursive preconditioners where matrix partition can be based on functional space nesting or graph nesting (both geometrically based and algebraically based).

Chapter 8. (Sparse wavelet preconditioners: Type 5) first introduces the continuous wavelets and then considers to how construct preconditioners under the wavelet basis in which an underlying operator is more amenable to approximation by the techniques of Chapters 4–7. Finally we discuss various permutations for the standard wavelet transforms and their use in designing banded arrow (wavelet) preconditioners.

Chapter 9. (Wavelet Schur preconditioners: Type 6) generalizes the Schur preconditioner of Chapter 7 to wavelet discretization. Here we propose to combine the non-standard form with Schur complement ideas to avoid finger-patterned matrices.

Chapter 10. (Implicit wavelet preconditioners: Type 7) presents some recent results that propose to combine the advantages of sparsity of finite elements, sparse approximate inverses and wavelets compression. Effectively the wavelet theory is used to justify the a priori patterns that are needed to enable approximate inverses to be efficient; this strategy is different from Chapter 9 which does not use approximate inverses.

► **About the selected applications.**

Chapter 11. (Application I) discusses the iterative solution of boundary integral equations reformulating the Helmholtz equation in an infinite domain modelling the acoustic scattering problem. We include some recent results on high order formulations to overcome the hyper-singularity. The chapter is concluded with a discussion of the open challenge of modelling high wavenumber problems.

Chapter 12. (Application II) surveys some recent work on preconditioning coupled matrix problems. These include Hermitian and skew-Hermitian splitting, continuous operators based Schur approximations for Oseen problems, the block diagonal approximate inverse preconditioners for a coupled fluid structure interaction problem, and FWT based sparse preconditioners for EHL equations modelling the isothermal (two dependent variables) and thermal (three dependent variables) cases.

Chapter 13. (Application III) surveys some recent results for iterative solution of inverse problems. We take the example of the nonlinear total variation (TV) equation for image restoration using operator splitting and circulant preconditioners. We show some new results based on combining FWT and FFT preconditioners for possibly more robust and faster solution and results on developing nonlinear multigrid methods for optimization. Also discussed is the 'matrix-free' idea of solving an elliptic PDE via an explicit scheme of a parabolic PDE, which is widely used in evolving level set functions for interfaces tracking; the related variational formulation of image segmentation is discussed.

Chapter 14. (Application IV) shows an example from scientific computing that typifies the challenge facing computational mathematics – the bifurcation problem. It comes from studying voltage stability in electrical power transmission systems. We have developed two-level preconditioners (approximate inverses with deflation) for solving the fold bifurcation while the Hopf problem remains an open problem as the problem dimension is 'squared'!

Chapter 15. (Parallel computing) gives a brief introduction to the important subject of parallel computing. Instead of parallelizing many algorithms, we motivate two fundamental issues here: firstly how to implement a parallel algorithm in a step-by-step manner and with complete MPI Fortran programs, and secondly what to consider when adapting a sequential algorithm for parallel computing. We take four relatively simple tasks for discussing the underlying ideas.

The Appendices give some useful background material, for reference purpose, on introductory linear algebra, the Harwell–Boeing data format, a MATLAB tutorial, the supplied Mfiles and Internet resources relevant to this book.

► **Use of the book.** The book should be accessible to graduate students in various scientific computing disciplines who have a basic linear algebra and computing knowledge. It will be useful to researchers and computational practitioners. It is anticipated that the reader can build intuition, gain insight and get enough hands on experience with the discussed methods, using the supplied Mfiles and programs from

http : //www.cambridge.org/9780521838283
http : //www.liv.ac.uk/maths/ETC/mpta

while reading. As a reference for researchers, the book provides a toolkit and with it the reader is expected to experiment with a matrix under consideration and identify the suitable methods first before embarking on serious analysis of a new problem.

► **Acknowledgements.** Last but not least, the author is grateful to many colleagues (including Joan E. Walsh, Siamiak Amini, Michael J. Baines, Tony F. Chan and Gene H. Golub) for their insight, guidance, and encouragement and to all my graduate students and research fellows (with whom he has collaborated) for their commitment and hard work, on topics relating to this book over the years. In particular, Stuart Hawkins and Martyn Hughes have helped and drafted earlier versions of some Mfiles, as individually acknowledged in these files. Several colleagues have expressed encouragement and comments as well as corrected on parts of the first draft of the manuscript – these include David J. Evans, Henk A. van der Vorst, Raymond H. Chan, Tony F. Chan, Gene H. Golub and Yimin Wei; the author thanks them all. Any remaining errors in the book are all mine. The handy author index was produced using the authorindex.sty (which is available from the usual LaTeX sites) as developed by Andreas Wettstein (ISE AG, Zurich, Switzerland); the author thanks him for writing a special script for me to convert bibitem entries to a bibliography style. The CUP editorial staff (especially Dr Ken Blake), the series editors and the series reviewers have been very helpful and supportive. The author thanks them for their professionalism and quality support. The continual funding in recent years by the UK EPSRC and other UK funding bodies for various projects related to this book has been gratefully received.

► **Feedback.** As the book involves an ambitious number of topics with preconditioning connection, inevitably, there might be errors and typos. The author is happy to hear from any readers who could point these out for future editions. Omission is definitely inevitable: to give a sense of depth and width of the subject area, a search on `www.google.com` in April 2004 (using keywords like 'preconditioned iterative' or 'preconditioning') resulted in hits ranging from 19000 to 149000 sites. Nevertheless, suggestions and comments are always welcome. The author is also keen to include more links to suitable software that are readable and helpful to other researchers, and are in the spirit of this book. Many thanks and happy computing.

Ke Chen
Liverpool, September 2004
EMAIL = *k.chen@liverpool.ac.uk*
URL = *http://www.liv.ac.uk/~cmchenke*

Nomenclature

All the beautiful mathematical ideas can be found in Numerical Linear Algebra. However, the subject is better to be enjoyed by researchers than to teach to students as many excellent ideas are often buried in the complicated notation. A researcher must be aware of this fact.

GENE H. GOLUB. Lecture at University of Liverpool (1995)

Throughout the book, capital letters such as A denote a rectangular matrix $m \times n$ (or a square matrix of size n), whose (i, j) entry is denoted by $A(i, j) = a_{ij}$, and small letters such as x, b denote vectors of size n unless stated otherwise i.e. $A \in \mathbb{R}^{m\times n}$ and x, $b \in \mathbb{R}^n$.

Some (common) abbreviations and notations are listed here

$\mathbb{C}^n$	$\rightarrow$	the space of all complex vectors of size n
$\mathbb{R}^n$	$\rightarrow$	the space of all real vectors of size n [note $\mathbb{R}^n \subset \mathbb{C}^n$ and $\mathbb{R}^{n\times n} \subset \mathbb{C}^{n\times n}$)]
$\|A\|$	$\rightarrow$	A norm of matrix A (see §1.5)
$\lvert A\rvert$	$\rightarrow$	The matrix of absolute values of A i.e. $(\lvert A\rvert)_{ij} = \lvert A(i, j)\rvert = \lvert a_{ij}\rvert$.
A^T	$\rightarrow$	The transpose of A i.e. $A^T(i, j) = A(j, i)$. [A is symmetric if $A^T = A$]
A^H	$\rightarrow$	The transpose conjugate for complex A i.e. $A^H(i, j) = \overline{A(j, i)}$. [$A$ is Hermitian if $A^H = A$. Some books write $A^* = A^H$]
$det(A)$	$\rightarrow$	The determinant of matrix A
$\text{diag}(\alpha_j)$	$\rightarrow$	A diagonal matrix made up of scalars α_j
$\lambda(A)$	$\rightarrow$	An eigenvalue of A
$A \oplus B$	$\rightarrow$	The direct sum of orthogonal quantities A, B
$A \odot B$	$\rightarrow$	The biproduct of matrices A, B (Definition 14.3.8)
$A \otimes B$	$\rightarrow$	The tensor product of matrices A, B (Definition 14.3.3)

$\sigma(A)$	$\to$	A singular value of A
$\kappa(A)$	$\to$	The condition number $cond(A)$ of A (in some norm)
$\Lambda(A)$	$\to$	Eigenspectrum for A
$\Lambda_\epsilon(A)$	$\to$	ϵ-Eigenspectrum for A
$\Sigma(A)$	$\to$	Spectrum of singular values of A
$\mathcal{Q}_k$	$\to$	Set of all degree k polynomials with $q(0) = 1$ for $q \in \mathcal{Q}_k$
$\mathcal{W}(A)$	$\to$	Field of values (FoA) spectrum
AINV	$\to$	Approximate inverse [55]
BEM	$\to$	Boundary element method
BIE	$\to$	Boundary integral equation
BCCB	$\to$	Block circulant with circulant blocks
BTTB	$\to$	Block Toeplitz with Toeplitz blocks
BPX	$\to$	Bramble–Pasciak–Xu (preconditioner)
CG	$\to$	Conjugate gradient
CGM	$\to$	CG Method
CGN	$\to$	Conjugate gradient normal method
DBAI	$\to$	Diagonal block approximate inverse (preconditioner)
DDM	$\to$	Domain decomposition method
DFT	$\to$	Discrete Fourier transform
DWT	$\to$	Discrete wavelet transform
FDM	$\to$	Finite difference method
FEM	$\to$	Finite element method
FFT	$\to$	Fast Fourier transform
FFT2	$\to$	Fast Fourier transform in 2D (tensor products)
FMM	$\to$	Fast multipole method
FoV	$\to$	Field of values
FSAI	$\to$	Factorized approximate inverse (preconditioner) [321]
FWT	$\to$	Fast wavelet transform
GMRES	$\to$	Generalised minimal residual method
GJ	$\to$	Gauss–Jordan decomposition
GS	$\to$	Gauss–Seidel iterations (or Gram–Schmidt method)
HB	$\to$	Hierachical basis (finite elements)
ILU	$\to$	Incomplete LU decomposition
LU	$\to$	Lower upper triangular matrix decomposition
LSAI	$\to$	Least squares approximate inverse (preconditioner)
MGM	$\to$	Multigrid method
MRA	$\to$	Multilresolution analysis
OSP	$\to$	Operator splitting preconditioner
PDE	$\to$	Partial differential equation

PSM	$\rightarrow$	Powers of sparse matrices
QR	$\rightarrow$	Orthogonal upper triangular decomposition
SDD	$\rightarrow$	Strictly diagonally dominant
SOR	$\rightarrow$	Successive over-relaxation
SSOR	$\rightarrow$	Symmetric SOR
SPAI	$\rightarrow$	Sparse approximate inverse [253]
SPD	$\rightarrow$	Symmetric positive definite matrix ($\lambda_j(A) > 0$)
SVD	$\rightarrow$	Singular value decomposition
WSPAI	$\rightarrow$	Wavelet SPAI