Series in Information and Computational Science — 35

Numerical Linear Algebra and Its Applications

Xiao-qing Jin and Yi-min Wei

（数值线性代数及其应用）

SCIENCE PRESS
SCIENCE PRESS USA Inc.

Responsible Editors: Liu Jiashan Fan Qingkui

Published by Science Press
16 Donghuangchenggen North Street
Beijing 100717, China

Printed in Beijing

ISBN 7-03-013954-2 (Beijing)

Series in Information and Computational Science 35

Editorial Board

To Our Families

Preface to the Series in Information and Computational Science

Since the 1970s, Science Press has published more than thirty volumes in its series Monographs in Computational Methods. This series was established and led by the late academician, Feng Kang, the founding director of the Computing Center of the Chinese Academy of Sciences. The monograph series has provided timely information of the frontier directions and latest research results in computational mathematics. It has had great impact on young scientists and the entire research community, and has played a very important role in the development of computational mathematics in China.

To cope with these new scientific developments, the Ministry of Education of the People's Republic of China in 1998 combined several subjects, such as computational mathematics, numerical algorithms, information science, and operations research and optimal control, into a new discipline called Information and Computational Science. As a result, Science Press also reorganized the editorial board of the monograph series and changed its name to Series in Information and Computational Science. The first editorial board meeting was held in Beijing in September 2004, and it discussed the new objectives, and the directions and contents of the new monograph series.

The aim of the new series is to present the state of the art in Information and Computational Science to senior undergraduate and graduate students, as well as to scientists working in these fields. Hence, the series will provide concrete and systematic expositions of the advances in information and computational science, encompassing also related interdisciplinary developments.

I would like to thank the previous editorial board members and assistants, and all the mathematicians who have contributed significantly to the monograph series on Computational Methods. As a result of their contributions the monograph series achieved an outstanding reputation in the community. I sincerely wish that we will extend this support to the new Series in Information and Computational Science, so that the new series can equally enhance the scientific development in information and computational science in this century.

Shi Zhongci

2005.7

Preface

Numerical linear algebra, also called matrix computation, has been a center of scientific and engineering computing since 1946, the first modern computer was born. Most of problems in science and engineering finally become problems in matrix computation. Therefore, it is important for us to study numerical linear algebra. This book gives an elementary introduction to matrix computation and it also includes some new results obtained in recent years. In the beginning of this book, we first give an outline of numerical linear algebra in Chapter 1.

In Chapter 2, we introduce Gaussian elimination, a basic direct method, for solving general linear systems. Usually, Gaussian elimination is used for solving a dense linear system with median size and no special structure. The operation cost of Gaussian elimination is $O(n^3)$ where n is the size of the system. The pivoting technique is also studied.

In Chapter 3, in order to discuss effects of perturbation and error on numerical solutions, we introduce vector and matrix norms and study their properties. The error analysis on floating point operations and on partial pivoting technique is also given.

In Chapter 4, linear least squares problems are studied. We will concentrate on the problem of finding the least squares solution of an overdetermined linear system $Ax = b$ where A has more rows than columns. Some orthogonal transformations and the QR decomposition are used to design efficient algorithms for solving least squares problems.

We study classical iterative methods for the solution of $Ax = b$ in Chapter 5. Iterative methods are quite different from direct methods such as Gaussian elimination. Direct methods based on an LU factorization of the matrix A are prohibitive in terms of computing time and computer storage if A is quite large. Usually, in most large problems, the matrices are sparse. The sparsity may be lost during the LU factorization procedure and then at the end of LU factorization, the storage becomes a crucial issue. For such kind of problem, we can use a class of methods called iterative methods. We only consider some classical iterative methods in this chapter.

In Chapter 6, we introduce another class of iterative methods called Krylov subspace methods proposed recently. We will only study two versions among

those Krylov subspace methods: the conjugate gradient (CG) method and the generalized minimum residual (GMRES) method. The CG method proposed in 1952 is one of the best known iterative method for solving symmetric positive definite linear systems. The GMRES method was proposed in 1986 for solving nonsymmetric linear systems. The preconditioning technique is also studied.

Eigenvalue problems are particularly interesting in scientific computing. In Chapter 7, nonsymmetric eigenvalue problems are studied. We introduce some well-known methods such as the power method, the inverse power method and the QR method.

The symmetric eigenvalue problem with its nice properties and rich mathematical theory is one of the most interesting topics in numerical linear algebra. In Chapter 8, we will study this topic. The symmetric QR iteration method, the Jacobi method, the bisection method and a divide-and-conquer technique will be discussed in this chapter.

In Chapter 9, we will briefly survey some of the latest developments in using boundary value methods for solving systems of ordinary differential equations with initial values. These methods require the solutions of one or more nonsymmetric, large and sparse linear systems. Therefore, we will use the GMRES method in Chapter 6 with some preconditioners for solving these linear systems. One of the main results is that if an A_{ν_1,ν_2}-stable boundary value method is used for an m-by-m system of ODEs, then the preconditioned matrix can be decomposed as $I + L$ where I is the identity matrix and the rank of L is at most $2m(\nu_1 + \nu_2)$. It follows that when the GMRES method is applied to the preconditioned system, the method will converge in at most $2m(\nu_1+\nu_2)+1$ iterations. Applications to different delay differential equations are also given.

" If any other mathematical topic is as fundamental to the mathematical sciences as calculus and differential equations, it is numerical linear algebra. " — L. Trefethen and D. Bau III

Acknowledgments: We would like to thank Professor Raymond H. F. Chan of the Department of Mathematics, Chinese University of Hong Kong, for his constant encouragement, long-standing friendship, financial support; Professor Z. H. Cao of the Department of Mathematics, Fudan University, for his many helpful discussions and useful suggestions. We also would like to thank our friend Professor Z. C. Shi for his encouraging support and valuable comments. Of course, special appreciation goes to two important institutions in the authors' life: University of Macau and Fudan University for providing

a wonderful intellectual atmosphere for writing this book. Most of the writing was done during evenings, weekends and holidays. Finally, thanks are also due to our families for their endless love, understanding, encouragement and support essential to the completion of this book. The most heartfelt thanks to all of them!

The publication of the book is supported in part by the research grants No.RG024/01-02S/JXQ/FST, No.RG031/02-03S/JXQ/FST and No.RG064/03-04S/JXQ/FST from University of Macau; the research grant No.10471027 from the National Natural Science Foundation of China and some financial support from Shanghai Education Committee and Fudan University.

Authors' words on the corrected and revised second printing: In its second printing, we corrected some minor mathematical and typographical mistakes in the first printing of the book. We would like to thank all those people who pointed these out to us. Additional comments and some revision have been made in Chapter 7. The references have been updated. More exercises are also to be found in the book. The second printing of the book is supported by the research grant No.RG081/04-05S/JXQ/FST.

Contents

Chapter 1
Introduction

Numerical linear algebra (NLA) is also called matrix computation. It has been a center of scientific and engineering computing since the first modern computer came to this world around 1946. Most of problems in science and engineering are finally transferred into problems in NLA. Thus, it is very important for us to study NLA. This book gives an elementary introduction to NLA and it also includes some new results obtained in recent years.

1.1 Basic symbols

We will use the following symbols throughout this book.

- Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{C}$ denote the set of complex numbers and $\mathbf{i} \equiv \sqrt{-1}$.
- Let $\mathbb{R}^n$ denote the set of real n-vectors and $\mathbb{C}^n$ denote the set of complex n-vectors. Vectors will almost always be column vectors.
- Let $\mathbb{R}^{m\times n}$ denote the linear vector space of m-by-n real matrices and $\mathbb{C}^{m\times n}$ denote the linear vector space of m-by-n complex matrices.
- We will use the upper case letters such as A, B, C, Δ and Λ, etc, to denote matrices and use the lower case letters such as x, y, z, etc, to denote vectors.
- The symbol a_{ij} will denote the ij-th entry in a matrix A.
- The symbol A^T will denote the transpose of the matrix A and A^* will denote the conjugate transpose of the matrix A.
- Let $a_1, \cdots, a_m \in \mathbb{R}^n$ (or $\mathbb{C}^n$). We will use $\text{span}\{a_1, \cdots, a_m\}$ to denote the linear vector space of all the linear combinations of $a_1, \cdots, a_m$.
- Let $\text{rank}(A)$ denote the rank of the matrix A.
- Let $\dim(S)$ denote the dimension of the vector space S.

- We will use $\det(A)$ to denote the determinant of the matrix A and use $\mathrm{diag}(a_{11}, \cdots, a_{nn})$ to denote the n-by-n diagonal matrix:

$$\mathrm{diag}(a_{11}, \cdots, a_{nn}) = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix}.$$

- For matrix $A = [a_{ij}]$, the symbol $|A|$ will denote the matrix with entries $(|A|)_{ij} = |a_{ij}|$.

- The symbol I will denote the identity matrix, i.e.,

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix},$$

  and e_i will denote the i-th unit vector, i.e., the i-th column vector of I.

- We will use $\|\cdot\|$ to denote a norm of matrix or vector. The symbols $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ will denote the p-norm with $p = 1, 2, \infty$, respectively.

- As in MATLAB, in algorithms, $A(i,j)$ will denote the (i,j)-th entry of matrix A; $A(i,:)$ and $A(:,j)$ will denote the i-th row and the j-th column of A, respectively; $A(i_1 : i_2, k)$ will express the column vector constructed by using entries from the i_1-th entry to the i_2-th entry in the k-th column of A; $A(k, j_1 : j_2)$ will express the row vector constructed by using entries from the j_1-th entry to the j_2-th entry in the k-th row of A; $A(k : l, p : q)$ will denote the $(l-k+1)$-by-$(q-p+1)$ submatrix constructed by using the rows from the k-th row to the l-th row and the columns from the p-th column to the q-th column.

1.2 Basic problems in NLA

NLA includes the following three main important problems which will be studied in this book:

(1) Find the solution of linear systems

$$Ax = b$$

where A is an n-by-n nonsingular matrix and b is an n-vector.

(2) Linear least squares problems: For any m-by-n matrix A and an m-vector b, find an n-vector x such that

$$\|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2.$$

(3) Eigenvalues problems: For any n-by-n matrix A, find a part (or all) of its eigenvalues and corresponding eigenvectors. We remark here that a complex number λ is called an eigenvalue of A if there exists a nonzero vector $x \in \mathbb{C}^n$ such that

$$Ax = \lambda x,$$

where x is called the eigenvector of A associated with λ.

Besides these main problems, there are many other fundamental problems in NLA, for instance, total least squares problems, matrix equations, generalized inverses, inverse problems of eigenvalues, and singular value problems, etc.

1.3 Why shall we study numerical methods?

To answer this question, let us consider the following linear system,

$$Ax = b$$

where A is an n-by-n nonsingular matrix and $x = (x_1, x_2, \cdots, x_n)^T$. If we use the well-known Cramer rule, then we have the following solution:

$$x_1 = \frac{\det(A_1)}{\det(A)}, \quad x_2 = \frac{\det(A_2)}{\det(A)}, \cdots, \quad x_n = \frac{\det(A_n)}{\det(A)},$$

where A_i, for $i = 1, 2, \cdots, n$, are matrices with the i-th column replaced by the vector b. Then we should compute $n+1$ determinants $\det(A_i)$, $i = 1, 2, \cdots, n$, and $\det(A)$. There are

$$[n!(n-1)](n+1) = (n-1)(n+1)!$$

multiplications. When $n = 25$, by using a computer with 10 billion operations/sec., we need

$$\frac{24 \times 26!}{10^{10} \times 3600 \times 24 \times 365} \approx 30.6 \quad \text{billion years.}$$

If one uses Gaussian elimination, it requires

$$\sum_{i=1}^{n}(i-1)(i+1) = \sum_{i=1}^{n} i^2 - n = \frac{1}{6}n(n+1)(2n+1) - n = O(n^3)$$

multiplications. Then less than 1 second, we could solve 25-by-25 linear systems by using the same computer. From above discussions, we note that for solving the same problem by using different numerical methods, the results are much different. Therefore, it is essential for us to study the properties of numerical methods.

1.4 Matrix factorizations (decompositions)

For any linear system $Ax = b$, if we can factorize (decompose) A as $A = LU$ where L is a lower triangular matrix and U is an upper triangular matrix, then we have

$$\begin{cases} Ly = b \\ \\ Ux = y. \end{cases} \tag{1.1}$$

By substituting, we can easily solve (1.1) and then $Ax = b$. Therefore, matrix factorizations (decompositions) are very important tools in NLA. The following theorem is basic and important in linear algebra, see [17].

Theorem 1.1 (Jordan Decomposition Theorem) *If $A \in \mathbb{C}^{n\times n}$, then there exists a nonsingular matrix $X \in \mathbb{C}^{n\times n}$ such that*

$$X^{-1}AX = J \equiv \mathrm{diag}(J_1, J_2, \cdots, J_p),$$

or $A = XJX^{-1}$, where J is called the Jordan canonical form of A and

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \lambda_i \end{bmatrix} \in \mathbb{C}^{n_i\times n_i},$$

for $i = 1, 2, \cdots, p$, are called Jordan blocks with $n_1 + \cdots + n_p = n$. The Jordan canonical form of A is unique up to the permutation of diagonal Jordan blocks. If $A \in \mathbb{R}^{n\times n}$ with only real eigenvalues, then the matrix X can be taken to be real.

1.5 Perturbation and error analysis

The solutions provided by numerical algorithms are seldom absolutely correct. Usually, there are two kinds of errors. First, errors appear in input data caused by prior computations or measurements. Second, there may be errors caused by algorithms themselves because of approximations made within algorithms. Thus, we need to carry out a perturbation and error analysis.

(1) Perturbation.

For a given x, we want to compute the value of function $f(x)$. Suppose there is a perturbation δx of x and $|\delta x|/|x|$ is very small. We want to find a positive number $c(x)$ as small as possible such that

$$\frac{|f(x+\delta x)-f(x)|}{|f(x)|} \leq c(x)\frac{|\delta x|}{|x|}.$$

Then $c(x)$ is called the condition number of $f(x)$ at x. If $c(x)$ is large, we say that the function f is ill-conditioned at x; if $c(x)$ is small, we say that the function f is well-conditioned at x.

Remark: A computational problem being ill-conditioned or not has no relation with numerical methods that we used.

(2) Error.

By using some numerical methods, we calculate the value of a function f at a point x and we obtain $\hat{y}$. Because of the rounding error (or chopping error), usually

$$\hat{y} \neq f(x).$$

If there exists δx such that

$$\hat{y} = f(x+\delta x), \qquad |\delta x| \leq \epsilon |x|,$$

where ϵ is a positive constant having a closed relation with numerical methods and computers used, then we say that the method is stable if ϵ is small; the method is unstable if ϵ is large.

Remark: A numerical method being stable or not has no relation with computational problems that we faced.

With the perturbation and error analysis, we obtain

$$\frac{|\hat{y}-f(x)|}{|f(x)|} = \frac{|f(x+\delta x)-f(x)|}{|f(x)|} \leq c(x)\frac{|\delta x|}{|x|} \leq \epsilon c(x).$$

Therefore, whether a numerical result is accurate depends on both the stability of the numerical method and the condition number of the computational problem.