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.

Xiao-qing Jin and Yi-min Wei

Numerical Linear Algebra and Its Applications

（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.

1.6 Operation cost and convergence rate

Usually, numerical algorithms are divided into two classes:

$$\begin{cases} \text{(i) direct methods;} \\ \\ \text{(ii) iterative methods.} \end{cases}$$

By using direct methods, one can obtain an accurate solution of computational problems within finite steps in exact arithmetic. By using iterative methods, one can only obtain an approximate solution of computational problems within finite steps.

The operation cost is an important measurement of algorithms. The operation cost of an algorithm is the total operations of "$+$, $-$, $\times$, $\div$" used in the algorithm. We remark that the speed of algorithms is only partially depending on the operation cost. In modern computers, the speed of operations is much faster than that of data transfer. Therefore, sometimes, the speed of an algorithm is mainly depending on the total amount of data transfers.

For direct methods, usually, we use the operation cost as a main measurement of the speed of algorithms. For iterative methods, we need to consider

$$\begin{cases} \text{(i) operation cost in each iteration;} \\ \\ \text{(ii) convergence rate of the method.} \end{cases}$$

For a sequence $\{x_k\}$ provided by an iterative algorithm, if $\{x_k\} \longrightarrow x$, the exact solution, and if $\{x_k\}$ satisfies

$$\|x_k - x\| \leq c\|x_{k-1} - x\|, \quad k = 1, 2, \cdots,$$

where $0 < c < 1$ and $\|\cdot\|$ is any vector norm (see Chapter 3 for a detail), then we say that the convergence rate is linear. If it satisfies

$$\|x_k - x\| \leq c\|x_{k-1} - x\|^p, \quad k = 1, 2, \cdots,$$

where $0 < c < 1$ and $p > 1$, then we say that the convergence rate is superlinear.

Exercises:

1. A matrix is strictly upper triangular if it is upper triangular with zero diagonal elements. Show that if A is a strictly upper triangular matrix of order n, then $A^n = 0$.

2. Let $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{m\times l}$. Prove that

$$\operatorname{rank}(AB) \geq \operatorname{rank}(A) + \operatorname{rank}(B) - m.$$

3. Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

where A_{ij}, for $i, j = 1, 2$, are square matrices with $\det(A_{11}) \neq 0$, and satisfy

$$A_{11}A_{21} = A_{21}A_{11}.$$

Then

$$\det(A) = \det(A_{11}A_{22} - A_{21}A_{12}).$$

4. Show that $\det(I - uv^*) = 1 - v^*u$ where $u, v \in \mathbb{C}^m$ are column vectors.

5. Prove Hadamard's inequality for $A \in \mathbb{C}^{n\times n}$:

$$|\det(A)| \leq \prod_{k=1}^{n} \|a_k\|_2,$$

where $a_k = A(:, k)$. When does the equality hold?

6. Let B be nilpotent, i.e., there exists an integer $k > 0$ such that $B^k = 0$. Show that if $AB = BA$, then

$$\det(A + B) = \det(A).$$

7. Let A be an m-by-n matrix and B be an n-by-m matrix. Show that the matrices

$$\begin{bmatrix} AB & \mathbf{0} \\ B & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ B & BA \end{bmatrix}$$

are similar. Conclude that the nonzero eigenvalues of AB are the same as those of BA, and

$$\det(I_m + AB) = \det(I_n + BA).$$

8. A matrix $M \in \mathbb{C}^{n\times n}$ is Hermitian positive definite if it satisfies

$$M = M^*, \qquad x^*Mx > 0,$$

for all $x \neq 0 \in \mathbb{C}^n$. Let A and B be Hermitian positive definite matrices.

   (1) Show that the matrix product AB has positive eigenvalues.

   (2) Show that AB is Hermitian if and only if A and B commute.

9. Show that any matrix $A \in \mathbb{C}^{n\times n}$ can be written uniquely in the form

$$A = B + \mathbf{i}C,$$

where B and C are Hermitian.

10. Show that if A is skew-Hermitian, i.e., $A^* = -A$, then all its eigenvalues lie on the imaginary axis.

11. Let
$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$
Assume that A_{11}, A_{22} are square, and A_{11}, $A_{22} - A_{21}A_{11}^{-1}A_{12}$ are nonsingular. Let
$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$
be the inverse of A. Show that
$$B_{22} = (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}, \quad B_{12} = -A_{11}^{-1}A_{12}B_{22},$$
$$B_{21} = -B_{22}A_{21}A_{11}^{-1}, \quad B_{11} = A_{11}^{-1} - B_{12}A_{21}A_{11}^{-1}.$$

12. Suppose that A and B are Hermitian with A being positive definite. Show that $A + B$ is positive definite if and only if all the eigenvalues of $A^{-1}B$ are greater than -1.

13. Let A be idempotent, i.e., $A^2 = A$. Show that the eigenvalues of A is either 0 or 1.

14. Let A be a matrix with all entries equal to one. Show that A can be written as $A = ee^T$, where $e^T = (1, 1, \cdots, 1)$, and A is positive semi-definite. Find the eigenvalues and eigenvectors of A.

15. Prove that any matrix $A \in \mathbb{C}^{n\times n}$ has a polar decomposition $A = HQ$, where H is Hermitian positive semi-definite and Q is unitary. We recall that $M \in \mathbb{C}^{n\times n}$ is a unitary matrix if $M^{-1} = M^*$. Moreover, if A is nonsingular, then H is Hermitian positive definite and the polar decomposition of A is unique.

Chapter 2
Direct Methods for Linear Systems

The problem of solving linear systems is central in NLA. For solving linear systems, in general, we have two classes of methods. One is called the direct method and the other is called the iterative method. By using direct methods, within finite steps, one can obtain an accurate solution of computational problems in exact arithmetic. By using iterative methods, within finite steps, one can only obtain an approximate solution of computational problems.

In this chapter, we will introduce a basic direct method called Gaussian elimination for solving general linear systems. Usually, Gaussian elimination is used for solving a dense linear system with median size and no special structure.

2.1 Triangular linear systems and LU factorization

We first study triangular linear systems.

2.1.1 Triangular linear systems

We consider the following nonsingular lower triangular linear system

$$Ly = b \tag{2.1}$$

where $b = (b_1, b_2, \cdots, b_n)^T \in \mathbb{R}^n$ is a known vector, $y = (y_1, y_2, \cdots, y_n)^T$ is an unknown vector, and $L = [l_{ij}] \in \mathbb{R}^{n\times n}$ is given by

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & \vdots \\ l_{31} & l_{32} & l_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}$$

with $l_{ii} \neq 0$, $i = 1, 2, \cdots, n$. By the first equation in (2.1), we have

$$l_{11}y_1 = b_1,$$

and then

$$y_1 = \frac{b_1}{l_{11}}.$$

Similarly, by the second equation in (2.1), we have

$$y_2 = \frac{1}{l_{22}}(b_2 - l_{21}y_1).$$

In general, if we have already obtained $y_1, y_2, \cdots, y_{i-1}$, then by using the i-th equation in (2.1), we have

$$y_i = \frac{1}{l_{ii}}\Big(b_i - \sum_{j=1}^{i-1} l_{ij}y_j\Big).$$

This algorithm is called the forward substitution method which needs $O(n^2)$ operations.

Now, we consider the following nonsingular upper triangular linear system

$$Ux = y \tag{2.2}$$

where $x = (x_1, x_2, \cdots, x_n)^T$ is an unknown vector, and $U \in \mathbb{R}^{n\times n}$ is given by

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & \vdots \\ 0 & 0 & u_{33} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & u_{nn} \end{bmatrix}$$

with $u_{ii} \neq 0$, $i = 1, 2, \cdots, n$. Beginning from the last equation of (2.2), we can obtain $x_n, x_{n-1}, \cdots, x_1$ step by step. The $x_n = y_n/u_{nn}$ and x_i is given by

$$x_i = \frac{1}{u_{ii}}\Big(y_i - \sum_{j=i+1}^{n} u_{ij}x_j\Big)$$

for $i = n-1, \cdots, 1$. This algorithm is called the backward substitution method which also needs $O(n^2)$ operations.

For general linear systems

$$Ax = b \tag{2.3}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are known. If we can factorize the matrix A into

$$A = LU$$

where L is a lower triangular matrix and U is an upper triangular matrix, then we could find the solution of (2.3) by the following two steps:

(1) By using the forward substitution method to find solution y of $Ly = b$.

(2) By using the backward substitution method to find solution x of $Ux = y$.

Now the problem that we are facing is how to factorize the matrix A into $A = LU$. We therefore introduce Gaussian transform matrices.

2.1.2 Gaussian transform matrix

Let

$$L_k = I - l_k e_k^T$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix, $l_k = (0, \cdots, 0, l_{k+1,k}, \cdots, l_{nk})^T \in \mathbb{R}^n$ and $e_k \in \mathbb{R}^n$ is the k-th unit vector. Then for any k,

$$L_k = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & & 1 & & & \vdots \\ \vdots & & -l_{k+1,k} & 1 & & \vdots \\ \vdots & & \vdots & & \ddots & 0 \\ 0 & \cdots & -l_{nk} & \cdots & 0 & 1 \end{bmatrix}$$

is called the Gaussian transform matrix. Such a matrix is a unit lower triangular matrix. We remark that a unit triangular matrix is a triangular matrix with ones on its diagonal. For any given vector

$$x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n,$$

we have

$$\begin{aligned} L_k x &= (x_1, \cdots, x_k, x_{k+1} - x_k l_{k+1,k}, \cdots, x_n - x_k l_{nk})^T \\ &= (x_1, \cdots, x_k, 0, \cdots, 0)^T \end{aligned}$$

if we take

$$l_{ik} = \frac{x_i}{x_k}, \qquad i = k+1, \cdots, n$$

with $x_k \neq 0$. It is easy to check that

$$L_k^{-1} = I + l_k e_k^T$$

by noting that $e_k^T l_k = 0$.

For a given matrix $A \in \mathbb{R}^{n\times n}$, we have

$$L_k A = (I - l_k e_k^T)A = A - l_k(e_k^T A)$$

and

$$\text{rank}(l_k(e_k^T A)) = 1. .$$

Therefore, $L_k A$ is a rank-one modification of the matrix A.

2.1.3 Computation of LU factorization

We consider the following simple example. Let

$$A = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 4 & 7 \\ 3 & 3 & 10 \end{bmatrix}.$$

By using the Gaussian transform matrix

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix},$$

we have

$$L_1 A = \begin{bmatrix} 1 & 5 & 9 \\ 0 & -6 & -11 \\ 0 & -12 & -17 \end{bmatrix}.$$

Followed by using the Gaussian transform matrix

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix},$$

we have

$$L_2(L_1 A) \equiv U = \begin{bmatrix} 1 & 5 & 9 \\ 0 & -6 & -11 \\ 0 & 0 & 5 \end{bmatrix}.$$

Therefore, we finally have

$$A = LU$$

where

$$L \equiv (L_2 L_1)^{-1} = L_1^{-1} L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}.$$

For general n-by-n matrix A, we can use $n-1$ Gaussian transform matrices $L_1, L_2, \cdots, L_{n-1}$ such that $L_{n-1} \cdots L_1 A$ is an upper triangular matrix. In fact, let $A^{(0)} \equiv A$ and assume that we have already found $k-1$ Gaussian transform matrices $L_1, \cdots, L_{k-1} \in \mathbb{R}^{n \times n}$ such that

$$A^{(k-1)} = L_{k-1} \cdots L_1 A = \begin{bmatrix} A_{11}^{(k-1)} & A_{12}^{(k-1)} \\ \mathbf{0} & A_{22}^{(k-1)} \end{bmatrix}$$

where $A_{11}^{(k-1)}$ is a $(k-1)$-by-$(k-1)$ upper triangular matrix and

$$A_{22}^{(k-1)} = \begin{bmatrix} a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & \ddots & \vdots \\ a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{bmatrix}.$$

If $a_{kk}^{(k-1)} \neq 0$, then we can use the Gaussian transform matrix

$$L_k = I - l_k e_k^T$$

where

$$l_k = (0, \cdots, 0, l_{k+1,k}, \cdots, l_{nk})^T$$

with

$$l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \qquad i = k+1, \cdots, n,$$

such that the last $n-k$ entries in the k-th column of $L_k A^{(k-1)}$ become zeros. We therefore have

$$A^{(k)} \equiv L_k A^{(k-1)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ \mathbf{0} & A_{22}^{(k)} \end{bmatrix}$$

where $A_{11}^{(k)}$ is a k-by-k upper triangular matrix. After $n-1$ steps, we obtain $A^{(n-1)}$ which is an upper triangular matrix that we need. Let

$$L = (L_{n-1} \cdots L_1)^{-1}, \qquad U = A^{(n-1)},$$

then $A = LU$. Now we want to show that L is a unit lower triangular matrix. By noting that $e_j^T l_i = 0$ for $j < i$, we have

$$\begin{aligned} L &= L_1^{-1} \cdots L_{n-1}^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= I + l_1 e_1^T + \cdots + l_{n-1} e_{n-1}^T \\ &= I + [l_1, l_2, \cdots, l_{n-1}, 0] \\ &= \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}. \end{aligned}$$

This computational process of the LU factorization is called Gaussian elimination. Thus, we have the following algorithm.

Algorithm 2.1 (Gaussian elimination)

$$\left\{\begin{array}{l} \textbf{for} \quad k = 1 : n-1 \\ \qquad A(k+1:n,k) = A(k+1:n,k)/A(k,k) \\ \qquad A(k+1:n,k+1:n) = A(k+1:n,k+1:n) \\ \qquad\quad -A(k+1:n,k)A(k,k+1:n) \\ \textbf{end} \end{array}\right.$$

The operation cost of Gaussian elimination is

$$\begin{aligned} \sum_{k=1}^{n-1}\Big((n-k)+2(n-k)^2\Big) &= \frac{n(n-1)}{2}+\frac{n(n-1)(2n-1)}{3} \\ &= \frac{2}{3}n^3+O(n^2)=O(n^3). \end{aligned}$$

We remark that in Gaussian elimination, $a_{kk}^{(k-1)}$, $k=1,\cdots,n-1$, are required to be nonzero. We have the following theorem.

Theorem 2.1 *The entries $a_{ii}^{(i-1)} \neq 0$, $i=1,\cdots,k$, if and only if all the leading principal submatrices A_i of A, $i=1,\cdots,k$, are nonsingular.*

Proof By induction, for $k=1$, it is obviously true. Assume that the statement is true until $k-1$. We want to show that if $A_1,\cdots,A_{k-1}$ are nonsingular, then

$$\text{“ } A_k \text{ is nonsingular} \Longleftrightarrow a_{kk}^{(k-1)} \neq 0 \text{ ”}.$$

By assumption, we know that

$$a_{ii}^{(i-1)} \neq 0, \qquad i=1,\cdots,k-1.$$

By using $k-1$ Gaussian transform matrices $L_1,\cdots,L_{k-1}$, we obtain

$$A^{(k-1)} = L_{k-1}\cdots L_1 A = \begin{bmatrix} A_{11}^{(k-1)} & A_{12}^{(k-1)} \\ \mathbf{0} & A_{22}^{(k-1)} \end{bmatrix} \tag{2.4}$$

where $A_{11}^{(k-1)}$ is an upper triangular matrix with nonzero diagonal entries $a_{ii}^{(i-1)}$, $i=1,\cdots,k-1$. Therefore, the k-th leading principal submatrix of $A^{(k-1)}$ has the following form

$$\begin{bmatrix} A_{11}^{(k-1)} & * \\ \mathbf{0} & a_{kk}^{(k-1)} \end{bmatrix}$$

Let $(L_1)_k, \cdots, (L_{k-1})_k$ denote the k-th leading principal submatrices of $L_1, \cdots, L_{k-1}$, respectively. By using (2.4), we obtain

$$(L_{k-1})_k \cdots (L_1)_k A_k = \begin{bmatrix} A_{11}^{(k-1)} & * \\ \mathbf{0} & a_{kk}^{(k-1)} \end{bmatrix}.$$

By noting that L_i, $i = 1, \cdots, k-1$, are unit lower triangular matrices, we immediately know that

$$\det(A_k) = a_{kk}^{(k-1)} \det(A_{11}^{(k-1)}) \neq 0$$

if and only if $a_{kk}^{(k-1)} \neq 0$. □

Thus, we have

Theorem 2.2 *If all the leading principal submatrices A_i of a matrix $A \in \mathbb{R}^{n \times n}$ are nonsingular for $i = 1, \cdots, n-1$, then there exists a unique LU factorization of A.*

2.2 *LU* factorization with pivoting

Before we study pivoting techniques, we first consider the following simple example:

$$\begin{bmatrix} 0.3 \times 10^{-11} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix}.$$

If we using Gaussian elimination with the 10-decimal-digit floating point arithmetic, we have

$$\widehat{L} = \begin{bmatrix} 1 & 0 \\ 0.3333333333 \times 10^{12} & 1 \end{bmatrix}$$

and

$$\widehat{U} = \begin{bmatrix} 0.3 \times 10^{-11} & 1 \\ 0 & -0.3333333333 \times 10^{12} \end{bmatrix}.$$

Then the computational solution is

$$\hat{x} = (0.0000000000, 0.7000000000)^T$$

which is not good comparing with the accurate solution

$$x = (0.2000000000006 \cdots, 0.6999999999994 \cdots)^T.$$

If we just interchange the first equation and the second equation, we have

$$\begin{bmatrix} 1 & 1 \\ 0.3 \times 10^{-11} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix}.$$

By using Gaussian elimination with the 10-decimal-digit floating point arithmetic again, we have

$$\widehat{L} = \begin{bmatrix} 1 & 0 \\ 0.3 \times 10^{-11} & 1 \end{bmatrix}, \qquad \widehat{U} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Then the computational solution is

$$\hat{x} = (0.2000000000, 0.7000000000)^T$$

which is very good. So we need to introduce permutations into Gaussian elimination. We first define a permutation matrix.

Definition 2.1 *A permutation matrix P is an identity matrix with permuted rows.*

The important properties of the permutation matrix are included in the following lemma. Its proof is straightforward.

Lemma 2.1 *Let P, P_1, $P_2 \in \mathbb{R}^{n\times n}$ be permutation matrices and $X \in \mathbb{R}^{n\times n}$. Then*

(i) *PX is the same as X with its rows permuted. XP is the same as X with its columns permuted.*

(ii) $P^{-1} = P^T$.

(iii) $det(P) = \pm 1$.

(iv) *$P_1 \cdot P_2$ is also a permutation matrix.*

Now we introduce the main theorem of this section.

Theorem 2.3 *If A is nonsingular, then there exist permutation matrices P_1 and P_2, a unit lower triangular matrix L, and a nonsingular upper triangular matrix U such that*

$$P_1 A P_2 = LU.$$

Only one of P_1 and P_2 is necessary.

Proof We use induction on the dimension n. For $n = 1$, it is obviously true. Assume that the statement is true for $n-1$. If A is nonsingular, then it has a nonzero entry. Choose permutation matrices P_1' and P_2' such that the $(1,1)$-th position of $P_1'AP_2'$ is nonzero. Now we write a desired factorization and solve for unknown components:

$$\begin{aligned} P_1'AP_2' &= \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \\ L_{21} & I \end{bmatrix} \cdot \begin{bmatrix} u_{11} & U_{12} \\ \mathbf{0} & \widetilde{A}_{22} \end{bmatrix} \\ &= \begin{bmatrix} u_{11} & U_{12} \\ L_{21}u_{11} & L_{21}U_{12} + \widetilde{A}_{22} \end{bmatrix}, \end{aligned} \tag{2.5}$$

where A_{22}, $\widetilde{A}_{22}$ are $(n-1)$-by-$(n-1)$ matrices, and L_{21}, U_{12}^T are $(n-1)$-by-1 matrices.

Solving for the components of this 2-by-2 block factorization, we get

$$u_{11} = a_{11} \neq 0, \qquad U_{12} = A_{12},$$

and

$$L_{21}u_{11} = A_{21}, \qquad A_{22} = L_{21}U_{12} + \widetilde{A}_{22}.$$

Therefore, we obtain

$$L_{21} = \frac{A_{21}}{a_{11}}, \qquad \widetilde{A}_{22} = A_{22} - L_{21}U_{12}.$$

We want to apply induction to $\widetilde{A}_{22}$, but to do so we need to check that

$$\det(\widetilde{A}_{22}) \neq 0.$$

Since

$$\det(P_1'AP_2') = \pm\det(A) \neq 0$$

and also

$$\begin{aligned} \det(P_1'AP_2') &= \det \begin{bmatrix} 1 & \mathbf{0} \\ L_{21} & I \end{bmatrix} \cdot \det \begin{bmatrix} u_{11} & U_{12} \\ \mathbf{0} & \widetilde{A}_{22} \end{bmatrix} \\ &= u_{11} \cdot \det(\widetilde{A}_{22}), \end{aligned}$$

we know that

$$\det(\widetilde{A}_{22}) \neq 0.$$

Therefore, by the assumption of induction, there exist permutation matrices $\widetilde{P}_1$ and $\widetilde{P}_2$ such that

$$\widetilde{P}_1\widetilde{A}_{22}\widetilde{P}_2 = \widetilde{L}\widetilde{U}, \tag{2.6}$$

where $\widetilde{L}$ is a unit lower triangular matrix and $\widetilde{U}$ is a nonsingular upper triangular matrix. Substituting (2.6) into (2.5) yields

$$
\begin{aligned}
P_1' A P_2' &= \begin{bmatrix} 1 & \mathbf{0} \\ L_{21} & I \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ \mathbf{0} & \widetilde{P}_1^T \widetilde{L} \widetilde{U} \widetilde{P}_2^T \end{bmatrix} \\
&= \begin{bmatrix} 1 & \mathbf{0} \\ L_{21} & I \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{P}_1^T \widetilde{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ \mathbf{0} & \widetilde{U} \widetilde{P}_2^T \end{bmatrix} \\
&= \begin{bmatrix} 1 & \mathbf{0} \\ L_{21} & \widetilde{P}_1^T \widetilde{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12}\widetilde{P}_2 \\ \mathbf{0} & \widetilde{U} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{P}_2^T \end{bmatrix} \\
&= \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{P}_1^T \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \widetilde{P}_1 L_{21} & \widetilde{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12}\widetilde{P}_2 \\ \mathbf{0} & \widetilde{U} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{P}_2^T \end{bmatrix},
\end{aligned}
$$

so we get a desired factorization of A:

$$
\begin{aligned}
P_1 A P_2 &= \left(\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{P}_1 \end{bmatrix} P_1' \right) A \left(P_2' \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{P}_2 \end{bmatrix} \right) \\
&= \begin{bmatrix} 1 & \mathbf{0} \\ \widetilde{P}_1 L_{21} & \widetilde{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12}\widetilde{P}_2 \\ \mathbf{0} & \widetilde{U} \end{bmatrix}. \qquad \square
\end{aligned}
$$

This row-column interchange strategy is called complete pivoting. We therefore have the following algorithm.

Algorithm 2.2 (Gaussian elimination with complete pivoting)

for $k = 1 : n-1$
  choose $p,\ q,\ (k \le p, q \le n)$ *such that*
    $|A(p,q)| = \max\{|A(i,j)| \ :\ i = k:n,\ j = k:n\}$
  $A(k, 1:n) \longleftrightarrow A(p, 1:n)$
  $A(1:n, k) \longleftrightarrow A(1:n, q)$
  if $A(k,k) \neq 0$
    $A(k+1:n, k) = A(k+1:n, k)/A(k,k)$
    $A(k+1:n, k+1:n) = A(k+1:n, k+1:n)$
      $-A(k+1:n, k)A(k, k+1:n)$
  else
    stop
  end
end

We remark that although the LU factorization with complete pivoting can overcome some shortcomings of the LU factorization without pivoting, the cost of complete pivoting is very high. Usually, it requires $O(n^3)$ operations in comparison with entries of the matrix for pivoting.

In order to reduce the operation cost of pivoting, the LU factorization with partial pivoting is proposed. In partial pivoting, at the k-th step, we choose $a_{pk}^{(k-1)}$ from the submatrix $A_{22}^{(k-1)}$ which satisfies

$$|a_{pk}^{(k-1)}| = \max\left\{|a_{ik}^{(k-1)}| : k \le i \le n\right\}.$$

When A is nonsingular, the LU factorization with partial pivoting can be carried out until we finally obtain

$$PA = LU.$$

In this algorithm, the operation cost in comparison with entries of the matrix for pivoting is $O(n^2)$. We have

Algorithm 2.3 (Gaussian elimination with partial pivoting)

for $k = 1 : n-1$
  choose p, $(k \le p \le n)$ *such that*
    $|A(p,k)| = \max\{|A(i,k)| \ : \ i = k : n\}$
  $A(k, 1:n) \longleftrightarrow A(p, 1:n)$
  if $A(k,k) \neq 0$
    $A(k+1:n, k) = A(k+1:n, k)/A(k,k)$
    $A(k+1:n, k+1:n) = A(k+1:n, k+1:n)$
      $-A(k+1:n, k)A(k, k+1:n)$
  else
    stop
  end
end

2.3 Cholesky factorization

Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite, i.e., it satisfies

$$A = A^T, \qquad x^T Ax > 0,$$

for all $x \neq 0 \in \mathbb{R}^n$. We have

Theorem 2.4 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Then there exists a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that*

$$A = LL^T.$$

This factorization is called the Cholesky factorization.

Proof Since A is positive definite, all the principal submatrices of A should be positive definite. By Theorem 2.2, there exist a unit lower triangular matrix $\widetilde{L}$ and an upper triangular matrix U such that

$$A = \widetilde{L}U.$$

Let

$$D = \text{diag}(u_{11}, \cdots, u_{nn}), \qquad \widetilde{U} = D^{-1}U,$$

where $u_{ii} > 0$, for $i = 1, \cdots, n$. Then we have

$$\widetilde{U}^T D\widetilde{L}^T = A^T = A = \widetilde{L}D\widetilde{U}.$$

Therefore,

$$\widetilde{L}^T\widetilde{U}^{-1} = D^{-1}\widetilde{U}^{-T}\widetilde{L}D.$$

We note that $\widetilde{L}^T\widetilde{U}^{-1}$ is a unit upper triangular matrix and $D^{-1}\widetilde{U}^{-T}\widetilde{L}D$ is a lower triangular matrix. Hence

$$\widetilde{L}^T\widetilde{U}^{-1} = I = D^{-1}\widetilde{U}^{-T}\widetilde{L}D$$

which implies $\widetilde{U} = \widetilde{L}^T$. Thus

$$A = \widetilde{L}D\widetilde{L}^T.$$

Let

$$L = \widetilde{L}\text{diag}(\sqrt{u_{11}}, \cdots, \sqrt{u_{nn}}).$$

We finally have

$$A = LL^T. \qquad \square$$

Thus, when a matrix A is symmetric positive definite, we could find the solution of the system $Ax = b$ by the following three steps:

(1) Find the Cholesky factorization of A: $A = LL^T$.

(2) Find solution y of $Ly = b$.

(3) Find solution x of $L^T x = y$.

From Theorem 2.4, we know that we do not need a pivoting in Cholesky factorization. Also we could calculate L directly through a comparison in the corresponding entries between two sides of $A = LL^T$. We have the following algorithm.

Algorithm 2.4 (Cholesky factorization)

```
for  k = 1 : n
     A(k,k) = sqrt(A(k,k))
     A(k+1:n,k) = A(k+1:n,k)/A(k,k)
     for  j = k+1 : n
          A(j:n,j) = A(j:n,j) - A(j:n,k)A(j,k)
     end
end
```

The operation cost of Cholesky factorization is $n^3/3$.

Exercises:

1. Let $S, T \in \mathbb{R}^{n\times n}$ be upper triangular matrices such that

$$(ST - \lambda I)x = b$$

is a nonsingular system. Find an algorithm of $O(n^2)$ operations for computing x.

2. Show that the LDL^T factorization of a symmetric positive definite matrix A is unique.

3. Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Find an algorithm for computing an upper triangular matrix $U \in \mathbb{R}^{n\times n}$ such that $A = UU^T$.

4. Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ be strictly diagonally dominant matrix, i.e.,

$$|a_{kk}| > \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}|, \qquad k = 1, 2, \cdots, n.$$

Prove that a strictly diagonally dominant matrix is nonsingular, and a strictly diagonally dominant symmetric matrix with positive diagonal entries is positive definite.

5. Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

with A_{11} being a k-by-k nonsingular matrix. Then

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

is called the Schur complement of A_{11} in A. Show that after k steps of Gaussian elimination without pivoting, $A_{22}^{(k-1)} = S$.

6. Let A be a symmetric positive definite matrix. At the end of the first step of Gaussian elimination, we have

$$\begin{bmatrix} a_{11} & a_1^T \\ \mathbf{0} & A_{22} \end{bmatrix}.$$

Prove that A_{22} is also symmetric positive definite.

7. Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ be a strictly diagonally dominant matrix. After one step of Gaussian elimination, we have

$$\begin{bmatrix} a_{11} & a_1^T \\ \mathbf{0} & A_{22} \end{bmatrix}.$$

Show that A_{22} is also strictly diagonally dominant.

8. Show that if $PAQ = LU$ is obtained via Gaussian elimination with pivoting, then $|u_{ii}| \geq |u_{ij}|$, for $j = i+1, \cdots, n$.

9. Let $H = A + \mathbf{i}B$ be a Hermitian positive definite matrix, where $A, B \in \mathbb{R}^{n\times n}$.

   (1) Prove the matrix

$$C = \begin{bmatrix} A & -B \\ B & A \end{bmatrix}$$

   is symmetric positive definite.

   (2) How to solve

$$(A + \mathbf{i}B)(x + \mathbf{i}y) = b + \mathbf{i}c, \qquad x,\ y,\ b,\ c \in \mathbb{R}^n$$

   by real number computation only?

10. Develop an algorithm to solve a tridiagonal system by using Gaussian elimination with partial pivoting.

11. Show that if a singular matrix $A \in \mathbb{R}^{n\times n}$ has a unique LU factorization, then A_k is nonsingular for $k = 1, 2, \cdots, n-1$.

Chapter 3
Perturbation and Error Analysis

In this chapter, we will discuss effects of perturbation and error on numerical solutions. The error analysis on floating point operations and on partial pivoting technique is also given. It is well-known that the essential notions of distance and size in linear vector spaces are captured by norms. We therefore need to introduce vector and matrix norms and study their properties before we develop our perturbation and error analysis.

3.1 Vector and matrix norms

We first introduce vector norms.

3.1.1 Vector norms

Let

$$x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n.$$

Definition 3.1 *A vector norm on $\mathbb{R}^n$ is a function that assigns to each $x \in \mathbb{R}^n$ a real number $\|x\|$, called the norm of x, such that the following three properties are satisfied for all $x, y \in \mathbb{R}^n$ and all $\alpha \in \mathbb{R}$:*

(i) $\|x\| > 0$ *if* $x \neq 0$, *and* $\|x\| = 0$ *if and only if* $x = 0$;

(ii) $\|\alpha x\| = |\alpha| \cdot \|x\|$;

(iii) $\|x + y\| \leq \|x\| + \|y\|$.

A useful class of vector norms is the p-norm defined by

$$\|x\|_p \equiv \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

where $1 \leq p$. The following p-norms are the most commonly used norms in practice:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}}, \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

The Cauchy-Schwarz inequality concerning $\|\cdot\|_2$ is given as follows,

$$|x^T y| \leq \|x\|_2 \|y\|_2$$

for x, $y \in \mathbb{R}^n$, which is a special case of the Hölder inequality given as follows,

$$|x^T y| \leq \|x\|_p \|y\|_q$$

where $1/p + 1/q = 1$.

A very important property of vector norms on $\mathbb{R}^n$ is that all the vector norms on $\mathbb{R}^n$ are equivalent as the following theorem said, see [35].

Theorem 3.1 *If $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are two norms on $\mathbb{R}^n$, then there exist two positive constants c_1 and c_2 such that*

$$c_1 \|x\|_\alpha \leq \|x\|_\beta \leq c_2 \|x\|_\alpha$$

for all $x \in \mathbb{R}^n$.

For example, if $x \in \mathbb{R}^n$, then we have

$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \|x\|_2,$$

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty$$

and

$$\|x\|_\infty \leq \|x\|_1 \leq n \|x\|_\infty.$$

We remark that for any sequence of vectors $\{x^k\}$ where $x^k = (x_1^{(k)}, \cdots, x_n^{(k)})^T \in \mathbb{R}^n$, and $x = (x_1, \cdots, x_n)^T \in \mathbb{R}^n$, by Theorem 3.1, one can prove that

$$\lim_{k \to \infty} \|x^k - x\| = 0 \Longleftrightarrow \lim_{k \to \infty} |x_i^{(k)} - x_i| = 0,$$

for $i = 1, \cdots, n$.

3.1.2 Matrix norms

Let

$$A = [a_{ij}]_{i,j=1}^{n} \in \mathbb{R}^{n\times n}.$$

We now turn our attention to matrix norms.

Definition 3.2 *A matrix norm is a function that assigns to each $A \in \mathbb{R}^{n\times n}$ a real number $\|A\|$, called the norm of A, such that the following four properties are satisfied for all $A, B \in \mathbb{R}^{n\times n}$ and all $\alpha \in \mathbb{R}$:*

(i) $\|A\| > 0$ *if* $A \neq 0$, *and* $\|A\| = 0$ *if and only if* $A = 0$;

(ii) $\|\alpha A\| = |\alpha| \cdot \|A\|$;

(iii) $\|A + B\| \leq \|A\| + \|B\|$;

(iv) $\|AB\| \leq \|A\| \cdot \|B\|$.

An important property of matrix norms on $\mathbb{R}^{n\times n}$ is that all the matrix norms on $\mathbb{R}^{n\times n}$ are equivalent. For the relation between a vector norm and a matrix norm, we have

Definition 3.3 *If a matrix norm $\|\cdot\|_M$ and a vector norm $\|\cdot\|_v$ satisfy*

$$\|Ax\|_v \leq \|A\|_M \|x\|_v,$$

for $A \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$, then these norms are called mutually consistent.

For any vector norm $\|\cdot\|_v$, we can define a matrix norm in the following natural way:

$$\|A\|_M \equiv \max_{x\neq 0} \frac{\|Ax\|_v}{\|x\|_v} = \max_{\|x\|_v=1} \|Ax\|_v.$$

The most important matrix norms are the matrix p-norms induced by the vector p-norms for $p = 1, 2, \infty$. We have the following theorem.

Theorem 3.2 *Let*

$$A = [a_{ij}]_{i,j=1}^{n} \in \mathbb{R}^{n\times n}.$$

Then we have

(i) $\|A\|_1 = \max\limits_{1\leq j\leq n} \sum\limits_{i=1}^{n} |a_{ij}|$.

(ii) $\|A\|_\infty = \max\limits_{1\le i\le n} \sum\limits_{j=1}^{n} |a_{ij}|$.

(iii) $\|A\|_2 = \sqrt{\lambda_{\max}(A^TA)}$, *where* $\lambda_{\max}(A^TA)$ *is the largest eigenvalue of* A^TA.

Proof We only give the proof of (i) and (iii). In the following, we always assume that $A \neq 0$.

For (i), we partition the matrix A by columns:

$$A = [a_1, \cdots, a_n].$$

Let

$$\delta = \|a_{j_0}\|_1 = \max_{1\le j\le n} \|a_j\|_1.$$

Then for any vector $x \in \mathbb{R}^n$ which satisfies $\|x\|_1 = \sum\limits_{i=1}^{n} |x_i| = 1$, we have

$$\begin{aligned} \|Ax\|_1 &= \Big\| \sum_{j=1}^{n} x_j a_j \Big\|_1 \le \sum_{j=1}^{n} |x_j| \cdot \|a_j\|_1 \\ &\le (\sum_{j=1}^{n} |x_j|) \cdot \max_{1\le j\le n} \|a_j\|_1 \\ &= \|a_{j_0}\|_1 = \delta. \end{aligned}$$

Let e_{j_0} denote the j_0-th unit vector and then

$$\|Ae_{j_0}\|_1 = \|a_{j_0}\|_1 = \delta.$$

Therefore

$$\|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1 = \delta = \max_{1\le j\le n} \|a_j\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|.$$

For (iii), we have

$$\begin{aligned} \|A\|_2 &= \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} [(Ax)^T(Ax)]^{1/2} \\ &= \max_{\|x\|_2=1} [x^T(A^TA)x]^{1/2}. \end{aligned}$$

Since A^TA is positive semi-definite, its eigenvalues can be assumed to be in the following order:

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0.$$

Let

$$v_1, v_2, \cdots, v_n \in \mathbb{R}^n$$

denote the orthonormal eigenvectors corresponding to $\lambda_1, \lambda_2, \cdots, \lambda_n$, respectively. Then for any vector $x \in \mathbb{R}^n$ with $\|x\|_2 = 1$, we have

$$x = \sum_{i=1}^{n} \alpha_i v_i, \qquad \sum_{i=1}^{n} \alpha_i^2 = 1.$$

Therefore,

$$x^T A^T A x = \sum_{i=1}^{n} \lambda_i \alpha_i^2 \leq \lambda_1.$$

On the other hand, let $x = v_1$, we have

$$x^T A^T A x = v_1^T A^T A v_1 = v_1^T \lambda_1 v_1 = \lambda_1.$$

Thus

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\lambda_1} = \sqrt{\lambda_{\max}(A^T A)}. \qquad \square$$

We have the following theorem for the norm $\|\cdot\|_2$.

Theorem 3.3 *Let $A \in \mathbb{R}^{n\times n}$. Then we have*

(i) $\|A\|_2 = \max\limits_{\|x\|_2=1} \max\limits_{\|y\|_2=1} |y^* A x|$, *where* $x, y \in \mathbb{C}^n$.

(ii) $\|A^T\|_2 = \|A\|_2 = \sqrt{\|A^T A\|_2}$.

(iii) $\|A\|_2 = \|QAZ\|_2$, *for any orthogonal matrices Q and Z. We recall that a matrix $M \in \mathbb{R}^{n\times n}$ is called orthogonal if $M^{-1} = M^T$.*

Proof We only prove (i). We first introduce the dual norm $\|\cdot\|^D$ of a vector norm $\|\cdot\|$ defined as follows,

$$\|y\|^D = \max_{\|x\|=1} |y^* x|.$$

For $\|\cdot\|_2$, we have by the Cauchy-Schwarz inequality,

$$|y^* x| \leq \|y\|_2 \|x\|_2$$

with equality when $x = \frac{1}{\|y\|_2} y$. Therefore, the dual norm of $\|\cdot\|_2$ is given by

$$\|y\|_2^D = \max_{\|x\|_2=1} |y^* x| = \max_{\|x\|_2=1} \|y\|_2 \|x\|_2 = \|y\|_2.$$

So, $\|\cdot\|_2$ is its own dual. Now, we consider

$$\begin{aligned}\|A\|_2 &= \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} \|Ax\|_2^D \\ &= \max_{\|x\|_2=1} \max_{\|y\|_2=1} |(Ax)^* y| \\ &= \max_{\|x\|_2=1} \max_{\|y\|_2=1} |y^* Ax|. \qquad \square\end{aligned}$$

Another useful norm is the Frobenius norm which is defined by

$$\|A\|_F \equiv \left(\sum_{j=1}^{n} \sum_{i=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}}$$

One of the most important properties of $\|\cdot\|_F$ is that for any orthogonal matrices Q and Z,

$$\|A\|_F = \|QAZ\|_F.$$

In the following, we will extend our discussion on norms to the field of $\mathbb{C}$. We remark that from the viewpoint of norms, there is no essential difference between matrices or vectors in the field of $\mathbb{R}$ and matrices or vectors in the field of $\mathbb{C}$.

Definition 3.4 *Let $A \in \mathbb{C}^{n\times n}$. Then the set of all the eigenvalues of A is called the spectrum of A and*

$$\rho(A) = \max\{|\lambda| : \lambda \text{ belongs to the spectrum of } A\}$$

is called the spectral radius of A.

For the relation between the spectral radius and matrix norms, we have

Theorem 3.4 *Let $A \in \mathbb{C}^{n\times n}$. Then*

(i) *For any matrix norm, we have*

$$\rho(A) \le \|A\|.$$

(ii) *For any $\epsilon > 0$, there exists a norm defined on $\mathbb{C}^{n\times n}$ such that*

$$\|A\| \le \rho(A) + \epsilon.$$

Proof For (i), let $x \in \mathbb{C}^n$ satisfy

$$x \neq 0, \qquad Ax = \lambda x, \qquad |\lambda| = \rho(A).$$

Then we have

$$\rho(A)\|xe_1^T\| = \|\lambda xe_1^T\| = \|Axe_1^T\| \leq \|A\| \cdot \|xe_1^T\|.$$

Hence

$$\rho(A) \leq \|A\|.$$

For (ii), by using Theorem 1.1 (Jordan Decomposition Theorem), we know that there is a nonsingular matrix $X \in \mathbb{C}^{n\times n}$ such that

$$X^{-1}AX = \begin{bmatrix} \lambda_1 & \delta_1 & & & \\ & \lambda_2 & \delta_2 & & \\ & & \ddots & \ddots & \\ & & & \lambda_{n-1} & \delta_{n-1} \\ & & & & \lambda_n \end{bmatrix}$$

where $\delta_i = 1$ or 0. For any given $\epsilon > 0$, let

$$D_\epsilon = \operatorname{diag}(1, \epsilon, \epsilon^2, \cdots, \epsilon^{n-1}),$$

then

$$D_\epsilon^{-1}X^{-1}AXD_\epsilon = \begin{bmatrix} \lambda_1 & \epsilon\delta_1 & & & \\ & \lambda_2 & \epsilon\delta_2 & & \\ & & \ddots & \ddots & \\ & & & \lambda_{n-1} & \epsilon\delta_{n-1} \\ & & & & \lambda_n \end{bmatrix}$$

Now, define

$$\|G\|_\epsilon = \|D_\epsilon^{-1}X^{-1}GXD_\epsilon\|_\infty, \qquad G \in \mathbb{C}^{n\times n}.$$

It is easy to see this matrix norm $\|\cdot\|_\epsilon$ actually is induced by the vector norm defined as follows:

$$\|x\|_{XD_\epsilon} = \|(XD_\epsilon)^{-1}x\|_\infty, \qquad x \in \mathbb{C}^n.$$

Therefore,

$$\|A\|_\epsilon = \|D_\epsilon^{-1}X^{-1}AXD_\epsilon\|_\infty = \max_{1\leq i\leq n}(|\lambda_i| + |\epsilon\delta_i|) \leq \rho(A) + \epsilon,$$

where $\delta_n = 0$. □

We remark that for any sequence of matrices $\{A_{(k)}\}$ where $A_{(k)} = [a_{ij}^{(k)}] \in \mathbb{R}^{n\times n}$, and $A = [a_{ij}] \in \mathbb{R}^{n\times n}$,

$$\lim_{k\to\infty} \|A_{(k)} - A\| = 0 \Longleftrightarrow \lim_{k\to\infty} a_{ij}^{(k)} = a_{ij},$$

for $i, j = 1, \cdots, n$.

Theorem 3.5 *Let $A \in \mathbb{C}^{n\times n}$. Then*

$$\lim_{k\to\infty} A^k = 0 \Longleftrightarrow \rho(A) < 1.$$

Proof We first assume that

$$\lim_{k\to\infty} A^k = 0.$$

Let λ be an eigenvalue of A such that $\rho(A) = |\lambda|$. Then λ^k is an eigenvalue of A^k for any k. By Theorem 3.4 (i), we know that for any k,

$$\rho(A)^k = |\lambda|^k = |\lambda^k| \leq \rho(A^k) \leq \|A^k\|.$$

Therefore,

$$\lim_{k\to\infty} \rho(A)^k = 0$$

which implies $\rho(A) < 1$.

Conversely, assume that $\rho(A) < 1$. By Theorem 3.4 (ii), there exists a matrix norm $\|\cdot\|$ such that $\|A\| < 1$. Therefore, we have

$$0 \leq \|A^k\| \leq \|A\|^k \longrightarrow 0, \qquad k \longrightarrow \infty,$$

i.e.,

$$\lim_{k\to\infty} A^k = 0. \qquad \square$$

By using Theorem 3.5, one can easily prove that following important theorem.

Theorem 3.6 *Let $A \in \mathbb{C}^{n\times n}$. Then*

(i) $\sum_{k=0}^{\infty} A^k$ *is convergent if and only if* $\rho(A) < 1$.

(ii) *When* $\sum_{k=0}^{\infty} A^k$ *converges, we have*

$$\sum_{k=0}^{\infty} A^k = (I - A)^{-1}.$$

Moreover, there exists a norm defined on $\mathbb{C}^{n\times n}$ *such that for any* m,

$$\left\|(I - A)^{-1} - \sum_{k=0}^{m} A^k\right\| \leq \frac{\|A\|^{m+1}}{1 - \|A\|}.$$

Corollary 3.1 *Let* $\|\cdot\|$ *be a norm defined on* $\mathbb{C}^{n\times n}$ *with* $\|I\| = 1$ *and* $A \in \mathbb{C}^{n\times n}$ *satisfy* $\|A\| < 1$. *Then* $I - A$ *is nonsingular and satisfies*

$$\|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|}.$$

Proof Just note that

$$\|(I - A)^{-1}\| = \left\|\sum_{k=0}^{\infty} A^k\right\| \leq 1 + \sum_{k=1}^{\infty} \|A\|^k = \frac{1}{1 - \|A\|}. \qquad \square$$

3.2 Perturbation analysis for linear systems

We first consider the following simple example $Ax = b$ given by:

$$\begin{bmatrix} 2.0001 & 1.9999 \\ 1.9999 & 2.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}.$$

The solution of this linear system is $x = (1, 1)^T$. If there is a small perturbation on b, say,

$$\beta = (1 \times 10^{-4}, -1 \times 10^{-4})^T,$$

the system becomes

$$\begin{bmatrix} 2.0001 & 1.9999 \\ 1.9999 & 2.0001 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 4.0001 \\ 3.9999 \end{bmatrix}.$$

The solution of this perturbed system is $\tilde{x} = (1.5, 0.5)^T$. Therefore, we have

$$\frac{\|x - \tilde{x}\|_\infty}{\|x\|_\infty} = \frac{1}{2}, \qquad \frac{\|\beta\|_\infty}{\|b\|_\infty} = \frac{1}{40000},$$

i.e., the relative error of the solution is 20000 times of that of the perturbation on b.

Thus, when we solve a linear system $Ax = b$, a good measurement, which can tell us how sensitive the computed solution is to input small perturbations, is needed. The condition number of matrices is then defined. It relates perturbations of x to perturbations of A and b.

Definition 3.5 *Let $\|\cdot\|$ be any p-norm of matrix and A be a nonsingular matrix. The condition number of A is defined as follows,*

$$\kappa(A) \equiv \|A\| \cdot \|A^{-1}\|. \tag{3.1}$$

Obviously, the condition number depends on the matrix norm used. Since

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\|,$$

it follows that $\kappa(A) \geq 1$. When $\kappa(A)$ is small, then A is said to be well-conditioned, whereas if $\kappa(A)$ is large, then A is said to be ill-conditioned.

Let $\hat{x}$ be an approximation of the exact solution x of $Ax = b$. The error vector is defined as follows,

$$e = x - \hat{x},$$

i.e.,

$$x = \hat{x} + e. \tag{3.2}$$

The absolute error is given by

$$\|e\| = \|x - \hat{x}\|$$

for any vector norm. If $x \neq 0$, then the relative error is defined by

$$\frac{\|e\|}{\|x\|} = \frac{\|x - \hat{x}\|}{\|x\|}.$$

We have by substituting (3.2) into $Ax = b$,

$$A(\hat{x} + e) = A\hat{x} + Ae = b.$$

Therefore,

$$A\hat{x} = b - Ae = \hat{b}.$$

The $\hat{x}$ is the exact solution of $A\hat{x} = \hat{b}$ where $\hat{b}$ is a perturbed vector of b. Since $x = A^{-1}b$ and $\hat{x} = A^{-1}\hat{b}$, we have

$$\|x - \hat{x}\| = \|A^{-1}(b - \hat{b})\| \leq \|A^{-1}\| \cdot \|b - \hat{b}\|. \tag{3.3}$$

Similarly,

$$\|b\| = \|Ax\| \leq \|A\| \cdot \|x\|,$$

i.e.,

$$\frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}. \tag{3.4}$$

Combining (3.3), (3.4) and (3.1), we obtain the following theorem which gives the effect of perturbations of the vector b on the solution of $Ax = b$ in terms of the condition number.

Theorem 3.7 *Let $\hat{x}$ be an approximate solution of the exact solution x of $Ax = b$. Then*

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \kappa(A) \frac{\|b - \hat{b}\|}{\|b\|}.$$

The next theorem includes the effect of perturbations of the coefficient matrix A on the solution of $Ax = b$ in terms of the condition number.

Theorem 3.8 *Let A be a nonsingular matrix and $\widehat{A}$ be a perturbed matrix of A such that*

$$\|A - \widehat{A}\| \cdot \|A^{-1}\| < 1.$$

If $Ax = b$ and $\widehat{A}\hat{x} = \hat{b}$ where $\hat{b}$ is a perturbed vector of b, then

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|A - \widehat{A}\|}{\|A\|}} \left(\frac{\|A - \widehat{A}\|}{\|A\|} + \frac{\|b - \hat{b}\|}{\|b\|} \right).$$

Proof Let

$$E = A - \widehat{A} \quad \text{and} \quad \beta = b - \hat{b}.$$

By subtracting $Ax = b$ from $\widehat{A}\hat{x} = \hat{b}$, we have

$$A(x - \hat{x}) = -E\hat{x} + \beta.$$

Furthermore, we get

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \|A^{-1}E\| \frac{\|\hat{x}\|}{\|x\|} + \|A^{-1}\| \frac{\|Ax\|}{\|x\|} \frac{\|\beta\|}{\|b\|}.$$

By using

$$\|\hat{x}\| \leq \|\hat{x} - x\| + \|x\| \quad \text{and} \quad \|Ax\| \leq \|A\| \cdot \|x\|,$$

we then have

$$\frac{\|x-\hat{x}\|}{\|x\|} \leq \|A^{-1}E\|\frac{\|x-\hat{x}\|}{\|x\|} + \|A^{-1}E\| + \|A^{-1}\|\|A\|\frac{\|\beta\|}{\|b\|},$$

i.e.,

$$(1-\|A^{-1}E\|)\frac{\|x-\hat{x}\|}{\|x\|} \leq \|A^{-1}E\| + \kappa(A)\frac{\|\beta\|}{\|b\|}.$$

Since

$$\|A^{-1}E\| \leq \|A^{-1}\| \cdot \|E\| = \|A^{-1}\| \cdot \|A-\widehat{A}\| < 1,$$

we get

$$\frac{\|x-\hat{x}\|}{\|x\|} \leq (1-\|A^{-1}E\|)^{-1}\left(\|A^{-1}E\| + \kappa(A)\frac{\|\beta\|}{\|b\|}\right).$$

By using

$$\|A^{-1}E\| \leq \|A^{-1}\| \cdot \|E\| = \kappa(A)\frac{\|E\|}{\|A\|},$$

we finally have

$$\frac{\|x-\hat{x}\|}{\|x\|} \leq \frac{\kappa(A)}{1-\kappa(A)\frac{\|E\|}{\|A\|}}\left(\frac{\|E\|}{\|A\|} + \frac{\|\beta\|}{\|b\|}\right). \qquad \square$$

Theorems 3.7 and 3.8 give upper bounds for the relative error of x in terms of the condition number of A. From Theorems 3.7 and 3.8, we know that if A is well-conditioned, i.e., $\kappa(A)$ is small, the relative error in x will be small if the relative errors in both A and b are small.

Corollary 3.2 *Let A be a nonsingular matrix and $A+\widetilde{A}$ be a perturbed matrix of A such that*

$$\|A^{-1}\widetilde{A}\| < 1.$$

Then $A+\widetilde{A}$ is nonsingular and

$$\frac{\|(A+\widetilde{A})^{-1}-A^{-1}\|}{\|A^{-1}\|} \leq \frac{\kappa(A)}{1-\kappa(A)\frac{\|\widetilde{A}\|}{\|A\|}}\frac{\|\widetilde{A}\|}{\|A\|}.$$

Proof We first prove that

$$\|(A+\widetilde{A})^{-1}-A^{-1}\| \leq \frac{\|\widetilde{A}\| \cdot \|A^{-1}\|^2}{1-\|A^{-1}\widetilde{A}\|}.$$

Note that

$$A + \widetilde{A} = A(I + A^{-1}\widetilde{A}) = A[I - (-A^{-1}\widetilde{A})].$$

Let $F = -A^{-1}\widetilde{A}$ and $r = \|A^{-1}\widetilde{A}\|$. Now,

$$(A + \widetilde{A})^{-1} = (I - F)^{-1}A^{-1}.$$

Therefore by noting that $\|F\| = r < 1$ and Corollary 3.1, we have

$$\|(A + \widetilde{A})^{-1}\| = \|(I - F)^{-1}A^{-1}\| \leq \|(I - F)^{-1}\| \cdot \|A^{-1}\| < \frac{\|A^{-1}\|}{1 - \|F\|} = \frac{\|A^{-1}\|}{1 - r}.$$

By using identity

$$B^{-1} = A^{-1} - B^{-1}(B - A)A^{-1},$$

we have,

$$(A + \widetilde{A})^{-1} - A^{-1} = -(A + \widetilde{A})^{-1}\widetilde{A}A^{-1}.$$

Then

$$\|(A + \widetilde{A})^{-1} - A^{-1}\| \leq \|A^{-1}\| \cdot \|\widetilde{A}\| \cdot \|(A + \widetilde{A})^{-1}\| \leq \frac{\|A^{-1}\|^2\|\widetilde{A}\|}{1 - r}.$$

Finally, we obtain

$$\frac{\|(A + \widetilde{A})^{-1} - A^{-1}\|}{\|A^{-1}\|} \leq \frac{\|A^{-1}\| \cdot \|\widetilde{A}\|}{1 - r} \leq \frac{\|A^{-1}\| \cdot \|\widetilde{A}\|}{1 - \|A^{-1}\| \cdot \|\widetilde{A}\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\widetilde{A}\|}{\|A\|}} \frac{\|\widetilde{A}\|}{\|A\|}.$$

□

3.3 Error analysis on floating point arithmetic

In computers, the floating point numbers f are expressed as

$$f = \pm\omega \times \beta^J, \qquad L \leq J \leq U,$$

where β is the base, J is the order, and ω is the fraction. Usually, ω has the following form:

$$\omega = 0.d_1 d_2 \cdots d_t$$

where t is the length (precision) of ω, $d_1 \neq 0$, and $0 \leq d_i < \beta$, for $i = 2, \cdots, t$.

Let

$$\mathcal{F} = \{0\} \cup \{f : f = \pm\omega \times \beta^J,\ 0 \leq d_i < \beta,\ d_1 \neq 0,\ L \leq J \leq U\}.$$

Then $\mathcal{F}$ contains

$$2(\beta - 1)\beta^{t-1}(U - L + 1) + 1$$

floating point numbers. These numbers are symmetrically distributed in the intervals $[m, M]$ and $[-M, -m]$, where

$$m = \beta^{L-1}, \qquad M = \beta^{U}(1 - \beta^{-t}). \tag{3.5}$$

We remark that $\mathcal{F}$ is only a finite set which cannot contain all the real numbers in these two intervals.

Let $fl(x)$ denote the floating point number of any real number x. Then

$$fl(x) = 0, \qquad \text{for } x = 0.$$

If $m \le |x| \le M$, by rounding, $fl(x)$ is the minimum of

$$|fl(x) - x| = \min_{f \in \mathcal{F}} |f - x|.$$

By chopping, $fl(x)$ is the minimum of

$$|fl(x) - x| = \min_{|f| \le |x|} |f - x|.$$

For example, let $\beta = 10$, $t = 3$, $L = 0$ and $U = 2$. We consider the floating point expression of $x = 5.45627$. By rounding, we have $fl(x) = 0.546 \times 10$. By chopping, we have $fl(x) = 0.545 \times 10$. The following theorem gives an estimate of the relative error of floating point expressions.

Theorem 3.9 *Let $m \le |x| \le M$, where m and M are defined by (3.5). Then*

$$fl(x) = x(1 + \delta), \qquad |\delta| \le \mathbf{u},$$

where $\mathbf{u}$ is the machine precision, i.e.,

$$\mathbf{u} = \begin{cases} \frac{1}{2}\beta^{1-t}, & \textit{by rounding,} \\ \beta^{1-t}, & \textit{by chopping.} \end{cases}$$

Proof In the following, we assume that $x \neq 0$ and $x > 0$. Let α be an integer and satisfy

$$\beta^{\alpha-1} \le x < \beta^{\alpha}. \tag{3.6}$$

Since the order of floating point numbers in $[\beta^{\alpha-1}, \beta^{\alpha})$ is α, all the numbers

$$0.d_1 d_2 \cdots d_t \times \beta^{\alpha}$$

are distributed in the interval with distance $\beta^{\alpha-t}$. For the rounding error, by (3.6), we have

$$|fl(x) - x| \le \frac{1}{2}\beta^{\alpha-t} = \frac{1}{2}\beta^{\alpha-1}\beta^{1-t} \le \frac{1}{2}x\beta^{1-t},$$

i.e.,

$$\frac{|fl(x) - x|}{x} \leq \frac{1}{2}\beta^{1-t}.$$

For the chopping error, we have

$$|fl(x) - x| \leq \beta^{\alpha-t} = \beta^{\alpha-1}\beta^{1-t} \leq x\beta^{1-t},$$

i.e.,

$$\frac{|fl(x) - x|}{x} \leq \beta^{1-t}.$$

The proof is complete. □

Let us now consider the rounding error of elementary operations. Let a, $b \in \mathcal{F}$ and "$\circ$" represent any elementary operations: "$+, -, \times, \div$". By Theorem 3.9, we immediately have

Theorem 3.10 *We have*

$$fl(a \circ b) = (a \circ b)(1 + \delta), \qquad |\delta| \leq \mathbf{u}.$$

Theorem 3.11 *If* $|\delta_i| \leq \mathbf{u}$ *and* $n\mathbf{u} \leq 0.01$, *then*

$$1 - 1.01n\mathbf{u} \leq \prod_{i=1}^{n}(1 + \delta_i) \leq 1 + 1.01n\mathbf{u}.$$

Proof Since $|\delta_i| \leq \mathbf{u}$, we have

$$(1 - \mathbf{u})^n \leq \prod_{i=1}^{n}(1 + \delta_i) \leq (1 + \mathbf{u})^n. \tag{3.7}$$

For the lower bound of $(1 - \mathbf{u})^n$, by using the Taylor expansion of $(1 - x)^n$, i.e.,

$$(1 - x)^n = 1 - nx + \frac{n(n-1)}{2}(1 - \theta x)^{n-2}x^2,$$

we have

$$1 - nx \leq (1 - x)^n.$$

Therefore,

$$1 - 1.01n\mathbf{u} \leq 1 - n\mathbf{u} \leq (1 - \mathbf{u})^n. \tag{3.8}$$

Now, we estimate the upper bound of $(1+\mathbf{u})^n$. By using the Taylor expansion of e^x, we have

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \\ &= 1 + x + \frac{x}{2} \cdot x \cdot \left(1 + \frac{x}{3} + \cdots\right). \end{aligned}$$

Therefore, when $0 \le x \le 0.01$, we know that by using $e^{0.01} < 2$,

$$1 + x \le e^x \le 1 + x + \frac{0.01}{2} x e^x \le 1 + 1.01x. \tag{3.9}$$

Let $x = \mathbf{u}$. By the left inequality of (3.9), we have

$$(1 + \mathbf{u})^n \le e^{n\mathbf{u}}. \tag{3.10}$$

Let $x = n\mathbf{u}$. By the right inequality of (3.9), we have

$$e^{n\mathbf{u}} \le 1 + 1.01n\mathbf{u}. \tag{3.11}$$

Combining (3.10) and (3.11), we have

$$(1 + \mathbf{u})^n \le 1 + 1.01n\mathbf{u}. \tag{3.12}$$

By (3.7), (3.8) and (3.12), the proof is complete. □

We consider the following example.

Example 3.1. For given x, $y \in \mathbb{R}^n$, estimate the upper bound of

$$|fl(x^T y) - x^T y|.$$

Let

$$S_k = fl\left(\sum_{i=1}^{k} x_i y_i\right).$$

By Theorem 3.10, we have

$$S_1 = x_1 y_1 (1 + \gamma_1), \qquad |\gamma_1| \le \mathbf{u},$$

and

$$\begin{aligned} S_k &= fl(S_{k-1} + fl(x_k y_k)) \\ &= [S_{k-1} + x_k y_k (1 + \gamma_k)](1 + \delta_k), \qquad |\delta_k|,\ |\gamma_k| \le \mathbf{u}. \end{aligned}$$

Therefore,

$$fl(x^T y) = S_n = \sum_{i=1}^{n} x_i y_i (1+\gamma_i) \prod_{j=i}^{n} (1+\delta_j)$$

$$= \sum_{i=1}^{n} (1+\epsilon_i) x_i y_i,$$

where

$$1+\epsilon_i = (1+\gamma_i) \prod_{j=i}^{n} (1+\delta_j)$$

with $\delta_1 = 0$. Thus, if $n\mathbf{u} \leq 0.01$, we then have by Theorem 3.11,

$$|fl(x^T y) - x^T y| \leq \sum_{i=1}^{n} |\epsilon_i| \cdot |x_i y_i| \leq 1.01 n\mathbf{u} \sum_{i=1}^{n} |x_i y_i|.$$

Before we finish this section, let us briefly discuss the floating point analysis on elementary matrix operations. We first introduce the following notations:

$$|E| = [|e_{ij}|],$$

where $E = [e_{ij}] \in \mathbb{R}^{n \times n}$ and

$$|E| \leq |F| \Longleftrightarrow |e_{ij}| \leq |f_{ij}|$$

for $i, j = 1, 2, \cdots, n$. Let A, $B \in \mathbb{R}^{n \times n}$ be matrices with entries in $\mathcal{F}$, and $\alpha \in \mathcal{F}$. By Theorem 3.10, we have

$$fl(\alpha A) = \alpha A + E, \qquad |E| \leq \mathbf{u}|\alpha A|,$$

and

$$fl(A+B) = (A+B) + E, \qquad |E| \leq \mathbf{u}|A+B|.$$

From Example 3.1, we also have

$$fl(AB) = AB + E, \qquad |E| \leq 1.01 n\mathbf{u}|A| \cdot |B|.$$

Note that $|A| \cdot |B|$ maybe is much larger than $|AB|$. Therefore the relative error of AB may not be small.

3.4 Error analysis on partial pivoting

We will show that if Gaussian elimination with partial pivoting is used to solve $Ax = b$, then the computational solution x satisfies

$$(A+E)x = b,$$

where E is an error matrix. An upper bound of E is also given. We first study the rounding error of the LU factorization of A.

Lemma 3.1 *Let $A \in \mathbb{R}^{n\times n}$ with floating point entries. Assume that A has an LU factorization and $6n\mathbf{u} \le 1$ where $\mathbf{u}$ is the machine precision. Then by using Gaussian elimination, we have*

$$\widetilde{L}\widetilde{U} = A + E$$

where

$$|E| \le 3n\mathbf{u}(|A| + |\widetilde{L}| \cdot |\widetilde{U}|).$$

Proof We use induction on n. Obviously, Lemma 3.1 is true for $n = 1$. Assume that the lemma holds for $n-1$. Now, we consider a matrix $A \in \mathbb{R}^{n\times n}$:

$$A = \begin{bmatrix} \alpha & w^T \\ v & A_1 \end{bmatrix},$$

where $A_1 \in \mathbb{R}^{(n-1)\times(n-1)}$. At the first step of Gaussian elimination, we should compute the vector $\tilde{l}_1 = fl(v/\alpha)$ and modify the matrix A_1 as

$$\widetilde{A}_1 = fl(A_1 - fl(\tilde{l}_1 w^T)).$$

By Theorem 3.10, we have

$$\tilde{l}_1 = v/\alpha + f, \qquad |f| \le \frac{\mathbf{u}}{|\alpha|}|v| \tag{3.13}$$

and

$$\widetilde{A}_1 = A_1 - \tilde{l}_1 w^T + F, \qquad |F| \le (2+\mathbf{u})\mathbf{u}(|A_1| + |\tilde{l}_1| \cdot |w|^T). \tag{3.14}$$

For $\widetilde{A}_1$, by using the assumption, we obtain an LU factorization with a unit lower triangular matrix $\widetilde{L}_1$ and an upper triangular matrix $\widetilde{U}_1$ such that

$$\widetilde{L}_1\widetilde{U}_1 = \widetilde{A}_1 + E_1$$

where

$$|E_1| \le 3(n-1)\mathbf{u}(|\widetilde{A}_1| + |\widetilde{L}_1| \cdot |\widetilde{U}_1|).$$

Thus, we have

$$\widetilde{L}\widetilde{U} = \begin{bmatrix} 1 & \mathbf{0} \\ \tilde{l}_1 & \widetilde{L}_1 \end{bmatrix} \begin{bmatrix} \alpha & w^T \\ \mathbf{0} & \widetilde{U}_1 \end{bmatrix} = A + E,$$

where

$$E = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \alpha f & E_1 + F \end{bmatrix}.$$

By using (3.14), we obtain

$$|\widetilde{A}_1| \le (1 + 2\mathbf{u} + \mathbf{u}^2)(|A_1| + |\tilde{l}_1| \cdot |w|^T).$$

Therefore, by using the condition $6n\mathbf{u} \leq 1$, we have

$$\begin{aligned}
|E_1 + F| &\leq |E_1| + |F| \\
&\leq 3(n-1)\mathbf{u}(|\widetilde{A}_1| + |\widetilde{L}_1| \cdot |\widetilde{U}_1|) + (2+\mathbf{u})\mathbf{u}(|A_1| + |\tilde{l}_1| \cdot |w|^T) \\
&\leq 3(n-1)\mathbf{u}\left((1+2\mathbf{u}+\mathbf{u}^2)(|A_1| + |\tilde{l}_1| \cdot |w|^T) + |\widetilde{L}_1| \cdot |\widetilde{U}_1|\right) \\
&\quad +(2+\mathbf{u})\mathbf{u}(|A_1| + |\tilde{l}_1| \cdot |w|^T) \\
&\leq \mathbf{u}\Big(3n - 1 + [6n + 3(n-1)\mathbf{u} - 5]\mathbf{u}\Big)(|A_1| + |\tilde{l}_1| \cdot |w|^T) \\
&\quad +3(n-1)\mathbf{u}(|\widetilde{L}_1| \cdot |\widetilde{U}_1|) \\
&\leq 3n\mathbf{u}(|A_1| + |\tilde{l}_1| \cdot |w|^T + |\widetilde{L}_1| \cdot |\widetilde{U}_1|).
\end{aligned}$$

Combining with (3.13), we obtain

$$\begin{aligned}
|E| &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ |\alpha||f| & |E_1 + F| \end{bmatrix} \\
&\leq 3n\mathbf{u} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ |v| & |A_1| + |\tilde{l}_1| \cdot |w|^T + |\widetilde{L}_1| \cdot |\widetilde{U}_1| \end{bmatrix} \\
&\leq 3n\mathbf{u} \left(\begin{bmatrix} |\alpha| & |w|^T \\ |v| & |A_1| \end{bmatrix} + \begin{bmatrix} 1 & \mathbf{0} \\ |\tilde{l}_1| & |\widetilde{L}_1| \end{bmatrix} \begin{bmatrix} |\alpha| & |w|^T \\ \mathbf{0} & |\widetilde{U}_1| \end{bmatrix} \right) \\
&= 3n\mathbf{u}(|A| + |\widetilde{L}| \cdot |\widetilde{U}|).
\end{aligned}$$

The proof is complete. □

Corollary 3.3 *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular with floating point entries and $6n\mathbf{u} \leq 1$. Assume that by using Gaussian elimination with partial pivoting, we obtain*

$$\widetilde{L}\widetilde{U} = PA + E$$

where $\widetilde{L} = [l_{ij}]$ is a unit lower triangular matrix with $|l_{ij}| \leq 1$, $\widetilde{U}$ is an upper triangular matrix and P is a permutation matrix. Then E satisfies the following inequality:

$$|E| \leq 3n\mathbf{u}(|PA| + |\widetilde{L}| \cdot |\widetilde{U}|).$$

After we obtain the LU factorization of A, the problem of solving $Ax = b$ becomes the problem of solving the following two triangular systems:

$$\widetilde{L}y = Pb, \qquad \widetilde{U}x = y.$$

Therefore, we need to estimate the rounding error of solving triangular systems.

Lemma 3.2 *Let $S \in \mathbb{R}^{n\times n}$ be a nonsingular triangular matrix with floating point entries and $1.01n\mathbf{u} \leq 0.01$. By using the method proposed in Section 2.1.1 to solve $Sx = b$, we then obtain a computational solution $\tilde{x}$ which satisfies*

$$(S + H)\tilde{x} = b,$$

where

$$|H| \leq 1.01n\mathbf{u}|S|.$$

Proof We use induction on n. Without loss of generality, let $S = L$ be a lower triangular matrix. Obviously, Lemma 3.2 is true for $n = 1$. Assume that the lemma is true for $n - 1$. Now, we consider a lower triangular matrix $L \in \mathbb{R}^{n\times n}$. Let $\tilde{x}$ be the computational solution of $Lx = b$ and we partition L, b and $\tilde{x}$ as follows:

$$L = \begin{bmatrix} l_{11} & \mathbf{0} \\ l_1 & L_1 \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ c \end{bmatrix}, \qquad \tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{y} \end{bmatrix},$$

where c, $\tilde{y} \in \mathbb{R}^{n-1}$ and $L_1 \in \mathbb{R}^{(n-1)\times(n-1)}$. By Theorem 3.10, we have

$$\tilde{x}_1 = fl(b_1/l_{11}) = \frac{b_1}{l_{11}(1+\delta_1)}, \qquad |\delta_1| \leq \mathbf{u}. \tag{3.15}$$

Note that $\tilde{y}$ is the computational solution of the $(n-1)$-by-$(n-1)$ system

$$L_1 y = fl(c - \tilde{x}_1 l_1).$$

By assumption, we have

$$(L_1 + H_1)\tilde{y} = fl(c - \tilde{x}_1 l_1)$$

where

$$|H_1| \leq 1.01(n-1)\mathbf{u}|L_1|. \tag{3.16}$$

By Theorem 3.10 again, we obtain

$$fl(c - \tilde{x}_1 l_1) = fl(c - fl(\tilde{x}_1 l_1)) = (I + D_\gamma)^{-1}(c - \tilde{x}_1 l_1 - \tilde{x}_1 D_\delta l_1),$$

where

$$D_\gamma = \mathrm{diag}(\gamma_2, \cdots, \gamma_n), \qquad D_\delta = \mathrm{diag}(\delta_2, \cdots, \delta_n)$$

with

$$|\gamma_i|,\ |\delta_i| \le \mathbf{u}, \qquad i = 2, \cdots, n.$$

Therefore,

$$\tilde{x}_1 l_1 + \tilde{x}_1 D_\delta l_1 + (I + D_\gamma)(L_1 + H_1)\tilde{y} = c,$$

and then

$$(L + H)\tilde{x} = b,$$

where

$$H = \begin{bmatrix} \delta_1 l_{11} & \mathbf{0} \\ D_\delta l_1 & H_1 + D_\gamma(L_1 + H_1) \end{bmatrix}.$$

By using (3.15), (3.16) and the condition $1.01n\mathbf{u} \le 0.01$, we have

$$\begin{aligned} |H| &\le \begin{bmatrix} |\delta_1| \cdot |l_{11}| & \mathbf{0} \\ |D_\delta| \cdot |l_1| & |H_1| + |D_\gamma|(|L_1| + |H_1|) \end{bmatrix} \\ &\le \begin{bmatrix} \mathbf{u}|l_{11}| & \mathbf{0} \\ \mathbf{u}|l_1| & |H_1| + \mathbf{u}(|L_1| + |H_1|) \end{bmatrix} \\ &\le \mathbf{u} \begin{bmatrix} |l_{11}| & \mathbf{0} \\ |l_1| & [1.01(n-1) + 1 + 1.01(n-1)\mathbf{u}]|L_1| \end{bmatrix} \\ &\le 1.01n\mathbf{u}|L|. \qquad \square \end{aligned}$$

We then have the main theorem of this section.

Theorem 3.12 *Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix with floating point entries and $1.01n\mathbf{u} \le 0.01$. If Gaussian elimination with partial pivoting is used to solve $Ax = b$, we then obtain a computational solution $\tilde{x}$ which satisfies*

$$(A + \delta A)\tilde{x} = b,$$

where

$$\|\delta A\|_\infty \le \mathbf{u}(3n + 5.04n^3\rho)\|A\|_\infty \tag{3.17}$$

with the growth factor

$$\rho \equiv \frac{1}{\|A\|_\infty} \max_{i,j,k} |a_{ij}^{(k)}|.$$

Proof By using Gaussian elimination with partial pivoting, we have the following two triangular systems:

$$\widetilde{L}y = Pb, \qquad \widetilde{U}x = y.$$

By using Lemma 3.2, the computational solution $\tilde{x}$ should satisfy

$$(\widetilde{L} + F)(\widetilde{U} + G)\tilde{x} = Pb,$$

i.e.,

$$(\widetilde{L}\widetilde{U} + F\widetilde{U} + \widetilde{L}G + FG)\tilde{x} = Pb, \tag{3.18}$$

where

$$|F| \leq 1.01n\mathbf{u}|\widetilde{L}|, \qquad |G| \leq 1.01n\mathbf{u}|\widetilde{U}|. \tag{3.19}$$

Substituting $\widetilde{L}\widetilde{U} = PA + E$ into (3.18), we have

$$(A + \delta A)\tilde{x} = b,$$

where

$$\delta A = P^T(E + F\widetilde{U} + \widetilde{L}G + FG).$$

By using (3.19), Corollary 3.3 and the condition $1.01n\mathbf{u} \leq 0.01$, we have

$$\begin{aligned} |\delta A| &\leq P^T(3n\mathbf{u}|PA| + (3n + 2.04n)\mathbf{u}|\widetilde{L}| \cdot |\widetilde{U}|) \\ &= n\mathbf{u}P^T(3|PA| + 5.04|\widetilde{L}| \cdot |\widetilde{U}|). \end{aligned} \tag{3.20}$$

By Corollary 3.3 again, the absolute values of entries in $\widetilde{L}$ are less than or equal to 1. Therefore, we have

$$\|\widetilde{L}\|_\infty \leq n. \tag{3.21}$$

We now define

$$\rho \equiv \frac{1}{\|A\|_\infty} \max_{i,j,k} |a_{ij}^{(k)}|$$

and then we have

$$\|\widetilde{U}\|_\infty \leq n\rho\|A\|_\infty. \tag{3.22}$$

Substituting (3.21) and (3.22) into (3.20), we have (3.17). The proof is complete. □

We remark that $\|\delta A\|_\infty$ usually is very small comparing with the initial error from given data. Thus, Gaussian elimination with partial pivoting is numerically stable.

Exercises:

1. Let
$$A = \begin{bmatrix} 1 & 0.999999 \\ 0.999999 & 1 \end{bmatrix}.$$
Compute A^{-1}, $\det(A)$ and the condition number of A.

2. Prove that $\|AB\|_F \leq \|A\|_2\|B\|_F$ and $\|AB\|_F \leq \|A\|_F\|B\|_2$.

3. Prove that $\|A\|_2^2 \leq \|A\|_1\|A\|_\infty$ for any square matrix A.

4. Show that
$$\left\| \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \right\|_2 \leq \left\| \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \right\|_2.$$

5. Let A be nonsingular. Show that
$$\|A^{-1}\|_2^{-1} = \min_{\|x\|_2=1} \|Ax\|_2.$$

6. Show that if S is real and $S = -S^T$, then $I - S$ is nonsingular and the matrix
$$(I - S)^{-1}(I + S)$$
is orthogonal. This is known as the Cayley transform of S.

7. Prove that if both A and $A + E$ are nonsingular, then
$$\frac{\|(A+E)^{-1} - A^{-1}\|}{\|A^{-1}\|} \leq \|(A+E)^{-1}\| \cdot \|E\|.$$

8. Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and let x, y, $z \in \mathbb{R}^n$ such that $Ax = b$ and $Ay = b + z$. Show that
$$\frac{\|z\|_2}{\|A\|_2} \leq \|x - y\|_2 \leq \|A^{-1}\|_2\|z\|_2.$$

9. Let $A = [a_{ij}]$ be an m-by-n matrix. Define
$$|||A|||_l = \max_{i,j} |a_{ij}|.$$
Is $|||\cdot|||_l$ a matrix norm? Give a reason for your answer.

10. Show that if $X \in \mathbb{C}^{n\times n}$ is nonsingular, then $\|A\|_X = \|X^{-1}AX\|_2$ defines a matrix norm.

11. Let $A = LDL^T \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix and
$$D = \operatorname{diag}(d_{11}, \cdots, d_{nn}).$$
Show that
$$\kappa_2(A) \geq \frac{\max_i\{d_{ii}\}}{\min_i\{d_{ii}\}}.$$

12. Verify that
$$\|xy^*\|_F = \|xy^*\|_2 = \|x\|_2\|y\|_2,$$
for any x, $y \in \mathbb{C}^n$.

13. Show that if $0 \neq v \in \mathbb{R}^n$ and $E \in \mathbb{R}^{n\times n}$, then
$$\left\|E\left(I - \frac{vv^T}{v^Tv}\right)\right\|_F^2 = \|E\|_F^2 - \frac{\|Ev\|_2^2}{v^Tv}.$$

14. Let $A \in \mathbb{R}^{m\times n}$ and $x \in \mathbb{R}^n$. Show that
$$|A| \cdot |x| \leq \tau\|A\|_\infty |x|$$
where $\tau = \max_i |x_i| / \min_i |x_i|$.

15. Prove the Sherman-Morrison-Woodbury formula. Let U, V be n-by-k rectangular matrices with $k \leq n$ and A be an n-by-n matrix. Then
$$T = I + V^TA^{-1}U$$
is nonsingular if and only if $A + UV^T$ is nonsingular. In this case, we have
$$(A + UV^T)^{-1} = A^{-1} - A^{-1}UT^{-1}V^TA^{-1}.$$

Chapter 4
Least Squares Problems

In this chapter, we study linear least squares problems:

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

where the data matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and the observation vector $b \in \mathbb{R}^m$ are given. We introduce some well-known orthogonal transformations and the QR decomposition for constructing efficient algorithms for these problems. For a literature on least squares problems, we refer to [15, 21, 42, 44, 45, 48].

4.1 Least squares problems

In practice, if we are given m points $t_1, t_2, \cdots, t_m$ with data on these points $y_1, y_2, \cdots, y_m$, and functions $\phi_1(t), \phi_2(t), \cdots, \phi_n(t)$ defined on these points, we then try to find $f(x,t)$ defined by

$$f(x,t) \equiv \sum_{j=1}^{n} x_j \phi_j(t)$$

such that residuals defined by

$$r_i(x) \equiv y_i - f(x,t_i) = y_i - \sum_{j=1}^{n} x_j \phi_j(t_i), \qquad i = 1, 2, \cdots, m,$$

can be as small as possible. In matrix form, we have

$$r(x) = b - Ax$$

where

$$A = \begin{bmatrix} \phi_1(t_1) & \cdots & \phi_n(t_1) \\ \vdots & & \vdots \\ \phi_1(t_m) & \cdots & \phi_n(t_m) \end{bmatrix}$$

and

$$b = (y_1, \cdots, y_m)^T, \quad x = (x_1, \cdots, x_n)^T, \quad r(x) = (r_1(x), \cdots, r_m(x))^T.$$

When $m = n$, we can require that $r(x) = 0$ and x can be found by solving the system $Ax = b$. When $m > n$, we require that $r(x)$ can reach its minimum under the norm $\|\cdot\|_2$. We therefore introduce the following definition of the least squares problem.

Definition 4.1 *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Find $x \in \mathbb{R}^n$ such that*

$$\|b - Ax\|_2 = \|r(x)\|_2 = \min_{y\in\mathbb{R}^n} \|r(y)\|_2 = \min_{y\in\mathbb{R}^n} \|b - Ay\|_2. \tag{4.1}$$

It is called the least squares (LS) problem and $r(x)$ is called the residual.

In the following, we only consider the case of

$$\text{rank}(A) = n < m.$$

We first study the solution x of the following equation

$$Ax = b, \qquad A \in \mathbb{R}^{m\times n}. \tag{4.2}$$

The range of matrix A is defined by

$$\mathcal{R}(A) \equiv \{y \in \mathbb{R}^m : y = Ax,\ x \in \mathbb{R}^n\}.$$

It is easy to see that

$$\mathcal{R}(A) = \text{span}\{a_1, \cdots, a_n\}$$

where a_i, $i = 1, \cdots, n$, are column vectors of A. The nullspace of A is defined by

$$\mathcal{N}(A) \equiv \{x \in \mathbb{R}^n : Ax = 0\}.$$

The dimension of $\mathcal{N}(A)$ is denoted by null(A). The orthogonal complement of a subspace $\mathcal{S} \subset \mathbb{R}^n$ is defined by

$$\mathcal{S}^\perp \equiv \{y \in \mathbb{R}^n : y^T x = 0, \text{ for all } x \in \mathcal{S}\}.$$

We have the following theorems for (4.2).

Theorem 4.1 *The equation (4.2) has solutions $\Longleftrightarrow$ rank(A) = rank$([A, b])$.*

Theorem 4.2 *Let x be a special solution of (4.2). Then the solution set of (4.2) is given by $x + \mathcal{N}(A)$.*

Corollary 4.1 *Assume that the equation (4.2) has some solution. The solution is unique* $\Longleftrightarrow$ *null*$(A) = 0$.

We have the following essential theorem for the solution of (4.1).

Theorem 4.3 *The LS problem (4.1) always has solutions. The solution is unique if and only if null*$(A) = 0$.

Proof Since

$$\mathbb{R}^m = \mathcal{R}(A) \oplus \mathcal{R}(A)^{\perp},$$

the vector b can be expressed uniquely by

$$b = b_1 + b_2$$

where $b_1 \in \mathcal{R}(A)$ and $b_2 \in \mathcal{R}(A)^{\perp}$. For any $x \in \mathbb{R}^n$, since $b_1 - Ax \in \mathcal{R}(A)$ and is orthogonal to b_2, we therefore have

$$\begin{aligned} \|r(x)\|_2^2 &= \|b - Ax\|_2^2 = \|(b_1 - Ax) + b_2\|_2^2 \\ &= \|b_1 - Ax\|_2^2 + \|b_2\|_2^2. \end{aligned}$$

Note that $\|r(x)\|_2^2$ reaches the minimum if and only if $\|b_1 - Ax\|_2^2$ reaches the minimum. Since $b_1 \in \mathcal{R}(A)$, $\|r(x)\|_2^2$ reaches its minimum if and only if

$$Ax = b_1,$$

i.e.,

$$\|b_1 - Ax\|_2^2 = 0.$$

Thus, by Corollary 4.1, we know that the solution of $Ax = b_1$ is unique, i.e., the solution of (4.1) is unique, if and only if

$$\text{null}(A) = 0. \qquad \square$$

Let

$$\mathcal{X} = \{x \in \mathbb{R}^n : x \text{ is a solution of (4.1)}\}.$$

We have

Theorem 4.4 *A vector* $x \in \mathcal{X}$ *if and only if*

$$A^T Ax = A^T b. \tag{4.3}$$

Proof Let $x \in \mathcal{X}$. By Theorem 4.3, we know that $Ax = b_1$ where $b_1 \in \mathcal{R}(A)$ and

$$r(x) = b - Ax = b - b_1 = b_2 \in \mathcal{R}(A)^{\perp}.$$

Therefore

$$A^T r(x) = A^T b_2 = 0.$$

Substituting $r(x) = b - Ax$ into $A^T r(x) = 0$, we obtain (4.3).

Conversely, let $x \in \mathbb{R}^n$ satisfy

$$A^T Ax = A^T b,$$

then for any $y \in \mathbb{R}^n$, we have

$$\begin{aligned} \|b - A(x+y)\|_2^2 &= \|b - Ax\|_2^2 - 2y^T A^T (b - Ax) + \|Ay\|_2^2 \\ &= \|b - Ax\|_2^2 + \|Ay\|_2^2 \\ &\geq \|b - Ax\|_2^2. \end{aligned}$$

Thus, $x \in \mathcal{X}$. □

We therefore have the following algorithm for LS problems:

(1) Compute $C = A^T A$ and $d = A^T b$.

(2) Find the Cholesky factorization of $C = LL^T$.

(3) Solve triangular linear systems: $Ly = d$ and $L^T x = y$.

We remark that in computation of $A^T A$, usually, the operation cost is $O(n^2 m)$, and some information of matrix A could be lost. For example, we consider

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}.$$

We have

$$A^T A = \begin{bmatrix} 1+\epsilon^2 & 1 & 1 \\ 1 & 1+\epsilon^2 & 1 \\ 1 & 1 & 1+\epsilon^2 \end{bmatrix}.$$

Assume that $\epsilon = 10^{-3}$ and a 6-digital decimal floating system is used. Then $1 + \epsilon^2 = 1 + 10^{-6}$ is rounded off to be 1, which means that $A^T A$ is singular!

We note that the solution x of (4.3) can be expressed as

$$x = (A^T A)^{-1} A^T b.$$

If we let

$$A^\dagger = (A^T A)^{-1} A^T,$$

then the LS solution x could be written as

$$x = A^\dagger b.$$

Actually, the n-by-m matrix $A^\dagger$ is the Moore-Penrose generalized inverse of A, which is unique, see [14, 17, 42]. In general, we have

Definition 4.2 *Let $X \in \mathbb{R}^{n\times m}$. If it satisfies the following conditions:*

$$AXA = A, \quad XAX = X, \quad (AX)^T = AX, \quad (XA)^T = XA,$$

then X is called the Moore-Penrose generalized inverse of A and denoted by $A^\dagger$.

Now we develop the perturbation analysis of LS problems. Assume that there is a perturbation δb on b and let x, $x + \delta x$ denote the solutions of the following LS problems, respectively,

$$\min_x \|b - Ax\|_2, \qquad \min_x \|(b + \delta b) - Ax\|_2.$$

Then

$$x = A^\dagger b,$$

and

$$x + \delta x = A^\dagger(b + \delta b) = A^\dagger \tilde{b}$$

where $\tilde{b} = b + \delta b$. We have

Theorem 4.5 *Let b_1 and $\tilde{b}_1$ denote orthogonal projections of b and $\tilde{b}$ on $\mathcal{R}(A)$, respectively. If $b_1 \neq 0$, then*

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \kappa_2(A) \frac{\|\delta b_1\|_2}{\|b_1\|_2}$$

where $\kappa_2(A) = \|A\|_2 \|A^\dagger\|_2$ and $\tilde{b}_1 = b_1 + \delta b_1$.

Proof Let b_2 denote the orthogonal projection of b on $\mathcal{R}(A)^\perp$. Then $b = b_1 + b_2$ and $A^T b_2 = 0$. Note that

$$A^\dagger b = A^\dagger b_1 + A^\dagger b_2 = A^\dagger b_1 + (A^T A)^{-1} A^T b_2 = A^\dagger b_1.$$

Similarly, $A^\dagger \tilde{b} = A^\dagger \tilde{b}_1$. Therefore,

$$\begin{aligned} \|\delta x\|_2 &= \|A^\dagger b - A^\dagger \tilde{b}\|_2 = \|A^\dagger (b_1 - \tilde{b}_1)\|_2 \\ &\leq \|A^\dagger\|_2 \|b_1 - \tilde{b}_1\|_2 = \|A^\dagger\|_2 \|\delta b_1\|_2. \end{aligned} \tag{4.4}$$

Since $Ax = b_1$, we have

$$\|b_1\|_2 \leq \|A\|_2 \|x\|_2. \tag{4.5}$$

By combining (4.4) and (4.5), the proof is complete. □

We remark that the condition number $\kappa_2(A)$ is important for LS problems. When $\kappa_2(A)$ is large, we say that the LS problem is ill-conditioned. When $\kappa_2(A)$ is small, we say that the LS problem is well-conditioned.

Theorem 4.6 *Suppose that column vectors of A are linearly independent. Then*

$$\kappa_2(A)^2 = \kappa_2(A^T A).$$

Proof By Theorem 3.3 (ii), and the given condition we have

$$\|A\|_2^2 = \|A^T A\|_2, \qquad \|A^\dagger\|_2^2 = \|A^\dagger (A^\dagger)^T\|_2 = \|(A^T A)^{-1}\|_2.$$

Therefore,

$$\kappa_2(A)^2 = \|A\|_2^2 \|A^\dagger\|_2^2 = \|A^T A\|_2 \|(A^T A)^{-1}\|_2 = \kappa_2(A^T A). \qquad \square$$

4.2 Orthogonal transformations

In order to construct efficient algorithms for solving LS problems, we introduce some well-known orthogonal transformations.

4.2.1 Householder transformation

We first introduce the following definition of Householder transformation.

Definition 4.3 *Let $\omega \in \mathbb{R}^n$ with $\|\omega\|_2 = 1$. Define $H \in \mathbb{R}^{n \times n}$ as follows:*

$$H = I - 2\omega\omega^T. \tag{4.6}$$

The matrix H is called the Householder transformation.

Theorem 4.7 *Let H be defined as in (4.6). Then H has the following properties:*

(i) *H is a symmetric orthogonal matrix.*

(ii) $H^2 = I$.

(iii) *H is called a reflection because Hx is the reflection of $x \in \mathbb{R}^n$ in the plane through 0 perpendicular to ω.*

Proof We only prove (iii). Note that for any vector $x \in \mathbb{R}^n$, it can be expressed as:

$$x = u + \alpha\omega$$

where $u \in \text{span}\{\omega\}^\perp$ and $\alpha \in \mathbb{R}$. By using $u^T\omega = 0$ and $\omega^T\omega = 1$, we have

$$Hx = (I - 2\omega\omega^T)(u + \alpha\omega) = u + \alpha\omega - 2\omega\omega^T u - 2\alpha\omega\omega^T\omega = u - \alpha\omega. \qquad \square$$

Theorem 4.8 *For any $0 \neq x \in \mathbb{R}^n$, we can construct a unit vector ω such that the Householder transformation defined as in (4.6) satisfies*

$$Hx = \alpha e_1$$

where $\alpha = \pm\|x\|_2$.

Proof Note that $Hx = (I - 2\omega\omega^T)x = x - 2(\omega^T x)\omega$. Let

$$\omega = \frac{x - \alpha e_1}{\|x - \alpha e_1\|_2}.$$

We then have

$$\begin{aligned} Hx &= x - 2(\omega^T x)\omega \\ &= x - \frac{2}{\|x - \alpha e_1\|_2^2}\left[(x^T - \alpha e_1^T)x\right](x - \alpha e_1) \\ &= x - \frac{2(\|x\|_2^2 - \alpha x_1)}{\|x - \alpha e_1\|_2^2}x + \frac{2(\|x\|_2^2 - \alpha x_1)\alpha}{\|x - \alpha e_1\|_2^2}e_1 \\ &= \left[1 - \frac{2(\|x\|_2^2 - \alpha x_1)}{\|x - \alpha e_1\|_2^2}\right]x + \frac{2(\|x\|_2^2 - \alpha x_1)\alpha}{\|x - \alpha e_1\|_2^2}e_1, \end{aligned} \tag{4.7}$$

where x_1 is the first component of the vector x. Let the coefficient of x be zero and then we have the following equation:

$$1 - \frac{2(\|x\|_2^2 - \alpha x_1)}{\|x - \alpha e_1\|_2^2} = 0.$$

Solving this equation for α, we have $\alpha = \pm\|x\|_2$. Substituting it into (4.7), we therefore have

$$Hx = \mp\|x\|_2 e_1. \qquad \square$$

We remark that for any vector $0 \neq x \in \mathbb{R}^n$, by Theorem 4.8, one can construct a Householder matrix H such that the last $n-1$ components of Hx are zeros. We can use the following two steps to construct the unit vector ω of H:

(1) compute $v = x \pm \|x\|_2 e_1$;

(2) compute $\omega = v/\|v\|_2$.

Now a natural question is: how to choose the sign in front of $\|x\|_2$? Usually, we choose

$$v = x + \text{sign}(x_1)\|x\|_2 e_1,$$

where $x_1 \neq 0$ is the first component of the vector x, see [38]. Since

$$H = I - 2\omega\omega^T = I - \frac{2}{v^T v} vv^T = I - \beta vv^T$$

where $\beta = 2/v^T v$, we only need to compute β and v instead of forming ω. Thus, we have the following algorithm.

Algorithm 4.1 (Householder transformation)

function: $[v, \beta] =$ **house**(x)
$n = length(x)$
$\sigma = x(2:n)^T x(2:n)$
$v(1) = x(1) + sign(x(1))\sqrt{x(1)^2 + \sigma}$
$v(2:n) = x(2:n)$
if $\sigma = 0$
$\beta = 0$
else
$\beta = 2/(v(1)^2 + \sigma)$
end

4.2.2 Givens rotation

A Givens rotation is defined as follows:

$$G(i,k,\theta) = I + s(e_i e_k^T - e_k e_i^T) + (c-1)(e_i e_i^T + e_k e_k^T)$$

$$= \begin{bmatrix} 1 & & \vdots & & \vdots & & \\ & \ddots & \vdots & & \vdots & & \\ \cdots & \cdots & c & \cdots & s & \cdots & \cdots \\ & & \vdots & & \vdots & & \\ \cdots & \cdots & -s & \cdots & c & \cdots & \cdots \\ & & \vdots & & \vdots & \ddots & \\ & & \vdots & & \vdots & & 1 \end{bmatrix},$$

where $c = \cos\theta$ and $s = \sin\theta$. It is easy to prove that $G(i,k,\theta)$ is an orthogonal matrix.

Let $x \in \mathbb{R}^n$ and $y = G(i,k,\theta)x$. We then have

$$y_i = cx_i + sx_k, \qquad y_k = -sx_i + cx_k, \qquad y_j = x_j, \quad j \neq i,k.$$

If we want to make $y_k = 0$, then we only need to take

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \qquad s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}}.$$

Therefore,

$$y_i = \sqrt{x_i^2 + x_k^2}, \qquad y_k = 0.$$

We remark that for any vector $0 \neq x \in \mathbb{R}^n$, one can construct a Givens rotation $G(i,k,\theta)$ acting on x to make a nonzero component of x be zero.

4.3 QR decomposition

Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. By Theorem 3.3 (iii), for any orthogonal matrix Q, we have

$$\|Ax - b\|_2 = \|Q^T(Ax - b)\|_2.$$

Therefore, the LS problem

$$\min_x \|Q^T Ax - Q^T b\|_2$$

is equivalent to (4.1). We wish that we could find a suitable orthogonal matrix Q such that the original LS problem becomes an easier solvable LS problem. We have

Theorem 4.9 (QR decomposition) *Let $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$. Then A has a QR decomposition:*

$$A = Q \begin{bmatrix} R \\ \mathbf{0} \end{bmatrix}, \tag{4.8}$$

where $Q \in \mathbb{R}^{m\times m}$ is an orthogonal matrix and $R \in \mathbb{R}^{n\times n}$ is an upper triangular matrix with nonnegative diagonal entries. The decomposition is unique when $m = n$ and A is nonsingular.

Proof We use induction. When $n = 1$, we note that it is true by using Theorem 4.8. Now, we assume that the theorem is true for all the matrices in $\mathbb{R}^{p\times(n-1)}$ with $p \geq n-1$. Let the first column of $A \in \mathbb{R}^{m\times n}$ be a_1. By Theorem 4.8 again, there exists an orthogonal matrix $Q_1 \in \mathbb{R}^{m\times m}$ such that

$$Q_1^T a_1 = \|a_1\|_2 e_1.$$

Therefore, we have

$$Q_1^T A = \begin{bmatrix} \|a_1\|_2 & v^T \\ \mathbf{0} & A_1 \end{bmatrix}.$$

For the matrix $A_1 \in \mathbb{R}^{(m-1)\times(n-1)}$, we obtain by assumption,

$$A_1 = Q_2 \begin{bmatrix} R_2 \\ \mathbf{0} \end{bmatrix},$$

where $Q_2 \in \mathbb{R}^{(m-1)\times(m-1)}$ is an orthogonal matrix and R_2 is an upper triangular matrix with nonnegative diagonal entries. Thus, let

$$Q = Q_1 \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & Q_2 \end{bmatrix}, \qquad R = \begin{bmatrix} \|a_1\|_2 & v^T \\ \mathbf{0} & R_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Then Q and R are the matrices satisfying the conditions of the theorem.

When $A \in \mathbb{R}^{m\times m}$ is nonsingular, we want to show that the QR decomposition is unique. Let

$$A = QR = \widetilde{Q}\widetilde{R}$$

where Q, $\widetilde{Q} \in \mathbb{R}^{m\times m}$ are orthogonal matrices, and R, $\widetilde{R} \in \mathbb{R}^{m\times m}$ are upper triangular matrices with nonnegative diagonal entries. Since A is nonsingular, we know that the diagonal entries of R and $\widetilde{R}$ are positive. Therefore, the matrices

$$\widetilde{Q}^T Q = \widetilde{R} R^{-1}$$

are both orthogonal and upper triangular matrices with positive diagonal entries. Thus

$$\widetilde{Q}^T Q = \widetilde{R} R^{-1} = I,$$

i.e.,

$$\widetilde{Q} = Q, \qquad \widetilde{R} = R. \qquad \square$$

A complex version of the QR decomposition is needed later on.

Corollary 4.2 *Let $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$. Then A has a QR decomposition:*

$$A = Q \begin{bmatrix} R \\ \mathbf{0} \end{bmatrix},$$

where $Q \in \mathbb{C}^{m\times m}$ is a unitary matrix and $R \in \mathbb{C}^{n\times n}$ is an upper triangular matrix with nonnegative diagonal entries. The decomposition is unique when $m = n$ and A is nonsingular.

Now we use the QR decomposition to solve the LS problem (4.1). Suppose that $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ has linearly independent columns, $b \in \mathbb{R}^m$, and A has a QR decomposition (4.8). Let Q be partitioned as

$$Q = [Q_1 \; Q_2],$$

and

$$Q^T b = \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

Then

$$\|Ax - b\|_2^2 = \|Q^T Ax - Q^T b\|_2^2 = \|Rx - c_1\|_2^2 + \|c_2\|_2^2.$$

The x is the solution of the LS problem (4.1) if and only if it is the solution of $Rx = c_1$. Note that it is much easier to get the solution of (4.1) by solving $Rx = c_1$ since R is an upper triangular matrix. We have the following algorithm for LS problems:

(1) Compute a QR decomposition of A.

(2) Compute $c_1 = Q_1^T b$.

(3) Solve the upper triangular system $Rx = c_1$.

Finally, we discuss how to use Householder transformations to compute the QR decomposition of A. Let $m = 7$ and $n = 5$. Assume that we have

already found Householder transformations H_1 and H_2 such that

$$H_2H_1A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & + & \times & \times \\ 0 & 0 & + & \times & \times \\ 0 & 0 & + & \times & \times \\ 0 & 0 & + & \times & \times \\ 0 & 0 & + & \times & \times \end{bmatrix}$$

Now we construct a Householder transformation $\widetilde{H}_3 \in \mathbb{R}^{5\times 5}$ such that

$$\widetilde{H}_3 \begin{bmatrix} + \\ + \\ + \\ + \\ + \end{bmatrix} = \begin{bmatrix} \times \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Let $H_3 = \mathrm{diag}(I_2, \widetilde{H}_3)$. We obtain

$$H_3H_2H_1A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

In general, after n such steps, we can reduce the matrix A into the following form,

$$H_nH_{n-1}\cdots H_1A = \begin{bmatrix} R \\ \mathbf{0} \end{bmatrix},$$

where R is an upper triangular matrix with nonnegative diagonal entries. By setting $Q = H_1 \cdots H_n$, we obtain

$$A = Q \begin{bmatrix} R \\ \mathbf{0} \end{bmatrix}.$$

Thus, we have the following algorithm.

Algorithm 4.2 (QR decomposition: Householder transformation)

$$
\begin{cases}
\textbf{for } j = 1 : n \\
\quad [v, \beta] = \textbf{house}(A(j : m, j)) \\
\quad A(j : m, j : n) = (I_{m-j+1} - \beta v v^T) A(j : m, j : n) \\
\quad \textbf{if } j < m \\
\quad\quad A(j+1 : m, j) = v(2 : m - j + 1) \\
\quad \textbf{end} \\
\textbf{end}
\end{cases}
$$

We remark that the QR decomposition is not only a basic tool for solving LS problems but also an important tool for solving some other fundamental problems in NLA.

Exercises:

1. Let $A \in \mathbb{R}^{m \times n}$ have full column rank. Prove that $A + E$ also has full column rank if E satisfies $\|E\|_2 \leq \frac{1}{\|A^\dagger\|_2}$, where $A^\dagger = (A^T A)^{-1} A^T$.

2. Let $U = [u_{ij}]$ be a nonsingular upper triangular matrix. Show that

$$\kappa_\infty(U) \geq \frac{\max\limits_i |u_{ii}|}{\min\limits_i |u_{ii}|},$$

   where $\kappa_\infty(U) = \|U\|_\infty \|U^{-1}\|_\infty$.

3. Let $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and have full column rank. Show that

$$\begin{bmatrix} I & A \\ A^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix}$$

   has a solution where x minimizes $\|Ax - b\|_2$.

4. Let $x \in \mathbb{R}^n$ and P be a Householder transformation such that

$$Px = \pm \|x\|_2 e_1.$$

   Let $G_{12}, G_{23}, \cdots, G_{n-1,n}$ be Givens rotations, and let

$$Q = G_{12} G_{23} \cdots G_{n-1,n}.$$

   Suppose $Qx = \pm \|x\|_2 e_1$. Is P equal to Q? Give a proof or a counterexample.

5. Let $A \in \mathbb{R}^{m \times n}$. Show that $X = A^\dagger$ minimizes $\|AX - I\|_F$ over all $X \in \mathbb{R}^{n \times m}$. What is the minimum?

6. Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{C}^2$. Find an algorithm to compute the following unitary matrix

$$Q = \begin{bmatrix} c & \bar{s} \\ -s & c \end{bmatrix}, \quad c \in \mathbb{R}, \quad c^2 + |s|^2 = 1$$

such that $Qx = \begin{bmatrix} * \\ 0 \end{bmatrix}$.

7. Suppose an m-by-n matrix A has the form

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where A_1 is an n-by-n nonsingular matrix and A_2 is an $(m-n)$-by-n arbitrary matrix. Prove that $\|A^\dagger\|_2 \le \|A_1^{-1}\|_2$.

8. Consider the following well-known ill-conditioned matrix

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}, \qquad |\epsilon| \ll 1.$$

   (a) Choose a small ϵ such that $\operatorname{rank}(A) = 3$. Then compute $\kappa_2(A)$ to show that A is ill-conditioned.

   (b) Find the LS solution with A given as above and $b = (3, \epsilon, \epsilon, \epsilon)^T$ by using

      (i) the normalized equation method;

      (ii) the QR method.

9. Let $A = BC$ where $B \in \mathbb{C}^{m \times r}$ and $C \in \mathbb{C}^{r \times n}$ with

$$r = \operatorname{rank}(A) = \operatorname{rank}(B) = \operatorname{rank}(C).$$

Show that

$$A^\dagger = C^*(CC^*)^{-1}(B^*B)^{-1}B^*.$$

10. Let $A = U\Sigma V^* \in \mathbb{C}^{m \times n}$, where $U \in \mathbb{C}^{m \times n}$ satisfies $U^*U = I$, $V \in \mathbb{C}^{n \times n}$ satisfies $V^*V = I$ and Σ is an n-by-n diagonal matrix. Show that

$$A^\dagger = V\Sigma^\dagger U^*.$$

11. Prove that

$$A^\dagger = \lim_{\epsilon \to 0} (A^*A + \epsilon I)^{-1} A^* = \lim_{\epsilon \to 0} A^*(AA^* + \epsilon I)^{-1}.$$

12. Show that

$$\mathcal{R}(A) \cap \mathcal{N}(A^*) = \{0\}.$$

13. Let $A = [a_{ij}] \in \mathbb{C}^{n\times n}$ be idempotent. Then

$$\mathcal{R}(A) \oplus \mathcal{N}(A) = \mathbb{C}^n, \qquad \text{rank}(A) = \sum_{i=1}^{n} a_{ii}.$$

14. Let $A \in \mathbb{C}^{m\times n}$. Prove that

$$\mathcal{R}(AA^\dagger) = \mathcal{R}(AA^*) = \mathcal{R}(A),$$

$$\mathcal{R}(A^\dagger A) = \mathcal{R}(A^*A) = \mathcal{R}(A^\dagger) = \mathcal{R}(A^*),$$

$$\mathcal{N}(AA^\dagger) = \mathcal{N}(AA^*) = \mathcal{N}(A^\dagger) = \mathcal{N}(A^*),$$

$$\mathcal{N}(A^\dagger A) = \mathcal{N}(A^*A) = \mathcal{N}(A).$$

Therefore $AA^\dagger$ and $A^\dagger A$ are orthogonal projectors.

15. Prove Corollary 4.2.

Chapter 5
Classical Iterative Methods

We study classical iterative methods for the solution of $Ax = b$. Iterative methods, originally proposed by Gauss in 1823, Liouville in 1837, and Jacobi in 1845, are quite different from direct methods such as Gaussian elimination, see [2].

Direct methods based on an LU factorization of A become prohibitive in terms of computing time and computer storage if the matrix A is quite large. In some practical situation such as the discretization of partial differential equations, the matrix size can be as large as several hundreds of thousands. For such problems, direct methods become impractical. Furthermore, most large problems are sparse, and usually the sparsity is lost during LU factorizations. Therefore, we have to face a very large matrix with many nonzero entries at the end of LU factorizations, and then the storage becomes a crucial issue. For such kind of problems, we can use a class of methods called iterative methods. In this chapter, we only consider some classical iterative methods.

We remark that the disadvantage with classical iterative methods is that the convergence rate maybe is slow or they may even diverge, and a stopping criterion is needed to be found.

5.1 Jacobi and Gauss-Seidel method

5.1.1 Jacobi method

Consider the following linear system

$$Ax = b$$

where $A = [a_{ij}] \in \mathbb{R}^{n \times n}$. We can write the matrix A in the following form

$$A = D - L - U,$$

where

$$D = \text{diag}(a_{11}, a_{22}, \cdots, a_{nn}),$$

$$L = \begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & 0 & -a_{n-1,n} \\ & & & & 0 \end{bmatrix}$$

Then it is easy to see that

$$x = B_J x + g,$$

where

$$B_J = D^{-1}(L+U), \qquad g = D^{-1}b.$$

The matrix B_J is called the Jacobi iteration matrix. The corresponding iteration

$$x_k = B_J x_{k-1} + g, \qquad k = 1, 2, \cdots, \tag{5.1}$$

is known as the Jacobi method if an initial vector $x_0 = \left(x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)}\right)^T$ is given.

5.1.2 Gauss-Seidel method

In the Jacobi method, to compute the components of the vector

$$x_{k+1} = \left(x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_n^{(k+1)}\right)^T,$$

only the components of the vector x_k are used. However, note that to compute $x_i^{(k+1)}$, we could use $x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$, which were already available for us. Thus a natural modification of the Jacobi method is to rewrite the Jacobi iteration (5.1) in the following form

$$x_k = (D-L)^{-1}Ux_{k-1} + (D-L)^{-1}b, \qquad k = 1, 2, \cdots. \tag{5.2}$$

The idea is to use each new component as soon as it is available in the computation of the next component. The iteration (5.2) is known as the Gauss-Seidel method.

Note that the matrix $D - L$ is a lower triangular matrix with $a_{11}, \cdots, a_{nn}$ on the diagonal. Because these entries are assumed to be nonzero, the matrix $D - L$ is nonsingular. The matrix

$$B_{GS} = (D - L)^{-1}U$$

is called the Gauss-Seidel iteration matrix.

5.2 Convergence analysis

5.2.1 Convergence theorems

It is often hard to make a good initial approximation x_0. Thus, it will be nice to have conditions that will guarantee the convergence of Jacobi, Gauss-Seidel methods for any arbitrary choice of the initial approximation.

Both of the Jacobi iteration and the Gauss-Seidel iteration can be expressed by

$$x_{k+1} = Bx_k + g, \qquad k = 0, 1, \cdots. \tag{5.3}$$

For the Jacobi iteration, we have

$$B_J = D^{-1}(L + U), \qquad g = D^{-1}b;$$

and for the Gauss-Seidel iteration, we have

$$B_{GS} = (D - L)^{-1}U, \qquad g = (D - L)^{-1}b.$$

The iteration (5.3) is called linear stationary iteration, where $B \in \mathbb{R}^{n\times n}$ is called the iteration matrix, $g \in \mathbb{R}^n$ the constant term, and $x_0 \in \mathbb{R}^n$ the initial vector. In the following, we give a convergence theorem.

Theorem 5.1 *The iteration (5.3) converges with an arbitrary initial guess x_0 if and only if the matrix $B^k \to 0$ as $k \to \infty$.*

Proof From $x = Bx + g$ and $x_{k+1} = Bx_k + g$, we have

$$x - x_{k+1} = B(x - x_k). \tag{5.4}$$

Because it is true for any value of k, we can write

$$x - x_k = B(x - x_{k-1}). \tag{5.5}$$

Substituting (5.5) into (5.4), we have

$$x - x_{k+1} = B^2(x - x_{k-1}).$$

Continuing this process k times, we can write

$$x - x_{k+1} = B^{k+1}(x - x_0).$$

This shows that $\{x_k\}$ converges to the solution x for any choice x_0 if and only if $B^k \to 0$ as $k \to \infty$. □

Recall that $B^k \to 0$ as $k \to \infty$ if and only if the spectral radius $\rho(B) < 1$. Since $|\lambda_i| \leq \|B\|$, a good way to see whether $\rho(B) < 1$ is to see whether $\|B\| < 1$ by computing $\|B\|$ with a row-sum or column-sum norm. Note that the converse is not true. Combining the result of Theorem 5.1 with the above observation, we have the following theorem.

Theorem 5.2 *The iteration (5.3) converges for any choice of x_0 if and only if $\rho(B) < 1$. Moreover, if $\|B\| < 1$ for some matrix norm, then the iteration (5.3) converges.*

Let us consider the examples:

$$A_1 = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & -2 \end{bmatrix}.$$

It is easy to verify that for A_1, the Jacobi method converges even if the Gauss-Seidel method does not. For A_2, the Jacobi method diverges while the Gauss-Seidel method converges.

5.2.2 Sufficient conditions for convergence

We now apply Theorem 5.2 to give a sequence of criteria that guarantee the convergence of the Jacobi and (or) Gauss-Seidel methods with any choice of initial approximation x_0.

Definition 5.1 *Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$. If*

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \qquad i = 1, 2, \cdots, n,$$

then the matrix A is called strictly diagonally dominant. The matrix A is called weakly diagonally dominant if for $i = 1, 2, \cdots, n$,

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

with at least one strict inequality.

Theorem 5.3 *If A is strictly diagonally dominant, then the Jacobi method converges for any initial approximation x_0.*

Proof Because $A = [a_{ij}]$ is strictly diagonally dominant, we have

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad i = 1, 2, \cdots, n.$$

Recall that the Jacobi iteration matrix

$$B_J = D^{-1}(L + U)$$

is given by

$$B_J = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & -\frac{a_{n-1,n}}{a_{n-1,n-1}} \\ -\frac{a_{n1}}{a_{nn}} & \cdots & \cdots & \frac{a_{n,n-1}}{a_{nn}} & 0 \end{bmatrix}.$$

We know that the absolute row sum of each row is less than 1, which means

$$\|B_J\|_\infty < 1.$$

Thus by Theorem 5.2, the Jacobi method converges. □

Theorem 5.4 *If A is strictly diagonally dominant, then the Gauss-Seidel method converges for any initial approximation x_0.*

Proof The Gauss-Seidel iteration matrix is given by

$$B_{GS} = (D - L)^{-1}U.$$

Let λ be an eigenvalue of this matrix and $x = (x_1, x_2, \cdots, x_n)^T$ be the corresponding eigenvector with the largest component having the magnitude 1. Then from

$$B_{GS}x = \lambda x,$$

we have

$$Ux = \lambda(D - L)x,$$

i.e.,

$$-\sum_{j=i+1}^{n} a_{ij}x_j = \lambda \sum_{j=1}^{i} a_{ij}x_j, \qquad 1 \le i \le n,$$

which can be rewritten as

$$\lambda a_{ii}x_i = -\lambda \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j, \qquad 1 \le i \le n. \tag{5.6}$$

Let x_k be the largest component having the magnitude 1 of the vector x. Then by (5.6), we have

$$|\lambda| \cdot |a_{kk}| \le |\lambda| \sum_{j=1}^{k-1} |a_{kj}| + \sum_{j=k+1}^{n} |a_{kj}|$$

i.e.,

$$|\lambda|(|a_{kk}| - \sum_{j=1}^{k-1} |a_{kj}|) \le \sum_{j=k+1}^{n} |a_{kj}|$$

or

$$|\lambda| \le \frac{\sum_{j=k+1}^{n} |a_{kj}|}{|a_{kk}| - \sum_{j=1}^{k-1} |a_{kj}|} \tag{5.7}$$

Since A is strictly diagonally dominant, we have

$$|a_{kk}| - \sum_{j=1}^{k-1} |a_{kj}| > \sum_{j=k+1}^{n} |a_{kj}|.$$

Thus from (5.7), we conclude that $|\lambda| < 1$, i.e., $\rho(B_{GS}) < 1$. By Theorem 5.2, the Gauss-Seidel method converges. □

We now discuss the convergence of the Jacobi, Gauss-Seidel methods for the symmetric positive definite matrices.

Theorem 5.5 *Let A be symmetric with diagonal entries $a_{ii} > 0$, $i = 1, 2, \cdots, n$. Then the Jacobi method converges if and only if both A and $2D - A$ are positive definite.*

Proof Since

$$B_J = D^{-1}(L+U) = D^{-1}(D-A) = I - D^{-1}A,$$

and

$$D = \mathrm{diag}(a_{11}, a_{22}, \cdots, a_{nn})$$

with $a_{ii} > 0$, $i = 1, 2, \cdots, n$, then

$$B_J = I - D^{-1}A = D^{-1/2}(I - D^{-1/2}AD^{-1/2})D^{1/2}.$$

It is easy to see that $I - D^{-1/2}AD^{-1/2}$ symmetric and similar to B_J. Then the eigenvalues of B_J are real.

Now, we first suppose that the Jacobi method converges, then $\rho(B_J) < 1$ by Theorem 5.2. The absolute value of the eigenvalues of

$$I - D^{-1/2}AD^{-1/2}$$

is less than 1, i.e., the eigenvalues of $D^{-1/2}AD^{-1/2}$ lies on $(0, 2)$. Thus A is positive definite. On the other hand, the eigenvalues of $2I - D^{-1/2}AD^{-1/2}$ are positive, so the matrix

$$2I - D^{-1/2}AD^{-1/2}$$

is positive definite. Since

$$D^{-1/2}(2D - A)D^{-1/2} = 2I - D^{-1/2}AD^{-1/2},$$

we know that $2D - A$ is positive definite too.

Conversely, since

$$D^{1/2}(I - B_J)D^{-1/2} = D^{-1/2}AD^{-1/2}$$

and A is positive definite, it follows that the eigenvalues of $I - B_J$ are positive, i.e., the eigenvalues of B_J are less than 1. Because $2D - A$ is positive definite and

$$D^{-1/2}(2D - A)D^{-1/2} = D^{1/2}(I + B_J)D^{-1/2},$$

we can deduce that the eigenvalues of $I + B_J$ are positive, i.e., the eigenvalues of B_J are greater than -1. Thus $\rho(B_J) < 1$. By Theorem 5.2 again, the Jacobi method converges. □

Theorem 5.6 *Let A be a symmetric positive definite matrix. Then the Gauss-Seidel method converges for any initial approximation x_0.*

Proof Let λ be an eigenvalue of the iteration matrix B_{GS} and u the corresponding eigenvector. Then

$$(D - L)^{-1}Uu = \lambda u.$$

Since A is symmetric positive definite, we have $U = L^T$ and

$$\lambda(D - L)u = L^T u.$$

Therefore,

$$\lambda u^*(D - L)u = u^* L^T u.$$

Let

$$u^* Du = \delta, \qquad u^* Lu = \alpha + \mathbf{i}\beta.$$

We then have

$$u^* L^T u = (Lu)^* u = \overline{u^* Lu} = \alpha - \mathbf{i}\beta.$$

Thus

$$\lambda[\delta - (\alpha + \mathbf{i}\beta)] = \alpha - \mathbf{i}\beta.$$

Taking the modulus of both sides, we have

$$|\lambda|^2 = \frac{\alpha^2 + \beta^2}{(\delta - \alpha)^2 + \beta^2}.$$

On the other hand,

$$0 < u^* Au = u^*(D - L - L^T)u = \delta - 2\alpha.$$

Hence,

$$\begin{aligned}(\delta - \alpha)^2 + \beta^2 &= \delta^2 + \alpha^2 + \beta^2 - 2\delta\alpha \\ &= \delta(\delta - 2\alpha) + \alpha^2 + \beta^2 \\ &> \alpha^2 + \beta^2.\end{aligned}$$

So we get $|\lambda| < 1$. By Theorem 5.2, the Gauss-Seidel method converges. □

Definition 5.2 *A matrix A is called irreducible if there is no permutation matrix P such that*

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ \mathbf{0} & A_{22} \end{bmatrix},$$

where A_{11} and A_{22} are square matrices.

We have the following lemma, see [41].

Lemma 5.1 *If a matrix A is strictly diagonally dominant; or irreducible and weakly diagonally dominant, then A is nonsingular.*

Furthermore,

Theorem 5.7 *We have*

(i) *If A is strictly diagonally dominant, then both the Jacobi and the Gauss-Seidel methods converge. In fact,*

$$\|B_{GS}\|_\infty \leq \|B_J\|_\infty < 1.$$

(ii) *If A is irreducible and weakly diagonally dominant, then both the Jacobi and the Gauss-Seidel methods converge. Moreover,*

$$\rho(B_{GS}) < \rho(B_J) < 1.$$

For the proof of this theorem, we refer to [14, 41].

5.3 Convergence rate

We consider a stationary linear iteration method,

$$x_{k+1} = Mx_k + g, \qquad k = 0, 1, \cdots,$$

where M is an n-by-n matrix. If $I - M$ is nonsingular, then there exists a unique solution of

$$(I - M)x_* = g.$$

The error vectors y_k are defined as $y_k = x_k - x_*$, for $k = 0, 1, \cdots$, and then

$$y_k = My_{k-1} = \cdots = M^k y_0.$$

Using matrix and vector norms, we have

$$\|y_k\| \leq \|M^k\| \, \|y_0\|$$

with the equality possible for each k for some vector y_0. Thus, if y_0 is not the zero vector, then $\|M^k\|$ gives us a sharp upper-bound estimate for the ratio $\|y_k\|/\|y_0\|$. Since the initial vector y_0 is unknown in practical problems, $\|M^k\|$ serves as a measurement of comparison of different iterative methods.

Definition 5.3 *Let M be an n-by-n iteration matrix. If $\|M^k\| < 1$ for some positive integer k, then*

$$R_k(M) \equiv -\ln[(\|M^k\|)^{1/k}] = -\frac{\ln \|M^k\|}{k}$$

is called the average rate of convergence for k iterations of M.

In terms of actual computations, the significance of the average rate of convergence $R_k(M)$ is given as follows. Clearly, the quantity

$$\sigma = \left(\frac{\|y_k\|}{\|y_0\|}\right)^{1/k}$$

is the average reduction factor per iteration for the norm of error. If $\|M^k\| < 1$, then by Definition 5.3, we have

$$\sigma \leq (\|M^k\|)^{1/k} = e^{-R_k(M)}$$

where e is the base of the natural logarithm.

If M is symmetric (or Hermitian, or normal, i.e., $MM^* = M^*M$), by using the spectral radius of the iteration matrix M, we then have

$$\|M^k\|_2 = [\rho(M)]^k,$$

and thus,

$$R_k(M) = -\ln \rho(M),$$

which is independent of k.

Next we will consider the asymptotic convergence rate

$$R_\infty(M) \equiv \lim_{k\to\infty} R_k(M).$$

Theorem 5.8 *We have*

$$R_\infty(M) = -\ln \rho(M).$$

Proof We only need to prove that

$$\lim_{k\to\infty} \|M^k\|^{1/k} = \rho(M).$$

Since

$$[\rho(M)]^k = \rho(M^k) \leq \|M^k\|,$$

we have

$$\rho(M) \leq \|M^k\|^{1/k}.$$

On the other hand, for any $\epsilon > 0$, consider the matrix

$$B_\epsilon = \frac{1}{\rho(M) + \epsilon} M.$$

It is obvious that $\rho(B_\epsilon) < 1$ and then $\lim_{k\to\infty} B_\epsilon^k = 0$. Hence, there exists a natural number K, for $k \geq K$, we have

$$\|B_\epsilon^k\| \leq 1,$$

i.e.,

$$\|M^k\| \leq [\rho(M) + \epsilon]^k.$$

Thus,

$$\rho(M) \leq \|M^k\|^{1/k} \leq \rho(M) + \epsilon,$$

which means

$$\lim_{k\to\infty} \|M^k\|^{1/k} = \rho(M). \qquad \square$$

5.4 SOR method

The Gauss-Seidel method is very slow when $\rho(B_{GS})$ is close to unity. However, the convergence rate of the Gauss-Seidel iteration, in certain cases, can be improved by introducing a parameter ω, known as the relaxation parameter. This method is called the successive overrelaxation (SOR) method.

5.4.1 Iterative form

Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ and $A = D - L - U$ defined as in Section 5.1.1. Consider the solution of $Ax = b$ again. The motivation of the SOR method is to improve the Gauss-Seidel iteration by taking an appropriately weighted average of the $x_i^{(k)}$ and $x_i^{(k+1)}$ yielding the following algorithm

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega\left(\sum_{j=1}^{i-1} c_{ij}x_j^{(k+1)} + \sum_{j=i+1}^{n} c_{ij}x_j^{(k)} + g_i\right). \tag{5.8}$$

Here $D^{-1}(L+U) = [c_{ij}]$, $x_k = \left(x_1^{(k)}, x_2^{(k)}, \cdots, x_n^{(k)}\right)^T$, and $g = D^{-1}b = (g_1, g_2, \cdots, g_n)^T$. In matrix form, we have

$$x_{k+1} = L_\omega x_k + \omega(D - \omega L)^{-1}b$$

where

$$L_\omega = (D - \omega L)^{-1}[(1-\omega)D + \omega U]$$

is called the iteration matrix of the SOR method and ω is the relaxation parameter.

We have three cases depending on the values of ω:

(1) if $\omega = 1$, (5.8) is equivalent to the Gauss-Seidel method;

(2) if $\omega < 1$, (5.8) is called underrelaxation;

(3) if $\omega > 1$, (5.8) is called overrelaxation.

5.4.2 Convergence criteria

It is natural to ask for which range of ω the SOR iteration converges. To this end, we first prove the following important result, see [14, 41].

Theorem 5.9 *The SOR iteration cannot converge for any initial approximation if ω lies outside the interval (0,2).*

Proof Recall that the SOR iteration matrix L_ω is given by

$$L_\omega = (D - \omega L)^{-1}[(1-\omega)D + \omega U],$$

where $A = [a_{ij}] = D - L - U$.

The matrix $(D - \omega L)^{-1}$ is a lower triangular matrix with $1/a_{ii}$, $i = 1, 2, \cdots, n$, as diagonal entries, and the matrix

$$(1-\omega)D + \omega U$$

is an upper triangular matrix with $(1-\omega)a_{ii}$, $i = 1, 2, \cdots, n$, as diagonal entries. Therefore,

$$\det(L_\omega) = (1-\omega)^n.$$

Since the determinant of a matrix is equal to the product of its eigenvalues, we conclude that

$$\rho(L_\omega) \geq |1-\omega|,$$

where $\rho(L_\omega)$ is the spectral radius of L_ω. By Theorem 5.2, the spectral radius of the iteration matrix should be less than 1 for convergence. We then conclude that $0 < \omega < 2$ is required for the convergence of the SOR method. □

Theorem 5.10 *If A is symmetric positive definite, then*

$$\rho(L_\omega) < 1$$

for all $0 < \omega < 2$.

Proof Let λ be any eigenvalue of the SOR iteration matrix

$$L_\omega = (D - \omega L)^{-1}[(1-\omega)D + \omega L^T]$$

and x be the corresponding eigenvector. We have

$$[(1-\omega)D + \omega L^T]x = \lambda(D - \omega L)x$$

or

$$x^*[(1-\omega)D + \omega L^T]x = \lambda x^*(D - \omega L)x.$$

Let

$$x^*Dx = \delta, \qquad x^*Lx = \alpha + \mathbf{i}\beta.$$

Therefore, $x^*L^Tx = \alpha - \mathbf{i}\beta$ and then

$$(1-\omega)\delta + \omega(\alpha - \mathbf{i}\beta) = \lambda[\delta - \omega(\alpha + \mathbf{i}\beta)].$$

Taking modulus of both sides, we obtain

$$|\lambda|^2 = \frac{[(1-\omega)\delta + \omega\alpha]^2 + \omega^2\beta^2}{(\delta - \omega\alpha)^2 + \omega^2\beta^2}. \tag{5.9}$$

Note that

$$\begin{aligned}
&[(1-\omega)\delta + \omega\alpha]^2 + \omega^2\beta^2 - (\delta - \omega\alpha)^2 - \omega^2\beta^2 \\
&= [\delta - \omega(\delta - \alpha)]^2 - (\delta - \omega\alpha)^2 \\
&= \omega\delta(\delta - 2\alpha)(\omega - 2).
\end{aligned}$$

Since A is symmetric positive definite, we have

$$\delta > 0, \qquad \delta - 2\alpha > 0.$$

Therefore, if $0 < \omega < 2$, we have

$$[(1-\omega)\delta + \omega\alpha]^2 + \omega^2\beta^2 < (\delta - \omega\alpha)^2 + \omega^2\beta^2. \tag{5.10}$$

Thus, for $0 < \omega < 2$, we obtain by (5.9) and (5.10),

$$|\lambda|^2 < 1,$$

i.e., the SOR method converges. □

5.4.3 Optimal ω in SOR iteration

For further comparison of the Jacobi, Gauss-Seidel and SOR methods, we impose another condition on matrices. This condition allows us to compute $\rho(B_{GS})$ and $\rho(L_\omega)$ explicitly in terms of $\rho(B_J)$.

Definition 5.4 *A matrix M has property A if there exists a permutation P such that*

$$PMP^T = \begin{bmatrix} D_{11} & M_{12} \\ M_{21} & D_{22} \end{bmatrix}$$

where D_{11} and D_{22} are diagonal matrices.

If M has property A, then we can write

$$PMP^T = \widehat{D} - \widehat{L} - \widehat{U}$$

where

$$\widehat{D} = \begin{bmatrix} D_{11} & \mathbf{0} \\ \mathbf{0} & D_{22} \end{bmatrix}, \qquad \widehat{L} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -M_{21} & \mathbf{0} \end{bmatrix}, \qquad \widehat{U} = \begin{bmatrix} \mathbf{0} & -M_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Let

$$\widehat{B_J}(\alpha) \equiv \alpha \widehat{D}^{-1}\widehat{L} + \frac{1}{\alpha}\widehat{D}^{-1}\widehat{U}.$$

We have

Theorem 5.11 *The eigenvalues of $\widehat{B_J}(\alpha)$ are independent of α.*

Proof Just note that the matrix

$$\widehat{B_J}(\alpha) = -\begin{bmatrix} \mathbf{0} & \frac{1}{\alpha}D_{11}^{-1}M_{12} \\ \alpha D_{22}^{-1}M_{21} & \mathbf{0} \end{bmatrix}$$

is similar to the matrix

$$\begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \alpha I \end{bmatrix}^{-1} \widehat{B_J}(\alpha) \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \alpha I \end{bmatrix} = -\begin{bmatrix} \mathbf{0} & D_{11}^{-1}M_{12} \\ D_{22}^{-1}M_{21} & \mathbf{0} \end{bmatrix} = \widehat{B_J}(1). \qquad \square$$

Definition 5.5 *Let $M = D - L - U$ and*

$$B_J(\alpha) = \alpha D^{-1}L + \frac{1}{\alpha}D^{-1}U.$$

If the eigenvalues of $B_J(\alpha)$ are independent of α, then M is called consistent ordering (or said to be consistently ordered).

Note that $B_J(1) = B_J$ is the Jacobi iteration matrix. From Theorem 5.11, we have known that if M has property A, then PMP^T is consistently ordered where P is a permutation matrix such that

$$PMP^T = \begin{bmatrix} D_{11} & M_{12} \\ M_{21} & D_{22} \end{bmatrix}$$

with D_{11} and D_{22} being diagonal. It is not true that consistent ordering implies property A.

Example 5.1. Any block tridiagonal matrix

$$\begin{bmatrix} D_1 & A_1 & & \\ B_1 & \ddots & \ddots & \\ & \ddots & \ddots & A_{n-1} \\ & & B_{n-1} & D_n \end{bmatrix}$$

is consistently ordered when D_i are diagonal.

The following theorem gives a relation between the eigenvalues of B_J and the eigenvalues of L_ω, see [14, 41].

Theorem 5.12 *If A is consistently ordered and $\omega \neq 0$, then*

(i) *The eigenvalues of B_J appear in $\pm$ pairs.*

(ii) *If μ is an eigenvalue of B_J and λ satisfies*

$$(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2, \tag{5.11}$$

then λ is an eigenvalue of L_ω.

(iii) *If $\lambda \neq 0$ is an eigenvalue of L_ω and μ satisfies (5.11), then μ is an eigenvalue of B_J.*

Corollary 5.1 *If A is consistently ordered, then*

$$\rho(B_{GS}) = [\rho(B_J)]^2.$$

This means that the convergence rate of the Gauss-Seidel method is twice as fast as that of the Jacobi method.

Proof The choice of $\omega = 1$ is equivalent to the Gauss-Seidel method. Therefore, by (5.11), we have $\lambda^2 = \lambda\mu^2$ and then

$$\lambda = \mu^2. \qquad \square$$

To get the most benefit from overrelaxation, we would like to find an optimal ω, denoted by ω_{opt}, minimizing $\rho(L_\omega)$. We have the following theorem, see [14, 41].

Theorem 5.13 *Suppose that A is consistently ordered and B_J has real eigenvalues with $\mu = \rho(B_J) < 1$. Then*

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \mu^2}},$$

$$\rho(L_{\omega_{opt}}) = \frac{\mu^2}{(1 + \sqrt{1 - \mu^2})^2},$$

and

$$\rho(L_\omega) = \begin{cases} \omega - 1, & \omega_{opt} \le \omega < 2, \\ 1 - \omega + \frac{1}{2}\omega^2\mu^2 + \omega\mu\sqrt{1 - \omega + \frac{1}{4}\omega^2\mu^2}, & 0 < \omega \le \omega_{opt}. \end{cases}$$

Exercises:

1. Judge the convergence of the Jacobi method and the Gauss-Seidel method for the following examples:

$$A_1 = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & -2 \end{bmatrix}.$$

2. Show that the Jacobi method converges for 2-by-2 symmetric positive definite systems.

3. Show that if $A = M - N$ is singular, then we can never have $\rho(M^{-1}N) < 1$ even if M is nonsingular.

4. Consider $Ax = b$ where

$$A = \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & 0 \\ \alpha & 0 & 1 \end{bmatrix}.$$

(1) For which α, is A positive definite?

(2) For which α, does the Jacobi method converge?

(3) For which α, does the Gauss-Seidel method converge?

5. Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. Show that there exists a permutation matrix P such that the diagonal entries of PA are nonzero.

6. Prove that

$$A = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}$$

is consistently ordered.

7. Let $A \in \mathbb{R}^{n\times n}$. Then $\rho(A) < 1$ if and only if $I - A$ nonsingular and each eigenvalue of $(I - A)^{-1}(I + A)$ has a positive real part.

8. Let $B \in \mathbb{R}^{n\times n}$ satisfy $\rho(B) = 0$. Show that for any g, $x_0 \in \mathbb{R}^n$, the iterative formula

$$x_{k+1} = Bx_k + g, \qquad k = 0, 1, \cdots$$

converges to the exact solution of $x = Bx + g$ for at most n iterations.

9. If $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ is irreducible with $a_{ij} \geq 0$ and $B = [b_{ij}] \in \mathbb{R}^{n\times n}$ with $b_{ij} \geq 0$. Show that $A + B$ is irreducible.

10. Let

$$A = \begin{bmatrix} 0 & a & 0 \\ 0 & 0 & b \\ c & 0 & 0 \end{bmatrix}.$$

Is A^k irreducible $(k = 1, 2, 3)$?

11. Let

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}.$$

Show that

(1)

$$A^k = k \begin{bmatrix} 1 + 1/k & -1 \\ 1 & 1/k - 1 \end{bmatrix}, \qquad k = 1, 2, \cdots.$$

(2)

$$\lim_{k\to\infty} \frac{\|A^k\|_p}{k} = 2, \qquad p = 1, 2, \infty.$$

(3) $\rho(A^k) = 1$.

12. Let

$$B_k = B_{k-1} + B_{k-1}(I - AB_{k-1}), \quad k = 1, 2, \cdots.$$

Show that if $\|I - AB_0\| = c < 1$, then

$$\lim_{k \to \infty} B_k = A^{-1}$$

and

$$\|A^{-1} - B_k\| \leq \frac{c^{2^k}}{1 - c} \|B_0\|.$$

13. Prove Theorem 5.12.

14. Prove Theorem 5.13.

Chapter 6
Krylov Subspace Methods

In this chapter, we will introduce a class of iterative methods called Krylov subspace methods. Among Krylov subspace methods developed for large sparse problems, we will mainly study two methods: the conjugate gradient (CG) method and the generalized minimum residual (GMRES) method. The CG method proposed by Hestenes and Stiefel in 1952 is one of the best known iterative methods for solving symmetric positive definite linear systems, see [16]. The GMRES method was proposed by Saad and Schultz in 1986 for solving nonsymmetric linear systems, see [34]. As usual, let us begin our discussion from the steepest descent method.

6.1 Steepest descent method

We consider the linear system

$$Ax = b$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix and $b \in \mathbb{R}^n$ is a known vector. We define the following quadratic function

$$\phi(x) \equiv x^T Ax - 2b^T x. \tag{6.1}$$

Theorem 6.1 *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Then finding the solution of $Ax = b$ is equivalent to finding the minimum of function (6.1).*

Proof Note that

$$\frac{\partial \phi}{\partial x_i} = 2(a_{i1}x_1 + \cdots + a_{in}x_n) - 2b_i, \qquad i = 1, 2, \cdots, n.$$

We therefore have

$$\text{grad}\phi(x) = 2(Ax - b) = -2r, \tag{6.2}$$

where $\text{grad}\phi(x)$ denotes the gradient of $\phi(x)$ and $r = b - Ax$. If $\phi(x)$ reaches its minimum at a point x_*, then

$$\text{grad}\phi(x_*) = 0,$$

i.e., $Ax_* = b$ which means that x_* is the solution of the system.

Conversely, if x_* is the solution of the system, then for any vector y, we have

$$\begin{aligned}\phi(x_* + y) &= (x_* + y)^T A(x_* + y) - 2b^T(x_* + y) \\ &= x_*^T Ax_* - 2b^T x_* + y^T Ay = \phi(x_*) + y^T Ay.\end{aligned}$$

Since A is symmetric positive definite, we have $y^T Ay \geq 0$. Hence,

$$\phi(x_* + y) \geq \phi(x_*),$$

i.e., $\phi(x)$ reaches its minimum at the point x_*. □

How to find the minimum of (6.1)? Usually, for any given initial vector x_0, we choose a direction p_0, and then we try to find a point

$$x_1 = x_0 + \alpha_0 p_0$$

on the line $x = x_0 + \alpha p_0$ such that

$$\phi(x_1) = \phi(x_0 + \alpha_0 p_0) \leq \phi(x_0 + \alpha p_0).$$

It means that along this line, $\phi(x)$ reaches its minimum at point x_1. Afterwards, starting from x_1, we choose another direction p_1, and then we try to find a point

$$x_2 = x_1 + \alpha_1 p_1$$

on the line $x = x_1 + \alpha p_1$ such that

$$\phi(x_2) = \phi(x_1 + \alpha_1 p_1) \leq \phi(x_1 + \alpha p_1),$$

i.e., along this line, $\phi(x)$ reaches its minimum at point x_2. Step by step, we have

$$p_0, p_1, \cdots, \quad \text{and} \quad \alpha_0, \alpha_1, \cdots,$$

where $\{p_k\}$ are line search directions and $\{\alpha_k\}$ are step sizes. In general, at a point x_k, we choose a direction p_k and then determine a step size α_k along the line $x = x_k + \alpha p_k$ such that

$$\phi(x_k + \alpha_k p_k) \leq \phi(x_k + \alpha p_k).$$

We then obtain $x_{k+1} = x_k + \alpha_k p_k$. We remark that different ways for choosing line search directions and step sizes give different algorithms for solving (6.1).

In the following, we first consider how to determine a step size α_k. Starting from a point x_k along a direction p_k, we want to find a step size α_k on the line $x = x_k + \alpha p_k$ such that

$$\phi(x_k + \alpha_k p_k) \leq \phi(x_k + \alpha p_k).$$

Let

$$\begin{aligned} f(\alpha) &= \phi(x_k + \alpha p_k) \\ &= (x_k + \alpha p_k)^T A(x_k + \alpha p_k) - 2b^T(x_k + \alpha p_k) \\ &= \alpha^2 p_k^T A p_k - 2\alpha r_k^T p_k + \phi(x_k) \end{aligned}$$

where $r_k = b - Ax_k$. We have

$$\frac{df}{d\alpha} = 2\alpha p_k^T A p_k - 2r_k^T p_k = 0.$$

Therefore,

$$\alpha_k = \frac{r_k^T p_k}{p_k^T A p_k}. \tag{6.3}$$

Once we get α_k, we can compute

$$x_{k+1} = x_k + \alpha_k p_k.$$

Is $\phi(x_{k+1}) \leq \phi(x_k)$? We consider

$$\begin{aligned} \phi(x_{k+1}) - \phi(x_k) &= \phi(x_k + \alpha_k p_k) - \phi(x_k) \\ &= \alpha_k^2 p_k^T A p_k - 2\alpha_k r_k^T p_k \\ &= -\frac{(r_k^T p_k)^2}{p_k^T A p_k} \leq 0. \end{aligned}$$

If $r_k^T p_k \neq 0$, then $\phi(x_{k+1}) < \phi(x_k)$.

Now we consider how to choose a direction p_k. It is well-known that the steepest descent direction of $\phi(x)$ is the negative direction of the gradient, i.e., $p_k = r_k$ by (6.2). Thus we have the steepest descent method. In order to discuss the convergence rate of the method, we introduce the following lemma first.

Lemma 6.1 *Let $0 < \lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of a symmetric positive definite matrix A and $P(t)$ be a real polynomial of t. Then*

$$\|P(A)x\|_A \leq \max_{1 \leq j \leq n} |P(\lambda_j)| \cdot \|x\|_A, \qquad x \in \mathbb{R}^n,$$

where $\|x\|_A \equiv \sqrt{x^T A x}$.

Proof Let $y_1, y_2, \cdots, y_n$ be the eigenvectors of A corresponding to the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, respectively. Suppose that $y_1, y_2, \cdots, y_n$ also form an orthonormal basis of $\mathbb{R}^n$. Therefore, for any $x \in \mathbb{R}^n$, we have $x = \sum_{i=1}^{n} \beta_i y_i$ and furthermore,

$$\begin{aligned} x^T P(A)AP(A)x &= \left(\sum_{i=1}^{n} \beta_i P(\lambda_i) y_i\right)^T A \left(\sum_{i=1}^{n} \beta_i P(\lambda_i) y_i\right) \\ &= \sum_{i=1}^{n} \lambda_i \beta_i^2 P^2(\lambda_i) \leq \max_{1 \leq j \leq n} P^2(\lambda_j) \sum_{i=1}^{n} \lambda_i \beta_i^2 \\ &= \max_{1 \leq j \leq n} P^2(\lambda_j) x^T A x. \end{aligned}$$

Then

$$\|P(A)x\|_A \leq \max_{1 \leq j \leq n} |P(\lambda_j)| \cdot \|x\|_A. \qquad \square$$

For the steepest descent method, we have the following convergence theorem.

Theorem 6.2 *Let $0 < \lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of a symmetric positive definite matrix A. Then the sequence $\{x_k\}$ produced by the steepest descent method satisfies*

$$\|x_k - x_*\|_A \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^k \|x_0 - x_*\|_A,$$

where x_* is the exact solution of $Ax = b$.

Proof The x_k produced by the steepest descent method satisfies

$$\phi(x_k) \leq \phi(x_{k-1} + \alpha r_{k-1}), \qquad \alpha \in \mathbb{R}.$$

By noting that

$$\phi(x) + x_*^T A x_* = (x - x_*)^T A (x - x_*),$$

we have

$$\begin{aligned} &(x_k - x_*)^T A (x_k - x_*) \leq (x_{k-1} + \alpha r_{k-1} - x_*)^T A (x_{k-1} + \alpha r_{k-1} - x_*) \\ &= [(I - \alpha A)(x_{k-1} - x_*)]^T A [(I - \alpha A)(x_{k-1} - x_*)], \end{aligned} \tag{6.4}$$

for any $\alpha \in \mathbb{R}$. Let $P_\alpha(t) = 1 - \alpha t$. By using Lemma 6.1, we have from (6.4),

$$\begin{aligned} \|x_k - x_*\|_A &\leq \|P_\alpha(A)(x_{k-1} - x_*)\|_A \\ &\leq \max_{1\leq j\leq n} |P_\alpha(\lambda_j)| \cdot \|x_{k-1} - x_*\|_A, \end{aligned} \tag{6.5}$$

for any $\alpha \in \mathbb{R}$. By using properties of the Chebyshev approximation, see [33], we have

$$\min_\alpha \max_{\lambda_1\leq t\leq\lambda_n} |1 - \alpha t| = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}. \tag{6.6}$$

Substituting (6.6) into (6.5), we obtain

$$\|x_k - x_*\|_A \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \|x_{k-1} - x_*\|_A.$$

Thus,

$$\|x_k - x_*\|_A \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^k \|x_0 - x_*\|_A. \qquad \square$$

6.2 Conjugate gradient method

In this section, we introduce the conjugate gradient (CG) method which is one of the most important Krylov subspace methods.

6.2.1 Conjugate gradient method

The basic idea of the CG method is given as follows. For a given initial vector x_0, in the first step, we still choose the direction of negative gradient, i.e., $p_0 = r_0$. Then we have

$$\alpha_0 = \frac{r_0^T p_0}{p_0^T A p_0}, \qquad x_1 = x_0 + \alpha_0 p_0, \qquad r_1 = b - A x_1.$$

Afterwards, at the $(k+1)$-th step ($k \geq 1$), we want to choose a direction p_k on the plane

$$\pi_2 = \{x = x_k + \xi r_k + \eta p_{k-1} : \xi, \eta \in \mathbb{R}\}$$

such that ϕ decreases most rapidly. Consider ϕ on π_2:

$$\begin{aligned} \psi(\xi, \eta) &= \phi(x_k + \xi r_k + \eta p_{k-1}) \\ &= (x_k + \xi r_k + \eta p_{k-1})^T A (x_k + \xi r_k + \eta p_{k-1}) \\ &\quad -2b^T (x_k + \xi r_k + \eta p_{k-1}). \end{aligned}$$

By directly computing, we have

$$\frac{\partial \psi}{\partial \xi} = 2(\xi r_k^T A r_k + \eta r_k^T A p_{k-1} - r_k^T r_k),$$

$$\frac{\partial \psi}{\partial \eta} = 2(\xi r_k^T A p_{k-1} + \eta p_{k-1}^T A p_{k-1}),$$

where we use $r_k^T p_{k-1} = 0$ (see Theorem 6.3 later). Let

$$\frac{\partial \psi}{\partial \xi} = \frac{\partial \psi}{\partial \eta} = 0,$$

then we find a unique minimum point

$$\tilde{x} = x_k + \xi_0 r_k + \eta_0 p_{k-1}$$

of ϕ on the plane π_2, where ξ_0 and η_0 satisfy:

$$\begin{aligned} \xi_0 r_k^T A r_k + \eta_0 r_k^T A p_{k-1} &= r_k^T r_k \\ \xi_0 r_k^T A p_{k-1} + \eta_0 p_{k-1}^T A p_{k-1} &= 0. \end{aligned} \tag{6.7}$$

Note that from (6.7), if $r_k \neq 0$ then $\xi_0 \neq 0$. We therefore can choose

$$p_k = \frac{1}{\xi_0}(\tilde{x} - x_k) = r_k + \frac{\eta_0}{\xi_0} p_{k-1}$$

as a new direction which is the optimal direction for minimizing ϕ on the plane π_2. Let $\beta_{k-1} = \frac{\eta_0}{\xi_0}$. Then by using the second equation in (6.7), we have

$$\beta_{k-1} = -\frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

Note that p_k satisfies $p_k^T A p_{k-1} = 0$ (see Theorem 6.3 later), i.e., p_k and p_{k-1} are mutually A-conjugate.

Once we get p_k, we can determine α_k by using (6.3) and then compute

$$x_{k+1} = x_k + \alpha_k p_k.$$

In conclusion, we have the following formulas:

$$\alpha_k = \frac{r_k^T p_k}{p_k^T A p_k},$$

$$x_{k+1} = x_k + \alpha_k p_k,$$

$$r_{k+1} = b - A x_{k+1},$$

$$\beta_k = -\frac{r_{k+1}^T A p_k}{p_k^T A p_k},$$

$$p_{k+1} = r_{k+1} + \beta_k p_k.$$

After a few elementary computations, we obtain

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}, \qquad \beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

Thus, the scheme of the CG method, one of the most popular and successful iterative methods for solving symmetric positive definite systems $Ax = b$, is given as follows. At the initialization step, for $k = 0$, we choose x_0 and then calculate

$$r_0 = b - A x_0.$$

While $r_k \neq 0$, in iteration steps, we have

$$\begin{cases} k = k + 1 \\ \textbf{if } k = 1 \\ \qquad p_0 = r_0 \\ \textbf{else} \\ \qquad \beta_{k-2} = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2} \\ \qquad p_{k-1} = r_{k-1} + \beta_{k-2} p_{k-2} \\ \textbf{end} \\ \alpha_{k-1} = r_{k-1}^T r_{k-1} / p_{k-1}^T A p_{k-1} \\ x_k = x_{k-1} + \alpha_{k-1} p_{k-1} \\ r_k = r_{k-1} - \alpha_{k-1} A p_{k-1} \end{cases}$$

where r_k, p_k are vectors and α_k, β_k are scalars, $k = 0, 1, \cdots$. The x_k is the approximation to the exact solution after the k-th iteration. When $r_k = 0$, then the solution is $x = x_k$.

6.2.2 Basic properties

Theorem 6.3 *The vectors $\{r_i\}$ and $\{p_i\}$ satisfy the following properties:*

(1) $p_i^T r_j = 0, \quad 0 \le i < j \le k;$

(2) $r_i^T r_j = 0, \quad i \ne j, \quad 0 \le i, j \le k;$

(3) $p_i^T A p_j = 0, \quad i \ne j, \quad 0 \le i, j \le k;$

(4) $span\{r_0, \cdots, r_k\} = span\{p_0, \cdots, p_k\} = \mathcal{K}(A, r_0, k+1)$, *where*

$$\mathcal{K}(A, r_0, k+1) \equiv span\{r_0, Ar_0, \cdots, A^k r_0\}$$

is called the Krylov subspace.

Proof By using induction, for $k = 1$, we have

$$p_0 = r_0, \qquad r_1 = r_0 - \alpha_0 A p_0, \qquad p_1 = r_1 + \beta_0 p_0.$$

Then

$$p_0^T r_1 = r_0^T r_1 = r_0^T (r_0 - \alpha_0 A p_0) = r_0^T r_0 - \alpha_0 p_0^T A p_0 = 0$$

provided $\alpha_0 = r_0^T r_0 / p_0^T A p_0$, and

$$p_1^T A p_0 = (r_1 + \beta_0 r_0)^T A r_0 = r_1^T A r_0 - \frac{r_1^T A r_0}{r_0^T A r_0} r_0^T A r_0 = 0.$$

Now, we assume that the theorem is true for k and we try to prove that it also holds for $k+1$.

For (1), by using assumption and $r_{k+1} = r_k - \alpha_k A p_k$, we have

$$p_i^T r_{k+1} = p_i^T r_k - \alpha_k p_i^T A p_k = 0, \qquad 0 \le i \le k-1,$$

and also

$$p_k^T r_{k+1} = p_k^T r_k - \frac{p_k^T r_k}{p_k^T A p_k} p_k^T A p_k = 0.$$

Thus, (1) is true for $k+1$.

For (2), we have by assumption,

$$\text{span}\{r_0, \cdots, r_k\} = \text{span}\{p_0, \cdots, p_k\}.$$

By (1), we know that r_{k+1} is orthogonal to this subspace. Therefore, (2) is true for $k+1$.

For (3), by using assumption, (2) and

$$p_{k+1} = r_{k+1} + \beta_k p_k, \qquad r_{i+1} = r_i - \alpha_i A p_i,$$

we have

$$p_{k+1}^T A p_i = \frac{1}{\alpha_i} r_{k+1}^T (r_i - r_{i+1}) + \beta_k p_k^T A p_i = 0, \qquad i = 0, 1, \cdots, k-1.$$

By the definition of β_k, we have

$$p_{k+1}^T A p_k = (r_{k+1} + \beta_k p_k)^T A p_k = r_{k+1}^T A p_k - \frac{r_{k+1}^T A p_k}{p_k^T A p_k} p_k^T A p_k = 0.$$

Then, (3) holds for $k+1$.

For (4), we know by using assumption that

$$r_k,\ p_k \in \mathcal{K}(A, r_0, k+1) = \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}.$$

Therefore

$$r_{k+1} = r_k - \alpha_k A p_k \in \mathcal{K}(A, r_0, k+2) = \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\},$$

and

$$p_{k+1} = r_{k+1} + \beta_k p_k \in \mathcal{K}(A, r_0, k+2) = \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}.$$

By (2) and (3), we note that the vectors $r_0, \cdots, r_{k+1}$ and $p_0, \cdots, p_{k+1}$ are linearly independent. Thus, (4) is true for $k+1$. □

We remark that by Theorem 6.3, at most n steps, we can obtain the exact solution of an n-by-n system by using the CG method. Therefore, from a theoretical viewpoint, the CG method is a direct method.

Theorem 6.4 *The x_k obtained from the CG method satisfies*

$$\phi(x_k) = \min\{\phi(x) \ : \ x \in x_0 + \mathcal{K}(A, r_0, k)\} \tag{6.8}$$

or

$$\|x_k - x_*\|_A = \min\{\|x - x_*\|_A \ : \ x \in x_0 + \mathcal{K}(A, r_0, k)\}, \tag{6.9}$$

where $\|x\|_A = \sqrt{x^T A x}$ and x_* is the exact solution of $Ax = b$.

Proof Since (6.8) and (6.9) are equivalent, we only need to prove (6.9). Suppose that $r_l = 0$ at the l-th step of the CG method, then we have

$$\begin{aligned} x_* &= x_l = x_{l-1} + \alpha_{l-1}p_{l-1} \\ &= x_{l-2} + \alpha_{l-2}p_{l-2} + \alpha_{l-1}p_{l-1} \\ &\cdots\cdots\cdots \\ &= x_0 + \alpha_0 p_0 + \cdots + \alpha_{l-1}p_{l-1}. \end{aligned}$$

For $k < l$, we have

$$x_k = x_0 + \alpha_0 p_0 + \cdots + \alpha_{k-1}p_{k-1} \in x_0 + \mathcal{K}(A, r_0, k).$$

Let x be any vector in $x_0 + \mathcal{K}(A, r_0, k)$. Then by Theorem 6.3 (4), we have

$$x = x_0 + \gamma_0 p_0 + \cdots + \gamma_{k-1}p_{k-1}.$$

Moreover,

$$x_* - x = (\alpha_0 - \gamma_0)p_0 + \cdots + (\alpha_{k-1} - \gamma_{k-1})p_{k-1} + \alpha_k p_k + \cdots + \alpha_{l-1}p_{l-1}.$$

Since

$$x_* - x_k = \alpha_k p_k + \cdots + \alpha_{l-1}p_{l-1},$$

we have by using Theorem 6.3 (3),

$$\begin{aligned} \|x_* - x\|_A^2 &= \|(\alpha_0 - \gamma_0)p_0 + \cdots + (\alpha_{k-1} - \gamma_{k-1})p_{k-1}\|_A^2 \\ &\quad + \|\alpha_k p_k + \cdots + \alpha_{l-1}p_{l-1}\|_A^2 \\ &\geq \|\alpha_k p_k + \cdots + \alpha_{l-1}p_{l-1}\|_A^2 \\ &= \|x_* - x_k\|_A^2. \qquad \square \end{aligned}$$

6.3 Practical CG method and convergence analysis

In this section, we give a practical algorithm of the CG method and analyze the convergence rate of the CG method.

6.3.1 Practical CG method

From Theorem 6.3, we know that the CG method would obtain an accurate solution after n steps in exact arithmetic, where n is the size of the system. In other words, the CG method is thought as a direct method rather than an iterative method. When n is very large, in practice, we use the CG method as an iterative method and stop iterations when

(i) $\|r_k\|$ is less than ϵ, where $r_k = b - Ax_k$ and ϵ is a given error bound; or

(ii) the number of iterations reaches $k_{\max}$, the largest number of iterations provided by us, where $k_{\max} \ll n$.

We then have the following practical algorithm for solving symmetric positive definite systems $Ax = b$. At the initialization step $k = 0$, we choose a initial vector x and calculate

$$r = b - Ax, \qquad \rho = r^T r.$$

While $\sqrt{\rho} > \epsilon \|b\|_2$ and $k < k_{\max}$, in iteration steps, we have

$$\begin{cases} k = k + 1 \\ \textbf{if } k = 1 \\ \quad p = r \\ \textbf{else} \\ \quad \beta = \rho/\tilde{\rho}; \qquad p = r + \beta p \\ \textbf{end} \\ w = Ap; \qquad \alpha = \rho/p^T w; \qquad x = x + \alpha p \\ r = r - \alpha w; \qquad \tilde{\rho} = \rho; \qquad \rho = r^T r \end{cases}$$

where r, p, w are vectors and ρ, α, β are scalars.

We remark that:

(1) We only have the matrix-vector multiplications in the algorithm. If the matrix is sparse or it has a special structure, then these multiplications can be done efficiently by using some sparse solvers or fast solvers.

(2) We do not need to estimate any parameter in the algorithm unlike the SOR method.

(3) For each iteration, we could use the parallel algorithms for the vector operations.

Now, we briefly discuss how to use the CG method to solve general linear systems $Ax = b$. Since we cannot use the CG method directly to the system,

instead of solving $Ax = b$, we can use the CG method to solve the normalized system,

$$A^T Ax = A^T b.$$

When the system is well-conditioned, then the normalized CG method is suitable. But if the system is ill-conditioned, then the condition number of the normalized system could become very large because of $\kappa_2(A^T A) = (\kappa_2(A))^2$. Hence the normalized CG method is not suitable for ill-conditioned systems.

6.3.2 Convergence analysis

We have the following theorem for the convergence rate of the CG method.

Theorem 6.5 *If $A = I + B$ where I is the identity matrix and rank$(B) = p$, then by using at most $p + 1$ iterations, the CG method can obtain the exact solution of $Ax = b$.*

Proof Since $A = I + B$, it is easy to show that

$$\text{span}\{r_0, Ar_0, \cdots, A^k r_0\} = \text{span}\{r_0, Br_0, \cdots, B^k r_0\}.$$

Note that $\text{rank}(B) = p$ and then

$$\dim\left(\text{span}\{r_0, Br_0, \cdots, B^k r_0\}\right) \leq p + 1.$$

By Theorem 6.3, we know that

$$\dim\left(\text{span}\{r_0, \cdots, r_p, r_{p+1}\}\right) \leq p + 1$$

and also

$$r_i^T r_{p+1} = 0, \qquad i = 0, 1, \cdots, p.$$

Therefore $r_{p+1} = 0$, i.e., $Ax_{p+1} = b$. □

Theorem 6.6 *Let $A \in \mathbb{R}^{n \times n}$ be symmetric positive definite and x_* be the exact solution of $Ax = b$. We then have*

$$\|x_* - x_k\|_A \leq 2\left(\frac{\sqrt{\kappa_2} - 1}{\sqrt{\kappa_2} + 1}\right)^k \|x_* - x_0\|_A$$

where x_k is produced by the CG method and

$$\kappa_2 = \kappa_2(A) = \|A\|_2 \|A^{-1}\|_2.$$

Proof By Theorem 6.3, we know that for any $x \in x_0 + \mathcal{K}(A, r_0, k)$,

$$\begin{aligned} x_* - x &= x_* - x_0 + a_{k1} r_0 + a_{k2} A r_0 + \cdots + a_{kk} A^{k-1} r_0 \\ &= A^{-1}(r_0 + a_{k1} A r_0 + a_{k2} A^2 r_0 + \cdots + a_{kk} A^k r_0) \\ &= A^{-1} P_k(A) r_0, \end{aligned}$$

where $P_k(\lambda) = \sum_{j=0}^{k} a_{kj} \lambda^j$ with $P_k(0) = 1$. Let $\mathcal{P}_k$ be the set of all the polynomials P_k with order less than or equal to k and $P_k(0) = 1$. By Theorem 6.4 and Lemma 6.1, we have

$$\begin{aligned} \|x_k - x_*\|_A &= \min\{\|x - x_*\|_A \ : \ x \in x_0 + \mathcal{K}(A, r_0, k)\} \\ &= \min_{P_k \in \mathcal{P}_k} \|A^{-1} P_k(A) r_0\|_A = \min_{P_k \in \mathcal{P}_k} \|P_k(A) A^{-1} r_0\|_A \\ &\leq \min_{P_k \in \mathcal{P}_k} \max_{1 \leq i \leq n} |P_k(\lambda_i)| \cdot \|A^{-1} r_0\|_A \\ &\leq \min_{P_k \in \mathcal{P}_k} \max_{a_1 \leq \lambda \leq a_2} |P_k(\lambda)| \cdot \|x_* - x_0\|_A, \end{aligned}$$

where

$$0 < a_1 = \lambda_1 \leq \cdots \leq \lambda_n = a_2$$

are the eigenvalues of A. By the well-known Approximation Theorem of Chebyshev polynomials, see [33], we know that there exists a unique solution of the optimal problem

$$\min_{P_k \in \mathcal{P}_k} \max_{a_1 \leq \lambda \leq a_2} |P_k(\lambda)|$$

given by

$$\widetilde{P}_k(\lambda) = \frac{T_k(\frac{a_2 + a_1 - 2\lambda}{a_2 - a_1})}{T_k(\frac{a_2 + a_1}{a_2 - a_1})}.$$

Here $T_k(z)$ is the k-th Chebyshev polynomial defined recursively by

$$T_k(z) = 2z T_{k-1}(z) - T_{k-2}(z)$$

with $T_0(z) = 1$ and $T_1(z) = z$. By the properties of Chebyshev polynomials, see [33] again, we know that

$$\max_{a_1 \leq \lambda \leq a_2} |\widetilde{P}_k(\lambda)| = \frac{1}{T_k(\frac{a_2 + a_1}{a_2 - a_1})} \leq 2 \left(\frac{\sqrt{\kappa_2} - 1}{\sqrt{\kappa_2} + 1} \right)^k,$$

where $\kappa_2 = \kappa_2(A) = \|A\|_2\|A^{-1}\|_2$. Therefore,

$$\|x_* - x_k\|_A \leq 2\left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)^k \|x_* - x_0\|_A. \qquad \square$$

Furthermore, we have the following theorem, see [2].

Theorem 6.7 *If the eigenvalues λ_j of a symmetric positive definite matrix A are ordered such that*

$$0 < \lambda_1 \leq \ldots \leq \lambda_p \leq b_1 \leq \lambda_{p+1} \leq \ldots \leq \lambda_{n-q} \leq b_2 \leq \lambda_{n-q+1} \leq \ldots \leq \lambda_n$$

where b_1 and b_2 are two constants, then

$$\frac{\|x_* - x_k\|_A}{\|x_* - x_0\|_A} \leq 2\left(\frac{\alpha-1}{\alpha+1}\right)^{k-p-q} \cdot \max_{\lambda\in[b_1,b_2]} \prod_{j=1}^{p}\left(\frac{\lambda-\lambda_j}{\lambda_j}\right),$$

where $\alpha \equiv (b_2/b_1)^{1/2} \geq 1$.

From Theorem 6.7, we note that when n is increased, if p, q are constants that do not depend on n and λ_1 is uniformly bounded from zero, then the convergence rate is linear, i.e., the number of iterations is independent of n. We also notice that the more clustered the eigenvalues are, the faster the convergence rate will be.

Corollary 6.1 *If the eigenvalues λ_j of a symmetric positive definite matrix A are ordered such that*

$$0 < \delta < \lambda_1 \leq \ldots \leq \lambda_p \leq 1-\epsilon \leq \lambda_{p+1} \leq \ldots \leq \lambda_{n-q} \leq 1+\epsilon \leq \lambda_{n-q+1} \leq \ldots \leq \lambda_n$$

where $0 < \epsilon < 1$, then

$$\frac{\|x_* - x_k\|_A}{\|x_* - x_0\|_A} \leq 2\left(\frac{1+\epsilon}{\delta}\right)^p \epsilon^{k-p-q}$$

where $k \geq p+q$.

Proof For α given in Theorem 6.7, we have

$$\alpha \equiv \left(\frac{b_2}{b_1}\right)^{\frac{1}{2}} = \left(\frac{1+\epsilon}{1-\epsilon}\right)^{\frac{1}{2}}.$$

Therefore,

$$\frac{\alpha-1}{\alpha+1}=\frac{1-\sqrt{1-\epsilon^2}}{\epsilon}<\epsilon.$$

For $1\le j\le p$ and $\lambda\in[1-\epsilon,1+\epsilon]$, we have

$$0\le\frac{\lambda-\lambda_j}{\lambda_j}\le\frac{1+\epsilon}{\delta}.$$

Thus, by using Theorem 6.7, we obtain

$$\begin{aligned}\frac{\|x_*-x_k\|_A}{\|x_*-x_0\|_A} &\le 2\left(\frac{\alpha-1}{\alpha+1}\right)^{k-p-q}\cdot\max_{\lambda\in[b_1,b_2]}\prod_{j=1}^{p}\left(\frac{\lambda-\lambda_j}{\lambda_j}\right)\\ &\le 2\left(\frac{1+\epsilon}{\delta}\right)^{p}\epsilon^{k-p-q}.\quad\square\end{aligned}$$

6.4 Preconditioning

From Section 6.3, we know that if the matrix A of the system

$$Ax=b$$

is well-conditioned or its spectrum is clustered, then the convergence rate of the CG method will be very quick. Therefore, in order to speed up the convergence rate, we usually precondition the system, i.e., instead of solving the original system, we solve the following preconditioned system

$$\widetilde{A}\tilde{x}=\tilde{b},\tag{6.10}$$

where

$$\widetilde{A}=C^{-1}AC^{-1},\qquad \tilde{x}=Cx,\qquad \tilde{b}=C^{-1}b,$$

and C is symmetric positive definite. We wish that the preconditioned matrix $\widetilde{A}$ could have better spectral properties than those of A.

By using the CG method on (6.10), we have

$$\begin{cases}\alpha_k=\dfrac{\tilde{r}_k^T\tilde{r}_k}{\tilde{p}_k^T\widetilde{A}\tilde{p}_k},\\[2ex] \tilde{x}_{k+1}=\tilde{x}_k+\alpha_k\tilde{p}_k,\\[2ex] \tilde{r}_{k+1}=\tilde{r}_k-\alpha_k\widetilde{A}\tilde{p}_k,\\[2ex] \beta_k=\dfrac{\tilde{r}_{k+1}^T\tilde{r}_{k+1}}{\tilde{r}_k^T\tilde{r}_k},\\[2ex] \tilde{p}_{k+1}=\tilde{r}_{k+1}+\beta_k\tilde{p}_k,\end{cases}\tag{6.11}$$

where $\tilde{x}_0$ is any given initial vector, $\tilde{r}_0 = \tilde{b} - \tilde{A}\tilde{x}_0$ and $\tilde{p}_0 = \tilde{r}_0$. Let

$$\tilde{x}_k = Cx_k, \qquad \tilde{r}_k = C^{-1}r_k, \qquad \tilde{p}_k = Cp_k, \qquad M = C^2.$$

Substituting them into (6.11), we actually have

$$\begin{aligned} w_k &= Ap_k, & \alpha_k &= \rho_k / p_k^T w_k, \\ x_{k+1} &= x_k + \alpha_k p_k, & r_{k+1} &= r_k - \alpha_k w_k, \\ z_{k+1} &= M^{-1} r_{k+1}, & \rho_{k+1} &= r_{k+1}^T z_{k+1}, \\ \beta_k &= \rho_{k+1}/\rho_k, & p_{k+1} &= z_{k+1} + \beta_k p_k, \end{aligned}$$

where x_0 is any given initial vector, $r_0 = b - Ax_0$, $z_0 = M^{-1}r_0$, $\rho_0 = r_0^T z_0$ and $p_0 = z_0$.

We then have the following preconditioned algorithm. At the initialization step $k = 0$, we choose a initial vector x and calculate $r = b - Ax$. While $\sqrt{r^T r} > \epsilon \|b\|_2$ and $k < k_{\max}$, in iteration steps, we have

$$\left\{ \begin{array}{l} \text{Solve } Mz = r \text{ for } z \\ k = k + 1 \\ \textbf{if } k = 1 \\ \qquad p = z \\ \textbf{else} \\ \qquad \beta = \rho/\tilde{\rho}; \qquad p = z + \beta p \\ \textbf{end} \\ w = Ap; \qquad \alpha = \rho/p^T w; \qquad x = x + \alpha p \\ r = r - \alpha w; \qquad \tilde{\rho} = \rho; \qquad \rho = r^T z \end{array} \right.$$

where z, r, p, w are vectors and ρ, α, β are scalars. This algorithm is called the preconditioned conjugate gradient (PCG) method. Note that the PCG method has the following properties:

(i) $r_i^T M^{-1} r_j = 0$, for $i \neq j$.

(ii) $p_i^T A p_j = 0$, for $i \neq j$.

(iii) The approximation x_k satisfies:

$$\|x_* - x_k\|_{\tilde{A}} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|x_* - x_0\|_{\tilde{A}}$$

where $\kappa = \lambda_n/\lambda_1$ with λ_n being the largest eigenvalue of $M^{-1}A$ and λ_1 being the smallest eigenvalue of $M^{-1}A$.

A good preconditioner $M = C^2$ is chosen with two criteria in mind, see [2, 19, 22]:

(1) $Mz = d$ is easy to solve.

(2) The spectrum of $M^{-1}A$ is clustered and (or) $M^{-1}A$ is well-conditioned compared to A.

Usually, it is not easy to choose a preconditioner which satisfies all these two criteria. Now we briefly discuss the following three classes of preconditioners.

1. **Diagonal preconditioners.** If diagonal entries of the coefficient matrix A are much different, one could use the matrix

$$M = \text{diag}(a_{11}, \cdots, a_{nn})$$

as a preconditioner to speed up the convergence rate of the CG method hopefully. For block matrix

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1k} \\ \vdots & \ddots & \vdots \\ A_{k1} & \cdots & A_{kk} \end{bmatrix},$$

if A_{ii}^{-1} are easily to be obtained, then one could use the block diagonal matrix

$$M = \text{diag}(A_{11}, \cdots, A_{kk})$$

as a preconditioner.

2. **Preconditioners based on incomplete Cholesky factorization.** If one computes the incomplete Cholesky factorization first,

$$A = LL^T + R,$$

then one can use the matrix $M = LL^T$ as a preconditioner. We could require that the matrix L has the same sparse structure as the matrix A and also the matrix $LL^T \approx A$.

3. **Optimal (circulant) preconditioners.** This class of preconditioners is proposed very recently, see [9, 22, 37]. The circulant matrix is defined as follows:

$$C_n = \begin{bmatrix} c_0 & c_{-1} & \cdots & c_{2-n} & c_{1-n} \\ c_1 & c_0 & c_{-1} & \cdots & c_{2-n} \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \cdots & \ddots & \ddots & c_{-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}$$

where $c_{-k} = c_{n-k}$ for $1 \leq k \leq n-1$. It is well-known that circulant matrices can be diagonalized by the Fourier matrix F_n, see [13], i.e.,

$$C_n = F_n^* \Lambda_n F_n, \tag{6.12}$$

where the entries of F_n are given by

$$(F_n)_{j,k} = \frac{1}{\sqrt{n}} e^{2\pi \mathbf{i}(j-1)(k-1)/n}, \qquad \mathbf{i} \equiv \sqrt{-1},$$

for $1 \leq j, k \leq n$, and Λ_n is a diagonal matrix holding the eigenvalues of C_n. We note that Λ_n can be obtained in $O(n \log n)$ operations by taking the fast Fourier transform (FFT) of the first column of C_n. Once Λ_n is obtained, the products $C_n y$ and $C_n^{-1} y$ for any vector y can be computed by FFTs in $O(n \log n)$ operations. For the Fourier matrix F_n, when there is no ambiguity, we shall denote F.

Now, we study a kind of preconditioner called the optimal preconditioner, see [9, 11, 22]. Given any unitary matrix $U \in \mathbb{C}^{n \times n}$, let $\mathcal{M}_U$ be the set of all matrices simultaneously diagonalized by U, i.e.,

$$\mathcal{M}_U = \{U^* \Lambda U \mid \Lambda \text{ is an } n\text{-by-}n \text{ diagonal matrix}\}. \tag{6.13}$$

We note that in (6.13), when $U = F$, the Fourier matrix, $\mathcal{M}_F$ is the set of all the circulant matrices. Let $\delta(A)$ denote the diagonal matrix whose diagonal is equal to the diagonal of the matrix A. We have the following lemma, see [22, 27, 39].

Lemma 6.2 *For any arbitrary $A = [a_{pq}] \in \mathbb{C}^{n \times n}$, let $c_U(A)$ be the minimizer of $\|W - A\|_F$ over all $W \in \mathcal{M}_U$. Then*

(i) *$c_U(A)$ is uniquely determined by A and is given by*

$$c_U(A) = U^* \delta(UAU^*) U.$$

(ii) *If A is Hermitian, then $c_U(A)$ is also Hermitian. Furthermore, if $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalues respectively, then we have*

$$\lambda_{\min}(A) \leq \lambda_{\min}(c_U(A)) \leq \lambda_{\max}(c_U(A)) \leq \lambda_{\max}(A).$$

In particular, if A is positive definite, then so is $c_U(A)$.

(iii) *If A is normal and stable, i.e., $A^*A = AA^*$ and the real parts of all the eigenvalues of A are negative, then $c_U(A)$ is also stable.*

(iv) *When U is the Fourier matrix F, we then have*

$$c_F(A) = \sum_{j=0}^{n-1} \left(\frac{1}{n} \sum_{p-q \equiv j \,(\mathrm{mod}\, n)} a_{pq} \right) Q^j,$$

where Q is an n-by-n circulant matrix given by

$$Q \equiv \begin{bmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ 0 & 1 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $c_U(A)$ is called the optimal preconditioner of A and the matrix $c_F(A)$ is called the optimal circulant preconditioner of A. We remark that $c_F(A)$ is a good preconditioner for solving a large class of structured linear systems $Ax = b$, for instance, Toeplitz systems, Hankel systems, etc, see [9, 11, 12, 22].

6.5 GMRES method

In this section, we introduce the generalized minimum residual (GMRES) method to solve general systems

$$Ax = b$$

where $A \in \mathbb{R}^{n \times n}$ is nonsingular. The GMRES method was proposed by Saad and Schultz in 1986, which is one of the most important Krylov subspace methods for nonsymmetric systems, see [33, 34].

6.5.1 Basic properties of GMRES method

For the GMRES method, in the k-th iteration, we are going to find a solution x_k of the LS problem

$$\min_{x \in x_0 + \mathcal{K}(A, r_0, k)} \|b - Ax\|_2$$

where

$$\mathcal{K}(A, r_0, k) \equiv \mathrm{span}\{r_0, Ar_0, \cdots, A^{k-1} r_0\}$$

with $r_0 = b - Ax_0$. Let $x \in x_0 + \mathcal{K}(A, r_0, k)$. We have

$$x = x_0 + \sum_{j=0}^{k-1} \gamma_j A^j r_0$$

and then

$$r = b - Ax = b - Ax_0 - \sum_{j=0}^{k-1} \gamma_j A^{j+1} r_0 = r_0 - \sum_{j=1}^{k} \gamma_{j-1} A^j r_0.$$

Hence

$$r = \bar{P}_k(A) r_0$$

where $\bar{P}_k \in \mathcal{P}_k$ with $\mathcal{P}_k$ being the set of all the polynomials P_k with order less than or equal to k and $P_k(0) = 1$. We therefore have the following theorem.

Theorem 6.8 *Let x_k be the solution after the k-th GMRES iteration. Then we have*

$$\|r_k\|_2 = \min_{P \in \mathcal{P}_k} \|P(A) r_0\|_2 \leq \|\bar{P}_k(A) r_0\|_2.$$

Furthermore,

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \|\bar{P}_k(A)\|_2.$$

Moreover, we have

Theorem 6.9 *The GMRES method will obtain the exact solution of $Ax = b$ within n iterations, where $A \in \mathbb{R}^{n \times n}$.*

Proof The characteristic polynomial of A is given by

$$p_A(z) = \det(zI - A)$$

where the order of $p_A(z)$ is n and $p_A(0) = (-1)^n \det(A) \neq 0$. Then

$$\bar{P}_n(z) = \frac{p_A(z)}{p_A(0)} \in \mathcal{P}_n.$$

By the Hamilton-Cayley Theorem, see [17], we know that

$$\bar{P}_n(A) = p_A(A) = 0.$$

By Theorem 6.8, we have

$$r_n = b - Ax_n = 0.$$

Thus x_n is the exact solution of $Ax = b$. □

If A is diagonalizable, i.e., $A = V \Lambda V^{-1}$ where Λ is a diagonal matrix, then we have

$$P(A) = V P(\Lambda) V^{-1}.$$

If V is orthogonal, then A is a normal matrix.

Theorem 6.10 *Let $A = V\Lambda V^{-1}$. We have*

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa_2(V) \min_{P\in\mathcal{P}_k} \max_{\lambda_i} |P(\lambda_i)|$$

where λ_i are the eigenvalues of the matrix A.

Proof By Theorem 6.8, we have

$$\begin{aligned}
\|r_k\|_2 &= \min_{P\in\mathcal{P}_k} \|P(A)r_0\|_2 = \min_{P\in\mathcal{P}_k} \|VP(\Lambda)V^{-1}r_0\|_2 \\
&\leq \min_{P\in\mathcal{P}_k} \|V\|_2\|V^{-1}\|_2\|P(\Lambda)\|_2\|r_0\|_2 \\
&= \kappa_2(V) \min_{P\in\mathcal{P}_k} \|P(\Lambda)\|_2\|r_0\|_2 \\
&= \kappa_2(V) \min_{P\in\mathcal{P}_k} \max_{\lambda_i} |P(\lambda_i)| \cdot \|r_0\|_2. \qquad \square
\end{aligned}$$

We remark that when A is normal, then $\kappa_2(V) = 1$. We therefore have

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \min_{P\in\mathcal{P}_k} \max_{\lambda_i} |P(\lambda_i)|.$$

Theorem 6.11 *If A is diagonalizable and has exactly k distinct eigenvalues, then the GMRES method will terminate in at most k iterations.*

Proof We construct a polynomial as follows,

$$P(z) = \prod_{i=1}^{k} \frac{\lambda_i - z}{\lambda_i} \in \mathcal{P}_k,$$

where λ_i are the eigenvalues of A. By Theorem 6.10, we know that $r_k = 0$, i.e., $Ax_k = b$. $\square$

We should emphasize that in general, the behavior of the GMRES method cannot be determined by eigenvalues alone. In fact, it is shown in [20] that any nonincreasing convergence rate is possible for the GMRES method applied to some problem with nonnormal matrix. Moreover, that problem can have any desired distribution of eigenvalues. Thus, for instance, eigenvalues tightly clustered around 1 are not necessarily good for nonnormal matrices, as they are for normal ones. However, we have the following two theorems.

Theorem 6.12 *If b is a linear combination of k eigenvectors of A, say*

$$b = \sum_{l=1}^{k} \gamma_l u_{i_l},$$

then the GMRES method will terminate in at most k iterations.

Proof We first extend the set of eigenvectors $\{u_{i_l}\}$ to be a basis of $\mathbb{R}^n$, i.e., the vectors

$$u_{i_1}, u_{i_2}, \cdots, u_{i_k}, v_1, \cdots, v_{n-k},$$

form a basis of $\mathbb{R}^n$. Let x_* be the exact solution of $Ax = b$. Then

$$x_* = \sum_{l=1}^{k} \alpha_l u_{i_l} + \sum_{j=1}^{n-k} \beta_j v_j.$$

Moreover, we have

$$\begin{aligned} Ax_* &= \sum_{l=1}^{k} \alpha_l A u_{i_l} + \sum_{j=1}^{n-k} \beta_j A v_j \\ &= \sum_{l=1}^{k} \alpha_l \lambda_{i_l} u_{i_l} + \sum_{j=1}^{n-k} \beta_j A v_j \\ &= b = \sum_{l=1}^{k} \gamma_l u_{i_l}. \end{aligned}$$

Hence,

$$\begin{cases} \beta_j = 0, \quad j = 1, 2, \cdots, n-k; \\ \alpha_l = \gamma_l / \lambda_{i_l}, \quad l = 1, 2, \cdots, k. \end{cases}$$

We therefore have

$$x_* = \sum_{l=1}^{k} (\gamma_l / \lambda_{i_l}) u_{i_l}.$$

Let

$$\bar{P}_k(z) = \prod_{l=1}^{k} \frac{\lambda_{i_l} - z}{\lambda_{i_l}} \in \mathcal{P}_k.$$

Note that $\bar{P}_k(\lambda_{i_l}) = 0$, for $1 \le l \le k$, and

$$\bar{P}_k(A) x_* = \sum_{l=1}^{k} \bar{P}_k(\lambda_{i_l})(\gamma_l / \lambda_{i_l}) u_{i_l} = 0.$$

We thus have

$$\begin{aligned} \|r_k\|_2 &\leq \|\bar{P}_k(A)r_0\|_2 = \|\bar{P}_k(A)b\|_2 \\ &= \|\bar{P}_k(A)Ax_*\|_2 = \|A\bar{P}_k(A)x_*\|_2 = 0, \end{aligned}$$

where we choose the initial vector $x_0 = 0$. □

Theorem 6.13 *When the GMRES method is applied for solving a linear system $Ax = b$ where $A = I+L$, the method will converge in at most rank$(L)+1$ iterations.*

Proof We first recall that the minimal polynomial of r_0 with respect to A is the nonzero monic polynomial p of the lowest degree such that $p(A)r_0 = 0$, see [33]. By Theorem 6.8, the GMRES method must converge within ν iterations, where ν is the degree of the minimal polynomials of the residual r_0 with respect to $A = I + L$. Let μ be the degree of the minimal polynomials of r_0 with respect to L. Since

$$\sum_{i=0}^{k} \alpha_i (I + L)^i r_0 = 0$$

implies

$$\sum_{i=0}^{k} \beta_i L^i r_0 = 0$$

for some constants β_i and vice versa, we have $\nu = \mu$. Moreover, from the definition of μ, the set

$$\{r_0, Lr_0, \cdots, L^{\mu-1}r_0\}$$

is linearly independent. Let B be the column vector space of L. Then, the dimension of B is equal to the rank of L. Since $L^i r_0 \in B$ for $i \geq 1$, we have

$$\{Lr_0, \cdots, L^{\mu-1}r_0\} \subseteq B.$$

Thus, $\mu - 1 \leq \text{rank}(L)$, i.e., $\mu \leq \text{rank}(L) + 1$. Hence, the GMRES method converges within

$$\nu = \mu \leq \text{rank}(L) + 1$$

iterations. □

6.5.2 Implementation of GMRES method

Recall that the LS problem from the k-th iteration of the GMRES method is

$$\min_{x \in x_0 + \mathcal{K}(A, r_0, k)} \|b - Ax\|_2.$$

Suppose that we have a matrix

$$V_k = [v_1^k, v_2^k, \cdots, v_k^k]$$

whose columns form an orthonormal basis of $\mathcal{K}(A, r_0, k)$. Then for any $z \in \mathcal{K}(A, r_0, k)$, it can be written as

$$z = \sum_{l=1}^{k} u_l v_l^k = V_k u,$$

where $u = (u_1, u_2, \cdots, u_k)^T \in \mathbb{R}^k$. Thus, once we find V_k, we can convert the original LS problem in the Krylov subspace into a LS problem in $\mathbb{R}^k$ as follows. Let x_k be the solution after the k-th iteration. We then have

$$x_k = x_0 + V_k y_k$$

where the vector y_k minimizes

$$\min_{y \in \mathbb{R}^k} \|b - A(x_0 + V_k y)\|_2 = \min_{y \in \mathbb{R}^k} \|r_0 - AV_k y\|_2.$$

This is a standard linear LS problem that can be solved by a QR decomposition.

One could use the Gram-Schmidt orthogonalization to find an orthonormal basis of $\mathcal{K}(A, r_0, k)$. The algorithm is given as follows:

(1) Define $r_0 = b - Ax_0$ and $v_1 = \dfrac{r_0}{\|r_0\|_2}$.

(2) Compute

$$v_{i+1} = \frac{Av_i - \sum_{j=1}^{i} ((Av_i)^T v_j) v_j}{\left\| Av_i - \sum_{j=1}^{i} ((Av_i)^T v_j) v_j \right\|_2}$$

for $i = 1, 2, \cdots, k-1$.

This algorithm produces the columns of the matrix V_k which are also an orthonormal basis for $\mathcal{K}(A, r_0, k)$. We note that a breakdown happens when a division by zero occurs. We have the following theorem for the breakdown happening.

Theorem 6.14 *Let A be nonsingular, the vectors v_j be generated by the above algorithm, and i be the smallest integer for which*

$$Av_i - \sum_{j=1}^{i}((Av_i)^T v_j)v_j = 0. \tag{6.14}$$

Then $x = A^{-1}b \in x_0 + \mathcal{K}(A, r_0, i)$.

Proof Since by (6.14),

$$Av_i = \sum_{j=1}^{i}((Av_i)^T v_j)v_j \in \mathcal{K}(A, r_0, i),$$

we know that

$$A\mathcal{K}(A, r_0, i) \subset \mathcal{K}(A, r_0, i).$$

Note that the columns of $V_i = [v_1, v_2, \cdots, v_i]$ form an orthonormal basis for $\mathcal{K}(A, r_0, i)$, i.e.,

$$\mathcal{K}(A, r_0, i) = \text{span}\{v_1, v_2, \cdots, v_i\},$$

and then

$$AV_i = V_i H \tag{6.15}$$

where $H \in \mathbb{R}^{i\times i}$ is nonsingular since A is nonsingular. There exists a vector $y \in \mathbb{R}^i$ such that $x_i - x_0 = V_i y$ because $x_i - x_0 \in \mathcal{K}(A, r_0, i)$. We therefore have

$$\|r_i\|_2 = \|b - Ax_i\|_2 = \|r_0 - A(x_i - x_0)\|_2 = \|r_0 - AV_i y\|_2. \tag{6.16}$$

Let $\beta = \|r_0\|_2$ and $e_1 = (1, 0, \cdots, 0)^T \in \mathbb{R}^i$. Then $r_0 = \beta V_i e_1$. Since V_i is a matrix with orthonormal columns, we have by (6.15) and (6.16),

$$\|r_i\|_2 = \|V_i(\beta e_1 - Hy)\|_2 = \|\beta e_1 - Hy\|_2.$$

Setting $y = \beta H^{-1}e_1$ and hence $r_i = 0$, i.e.,

$$x_i = A^{-1}b \in x_0 + \mathcal{K}(A, r_0, i). \qquad \square$$

If the Gram-Schmidt process does not breakdown, we can use it to carry out the GMRES method in the following efficient way. Let $h_{ij} = (Av_j)^T v_i$. By the Gram-Schmidt algorithm, we have a $(k+1)$-by-k matrix H_k which is

upper Hessenberg, i.e., its entries h_{ij} satisfy $h_{ij} = 0$ if $i > j + 1$. This process produces a sequence of matrices $\{V_k\}$ with orthonormal columns such that

$$AV_k = V_{k+1}H_k.$$

Therefore, we have

$$r_k = b - Ax_k = r_0 - A(x_k - x_0) = \beta V_{k+1}e_1 - AV_k y_k = V_{k+1}(\beta e_1 - H_k y_k),$$

where y_k is the solution of

$$\min_{y \in \mathbb{R}^k} \|\beta e_1 - H_k y\|_2.$$

Hence

$$x_k = x_0 + V_k y_k.$$

Let the GMRES iterations be ended when one finds a vector x such that for a given ϵ,

$$\|r\|_2 = \|b - Ax\|_2 \leq \epsilon \|b\|_2.$$

We then have the following GMRES algorithm for solving

$$Ax = b,$$

where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ are known, see [33]. At the initialization step, let

$$r_0 = b - Ax_0, \quad \beta = \|r_0\|_2, \quad v_1 = r_0/\beta.$$

In the iteration steps, we have

$$\begin{cases} \textbf{for } j = 1 : m \\ \quad w_j = Av_j \\ \quad \textbf{for } i = 1 : j \\ \quad\quad h_{ij} = w_j^T v_i \\ \quad\quad w_j = w_j - h_{ij} v_i \\ \quad \textbf{end} \\ \quad h_{j+1,j} = \|w_j\|_2; \text{ if } h_{j+1,j} = 0, \text{ set } m = j \text{ and go to } (*) \\ \quad v_{j+1} = w_j / h_{j+1,j} \\ \textbf{end} \\ (*) \text{ compute } y_m \text{ the minimizer of } \|\beta e_1 - \bar{H}_m y\|_2 \\ x_m = x_0 + V_m y_m \end{cases}$$

where $\bar{H}_m = (h_{ij})_{1 \leq i \leq m+1, 1 \leq j \leq m}$. The matrix $V_m = [v_1, v_2, \cdots, v_m] \in \mathbb{R}^{n \times m}$ with $m \leq n$ in the algorithm is a matrix with orthonormal columns.

Exercises:

1. Suppose that x_k is generated by the steepest descent method. Prove that

$$\phi(x_k) \leq \left(1 - \frac{1}{\kappa_2(A)}\right) \phi(x_{k-1}),$$

where $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$.

2. Let A be a symmetric positive definite matrix. Define $\|x\|_A \equiv \sqrt{x^T A x}$. Show that it is a vector norm.

3. Let $A \in \mathbb{R}^{n \times n}$ be symmetric positive definite, and $p_1, p_2, \cdots, p_k \in \mathbb{R}^n$ be mutually A-conjugate, i.e., $p_i^T A p_j = 0, i \neq j$. Prove that $\{p_1, p_2, \cdots, p_k\}$ is linearly independent.

4. Let x_k be produced by the CG method. Show that

$$\|x_k - x_*\|_2 \leq 2\sqrt{\kappa_2} \left(\frac{\sqrt{\kappa_2} - 1}{\sqrt{\kappa_2} + 1}\right)^k \|x_0 - x_*\|_2,$$

where $\kappa_2 = \kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$.

5. Suppose z_k and r_k are generated by the PCG method. Show that if $r_k \neq 0$, then $z_k^T r_k > 0$.

6. Find an efficient algorithm for solving $A^T A x = A^T b$ by the CG method.

7. Show that if $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite and has exactly k distinct eigenvalues, then the CG method will terminate in at most k iterations.

8. Let the initial vector $x_0 = 0$. When the GMRES method is used to solve the linear system $Ax = b$ where

$$A = \begin{bmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ & 1 & 0 & & \\ & & 1 & 0 & \\ & & & 1 & 0 \end{bmatrix}$$

and $b = (1, 0, 0, 0, 0)^T$, what is the convergence rate?

9. Let

$$A = \begin{bmatrix} I & Y \\ \mathbf{0} & I \end{bmatrix}.$$

When the GMRES method is used to solve $Ax = b$, what is the maximum number of iterations required to converge?

10. Prove that the LS problem in the GMRES method has full column rank.

11. Show that $c_U(A) = U^* \delta(UAU^*) U$.

Chapter 7
Nonsymmetric Eigenvalue Problems

Eigenvalue problems are particularly interesting in NLA. In this chapter, we study nonsymmetric eigenvalue problems. Some well-known methods, such as the power method, the inverse power method and the QR method, are discussed.

7.1 Basic properties

Let $A \in \mathbb{C}^{n\times n}$. We recall that a complex number λ is called an eigenvalue of A if there exists a nonzero vector $x \in \mathbb{C}^n$ such that

$$Ax = \lambda x.$$

Here x is called the eigenvector of A associated with λ. It is well-known that λ is an eigenvalue of A if and only if

$$\det(\lambda I - A) = 0.$$

Let

$$p_A(\lambda) = \det(\lambda I - A)$$

be the characteristic polynomial of A. By the Fundamental Theorem of Algebra, we know that $p_A(\lambda)$ has n roots in $\mathbb{C}$, i.e., A has n eigenvalues.

Now suppose that $p_A(\lambda)$ has the following factorization:

$$p_A(\lambda) = (\lambda - \lambda_1)^{n_1}(\lambda - \lambda_2)^{n_2}\cdots(\lambda - \lambda_p)^{n_p},$$

where $n_1 + n_2 + \cdots + n_p = n$, $\lambda_i \neq \lambda_j$ for $i \neq j$. The n_i is called the algebraic multiplicity of λ_i and the number

$$m_i = n - \operatorname{rank}(\lambda_i I - A)$$

is called the geometric multiplicity of λ_i. Actually, m_i is the dimension of the eigenspace of λ_i. We remark that the eigenspace of λ_i is the solution space of $(\lambda_i I - A)x = 0$. It is easy to see that $m_i \leq n_i$ for $i = 1, 2, \cdots, p$. If $n_i = 1$, then λ_i is called a simple eigenvalue. If $m_i < n_i$ for some eigenvalue λ_i, then

A is called defective. If the geometric multiplicity is equal to the algebraic multiplicity for each eigenvalue of A, then A is said to be nondefective. Note that A is diagonalizable if and only if A is nondefective.

Let $A, B \in \mathbb{C}^{n\times n}$. We recall that A and B are called similar matrices if there is a nonsingular matrix $X \in \mathbb{C}^{n\times n}$ such that

$$B = XAX^{-1}.$$

The transformation

$$A \to B = XAX^{-1}$$

is called a similarity transformation by the similarity matrix X. The similar matrices have the same eigenvalues. If x is an eigenvector of A, then $y = Xx$ is an eigenvector of B. By the Jordan Decomposition Theorem (Theorem 1.1), we know that any n-by-n matrix is similar to its Jordan canonical form. If the similarity matrix is required to be a unitary matrix, we then have the following perhaps the most fundamentally useful theorem in NLA, see [17].

Theorem 7.1 (Schur Decomposition Theorem) *Let $A \in \mathbb{C}^{n\times n}$ with the eigenvalues $\lambda_1, \cdots, \lambda_n$ in any prescribed order. Then there exists a unitary matrix $U \in \mathbb{C}^{n\times n}$ such that*

$$U^*AU = T = [t_{ij}]$$

where T is an upper triangular matrix with diagonal entries $t_{ii} = \lambda_i$, $i = 1, \cdots, n$. Furthermore, if $A \in \mathbb{R}^{n\times n}$ and if all the eigenvalues of A are real, then U may be chosen to be real and orthogonal.

Proof Let $x_1 \in \mathbb{C}^n$ be a normalized eigenvector of A associated with the eigenvalue λ_1. The vector x_1 can be extended to a basis of $\mathbb{C}^n$:

$$x_1, y_2, \cdots, y_n.$$

By applying the Gram-Schmidt orthonormalization procedure to this basis, we therefore have an orthonormal basis of $\mathbb{C}^n$:

$$x_1, z_2, \cdots, z_n.$$

Let

$$U_1 = [x_1, z_2, \cdots, z_n]$$

be a unitary matrix. By a simple calculation, we obtain

$$U_1^*AU_1 = \begin{bmatrix} \lambda_1 & * \\ \mathbf{0} & A_1 \end{bmatrix}.$$

The matrix $A_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ has the eigenvalues $\lambda_2, \cdots, \lambda_n$. Let $x_2 \in \mathbb{C}^{n-1}$ be a normalized eigenvector of A_1 associated with λ_2, and do it all over again. Determine a unitary matrix $V_2 \in \mathbb{C}^{(n-1)\times(n-1)}$ such that

$$V_2^* A_1 V_2 = \begin{bmatrix} \lambda_2 & * \\ \mathbf{0} & A_2 \end{bmatrix}.$$

Let

$$U_2 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & V_2 \end{bmatrix}.$$

Then the matrices U_2 and $U_1 U_2$ are unitary, and

$$U_2^* U_1^* A U_1 U_2 = \begin{bmatrix} \lambda_1 & & * \\ & \lambda_2 & \\ \mathbf{0} & & A_2 \end{bmatrix}.$$

Continuing this process, we can produce unitary matrices $U_1, U_2, \cdots, U_{n-1}$ such that the matrix

$$U = U_1 U_2 \cdots U_{n-1}$$

is unitary and $U^* A U$ yields the desired form. □

Theorem 7.2 (Real Schur Decomposition Theorem [17]) *Let $A \in \mathbb{R}^{n\times n}$. Then there exists an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that*

$$Q^T A Q = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ & R_{22} & \cdots & R_{2m} \\ & & \ddots & \vdots \\ \mathbf{0} & & & R_{mm} \end{bmatrix}$$

where R_{ii} is either a real number or a 2-by-2 matrix having a pair of complex conjugate eigenvalues.

In general, one cannot hope to reduce a real matrix to a strictly upper triangular form by using an orthogonal similarity transformation because the diagonal entries would then be eigenvalues, which could not be real.

7.2 Power method

The power method is an iterative algorithm for computing the eigenvalue with the largest absolute value and its corresponding eigenvector. Now, we introduce the basic idea of this method. For simplicity, we first suppose that the

matrix $A \in \mathbb{C}^{n\times n}$ is diagonalizable, i.e., A has the following Jordan decomposition:

$$A = X\Lambda X^{-1}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ with the order

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|,$$

and $X = [x_1, \cdots, x_n] \in \mathbb{C}^{n\times n}$. Let $u_0 \in \mathbb{C}^n$ be any vector. Since the column vectors $x_1, \cdots, x_n$ form a basis of $\mathbb{C}^n$, we have

$$u_0 = \sum_{j=1}^{n} \alpha_j x_j$$

where $\alpha_j \in \mathbb{C}$. We therefore have,

$$\begin{aligned} A^k u_0 &= \sum_{j=1}^{n} \alpha_j A^k x_j = \sum_{j=1}^{n} \alpha_j \lambda_j^k x_j \\ &= \lambda_1^k \left(\alpha_1 x_1 + \sum_{j=2}^{n} \alpha_j \Big(\frac{\lambda_j}{\lambda_1}\Big)^k x_j \right), \end{aligned}$$

and then

$$\lim_{k\to\infty} \frac{A^k u_0}{\lambda_1^k} = \alpha_1 x_1.$$

When $\alpha_1 \neq 0$ and k is sufficiently large, we know that the vector

$$u_k = \frac{A^k u_0}{\lambda_1^k} \tag{7.1}$$

is a good approximate eigenvector of A.

In practice, we cannot use (7.1) directly to compute an approximate eigenvector since we do not know the eigenvalue λ_1 in advance and the operation cost of A^k is very large when k is large. We therefore propose the following iterative algorithm:

$$\begin{cases} y_k = A u_{k-1}, \\ \mu_k = \zeta_j^{(k)}, \\ u_k = y_k/\mu_k, \end{cases} \tag{7.2}$$

where $u_0 \in \mathbb{C}^n$ is any given initial vector with $\|u_0\|_\infty = 1$ usually, and $\zeta_j^{(k)}$ is the largest absolute value of components of y_k. This iterative algorithm is called the power method. We have the following theorem for the convergence of the power method.

Theorem 7.3 *Let $A \in \mathbb{C}^{n\times n}$ with p distinct eigenvalues which satisfy*

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_p|.$$

The geometric multiplicity of λ_1 is equal to its algebraic multiplicity. If the projection of the initial vector u_0 on the eigenspace of λ_1 is nonzero, then $\{u_k\}$ produced by (7.2) converges to an eigenvector x_1 associated with λ_1. Also $\{\mu_k\}$ produced by (7.2) converges to λ_1.

Proof We note that A has the Jordan decomposition as follows,

$$A = X\text{diag}(J_1, \cdots, J_p)X^{-1}, \tag{7.3}$$

where $X \in \mathbb{C}^{n\times n}$, $J_i \in \mathbb{C}^{n_i\times n_i}$ is the Jordan block associated with λ_i ($i = 1, \cdots, p$), and

$$n_1 + n_2 + \cdots + n_p = n.$$

Since the geometric multiplicity of λ_1 is the same as its algebraic multiplicity, we have

$$J_1 = \lambda_1 I_{n_1}$$

where $I_{n_1} \in \mathbb{R}^{n_1\times n_1}$ is the identity matrix. Let $y = X^{-1}u_0$ and then decompose y and X as follows:

$$y = (y_1^T, y_2^T, \cdots, y_p^T)^T, \qquad X = [X_1, X_2, \cdots, X_p]$$

where $y_i \in \mathbb{C}^{n_i}$ and $X_i \in \mathbb{C}^{n\times n_i}$, for $i = 1, \cdots, p$. By using (7.3), we have

$$\begin{aligned} A^k u_0 &= X\text{diag}(J_1^k, \cdots, J_p^k)X^{-1}u_0 \\ &= X_1 J_1^k y_1 + X_2 J_2^k y_2 + \cdots + X_p J_p^k y_p \\ &= \lambda_1^k X_1 y_1 + X_2 J_2^k y_2 + \cdots + X_p J_p^k y_p \\ &= \lambda_1^k \left(X_1 y_1 + X_2(\lambda_1^{-1}J_2)^k y_2 + \cdots + X_p(\lambda_1^{-1}J_p)^k y_p\right). \end{aligned}$$

Note that the spectral radius of $\lambda_1^{-1}J_i$ satisfies

$$\rho(\lambda_1^{-1}J_i) = |\lambda_i|/|\lambda_1| < 1,$$

for $i = 2, 3, \cdots, p$. Therefore,

$$\lim_{k\to\infty} \frac{1}{\lambda_1^k} A^k u_0 = X_1 y_1. \tag{7.4}$$

Since the projection of the initial vector u_0 on the eigenspace of λ_1 is nonzero, we have $X_1 y_1 \neq 0$. Let

$$x_1 = \zeta^{-1} X_1 y_1$$

where ζ is the largest absolute value of components of $X_1 y_1$. Obviously, x_1 is an eigenvector of A associated with λ_1. Let ζ_k be the largest absolute value of components of $A^k u_0$. Then the largest absolute value of components of $\lambda_1^{-k} A^k u_0$ is $\zeta_k \lambda_1^{-k}$. By (7.2), we have

$$u_k = \frac{Au_{k-1}}{\mu_k} = \frac{A^k u_0}{\mu_k \mu_{k-1} \cdots \mu_1} = \frac{A^k u_0}{\zeta_k} = \frac{A^k u_0 / \lambda_1^k}{\zeta_k / \lambda_1^k}.$$

By using (7.4), we know that $\{u_k\}$ is convergent and

$$\lim_{k \to \infty} u_k = \frac{\lim_{k \to \infty} (A^k u_0 / \lambda_1^k)}{\lim_{k \to \infty} (\zeta_k / \lambda_1^k)} = \frac{X_1 y_1}{\zeta} = x_1.$$

By using $Au_{k-1} = \mu_k u_k$ and the fact that $\{u_k\}$ converges to an eigenvector associated with λ_1 having the largest absolute value of components equaling to 1, we immediately know that $\{\mu_k\}$ converges to λ_1. □

We remark that from the proof of Theorem 7.3, the convergence rate of the power method is determined by the value of $|\lambda_2|/|\lambda_1|$. Under the conditions of the theorem, we know that

$$\frac{|\lambda_2|}{|\lambda_1|} < 1.$$

The smaller the $|\lambda_2|/|\lambda_1|$ is, the faster the convergence rate will be. When $|\lambda_2|/|\lambda_1|$ is closed to 1, then the convergence rate will be very slow. In order to speed up the convergence of the power method, we could use the method on $A - \mu I$ where μ is called a shift. The μ could be chosen such that the distance between the eigenvalue with the largest absolute value and the other eigenvalues becomes larger. Therefore, the convergence rate of the power method can be increased.

7.3 Inverse power method

If one uses the power method on A^{-1} to obtain an eigenvalue of A with the smallest absolute value and its corresponding eigenvector, then this algorithm is called the inverse power method. Its basic iterative scheme is given as

follows:

$$\begin{cases} Ay_k = z_{k-1}, \\ \mu_k = \zeta_j^{(k)}, \\ z_k = y_k/\mu_k, \end{cases}$$

where $z_0 \in \mathbb{C}^n$ is any given initial vector and $\zeta_j^{(k)}$ is the largest absolute value of components of y_k. From Theorem 7.3, we know that if the eigenvalues of A satisfy

$$|\lambda_n| < |\lambda_{n-1}| \leq \cdots \leq |\lambda_1|,$$

then $\{z_k\}$ converges to an eigenvector associated with λ_n and $\{\mu_k\}$ converges to λ_n^{-1}. The convergence rate of the inverse power method is determined by $|\lambda_n|/|\lambda_{n-1}|$.

In practice, usually, the inverse power method is used on the matrix

$$A - \mu I$$

to compute an approximate eigenvector when an approximate eigenvalue μ of a distinct eigenvalue λ_i of A is obtained in advance. Therefore, we have the following inverse power method with a shift μ:

$$\begin{cases} (A - \mu I)v_k = z_{k-1}, \\ z_k = v_k/\|v_k\|_2. \end{cases} \tag{7.5}$$

From (7.5), we know that in each iteration of the inverse power method, one needs to solve a linear system. Hence, its operation cost is much larger than that of the power method. However, one could use the LU factorization with partial pivoting in advance and then in each iteration later on, one only needs to solve two triangular systems.

Suppose that the eigenvalues of the $A - \mu I$ are ordered as follows:

$$0 < |\lambda_1 - \mu| < |\lambda_2 - \mu| \leq |\lambda_3 - \mu| \leq \cdots \leq |\lambda_n - \mu|.$$

From Theorem 7.3 again, we know that the sequence $\{z_k\}$ produced by (7.5) converges to an eigenvector associated with λ_1. The convergence rate is determined by the value of $|\lambda_1 - \mu|/|\lambda_2 - \mu|$. The more closer of μ to λ_1, the faster the convergence rate will be. But if μ is closed to an eigenvalue of A, then $A - \mu I$ is closed to a singular matrix. Therefore, one needs to solve an ill-conditioned linear system in each iteration of the inverse power method. However, from practical computations, the illness of systems has no effect on the convergence rate of the method. Usually, only one iteration could produce a good approximate eigenvector of A if μ is closed to an eigenvalue of A.

7.4 QR method

In this section, we introduce the well-known QR method which is one of main important developments in matrix computations. For any given $A_0 = A \in \mathbb{C}^{n\times n}$, the basic iterative scheme of the QR algorithm is given as follows:

$$\begin{cases} A_{m-1} = Q_m R_m, \\ A_m = R_m Q_m, \end{cases} \tag{7.6}$$

for $m = 1, 2, \cdots$, where Q_m is a unitary matrix and R_m is an upper triangular matrix. For simplicity in later analysis, we require that diagonal entries of R_m are nonnegative. By (7.6), one can easily obtain

$$A_m = Q_m^* A_{m-1} Q_m, \tag{7.7}$$

i.e., each matrix in the sequence $\{A_m\}$ is similar to the matrix A. By using (7.7) again and again, we have

$$A_m = \widetilde{Q}_m^* A \widetilde{Q}_m, \tag{7.8}$$

where $\widetilde{Q}_m = Q_1 Q_2 \cdots Q_m$. Substituting $A_m = Q_{m+1} R_{m+1}$ into (7.8), we obtain

$$\widetilde{Q}_m Q_{m+1} R_{m+1} = A \widetilde{Q}_m.$$

Therefore,

$$\widetilde{Q}_m Q_{m+1} R_{m+1} R_m \cdots R_1 = A \widetilde{Q}_m R_m \cdots R_1,$$

i.e.,

$$\widetilde{Q}_{m+1} \widetilde{R}_{m+1} = A \widetilde{Q}_m \widetilde{R}_m,$$

where $\widetilde{R}_k = R_k R_{k-1} \cdots R_1$, for $k = m, m+1$. Moreover, we have

$$A^m = \widetilde{Q}_m \widetilde{R}_m. \tag{7.9}$$

Theorem 7.4 *Suppose that the eigenvalues of A satisfy:*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0.$$

Let Y be a matrix and y_i^T denote the i-th row of Y satisfying

$$y_i^T A = \lambda_i y_i^T.$$

If Y has an LU factorization, then the entries under the diagonal of the matrix

$$A_m = [\alpha_{ij}^{(m)}]$$

produced by (7.6) tend to zero as $m \longrightarrow \infty$. *At the same time,*

$$\alpha_{ii}^{(m)} \longrightarrow \lambda_i,$$

for $i = 1, 2, \cdots, n$.

Proof Let

$$X = Y^{-1}, \qquad \Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n).$$

Then $A = X\Lambda Y$. By assumption, Y has an LU factorization

$$Y = LU$$

where L is a unit lower triangular matrix and U is an upper triangular matrix. Hence

$$\begin{aligned} A^m &= X\Lambda^m Y = X\Lambda^m LU = X(\Lambda^m L\Lambda^{-m})\Lambda^m U \\ &= X(I + E_m)\Lambda^m U, \end{aligned} \tag{7.10}$$

where $I + E_m = \Lambda^m L\Lambda^{-m}$. Since L is a unit lower triangular matrix and $|\lambda_i| < |\lambda_j|$ for $i > j$, we have

$$\lim_{m\to\infty} E_m = 0. \tag{7.11}$$

Let $X = QR$ where Q is a unitary matrix and R is an upper triangular matrix. Since X is nonsingular, we can require that diagonal entries of R are positive. Substituting $X = QR$ into (7.10), we obtain

$$A^m = QR(I + E_m)\Lambda^m U = Q(I + RE_mR^{-1})R\Lambda^m U. \tag{7.12}$$

When m is sufficiently large, $I + RE_mR^{-1}$ is nonsingular and has the following QR decomposition,

$$I + RE_mR^{-1} = \widehat{Q}_m\widehat{R}_m, \tag{7.13}$$

where diagonal entries of $\widehat{R}_m$ are positive. By using (7.11) and (7.13), it is easy to show that

$$\lim_{m\to\infty} \widehat{Q}_m = \lim_{m\to\infty} \widehat{R}_m = I. \tag{7.14}$$

Substituting (7.13) into (7.12), we have

$$A^m = (Q\widehat{Q}_m)(\widehat{R}_m R\Lambda^m U),$$

which is a QR decomposition of A^m. In order to guarantee all the diagonal entries of the upper triangular matrix to be positive, we define

$$D_1 = \mathrm{diag}\left(\frac{\lambda_1}{|\lambda_1|}, \cdots, \frac{\lambda_n}{|\lambda_n|}\right),$$

and

$$D_2 = \text{diag}\left(\frac{u_{11}}{|u_{11}|}, \cdots, \frac{u_{nn}}{|u_{nn}|}\right),$$

where u_{ii} is the i-th diagonal entry of U. Hence, we have

$$A^m = (Q\widehat{Q}_m D_1^m D_2)(D_2^{-1} D_1^{-m} \widehat{R}_m R \Lambda^m U).$$

Comparing with (7.9) and noting that the QR decomposition is unique, we obtain

$$\widetilde{Q}_m = Q\widehat{Q}_m D_1^m D_2, \qquad \widetilde{R}_m = D_2^{-1} D_1^{-m} \widehat{R}_m R \Lambda^m U.$$

Substituting them into (7.8), we have

$$A_m = D_2^*(D_1^*)^m \widehat{Q}_m^* Q^* A Q \widehat{Q}_m D_1^m D_2.$$

Note that

$$A = X\Lambda Y = X\Lambda X^{-1} = QR\Lambda R^{-1} Q^*.$$

We finally have

$$A_m = D_2^*(D_1^*)^m \widehat{Q}_m^* R \Lambda R^{-1} \widehat{Q}_m D_1^m D_2.$$

When $m \longrightarrow \infty$, we know that by (7.14) the entries under the diagonal of the matrix A_m produced by (7.6) tend to zero. At the same time,

$$\alpha_{ii}^{(m)} \longrightarrow \lambda_i,$$

for $i = 1, 2, \cdots, n$. □

From Theorem 7.4, we know that the sequence $\{A_m\}$ produced by (7.6) converges to the Schur decomposition of A.

7.5 Real version of QR algorithm

If $A \in \mathbb{R}^{n\times n}$, we wish that we could find a fast and effective QR algorithm with real number operations only. We concentrate on developing the real analog of (7.6) as follows: let $A_1 = A$ and then construct an iterative algorithm:

$$\begin{cases} A_m = Q_m R_m, \\ \\ A_{m+1} = R_m Q_m, \end{cases} \tag{7.15}$$

for $m = 1, 2, \cdots$, where $Q_m \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $R_m \in \mathbb{R}^{n\times n}$ is an upper triangular matrix. A difficulty associated with (7.15) is that A_m can never converge to a strictly upper triangular form in the case that A has

complex eigenvalues. By Theorem 7.2, we can expect that (7.15) converges to the real Schur decomposition of A.

We note that in practice that (7.15) is not a good iterative method because the following two reasons:

(i) the operation cost in each iteration is too large;

(ii) the convergence rate is too slow.

We therefore need to reduce the operation cost in each iteration and increase the convergence rate of the method. Thus, the upper Hessenberg reduction and the shift strategy are introduced.

7.5.1 Upper Hessenberg reduction

We construct an orthogonal matrix Q_0 such that

$$Q_0^T A Q_0 = H \tag{7.16}$$

has some special structure (Hessenberg) with many zero entries. Afterwards, we can apply the QR algorithm (7.15) to the matrix H in (7.16). Then the operation cost per iteration can be dramatically reduced.

For $A = [\alpha_{ij}] \in \mathbb{R}^{n\times n}$, at the first step, we can choose a Householder transformation H_1 such that the first column of $H_1 A$ has many zero entries (at most $n-1$ zero entries). In order to keep a similarity transformation, we need to add one more column transformation:

$$H_1 A H_1.$$

Hence H_1 could have the following form

$$H_1 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{H}_1 \end{bmatrix} \tag{7.17}$$

to keep the zero entries in the first column unchanged. By using H_1 defined as in (7.17), we have

$$H_1 A H_1 = \begin{bmatrix} \alpha_{11} & a_2^T \widetilde{H}_1 \\ \widetilde{H}_1 a_1 & \widetilde{H}_1 A_{22} \widetilde{H}_1 \end{bmatrix} \tag{7.18}$$

where $a_1^T = (\alpha_{21}, \alpha_{31}, \cdots \alpha_{n1})$, $a_2^T = (\alpha_{12}, \alpha_{13}, \cdots \alpha_{1n})$, and A_{22} is the $(n-1)$-by-$(n-1)$ principal submatrix in the lower right corner of A. We know by (7.18) that the best choice of the Householder transformation $\widetilde{H}_1$ should be

$$\widetilde{H}_1 a_1 = p e_1 \tag{7.19}$$

where $p \in \mathbb{R}$ and $e_1 \in \mathbb{R}^{n-1}$ is the first unit vector. Therefore, the first column of the matrix in (7.18) has $n-2$ zeros by using H_1 defined by (7.17) and (7.19).

Afterwards, for $\widetilde{A}_{22} = \widetilde{H}_1 A_{22} \widetilde{H}_1$, we could find a Householder transformation

$$\widetilde{H}_2 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widehat{H}_2 \end{bmatrix}$$

such that

$$(\widetilde{H}_2 \widetilde{A}_{22} \widetilde{H}_2) e_1 = (*, *, 0, \cdots, 0)^T.$$

Let

$$H_2 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{H}_2 \end{bmatrix}.$$

We therefore have

$$H_2 H_1 A H_1 H_2 = \begin{bmatrix} h_{11} & h_{12} & \cdots & * \\ h_{21} & h_{22} & \vdots & \vdots \\ 0 & h_{32} & \vdots & * \\ 0 & 0 & \vdots & * \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & * \end{bmatrix}.$$

After $n-2$ steps, we have found $n-2$ Householder transformations H_1, $H_2, \cdots, H_{n-2}$, such that

$$H_{n-2} \cdots H_2 H_1 A H_1 H_2 \cdots H_{n-2} = H$$

where

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1,n-1} & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2,n-1} & h_{2n} \\ 0 & h_{32} & h_{33} & \cdots & h_{3,n-1} & h_{3n} \\ \vdots & 0 & h_{43} & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n,n-1} & h_{nn} \end{bmatrix}$$

with $h_{ij} = 0$ for $i > j+1$, and is called the upper Hessenberg matrix. Let

$$Q_0 = H_1 H_2 \cdots H_{n-2}$$

and therefore,

$$Q_0^T A Q_0 = H,$$

which is called the upper Hessenberg decomposition of A. We have the following algorithm by using Householder transformations.

Algorithm 7.1 (Upper Hessenberg decomposition)

$$\begin{cases} \textbf{for} \quad k = 1 : n-2 \\ \qquad [v, \beta] = \textbf{house}(A(k+1:n, k)) \\ \qquad A(k+1:n, k:n) = (I - \beta v v^T) A(k+1:n, k:n) \\ \qquad A(1:n, k+1:n) = A(1:n, k+1:n)(I - \beta v v^T) \\ \textbf{end} \end{cases}$$

The operation cost of the Hessenberg decomposition by Householder transformations is $O(n^3)$. Of course, the Hessenberg decomposition can also be obtained by using Givens rotations but the operation cost will be double of that by using Householder transformations. Although the Hessenberg decomposition of a matrix is not unique, we have the following theorem.

Theorem 7.5 *Suppose that $A \in \mathbb{R}^{n\times n}$ has the following two upper Hessenberg decompositions:*

$$U^T A U = H, \qquad V^T A V = G, \tag{7.20}$$

where

$$U = [u_1, u_2, \cdots, u_n], \qquad V = [v_1, v_2, \cdots, v_n]$$

are n-by-n orthogonal matrices, and $H = [h_{ij}]$, $G = [g_{ij}]$ are upper Hessenberg matrices. If $u_1 = v_1$ and all the entries $h_{i+1,i}$ are not zero, then there exists a diagonal matrix D with diagonal entries being either 1 or -1 such that

$$U = VD, \qquad H = DGD.$$

Proof Assume that we have proved for some m, $1 \le m < n$, that

$$u_j = \epsilon_j v_j, \qquad j = 1, 2, \cdots, m, \tag{7.21}$$

where $\epsilon_1 = 1$, $\epsilon_j = 1$ or -1. Now, we want to show that there exists $\epsilon_{m+1} = 1$ or -1 such that

$$u_{m+1} = \epsilon_{m+1} v_{m+1}.$$

From (7.20), we have

$$AU = UH, \qquad AV = VG.$$

Comparing with the m-th column of above matrix equalities respectively, we obtain

$$Au_m = h_{1m}u_1 + \cdots + h_{mm}u_m + h_{m+1,m}u_{m+1}, \tag{7.22}$$

and

$$Av_m = g_{1m}v_1 + \cdots + g_{mm}v_m + g_{m+1,m}v_{m+1}. \tag{7.23}$$

Multiplying (7.22) and (7.23) by u_i^T and v_i^T, respectively, we have

$$h_{im} = u_i^T A u_m, \qquad g_{im} = v_i^T A v_m, \qquad i = 1, 2, \cdots, m.$$

Therefore by (7.21),

$$h_{im} = \epsilon_i \epsilon_m g_{im}, \qquad i = 1, 2, \cdots, m. \tag{7.24}$$

Substituting (7.24) into (7.22), and using (7.21) and (7.23), we obtain

$$\begin{aligned} h_{m+1,m}u_{m+1} &= \epsilon_m(Av_m - \epsilon_1^2 g_{1m}v_1 - \cdots - \epsilon_m^2 g_{mm}v_m) \\ &= \epsilon_m(Av_m - g_{1m}v_1 - \cdots - g_{mm}v_m) \qquad (7.25) \\ &= \epsilon_m g_{m+1,m}v_{m+1}. \end{aligned}$$

Hence,

$$|h_{m+1,m}| = |g_{m+1,m}|.$$

Since $h_{m+1,m} \neq 0$, from (7.25), we know that

$$u_{m+1} = \epsilon_{m+1}v_{m+1},$$

where $\epsilon_{m+1} = 1$ or -1. By induction, the proof is complete. □

We note that for an upper Hessenberg matrix $H = [h_{ij}]$, if $h_{i+1,i} \neq 0$, $i = 1, 2, \cdots, n-1$, then it is irreducible. Theorem 7.5 said that if $Q^T A Q = H$ is irreducible upper Hessenberg where Q is an orthogonal matrix, then Q and H are determined completely by the first column of Q (up to ± 1).

Now, suppose that $H \in \mathbb{R}^{n\times n}$ is an upper Hessenberg matrix, we want to apply a QR iteration to H. Firstly, we need to construct a QR decomposition for H. Since H has a special structure, we can use $n-1$ Givens rotations to obtain the QR decomposition. For simplicity, in the following, we consider the case of $n = 5$. Suppose that we have already found two Givens rotations P_{12} and P_{23} such that

$$P_{23}P_{12}H = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & h_{33} & \times & \times \\ 0 & 0 & h_{43} & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

Then we construct a Givens rotation $P_{34} = G(3, 4, \theta_3)$ such that the θ_3 satisfies

$$\begin{bmatrix} \cos\theta_3 & \sin\theta_3 \\ -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} h_{33} \\ h_{43} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}.$$

Hence,

$$P_{34}P_{23}P_{12}H = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

Therefore, it is easy to see that for n-by-n upper Hessenberg matrix H, we can construct $n-1$ Givens rotations P_{12}, $P_{23}, \cdots, P_{n-1,n}$ such that

$$P_{n-1,n}P_{n-2,n-1}\cdots P_{12}H = R$$

is an upper triangular matrix. Let

$$Q = (P_{n-1,n}P_{n-2,n-1}\cdots P_{12})^T.$$

Then we have $H = QR$. In order to complete a QR iteration, we have to compute

$$\widetilde{H} = RQ = RP_{12}^T P_{23}^T \cdots P_{n-1,n}^T.$$

Note that RP_{12}^T is different from R only in the first two columns. Since R is an upper triangular matrix, RP_{12}^T should have the following form (for $n = 5$),

$$RP_{12}^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}.$$

Similarly, $RP_{12}^T P_{23}^T$ is different from RP_{12}^T only in the second column and the third column. Hence,

$$RP_{12}^T P_{23}^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}.$$

Continuously, we finally obtain the matrix $\widetilde{H}$ which is also an upper Hessenberg matrix. It is easy to know that the operation cost of a QR iteration for an upper Hessenberg matrix is $O(n^2)$. Note that the operation cost of a QR iteration for a general matrix is $O(n^3)$.

7.5.2 QR iteration with single shift

Now, we only need to discuss the Hessenberg form. From Theorem 7.4, we know that the convergence rate of the basic QR algorithm is linear and is depending on the distance between eigenvalues. In order to speed up the convergence rate, we introduce the shift strategy. The QR iteration with a single shift is given as follows:

$$\begin{cases} H_m - \mu_m I = Q_m R_m, \\ \\ H_{m+1} = R_m Q_m + \mu_m I, \end{cases} \tag{7.26}$$

for $m = 1, 2, \cdots$, where $H_1 = H \in \mathbb{R}^{n \times n}$ is a given upper Hessenberg matrix satisfying the conditions of Theorem 7.4 and $\mu_m \in \mathbb{R}$. We consider how to choose a shift if all the eigenvalues of H are assumed to be real. Since H_m is upper Hessenberg, there are only two nonzero entries $h_{n,n-1}^{(m)}$ and $h_{nn}^{(m)}$ in the last row (for $n = 5$):

$$H_m = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & h_{n,n-1}^{(m)} & h_{nn}^{(m)} \end{bmatrix}$$

When the QR algorithm converges, $h_{n,n-1}^{(m)}$ will be very small and $h_{nn}^{(m)}$ will approach to an eigenvalue of H. Therefore, we can choose a shift $\mu_m = h_{nn}^{(m)}$. In fact, we can prove that if $h_{n,n-1}^{(m)} = \epsilon$ is very small, then after one iteration of the QR algorithm, we have

$$h_{n,n-1}^{(m+1)} = O(\epsilon^2). \tag{7.27}$$

From the discussion above, we know that there are $n - 1$ steps to reduce $H_m - h_{nn}^{(m)} I$ to be an upper triangular matrix. Assume that the first $n - 2$ steps are completed:

$$\widehat{H} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \alpha & \beta \\ 0 & 0 & 0 & \epsilon & 0 \end{bmatrix}.$$

Actually, we only need to study the 2-by-2 submatrix

$$H_{m2} = \begin{bmatrix} \alpha & \beta \\ \epsilon & 0 \end{bmatrix}$$

in the lower right corner of the matrix $\widehat{H}$. In the $(n-1)$-th step of reduction, we want to determine $c = \cos\theta$ and $s = \sin\theta$ such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \alpha \\ \epsilon \end{bmatrix} = \begin{bmatrix} \sigma \\ 0 \end{bmatrix}.$$

It is easy to see that

$$c = \alpha/\sigma, \qquad s = \epsilon/\sigma, \qquad \sigma = \sqrt{\alpha^2 + \epsilon^2}.$$

Thus, after a few simple computations on the 2-by-2 submatrix in the lower right corner of the matrix $H_{m+1} = R_m Q_m + h_{nn}^{(m)} I$, we have

$$h_{n,n-1}^{(m+1)} = -s^2\beta = -\beta\epsilon^2/\sigma^2 = O(\epsilon^2),$$

i.e., (7.27) holds. Through a shift, the convergence rate of the QR iteration is expected to be quadratic. If H has complex eigenvalues, then a double shift strategy can be used to speed up the convergence rate of the QR iteration.

7.5.3 QR iteration with double shift

Note that difficulties with (7.26) can be expected if the submatrix

$$G = \begin{bmatrix} h_{pp}^{(m)} & h_{pn}^{(m)} \\ h_{np}^{(m)} & h_{nn}^{(m)} \end{bmatrix}, \qquad p = n-1$$

in the lower right corner of H_m has a pair of complex conjugate eigenvalues μ_1 and μ_2. We cannot expect that $h_{nn}^{(m)}$ tends to an eigenvalue of A. A way around this difficulty is to perform the following QR algorithm with double shifts:

$$\begin{cases} H - \mu_1 I = Q_1 R_1, \\ H_1 = R_1 Q_1 + \mu_1 I, \\ H_1 - \mu_2 I = Q_2 R_2, \\ H_2 = R_2 Q_2 + \mu_2 I, \end{cases} \tag{7.28}$$

where $H = H_m$. Let

$$M \equiv (H - \mu_1 I)(H - \mu_2 I). \tag{7.29}$$

By a few simple computations, we have

$$M = QR, \tag{7.30}$$

and

$$H_2 = Q^T H Q, \tag{7.31}$$

where

$$Q = Q_1 Q_2, \qquad R = R_2 R_1.$$

By (7.29), we obtain

$$M = H^2 - sH + tI,$$

where

$$s = \mu_1 + \mu_2 = h_{pp}^{(m)} + h_{nn}^{(m)} \in \mathbb{R},$$

and

$$t = \mu_1 \mu_2 = \det(G) \in \mathbb{R}.$$

Hence M is also real. If μ_1, μ_2 are not eigenvalues of H and diagonal entries of R_1, R_2 in each iteration are chosen to be positive, then by using (7.30), Q is also real. By (7.31), it then follows that H_2 is real. Therefore, under an assumption of without rounding error, by using (7.28), H_2 is still a real upper Hessenberg matrix. In practice, because of rounding error, usually, H_2 is not real. In order to keep the reality of H_2, by using (7.30) and (7.31), we propose the following process to compute H_2:

(1) Compute $M = H^2 - sH + tI$.

(2) Compute the QR decomposition of M: $M = QR$.

(3) Compute $H_2 = Q^T H Q$.

Note that the operation cost of forming the matrix M in (1) needs $O(n^3)$. We remark that actually, we do not need to form matrix M explicitly, see [14]. For a practical implementation of the QR algorithm with the shift strategy, we refer to [14, 19, 47].

Exercises:

1. Show that if $T \in \mathbb{C}^{n\times n}$ is upper triangular and normal, then T is diagonal.
2. Let A, $B \in \mathbb{C}^{n\times n}$. Prove that the spectrum of AB is equal to the spectrum of BA.
3. Let $A \in \mathbb{C}^{n\times n}$, $x \in \mathbb{C}^n$ and $X = [x, Ax, \cdots, A^{n-1}x]$. Show that if X is nonsingular, then $X^{-1}AX$ is an upper Hessenberg matrix.
4. Suppose that $A \in \mathbb{C}^{n\times n}$ has distinct eigenvalues. Show that if $Q^*AQ = T$ is the Schur decomposition and $AB = BA$, then Q^*BQ is upper triangular.

5. Suppose that $A \in \mathbb{R}^{n\times n}$ and $z \in \mathbb{R}^n$. Find a detailed algorithm for computing an orthogonal matrix Q such that Q^TAQ is upper Hessenberg and Q^Tz is a multiple of e_1 where e_1 is the first unit vector.

6. Suppose that $W, Y \in \mathbb{R}^{n\times n}$ and define matrices C, B by

$$C = W + \mathbf{i}Y, \quad B = \begin{bmatrix} W & -Y \\ Y & W \end{bmatrix}.$$

Show that if $\lambda \in \mathbb{R}$ is an eigenvalue of C, then λ is also an eigenvalue of B. What is the relation between two corresponding eigenvectors?

7. Suppose that

$$A = \begin{bmatrix} w & x \\ y & z \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

has eigenvalues $\lambda \pm \mathbf{i}\mu$, where $\mu \neq 0$. Find an algorithm that determines $c = \cos\theta$ and $s = \sin\theta$ stably such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} w & x \\ y & z \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \lambda & \beta \\ \alpha & \lambda \end{bmatrix},$$

where $\alpha\beta = -\mu^2$.

8. Find a 2-by-2 diagonal matrix D that minimizes $\|D^{-1}AD\|_F$ where

$$A = \begin{bmatrix} w & x \\ y & z \end{bmatrix}.$$

9. Let $H = H_1$ be a given matrix. We generate matrices H_k via

$$H_k - \mu_k I = Q_k R_k, \quad H_{k+1} = R_k Q_k + \mu_k I.$$

Show that

$$(Q_1 \cdots Q_j)(R_j \cdots R_1) = (H - \mu_1 I) \cdots (H - \mu_j I).$$

10. Show that if

$$Y = \begin{bmatrix} I & Z \\ \mathbf{0} & I \end{bmatrix},$$

then

$$\kappa_2(Y) \equiv \|Y\|_2 \|Y^{-1}\|_2 = \frac{1}{2}\left(2 + \sigma^2 + \sqrt{4\sigma^2 + \sigma^4}\right),$$

where $\sigma = \|Z\|_2$.

11. Let A be a matrix with real diagonal entries. Show that

$$|\mathrm{Im}(\lambda)| \leq \max_i \sum_{j\neq i} |a_{ij}|,$$

where λ is any eigenvalue of A and $\mathrm{Im}(\cdot)$ denotes the imaginary part of a complex number.

12. Show that if

$$\frac{1}{2}(A + A^T)$$

is positive definite, then $\mathrm{Re}(\lambda) > 0$, where λ is any eigenvalue of A and $\mathrm{Re}(\cdot)$ denotes the real part of a complex number.

13. Let B be a matrix with $\|B\|_2 < 1$. Show that $I - B$ is nonsingular and the eigenvalues of $I + 2(B - I)^{-1}$ have negative real parts.

Chapter 8
Symmetric Eigenvalue Problems

The symmetric eigenvalue problem with its nice properties and rich mathematical theory is one of the most pleasing topics in NLA. In this chapter, we will study symmetric eigenvalue problems. We begin by introducing some basic spectral properties of symmetric matrices. Then the symmetric QR method, the Jacobi method and the bisection method are discussed. Finally, a divide-and-conquer algorithm is described.

8.1 Basic spectral properties

We first introduce some basic properties of eigenvalues and eigenvectors of any symmetric matrix $A \in \mathbb{R}^{n\times n}$. It is well known that the eigenvalues of any symmetric matrix A are real and there is an orthonormal basis of $\mathbb{R}^n$ formed by the eigenvectors of A.

Theorem 8.1 (Spectral Decomposition Theorem [17]) *If $A \in \mathbb{R}^{n\times n}$ is symmetric, then there exists an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that*

$$Q^T A Q = \mathrm{diag}(\lambda_1, \cdots, \lambda_n).$$

The eigenvalues of a symmetric matrix have minimax properties based on the values called the Rayleigh quotient:

$$\frac{x^T A x}{x^T x}.$$

We have the following theorem and its proof can be found in [19].

Theorem 8.2 (Courant-Fischer Minimax Theorem) *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and its eigenvalues be ordered as*

$$\lambda_1 \geq \cdots \geq \lambda_n.$$

Then

$$\lambda_i = \max_{dim(S)=i} \min_{0\neq u\in S} \frac{u^T A u}{u^T u} = \min_{dim(S)=n-i+1} \max_{0\neq u\in S} \frac{u^T A u}{u^T u},$$

where S is any subspace of $\mathbb{R}^n$.

The next theorem [46] shows the sensitivity of eigenvalues of symmetric matrices.

Theorem 8.3 (Weyl, Wielandt-Hoffman Theorem) *If A, $B \in \mathbb{R}^{n\times n}$ are symmetric matrices and the eigenvalues of A, B are ordered respectively as follows,*

$$\lambda_1(A) \geq \cdots \geq \lambda_n(A), \qquad \mu_1(B) \geq \cdots \geq \mu_n(B),$$

then

$$|\lambda_i(A) - \mu_i(B)| \leq \|A - B\|_2, \qquad i = 1, 2, \cdots, n,$$

and

$$\sum_{i=1}^{n} (\lambda_i(A) - \mu_i(B))^2 \leq \|A - B\|_F^2.$$

Theorem 8.3 tells us that the eigenvalues of any symmetric matrix are well conditioned, i.e., small perturbations on the entries of A cause only small changes in the eigenvalues of A.

Theorem 8.4 (Cauchy Interlace Theorem [17, 46]) *If $A \in \mathbb{R}^{n\times n}$ is symmetric and A_r denotes the r-by-r leading principal submatrix of A, then*

$$\lambda_{r+1}(A_{r+1}) \leq \lambda_r(A_r) \leq \lambda_r(A_{r+1}) \leq \cdots \leq \lambda_2(A_{r+1}) \leq \lambda_1(A_r) \leq \lambda_1(A_{r+1}),$$

for $r = 1, 2, \cdots, n-1$.

As for the sensitivity of eigenvectors, we have the following theorem, see [46].

Theorem 8.5 *Suppose A, $A + E \in \mathbb{R}^{n\times n}$ are symmetric matrices and*

$$Q = [q_1, Q_2] \in \mathbb{R}^{n\times n}$$

is an orthogonal matrix where q_1 is a unit eigenvector of A. Partition the matrices Q^TAQ and Q^TEQ as follows:

$$Q^TAQ = \begin{bmatrix} \lambda & \mathbf{0} \\ \mathbf{0} & D_{22} \end{bmatrix}, \qquad Q^TEQ = \begin{bmatrix} \epsilon & e^T \\ e & E_{22} \end{bmatrix},$$

where D_{22}, $E_{22} \in \mathbb{R}^{(n-1)\times(n-1)}$. If

$$d = \min_{\mu} |\lambda - \mu| > 0, \qquad \|E\|_2 \leq d/4,$$

where μ is any eigenvalue of D_{22}, then there exists a unit eigenvector $\tilde{q}_1$ such that

$$\sin\theta = \sqrt{1 - |q_1^T \tilde{q}_1|^2} \leq \frac{4}{d}\|e\|_2 \leq \frac{4}{d}\|E\|_2,$$

where $\theta = \arccos|q_1^T \tilde{q}_1|$.

It seems that θ could be a good measurement between q_1 and $\tilde{q}_1$. We can see that the sensitivity to a perturbation of a single eigenvector depends on the separation of its corresponding eigenvalue from the rest of eigenvalues.

The eigenvalues of any symmetric matrix are closely related with the singular values of the matrix. The singular value decomposition [19] is essential in NLA.

Theorem 8.6 (Singular Value Decomposition Theorem) *Let $A \in \mathbb{R}^{m\times n}$ with $rank(A) = r$. Then there exist orthogonal matrices*

$$U = [u_1, \cdots, u_m] \in \mathbb{R}^{m\times m}, \qquad V = [v_1, \cdots, v_n] \in \mathbb{R}^{n\times n}$$

such that

$$U^T A V = \begin{bmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where

$$\Sigma_r = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

The σ_i are called the singular values of A. The vectors u_i and v_i are called the i-th left singular vector and the i-th right singular vector respectively. The next corollary is the Weyl, Wielandt-Hoffman Theorem for the singular values.

Corollary 8.1 *Let A, $B \in \mathbb{R}^{n\times n}$ and their singular values be ordered respectively as follows:*

$$\sigma_1(A) \geq \cdots \geq \sigma_n(A), \qquad \tau_1(B) \geq \cdots \geq \tau_n(B),$$

then

$$|\sigma_i(A) - \tau_i(B)| \leq \|A - B\|_2, \quad i = 1, 2, \cdots, n,$$

and

$$\sum_{i=1}^{n} (\sigma_i(A) - \tau_i(B))^2 \leq \|A - B\|_F^2.$$

The corollary shows that the singular values of any real matrix are also well conditioned, i.e., small perturbations on the entries of A cause only small changes in the singular values of A.

8.2 Symmetric QR method

The symmetric QR method is a QR iteration for solving symmetric eigenvalue problem. It applies the QR algorithm to any symmetric matrix A by using its symmetry. In order to construct an efficient QR method, we first reduce the symmetric matrix A to a tridiagonal matrix T and then apply the QR iteration to the matrix T.

8.2.1 Tridiagonal QR iteration

Let A be symmetric and suppose that A can be decomposed as

$$Q^T A Q = T,$$

where Q is an orthogonal matrix and T is an upper Hessenberg matrix. Then, T should be symmetric tridiagonal. In this case we only need to handle with an eigenvalue problem of a symmetric tridiagonal matrix T.

Partition A as follows,

$$A = \begin{bmatrix} \alpha_1 & v_0^T \\ v_0 & A_0 \end{bmatrix}$$

where $A_0 \in \mathbb{R}^{(n-1)\times(n-1)}$. By using Householder transformations, we can reduce the matrix A into a symmetric tridiagonal matrix as follows: at the k-th step,

(1) Compute a Householder transformation $\tilde{H}_k \in \mathbb{R}^{(n-k)\times(n-k)}$ such that

$$\tilde{H}_k v_{k-1} = \beta_k e_1, \qquad \beta_k \in \mathbb{R}.$$

(2) Compute

$$\begin{bmatrix} \alpha_{k+1} & v_k^T \\ v_k & A_k \end{bmatrix} = \tilde{H}_k A_{k-1} \tilde{H}_k,$$

where $A_k \in \mathbb{R}^{(n-k-1)\times(n-k-1)}$.

If we use α_k, β_k and $\tilde{H}_k$ generated by the reduction above to define

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \mathbf{0} \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ \mathbf{0} & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

$$H_k = \mathrm{diag}(I_k, \tilde{H}_k) \in \mathbb{R}^{n\times n}, \qquad Q = H_1 H_2 \cdots H_{n-2},$$

where

$$\begin{bmatrix} \alpha_{n-1} & \beta_{n-1} \\ \beta_{n-1} & \alpha_n \end{bmatrix} = \tilde{H}_{n-2} A_{n-3} \tilde{H}_{n-2},$$

then we have

$$Q^T A Q = T.$$

From the reduction above, it is easy to see that the main operation cost of the k-th step is to compute $\tilde{H}_k A_{k-1} \tilde{H}_k$. Let

$$\tilde{H}_k = I - \beta v v^T, \qquad v \in \mathbb{R}^{n-k}.$$

By using the symmetry of A_{k-1}, we then have

$$\tilde{H}_k A_{k-1} \tilde{H}_k = A_{k-1} - v w^T - w v^T$$

where

$$w = u - \frac{1}{2}\beta(v^T u)v, \qquad u = \beta A_{k-1} v.$$

Since only the upper triangular portion of this matrix needs to be computed, we see that the transition from A_{k-1} to A_k can be computed in $4(n-k)^2$ operations only. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the following algorithm overwrites A with $T = Q^T A Q$, where T is a tridiagonal matrix and Q is a product of Householder transformations.

Algorithm 8.1 (Householder tridiagonalization)

for $k = 1 : n-2$
  $[v, \beta] = \mathbf{house}(A(k+1:n, k))$
  $u = \beta A(k+1:n, k+1:n) v$
  $w = u - (\beta u^T v/2) v$
  $A(k+1, k) = \|A(k+1:n, k)\|_2$
  $A(k, k+1) = A(k+1, k)$
  $A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - v w^T - w v^T$
end

This algorithm requires $4n^3/3$ operations. If Q is explicitly required, then it can be formed with additional $4n^3/3$ operations.

After a symmetric matrix A has been reduced to a symmetric tridiagonal matrix T, our aim turns to choose a suitable shift for the QR iteration. Consider the following QR iteration with a single shift μ_k:

$$\begin{cases} T_k - \mu_k I = Q_k R_k, \\ \\ T_{k+1} = R_k Q_k + \mu_k I, \end{cases} \tag{8.1}$$

for $k = 1, 2, \cdots$, where $T_1 = T$ is a symmetric tridiagonal matrix and so is each matrix T_k in (8.1). Just as the QR algorithm for nonsymmetric case, T_k is assumed to be irreducible, i.e., sub-diagonal entries are nonzero. Let us discuss how to choose the shift μ_k. We can take $\mu_k = T_k(n,n)$, the (n,n)-th entry at each iteration as a shift. However, a better way is to select

$$\mu_k = \alpha_n + \delta - sign(\delta)\sqrt{\delta^2 + \beta_{n-1}^2},$$

where $\delta = (\alpha_{n-1} - \alpha_n)/2$. This is the well-known Wilkinson shift, see [46]. Note that μ_k is just the eigenvalue of the matrix

$$T_k(n-1:n, n-1:n) = \begin{bmatrix} \alpha_{n-1} & \beta_{n-1} \\ \beta_{n-1} & \alpha_n \end{bmatrix}$$

which is closer to α_n.

8.2.2 Implicit symmetric QR iteration

For the explicit QR algorithm (8.1), it is possible to execute the transition from

$$T - \mu I = QR$$

to

$$\tilde{T} = RQ + \mu I$$

without explicitly forming the matrix $T - \mu I$. The essence of (8.1) is to transform T to $\tilde{T}$ by orthogonal similarity transformations. It follows from Theorem 7.5 that $\tilde{T}$ can be determined completely by the first column of Q. From the process of the QR decomposition of $T - \mu I$ by Givens rotations, we know that

$$Qe_1 = G_1 e_1,$$

where $G_1 = G(1, 2, \theta_1)$ is a rotation which makes the second entry in the first column of $T - \mu I$ to be zero. The θ_1 can be computed from

$$\begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} \alpha_1 - \mu \\ \beta_1 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}.$$

Let

$$B = G_1 T G_1^T.$$

Then B has the following form ($n = 4$),

$$B = \begin{bmatrix} * & * & + & 0 \\ * & * & * & 0 \\ + & * & * & * \\ 0 & 0 & * & * \end{bmatrix}.$$

Let $G_i = G(i, i+1, \theta_i)$, $i = 2, 3$. Then G_i of this form can be used to chase the unwanted nonzero entry "+" out of the matrix B as follows:

$$B \xrightarrow{G_2} \begin{bmatrix} * & * & 0 & 0 \\ * & * & * & + \\ 0 & * & * & * \\ 0 & + & * & * \end{bmatrix} \xrightarrow{G_3} \begin{bmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix}.$$

In general, if $Z = G_1 G_2 \cdots G_{n-1}$, then

$$Ze_1 = G_1 e_1 = Qe_1$$

and ZTZ^T is tridiagonal. Thus, from Theorem 7.5, the tridiagonal matrix ZTZ^T produced by this zero-chasing technique is essentially the same as the tridiagonal matrix T obtained by the explicit method (8.1). Overall, we obtain

Algorithm 8.2(Implicit symmetric QR iteration with Wilkinson shift)

$$\left\{ \begin{array}{l} d = (T(n-1, n-1) - T(n,n))/2 \\ \mu = T(n,n) - T(n,n-1)^2/(d + sign(d)\sqrt{d^2 + T(n,n-1)^2}) \\ x = T(1,1) - \mu; \quad z = T(2,1) \\ \textbf{for} \quad k = 1 : n-1 \\ \quad [c, s] = \textbf{givens}(x, z) \\ \quad T = G_k T G_k^T, \quad where \quad G_k = G(k, k+1, \theta_k) \\ \quad \textbf{if} \quad k < n-1 \\ \quad\quad x = T(k+1, k); \quad z = T(k+2, k) \\ \quad \textbf{end} \\ \textbf{end} \end{array} \right.$$

This algorithm requires about $30n$ operations and n square roots. Of course, the tridiagonal matrix T would be stored in a pair of n-vectors in any practical implementation.

8.2.3 Implicit symmetric QR algorithm

Algorithm 8.2 is a base of the symmetric QR algorithm – the standard means for computing the spectral decomposition of any symmetric matrix. By applying Algorithm 8.2, we develop the following algorithm.

Algorithm 8.3 (Implicit symmetric QR algorithm)

(1) *Input A (real symmetric matrix).*

(2) *Tridiagonalization: Compute the tridiagonalization of A by Algorithm 8.1,*

$$T = U_0^T A U_0.$$

Set $Q = U_0$.

(3) *Criterion for convergence:*

(i) *For* $i = 1, \cdots, n-1$, *let* $t_{i+1,i}$ *and* $t_{i,i+1}$ *be zero if*

$$|t_{i+1,i}| = |t_{i,i+1}| \leq (|t_{i,i}| + |t_{i+1,i+1}|)\mathbf{u},$$

where $\mathbf{u}$ *is the machine precision.*

(ii) *Find the largest integer* $m \geq 0$ *and the smallest integer* $l \geq 0$ *such that*

$$T = \begin{bmatrix} T_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & T_{33} \end{bmatrix},$$

where $T_{11} \in \mathbb{R}^{l \times l}$, $T_{22} \in \mathbb{R}^{(n-l-m)\times(n-l-m)}$ *is an irreducible tridiagonal matrix and* $T_{33} \in \mathbb{R}^{m \times m}$ *is a diagonal matrix.*

(iii) *If* $m = n$, *then output; otherwise*

(4) *QR iteration: Apply Algorithm 8.2 to* T_{22}*:*

$$T_{22} = G T_{22} G^T, \qquad G = G_1 G_2 \cdots G_{n-l-m-1}.$$

(5) *Set* $Q = Q\mathrm{diag}(I_l, G, I_m)$, *then go to (3).*

If we only need to compute the eigenvalues, this algorithm requires about $4n^3/3$ operations. If we need both the eigenvalues and eigenvectors, it requires about $9n^3$ operations. It can be shown [46] that the computed eigenvalues $\tilde{\lambda}_i$, $i = 1, 2, \cdots, n$, obtained by Algorithm 8.3, satisfy

$$Q^T(A + E)Q = \mathrm{diag}(\tilde{\lambda}_1, \cdots, \tilde{\lambda}_n),$$

where $Q \in \mathbb{R}^{n \times n}$ is orthogonal and $\|E\|_2 \approx \|A\|_2 \mathbf{u}$ where $\mathbf{u}$ is the machine precision. Using Theorem 8.3, we have

$$|\lambda_i - \tilde{\lambda}_i| \approx \|A\|_2 \mathbf{u}, \qquad i = 1, 2, \cdots, n,$$

where $\{\lambda_i\}$ are the eigenvalues of A. The absolute error in each $\tilde{\lambda}_i$ is small and the relative error is less than the machine precision $\mathbf{u}$. If $\tilde{Q} = [\tilde{q_1}, \cdots, \tilde{q_n}]$ is the matrix of computed orthonormal eigenvectors, then the accuracy of each $\tilde{q_i}$ depends on the separation of λ_i from the rest of eigenvalues.

8.3 Jacobi method

Jacobi method is one of the earliest methods for computing the symmetric eigenvalue problem. It was developed by Jacobi in 1846, see [19]. It is well-known that a real symmetric matrix can be reduced to a diagonal matrix by orthogonal similarity transformations. Jacobi method exploits the symmetry of the matrix and chooses suitable rotations to reduce a symmetric matrix to a diagonal form. Jacobi method is usually much slower than the symmetric QR algorithm. However, Jacobi method remains to be interesting because the method is capable of programming and inherently parallelism recognized in recent years.

8.3.1 Basic idea

Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ be a symmetric matrix and

$$\text{off}(A) \equiv \left(\|A\|_F^2 - \sum_{i=1}^{n} a_{ii}^2 \right)^{1/2} = \left(\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}^2 \right)^{1/2}$$

The idea of Jacobi method is to systematically reduce the off(A) to be zero. The basic tools for doing this are called Jacobi rotations defined as follows,

$$J(p,q,\theta) = I + \sin\theta(e_p e_q^T - e_q e_p^T) + (\cos\theta - 1)(e_p e_p^T + e_q e_q^T),$$

where $p < q$ and e_k denotes the k-th unit vector. Note that Jacobi rotations are no different from Givens rotations, see Section 4.2.2. We change the name in this section to honour the inventor. The basic step in a Jacobi procedure involves:

(1) Choose p and q for a rotation with $1 \le p < q \le n$.

(2) Compute a rotation angle θ such that

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \tag{8.2}$$

is diagonal, i.e., $b_{pq} = b_{qp} = 0$, where $c = \cos\theta$ and $s = \sin\theta$.

(3) Overwrite A with $B = [b_{ij}] = J^T A J$, where $J = J(p,q,\theta)$.

Note that the matrix B agrees with the matrix A except the p-th row (column) and the q-th row (column). The relations are:

$$b_{ip} = b_{pi} = ca_{ip} - sa_{iq}, \quad i \neq p, q$$

$$b_{iq} = b_{qi} = sa_{ip} + ca_{iq}, \quad i \neq p, q$$

$$b_{pp} = c^2 a_{pp} - 2sca_{pq} + s^2 a_{qq},$$

$$b_{qq} = s^2 a_{pp} + 2sca_{pq} + c^2 a_{qq},$$

$$b_{pq} = b_{qp} = (c^2 - s^2)a_{pq} + sc(a_{pp} - a_{qq}).$$

Let us first consider actual computations of $s = \sin\theta$ and $c = \cos\theta$ such that $b_{pq} = b_{qp} = 0$ in (8.2). Actually it is equivalent to

$$a_{pq}(c^2 - s^2) + (a_{pp} - a_{qq})cs = 0. \tag{8.3}$$

If $a_{pq} = 0$, then we just set $c = 1$ and $s = 0$. Otherwise define

$$\tau = \frac{a_{qq} - a_{pp}}{2a_{pq}}, \qquad t = \tan\theta = \frac{s}{c},$$

and conclude from (8.3) that t solves the quadratic equation

$$t^2 + 2\tau t - 1 = 0.$$

Then,

$$t = -\tau \pm \sqrt{1 + \tau^2}.$$

We select t to be the smaller of the two roots which ensures that $|\theta| \leq \pi/4$ and has the effect of minimizing of $\|B - A\|_F^2$ because

$$\|B - A\|_F^2 = 4(1 - c) \sum_{\substack{i=1 \\ i \neq p,q}}^{n} (a_{ip}^2 + a_{iq}^2) + \frac{2a_{pq}^2}{c^2}.$$

After t is determined, we can obtain c and s from the formulas

$$c = \frac{1}{\sqrt{1 + t^2}}, \qquad s = tc.$$

We summarize the computation of Jacobi rotation $J(p, q, \theta)$ as follows. Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$ and indices p, q with $1 \leq p < q \leq n$, the following algorithm computes a cosine-sine pair such that $b_{pq} = b_{qp} = 0$ where b_{jk} is the (j, k)-th entry of the matrix $B = J(p, q, \theta)^T A J(p, q, \theta)$.

Algorithm 8.4

function : $[c, s] = \mathbf{sym}(A, p, q)$
  if $A(p, q) \neq 0$
    $\tau = (A(q, q) - A(p, p))/(2A(p, q))$
    if $\tau \geq 0$
      $t = 1/(\tau + \sqrt{1 + \tau^2})$
    else
      $t = -1/(-\tau + \sqrt{1 + \tau^2})$
    end
    $c = 1/\sqrt{1 + t^2}$
    $s = tc$
  else
    $c = 1$
    $s = 0$
  end

Once $J(p, q, \theta)$ is determined, then the updated

$$A \leftarrow J(p, q, \theta)^T A J(p, q, \theta)$$

can be computed in $6n$ operations.

How can we choose the integers p and q? Since the Frobenius norm is preserved by the orthogonal transformation, we have $\|B\|_F = \|A\|_F$. Note that

$$a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 = b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2 = b_{pp}^2 + b_{qq}^2,$$

and then

$$\begin{aligned} \text{off}(B)^2 &= \|B\|_F^2 - \sum_{i=1}^{n} b_{ii}^2 \\ &= \|A\|_F^2 - \sum_{i=1}^{n} a_{ii}^2 + (a_{pp}^2 + a_{qq}^2 - b_{pp}^2 - b_{qq}^2) \\ &= \text{off}(A)^2 - 2a_{pq}^2. \end{aligned} \tag{8.4}$$

Our goal is to minimize off(B), the best choice of p, q should be

$$|a_{pq}| = \max_{1 \leq i < j \leq n} |a_{ij}|.$$

This is the basic idea of the classical Jacobi method.

8.3.2 Classical Jacobi method

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a tolerance $\epsilon > 0$, the following algorithm overwrites A with U^TAU, where U is orthogonal and $\text{off}(U^TAU) \leq \epsilon\|A\|_F$.

Algorithm 8.5 (Classical Jacobi method)

$$\begin{cases} U = I_n; \quad eps = \epsilon\|A\|_F \\ \textbf{while} \quad \mathit{off}(A) > eps \\ \qquad \mathit{choose} \quad p, q \quad \mathit{such\ that} \quad |a_{pq}| = \max\limits_{i\neq j} |a_{ij}| \\ \qquad [c, s] = \textbf{sym}(A, p, q) \\ \qquad A = J(p, q, \theta)^T A J(p, q, \theta) \\ \qquad U = UJ(p, q, \theta) \\ \textbf{end} \end{cases}$$

Since $|a_{pq}|$ is the largest off-diagonal entry, we have

$$\text{off}(A)^2 \leq N(a_{pq}^2 + a_{qp}^2),$$

where $N = n(n-1)/2$. It follows from (8.4) that

$$\text{off}(B)^2 \leq \left(1 - \frac{1}{N}\right)\text{off}(A)^2.$$

If A_k denotes the matrix A after k-th Jacobi iteration and $A_0 = A$, we have by induction,

$$\text{off}(A_k)^2 \leq \left(1 - \frac{1}{N}\right)^k \text{off}(A_0)^2.$$

This implies that the classical Jacobi method converges linearly. However, actually, the asymptotic convergence rate of the Jacobi method is quadratic. We can prove that for k large enough, there is a constant c such that

$$\text{off}(A_{k+N}) \leq c \cdot \text{off}(A_k)^2,$$

see [19] and references therein. Therefore, the off-diagonal "norm" will approach to zero at a quadratic rate after a sufficient number of iterations.

Another advantage of the Jacobi method is easy to compute the eigenvectors. If the iteration stops after the k-th rotation, we then have

$$A_k = J_k^T J_{k-1}^T \cdots J_1^T A J_1 J_2 \cdots J_k.$$

Denote

$$Q_k = J_1 J_2 \cdots J_k.$$

Thus

$$AQ_k = Q_k A_k.$$

Since off-diagonal entries of A_k are tiny and then diagonal entries of A_k are good approximations to the eigenvalues of A, the identity above shows that the columns of Q_k are good approximations to the eigenvectors of A and all the approximate eigenvectors are orthonormal. We can obtain Q_k, the approximate eigenvectors, during Jacobi iterative process.

8.3.3 Parallel Jacobi method

Jacobi method to the symmetric eigenvalue problem is inherently parallelism. To illustrate this, let $A \in \mathbb{R}^{8\times 8}$ be symmetric. If one has a parallel computer with 4 processors, then one can group the 28 subproblems into 7 groups of rotations as follows:

$$\begin{aligned}
&\text{group (1)}: \quad (1,2),(3,4),(5,6),(7,8);\\
&\text{group (2)}: \quad (1,3),(2,4),(5,7),(6,8);\\
&\text{group (3)}: \quad (1,4),(2,3),(5,8),(6,7);\\
&\text{group (4)}: \quad (1,5),(2,6),(3,7),(4,8);\\
&\text{group (5)}: \quad (1,6),(2,5),(3,8),(4,7);\\
&\text{group (6)}: \quad (1,7),(2,8),(3,5),(4,6);\\
&\text{group (7)}: \quad (1,8),(2,7),(3,6),(4,5).
\end{aligned}$$

Note that all 4 rotations within each group are nonconflicting. For instance, the subproblems $J(2i-1,2i,\theta_i)$, $i=1,2,3,4$, in the first group can be carried out in parallel. When we compute $J(1,2,\theta_1)^T A J(1,2,\theta_1)$, it has no effect on the rotations $(3,4)$, $(5,6)$ and $(7,8)$. Then the computation of

$$A = AJ(1,2,\theta_1), \qquad A = AJ(3,4,\theta_2),$$

$$A = AJ(5,6,\theta_3), \qquad A = AJ(7,8,\theta_4),$$

can be executed in parallel by 4 processors. Similarly, the computation of

$$A = J(1,2,\theta_1)^T A, \qquad A = J(3,4,\theta_2)^T A,$$

$$A = J(5,6,\theta_3)^T A, \qquad A = J(7,8,\theta_4)^T A,$$

can also be carried out in parallel by 4 processors. For the example above, it only needs 1/4 computing time of a computer with a single processor. A parallel Jacobi algorithm can be found in [19].

8.4 Bisection method

In this section we present the bisection method for symmetric tridiagonal eigenvalue problems. Combining the bisection method with the tridiagonal skill, we can obtain a numerical method for a specified eigenvalue and its corresponding eigenvector of a symmetric matrix. For a given tridiagonal matrix

$$T = \begin{bmatrix} a_1 & b_1 & & \mathbf{0} \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ \mathbf{0} & & b_{n-1} & a_n \end{bmatrix} \in \mathbb{R}^{n\times n}, \tag{8.5}$$

we consider the computation of eigenvalues of T. Without loss of generality, we assume that $b_i \neq 0$, $i = 1, 2, \cdots, n-1$, i.e., T is an irreducible symmetric tridiagonal matrix. Otherwise, T can be divided into several smaller irreducible symmetric tridiagonal matrices.

Let $p_i(\lambda)$ be the characteristic polynomial of the i-by-i leading principal submatrix T_i of T, $i = 1, 2, \cdots, n$. Then these polynomials satisfy a three-term recurrence:

$$\begin{aligned} &p_0(\lambda) \equiv 1, \qquad p_1(\lambda) = a_1 - \lambda, \\ &p_i(\lambda) = (a_i - \lambda)p_{i-1}(\lambda) - b_{i-1}^2 p_{i-2}(\lambda), \qquad i = 2, 3, \cdots, n. \end{aligned} \tag{8.6}$$

Since T is symmetric, the roots of polynomial $p_i(\lambda)$ $(i = 1, 2\cdots, n)$ are real. The following interlacing property is very important.

Theorem 8.7 (Sturm Sequence Property) *Let the symmetric tridiagonal matrix T in (8.5) be irreducible. Then the eigenvalues of T_{i-1} strictly separate the eigenvalues of T_i:*

$$\lambda_i(T_i) < \lambda_{i-1}(T_{i-1}) < \lambda_{i-1}(T_i) < \cdots < \lambda_2(T_i) < \lambda_1(T_{i-1}) < \lambda_1(T_i).$$

Moreover, if $s_n(\lambda)$ denotes the number of sign changes in the sequence

$$\{p_0(\lambda), p_1(\lambda), \cdots, p_n(\lambda)\}$$

then $s_n(\lambda)$ is equal to the number of eigenvalues of T that are less than λ, where $p_i(\lambda)$ are defined by (8.6). If $p_i(\lambda) = 0$, then $p_{i-1}(\lambda)p_{i+1}(\lambda) < 0$.

Proof It follows from Theorem 8.4 that the eigenvalues of T_{i-1} weakly separate those of T_i. Next we will show that the separation must be strict.

Assume that $p_i(\mu) = p_{i-1}(\mu) = 0$ for some i and μ. Since T is irreducible and, we note that by (8.6),

$$p_0(\mu) = p_1(\mu) = \cdots = p_i(\mu) = 0,$$

which is a contradiction with $p_0(\lambda) \equiv 1$. Thus we have a strict separation. The assertion about $s_n(\lambda)$ is developed in [46]. □

The bisection method for computing a specified eigenvalue of T can be stated as follows:

Algorithm 8.6 (Bisection algorithm) *Let $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ be the eigenvalues of T, i.e., the roots of $p_n(\lambda)$, and ϵ be a tolerance. Suppose that the desired eigenvalue is λ_m for a given $m \leq n$. Then*

(1) *Find an interval $[l_0, u_0]$ including λ_m. Since $|\lambda_i| \leq \rho(T) \leq \|T\|_\infty$, we can take $l_0 = -\|T\|_\infty$ and $u_0 = \|T\|_\infty$.*

(2) *Compute $r_1 = \dfrac{l_0 + u_0}{2}$ and $s_n(r_1)$.*

(3) *If $s_n(r_1) \geq m$, then $\lambda_m \in [l_0, r_1]$, set $l_1 = l_0$ and $u_1 = r_1$; otherwise $\lambda_m \in [r_1, u_0]$, set $l_1 = r_1$ and $u_1 = u_0$.*

(4) *If $|l_1 - u_1| < \epsilon$, take $r_2 = \dfrac{l_1 + u_1}{2}$ as an approximate value of λ_m. Otherwise go to (2).*

From the algorithm above, we can see that the main operation cost is to compute $s_n(\mu)$. However in practice, $s_n(\mu)$ cannot be obtained through computing the value of $p_i(\mu)$ because it is difficult to evaluate polynomials of high order. In order to avoid such a problem, we define

$$q_i(\lambda) = \frac{p_i(\lambda)}{p_{i-1}(\lambda)}, \qquad i = 1, 2, \cdots, n.$$

From (8.6), we have

$$q_1(\lambda) = p_1(\lambda) = a_1 - \lambda, \qquad q_i(\lambda) = a_i - \lambda - \frac{b_{i-1}^2}{q_{i-1}(\lambda)}, \qquad i = 2, 3, \cdots, n.$$

It is easy to check that $s_n(\mu)$ is exactly the number of negative values in the sequence of $q_1(\mu), \cdots, q_n(\mu)$. The following is a practical algorithm for computing $s_n(\mu)$.

Algorithm 8.7 (Compute sign changes)

$x = [a_1, a_2, \cdots, a_n]$
$y = [0, b_1, \cdots, b_{n-1}]$
$s = 0;\ q = x(1) - \mu$
for $k = 1 : n$
  if $q < 0$
    $s = s + 1$
  end
  if $k < n$
    if $q = 0$
      $q = |y(k+1)|\mathbf{u}$
    end
    $q = x(k+1) - \mu - y(k+1)^2/q$
  end
end

where $\mathbf{u}$ is the machine precision.

When $q_i(\mu) = 0$, we treat q_i as a positive number. Therefore, we can use a small positive number $|b_i|\mathbf{u}$ instead of $q_i(\mu)$ in Algorithm 8.7. If we store b_i^2 in advance, then Algorithm 8.7 needs $3n$ operations. If an eigenvalue is computed by using the bisection method m times on average, then the operation cost is $3nm$. Thus, it is efficient to compute the eigenvalues of any symmetric tridiagonal matrix by the bisection method. On the other hand, the rounding error analysis shows that the bisection method is numerically stable, see [46].

8.5 Divide-and-conquer method

A divide-and-conquer method is a numerical method developed by Dongarra and Sorensen in 1987 for computing all the eigenvalues and eigenvectors of symmetric tridiagonal matrices, see [19]. The basic idea is to tear the original symmetric tridiagonal matrix into 2^k symmetric tridiagonal matrices with smaller sizes and then compute the spectral decomposition of each smaller symmetric tridiagonal matrix. Once we obtained these smaller spectral decompositions, we then combine them together to form a spectral decomposition of the original matrix. Thus, this method is suitable for parallel computing.

8.5.1 Tearing

Let $T \in \mathbb{R}^{n\times n}$ be given as follows,

$$T = \begin{bmatrix} a_1 & b_1 & & \mathbf{0} \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ \mathbf{0} & & b_{n-1} & a_n \end{bmatrix}$$

Without loss of generality, assume $n = 2m$. Let

$$v = (\underbrace{0,\cdots,0}_{m-1},1,\theta,\underbrace{0,\cdots,0}_{m-1})^T \in \mathbb{R}^n.$$

Consider the matrix

$$\tilde{T} = T - \rho v v^T$$

where θ, $\rho \in \mathbb{R}$ are needed to be determined. It is easy to see that $\tilde{T}$ is identical to T except its 4 middle entries:

$$\begin{bmatrix} a_m - \rho & b_m - \rho\theta \\ b_m - \rho\theta & a_{m+1} - \rho\theta^2 \end{bmatrix}.$$

If we set $\rho\theta = b_m$, then

$$T = \begin{bmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{bmatrix} + \rho v v^T,$$

where

$$T_1 = \begin{bmatrix} a_1 & b_1 & & & \mathbf{0} \\ b_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{m-2} & a_{m-1} & b_{m-1} \\ \mathbf{0} & & & b_{m-1} & \tilde{a}_m \end{bmatrix}$$

and

$$T_2 = \begin{bmatrix} \tilde{a}_{m+1} & b_{m+1} & & & \mathbf{0} \\ b_{m+1} & a_{m+2} & b_{m+2} & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & b_{n-1} \\ \mathbf{0} & & & b_{n-1} & a_n \end{bmatrix}$$

with

$$\tilde{a}_m = a_m - \rho, \qquad \tilde{a}_{m+1} = a_{m+1} - \rho\theta^2.$$

Therefore, T is divided into a sum of a partitioned matrix and a rank-one matrix. If we divide T_1 and T_2 repeatedly, then finally we can divide T into 2^k blocks.

8.5.2 Combining

Once we obtained the spectral decompositions of T_1 and T_2:

$$Q_1^T T_1 Q_1 = D_1, \qquad Q_2^T T_2 Q_2 = D_2,$$

where Q_1, $Q_2 \in \mathbb{R}^{m\times m}$ are orthogonal matrices, and D_1, $D_2 \in \mathbb{R}^{m\times m}$ are diagonal matrices, our aim now is to compute the spectral decomposition of T from that of T_1 and T_2, i.e., to find an orthogonal matrix V such that

$$V^T T V = \text{diag}(\lambda_1, \cdots, \lambda_n).$$

Let

$$U = \begin{bmatrix} Q_1 & \mathbf{0} \\ \mathbf{0} & Q_2 \end{bmatrix},$$

then

$$\begin{aligned} U^T T U &= \begin{bmatrix} Q_1 & \mathbf{0} \\ \mathbf{0} & Q_2 \end{bmatrix}^T \left(\begin{bmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{bmatrix} + \rho v v^T \right) \begin{bmatrix} Q_1 & \mathbf{0} \\ \mathbf{0} & Q_2 \end{bmatrix} \\ &= D + \rho z z^T, \end{aligned}$$

where

$$D = \text{diag}(D_1, D_2), \qquad z = U^T v.$$

Now the problem of finding the spectral decomposition of T is reduced to the problem of computing the spectral decomposition of $D + \rho z z^T$. We will consider how to compute the spectral decomposition of $D + \rho z z^T$ quickly and stably.

Lemma 8.1 *Let $D = \text{diag}(d_1, \cdots, d_n) \in \mathbb{R}^{n\times n}$ with $d_1 > d_2 > \cdots > d_n$. Assume that $0 \neq \rho \in \mathbb{R}$ and $z = (z_1, z_2, \cdots, z_n)^T \in \mathbb{R}^n$ with $z_i \neq 0$ for all i. Let $u \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ satisfy*

$$(D + \rho z z^T)u = \lambda u, \qquad u \neq 0.$$

Then $z^T u \neq 0$ and $D - \lambda I$ is nonsingular.

Proof If $z^T u = 0$, then $Du = \lambda u$ with $u \neq 0$, i.e., λ is the eigenvalue of D and u is the eigenvector associated with λ. Since D is a diagonal matrix with distinct entries, there must exist some i such that $d_i = \lambda$ and $u = \alpha e_i$ with $\alpha \neq 0$, where e_i is the i-th unit vector. Thus

$$0 = z^T u = \alpha z^T e_i = \alpha z_i,$$

which implies $z_i = 0$, a contradiction. Therefore, $z^T u \neq 0$.

On the other hand, if $D - \lambda I$ is singular, then there exists some i such that $e_i^T(D - \lambda I) = 0$, and then

$$0 = e_i^T(D - \lambda I)u = -\rho z^T u e_i^T z.$$

Since $\rho z^T u \neq 0$, we have $e_i^T z = z_i = 0$, a contradiction. Thus, $D - \lambda I$ is nonsingular. □

Theorem 8.8 *Let $D = \mathrm{diag}(d_1, \cdots, d_n) \in \mathbb{R}^{n\times n}$ with $d_1 > d_2 > \cdots > d_n$. Assume that $0 \neq \rho \in \mathbb{R}$ and $z = (z_1, z_2, \cdots, z_n)^T \in \mathbb{R}^n$ with $z_i \neq 0$ for all i. Suppose that the spectral decomposition of $D + \rho z z^T$ is*

$$V^T(D + \rho z z^T)V = \mathrm{diag}(\lambda_1, \cdots, \lambda_n),$$

where $V = [v_1, \cdots, v_n]$ is an orthogonal matrix and $\lambda_1 \geq \cdots \geq \lambda_n$. Then

(i) *$\lambda_1, \cdots, \lambda_n$ are n roots of the function*

$$f(\lambda) = 1 + \rho z^T(D - \lambda I)^{-1} z.$$

(ii) *If $\rho > 0$, then*

$$\lambda_1 > d_1 > \lambda_2 > \cdots > \lambda_n > d_n;$$

if $\rho < 0$, then

$$d_1 > \lambda_1 > d_2 > \cdots > d_n > \lambda_n.$$

(iii) *There exists a constant $\alpha_i \neq 0$ such that*

$$v_i = \alpha_i(D - \lambda_i I)^{-1} z, \qquad i = 1, 2, \cdots, n.$$

Proof From the assumption, we have

$$(D + \rho z z^T)v_i = \lambda_i v_i, \qquad \|v_i\|_2 = 1.$$

It follows from Lemma 8.1 that $D - \lambda_i I$ is nonsingular. Thus,

$$v_i = -\rho z^T v_i (D - \lambda_i I)^{-1} z, \qquad i = 1, 2, \cdots, n, \tag{8.7}$$

thereby establishing (iii). Note that $D + \rho z z^T$ has distinct eigenvalues. Otherwise, if $\lambda_i = \lambda_j$, then v_i and v_j are linearly dependent, which contradicts with the orthogonality between v_i and v_j.

By multiplying z^T to the both sides of (8.7) and noting that $z^T v_i \neq 0$, we have

$$1 = -\rho z^T (D - \lambda_i I)^{-1} z,$$

i.e.,

$$f(\lambda_i) = 0, \qquad i = 1, 2, \cdots, n.$$

Thus, λ_i, $i = 1, 2, \cdots, n$, are the roots of $f(\lambda)$. Next we prove that $f(\lambda)$ has exactly n zeros. Note that

$$f(\lambda) = 1 + \rho \left(\frac{z_1^2}{d_1 - \lambda} + \cdots + \frac{z_n^2}{d_n - \lambda} \right),$$

and moreover,

$$f'(\lambda) = \rho \left(\frac{z_1^2}{(d_1 - \lambda)^2} + \cdots + \frac{z_n^2}{(d_n - \lambda)^2} \right).$$

Thus, $f(\lambda)$ is strictly monotone between the poles d_i and d_{i+1}. If $\rho > 0$, $f(\lambda)$ is strictly increasing; if $\rho < 0$, $f(\lambda)$ is strictly decreasing. Therefore, it is easy to see that $f(\lambda)$ has exactly n roots, one of each in the intervals

$$(d_n, d_{n-1}), \cdots, \ (d_2, d_1), \ (d_1, \infty),$$

if $\rho > 0$; and one of each in the intervals

$$(-\infty, d_n), \ (d_n, d_{n-1}), \cdots, \ (d_2, d_1),$$

if $\rho < 0$. Thus, (i) and (ii) are established. □

By Theorem 8.8, we can compute the spectral decomposition of $D + \rho z z^T$ efficiently in the following two steps:

(1) Find roots $\lambda_1, \lambda_2, \cdots, \lambda_n$ of $f(\lambda)$. There is a unique root of $f(\lambda)$ in each of the intervals (d_{i+1}, d_i) and $f(\lambda)$ is strictly monotone in the interval. Thus, this step can be implemented quickly and stably by using a Newton-like method [14].

(2) Compute

$$v_i = \frac{(D - \lambda_i I)^{-1} z}{\|(D - \lambda_i I)^{-1} z\|_2}, \qquad i = 1, 2, \cdots, n.$$

The spectral decomposition of a general $D + \rho z z^T$ can be turned into the case as in Theorem 8.8. To this end, we can prove the following theorem constructively.

Theorem 8.9 *Let $D = \mathrm{diag}(d_1, \cdots, d_n) \in \mathbb{R}^{n\times n}$ and $z \in \mathbb{R}^n$. Then there exists an orthogonal matrix V and a permutation π of $\{1, 2, \cdots, n\}$ such that*

(i) $V^T z = (\eta_1, \cdots, \eta_r, 0, \cdots, 0)^T$ *where* $\eta_i \neq 0$, *for* $i = 1, 2, \cdots, r$.

(ii) $V^T DV = \mathrm{diag}(d_{\pi(1)}, \cdots, d_{\pi(n)})$ *where* $d_{\pi(1)} > d_{\pi(2)} > \cdots > d_{\pi(r)}$.

Proof Suppose that two indices $i < j$ satisfy $d_i = d_j$. Then we can set a rotation $P_{ij} = G(i, j, \theta)$ such that the j-th component of $P_{ij}z$ is zero. It is easy to show that $P_{ij}^T D P_{ij} = D$. After several steps, we can find an orthogonal matrix V_1 which is a product of some rotations such that

$$V_1^T D V_1 = D, \qquad V_1^T z = (\xi_1, \cdots, \xi_n)^T$$

with the property that if $\xi_i \xi_j \neq 0$ $(i \neq j)$, then $d_i \neq d_j$.

If $\xi_i = 0$, $\xi_j \neq 0$ for $i < j$, then we can find a permutation matrix to interchange the columns i and j. In such a way, we can find a permutation matrix P_1 which permutes all the nonzero ξ_j to the front, i.e.,

$$P_1^T V_1^T z = (\xi_{\pi_1(1)}, \cdots, \xi_{\pi_1(n)})^T$$

with

$$\xi_{\pi_1(i)} \neq 0, \qquad i = 1, 2, \cdots, r,$$

and

$$\xi_{\pi_1(i)} = 0, \qquad i = r+1, \cdots, n.$$

Here π_1 is a permutation of $\{1, 2, \cdots, n\}$. It follows from the construction of P_1 that

$$P_1^T V_1^T D V_1 P_1 = P_1^T D P_1 = \mathrm{diag}(d_{\pi_1(1)}, \cdots, d_{\pi_1(n)}),$$

where the first r diagonal entries $d_{\pi_1(1)}, \cdots, d_{\pi_1(r)}$ are distinct.

Finally, we can find another permutation matrix P_2 of order r such that

$$P_2^T \mathrm{diag}(d_{\pi_1(1)}, \cdots, d_{\pi_1(r)}) P_2 = \mathrm{diag}(\mu_1, \cdots, \mu_r)$$

where $\mu_1 > \mu_2 > \cdots > \mu_r$. Let

$$V = V_1 P_1 \mathrm{diag}(P_2, I_{n-r})$$

and π be a permutation of $\{1, 2, \cdots, n\}$ determined by P_1 and P_2. Then,

$$V^T z = (\xi_{\pi(1)}, \cdots, \xi_{\pi(r)}, 0, \cdots, 0)^T = (\eta_1, \cdots, \eta_r, 0, \cdots, 0)^T,$$

where $\eta_i \neq 0$ for $i = 1, 2, \cdots, r$, and

$$V^T D V = \mathrm{diag}(d_{\pi(1)}, \cdots, d_{\pi(n)})$$

with

$$d_{\pi(1)} > d_{\pi(2)} > \cdots > d_{\pi(r)}.$$

The proof is complete. □

For any $D = \text{diag}(d_1, \cdots, d_n) \in \mathbb{R}^{n\times n}$ and $z \in \mathbb{R}^n$, by Theorem 8.9, we can construct an orthogonal matrix V such that

$$V^T(D + \rho zz^T)V = \begin{bmatrix} D_1 + \rho\omega\omega^T & \mathbf{0} \\ \mathbf{0} & D_2 \end{bmatrix},$$

where

$$D_1 = \text{diag}(d_{\pi(1)}, \cdots, d_{\pi(r)}) \in \mathbb{R}^{r\times r}, \qquad d_{\pi(1)} > \cdots > d_{\pi(r)};$$

$$D_2 = \text{diag}(d_{\pi(r+1)}, \cdots, d_{\pi(n)}) \in \mathbb{R}^{(n-r)\times(n-r)};$$

and

$$\omega = (\eta_1, \cdots, \eta_r)^T, \quad \eta_i \neq 0, \quad i = 1, 2, \cdots, r.$$

Then we only need to compute the spectral decomposition of $D_1 + \rho\omega\omega^T$ instead of $D + \rho zz^T$.

Finally, we briefly introduce the parallel computation of the divide-and-conquer method. For simplicity, we use a parallel computer with 4 processors to compute the spectral decomposition of a $4N$-by-$4N$ symmetric tridiagonal matrix T. It can be summarized by the following 4 steps:

(1) Tear

$$T = \begin{bmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{bmatrix} + \rho vv^T, \qquad T_1 \in \mathbb{R}^{2N\times 2N}, \qquad v \in \mathbb{R}^{4N};$$

and

$$T_i = \begin{bmatrix} T_{i1} & \mathbf{0} \\ \mathbf{0} & T_{i2} \end{bmatrix} + \rho_i\omega_i\omega_i^T,$$

where $T_{ij} \in \mathbb{R}^{N\times N}$ and $\omega_i \in \mathbb{R}^{2N}$, for $i = 1, 2$.

(2) Compute the spectral decompositions of T_{11}, T_{12}, T_{21} and T_{22} by 4 processors in parallel.

(3) Combine the spectral decompositions of T_{11}, T_{12} to form a spectral decomposition of T_1, and combine the spectral decompositions of T_{21}, T_{22} to form a spectral decomposition of T_2. These can be implemented by 4 processors at the same time.

(4) Combine the spectral decompositions of T_1 and T_2 to from a spectral decomposition of T. This can be derived by 4 processors in parallel.

From discussions above, we know that the divide-and-conquer method can be used for computing all the eigenvalues and eigenvectors of any large symmetric tridiagonal matrix in parallel.

Exercises:

1. Compute the Schur decomposition of

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}.$$

2. Show that if $X \in \mathbb{R}^{n\times r}$ with $r \leq n$, and $\|X^T X - I\|_2 = \tau < 1$, then

$$\sigma_{\min}(X) \geq 1 - \tau,$$

where $\sigma_{\min}$ denotes the smallest singular value.

3. Show that $A = B + \mathbf{i}C$ is Hermitian if and only if

$$M = \begin{bmatrix} B & -C \\ C & B \end{bmatrix}$$

is symmetric. Relate the eigenvalues and eigenvectors of A to those of M.

4. Relate the singular values and the singular vectors of $A = B + \mathbf{i}C$ to those of

$$\begin{bmatrix} B & -C \\ C & B \end{bmatrix},$$

where B, $C \in \mathbb{R}^{m\times n}$.

5. Use the singular value decomposition to show that if $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, then there exist a matrix $Q \in \mathbb{R}^{m\times n}$ with $Q^T Q = I$ and a positive semi-definite matrix $P \in \mathbb{R}^{n\times n}$ such that $A = QP$.

6. Let

$$A = \begin{bmatrix} I & B \\ B^* & I \end{bmatrix}$$

with $\|B\|_2 < 1$. Show that

$$\|A\|_2 \|A^{-1}\|_2 = \frac{1 + \|B\|_2}{1 - \|B\|_2}.$$

7. Let

$$A = \begin{bmatrix} a_1 & b_1 & & & \mathbf{0} \\ c_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ \mathbf{0} & & & c_{n-1} & a_n \end{bmatrix},$$

where $b_i c_i > 0$. Then there exists a diagonal D such that $D^{-1}AD$ is a symmetric tridiagonal matrix.

8. Let

$$T = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix}.$$

(1) Is T positive definite?

(2) How many eigenvalues of T lie in the interval $[0, 2]$?

9. Let $A, E \in \mathbb{R}^{n\times n}$ be two symmetric matrices. Show that if A is positive definite and $\|A^{-1}\|_2\|E\|_2 < 1$, then $A + E$ is also positive definite.

10. Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, and assume that the singular values of A are ordered as

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n.$$

Show that

$$\sigma_i = \max_{dim(S)=i} \min_{0\neq u\in S} \frac{\|Au\|_2}{\|u\|_2} = \min_{dim(S)=n-i+1} \max_{0\neq u\in S} \frac{\|Au\|_2}{\|u\|_2},$$

where S is any subspace of $\mathbb{R}^n$.

11. Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ be symmetric and satisfy

(1) $a_{ii} > 0, \quad i = 1, 2, \cdots, n,$

(2) $a_{ij} \leq 0, \quad i \neq j,$

(3) $\sum_{i=1}^{n} a_{i1} > 0,$

(4) $\sum_{i=1}^{n} a_{ij} = 0, \quad j = 2, 3, \cdots, n.$

Prove that the eigenvalues of A are nonnegative.

12. Let A be symmetric and have bandwidth p. Show that if we perform the shifted QR iteration $A - \mu I = QR$ and $\tilde{A} = RQ + \mu I$, then $\tilde{A}$ has bandwidth p.

Chapter 9
Applications

In this chapter, we will briefly survey some of the latest developments in using boundary value methods (BVMs) for solving initial value problems of systems of ordinary differential equations (ODEs). These methods require the solution of one or more nonsymmetric, large and sparse linear systems. Therefore, we will use the GMRES method studied 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 BVM is used for an n-by-n 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 $2n(\nu_1 + \nu_2)$. When the GMRES method is applied to the preconditioned systems, the method will converge in at most $2n(\nu_1 + \nu_2) + 1$ iterations. Applications to different kinds of delay differential equations (DDEs) are also given. For a literature on BVMs for ODEs and DDEs, we refer to [3, 4, 5, 8, 10, 23, 24, 25, 26, 29, 30].

9.1 Introduction

Let us begin with the initial value problem:

$$\begin{cases} \mathbf{y}'(t) = J_n\mathbf{y}(t) + \mathbf{g}(t), \quad t \in (t_0, T], \\ \\ \mathbf{y}(t_0) = \mathbf{z}, \end{cases} \tag{9.1}$$

where $\mathbf{y}(t)$, $\mathbf{g}(t) : \mathbb{R} \to \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^n$, and $J_n \in \mathbb{R}^{n\times n}$. The initial value methods (IVMs), such as the Runge-Kutta methods, are well-known methods for solving (9.1), see [40]. Recently, another class of methods called the boundary value methods (BVMs) has been proposed in [5]. Using BVMs to discretize (9.1), we obtain a linear system

$$M\mathbf{u} = \mathbf{b}.$$

The advantage of using BVMs is that the methods are more stable and the resulting linear system $M\mathbf{u} = \mathbf{b}$ is hence more well-conditioned. However, this system is in general large and sparse (with band-structure), and solving it

is a major problem in the application of BVMs. The GMRES method studied in Chapter 6 will be used for solving $M\mathbf{u} = \mathbf{b}$. In order to speed up the convergence of the GMRES iterations, a preconditioner S called the Strang-type block-circulant preconditioner [10] is used to precondition the discrete system. The advantage of the Strang-type preconditioner is that if an A_{ν_1,ν_2}-stable BVM is used for solving (9.1), then S is invertible and the preconditioned matrix can be decomposed as

$$S^{-1}M = I + L,$$

where the rank of L is at most $2n(\nu_1 + \nu_2)$ which is independent of the integration step size. It follows that the GMRES method applied to the preconditioned system will converge in at most $2n(\nu_1 + \nu_2) + 1$ iterations in exact arithmetic.

The outline of this chapter is as follows. In Section 9.2, we will give some background knowledge about the linear multistep formulas (LMFs) and BVMs. Then, we will investigate the properties of the Strang-type block-circulant preconditioner for ODEs in Section 9.3. The convergence and cost analysis of the method will also be given with a numerical example. Finally, we discuss the applications of the Strang-type preconditioner with BVMs for solving different kinds of delay differential equations (DDEs) in Sections 9.4–9.6.

9.2 Background of BVMs

We give some background knowledge on LMFs and BVMs in this section.

9.2.1 Linear multistep formulas

Consider an initial value problem

$$\begin{cases} y' = f(t, y), \quad t \in (t_0, T], \\ y(t_0) = y_0, \end{cases}$$

where $y(t) : \mathbb{R} \to \mathbb{R}$ and $f(t, y) : \mathbb{R}^2 \to \mathbb{R}$. The μ-step linear multistep formula (LMF) over a uniform mesh with step size h is defined as follows:

$$\sum_{j=0}^{\mu} \alpha_j y_{m+j} = h \sum_{j=0}^{\mu} \beta_j f_{m+j}, \qquad m = 0, 1, \cdots, \tag{9.2}$$

where y_m is the discrete approximation to $y(t_m)$ and f_m denotes $f(t_m, y_m)$.

To get the solution of (9.2), we need μ initial conditions

$$y_0, y_1, \cdots, y_{\mu-1}.$$

Since only y_0 is provided from the original problem, we have to find additional conditions for the remaining values

$$y_1, y_2, \cdots, y_{\mu-1}.$$

The equation (9.2) with $\mu - 1$ additional conditions is called initial value methods (IVMs). An IVM is called implicit if $\beta_\mu \neq 0$ and explicit if $\beta_\mu = 0$. If an IVM is applied to an initial value problem on the interval $[t_0, t_{N+\mu-1}]$, we have the following discrete problem

$$A_N \mathbf{y} = hB_N \mathbf{f} + \begin{pmatrix} \sum_{i=0}^{\mu-1} (\alpha_i y_i - h\beta_i f_i) \\ \vdots \\ \alpha_0 y_{\mu-1} - h\beta_0 f_{\mu-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{9.3}$$

where

$$\mathbf{y} = (y_\mu, y_{\mu+1}, \cdots, y_{N+\mu-1})^T, \qquad \mathbf{f} = (f_\mu, f_{\mu+1}, \cdots, f_{N+\mu-1})^T,$$

$$A_N = \begin{bmatrix} \alpha_\mu & & & & \\ \vdots & \ddots & & & \\ \alpha_0 & & \ddots & & \\ & \ddots & & \ddots & \\ & & \alpha_0 & \cdots & \alpha_\mu \end{bmatrix}, \quad B_N = \begin{bmatrix} \beta_\mu & & & & \\ \vdots & \ddots & & & \\ \beta_0 & & \ddots & & \\ & \ddots & & \ddots & \\ & & \beta_0 & \cdots & \beta_\mu \end{bmatrix}.$$

Note that the matrices A_N, $B_N \in \mathbb{R}^{N\times N}$ are lower triangular band Toeplitz matrices with lower bandwidth μ. We recall that a matrix is said to be Toeplitz if its entries are constant along its diagonals. Moreover, the linear system (9.3) can be solved easily by forward recursion. A classical example of IVM is the second order backward differentiation formula (BDF),

$$3y_{m+2} - 4y_{m+1} + y_m = 2hf_{m+1},$$

which is a two-step method with $\alpha_0 = 1$, $\alpha_1 = -4$, $\alpha_2 = 3$ and $\beta_1 = 2$.

Instead of using an IVM with μ initial conditions for solving (9.1), we can also use the so-called boundary value methods (BVMs). Given ν_1, $\nu_2 \geq 0$ such that $\nu_1 + \nu_2 = \mu$, then the corresponding BVM requires ν_1 initial additional conditions

$$y_0, y_1, \cdots, y_{\nu_1 - 1},$$

and ν_2 final additional conditions

$$y_N, y_{N+1}, \cdots, y_{N+\nu_2 - 1},$$

which are called (ν_1, ν_2)-boundary conditions. Note that the class of BVMs contains the class of IVMs (i.e., $\nu_1 = \mu$, $\nu_2 = 0$).

The discrete problem generated by a μ-step BVM with (ν_1, ν_2)-boundary conditions can be written in the following matrix form

$$A\mathbf{y} = hB\mathbf{f} + \begin{pmatrix} \sum_{i=0}^{\nu_1 - 1} (\alpha_i y_i - h\beta_i f_i) \\ \vdots \\ \alpha_0 y_{\nu_1 - 1} - h\beta_0 f_{\nu_1 - 1} \\ 0 \\ \vdots \\ 0 \\ \alpha_\mu y_N - h\beta_\mu f_N \\ \vdots \\ \sum_{i=1}^{\nu_2} (\alpha_{\nu_1 + i} y_{N-1+i} - h\beta_{\nu_1 + i} f_{N-1+i}) \end{pmatrix}$$

where

$$\mathbf{y} = (y_{\nu_1}, y_{\nu_1 + 1}, \cdots, y_{N-1})^T, \qquad \mathbf{f} = (f_{\nu_1}, f_{\nu_1 + 1}, \cdots, f_{N-1})^T,$$

A and $B \in \mathbb{R}^{(N-\nu_1)\times(N-\nu_1)}$ are defined as follows,

$$A = \begin{bmatrix} \alpha_{\nu_1} & \cdots & \alpha_\mu & & \\ \vdots & \ddots & \ddots & \ddots & \\ \alpha_0 & \ddots & \ddots & \ddots & \alpha_\mu \\ & \ddots & & \ddots & \vdots \\ & & \alpha_0 & \cdots & \alpha_{\nu_1} \end{bmatrix}, \quad B = \begin{bmatrix} \beta_{\nu_1} & \cdots & \beta_\mu & & \\ \vdots & \ddots & \ddots & \ddots & \\ \beta_0 & \ddots & \ddots & \ddots & \beta_\mu \\ & \ddots & & \ddots & \vdots \\ & & \beta_0 & \cdots & \beta_{\nu_1} \end{bmatrix}. \tag{9.4}$$

Note that the coefficient matrices are band Toeplitz with lower bandwidth ν_1 and upper bandwidth ν_2. An example of BVMs is the third order generalized backward differentiation formula (GBDF),

$$2y_{m+1} + 3y_m - 6y_{m-1} + y_{m-2} = 6hf_m,$$

which is a three-step method with (2, 1)-boundary conditions where

$$\alpha_0 = 1, \quad \alpha_1 = -6, \quad \alpha_2 = 3, \quad \alpha_3 = 2, \quad \beta_2 = 6.$$

Although IVMs are more efficient than BVMs (which cannot be solved by forward recursion), the advantage in using BVMs over IVMs comes from their stability properties. For example, the usual BDF are not A-stable for $\mu > 2$ but the GBDF are A_{ν_1,ν_2}-stable for any $\mu \geq 1$, see for instance [1] and [5, p. 79 and Figures 5.1–5.3].

9.2.2 Block-BVMs and their matrix forms

Let $\mu = \nu_1 + \nu_2$. By using the μ-step block-BVM based on LMF over a uniform mesh $h = (T - t_0)/s$ for solving (9.1), we have:

$$\sum_{i=-\nu_1}^{\nu_2} \alpha_{i+\nu_1} \mathbf{y}_{m+i} = h \sum_{i=-\nu_1}^{\nu_2} \beta_{i+\nu_1} \mathbf{f}_{m+i}, \quad m = \nu_1, \ldots, s - \nu_2. \tag{9.5}$$

Here, $\mathbf{y}_m$ is the discrete approximation to $\mathbf{y}(t_m)$,

$$\mathbf{f}_m = J_n \mathbf{y}_m + \mathbf{g}_m, \qquad \mathbf{g}_m = \mathbf{g}(i_m).$$

Also, (9.5) requires ν_1 initial conditions and ν_2 final conditions which are provided by the following $\mu - 1$ additional equations:

$$\sum_{i=0}^{\mu} \alpha_i^{(j)} \mathbf{y}_i = h \sum_{i=0}^{\mu} \beta_i^{(j)} \mathbf{f}_i, \quad j = 1, \ldots, \nu_1 - 1, \tag{9.6}$$

and

$$\sum_{i=0}^{\mu} \alpha_{\mu-i}^{(j)} \mathbf{y}_{s-i} = h \sum_{i=0}^{\mu} \beta_{\mu-i}^{(j)} \mathbf{f}_{s-i}, \quad j = s - \nu_2 + 1, \ldots, s. \tag{9.7}$$

The coefficients $\{\alpha^{(j)}\}$, $\{\beta^{(j)}\}$ in (9.6) and (9.7) should be chosen such that the truncation errors for these initial and final conditions are of the same order as that in (9.5). By combining (9.5), (9.6), (9.7) and the initial condition $\mathbf{y}(t_0) = \mathbf{y}_0 = \mathbf{z}$, the discrete system of (9.1) is given by the following block form

$$M\mathbf{y} \equiv (\widetilde{A} \otimes I_n - h\widetilde{B} \otimes J_n)\mathbf{y} = \mathbf{e}_1 \otimes \mathbf{z} + h(\widetilde{B} \otimes I_n)\mathbf{g}. \tag{9.8}$$

Here

$$\mathbf{e}_1 = (1, 0, \cdots, 0)^T \in \mathbb{R}^{s+1}, \qquad \mathbf{y} = (\mathbf{y}_0^T, \cdots, \mathbf{y}_s^T)^T \in \mathbb{R}^{(s+1)n},$$

$$\mathbf{g} = (\mathbf{g}_0^T, \cdots, \mathbf{g}_s^T)^T \in \mathbb{R}^{(s+1)n},$$

and $\widetilde{A}$, $\widetilde{B} \in \mathbb{R}^{(s+1)\times(s+1)}$ given by:

$$\widetilde{A} = \begin{bmatrix} 1 & \cdots & 0 & & & & \\ \alpha_0^{(1)} & \cdots & \alpha_\mu^{(1)} & & & & \\ \vdots & \vdots & \vdots & & & \mathbf{0} & \\ \alpha_0^{(\nu_1-1)} & \cdots & \alpha_\mu^{(\nu_1-1)} & & & & \\ \alpha_0 & \cdots & \alpha_\mu & & & & \\ & \alpha_0 & \cdots & \alpha_\mu & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \alpha_0 & \cdots & \alpha_\mu \\ & & \mathbf{0} & & \alpha_0^{(s-\nu_2+1)} & \cdots & \alpha_\mu^{(s-\nu_2+1)} \\ & & & & \vdots & & \vdots \\ & & & & \alpha_0^{(s)} & \cdots & \alpha_\mu^{(s)} \end{bmatrix},$$

$$\widetilde{B} = \begin{bmatrix} 0 & \cdots & 0 & & & & \\ \beta_0^{(1)} & \cdots & \beta_\mu^{(1)} & & & & \\ \vdots & \vdots & \vdots & & & \mathbf{0} & \\ \beta_0^{(\nu_1-1)} & \cdots & \beta_\mu^{(\nu_1-1)} & & & & \\ \beta_0 & \cdots & \beta_\mu & & & & \\ & \beta_0 & \cdots & \beta_\mu & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \beta_0 & \cdots & \beta_\mu \\ & & \mathbf{0} & & \beta_0^{(s-\nu_2+1)} & \cdots & \beta_\mu^{(s-\nu_2+1)} \\ & & & & \vdots & & \vdots \\ & & & & \beta_0^{(s)} & \cdots & \beta_\mu^{(s)} \end{bmatrix}.$$

We remark that usually the linear system (9.8) is large and sparse (with band-structure), and solving it is a major problem in the application of the BVMs. We will use the GMRES method in Chapter 6 for solving (9.8). In order to speed up the convergence rate of the GMRES iterations, we will use a preconidtioner S called the Strang-type block-circulant preconditioner.

9.3 Strang-type preconditioner for ODEs

Now, we construct the Strang-type block-circulant preconditioner S for solving (9.8). We will show that the main advantages of the Strang-type preconditioner are:

(1) S is invertible if an A_{ν_1,ν_2}-stable BVM is used.

(2) The spectrum of the preconditioned system is clustered.

(3) The operation cost for each iteration of the preconditioned GMRES method is smaller than that of direct solvers.

9.3.1 Construction of preconditioner

We first recall the definition of Strang's circulant preconditioner for Toeplitz matrices. Given any Toeplitz matrix

$$T_l = [t_{i-j}]_{i,j=1}^{l} = [t_q],$$

Strang's preconditioner $s(T_l)$ is a circulant matrix with diagonals given by

$$[s(T_l)]_q = \begin{cases} t_q, & 0 \le q \le \lfloor l/2 \rfloor, \\ t_{q-l}, & \lfloor l/2 \rfloor < q < l, \\ [s(T_l)]_{l+q}, & 0 < -q < l, \end{cases}$$

see [9, 22, 37].

Neglecting the perturbations of $\widetilde{A}$ and $\widetilde{B}$, we propose the following preconditioner $S \in \mathbb{R}^{(s+1)n\times(s+1)n}$ called the Strang-type block circulant preconditioner for M given in (9.8):

$$S = s(A) \otimes I_n - hs(B) \otimes J_n, \tag{9.9}$$

where

$$s(A) = \begin{bmatrix} \alpha_{\nu_1} & \cdots & \alpha_\mu & & & \alpha_0 & \cdots & \alpha_{\nu_1-1} \\ \vdots & \ddots & & \ddots & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & \alpha_0 \\ & \ddots & & \ddots & & \ddots & \mathbf{0} & \\ & & \ddots & & \ddots & & \ddots & \\ & \mathbf{0} & & \ddots & & \ddots & & \\ \alpha_\mu & & & & \ddots & & \ddots & \alpha_\mu \\ \vdots & \ddots & & & & \ddots & & \vdots \\ \alpha_{\nu_1+1} & \cdots & \alpha_\mu & & & \alpha_0 & \cdots & \alpha_{\nu_1} \end{bmatrix}$$

and $s(B)$ is defined similarly by using $\{\beta_i\}_{i=0}^{\mu}$ instead of $\{\alpha_i\}_{i=0}^{\mu}$ in $s(A)$. The $\{\alpha_i\}_{i=0}^{\mu}$ and $\{\beta_i\}_{i=0}^{\mu}$ here are the coefficients given in (9.5). We remark that

actually $s(A)$, $s(B)$ are just Strang's circulant preconditioners for Toeplitz matrices A, B respectively, where A, B are given by (9.4).

We will show that the preconditioner S is invertible provided that the given BVM is A_{ν_1,ν_2}-stable and the eigenvalues of J_n are in

$$\mathbb{C}^- \equiv \{q \in \mathbb{C} : \ \mathrm{Re}(q) < 0\}$$

where $\mathrm{Re}(\cdot)$ denotes the real part of a complex number. The stability of a BVM is closely related to two characteristic polynomials of degree $\mu = \nu_1 + \nu_2$, defined as follows:

$$\rho(z) \equiv \sum_{j=-\nu_1}^{\nu_2} \alpha_{j+\nu_1} z^{j+\nu_1} \quad \text{and} \quad \sigma(z) \equiv \sum_{j=-\nu_1}^{\nu_2} \beta_{j+\nu_1} z^{j+\nu_1}. \tag{9.10}$$

The A_{ν_1,ν_2}-stability polynomial is defined by

$$\pi(z,q) \equiv \rho(z) - q\sigma(z) \tag{9.11}$$

where z, $q \in \mathbb{C}$.

Consider now the equation $\pi(z,q) = 0$. It defines a mapping between the complex z-plane and the complex q-plane. For every $z \in \mathbb{C}$ which is a root of $\pi(z,q)$, (9.11) provides

$$q = q(z) = \frac{\rho(z)}{\sigma(z)}.$$

Let

$$\Gamma \equiv \left\{ q \in \mathbb{C} : q = \frac{\rho(e^{\mathrm{i}\theta})}{\sigma(e^{\mathrm{i}\theta})},\ 0 \le \theta < 2\pi \right\}. \tag{9.12}$$

The Γ is the set corresponding to the roots on the unit circumference and is called the boundary locus. We have the following definition and lemma, see [5].

Definition 9.1 *Consider a BVM with an A_{ν_1,ν_2}-stability polynomial $\pi(z,q)$ defined by (9.10). The region*

$$\mathcal{D}_{\nu_1,\nu_2} = \{q \in \mathbb{C} : \pi(z,q) \text{ has } \nu_1 \text{ zeros inside } |z| = 1 \text{ and } \nu_2 \text{ zeros outside } |z| = 1\}$$

is called the region of A_{ν_1,ν_2}-stability of the given BVM. Moreover, the BVM is said to be A_{ν_1,ν_2}-stable if

$$\mathbb{C}^- \subseteq \mathcal{D}_{\nu_1,\nu_2}.$$

Lemma 9.1 *If a BVM is A_{ν_1,ν_2}-stable and Γ is defined by (9.12), then $Re(q) \geq 0$ for all $q \in \Gamma$.*

Now, we want to show that the preconditioner S is invertible under the stability condition.

Theorem 9.1 *If the BVM for (9.1) is A_{ν_1,ν_2}-stable and $h\lambda_k(J_n) \in \mathcal{D}_{\nu_1,\nu_2}$ where $\lambda_k(J_n)$, $k = 1, \cdots, n$, are the eigenvalues of J_n, then the preconditioner S defined by (9.9) is invertible.*

Proof Since $s(A)$ and $s(B)$ are circulant matrices, their eigenvalues are given by

$$g_A(z) \equiv \alpha_\mu z^{\nu_2} + \ldots + \alpha_{\nu_1} + \alpha_{\nu_1-1}\frac{1}{z} + \ldots + \alpha_0\frac{1}{z^{\nu_1}} = \frac{\rho(z)}{z^{\nu_1}}$$

and

$$g_B(z) \equiv \beta_\mu z^{\nu_2} + \ldots + \beta_{\nu_1} + \beta_{\nu_1-1}\frac{1}{z} + \ldots + \beta_0\frac{1}{z^{\nu_1}} = \frac{\sigma(z)}{z^{\nu_1}},$$

evaluated at $\omega_j = e^{\frac{2\pi \mathbf{i} j}{s+1}}$ where $\mathbf{i} \equiv \sqrt{-1}$, for $j = 0, \cdots, s$, see [9, 13]. The eigenvalues $\lambda_{jk}(S)$ of S are therefore given by

$$\lambda_{jk}(S) = g_A(\omega_j) - h\lambda_k(J_n)g_B(\omega_j), \qquad j = 0, \cdots, s, \quad k = 1, \cdots, n.$$

Since the BVM is A_{ν_1,ν_2}-stable, the μ-degree polynomial

$$\pi[z, h\lambda_k(J_n)] = \rho(z) - h\lambda_k(J_n)\sigma(z)$$

has no roots on the unit circle $|z| = 1$ if $h\lambda_k(J_n) \in \mathcal{D}_{\nu_1,\nu_2}$. Thus for all $k = 1, \cdots, n$, and any arbitrary $|z| = 1$, we have

$$g_A(z) - h\lambda_k(J_n)g_B(z) = \frac{1}{z^{\nu_1}}\pi[z, h\lambda_k(J_n)] \neq 0.$$

It follows that

$$\lambda_{jk}(S) \neq 0, \qquad j = 0, \cdots, s, \quad k = 1, \cdots, n.$$

Thus S is invertible. □

In particular, we have

Corollary 9.1 *If the BVM is A_{ν_1,ν_2}-stable and $\lambda_k(J_n) \in \mathbb{C}^-$, then the preconditioner S is invertible.*

9.3.2 Convergence rate and operation cost

We have the following theorem for the convergence rate.

Theorem 9.2 *We have*

$$S^{-1}M = I_{n(s+1)} + L$$

where $\operatorname{rank}(L) \leq 2n\mu$.

Proof Let $E = M - S$. We have by (9.8) and (9.9),

$$E = \left(\widetilde{A} - s(A)\right) \otimes I_n - h\left(\widetilde{B} - s(B)\right) \otimes J_n = L_A \otimes I_n - hL_B \otimes J_n.$$

It is easy to check that L_A and L_B are $(s+1)$-by-$(s+1)$ matrices with nonzero entries only in the following four corners: a ν_1-by-$(\mu+1)$ block in the upper left; a ν_1-by-ν_1 block in the upper right; a ν_2-by-$(\mu+1)$ block in the lower right; and a ν_2-by-ν_2 block in the lower left. By noting that $\mu = \nu_1 + \nu_2$, we then have

$$\operatorname{rank}(L_A) \leq \mu, \qquad \operatorname{rank}(L_B) \leq \mu.$$

Therefore,

$$\operatorname{rank}(L_A \otimes I_n) = \operatorname{rank}(L_A) \cdot n \leq n\mu$$

and

$$\operatorname{rank}(L_B \otimes J_n) = \operatorname{rank}(L_B) \cdot n \leq n\mu.$$

Thus,

$$S^{-1}M = I_{n(s+1)} + S^{-1}E = I_{n(s+1)} + L,$$

where the rank of L is at most $2n\mu$. □

Therefore, when the GMRES method is applied to

$$S^{-1}M\mathbf{y} = S^{-1}\mathbf{b},$$

by Theorems 6.13 and 9.2, we know that the method will converge in at most $2n\mu + 1$ iterations in exact arithmetic.

Regarding the cost per iteration, the main work in each iteration for the GMRES method is the matrix-vector multiplication

$$S^{-1}M\mathbf{z} = (s(A) \otimes I_n - hs(B) \otimes J_n)^{-1} (\widetilde{A} \otimes I_n - h\widetilde{B} \otimes J_n)\mathbf{z},$$

see Section 6.5. Since $\widetilde{A}$, $\widetilde{B}$ are band matrices and J_n is assumed to be sparse, the matrix-vector multiplication

$$M\mathbf{z} = (\widetilde{A} \otimes I_n - h\widetilde{B} \otimes J_n)\mathbf{z}$$

can be done very fast.

Now we compute $S^{-1}(M\mathbf{z})$. Note that any circulant matrix can be diagonalized by the Fourier matrix F, see Section 6.4 and [13]. Since $s(A)$ and $s(B)$ are circulant, we have the following decompositions by (6.12),

$$s(A) = F\Lambda_A F^*, \qquad s(B) = F\Lambda_B F^*$$

where Λ_A, Λ_B are diagonal matrices containing the eigenvalues of $s(A)$, $s(B)$ respectively. It follows that

$$S^{-1}(M\mathbf{z}) = (F^* \otimes I_n)(\Lambda_A \otimes I_n - h\Lambda_B \otimes J_n)^{-1}(F \otimes I_n)(M\mathbf{z}).$$

This product can be obtained by using FFTs and solving $s+1$ linear systems of order n. Since J_n is sparse, the matrix

$$\Lambda_A \otimes I_n - h\Lambda_B \otimes J_n$$

will also be sparse. Thus $S^{-1}(M\mathbf{z})$ can be obtained by solving $s+1$ sparse linear systems of order n. It follows that the total number of operations per iteration is $\gamma_1 n(s+1)\log(s+1) + \gamma_2(s+1)nq$, where q is the number of nonzeros of J_n, and γ_1 and γ_2 are some positive constants. For comparing the computational cost of the method with direct solvers for the linear system (9.8), we refer to [10].

9.3.3 Numerical result

Now we give an example to illustrate the efficiency of the preconditioner S by solving a test problem given in [4]. The experiments were performed in MATLAB. We used the MATLAB-provided M-file "gmres" to solve the preconditioned systems. We should emphasize that in all of our tests in this chapter, the zero vector is the initial guess and the stopping criterion is

$$\frac{\|\mathbf{r}_q\|_2}{\|\mathbf{r}_0\|_2} < 10^{-6},$$

where $\mathbf{r}_q$ is the residual after q iterations. The BVM we used is the third order generalized Adam's method (GAM). Its formula and the initial and final additional conditions can be found in [5].

Example 9.1. Heat equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}, \\ u(0,t) = \dfrac{\partial u}{\partial x}(\pi,t) = 0, \quad t \in [0, 2\pi], \\ u(x,0) = x, \qquad\qquad x \in [0,\pi]. \end{cases}$$

We discretize the partial differential operator $\partial^2/\partial x^2$ with central differences and step size equals to $\pi/(n+1)$. The system of ODEs obtained is:

$$\begin{cases} \mathbf{y}'(t) = T_n \mathbf{y}(t), & t \in [0, 2\pi] \\ \mathbf{y}(0) = (x_1, x_2, \cdots, x_n)^T, \end{cases}$$

where T_n is a scaled discrete Laplacian matrix

$$T_n = \frac{(n+1)^2}{\pi^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}$$

Table 9.1 lists the number of iterations required for convergence of the GMRES method for different n and s. In the table, I means no preconditioner is used and S denotes the Strang-type block-circulant preconditioner defined by (9.9). We see that the number of iterations required for convergence, when S is used, is much less than that when no preconditioner is used. The numbers under the column S stay almost a constant for increasing s and n.

Table 9.1: Number of iterations for convergence.

n	s	I	S
24	6	19	4
	12	70	4
	24	152	4
	48	227	3
	96	314	3

n	s	I	S
48	6	47	4
	12	167	4
	24	359	4
	48	>400	3
	96	>400	3

9.4 Strang-type preconditioner for DDEs

Now, we study delay differential equations (DDEs).

9.4.1 Differential equations with multi-delays

Consider the solution of differential equations with multi-delays:

$$\begin{cases} \mathbf{y}'(t) = J_n \mathbf{y}(t) + D_n^{(1)} \mathbf{y}(t-\tau_1) + \cdots + D_n^{(s)} \mathbf{y}(t-\tau_s) + \mathbf{f}(t), & t \geq t_0, \\ \mathbf{y}(t) = \boldsymbol{\phi}(t), & t \leq t_0, \end{cases} \tag{9.13}$$

where $\mathbf{y}(t)$, $\mathbf{f}(t)$, $\boldsymbol{\phi}(t) : \mathbb{R} \to \mathbb{R}^n$; J_n, $D_n^{(1)}, \cdots, D_n^{(s)} \in \mathbb{R}^{n\times n}$; and $\tau_1, \cdots, \tau_s > 0$ are some rational numbers.

In order to find a reasonable numerical solution, we require that the solution of (9.13) is asymptotically stable. We have the following lemma, see [31, 43].

Lemma 9.2 *For any $s \geq 1$, if*

$$\eta(J_n) \equiv \frac{1}{2}\lambda_{\max}(J_n + J_n^T) < 0$$

and

$$\eta(J_n) + \sum_{j=1}^{s} \|D_n^{(j)}\|_2 < 0, \tag{9.14}$$

then the solution of (9.13) is asymptotically stable.

In the following, for simplicity, we only consider the case of $s = 2$ in (9.13). The generalization to any arbitrary s is straightforward. Let

$$h = \tau_1/m_1 = \tau_2/m_2$$

be the step size where m_1 and m_2 are positive integers with $m_2 > m_1$ ($\tau_2 > \tau_1$). For (9.13), by using a BVM with (ν_1, ν_2)-boundary conditions over a uniform mesh

$$t_j = t_0 + jh, \qquad j = 0, \cdots, r_1,$$

on the interval $[t_0, t_0 + r_1 h]$, we have

$$\sum_{i=0}^{\mu} \alpha_i \mathbf{y}_{p+i-\nu_1} = h \sum_{i=0}^{\mu} \beta_i (J_n \mathbf{y}_{p+i-\nu_1} + D_n^{(1)} \mathbf{y}_{p+i-\nu_1-m_1} + D_n^{(2)} \mathbf{y}_{p+i-\nu_1-m_2} + \mathbf{f}_{p+i-\nu_1}), \tag{9.15}$$

for $p = \nu_1, \cdots, r_1 - 1$, where $\mu = \nu_1 + \nu_2$. By providing the values

$$\mathbf{y}_{-m_2}, \cdots, \mathbf{y}_{-m_1}, \cdots, \mathbf{y}_0, \quad \mathbf{y}_1, \cdots, \mathbf{y}_{\nu_1-1}, \quad \mathbf{y}_{r_1}, \cdots, \mathbf{y}_{r_1+\nu_2-1}, \tag{9.16}$$

(9.15) can be written in a matrix form as

$$R\mathbf{y} = \mathbf{b}$$

where

$$R \equiv A \otimes I_n - hB \otimes J_n - hC^{(1)} \otimes D_n^{(1)} - hC^{(2)} \otimes D_n^{(2)}, \tag{9.17}$$

$$\mathbf{y} = (\mathbf{y}_{\nu_1}^T, \mathbf{y}_{\nu_1+1}^T, \cdots, \mathbf{y}_{r_1-1}^T)^T \in \mathbb{R}^{n(r_1-\nu_1)},$$

$\mathbf{b} \in \mathbb{R}^{n(r_1-\nu_1)}$ depends on $\mathbf{f}$, the boundary values and the coefficients of the method. The matrices A, $B \in \mathbb{R}^{(r_1-\nu_1)\times(r_1-\nu_1)}$ are defined as in (9.4) and $C^{(1)}$, $C^{(2)} \in \mathbb{R}^{(r_1-\nu_1)\times(r_1-\nu_1)}$ are defined as follows:

$$C^{(1)} = \begin{bmatrix} \mathbf{0} & & & & \\ \beta_\mu & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ \beta_0 & \cdots & \beta_\mu & \ddots & \\ & \ddots & & \ddots & \ddots \\ & & \beta_0 & \cdots & \beta_\mu \quad \mathbf{0} \end{bmatrix}, \quad C^{(2)} = \begin{bmatrix} \mathbf{0} & & & & \\ \beta_\mu & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ \beta_0 & \cdots & \beta_\mu & \ddots & \\ & \ddots & & \ddots & \ddots \\ & & \beta_0 & \cdots & \beta_\mu \quad \mathbf{0} \end{bmatrix}$$

We remark that the first column of $C^{(1)}$ is given by

$$(\underbrace{0,\cdots,0}_{m_1-\nu_2},\beta_\mu,\cdots,\beta_0,\ \underbrace{0,\cdots,0}_{r_1-m_1-2\nu_1-1})^T$$

and the first column of $C^{(2)}$ is given by

$$(\underbrace{0,\cdots,0}_{m_2-\nu_2},\beta_\mu,\cdots,\beta_0,\ \underbrace{0,\cdots,0}_{r_1-m_2-2\nu_1-1})^T.$$

9.4.2 Construction of preconditioner

The Strang-type block-circulant preconditioner for (9.17) is defined as follows:

$$\widetilde{S} \equiv s(A) \otimes I_n - hs(B) \otimes J_n - hs(C^{(1)}) \otimes D_n^{(1)} - hs(C^{(2)}) \otimes D_n^{(2)} \qquad (9.18)$$

where $s(E)$ is Strang's circulant preconditioner of Toeplitz matrix E, for $E = A$, B, $C^{(1)}$, $C^{(2)}$ respectively.

Now we discuss the invertibility of the Strang-type preconditioner $\widetilde{S}$ defined by (9.18). Since any circulant matrix can be diagonalized by the Fourier matrix F, we have by (6.12),

$$s(E) = F^* \Lambda_E F,$$

where Λ_E is the diagonal matrix holding the eigenvalues of $s(E)$, for $E = A$, B, $C^{(1)}$, $C^{(2)}$ respectively. Therefore, we obtain

$$\widetilde{S} = (F^* \otimes I_n)(\Lambda_A \otimes I_n - h\Lambda_B \otimes J_n - h\Lambda_{C^{(1)}} \otimes D_n^{(1)} - h\Lambda_{C^{(2)}} \otimes D_n^{(2)})(F \otimes I_n).$$

Note that the j-th block of

$$\Lambda_A \otimes I_n - h\Lambda_B \otimes J_n - h\Lambda_{C^{(1)}} \otimes D_n^{(1)} - h\Lambda_{C^{(2)}} \otimes D_n^{(2)}$$

is given by

$$\widetilde{S}_j = [\Lambda_A]_{jj} I_n - h[\Lambda_B]_{jj} J_n - h[\Lambda_{C^{(1)}}]_{jj} D_n^{(1)} - h[\Lambda_{C^{(2)}}]_{jj} D_n^{(2)},$$

for $j = 1, 2, \cdots, r_1 - \nu_1$. Let $w_j = e^{\frac{2\pi \mathbf{i} j}{r_1 - \nu_1}}$. We have

$$[\Lambda_A]_{jj} = \rho(w_j)/w_j^{\nu_1}, \qquad [\Lambda_B]_{jj} = \sigma(w_j)/w_j^{\nu_1},$$

$$[\Lambda_{C^{(1)}}]_{jj} = \beta_\mu w_j^{-m_1+\nu_2} + \cdots + \beta_0 w_j^{-m_1-\nu_1} = \sigma(w_j)/w_j^{m_1+\nu_1},$$

and

$$[\Lambda_{C^{(2)}}]_{jj} = \beta_\mu w_j^{-m_2+\nu_2} + \cdots + \beta_0 w_j^{-m_2-\nu_1} = \sigma(w_j)/w_j^{m_2+\nu_1},$$

where $\rho(z)$ and $\sigma(z)$ are defined as in (9.10). Therefore,

$$\widetilde{S}_j = \frac{1}{w_j^{m_2+\nu_1}} \left[w_j^{m_2} \left(\rho(w_j) I_n - h\sigma(w_j) J_n - h w_j^{-m_1} \sigma(w_j) D_n^{(1)} \right) - h\sigma(w_j) D_n^{(2)} \right].$$

We therefore only need to prove that $\widetilde{S}_j$, $j = 1, 2, \cdots, r_1 - \nu_1$, are invertible. We have the following theorem.

Theorem 9.3 *If the BVM with (ν_1, ν_2)-boundary conditions is A_{ν_1,ν_2}-stable and (9.14) holds, then for any arbitrary $\theta \in \mathbb{R}$, the matrix*

$$e^{\mathbf{i}m_2\theta} \left(\rho(e^{\mathbf{i}\theta}) I_n - h\sigma(e^{\mathbf{i}\theta}) J_n - h e^{-\mathbf{i}m_1\theta} \sigma(e^{\mathbf{i}\theta}) D_n^{(1)} \right) - h\sigma(e^{\mathbf{i}\theta}) D_n^{(2)}$$

is invertible. It follows that the Strang-type preconditioner $\widetilde{S}$ defined by (9.18) is also invertible.

Proof Suppose that there exist $\mathbf{x} \in \mathbb{C}^n$ with $\|\mathbf{x}\|_2 = 1$ and $\theta \in \mathbb{R}$ such that

$$\left[e^{\mathbf{i}m_2\theta} \left(\rho(e^{\mathbf{i}\theta}) I_n - h\sigma(e^{\mathbf{i}\theta}) J_n - h e^{-\mathbf{i}m_1\theta} \sigma(e^{\mathbf{i}\theta}) D_n^{(1)} \right) - h\sigma(e^{\mathbf{i}\theta}) D_n^{(2)} \right] \mathbf{x} = 0.$$

Then

$$\mathbf{x}^* \left[\rho(e^{\mathbf{i}\theta}) I_n - h\sigma(e^{\mathbf{i}\theta}) J_n - h e^{-\mathbf{i}m_1\theta} \sigma(e^{\mathbf{i}\theta}) D_n^{(1)} - h e^{-\mathbf{i}m_2\theta} \sigma(e^{\mathbf{i}\theta}) D_n^{(2)} \right] \mathbf{x} = 0,$$

i.e.,

$$\rho(e^{\mathbf{i}\theta}) - h\sigma(e^{\mathbf{i}\theta}) \mathbf{x}^* J_n \mathbf{x} - h e^{-\mathbf{i}m_1\theta} \sigma(e^{\mathbf{i}\theta}) \mathbf{x}^* D_n^{(1)} \mathbf{x} - h e^{-\mathbf{i}m_2\theta} \sigma(e^{\mathbf{i}\theta}) \mathbf{x}^* D_n^{(2)} \mathbf{x} = 0.$$

We therefore have

$$\rho(e^{\mathbf{i}\theta}) - (h\mathbf{x}^* J_n \mathbf{x} + h e^{-\mathbf{i}m_1\theta} \mathbf{x}^* D_n^{(1)} \mathbf{x} + h e^{-\mathbf{i}m_2\theta} \mathbf{x}^* D_n^{(2)} \mathbf{x}) \sigma(e^{\mathbf{i}\theta})$$

$$= \pi \left(e^{\mathbf{i}\theta}, h(\mathbf{x}^* J_n \mathbf{x} + e^{-\mathbf{i}m_1\theta} \mathbf{x}^* D_n^{(1)} \mathbf{x} + e^{-\mathbf{i}m_2\theta} \mathbf{x}^* D_n^{(2)} \mathbf{x}) \right) = 0$$

where $\pi(z,q)$ is given by (9.11). Thus,

$$h(\mathbf{x}^* J_n \mathbf{x} + e^{-\mathrm{i}m_1\theta}\mathbf{x}^* D_n^{(1)}\mathbf{x} + e^{-\mathrm{i}m_2\theta}\mathbf{x}^* D_n^{(2)}\mathbf{x}) \in \Gamma,$$

where Γ is the boundary locus defined by (9.12). Since the BVM is A_{ν_1,ν_2}-stable, from Lemma 9.1, we know that

$$\mathrm{Re}(\mathbf{x}^* J_n \mathbf{x} + e^{-\mathrm{i}m_1\theta}\mathbf{x}^* D_n^{(1)}\mathbf{x} + e^{-\mathrm{i}m_2\theta}\mathbf{x}^* D_n^{(2)}\mathbf{x}) \geq 0.$$

By Cauchy-Schwarz inequality, we have

$$\mathrm{Re}(e^{-\mathrm{i}m_1\theta}\mathbf{x}^* D_n^{(1)}\mathbf{x}) \leq |\mathbf{x}^* D_n^{(1)}\mathbf{x}| \leq \|\mathbf{x}\|_2 \|D_n^{(1)}\mathbf{x}\|_2 \leq \|D_n^{(1)}\|_2 \|\mathbf{x}\|_2 = \|D_n^{(1)}\|_2,$$

and similarly, $\mathrm{Re}(e^{-\mathrm{i}m_1\theta}\mathbf{x}^* D_n^{(2)}\mathbf{x}) \leq \|D_n^{(2)}\|_2$. Note that

$$\eta(J_n) = \max_{\|\mathbf{x}\|_2=1} \mathrm{Re}(\mathbf{x}^* J_n \mathbf{x}) \geq \mathrm{Re}(\mathbf{x}^* J_n \mathbf{x}).$$

Thus we have

$$\eta(J_n) + \|D_n^{(1)}\|_2 + \|D_n^{(2)}\|_2 \geq 0,$$

which is a contradiction to (9.14). Therefore, the matrix

$$e^{\mathrm{i}m_2\theta}\left(\rho(e^{\mathrm{i}\theta})I_n - h\sigma(e^{\mathrm{i}\theta})J_n - he^{-\mathrm{i}m_1\theta}\sigma(e^{\mathrm{i}\theta})D_n^{(1)}\right) - h\sigma(e^{\mathrm{i}\theta})D_n^{(2)}$$

is invertible and it follows that the Strang-type preconditioner $\widetilde{S}$ is also invertible. □

9.4.3 Convergence rate

Now, we discuss the convergence rate of the preconditioned GMRES method with the Strang-type block-circulant preconditioner. We have the following result for the spectra of preconditioned matrices, see [23].

Theorem 9.4 *Let R be given by (9.17) and $\widetilde{S}$ be given by (9.18). Then we have*

$$\widetilde{S}^{-1}R = I_{n(r_1-\nu_1)} + L$$

where $\mathrm{rank}(L) \leq (2\mu + m_1 + m_2 + 2\nu_1 + 2)n$.

By Theorems 6.13 and 9.4, when the GMRES method is applied to

$$\widetilde{S}^{-1}R\mathbf{y} = \widetilde{S}^{-1}\mathbf{b},$$

the method will converge in at most $(2\mu + m_1 + m_2 + 2\nu_1 + 2)n + 1$ iterations in exact arithmetic.

We know from Theorem 9.4 that if the step size $h = \tau_1/m_1 = \tau_2/m_2$ is fixed, the number of iterations for convergence of the GMRES method, when applied to the preconditioned system

$$\widetilde{S}^{-1}R\mathbf{y} = \widetilde{S}^{-1}\mathbf{b},$$

will be independent of r_1 and therefore is independent of the length of the interval that we considered. We should emphasize that the numerical example in Section 9.4.4 shows a much faster convergence rate than that predicted by the estimate provided by Theorem 9.4. For the operation cost of our algorithm, we refer to [23, 30].

9.4.4 Numerical result

We illustrate the efficiency of our preconditioner by solving the following example. The BVM we used is the third order GBDF for $t \in [0, 4]$.

Example 9.2. Consider

$$\begin{cases} \mathbf{y}'(t) = J_n\mathbf{y}(t) + D_n^{(1)}\mathbf{y}(t-0.5) + D_n^{(2)}\mathbf{y}(t-1), & t \geq 0, \\ \mathbf{y}(t) = (\sin t, 1, \cdots, 1)^T, & t \leq 0, \end{cases}$$

where

$$J_n = \begin{bmatrix} -10 & 2 & & & \\ 2 & \ddots & \ddots & & \\ 1 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & 2 \\ & & 1 & 2 & -10 \end{bmatrix}, \qquad D_n^{(1)} = \frac{1}{n}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix},$$

and

$$D_n^{(2)} = \frac{1}{n}\begin{bmatrix} 2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 2 \end{bmatrix}$$

In practice, we do not have the boundary values

$$\mathbf{y}_1, \cdots, \mathbf{y}_{\nu_1 - 1}, \qquad \mathbf{y}_{r_1}, \cdots, \mathbf{y}_{r_1+\nu_2-1},$$

provided by (9.16). Instead of giving the above values, as in Section 9.2, $\nu_1 - 1$ initial additional equations and ν_2 final additional equations are given. We remark that after introducing the additional equations, the matrices $A, B, C^{(1)}$

and $C^{(2)}$ in (9.17) are Toeplitz matrices with small rank perturbations. Neglecting the small rank perturbations, we can also construct the Strang-type preconditioner (9.18).

Table 9.2 shows the number of iterations required for convergence of the GMRES method with different combinations of matrix size n and step size h. In the table, I means no preconditioner is used and $\widetilde{S}$ denotes the Strang-type block-circulant preconditioner defined by (9.18). We see that the numbers of iterations required for convergence increase slowly for increasing n and decreasing h under the column $\widetilde{S}$.

Table 9.2: Number of iterations for convergence.

n	h	I	$\widetilde{S}$
24	1/10	52	9
	1/20	97	11
	1/40	185	15
	1/80	367	19

n	h	I	$\widetilde{S}$
48	1/10	53	12
	1/20	98	14
	1/40	189	14
	1/80	378	17

9.5 Strang-type preconditioner for NDDEs

In this section, we study neutral delay differential equations (NDDEs).

9.5.1 Neutral delay differential equations

We consider the solution of NDDE:

$$\begin{cases} \mathbf{y}'(t) = L_n\mathbf{y}'(t-\tau) + M_n\mathbf{y}(t) + N_n\mathbf{y}(t-\tau), & t \geq t_0, \\ \mathbf{y}(t) = \boldsymbol{\phi}(t), & t \leq t_0, \end{cases} \tag{9.19}$$

where $\mathbf{y}(t)$, $\boldsymbol{\phi}(t) : \mathbb{R} \to \mathbb{R}^n$; L_n, M_n, $N_n \in \mathbb{R}^{n\times n}$, and $\tau > 0$ is a constant.

As in Section 9.4.1, we want to find an asymptotically stable solution for (9.19). We have the following lemma, see [18, 28].

Lemma 9.3 *Let L_n, M_n and N_n be any matrices with $\|L_n\|_2 < 1$. Then the solution of (9.19) is asymptotically stable if $Re(\lambda_i) < 0$ where λ_i, $i = 1, \cdots, n$, are the eigenvalues of matrix*

$$(I_n - \eta L_n)^{-1}(M_n + \eta N_n)$$

with $|\eta| \leq 1$.

Let $h = \tau/k_1$ be the step size where k_1 is a positive integer. For (9.19), by using a BVM with (ν_1, ν_2)-boundary conditions over a uniform mesh

$$t_j = t_0 + jh, \qquad j = 0, \cdots, r_2,$$

on the interval $[t_0, t_0 + r_2 h]$, we have

$$\sum_{i=0}^{\mu} \alpha_i \mathbf{y}_{p+i-\nu_1} = \sum_{i=0}^{\mu} \alpha_i L_n \mathbf{y}_{p+i-\nu_1-k_1} + h \sum_{i=0}^{\mu} \beta_i (M_n \mathbf{y}_{p+i-\nu_1} + N_n \mathbf{y}_{p+i-\nu_1-k_1}), \tag{9.20}$$

for $p = \nu_1, \cdots, r_2 - 1$, where $\mu = \nu_1 + \nu_2$. By providing the values

$$\mathbf{y}_{-k_1}, \cdots, \mathbf{y}_0, \quad \mathbf{y}_1, \cdots, \mathbf{y}_{\nu_1 - 1}, \quad \mathbf{y}_{r_2}, \cdots, \mathbf{y}_{r_2+\nu_2-1}, \tag{9.21}$$

(9.20) can be written in a matrix form as

$$H\mathbf{y} = \mathbf{b}$$

where

$$H \equiv A \otimes I_n - A^{(1)} \otimes L_n - hB \otimes M_n - hB^{(1)} \otimes N_n, \tag{9.22}$$
$$\mathbf{y} = (\mathbf{y}_{\nu_1}^T, \mathbf{y}_{\nu_1+1}^T, \cdots, \mathbf{y}_{r_2-1}^T)^T \in \mathbb{R}^{n(r_2-\nu_1)},$$

$\mathbf{b} \in \mathbb{R}^{n(r_2-\nu_1)}$ depends on the boundary values and the coefficients of the method. In (9.22), the matrices A, $B \in \mathbb{R}^{(r_2-\nu_1)\times(r_2-\nu_1)}$ are defined as in (9.4), and $A^{(1)}$, $B^{(1)} \in \mathbb{R}^{(r_2-\nu_1)\times(r_2-\nu_1)}$ are given as follows:

$$A^{(1)} = \begin{bmatrix} \mathbf{0} & & & & & \\ \alpha_\mu & \ddots & & & & \\ \vdots & \ddots & \ddots & & & \\ \alpha_0 & \cdots & \alpha_\mu & \ddots & & \\ & \ddots & & \ddots & \ddots & \\ & & \alpha_0 & \cdots & \alpha_\mu & \mathbf{0} \end{bmatrix}, \quad B^{(1)} = \begin{bmatrix} \mathbf{0} & & & & & \\ \beta_\mu & \ddots & & & & \\ \vdots & \ddots & \ddots & & & \\ \beta_0 & \cdots & \beta_\mu & \ddots & & \\ & \ddots & & \ddots & \ddots & \\ & & \beta_0 & \cdots & \beta_\mu & \mathbf{0} \end{bmatrix}$$

see [3]. We remark that the first column of $A^{(1)}$ is given by:

$$(\underbrace{0, \cdots, 0}_{k_1-\nu_2}, \alpha_\mu, \cdots, \alpha_0, \underbrace{0, \cdots, 0}_{r_2-k_1-2\nu_1-1})^T$$

and the first column of $B^{(1)}$ is given by

$$(\underbrace{0, \cdots, 0}_{k_1-\nu_2}, \beta_\mu, \cdots, \beta_0, \underbrace{0, \cdots, 0}_{r_2-k_1-2\nu_1-1})^T.$$

9.5.2 Construction of preconditioner

The Strang-type block-circulant preconditioner for (9.22) is defined as follows:

$$\bar{S} \equiv s(A) \otimes I_n - s(A^{(1)}) \otimes L_n - hs(B) \otimes M_n - hs(B^{(1)}) \otimes N_n \qquad (9.23)$$

where $s(E)$ is Strang's circulant preconditioner of Toeplitz matrix E, for $E = A$, B, $A^{(1)}$, $B^{(1)}$ respectively. By using (6.12) again, we have

$$\bar{S} = (F^* \otimes I_n)\,(\Lambda_A \otimes I_n - \Lambda_{A^{(1)}} \otimes L_n - h\Lambda_B \otimes M_n - h\Lambda_{B^{(1)}} \otimes N_n)\,(F \otimes I_n),$$

where Λ_E is the diagonal matrix holding the eigenvalues of $s(E)$, for $E = A$, B, $A^{(1)}$, $B^{(1)}$ respectively.

Now we discuss the invertibility of the Strang-type preconditioner $\bar{S}$. Let $w_j = e^{\frac{2\pi ij}{r_2-\nu_1}}$. We have

$$[\Lambda_A]_{jj} = \rho(w_j)/w_j^{\nu_1}, \qquad [\Lambda_B]_{jj} = \sigma(w_j)/w_j^{\nu_1},$$

$$[\Lambda_{A^{(1)}}]_{jj} = \alpha_\mu w_j^{-k_1+\nu_2} + \cdots + \alpha_0 w_j^{-k_1-\nu_1} = \rho(w_j)/w_j^{k_1+\nu_1},$$

and

$$[\Lambda_{B^{(1)}}]_{jj} = \beta_\mu w_j^{-k_1+\nu_2} + \cdots + \beta_0 w_j^{-k_1-\nu_1} = \sigma(w_j)/w_j^{k_1+\nu_1},$$

where $\rho(z)$ and $\sigma(z)$ are defined as in (9.10). Thus the j-th block of

$$\Lambda_A \otimes I_n - \Lambda_{A^{(1)}} \otimes L_n - h\Lambda_B \otimes M_n - h\Lambda_{B^{(1)}} \otimes N_n$$

in $\bar{S}$ is given by

$$\begin{aligned}\bar{S}_j &= [\Lambda_A]_{jj} I_n - [\Lambda_{A^{(1)}}]_{jj} L_n - h[\Lambda_B]_{jj} M_n - h[\Lambda_{B^{(1)}}]_{jj} N_n \\ &= \frac{1}{w_j^{k_1+\nu_1}}\Big[w_j^{k_1}\,(\rho(w_j) I_n - h\sigma(w_j) M_n) - \rho(w_j) L_n - h\sigma(w_j) N_n\Big],\end{aligned}$$

for $j = 1, 2, \cdots, r_2 - \nu_1$. In order to prove that $\bar{S}$ is invertible, we only need to show that $\bar{S}_j$, $j = 1, 2, \cdots, r_2 - \nu_1$, are invertible. Let

$$\begin{aligned}\Delta &\equiv e^{\mathbf{i}k_1\theta}\left(\rho(e^{\mathbf{i}\theta}) I_n - h\sigma(e^{\mathbf{i}\theta}) M_n\right) - \rho(e^{\mathbf{i}\theta}) L_n - h\sigma(e^{\mathbf{i}\theta}) N_n \\ &= e^{\mathbf{i}k_1\theta}(I_n - e^{-\mathbf{i}k_1\theta} L_n) D\end{aligned}$$

where

$$D \equiv \rho(e^{\mathbf{i}\theta}) I_n - h(I_n - e^{-\mathbf{i}k_1\theta} L_n)^{-1}(M_n + e^{-\mathbf{i}k_1\theta} N_n)\sigma(e^{\mathbf{i}\theta}). \qquad (9.24)$$

Hence, we are required to show that Δ is invertible for any $\theta \in \mathbb{R}$ in order to prove that $\bar{S}_j$ is invertible. Assume that $\|L_n\|_2 < 1$, we have $I_n - e^{-\mathbf{i}k_1\theta}L_n$ is nonsingular for any $\theta \in \mathbb{R}$. Therefore, we only need to show D is invertible for any $\theta \in \mathbb{R}$. We have the following theorem, see [3].

Theorem 9.5 *If the BVM with (ν_1, ν_2)-boundary conditions is A_{ν_1,ν_2}-stable and $Re(\lambda_i) < 0$ where λ_i, $i = 1, \cdots, n$, are the eigenvalues of matrix*

$$(I_n - \eta L_n)^{-1}(M_n + \eta N_n)$$

with $|\eta| \leq 1$, then for any $\theta \in \mathbb{R}$, the matrix D defined by (9.24) is invertible. It follows that the Strang-type preconditioner $\bar{S}$ defined as in (9.23) is also invertible.

Proof Let

$$U \equiv (I_n - e^{-\mathbf{i}m\theta}L_n)^{-1}(M_n + e^{-\mathbf{i}m\theta}N_n).$$

Then D can be written as

$$D = \rho(z)I_n - hU\sigma(z).$$

Note that the eigenvalues of D are given by

$$\lambda_i(D) = \rho(z) - h\lambda_i(U)\sigma(z), \qquad i = 1, \cdots, n,$$

where $\lambda_i(U)$, $i = 1, \cdots, n$, denote the eigenvalues of U. Since we know that

$$\mathrm{Re}[\lambda_i(U)] < 0, \qquad i = 1, \cdots, n,$$

it follows that $h\lambda_i(U) \in \mathbb{C}^-$. Note that the BVM is A_{ν_1,ν_2}-stable and then we have

$$h\lambda_i(U) \in \mathbb{C}^- \subseteq \mathcal{D}_{\nu_1,\nu_2}.$$

Therefore, the A_{ν_1,ν_2}-stability polynomial defined by (9.11)

$$\pi[z, h\lambda_i(U)] \equiv \rho(z) - h\lambda_i(U)\sigma(z)$$

has no roots on the unit circle $|z| = 1$. Thus, for any $|z| = 1$, we have

$$\lambda_i(D) = \rho(z) - h\lambda_i(U)\sigma(z) = \pi[z, h\lambda_i(U)] \neq 0, \qquad i = 1, \cdots, n.$$

It follows that D is invertible. Therefore, the Strang-type preconditioner $\bar{S}$ defined as in (9.23) is also invertible. □

9.5.3 Convergence rate

We have the following result for the spectra of preconditioned matrices, see [3].

Theorem 9.6 *Let H be given by (9.22) and $\bar{S}$ be given by (9.23). Then we have*

$$\bar{S}^{-1}H = I_{n(r_2-\nu_1)} + L$$

where $\text{rank}(L) \leq 2(\mu + k_1 + \nu_1 + 1)n$.

By Theorems 6.13 and 9.6 , when the GMRES method is applied to

$$\bar{S}^{-1}H\mathbf{y} = \bar{S}^{-1}\mathbf{b},$$

the method will converge in at most $2(\mu+k_1+\nu_1+1)n+1$ iterations in exact arithmetic.

We observe from Theorem 9.6 that if the step size $h = \tau/k_1$ is fixed, the number of iterations for convergence of the GMRES method, when applied for solving

$$\bar{S}^{-1}H\mathbf{y} = \bar{S}^{-1}\mathbf{b},$$

is independent of r_2, i.e., the length of the interval that we considered.

9.5.4 Numerical result

We illustrate the efficiency of our preconditioner by solving the following example.

Example 9.3. Consider

$$\begin{cases} \mathbf{y}'(t) = L_n\mathbf{y}'(t-1) + M_n\mathbf{y}(t) + N_n\mathbf{y}(t-1), & t \geq 0, \\ \mathbf{y}(t) = (1, 1, \cdots, 1)^T, & t \leq 0, \end{cases}$$

where

$$L_n = \frac{1}{n}\begin{bmatrix} 2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 2 \end{bmatrix}, \quad M_n = \begin{bmatrix} -8 & 2 & 1 & & \\ 2 & \ddots & \ddots & \ddots & \\ 1 & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & 2 \\ & & 1 & 2 & -8 \end{bmatrix},$$

and

$$N_n = \frac{1}{n}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}.$$

Example 9.3 is solved by using the fifth order GAM for $t \in [0, 4]$. In practice, we do not have the boundary values

$$\mathbf{y}_1, \cdots, \mathbf{y}_{\nu_1 - 1}, \qquad \mathbf{y}_{r_2}, \cdots, \mathbf{y}_{r_2 + \nu_2 - 1},$$

provided by (9.21). Again as in Section 9.2, instead of giving the above values, $\nu_1 - 1$ initial additional equations and ν_2 final additional equations are given. After introducing the additional equations, the matrices $A, A^{(1)}, B$ and $B^{(2)}$ in (9.22) are Toeplitz matrices with small rank perturbations. We can also construct the Strang-type preconditioner (9.23) by neglecting the small rank perturbations.

Table 9.3 lists the number of iterations required for convergence of the GMRES method for different n and k_1. In the table, I means no preconditioner is used and $\bar{S}$ denotes the Strang-type block-circulant preconditioner defined by (9.23). We see that the number of iterations required for convergence, when $\bar{S}$ is used, is much less than that when no preconditioner is used. We should emphasize that our numerical example shows a much faster convergence rate than that predicted by the estimate provided by Theorem 9.6.

Table 9.3: Number of iterations for convergence ('$*$' means out of memory).

n	k_1	I	$\bar{S}$
24	10	43	7
	20	83	7
	40	161	7
	80	*	7

n	k_1	I	$\bar{S}$
48	10	44	6
	20	83	6
	40	163	6
	80	*	6

9.6 Strang-type preconditioner for SPDDEs

In this section, we study the solution of singular perturbation delay differential equations (SPDDEs).

9.6.1 Singular perturbation delay differential equations

We consider the solution of SPDDE:

$$\begin{cases} \mathbf{x}'(t) = V^{(1)}\mathbf{x}(t) + V^{(2)}\mathbf{x}(t-\tau) + C^{(1)}\mathbf{y}(t) + C^{(2)}\mathbf{y}(t-\tau), & t \geq t_0, \\ \epsilon\mathbf{y}'(t) = F^{(1)}\mathbf{x}(t) + F^{(2)}\mathbf{x}(t-\tau) + G^{(1)}\mathbf{y}(t) + G^{(2)}\mathbf{y}(t-\tau), & t \geq t_0, \\ \mathbf{x}(t) = \boldsymbol{\phi}(t), & t \leq t_0, \\ \mathbf{y}(t) = \boldsymbol{\psi}(t), & t \leq t_0, \end{cases} \tag{9.25}$$

where

$$\mathbf{x}(t), \quad \boldsymbol{\phi}(t) : \mathbb{R} \to \mathbb{R}^m; \qquad \mathbf{y}(t), \quad \boldsymbol{\psi}(t) : \mathbb{R} \to \mathbb{R}^n;$$

$$V^{(1)},\ V^{(2)} \in \mathbb{R}^{m\times m}; \qquad C^{(1)},\ C^{(2)} \in \mathbb{R}^{m\times n};$$

$$F^{(1)},\ F^{(2)} \in \mathbb{R}^{n\times m}; \qquad G^{(1)},\ G^{(2)} \in \mathbb{R}^{n\times n};$$

and $\tau > 0, \quad 0 < \epsilon \ll 1$ are constants. We can rewrite SPDDE (9.25) as the following initial value problem:

$$\begin{cases} \mathbf{z}'(t) = P\mathbf{z}(t) + Q\mathbf{z}(t-\tau), & t \geq t_0, \\ \mathbf{z}(t) = \begin{pmatrix} \boldsymbol{\phi}(t) \\ \boldsymbol{\psi}(t) \end{pmatrix}, & t \leq t_0, \end{cases} \tag{9.26}$$

where P and $Q \in \mathbb{R}^{(m+n)\times(m+n)}$ are defined as follows,

$$P = \begin{bmatrix} V^{(1)} & C^{(1)} \\ \epsilon^{-1}F^{(1)} & \epsilon^{-1}G^{(1)} \end{bmatrix}, \qquad Q = \begin{bmatrix} V^{(2)} & C^{(2)} \\ \epsilon^{-1}F^{(2)} & \epsilon^{-1}G^{(2)} \end{bmatrix}.$$

see [24]. Let

$$h = \tau/k_2$$

be the step size where k_2 is a positive integer. For (9.26), by using a BVM with (ν_1, ν_2)-boundary conditions over a uniform mesh

$$t_j = t_0 + jh, \qquad j = 0, \cdots, v,$$

on the interval $[t_0, t_0 + vh]$, we have

$$\sum_{i=0}^{\mu} \alpha_i \mathbf{z}_{p+i-\nu_1} = h \sum_{i=0}^{\mu} \beta_i (P\mathbf{z}_{p+i-\nu_1} + Q\mathbf{z}_{p+i-\nu_1-k_2}), \tag{9.27}$$

for $p = \nu_1, \cdots, v-1$, where $\mu = \nu_1 + \nu_2$. By providing the values

$$\mathbf{z}_{-k_2}, \cdots, \mathbf{z}_0, \quad \mathbf{z}_1, \cdots, \mathbf{z}_{\nu_1-1}, \quad \mathbf{z}_v, \cdots, \mathbf{z}_{v+\nu_2-1}, \tag{9.28}$$

(9.27) can be written in a matrix form as

$$K\mathbf{v} = \mathbf{b}, \tag{9.29}$$

where

$$K \equiv A \otimes I_{m+n} - hB \otimes P - hU \otimes Q. \tag{9.30}$$

The vector $\mathbf{v}$ in (9.29) is defined by

$$\mathbf{v} = (\mathbf{z}_{\nu_1}^T, \mathbf{z}_{\nu_1+1}^T, \cdots, \mathbf{z}_{v-1}^T)^T \in \mathbb{R}^{(m+n)(v-\nu_1)}.$$

The right-hand side $\mathbf{b} \in \mathbb{R}^{(m+n)(v-\nu_1)}$ of (9.29) depends on the boundary values and the coefficients of the method. The matrices A, $B \in \mathbb{R}^{(v-\nu_1)\times(v-\nu_1)}$ in (9.30) are defined as in (9.4) and $U \in \mathbb{R}^{(v-\nu_1)\times(v-\nu_1)}$ in (9.30) is defined as the matrix $C^{(1)}$ in (9.17).

9.6.2 Construction of preconditioner

The Strang-type block-circulant preconditioner can be constructed for solving (9.29):

$$\widehat{S} \equiv s(A) \otimes I_{m+n} - hs(B) \otimes P - hs(U) \otimes Q, \tag{9.31}$$

where $s(E)$ is Strang's circulant preconditioner of matrix E, for $E = A$, B, U respectively. We have the following theorem for the invertibility of our preconditioner $\widehat{S}$ and for the convergence rate of our method, see [24].

Theorem 9.7 *If the BVM with (ν_1, ν_2)-boundary conditions is A_{ν_1,ν_2}-stable,*

$$\eta(P) \equiv \frac{1}{2}\lambda_{\max}(P + P^T) < 0, \qquad \eta(P) + \|Q\|_2 < 0,$$

then the Strang-type preconditioner $\widehat{S}$ defined by (9.31) is invertible. Moreover, when the GMRES method is applied to

$$\widehat{S}^{-1}K\mathbf{v} = \widehat{S}^{-1}\mathbf{b},$$

the method will converge in at most $O(m+n)$ iterations in exact arithmetic.

9.6.3 Numerical result

We test the following SPDDE.

Example 9.4. Consider

$$\begin{cases} \mathbf{x}'(t) = A\mathbf{x}(t) + B\mathbf{x}(t-1) + C\mathbf{y}(t) + D\mathbf{y}(t-1), & t \geq 0, \\ \epsilon\mathbf{y}'(t) = E\mathbf{x}(t) + F\mathbf{x}(t-1) + G\mathbf{y}(t) + H\mathbf{y}(t-1), & t \geq 0, \quad \epsilon = 0.001, \\ \mathbf{x}(t) = (1, 1, \ldots, 1)^T, & t \leq 0, \\ \mathbf{y}(t) = (2, 2, \ldots, 2)^T, & t \leq 0, \end{cases}$$

where

$$A = \begin{bmatrix} -10 & 2 & 1 & & \\ 2 & \ddots & \ddots & \ddots & \\ 1 & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & 2 \\ & & 1 & 2 & -10 \end{bmatrix}_{m\times m}, \quad B = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}_{m\times m},$$

$$C = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}_{m\times n}, \quad D = 3C, \quad E = \begin{bmatrix} 2 & -1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & -1 & \\ & & 1 & 2 & -1 \end{bmatrix}_{n\times m},$$

$$F = \begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & -1 & \\ & & & 1 & -1 \end{bmatrix}_{n\times m}, \quad G = \begin{bmatrix} -2 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & -2 \end{bmatrix}_{n\times n},$$

and

$$H = \begin{bmatrix} -5 & -1 & & \\ 2 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & 2 & -5 \end{bmatrix}_{n\times n}.$$

Example 9.4 is solved by using the third order GAM for $t \in [0, 4]$. Table 9.4 lists the number of iterations required for convergence of the GMRES method for different m, n and k_2. In the table, $\widehat{S}$ denotes the Strang-type block-circulant preconditioner defined by (9.31).

Table 9.4: Number of iterations for convergence.

k_2	m	n	I	$\widehat{S}$
24	8	2	55	29
	16	4	83	57
	32	8	109	90

k_2	m	n	I	$\widehat{S}$
48	8	2	100	34
	16	4	131	63
	32	8	177	90

Bibliography

[1] P. Amodio, F. Mazzia and D. Trigiante, *Stability of Some Boundary Value Methods for the Solution of Initial Value Problems*, BIT, vol. 33 (1993), pp. 434–451.

[2] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1996.

[3] Z. Bai, X. Jin and L. Song, *Strang-type Preconditioners for Solving Linear Systems from Neutral Delay Differential Equations*, Calcolo, vol. 40 (2003), pp. 21–31.

[4] D. Bertaccini, *A Circulant Preconditioner for the Systems of LMF-Based ODE Codes*, SIAM J. Sci. Comput., vol. 22 (2000), pp.767–786.

[5] L. Brugnano and D. Trigiante, *Solving Differential Problems by Multistep Initial and Boundary Value Methods*, Gordon and Berach Science Publishers, Amsterdam, 1998.

[6] Z. Cao, *Numerical Linear Algebra* (in Chinese), Fudan University Press, Shanghai, 1996.

[7] R. Chan and X. Jin, *A Family of Block Preconditioners for Block Systems*, SIAM J. Sci. Statist. Comput., vol. 13 (1992), pp. 1218–1235.

[8] R. Chan, X. Jin and Y. Tam, *Strang-type Preconditioners for Solving System of ODEs by Boundary Value Methods*, Electron. J. Math. Phys. Sci., vol. 1 (2002), pp. 14–46.

[9] R. Chan and M. Ng, *Conjugate Gradient Methods for Toeplitz Systems*, SIAM Review, vol. 38 (1996), pp. 427–482.

[10] R. Chan, M. Ng and X. Jin, *Strang-type Preconditioners for Systems of LMF-Based ODE Codes*, IMA J. Numer. Anal., vol. 21 (2001), pp. 451–462.

[11] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Statist. Comput., vol. 9 (1988), pp. 766–771.

[12] W. Ching, *Iterative Methods for Queuing and Manufacturing Systems*, Springer-Verlag, London, 2001.

[13] P. Davis, *Circulant Matrices*, 2nd edition, AMS Chelsea Publishing, Rhode Island, 1994.

[14] J. Demmel, *Applied Numerical Linear Algebra*, SIAM Press, Philadelphia, 1997.

[15] H. Diao, Y. Wei and S. Qiao, *Displacement Rank of the Drazin Inverse*, J. Comput. Appl. Math., vol. 167 (2004), pp. 147–161.

[16] M. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, J. Res. Nat. Bur. Stand., vol. 49 (1952), pp. 409–436.

[17] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[18] G. Hu and T. Mitsui, *Stability Analysis of Numerical Methods for Systems of Neutral Delay-Differential Equations*, BIT, vol. 35 (1995), pp. 504–515.

[19] G. Golub and C. Van Loan, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore, 1996.

[20] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM Press, Philadephia, 1997.

[21] M. Gulliksson, X. Jin and Y. Wei, *Perturbation Bounds for Constrained and Weighted Least Squares Problems*, Linear Algebra Appl., vol. 349 (2002), pp. 221–232.

[22] X. Jin, *Developments and Applications of Block Toeplitz Iterative Solvers*, Kluwer Academic Publishers, Dordrecht; and Science Press, Beijing, 2002.

[23] X. Jin, S. Lei and Y. Wei, *Circulant Preconditioners for Solving Differential Equations with Multi-Delays*, Comput. Math. Appl., vol. 47 (2004), pp. 1429–1436.

[24] X. Jin, S. Lei and Y. Wei, *Circulant Preconditioners for Solving Singular Perturbation Delay Differential Equations.* Numer. Linear Algebra Appl., vol. 12 (2005), pp. 327–336.

[25] X. Jin, V. Sin and L. Song, *Circulant Preconditioned WR-BVM Methods for ODE systems*, J. Comput. Appl. Math., vol. 162 (2004), pp. 201–211.

[26] X. Jin, V. Sin and L. Song, *Preconditioned WR-LMF-Based Method for ODE systems*, J. Comput. Appl. Math., vol. 162 (2004), pp. 431–444.

[27] X. Jin, Y. Wei and W. Xu, *A Stability Property of T. Chan's Preconditioner*, SIAM J. Matrix Anal. Appl., vol. 25 (2003), pp. 627–629.

[28] J. Kuang, J. Xiang and H. Tian *The Asymptotic Stability of One Parameter Methods for Neutral Differential Equations*, BIT, vol. 34 (1994), pp. 400–408.

[29] S. Lei and X. Jin, *BCCB Preconditioners for Systems of BVM-Based Numerical Integrators*, Numer. Linear Algebra Appl., vol. 11 (2004), pp. 25–40.

[30] F. Lin, X. Jin and S. Lei, *Strang-type Preconditioners for Solving Linear Systems form Delay Differential Equations*, BIT, vol. 43 (2003), pp. 136–149.

[31] T. Mori, N. Fukuma and M. Kuwahara, *Simple Stability Criteria for Single and Composite Linear Systems with Time Delays*, Int. J. Control, vol. 34 (1981), pp. 1175–1184.

[32] T. Mori, E. Noldus, and M. Kuwahara, *A Way to Stabilize Linear Systems with Delayed State*, Automatica, vol. 19 (1983), pp. 571–573.

[33] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

[34] Y. Saad and M. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput., vol. 7 (1986), pp. 856–869.

[35] G. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, 1990.

[36] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1992.

[37] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Stud. Appl. Math., vol. 74 (1986), pp. 171–176.

[38] L. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM Press, Philadelphia, 1997.

[39] E. Trytyshnikov, *Optimal and Super-Optimal Circulant Preconditioners*, SIAM J. Matrix Anal. Appl., vol. 13 (1992), pp. 459–473.

[40] S. Vandewalle and R. Piessens, *On Dynamic Iteration Methods for Solving Time-Periodic Differential Equations*, SIAM J. Num. Anal., vol. 30 (1993), pp. 286–303.

[41] R. Varga, *Matrix Iterative Analysis*, 2nd edition, Springer-Verlag, Berlin, 2000.

[42] G. Wang, Y. Wei and S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, Beijing, 2004.

[43] S. Wang, *Further Results on Stability of* $\dot{X}(t) = AX(t) + BX(t-\tau)$, Syst. Cont. Letters, vol. 19 (1992), pp. 165–168.

[44] Y. Wei, J. Cai and M. Ng, *Computing Moore-Penrose Inverses of Toeplitz Matrices by Newton's Iteration*, Math. Comput. Modelling, vol. 40 (2004), pp. 181–191.

[45] Y. Wei and N. Zhang, *Condition Number Related with Generalized Inverse* $A^{(2)}_{T,S}$ *and Constrained Linear Systems*, J. Comput. Appl. Math., vol. 157 (2003), pp. 57–72.

[46] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[47] S. Xu, L. Gao and P. Zhang, *Numerical Linear Algebra* (in Chinese), Peking University Press, Beijing, 2000.

[48] N. Zhang and Y. Wei, *Solving EP Singular Linear Systems*, Int. J. Computer Mathematics, vol. 81 (2004), pp. 1395–1405.

Index

(O-2043.0103)

Series in Information and Computational Science

Numerical linear algebra, also called matrix computation, has been a center of scientific and engineering computing since 1946. Most of problems in science and engineering finally become problems in matrix computation. This book gives an elementary introduction to matrix computation and it also includes some new results obtained in recent years.

This book consists of nine chapters. It includes Gaussian elimination, classical iterative methods and Krylov subspace methods for solving linear systems; the rounding error analysis of Gaussian elimination; the perturbation analysis of linear systems; orthogonal decompositions for solving linear least squares problems; and some classical methods for eigen-problems. In the last chapter, a brief survey of the latest developments in using boundary value methods for solving initial value problems of ordinary differential equations is given.

This is a textbook for the senior students majoring in scientific computing and information science. It will also be useful to all who teach or study the subject.

Xiao-qing Jin is a professor at the Department of Mathematics, University of Macau. He is the author of *Developments and Applications of Block Toeplitz Iterative Solvers* (2002) and more than 60 research publications. He is also an editor of *Journal on Numerical Methods and Computer Applications*.

Yi-min Wei is an associate professor at the Department of Mathematics, Fudan University. He is the author of more than 90 research publications and the co-author of *Generalized Inverses: Theory and Computations* (2004). He is also a reviewer of *Mathematical Review*.

ISBN 7-03-013954-2
定 价：56.00 元