Fundamentals *of* Algorithms

An Introduction to Iterative Toeplitz Solvers

Raymond Hon-Fu Chan
Xiao-Qing Jin

siam

An Introduction to Iterative Toeplitz Solvers

Fundamentals of Algorithms

Editor-in-Chief: Nicholas J. Higham, University of Manchester

The SIAM series on Fundamentals of Algorithms is a collection of short user-oriented books on state-of-the-art numerical methods. Written by experts, the books provide readers with sufficient knowledge to choose an appropriate method for an application and to understand the method's strengths and limitations. The books cover a range of topics drawn from numerical analysis and scientific computing. The intended audiences are researchers and practitioners using the methods and upper level undergraduates in mathematics, engineering, and computational science.

Books in this series not only provide the mathematical background for a method or class of methods used in solving a specific problem but also explain how the method can be developed into an algorithm and translated into software. The books describe the range of applicability of a method and give guidance on troubleshooting solvers and interpreting results. The theory is presented at a level accessible to the practitioner. MATLAB® software is the preferred language for codes presented since it can be used across a wide variety of platforms and is an excellent environment for prototyping, testing, and problem solving.

The series is intended to provide guides to numerical algorithms that are readily accessible, contain practical advice not easily found elsewhere, and include understandable codes that implement the algorithms.

Editorial Board

Series Volumes

Chan, R. H.-F. and Jin, X.-Q., *An Introduction to Iterative Toeplitz Solvers*
Eldén, L., *Matrix Methods in Data Mining and Pattern Recognition*
Hansen, P. C., Nagy, J. G., and O'Leary, D. P., *Deblurring Images: Matrices, Spectra, and Filtering*
Davis, T. A., *Direct Methods for Sparse Linear Systems*
Kelley, C. T., *Solving Nonlinear Equations with Newton's Method*

Raymond Hon-Fu Chan
Chinese University of Hong Kong
Shatin, Hong Kong

Xiao-Qing Jin
University of Macau
Taipa, Macao

An Introduction to Iterative Toeplitz Solvers

siam®
Society for Industrial and Applied Mathematics
Philadelphia

10 9 8 7 6 5 4 3 2 1

The research results included in the book are supported by a number of grants provided by HKRGC of Hong Kong and research grant 050/2005/A provided by FDCT of Macao.

Library of Congress Cataloging-in-Publication Data

Chan, Raymond H., 1958-
An introduction to iterative Toeplitz solvers / Raymond Hon-Fu Chan, Xiao-Qing Jin.
p. cm. — (Fundamentals of algorithms)
Includes bibliographical references and index.
ISBN 978-0-898716-36-8 (alk. paper)
1. Iterative methods (Mathematics). 2. Toeplitz matrices. 3. Toeplitz operators. I. Jin, Xiao-Qing. II. Title.

QA297.8.C43 2007
518'.26—dc22

2007061742

To our families

Contents

Preface

In this book, we introduce current developments and applications in using iterative methods for solving Toeplitz systems. Toeplitz systems arise in a variety of applications in mathematics, scientific computing, and engineering, for instance, numerical partial and ordinary differential equations; numerical solution of convolution-type integral equations; stationary autoregressive time series in statistics; minimal realization problems in control theory; system identification problems in signal processing and image restoration problems in image processing; see [24, 36, 45, 55, 66].

In 1986, Strang [74] and Olkin [67] proposed independently the use of the preconditioned conjugate gradient (PCG) method with circulant matrices as preconditioners to solve Toeplitz systems. One of the main results of this iterative solver is that the complexity of solving a large class of n-by-n Toeplitz systems $T_n\mathbf{u} = \mathbf{b}$ is only $O(n \log n)$ operations. Since then, iterative Toeplitz solvers have garnered much attention and evolved rapidly over the last two decades.

This book is intended to be a short and quick guide to the development of iterative Toeplitz solvers based on the PCG method. Within limited space and time, we are forced to deal with only important aspects of iterative Toeplitz solvers and give special attention to the construction of efficient circulant preconditioners. Applications of iterative Toeplitz solvers to some practical problems will be briefly discussed. We wish that after reading the book, the readers can use our methods and algorithms to solve their own problems easily.

The book is organized into five chapters. In Chapter 1, we first introduce Toeplitz systems and discuss their applications. We give a brief survey of classical (direct) Toeplitz solvers. Some background knowledge of matrix analysis that will be used throughout the book is provided. A preparation for current developments in using the PCG method to solve Toeplitz systems is also given.

In Chapter 2, we study some well-known circulant preconditioners which have proven to be efficient for solving some well-conditioned Hermitian Toeplitz systems. We introduce the construction of Strang's preconditioner, T. Chan's preconditioner, and the superoptimal preconditioner. A detailed analysis of the convergence rate of the PCG method and some numerical tests with these three preconditioners are also given. Other useful preconditioners will be briefly introduced.

Chapter 3 develops a unified treatment of different circulant preconditioners. We consider circulant preconditioners for Hermitian Toeplitz systems from the viewpoint of function theory. Some well-known circulant preconditioners can be derived from convoluting the generating function of the Toeplitz matrix with some famous kernels. Several new circulant preconditioners are then constructed using this approach. An analysis of convergence rate is given with some numerical examples.

Chapter 4 describes how a family of efficient circulant preconditioners can be constructed for ill-conditioned Hermitian Toeplitz systems $T_n\mathbf{u} = \mathbf{b}$. Inspired by the unified theory developed in Chapter 3, the preconditioners are constructed by convoluting the generating function of T_n with the generalized Jackson kernels. When the generating function is nonnegative continuous with a zero of order $2p$, the condition number of T_n is known to grow as $O(n^{2p})$. We show, however, that the preconditioner is positive definite and the spectrum of the preconditioned matrix is uniformly bounded except for at most $2p + 1$ outliers. Moreover, the smallest eigenvalue is uniformly bounded away from zero. Hence the PCG method converges linearly when used to solve the system. Numerical examples are included to illustrate the effectiveness of the preconditioners.

Chapter 5 is devoted to the study of block circulant preconditioners for the solution of block systems $T_{mn}\mathbf{u} = \mathbf{b}$ by the PCG method, where T_{mn} are m-by-m block Toeplitz matrices with n-by-n Toeplitz blocks. Such a kind of system appears in many applications, especially in image processing [45, 66]. The preconditioners $c_F^{(1)}(T_{mn})$ and $c_{F,F}^{(2)}(T_{mn})$ are constructed to preserve the block structure of T_{mn} and are defined to be the minimizers of $\|T_{mn} - C_{mn}\|_{\mathscr{F}}$ over some special classes of matrices. We prove that if T_{mn} is positive definite, then $c_F^{(1)}(T_{mn})$ and $c_{F,F}^{(2)}(T_{mn})$ are positive definite too. We illustrate their effectiveness for solving block Toeplitz systems by some numerical examples.

To facilitate the use of the methods discussed in this book, we have included in the appendix the MATLAB programs that were used to generate our numerical results here.

This book contains some important parts of our research work in the past 20 years. Some research results are joint work with Prof. Michael K. Ng of the Department of Mathematics, Hong Kong Baptist University; Prof. Man-Chung Yeung of the Department of Mathematics, University of Wyoming; and Dr. Andy M. Yip of the Department of Mathematics, National University of Singapore. We are indebted to Prof. Tony F. Chan of the Department of Mathematics, University of California at Los Angeles; Prof. Gene Golub of the Department of Computer Science, Stanford University; Prof. Robert Plemmons of the Department of Mathematics and Computer Science, Wake Forest University; and Prof. Gilbert Strang of the Department of Mathematics, Massachusetts Institute of Technology, for their enlightening ideas, suggestions, and comments, from which we benefited a great deal. We are grateful to Prof. Fu-Rong Lin of the Department of Mathematics, Shantou University; and our student Mr. Wei Wang, for their help in the MATLAB programs used in the book. Thanks are also due to our families for their encouragement, great support, and patience, which are essential to the completion of the book.

The research results included in the book are supported by a number of grants provided by HKRGC of Hong Kong and research grant 050/2005/A provided by FDCT of Macao.

Chapter 1

Introduction

In this chapter, we first introduce the Toeplitz system, its history, and some remarks on classical (direct) Toeplitz solvers. We then develop the background knowledge in matrix analysis that will be used throughout the book. A preparation for the development of iterative Toeplitz solvers is also given.

1.1 Toeplitz systems

An n-by-n Toeplitz matrix is of the following form:

$$T_n = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \cdots & t_{2-n} \\ \vdots & t_1 & t_0 & \ddots & \vdots \\ t_{n-2} & \cdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{pmatrix}; \qquad (1.1)$$

i.e., $t_{ij} = t_{i-j}$ and T_n is constant along its diagonals. The name Toeplitz is in memorial of O. Toeplitz's early work [78] in 1911 on bilinear forms related to Laurent series; see [43] for details. We are interested in solving the Toeplitz system

$$T_n\mathbf{u} = \mathbf{b},$$

where $\mathbf{b}$ is a known vector and $\mathbf{u}$ is an unknown vector.

Toeplitz systems arise in a variety of applications in different fields of mathematics, scientific computing, and engineering [15, 24, 36, 45, 55, 57, 66]:

(1) Numerical partial and ordinary differential equations.

(2) Numerical solution of convolution-type integral equations.

(3) Statistics—stationary autoregressive time series.

(4) Signal processing—system identification and recursive filtering.

(5) Image processing—image restoration.

(6) Padé approximation—computation of coefficients.

(7) Control theory—minimal realization and minimal design problems.

(8) Networks—stochastic automata and neutral networks.

These applications have motivated mathematicians, scientists, and engineers to develop specifically fast algorithms for solving Toeplitz systems. Such kinds of algorithms are called Toeplitz solvers.

Most of the early works on Toeplitz solvers were focused on direct methods. A straightforward application of the Gaussian elimination method will result in an algorithm of $O(n^3)$ complexity. However, since n-by-n Toeplitz matrices are determined by only $2n-1$ entries rather than n^2 entries, it is expected that the solution of Toeplitz systems can be obtained in less than $O(n^3)$ operations. Levinson's algorithm [62] proposed in 1946 is the first algorithm which reduces the complexity to $O(n^2)$ operations. A number of algorithms with such complexity can be found in the literature; see, for instance, [46, 81, 88]. These algorithms require the invertibility of the $(n-1)$-by-$(n-1)$ principal submatrix of T_n.

Around 1980, fast direct Toeplitz solvers of complexity $O(n \log^2 n)$ were developed [2, 11, 13, 50]. These algorithms require the invertibility of the $\lfloor n/2 \rfloor$-by-$\lfloor n/2 \rfloor$ principal submatrix of T_n.

The stability properties of these direct methods for symmetric positive definite Toeplitz systems are discussed in Bunch [15]. It was noted that if T_n has a singular or ill-conditioned principal submatrix, then a breakdown (or near-breakdown) can occur in these algorithms. Such breakdowns will cause numerical instabilities in subsequent steps and result in inaccurate solutions.

The question of how to avoid breakdowns (or near-breakdowns) by skipping over singular submatrices or ill-conditioned submatrices has been studied extensively [38, 42, 47, 77, 87]. In particular, T. Chan and Hansen in 1992 derived the look-ahead Levinson algorithm [34]. The basic idea of the algorithm is to relax the inverse triangular decomposition slightly and to compute an inverse block factorization of the Toeplitz matrices with a block diagonal matrix instead of a scalar diagonal matrix. Other look-ahead extensions of fast Toeplitz solvers can be found in [39, 40].

Strang [74] and Olkin [67] in 1986 proposed independently the use of the preconditioned conjugate gradient (PCG) method with circulant matrices as preconditioners to solve symmetric positive definite Toeplitz systems. One of the main results of this iterative solver is that the complexity of solving a large class of n-by-n Toeplitz systems $T_n\mathbf{u} = \mathbf{b}$ is only $O(n \log n)$ operations. Since then, iterative Toeplitz solvers based on the PCG method have evolved rapidly with many papers appearing on the subject, especially in the past 10 years. It is difficult for us to include all of the different developments into this book. After compromising, we will deal only with important aspects of these iterative Toeplitz solvers and give some insight into how to design efficient preconditioners. Applications of iterative Toeplitz solvers to some practical problems will be briefly mentioned. We hope that the selection of topics and the style of presentation will be useful to anyone interested in iterative Toeplitz solvers. In the following, before we begin to study these iterative solvers in detail, we need to introduce some background knowledge in matrix analysis which will be used throughout the book.

1.2 Background in matrix analysis

An overview of the relevant concepts in matrix analysis is given here. The material will be helpful in developing our theory in later chapters.

1.2.1 Basic symbols

We will be using the following symbols throughout this book.

- Let $\mathbb{R}$ denote the set of real numbers and $\mathbb{C}$ the set of complex numbers, and let $\mathbf{i} \equiv \sqrt{-1}$.
- Let $\mathbb{R}^n$ denote the set of real n-vectors and $\mathbb{C}^n$ the set of complex n-vectors. Vectors will always be column vectors.
- Let $\mathbb{R}^{n\times n}$ denote the linear vector space of n-by-n real matrices and $\mathbb{C}^{n\times n}$ the linear vector space of n-by-n complex matrices.
- The uppercase letters such as A, B, C, Δ, and Λ denote matrices, and the boldface lowercase letters such as $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ denote vectors.
- The symbol $(A)_{ij} = a_{ij}$ denotes the (i,j)th entry of a matrix A. For an n-by-n matrix, the indexes i, j usually go from 1 to n, but sometimes they go from 0 to $n-1$ for convenience.
- Let A_{mn} denote any m-by-m block matrix with n-by-n blocks and $(A_{mn})_{i,j;k,l}$ denote the (i,j)th entry in the (k,l)th block of matrix A_{mn}.
- The symbol A^T denotes the transpose of a matrix A, and A^* denotes the conjugate transpose of A.
- Let $\operatorname{rank}(A)$ denote the rank of a matrix A.
- Let $\dim(\mathscr{X})$ denote the dimension of a vector space $\mathscr{X}$.
- We use $\operatorname{diag}(a_{11},\ldots,a_{nn})$ to denote the n-by-n diagonal matrix:

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

- The symbol I_n denotes the n-by-n identity matrix:

$$I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

  When there is no ambiguity, we shall write it as I. The symbol $\mathbf{e}_i$ will denote the ith unit vector, i.e., the ith column vector of I.

- Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the smallest and largest eigenvalues of A, respectively.

- We use $\rho[A] \equiv \max|\lambda_i(A)|$ to denote the spectral radius of A, where λ_i goes through the spectrum of A, i.e., the set of all eigenvalues of A.

- Let $\sigma_{\max}(A)$ denote the largest singular value of A.

- Let $\mathbf{C}_{2\pi}$ be the space of all 2π-periodic continuous real-valued functions $f(x)$ and $\mathbf{C}_{2\pi}^{+}$ denote the subspace of all functions $f(x) \geq 0$ in $\mathbf{C}_{2\pi}$ which are not identically zero.

- Let $\mathbf{C}_{2\pi\times 2\pi}$ denote the space of all 2π-periodic (in each direction) continuous real-valued functions $f(x, y)$.

1.2.2 Spectral properties of Hermitian matrix

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be Hermitian if $A^* = A$, and a matrix $A \in \mathbb{R}^{n\times n}$ is said to be symmetric if $A^T = A$. Hermitian and symmetric matrices have many elegant and important spectral properties (see [51, 56, 80, 86]), and here we present only several classical results that will be used later on.

Theorem 1.1 (spectral theorem). *Let $A \in \mathbb{C}^{n\times n}$. Then A is Hermitian if and only if there exist a unitary matrix $U \in \mathbb{C}^{n\times n}$ and a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ such that $A = U\Lambda U^*$.*

We recall that a matrix $M \in \mathbb{C}^{n\times n}$ is unitary if $M^{-1} = M^*$.

Theorem 1.2 (Courant–Fischer minimax theorem). *If $A \in \mathbb{C}^{n\times n}$ is Hermitian with eigenvalues*

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n,$$

then for each $k = 1, 2, \ldots, n$, we have

$$\lambda_k = \min_{\dim(\mathscr{X})=k} \max_{\mathbf{0}\neq\mathbf{x}\in\mathscr{X}} \frac{\mathbf{x}^*A\mathbf{x}}{\mathbf{x}^*\mathbf{x}} = \max_{\dim(\mathscr{X})=n-k+1} \min_{\mathbf{0}\neq\mathbf{x}\in\mathscr{X}} \frac{\mathbf{x}^*A\mathbf{x}}{\mathbf{x}^*\mathbf{x}},$$

where $\mathscr{X}$ denotes a subspace of $\mathbb{C}^n$. In particular, for the smallest and largest eigenvalues, we have

$$\lambda_{\min} = \lambda_1 = \min_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^* A\mathbf{x}}{\mathbf{x}^*\mathbf{x}}, \qquad \lambda_{\max} = \lambda_n = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^* A\mathbf{x}}{\mathbf{x}^*\mathbf{x}}.$$

We recall that a number $\lambda \in \mathbb{C}$ is called an eigenvalue of A if there exists a nonzero vector $\mathbf{x} \in \mathbb{C}^n$ such that

$$A\mathbf{x} = \lambda\mathbf{x}.$$

Here $\mathbf{x}$ is called the eigenvector of A associated with λ.

Theorem 1.3 (Cauchy's interlace theorem). *Suppose $A \in \mathbb{C}^{n\times n}$ is Hermitian and that its eigenvalues are arranged in an increasing order*

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

If

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_{n-1}$$

are the eigenvalues of a principal submatrix of A of order $n-1$, then

$$\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \mu_2 \leq \cdots \leq \mu_{n-1} \leq \lambda_n.$$

Theorem 1.4. (Weyl's theorem). *Let A, $E \in \mathbb{C}^{n\times n}$ be Hermitian and the eigenvalues $\lambda_i(A)$, $\lambda_i(E)$, $\lambda_i(A+E)$ be arranged in an increasing order. Then for each $k = 1, 2, \ldots, n$, we have*

$$\lambda_k(A) + \lambda_1(E) \leq \lambda_k(A+E) \leq \lambda_k(A) + \lambda_n(E).$$

Theorem 1.5 (singular value decomposition theorem). *Let $A \in \mathbb{C}^{m\times n}$ with* $\operatorname{rank}(A) = r$. *Then there exist unitary matrices*

$$U = (\mathbf{u}_1, \ldots, \mathbf{u}_m) \in \mathbb{C}^{m\times m}, \qquad V = (\mathbf{v}_1, \ldots, \mathbf{v}_n) \in \mathbb{C}^{n\times n}$$

such that

$$U^* A V = \begin{pmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where

$$\Sigma_r = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

The σ_i, $i = 1, 2, \ldots, r$, are called the singular values of A. The vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are called the ith left singular vector and the ith right singular vector, respectively.

1.2.3 Norms and condition number

Let

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{C}^n.$$

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

(i) $\|\mathbf{x}\| > 0$ if and only if $\mathbf{x} \neq \mathbf{0}$;

(ii) $\|\alpha \mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$;

(iii) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.

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

$$\|\mathbf{x}\|_p \equiv \Big(\sum_{i=1}^{n} |x_i|^p \Big)^{1/p}.$$

The following p-norms are the most commonly used norms in practice:

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

A very important property of vector norms on $\mathbb{C}^n$ is that all vector norms on $\mathbb{C}^n$ are equivalent; i.e., if $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are two norms on $\mathbb{C}^n$, then there exist two positive constants c_1 and c_2 such that

$$c_1 \|\mathbf{x}\|_\alpha \leq \|\mathbf{x}\|_\beta \leq c_2 \|\mathbf{x}\|_\alpha$$

for all $\mathbf{x} \in \mathbb{C}^n$. For instance, we have

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2, \quad \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty, \quad \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_1 \leq n\|\mathbf{x}\|_\infty.$$

Let

$$A = (a_{ij})_{i,j=1}^{n} \in \mathbb{C}^{n \times n}.$$

We now turn our attention to matrix norms. A matrix (consistent) norm is a function that assigns to each $A \in \mathbb{C}^{n \times n}$ a nonnegative number $\|A\|$, called the norm of A, such that the following four properties are satisfied for all $A, B \in \mathbb{C}^{n \times n}$ and all $\alpha \in \mathbb{C}$:

(i) $\|A\| > 0$ if and only if $A \neq \mathbf{0}$;

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

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

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

For every vector norm, we can define a matrix norm in a natural way. Given a vector norm $\|\cdot\|_v$, the matrix norm induced by $\|\cdot\|_v$ is defined by

$$\|A\|_v \equiv \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|A\mathbf{x}\|_v}{\|\mathbf{x}\|_v}.$$

The most important matrix norms are the matrix p-norms induced by the vector p-norms for $p = 1, 2, \infty$. One can show that [41, 56]

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|, \quad \|A\|_2 = \sigma_{\max}(A), \quad \|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|,$$

where $\sigma_{\max}(A)$ denotes the largest singular value of A. The Frobenius norm is defined by

$$\|A\|_{\mathscr{F}} \equiv \Big(\sum_{j=1}^{n} \sum_{i=1}^{n} |a_{ij}|^2 \Big)^{1/2}.$$

One of the most important properties of $\|\cdot\|_2$ and $\|\cdot\|_{\mathscr{F}}$ is that for any unitary matrices Q and Z,

$$\|A\|_2 = \|QAZ\|_2, \qquad \|A\|_{\mathscr{F}} = \|QAZ\|_{\mathscr{F}}.$$

When we solve a linear system $A\mathbf{u} = \mathbf{b}$, we need a good measurement on how sensitive the computed solution is to input perturbations. The condition number of matrices, defined by using matrix norms, relates the accuracy in $\mathbf{u}$ to perturbations in A and $\mathbf{b}$.

Definition 1.6. *Let $\|\cdot\|$ be any matrix norm and A be an invertible matrix. The condition number of A is defined as follows:*

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

Obviously, the condition number depends on the matrix norm, and usually $\|\cdot\|_2$ is used. 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. The following theorem concerns the effect of the perturbations in A and $\mathbf{b}$ on the solution of $A\mathbf{u} = \mathbf{b}$ in terms of the condition number. We refer readers to [41, 56] for a proof.

Theorem 1.7. *Let A be an invertible matrix and $\hat{A}$ be a perturbed matrix of A such that*

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

If

$$A\mathbf{u} = \mathbf{b}, \qquad \hat{A}\hat{\mathbf{u}} = \hat{\mathbf{b}},$$

where $\hat{\mathbf{b}}$ is a perturbed vector of $\mathbf{b}$, then

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

Theorem 1.7 gives the upper bounds for the relative error of $\mathbf{u}$ in terms of the condition number of A. From the theorem, we know that if A is well-conditioned, i.e., $\kappa(A)$ is small, then the relative error in $\mathbf{u}$ will be small, provided that the relative errors in A and $\mathbf{b}$ are both small.

1.3 Preparation for iterative Toeplitz solvers

In 1986, Strang [74] and Olkin [67] proposed independently the use of the PCG method with circulant preconditioners to solve symmetric positive definite Toeplitz systems. We first introduce the conjugate gradient (CG) method.

1.3.1 Conjugate gradient method

The scheme of the CG method, one of the most popular and successful iterative methods for solving Hermitian positive definite systems $H_n\mathbf{u} = \mathbf{b}$, is given as follows; see [3, 41, 56, 69]. At the initialization step, we choose $\mathbf{u}^{(0)}$, calculate

$$\mathbf{r}^{(0)} = \mathbf{b} - H_n\mathbf{u}^{(0)},$$

and put $\mathbf{d}^{(0)} = \mathbf{r}^{(0)}$. In the iteration steps, we have

$$\begin{cases} \mathbf{s}^{(k)} = H_n\mathbf{d}^{(k)}, \\ \tau_k := \dfrac{\mathbf{r}^{(k)^*}\mathbf{r}^{(k)}}{\mathbf{d}^{(k)^*}\mathbf{s}^{(k)}}, \\ \mathbf{u}^{(k+1)} := \mathbf{u}^{(k)} + \tau_k\mathbf{d}^{(k)}, \\ \mathbf{r}^{(k+1)} := \mathbf{r}^{(k)} - \tau_k\mathbf{s}^{(k)}, \\ \beta_k := \dfrac{\mathbf{r}^{(k+1)^*}\mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)^*}\mathbf{r}^{(k)}}, \\ \mathbf{d}^{(k+1)} := \mathbf{r}^{(k+1)} + \beta_k\mathbf{d}^{(k)}, \end{cases}$$

where $\mathbf{d}^{(k)}$, $\mathbf{r}^{(k)}$ are vectors and τ_k, β_k are scalars, $k = 0, 1, \ldots$. The vector $\mathbf{u}^{(k)}$ is the approximation to the true solution after the kth iteration. The main operation cost is the matrix-vector multiplication $H_n\mathbf{d}^{(k)}$, which usually needs $O(n^2)$ operations. A MATLAB implementation of the CG method is given as A.12 in the appendix.

For Hermitian matrices, by the spectral theorem we know that the spectra of the matrices are real. Therefore, in order to analyze the convergence rate of the CG method, we need to introduce the following definition of clustered spectrum on the real line [24, 55, 66, 83].

Definition 1.8. *A sequence of matrices $\{H_n\}_{n=1}^{\infty}$ is said to have clustered spectra around* 1 *if for any $\epsilon > 0$, there exist M and $N > 0$ such that for any $n > N$, at most M eigenvalues of $H_n - I_n$ have absolute values larger than ϵ; see Figure* 1.1. *Here I_n is the identity matrix.*

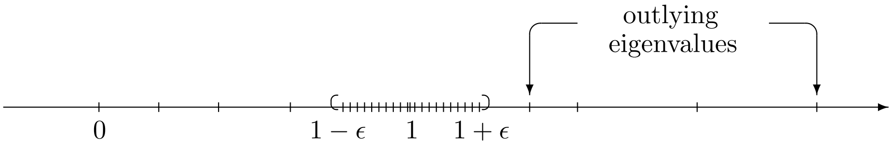

Figure 1.1. *Clustered spectra around* 1.

In the following, we study the convergence rate of the CG method. Let

$$\mathbf{e}^{(k)} \equiv \mathbf{u} - \mathbf{u}^{(k)}, \tag{1.2}$$

where $\mathbf{u}^{(k)}$ is the approximation after the kth iteration of the CG method applied to the system $H_n\mathbf{u} = \mathbf{b}$ and $\mathbf{u}$ is the true solution of the system. The following theorem can be found in [41, 56].

Theorem 1.9. *We have*

$$\frac{|||\mathbf{e}^{(k)}|||}{|||\mathbf{e}^{(0)}|||} \le 2\left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)^k,$$

where $\mathbf{e}^{(k)}$ *is defined by* (1.2), $|||\cdot|||$ *is the energy norm defined by* $|||\mathbf{v}|||^2 \equiv \mathbf{v}^*H_n\mathbf{v}$, *and*

$$\kappa_2 = \kappa_2(H_n) = \|H_n\|_2\|H_n^{-1}\|_2.$$

Furthermore, we have the following theorem [84].

Theorem 1.10. *If the eigenvalues* λ_j *of* H_n *are ordered such that*

$$0 < \lambda_1 \le \cdots \le \lambda_p \le b_1 \le \lambda_{p+1} \le \cdots \le \lambda_{n-q} \le b_2 \le \lambda_{n-q+1} \le \cdots \le \lambda_n,$$

where b_1 *and* b_2 *are two constants, then*

$$\frac{|||\mathbf{e}^{(k)}|||}{|||\mathbf{e}^{(0)}|||} \le 2\left(\frac{\alpha-1}{\alpha+1}\right)^{k-p-q} \cdot \max_{\lambda\in[b_1,b_2]} \prod_{j=1}^{p}\left(\frac{\lambda-\lambda_j}{\lambda_j}\right) = 2\left(\frac{\alpha-1}{\alpha+1}\right)^{k-p-q} \cdot \prod_{j=1}^{p}\left(\frac{b_2-\lambda_j}{\lambda_j}\right),$$

where $\mathbf{e}^{(k)}$ *is defined by* (1.2) *and* $\alpha \equiv (b_2/b_1)^{1/2} \ge 1$.

From Theorem 1.10, we notice that when n increases, if p and q are constants that do not depend on n and λ_1 is uniformly bounded away from zero, then the convergence rate is linear, i.e.,

$$\lim_{k\to\infty}\frac{|||\mathbf{e}^{(k+1)}|||}{|||\mathbf{e}^{(k)}|||} = c < 1.$$

Thus, the number of iterations to attain a given accuracy is independent of n. We also notice that the more clustered the eigenvalues are, the faster the convergence rate will be. In particular, if they are clustered around 1, we have the following corollary.

Corollary 1.11. *If the eigenvalues λ_j of H_n are ordered such that*

$$0 < \eta < \lambda_1 \le \cdots \le \lambda_p \le 1-\epsilon \le \lambda_{p+1} \le \cdots \le \lambda_{n-q} \le 1+\epsilon \le \lambda_{n-q+1} \le \cdots \le \lambda_n,$$

where $0 < \epsilon < 1$*, then*

$$\frac{|||\mathbf{e}^{(k)}|||}{|||\mathbf{e}^{(0)}|||} \le 2\left(\frac{1+\epsilon}{\eta}\right)^p \epsilon^{k-p-q},$$

where $\mathbf{e}^{(k)}$ *is defined by* (1.2) *and* $k \ge p+q$*.*

Proof. For α given in Theorem 1.10, 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

$$0 \le \frac{\lambda-\lambda_j}{\lambda_j} \le \frac{1+\epsilon}{\eta}.$$

Thus, by using Theorem 1.10, we obtain

$$\begin{aligned}\frac{|||\mathbf{e}^{(k)}|||}{|||\mathbf{e}^{(0)}|||} &\le 2\left(\frac{\alpha-1}{\alpha+1}\right)^{k-p-q} \cdot \max_{\lambda\in[b_1,b_2]} \prod_{j=1}^{p}\left(\frac{\lambda-\lambda_j}{\lambda_j}\right) \\ &\le 2\left(\frac{1+\epsilon}{\eta}\right)^p \epsilon^{k-p-q}. \qquad \square\end{aligned}$$

For arbitrary small $\epsilon > 0$, if p and q are constants that do not depend on n when n increases and λ_1 is uniformly bounded away from zero by η, then we have by Corollary 1.11, $|||\mathbf{e}^{(k+1)}|||/|||\mathbf{e}^{(k)}||| \le \epsilon$, i.e.,

$$\lim_{k\to\infty} \frac{|||\mathbf{e}^{(k+1)}|||}{|||\mathbf{e}^{(k)}|||} = 0.$$

Thus, the convergence rate is superlinear. Note that for any $\epsilon > 0$, if there exist M and $N > 0$ such that when $n > N$, the matrix H_n can be decomposed as

$$H_n = I_n + K_n + L_n, \tag{1.3}$$

where $\|K_n\|_2 < \epsilon$, $\text{rank}(L_n) < M$, and, moreover, $\lambda_{\min}(H_n)$ is uniformly bounded away from zero, then by Weyl's theorem, the spectrum of H_n will satisfy the conditions of Corollary 1.11. In later chapters, we will use this technique of decomposition (1.3) to establish the superlinear convergence rate of our methods.

When H_n is not of the form in (1.3), we can accelerate the convergence rate of the CG method by preconditioning the system; i.e., instead of solving the original system $H_n\mathbf{u} = \mathbf{b}$, we solve the preconditioned system

$$\widetilde{H}_n \tilde{\mathbf{u}} = \tilde{\mathbf{b}},$$

where $\widetilde{H}_n = M_n^{-\frac{1}{2}} H_n M_n^{-\frac{1}{2}}$, $\tilde{\mathbf{u}} = M_n^{\frac{1}{2}}\mathbf{u}$, $\tilde{\mathbf{b}} = M_n^{-\frac{1}{2}}\mathbf{b}$, and M_n is Hermitian positive definite. The main work involved in implementing the CG method for the preconditioned system $\widetilde{H}_n\tilde{\mathbf{u}} = \tilde{\mathbf{b}}$ is the matrix-vector product $M_n^{-1}H_n\mathbf{v}$ for some vector $\mathbf{v}$; see [41, 56, 69]. The resulting method is called the preconditioned conjugate gradient (PCG) method. The preconditioner M_n is chosen with three criteria in mind [3, 41, 55, 56, 66]:

I. M_n is easily constructible.

II. For any vector $\mathbf{d}$, the product $\mathbf{r} = M_n^{-1}\mathbf{d}$ is easy to compute or the system $M_n\mathbf{r} = \mathbf{d}$ is easy to solve.

III. The spectrum of $\widetilde{H}_n$, which is the same as that of $M_n^{-1}H_n$, is clustered and/or $\widetilde{H}_n$ is well-conditioned compared to H_n.

Strang [74] and Olkin [67] noted that for any Toeplitz matrix T_n with a circulant preconditioner C_n, the product $C_n^{-1}T_n\mathbf{v}$ can be computed efficiently in $O(n\log n)$ operations. Circulant matrices are Toeplitz matrices of the form

$$C_n = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \cdots & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix},$$

i.e., $c_{-k} = c_{n-k}$ for $1 \le k \le n-1$. It is well-known [37] that circulant matrices can be diagonalized by the Fourier matrix F_n, i.e.,

$$C_n = F_n^* \Lambda_n F_n, \tag{1.4}$$

where the entries of F_n are given by

$$(F_n)_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi \mathbf{i} jk}{n}}, \qquad \mathbf{i} \equiv \sqrt{-1},$$

for $0 \le j,k \le n-1$, and Λ_n is a diagonal matrix holding the eigenvalues of C_n.

By using (1.4), we note that the first column of F_n is $\frac{1}{\sqrt{n}}\mathbf{1}_n$, where $\mathbf{1}_n = (1,1,\ldots,1)^T \in \mathbb{R}^n$ is the vector of all ones. Hence

$$F_n C_n \mathbf{e}_1 = \frac{1}{\sqrt{n}} \Lambda_n \mathbf{1}_n, \tag{1.5}$$

where $\mathbf{e}_1 = (1,0,\ldots,0)^T \in \mathbb{R}^n$ is the first unit vector. Therefore, the entries of Λ_n can be obtained in $O(n\log n)$ operations by taking the celebrated fast Fourier transform (FFT) of the first column of C_n; see [14, p. 131] for a detailed introduction on FFT. From (1.5), we see that the eigenvalues λ_k of C_n are given by

$$\lambda_k = (\Lambda_n)_{kk} = \sum_{j=0}^{n-1} c_j e^{\frac{2\pi \mathbf{i} jk}{n}}, \qquad k = 0, 1, \ldots, n-1. \tag{1.6}$$

In MATLAB software, the command `fft(v)` computes the product $\sqrt{n} F_n \mathbf{v}$ for any vector $\mathbf{v}$. In view of (1.5), the eigenvalues λ_k can be computed by `fft(c)` when $\mathbf{c}$ is the first column of C_n; see how the vector `ev` is formed in A.3 in the appendix.

Once Λ_n is obtained, the products $C_n\mathbf{y}$ and $C_n^{-1}\mathbf{y}$ could be computed easily by FFTs in $O(n \log n)$ operations by using (1.4). In fact, $C_n^{-1}\mathbf{y} = F_n^* \Lambda_n^{-1} F_n \mathbf{y}$, which can be computed by the MATLAB command `ifft(fft(y)./ev)`; see A.6 in the appendix. We note that in MATLAB software, the operation "./" is the entrywise division.

The multiplication $T_n\mathbf{v}$ can also be computed by FFTs by first embedding T_n into a $2n$-by-$2n$ circulant matrix. More precisely, we construct a $2n$-by-$2n$ circulant matrix with T_n embedded inside as follows:

$$\begin{pmatrix} T_n & \times \\ \times & T_n \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} T_n\mathbf{v} \\ \dagger \end{pmatrix}, \tag{1.7}$$

and then the multiplication can be carried out by using the decomposition as in (1.4). Therefore, the cost of computing $T_n\mathbf{v}$ is $O(2n \log(2n))$ by using FFTs of length $2n$. In A.7, we give the program for computing $T_n\mathbf{v}$, where `gev`, computed via A.3, is the vector holding the eigenvalues of the $2n$-by-$2n$ circulant matrix.

Recall that when using the PCG method to solve $T_n\mathbf{u} = \mathbf{b}$ with the preconditioner C_n, the main areas of work in each iteration are the matrix-vector products $T_n\mathbf{v}$ and $C_n^{-1}\mathbf{w}$ for some vectors $\mathbf{v}$ and $\mathbf{w}$; see [41, 56, 69]. From the discussions above, the cost per iteration is therefore $O(n \log n)$. In particular, if the method converges linearly or superlinearly, then the complexity of the algorithm remains $O(n \log n)$. This is one of the important results of this algorithm when compared to the operation cost of $O(n \log^2 n)$ required by fast direct Toeplitz solvers. The MATLAB implementation of the PCG algorithm is given in A.5. Instead of requiring the matrices C_n and T_n, it requires the eigenvalues of C_n and the eigenvalues of the extended $2n$-by-$2n$ circulant matrix in (1.7). These eigenvalues are computed via A.3. We will illustrate how to use A.5 in Section 2.5.

We emphasize that the use of circulant preconditioners for Toeplitz systems allows the use of FFT throughout the computations, and FFT is highly parallelizable and has been implemented on multiprocessors efficiently [1, p. 238] and [76]. Since the CG method is also easily parallelizable [8, p. 165], the PCG method with circulant preconditioners is well adapted for parallel computing.

1.3.2 Generating function and spectral analysis

We need to introduce the technical term, generating function, in order to give the spectral analysis. From Theorem 1.10 and Corollary 1.11, we know that the convergence rate of the PCG method with circulant preconditioner C_n for solving the Toeplitz system $T_n\mathbf{u} = \mathbf{b}$ depends on the spectrum of $C_n^{-1}T_n$, which is a function of n. To link all T_n's together, we assume that the diagonals $\{t_k\}_{k=-n+1}^{n-1}$ of T_n, defined in (1.1), are the Fourier coefficients of a function f, i.e.,

$$t_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-\mathbf{i}kx}dx. \tag{1.8}$$

The function f is called the generating function of T_n. In some practical problems from science and engineering, we are usually given f first, not the Toeplitz matrices T_n [24, 55, 66]; note the following examples:

- Numerical differential equations: the equation gives f.
- Integral equation: the kernel gives f.
- Time series: the spectral density function gives f.
- Filter design: the transfer function gives f.
- Image restoration: the blurring function gives f.

The function f is assumed to be in a certain class of functions such that all T_n are invertible. We note the following:

(i) When f is real-valued, then T_n are Hermitian for all n.

(ii) When f is real-valued and even, then T_n are real symmetric for all n.

Let $\mathbf{C}_{2\pi}$ be the space of all 2π-periodic continuous real-valued functions defined on $[-\pi, \pi]$. The following theorem [43, pp. 64–65] gives the relation between the values of f and the eigenvalues of T_n.

Theorem 1.12 (Grenander–Szegö theorem). *Let T_n be given by* (1.1) *with a generating function $f \in \mathbf{C}_{2\pi}$. Let $\lambda_{\min}(T_n)$ and $\lambda_{\max}(T_n)$ denote the smallest and largest eigenvalues of T_n, respectively. Then we have*

$$f_{\min} \leq \lambda_{\min}(T_n) \leq \lambda_{\max}(T_n) \leq f_{\max}, \tag{1.9}$$

where $f_{\min}$ and $f_{\max}$ denote the minimum and maximum values of $f(x)$, respectively. In particular, if $f_{\min} > 0$, then T_n is positive definite. Moreover, the eigenvalues $\lambda_j(T_n)$, $j = 0, 1, \ldots, n-1$, are equally distributed as $f(\frac{2\pi j}{n})$, i.e.,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{j=0}^{n-1}\left[g(\lambda_j(T_n)) - g\Big(f\Big(\frac{2\pi j}{n}\Big)\Big)\right] = 0$$

for any $g \in \mathbf{C}_{2\pi}$.

Proof. We prove only (1.9). Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})^T \in \mathbb{C}^n$. Then we have

$$\mathbf{v}^* T_n \mathbf{v} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Big|\sum_{k=0}^{n-1} v_k e^{-\mathbf{i}kx}\Big|^2 f(x)dx. \tag{1.10}$$

Since $f_{\min} \le f(x) \le f_{\max}$ for all x, we have by (1.10),

$$f_{\min} \le \mathbf{v}^* T_n \mathbf{v} \le f_{\max},$$

provided that $\mathbf{v}$ satisfies the condition

$$\mathbf{v}^*\mathbf{v} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Big|\sum_{k=0}^{n-1} v_k e^{-\mathbf{i}kx}\Big|^2 dx = 1.$$

Hence, we have by the Courant–Fischer minimax theorem,

$$f_{\min} \le \lambda_{\min}(T_n) \le \lambda_{\max}(T_n) \le f_{\max}. \qquad \square$$

The equal distribution of eigenvalues of Toeplitz matrices indicates that their eigenvalues will not be clustered in general. To illustrate this, consider the 1-dimensional discrete Laplacian matrix

$$T_n = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

The generating function $f(x)$ is given by

$$f(x) = -\cos(-x) + 2 - \cos x = 4\sin^2\frac{x}{2}.$$

By Theorem 1.12, the spectrum of T_n is distributed as $4\sin^2(\pi j/n)$ for $0 \le j \le n-1$. In fact, the eigenvalues of T_n are given by

$$\lambda_j(T_n) = 4\sin^2\left[\frac{\pi(j+1)}{n+1}\right], \qquad 0 \le j \le n-1.$$

Obviously,

$$\begin{aligned} &\lim_{n\to\infty}\frac{1}{n}\sum_{j=0}^{n-1}\left[\lambda_j(T_n) - f\Big(\frac{2\pi j}{n}\Big)\right] \\ &= \lim_{n\to\infty}\frac{4}{n}\sum_{j=0}^{n-1}\left\{\sin^2\left[\frac{\pi(j+1)}{n+1}\right] - \sin^2\Big(\frac{\pi j}{n}\Big)\right\} = 0. \end{aligned}$$

For $n = 32$, the eigenvalues (tick marks) of T_n are depicted in Figure 1.2.

Let $\mathbf{C}_{2\pi}^{+}$ denote the subspace of all nonnegative functions in $\mathbf{C}_{2\pi}$ which are not identically zero. We say that $x_0 \in [-\pi, \pi]$ is a zero of f of order q if $f(x_0) = 0$ and

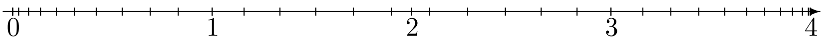

Figure 1.2. *Spectrum of the* 1*-dimensional discrete Laplacian* T_{32}.

q is the smallest positive integer such that $f^{(q)}(x_0) \neq 0$ and $f^{(q+1)}(x)$ is continuous in a neighborhood of x_0. By Taylor's theorem,

$$f(x) = \frac{f^{(q)}(x_0)}{q!}(x - x_0)^q + O\Big((x - x_0)^{q+1}\Big)$$

for all x in that neighborhood. Since f is nonnegative, q must be even and $f^{(q)}(x_0) > 0$. The following theorem is an improvement on Theorem 1.12. The proof of the theorem can be found in [18].

Theorem 1.13. *Let $f \in \mathbf{C}_{2\pi}^{+}$. If $f_{\min} < f_{\max}$, then for all $n > 0$,*

$$f_{\min} < \lambda_j(T_n) < f_{\max}, \qquad j = 1, \ldots, n,$$

where $\lambda_j(T_n)$ is the jth eigenvalue of T_n. In particular, if $f \geq 0$ and $f(x_0) = 0$ with $x_0 \in [-\pi, \pi]$ being a zero of order $2p$, then T_n are positive definite for all n. Moreover,

$$0 < \lambda_{\min}(T_n) \leq O(n^{-2p}).$$

From Theorems 1.12 and 1.13, we know that if $f \geq 0$ and has a zero of order $2p$, then T_n is always positive definite but the condition number of T_n will be $\kappa(T_n) = O(n^{2p})$, which is unbounded as n tends to infinity; i.e., T_n is ill-conditioned. If f is nondefinite, then T_n is also nondefinite. Also, the equal distribution of eigenvalues of T_n implies that the CG method, when applied to the system $T_n\mathbf{u} = \mathbf{b}$, will converge slowly. Therefore, some efficient preconditioners are needed. This is the main motivation in developing later chapters.

Chapter 2

Circulant preconditioners

"What is circular is eternal. What is eternal is circular [37]*."*

Since 1986, many circulant preconditioners have been proposed for solving Toeplitz systems. Here we introduce some of them that have proven to be good preconditioners in the literature [24, 36, 55, 66]. Other useful noncirculant preconditioners will be briefly discussed.

2.1 Strang's circulant preconditioner

Let T_n given by (1.1) be generated by a real-valued function

$$f(x) = \sum_{k=-\infty}^{\infty} t_k e^{\mathbf{i}kx}$$

in the Wiener class, i.e.,

$$\sum_{k=-\infty}^{\infty} |t_k| < \infty,$$

where

$$t_k \equiv \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-\mathbf{i}kx} dx.$$

We remark that T_n are Hermitian for all n and the Wiener class is a proper subset of $\mathbf{C}_{2\pi}$.

Strang's preconditioner $s(T_n)$ is defined to be the circulant matrix obtained by copying the central diagonals of T_n and bringing them around to complete the circulant requirement. More precisely, the diagonals s_k of $s(T_n)$ are given by

$$s_k = \begin{cases} t_k, & 0 \le k \le m, \\ t_{k-n}, & m < k \le n-1, \\ \bar{s}_{-k}, & 0 < -k \le n-1. \end{cases} \tag{2.1}$$

For simplicity, here we assume that $n = 2m + 1$. The case $n = 2m$ can be treated similarly and in that case, we define $s_m = 0$ or

$$s_m = \frac{1}{2}(t_m + t_{-m}).$$

Algorithm A.3 with `pchoice=2` generates the first column of $s(T_n)$ and then its eigenvalues `ev`.

According to the requirement III in Section 1.3.1, a necessary condition for $s(T_n)$ to be a good preconditioner for T_n is that $(s(T_n))^{-1}T_n$ has a clustered spectrum. The first step of proving this is to show the following lemma.

Lemma 2.1. *Let f be a positive real-valued function in the Wiener class. Then for large n, the matrices $s(T_n)$ and $(s(T_n))^{-1}$ are uniformly bounded in $\|\cdot\|_2$.*

Proof. By using (1.6), the jth eigenvalue of $s(T_n)$ is equal to

$$\lambda_j(s(T_n)) = \sum_{k=-m}^{m} t_k e^{\frac{2\pi \mathbf{i} jk}{n}}.$$

Since the series $\sum_{k=-\infty}^{\infty} t_k e^{\mathbf{i}kx}$ is absolutely convergent for $x \in [-\pi, \pi]$, for any given $\epsilon > 0$, there exists an N such that for $n > N$ (equivalently $m > (N-1)/2$),

$$\left| \sum_{|k|>m} t_k e^{\mathbf{i}kx} \right| < \epsilon.$$

Therefore, for any j,

$$\begin{aligned}
\lambda_j(s(T_n)) &= \sum_{k=-m}^{m} t_k e^{\frac{2\pi \mathbf{i} jk}{n}} - f\left(\frac{2\pi j}{n}\right) + f\left(\frac{2\pi j}{n}\right) \\
&= \sum_{k=-m}^{m} t_k e^{\frac{2\pi \mathbf{i} jk}{n}} - \sum_{k=-\infty}^{\infty} t_k e^{\frac{2\pi \mathbf{i} jk}{n}} + f\left(\frac{2\pi j}{n}\right) \\
&= f\left(\frac{2\pi j}{n}\right) - \sum_{|k|>m} t_k e^{\frac{2\pi \mathbf{i} jk}{n}} \\
&\geq f_{\min} - \left| \sum_{|k|>m} t_k e^{\mathbf{i}kx} \right| = f_{\min} - \epsilon.
\end{aligned}$$

By choosing $\epsilon = \frac{1}{2} f_{\min} > 0$, the result follows. □

Next we show that $T_n - s(T_n)$ has a clustered spectrum. The following theorem was first proved in [27] by using the theory of compact operators. Here we will use a purely linear algebra technique developed in [17].

Theorem 2.2. *Let f be a function in the Wiener class. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of $T_n - s(T_n)$ have absolute values exceeding ϵ.*

Proof. We note that $B_n \equiv T_n - s(T_n)$ is a Hermitian Toeplitz matrix with entries $b_{ij} = b_{i-j}$ given by

$$b_k = \begin{cases} 0, & 0 \le k \le m, \\ t_k - t_{k-n}, & m < k \le n-1, \\ \bar{b}_{-k}, & 0 < -k \le n-1. \end{cases}$$

Since f is in the Wiener class, for all given $\epsilon > 0$, there exists an $N > 0$ such that

$$\sum_{k=N+1}^{\infty} |t_k| < \epsilon.$$

In the following, we will use ϵ to denote a small positive generic constant. Let $U_n^{(N)}$ be the n-by-n matrix obtained from B_n by replacing the $(n-N)$-by-$(n-N)$ leading principal submatrix of B_n by the zero matrix. Then

$$\operatorname{rank}(U_n^{(N)}) \le 2N.$$

Let

$$W_n^{(N)} \equiv B_n - U_n^{(N)}.$$

The leading $(n-N)$-by-$(n-N)$ block of $W_n^{(N)}$ is the leading $(n-N)$-by-$(n-N)$ principal submatrix of B_n, and hence this block is a Toeplitz matrix. It is easy to see that the maximum absolute column sum of $W_n^{(N)}$ is attained at the first column (or the $(n-N-1)$th column). Thus

$$\|W_n^{(N)}\|_1 = \sum_{k=m+1}^{n-N-1} |b_k| = \sum_{k=m+1}^{n-N-1} |t_k - t_{k-n}| \le 2 \sum_{k=N+1}^{n-N-1} |t_k| < \epsilon.$$

Since $W_n^{(N)}$ is Hermitian, we have $\|W_n^{(N)}\|_\infty = \|W_n^{(N)}\|_1$. Thus

$$\|W_n^{(N)}\|_2 \le \left(\|W_n^{(N)}\|_1 \cdot \|W_n^{(N)}\|_\infty\right)^{\frac{1}{2}} < \epsilon.$$

Hence the spectrum of $W_n^{(N)}$ lies in $(-\epsilon, \epsilon)$. By Weyl's theorem, we see that at most $2N$ eigenvalues of $B_n = T_n - s(T_n)$ have absolute values exceeding ϵ. □

Combining Lemma 2.1 and Theorem 2.2, and using the fact that

$$(s(T_n))^{-1} T_n = I_n + (s(T_n))^{-1}(T_n - s(T_n)),$$

we have the following corollary.

Corollary 2.3. *Let f be a positive function in the Wiener class. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of $(s(T_n))^{-1}T_n - I_n$ have absolute values larger than ϵ.*

Thus the spectrum of $(s(T_n))^{-1}T_n$ is clustered around 1 for large n. It follows that the convergence rate of the PCG method is superlinear; refer to Section 1.3.1 for details.

If extra smoothness conditions are imposed on the generating function f, we can obtain more precise estimates on how

$$|||\mathbf{e}^{(k)}|||^2 = \mathbf{e}^{(k)^*}(s(T_n))^{-1/2}T_n(s(T_n))^{-1/2}\mathbf{e}^{(k)}$$

goes to zero. Here $\mathbf{e}^{(k)}$ is the error at the kth iteration of the PCG method. The following theorem can be found in [60, 61, 79].

Theorem 2.4. *Suppose f is a rational function of the form $f(z) = p(z)/q(z)$, where $p(z)$ and $q(z)$ are polynomials of degrees μ and ν, respectively. Then the number of outlying eigenvalues of $(s(T_n))^{-1}T_n$ is exactly equal to $2\max\{\mu,\nu\}$. Hence, the method converges in at most $2\max\{\mu,\nu\}+1$ steps for large n. If, however,*

$$f(z) = \sum_{j=0}^{\infty} a_j z^j$$

is analytic only in a neighborhood of $|z| = 1$, then there exist constants $c > 0$ and $0 \le r < 1$ such that

$$\frac{|||\mathbf{e}^{(k+1)}|||}{|||\mathbf{e}^{(0)}|||} \le c^k r^{k^2/4+k/2}.$$

For ν-times differentiable generating functions f, we have the following theorem for the convergence rate of the PCG method with Strang's preconditioner.

Theorem 2.5 (R. Chan [17]). *Let f be a ν-times differentiable function with $f^{(\nu)} \in L^1[-\pi,\pi]$, where $\nu > 1$. Then there exists a constant $c > 0$ which depends only on f and ν such that for large n,*

$$\frac{|||\mathbf{e}^{(2k)}|||}{|||\mathbf{e}^{(0)}|||} \le \frac{c^k}{((k-1)!)^{2\nu-2}}.$$

The theorem was proved by using Weyl's theorem. R. Chan and Yeung later used Jackson's theorems [35, pp. 143–148] in approximation theory to prove a stronger result than that in Theorem 2.5.

Theorem 2.6 (R. Chan and Yeung [31]). *Suppose f is a Lipschitz function of order ν for $0 < \nu \le 1$, or f has a continuous νth derivative for $\nu \ge 1$. Then there exists a constant $c > 0$ which depends only on f and ν such that for large n,*

$$\frac{|||\mathbf{e}^{(2k)}|||}{|||\mathbf{e}^{(0)}|||} \le \prod_{p=2}^{k} \frac{c\log^2 p}{p^{2\nu}}.$$

2.2 Optimal (circulant) preconditioner

T. Chan in [33] proposed a specific circulant preconditioner called the optimal circulant preconditioner for solving Toeplitz systems. His idea was then extended in [22, 82] for general matrices. Thus, we begin with the general case. Given a unitary matrix $U \in \mathbb{C}^{n\times n}$, let

$$\mathscr{M}_U \equiv \{U^*\Lambda_n U \mid \Lambda_n \text{ is any } n\text{-by-}n \text{ diagonal matrix}\}. \tag{2.2}$$

We note that if $U = F$, the Fourier matrix, $\mathscr{M}_F$ is the set of all circulant matrices [37]. T. Chan's preconditioner $c_U(A_n)$ for a general matrix A_n is defined to be the minimizer of

$$\min_{W_n \in \mathscr{M}_U} \|A_n - W_n\|_{\mathscr{F}},$$

where $\|\cdot\|_{\mathscr{F}}$ is the Frobenius norm. Let $\delta(A_n)$ denote the diagonal matrix whose diagonal is equal to the diagonal of the matrix A_n. The following theorem includes some important properties of T. Chan's preconditioner.

Theorem 2.7. *For any arbitrary $A_n = (a_{pq}) \in \mathbb{C}^{n\times n}$, let $c_U(A_n)$ be the minimizer of $\|A_n - W_n\|_{\mathscr{F}}$ over all $W_n \in \mathscr{M}_U$. Then the following hold:*

(i) *$c_U(A_n)$ is uniquely determined by A_n and is given by*

$$c_U(A_n) = U^*\delta(UA_nU^*)U. \tag{2.3}$$

(ii) *We have*

$$\sigma_{\max}\big(c_U(A_n)\big) \le \sigma_{\max}(A_n),$$

where $\sigma_{\max}(\cdot)$ denotes the largest singular value.

(iii) *If A_n is Hermitian, then $c_U(A_n)$ is also Hermitian. Furthermore, we have*

$$\lambda_{\min}(A_n) \le \lambda_{\min}\big(c_U(A_n)\big) \le \lambda_{\max}\big(c_U(A_n)\big) \le \lambda_{\max}(A_n),$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalues, respectively. In particular, if A_n is positive definite, then so is $c_U(A_n)$.

(iv) *c_U is a linear projection operator from $\mathbb{C}^{n\times n}$ into $\mathscr{M}_U$ and has the operator norms*

$$\|c_U\|_2 = \sup_{\|A_n\|_2=1} \|c_U(A_n)\|_2 = 1$$

and

$$\|c_U\|_{\mathscr{F}} = \sup_{\|A_n\|_{\mathscr{F}}=1} \|c_U(A_n)\|_{\mathscr{F}} = 1.$$

(v) *When U is the Fourier matrix F, we have*

$$c_F(A_n) = \sum_{j=0}^{n-1} \Big(\frac{1}{n} \sum_{p-q\equiv j(\bmod n)} a_{pq}\Big) Q^j, \tag{2.4}$$

where Q is an n-by-n circulant matrix given by

$$Q \equiv \begin{pmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ 0 & 1 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}. \tag{2.5}$$

Proof. We prove (i), (ii), (iii), and (iv). We refer readers to [82] for (v).

(i) Since the Frobenius norm is unitary invariant, we have

$$\|W_n - A_n\|_{\mathscr{F}} = \|U^*\Lambda_n U - A_n\|_{\mathscr{F}} = \|\Lambda_n - UA_nU^*\|_{\mathscr{F}}.$$

Thus the problem of minimizing $\|W_n - A_n\|_{\mathscr{F}}$ over $\mathscr{M}_U$ is equivalent to the problem of minimizing $\|\Lambda_n - UA_nU^*\|_{\mathscr{F}}$ over all diagonal matrices. Since Λ_n can affect only the diagonal entries of UA_nU^*, we see that the solution for the latter problem is $\Lambda_n = \delta(UA_nU^*)$. Hence $U^*\delta(UA_nU^*)U$ is the minimizer of $\|W_n - A_n\|_{\mathscr{F}}$. It is clear from the argument that Λ_n and hence $c_U(A_n)$ are uniquely determined by A_n.

(ii) Note that the set of the singular values of $c_U(A_n)$ is the same as that of $\delta(UA_nU^*)$. We have by Corollary 3.1.3 in [52, p. 149],

$$|[\delta(UA_nU^*)]_{ii}| \le \sigma_{\max}(UA_nU^*) = \sigma_{\max}(A_n).$$

Therefore,

$$\sigma_{\max}(c_U(A_n)) = \max_i |[\delta(UA_nU^*)]_{ii}| \le \sigma_{\max}(A_n).$$

(iii) It is clear that $c_U(A_n)$ is Hermitian when A_n is Hermitian. By (i), we know that the eigenvalues of $c_U(A_n)$ are given by $\delta(UA_nU^*)$. Suppose that

$$\delta(UA_nU^*) = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

with

$$\lambda_j = \lambda_{\min}(c_U(A_n)), \qquad \lambda_k = \lambda_{\max}(c_U(A_n)).$$

Let $\mathbf{e}_j$ and $\mathbf{e}_k \in \mathbb{R}^n$ denote the jth and the kth unit vectors, respectively. We have by the Courant–Fischer minimax theorem,

$$\begin{aligned} \lambda_{\min}(A_n) &= \min_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^*A_n\mathbf{x}}{\mathbf{x}^*\mathbf{x}} = \min_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^*UA_nU^*\mathbf{x}}{\mathbf{x}^*\mathbf{x}} \\ &\le \frac{\mathbf{e}_j^*UA_nU^*\mathbf{e}_j}{\mathbf{e}_j^*\mathbf{e}_j} = \lambda_j = \lambda_{\min}(c_U(A_n)) \\ &\le \lambda_{\max}(c_U(A_n)) = \lambda_k = \frac{\mathbf{e}_k^*UA_nU^*\mathbf{e}_k}{\mathbf{e}_k^*\mathbf{e}_k} \\ &\le \max_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^*UA_nU^*\mathbf{x}}{\mathbf{x}^*\mathbf{x}} = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^*A_n\mathbf{x}}{\mathbf{x}^*\mathbf{x}} = \lambda_{\max}(A_n). \end{aligned}$$

(iv) We have by (ii),

$$\|c_U(A_n)\|_2 = \sigma_{\max}(c_U(A_n)) \le \sigma_{\max}(A_n) = \|A_n\|_2.$$

However, for the identity matrix I_n, we have

$$\|c_U(I_n)\|_2 = \|I_n\|_2 = 1.$$

Hence $\|c_U\|_2 = 1$. For the Frobenius norm, since

$$\|c_U(A_n)\|_{\mathscr{F}} = \|\delta(UA_nU^*)\|_{\mathscr{F}} \le \|UA_nU^*\|_{\mathscr{F}} = \|A_n\|_{\mathscr{F}}$$

and

$$\Big\|c_U\Big(\frac{1}{\sqrt{n}}I_n\Big)\Big\|_{\mathscr{F}} = \frac{1}{\sqrt{n}}\|I_n\|_{\mathscr{F}} = 1,$$

it follows that $\|c_U\|_{\mathscr{F}} = 1$. □

We note that the matrix $c_F(A_n)$ in (2.4) was first named the optimal circulant preconditioner for Toeplitz matrices by T. Chan [33] in 1988. It has proven to be a good preconditioner for solving a large class of Toeplitz systems by the PCG method; see [22, 24, 55, 66]. For Toeplitz matrices T_n given by (1.1), the diagonals c_k of $c_F(T_n)$ are given by

$$c_k = \begin{cases} \dfrac{(n-k)t_k + kt_{k-n}}{n}, & 0 \le k \le n-1, \\ c_{n+k}, & 0 < -k \le n-1; \end{cases} \tag{2.6}$$

see (2.4) and (2.5). The construction of $c_F(T_n)$ therefore requires $O(n)$ operations. In contrast, the construction of $c_F(A_n)$ for general matrices A_n requires $O(n^2)$ operations in view of (2.4). Algorithm A.3 with `pchoice=1` generates the first column of $c_F(T_n)$ and then its eigenvalues `ev`.

We introduce the following two lemmas [16] in order to analyze the convergence rate of the PCG method with T. Chan's preconditioner.

Lemma 2.8. *Let f be a function in the Wiener class. Then*

$$\lim_{n\to\infty} \rho\,[s(T_n) - c_F(T_n)] = 0,$$

where $\rho[\cdot]$ denotes the spectral radius.

Proof. By (2.1) and (2.6), it is clear that $B_n \equiv s(T_n) - c_F(T_n)$ is circulant with entries

$$b_k = \begin{cases} \dfrac{k}{n}(t_k - t_{k-n}), & 0 \le k \le m, \\ \dfrac{n-k}{n}(t_{k-n} - t_k), & m < k \le n-1. \end{cases}$$

Here for simplicity, we assume $n = 2m$. By (1.6), the jth eigenvalue $\lambda_j(B_n)$ of B_n is given by $\sum_{k=0}^{n-1} b_k e^{\frac{2\pi \mathrm{i} jk}{n}}$, and we therefore have

$$\lambda_j(B_n) \le 2\sum_{k=1}^{m} \frac{k}{n}(|t_k| + |t_{k-n}|).$$

This implies

$$\rho[B_n] \leq 2\sum_{k=1}^{m} \frac{k}{n}|t_k| + 2\sum_{k=m}^{n-1} |t_k|.$$

Since f is in the Wiener class, for all $\epsilon > 0$, we can always find an $M_1 > 0$ and then an $M_2 > M_1$ such that

$$\sum_{k=M_1+1}^{\infty} |t_k| < \frac{\epsilon}{6}, \qquad \frac{1}{M_2}\sum_{k=1}^{M_1} k|t_k| < \frac{\epsilon}{6}.$$

Thus for all $m > M_2$,

$$\rho[B_n] < \frac{2}{M_2}\sum_{k=1}^{M_1} k|t_k| + 2\sum_{k=M_1+1}^{m} |t_k| + 2\sum_{k=m}^{\infty} |t_k| < \epsilon. \qquad \square$$

Lemma 2.9. *Let $f \in \mathbf{C}_{2\pi}$ be a positive function. Then the matrices $c_U(T_n)$ and $(c_U(T_n))^{-1}$ are uniformly bounded in the norm $\|\cdot\|_2$.*

Proof. Just use the Grenander–Szegö theorem and then Theorem 2.7(iii). □

Note that

$$(c_F(T_n))^{-1}T_n = I_n + (c_F(T_n))^{-1}[T_n - s(T_n)] + (c_F(T_n))^{-1}[s(T_n) - c_F(T_n)].$$

By using Theorem 2.2 and Lemmas 2.8 and 2.9, we have the following theorem.

Theorem 2.10. *Let f be a positive function in the Wiener class. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of $(c_F(T_n))^{-1}T_n - I_n$ have absolute values larger than ϵ.*

Thus the spectrum of $(c_F(T_n))^{-1}T_n$ is clustered around 1 for large n. It follows that the convergence rate of the PCG method is superlinear. In Section 3.1, we will extend the result in Theorem 2.10 from the Wiener class to $\mathbf{C}_{2\pi}$.

2.3 Superoptimal preconditioner

Like T. Chan's preconditioner, the superoptimal preconditioner is defined not only for Toeplitz matrices but also for general matrices. Thus we begin with the general case. The idea is to consider a minimization procedure concerning a kind of relative error in the matrix sense instead of the absolute error considered in Section 2.2. More precisely, the superoptimal preconditioner $t_U(A_n)$ for any matrix A_n is defined to be the minimizer of

$$\min \|I_n - C_n^{-1}A_n\|_{\mathscr{F}}$$

over all nonsingular matrices $C_n \in \mathscr{M}_U$, where $\mathscr{M}_U$ is given by (2.2). We need the following lemma in order to prove Theorem 2.12, which relates the superoptimal preconditioner $t_U(A_n)$ with T. Chan's preconditioner $c_U(A_n)$.

Lemma 2.11. *The matrix $c_U(A_nA_n^*) - c_U(A_n)c_U(A_n^*)$ is positive semidefinite.*

Proof. Define

$$\begin{aligned} D_n &\equiv c_U(A_nA_n^*) - c_U(A_n)c_U(A_n^*) \\ &= U^*[\delta(UA_nA_n^*U^*) - \delta(UA_nU^*)\delta(UA_n^*U^*)]U. \end{aligned}$$

It is sufficient to show that the eigenvalues of D_n, given by

$$[\delta(UA_nA_n^*U^*) - \delta(UA_nU^*)\delta(UA_n^*U^*)]_{kk}\,, \qquad k = 1, \ldots, n,$$

are all nonnegative. We notice that

$$\begin{aligned} [\delta(UA_nA_n^*U^*)]_{kk} &= [\delta(UA_nU^* \cdot UA_n^*U^*)]_{kk} \\ &= \sum_{p=1}^{n}(UA_nU^*)_{kp}(UA_n^*U^*)_{pk} = \sum_{p=1}^{n}(UA_nU^*)_{kp}\overline{(UA_nU^*)_{kp}} \\ &\geq (UA_nU^*)_{kk}\overline{(UA_nU^*)_{kk}} = [\delta(UA_nU^*)]_{kk} \cdot [\delta(UA_n^*U^*)]_{kk}\,. \end{aligned}$$

Therefore, the eigenvalues of D_n are all nonnegative. □

Theorem 2.12. *Let $A_n \in \mathbb{C}^{n\times n}$ be such that both A_n and $c_U(A_n)$ are nonsingular. Then the superoptimal preconditioner $t_U(A_n)$ exists and is equal to*

$$t_U(A_n) = c_U(A_nA_n^*)(c_U(A_n^*))^{-1}. \tag{2.7}$$

Proof. Instead of minimizing $\|I - C_n^{-1}A_n\|_{\mathscr{F}}$, we consider the problem of minimizing $\|I - \widehat{C}_nA_n\|_{\mathscr{F}}$ over all nonsingular $\widehat{C}_n$ in $\mathscr{M}_U$. Let $\widehat{C}_n = U^*\Lambda_nU$. We then have

$$\begin{aligned} \|I - \widehat{C}_nA_n\|_{\mathscr{F}} &= \|I - U^*\Lambda_nUA_n\|_{\mathscr{F}} = \|I - \Lambda_nUA_nU^*\|_{\mathscr{F}} \\ &= \operatorname{tr}(I - \Lambda_nUA_nU^* - UA_n^*U^*\Lambda_n^* + \Lambda_nUA_nA_n^*U^*\Lambda_n^*) \\ &= \operatorname{tr}[I - \Lambda_n\delta(UA_nU^*) - \delta(UA_n^*U^*)\Lambda_n^* + \Lambda_n\delta(UA_nA_n^*U^*)\Lambda_n^*], \end{aligned}$$

where $\operatorname{tr}(M)$ denotes the trace of the matrix M. Let

$$\Lambda_n \equiv \operatorname{diag}(\lambda_1, \ldots, \lambda_n), \qquad \delta(UA_nU^*) \equiv \operatorname{diag}(u_1, \ldots, u_n)$$

and

$$\delta(UA_nA_n^*U^*) \equiv \mathrm{diag}(w_1, \ldots, w_n).$$

We therefore have

$$\begin{aligned}
&\min \|I - \widehat{C}_n A_n\|_{\mathscr{F}} \\
&= \min \left\{ \mathrm{tr}[I - \Lambda_n \delta(UA_nU^*) - \delta(UA_n^*U^*)\Lambda_n^* + \Lambda_n \delta(UA_nA_n^*U^*)\Lambda_n^*] \right\} \\
&= \min_{\{\lambda_1,\ldots,\lambda_n\}} \sum_{k=1}^{n} (1 - \lambda_k u_k - \overline{u}_k \overline{\lambda}_k + \lambda_k w_k \overline{\lambda}_k).
\end{aligned}$$

Notice that by Lemma 2.11, $w_k \geq u_k \bar{u}_k$ for $k = 1, \ldots, n$. Hence for all complex scalars λ_k, $k = 1, \ldots, n$, the terms

$$1 - \lambda_k u_k - \overline{u}_k \overline{\lambda}_k + \lambda_k w_k \overline{\lambda}_k$$

are nonnegative. Differentiating them with respect to the real and imaginary parts of λ_k and setting the derivatives to zero, we obtain

$$\lambda_k = \frac{\overline{u}_k}{w_k}, \qquad k = 1, \ldots, n.$$

Since A_n and $c_U(A_n)$ are nonsingular, both w_k and u_k are nonzero. Hence λ_k are also nonzero. Thus the minimizer of $\|I - \widehat{C}_n A_n\|_{\mathscr{F}}$ is nonsingular and is given by

$$\begin{aligned}
\widehat{C}_n &= U^* \Lambda_n U = U^* \delta(UA_n^*U^*)[\delta(UA_nA_n^*U^*)]^{-1} U \\
&= U^* \delta(UA_n^*U^*) U \cdot [U^* \delta(UA_nA_n^*U^*) U]^{-1} \\
&= c_U(A_n^*)(c_U(A_nA_n^*))^{-1}.
\end{aligned}$$

Therefore, the superoptimal preconditioner is given by

$$t_U(A_n) = \widehat{C}_n^{-1} = c_U(A_nA_n^*)(c_U(A_n^*))^{-1}. \qquad \square$$

When U equals the Fourier matrix F, the matrix $t_F(A_n)$ was first named the superoptimal circulant preconditioner by Tyrtyshnikov [82] in 1992. The construction of $t_F(T_n)$ requires $O(n \log n)$ operations for Toeplitz matrices T_n, see [22], and the construction of $t_F(A_n)$ for general matrices A_n requires $O(n^3)$ operations by (2.7). Algorithm A.3 with `pchoice=12` generates the first column of $t_F(T_n)$ and then its eigenvalues `ev`.

Now we consider the solution of the Toeplitz system $T_n\mathbf{u} = \mathbf{b}$ by using the PCG method with the preconditioner $t_F(T_n)$. For Hermitian positive definite Toeplitz matrix T_n, we have by (2.7),

$$t_F(T_n) = c_F(T_n^2)(c_F(T_n))^{-1}.$$

Therefore,

$$\begin{aligned}
&(t_F(T_n))^{-1} T_n \\
&\quad = I + (t_F(T_n))^{-1}[T_n - c_F(T_n)] + (t_F(T_n))^{-1}[c_F(T_n) - t_F(T_n)] \\
&\quad = I + (t_F(T_n))^{-1}[T_n - c_F(T_n)] + (c_F(T_n^2))^{-1}[(c_F(T_n))^2 - c_F(T_n^2)]. \quad (2.8)
\end{aligned}$$

Lemma 2.13. *Let the generating function f be a positive function in the Wiener class. Then for n sufficient large, $c_F(T_n^2)$, $t_F(T_n)$, and their inverses are all uniformly bounded in $\|\cdot\|_2$.*

Proof. By the Grenander–Szegö theorem and Theorem 2.7(iii), we have

$$f_{\min}^2 \le \lambda_{\min}(T_n^2) \le \lambda_{\min}(c_F(T_n^2)) \le \lambda_{\max}(c_F(T_n^2)) \le \lambda_{\max}(T_n^2) \le f_{\max}^2.$$

Therefore,

$$\|(t_F(T_n))^{-1}\|_2 = \|c_F(T_n)(c_F(T_n^2))^{-1}\|_2 \le \|c_F(T_n)\|_2 \|(c_F(T_n^2))^{-1}\|_2 \le \frac{f_{\max}}{f_{\min}^2}$$

and

$$\|(t_F(T_n))\|_2 = \|c_F(T_n^2)(c_F(T_n))^{-1}\|_2 \le \|c_F(T_n^2)\|_2 \|(c_F(T_n))^{-1}\|_2 \le \frac{f_{\max}^2}{f_{\min}}. \qquad \square$$

Lemma 2.14. *Let f be a function in the Wiener class. Then*

$$\lim_{n\to\infty} \rho\left[(c_F(T_n))^2 - c_F(T_n^2)\right] = 0,$$

where $\rho[\cdot]$ denotes the spectral radius.

The proof of Lemma 2.14 can be found in [23]. By using (2.8), Theorem 2.10, and Lemmas 2.13 and 2.14, we have the following theorem for the convergence rate of the PCG method by using $t_F(T_n)$.

Theorem 2.15. *Let the generating function f be a positive function in the Wiener class. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of $(t_F(T_n))^{-1}T_n - I$ have absolute values larger than ϵ.*

Hence, if the CG method is applied to the preconditioned system, we can expect a superlinear convergence rate. We remark that the superoptimal circulant preconditioner was used by Di Benedetto, Estatico, and Serra Capizzano in 2005 to solve some ill-conditioned Toeplitz systems arising from image deblurring [5].

2.4 Other preconditioners

In this section, we briefly discuss some other well-known preconditioners which have proven to be useful in the literature [24, 36, 55, 66].

2.4.1 Huckle's circulant preconditioner

For T_n given by (1.1), Huckle's preconditioner $h^{(p)}(T_n)$ proposed in [53] is defined to be the circulant matrix with eigenvalues

$$\lambda_k(h^{(p)}(T_n)) = \sum_{j=-p+1}^{p-1} t_j \left(1 - \frac{|j|}{p}\right) e^{\frac{2\pi \mathrm{i} jk}{n}}, \qquad k = 0, \ldots, n-1. \tag{2.9}$$

When $p = n$, it is simply T. Chan's circulant preconditioner. If $f > 0$ is the generating function of T_n with Fourier coefficients t_k that satisfy

$$\sum_{k=0}^{\infty} |k||t_k|^2 < \infty,$$

then it was proved [53] that the spectra of $(h^{(p)}(T_n))^{-1}T_n$ are clustered around 1 for large n. Thus, the convergence rate of the PCG method is superlinear.

2.4.2 Preconditioners by embedding

Let the Toeplitz matrix T_n be embedded into a $2n$-by-$2n$ circulant matrix

$$\begin{pmatrix} T_n & B_n^* \\ B_n & T_n \end{pmatrix}. \tag{2.10}$$

R. Chan's [17] circulant preconditioner is defined as $r(T_n) = T_n + B_n$. Using the embedding (2.10), Ku and Kuo [59] constructed four different preconditioners $K_{(i)}$, $1 \leq i \leq 4$, based on different combinations of T_n and B_n. They are

$$\begin{aligned} K_{(1)} &= T_n + B_n = r(T_n), & K_{(2)} &= T_n - B_n, \\ K_{(3)} &= T_n + JB_n, & K_{(4)} &= T_n - JB_n, \end{aligned}$$

where J is the n-by-n anti-identity (reversal) matrix. Note that $K_{(2)}$, $K_{(3)}$, and $K_{(4)}$ are not circulant matrices. For the implementation of these preconditioners, we refer readers to [17, 59] for details.

2.4.3 Noncirculant optimal preconditioners

Besides FFT, many fast transforms are used in scientific computing and engineering. By letting U in (2.2) be other fast transform matrices, we can have new classes of optimal preconditioners.

Optimal preconditioner based on sine transform

Let $\mathcal{S} = \mathscr{M}_{\Phi^s}$ be the set of all n-by-n matrices that can be diagonalized by the discrete sine transform matrix Φ^s, i.e.,

$$\mathcal{S} = \{\Phi^s \Lambda_n \Phi^s \mid \Lambda_n \text{ is any } n\text{-by-}n \text{ diagonal matrix}\}.$$

Here the (j,k)th entry of Φ^s is given by

$$\sqrt{\frac{2}{n+1}} \sin\left(\frac{\pi j k}{n+1}\right)$$

for $1 \leq j, k \leq n$. Given any arbitrary matrix $A_n \in \mathbb{C}^{n \times n}$, we define the operator Ψ_s which maps A_n to $\Psi_s(A_n)$ that minimizes $\|A_n - B_n\|_{\mathscr{F}}$ over all $B_n \in \mathcal{S}$; see [9]. For the construction of $\Psi_s(A_n)$, we refer readers to [25].

Optimal preconditioner based on cosine transform

Let $\mathcal{C} = \mathscr{M}_{\Phi^c}$ be the set of all n-by-n matrices that can be diagonalized by the discrete cosine transform matrix Φ^c, i.e.,

$$\mathcal{C} = \{(\Phi^c)^T \Lambda_n \Phi^c \mid \Lambda_n \text{ is any } n\text{-by-}n \text{ diagonal matrix}\}.$$

Here the (j,k)th entry of Φ^c is given by

$$\sqrt{\frac{2-\delta_{j1}}{n}} \cos\left(\frac{(j-1)(2k-1)\pi}{2n}\right)$$

for $1 \le j,k \le n$. The symbol δ_{jk} is the Kronecker delta defined by

$$\delta_{jk} = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases}$$

Given any arbitrary matrix $A_n \in \mathbb{C}^{n\times n}$, we define the operator Ψ_c which maps A_n to $\Psi_c(A_n)$ that minimizes $\|A_n - B_n\|_{\mathscr{F}}$ over all $B_n \in \mathcal{C}$; see [20]. The construction of $\Psi_c(A_n)$ is also given in [20].

Optimal preconditioner based on Hartley transform

Let $\mathcal{H} = \mathscr{M}_{\Phi^h}$ be the set of all n-by-n matrices that can be diagonalized by the discrete Hartley transform matrix Φ^h, i.e.,

$$\mathcal{H} = \{\Phi^h \Lambda_n \Phi^h \mid \Lambda_n \text{ is any } n\text{-by-}n \text{ diagonal matrix}\}.$$

Here the (j,k)th entry of Φ^h is given by

$$\frac{1}{\sqrt{n}} \cos\left(\frac{2\pi(j-1)(k-1)}{n}\right) + \frac{1}{\sqrt{n}} \sin\left(\frac{2\pi(j-1)(k-1)}{n}\right)$$

for $1 \le j,k \le n$. Given any arbitrary matrix $A_n \in \mathbb{C}^{n\times n}$, we define the operator Ψ_h which maps A_n to $\Psi_h(A_n)$ that minimizes $\|A_n - B_n\|_{\mathscr{F}}$ over all $B_n \in \mathcal{H}$; see [10]. The construction of $\Psi_h(A_n)$ is also given in [10].

Convergence result and operation cost

Let $f \in \mathbf{C}_{2\pi}$ be a positive even function. We can show that the spectra of $(\Psi_\alpha(T_n))^{-1}T_n$ are clustered around 1 for large n, where $\alpha = s, c, h$; see [12, 20, 25, 54]. Thus, the convergence rate of the PCG method is superlinear. In each iteration of the PCG method, we have to compute the matrix-vector multiplications $T_n\mathbf{v}$ and $(\Psi_\alpha(T_n))^{-1}\mathbf{w}$ for some vectors $\mathbf{v}$ and $\mathbf{w}$; see Section 1.3.1. We have already known that $T_n\mathbf{v}$ can be computed in $O(n\log n)$ operations. Like circulant systems, the vector $(\Psi_\alpha(T_n))^{-1}\mathbf{w} = \Phi^\alpha\Lambda_n^{-1}\Phi^\alpha\mathbf{w}$ can also be computed in $O(n\log n)$ operations by using the fast sine transform for $\alpha = s$, the fast cosine transform for $\alpha = c$, or the fast Hartley transform for $\alpha = h$. Thus, the complexity of the PCG algorithm remains $O(n\log n)$.

2.4.4 Band-Toeplitz preconditioners

R. Chan and Tang proposed in [28] to use band-Toeplitz matrices B_n as preconditioners for solving symmetric positive definite Toeplitz systems $T_n\mathbf{u} = \mathbf{b}$ by the PCG method, where T_n are assumed to be generated by a function $f \in \mathbf{C}_{2\pi}^{+}$ with zeros. By Theorem 1.13, we know that T_n is ill-conditioned. Let g be the generating function of a band-Toeplitz matrix B_n. The function g is constructed not only to match the zeros of f but also to minimize

$$\Big|\Big|\frac{f-g}{f}\Big|\Big|_{\infty},$$

where for all $p \in \mathbf{C}_{2\pi}$,

$$\|p\|_{\infty} \equiv \max_{-\pi \le x \le \pi} |p(x)|$$

is the supremum norm. We remark that $\mathbf{C}_{2\pi}$ is a Banach space with the supremum norm. R. Chan and Tang proved the following theorem, which gives a bound on the condition number of the preconditioned matrix $B_n^{-1}T_n$.

Theorem 2.16. *Let $f \in \mathbf{C}_{2\pi}^{+}$ be the generating function of T_n with zeros and g be the generating function of a band-Toeplitz matrix B_n:*

$$g(x) = \sum_{k=-N}^{N} b_k e^{\mathbf{i}kx}$$

with $b_{-k} = \bar{b}_k$. If

$$\Big|\Big|\frac{f-g}{f}\Big|\Big|_{\infty} = h < 1,$$

then B_n is positive definite and

$$\kappa(B_n^{-1}T_n) \le \frac{1+h}{1-h}$$

for all $n > 0$.

Proof. By the assumption, we have

$$f(x)(1-h) \le g(x) \le f(x)(1+h)$$

for any $x \in [-\pi, \pi]$. It is clear that $g(x) \ge 0$. By Theorem 1.13, B_n is positive definite for all $n > 0$. Since both T_n and B_n are Toeplitz matrices, we have

$$\mathbf{v}^* T_n \mathbf{v} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Big|\sum_{k=0}^{n-1} v_k e^{-\mathbf{i}kx}\Big|^2 f(x)dx$$

and

$$\mathbf{v}^* B_n \mathbf{v} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Big|\sum_{k=0}^{n-1} v_k e^{-\mathbf{i}kx}\Big|^2 g(x)dx,$$

where

$$\mathbf{0} \neq \mathbf{v} = (v_0, v_1, \ldots, v_{n-1})^T \in \mathbb{C}^n.$$

Hence, we obtain

$$(1-h)\mathbf{v}^* T_n \mathbf{v} \leq \mathbf{v}^* B_n \mathbf{v} \leq (1+h)\mathbf{v}^* T_n \mathbf{v},$$

i.e.,

$$\frac{1}{1+h} \leq \frac{\mathbf{v}^* T_n \mathbf{v}}{\mathbf{v}^* B_n \mathbf{v}} \leq \frac{1}{1-h}.$$

By the Courant–Fischer minimax theorem, we know that

$$\frac{1}{1+h} \leq \lambda_{\min}(B_n^{-1}T_n) \leq \lambda_{\max}(B_n^{-1}T_n) \leq \frac{1}{1-h}.$$

We then have

$$\kappa(B_n^{-1}T_n) \leq \frac{1+h}{1-h}. \qquad \square$$

By using Theorem 1.9, we see that $|||\mathbf{e}^{(k)}|||/|||\mathbf{e}^0||| \leq \tau$, the tolerance, if the number of iterations

$$k > \frac{1}{2}\sqrt{\frac{1+h}{1-h}} \log\left(\frac{2}{\tau}\right).$$

The function g could be found by a version of the Remez algorithm with $O(N^3)$ operations. We stress that the construction of g is independent of n. Since h can be found explicitly in the Remez algorithm, we have a priori bound on the number of iterations for convergence.

To avoid the use of the Remez algorithm, Serra proposed in [70] to use the Chebyshev interpolation to construct g. We refer readers to [72] for a comparison between the optimal preconditioners based on fast transforms and the band-Toeplitz preconditioners.

2.4.5 $\{\omega\}$-circulant preconditioner

Potts and Steidl proposed in [68] to use $\{\omega\}$-circulant preconditioners to handle ill-conditioned Toeplitz systems $T_n\mathbf{u} = \mathbf{b}$. Here the generating function f of T_n is in $\mathbf{C}_{2\pi}^+$ with zeros in $[-\pi, \pi]$. The preconditioner P_n is constructed as follows. We choose uniform grids

$$x_k = w_n + \frac{2\pi k}{n},$$

where $w_n \in [-\pi, -\pi + 2\pi/n)$ such that

$$f(x_k) \neq 0, \qquad k = 0, \ldots, n-1.$$

Note that the choice of the grids requires some prior information about the zeros of f. Consider the preconditioner defined as

$$P_n = \Omega_n^* F_n^* \Lambda_n F_n \Omega_n, \tag{2.11}$$

where F_n is the Fourier matrix,

$$\Omega_n = \mathrm{diag}(1, e^{w_n \mathbf{i}}, e^{2w_n \mathbf{i}}, \ldots, e^{(n-1)w_n \mathbf{i}}),$$

and

$$\Lambda_n = \mathrm{diag}(f(x_0), f(x_1), f(x_2), \ldots, f(x_{n-1})). \tag{2.12}$$

The preconditioner P_n has the following properties (see [68]):

(i) P_n is Hermitian positive definite if $f \geq 0$.

(ii) P_n is an $\{e^{nw_n \mathbf{i}}\}$-circulant matrix [37]. Notice that $\{e^{nw_n \mathbf{i}}\}$-circulant matrices are Toeplitz matrices with the first entry of each column obtained by multiplying the last entry of the preceding column by $e^{nw_n \mathbf{i}}$.

(iii) Similar to that of circulant matrices, once the diagonal matrix Λ_n in (2.11) is obtained, the products of $P_n\mathbf{y}$ and $P_n^{-1}\mathbf{y}$ for any vector $\mathbf{y}$ can be computed by FFTs in $O(n \log n)$ operations.

(iv) In view of (2.11), P_n can be constructed in $O(n \log n)$ operations.

(v) The eigenvalues of $P_n^{-1}T_n$ are clustered around 1 and bounded away from zero.

The PCG method, when applied to the preconditioned system with the preconditioner P_n, will converge superlinearly. Therefore, the total complexity in solving the preconditioned system remains $O(n \log n)$.

Before the end of this section, we remark that R. Chan, Yip, and Ng [32] proposed a new family of circulant preconditioners called the best circulant preconditioner to solve ill-conditioned Toeplitz systems in 2001. Unlike B_n or P_n, the best circulant preconditioner can be constructed by using only the entries of the given matrix and does not require the explicit knowledge of the generating function f; see Chapter 4 for details.

2.5 Examples

In this section, we apply the PCG method with preconditioners $s(T_n)$, $c_F(T_n)$, and $t_F(T_n)$ to the Toeplitz system $T_n\mathbf{u} = \mathbf{b}$ with

$$t_k = \begin{cases} \dfrac{1+\sqrt{-1}}{(1+k)^{1.1}}, & k > 0, \\ 2, & k = 0, \\ \bar{t}_{-k}, & k < 0. \end{cases}$$

The right-hand side $\mathbf{b}$ is the vector of all ones, and the underlying generating function f is given by

$$f(x) = 2\sum_{k=0}^{\infty} \frac{\sin(kx) + \cos(kx)}{(1+k)^{1.1}}. \tag{2.13}$$

Clearly f is in the Wiener class. The MATLAB command to generate the first column of T_n is given in A.2 with `fchoice=1`. We should emphasize that in all our tests in the book, the zero vector is the initial guess and the stopping criterion is

$$\frac{\|\mathbf{r}^{(k)}\|_2}{\|\mathbf{r}^{(0)}\|_2} < 10^{-7},$$

where $\mathbf{r}^{(k)}$ is the residual vector after the kth iteration.

Table 2.1 shows the number of iterations required for convergence. The symbol I there signifies that no preconditioner is used. We see that as n increases, the number of iterations increases like $O(\log n)$ for the original matrix T_n, while it stays almost the same for the preconditioned matrices. Moreover, all preconditioned systems converge at the same rate for large n. The MATLAB programs used to generate Table 2.1 can be found in the appendix; see A.1–A.3 and A.5–A.7. To use them, one just has to run the algorithm A.1. It will prompt for the input of three parameters: `n`, the size of the system; `pchoice`, the choice of the preconditioner (e.g., enter 2 for Strang's preconditioner); and `fchoice`, the generating function used (in this case, we choose 1 for (2.13)).

To further illustrate Corollary 2.3 and Theorems 2.10 and 2.15, we give in Figure 2.1 the spectra of the matrices T_n, $(s(T_n))^{-1}T_n$, $(c_F(T_n))^{-1}T_n$, and $(t_F(T_n))^{-1}T_n$ for $n = 32$. We can see that the spectra of the preconditioned matrices are in a small interval around 1, except for few outliers, and that all the eigenvalues are well separated away from 0.

Table 2.1. *Preconditioners used and number of iterations.*

n	I	$s(T_n)$	$c_F(T_n)$	$t_F(T_n)$
32	15	7	6	8
64	17	7	7	7
128	19	7	7	7
256	20	7	7	7
512	21	7	7	7
1024	22	8	8	7

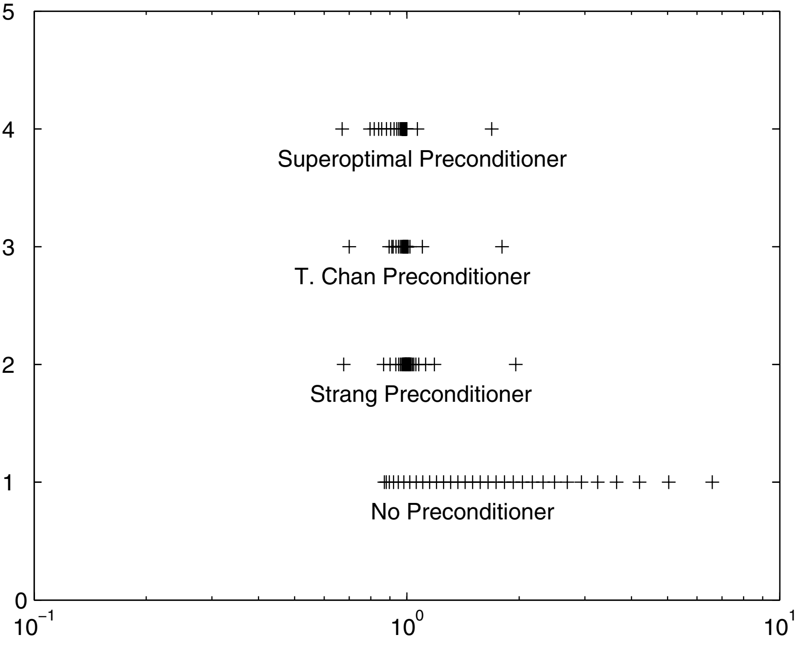

Figure 2.1. *Spectra of preconditioned matrices when* $n = 32$.

Chapter 3

Unified treatment from kernels

In this chapter, a unified treatment for constructing circulant preconditioners from the viewpoint of kernels is given [30]. We show that most of the well-known circulant preconditioners can be obtained from convoluting the generating function of the given Toeplitz matrix with some famous kernels. A convergence analysis is given together with some numerical examples.

3.1 Introduction

In this chapter, we use the symbol $\mathbf{C}_{2\pi}$ to denote the Banach space of all 2π-periodic continuous real-valued functions f equipped with the supremum norm $\|\cdot\|_\infty$. We first extend the result in Theorem 2.10 from the Wiener class to $\mathbf{C}_{2\pi}$ [29], as it will be used later to develop our theory. In the following, we use $T_n(f)$ to denote the n-by-n Toeplitz matrix generated by f; i.e., the diagonals of $T_n(f)$ are the Fourier coefficients of f (see (1.8)).

Theorem 3.1. *Let $f \in \mathbf{C}_{2\pi}$. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of $T_n(f) - c_F(T_n(f))$ have absolute values larger than ϵ.*

Proof. Since $f \in \mathbf{C}_{2\pi}$, for any $\epsilon > 0$, by the Weierstrass approximation theorem [58, p. 15], there is a trigonometric polynomial

$$p_N(x) = \sum_{k=-N}^{N} b_k e^{\mathbf{i}kx}$$

with $b_{-k} = \bar{b}_k$ for $|k| \le N$ such that

$$\|f - p_N\|_\infty \le \epsilon. \tag{3.1}$$

For all $n > 2N$, we have

$$\begin{aligned} &c_F(T_n(f)) - T_n(f) \\ &\quad = c_F(T_n(f - p_N)) - T_n(f - p_N) + c_F(T_n(p_N)) - T_n(p_N). \end{aligned} \tag{3.2}$$

For the first two terms in the right-hand side of (3.2), we note that by Theorem 2.7(iv), the Grenander–Szegö theorem, and (3.1),

$$\begin{aligned} &\|c_F(T_n(f - p_N)) - T_n(f - p_N)\|_2 \\ &\quad \le \|c_F(T_n(f - p_N))\|_2 + \|T_n(f - p_N)\|_2 \\ &\quad \le \|c_F\|_2 \cdot \|T_n(f - p_N)\|_2 + \|T_n(f - p_N)\|_2 \\ &\quad \le \|f - p_N\|_\infty + \|f - p_N\|_\infty \le 2\epsilon. \end{aligned}$$

Since p_N is a real-valued function in the Wiener class, we know that the matrix $c_F(T_n(p_N)) - T_n(p_N)$ has a clustered spectrum; see Section 2.2. Hence by using Weyl's theorem, the result follows. □

Since

$$(c_F(T_n))^{-1} T_n - I_n = (c_F(T_n))^{-1} \big[T_n - c_F(T_n)\big],$$

by Lemma 2.9, we have the following corollary.

Corollary 3.2. *Let T_n be a Toeplitz matrix with a positive generating function $f \in \mathbf{C}_{2\pi}$. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of the matrix $(c_F(T_n))^{-1} T_n - I_n$ have absolute values larger than ϵ.*

It follows that the convergence rate of the PCG method is superlinear. This result will be used in Section 3.4 for the convergence analysis of preconditioners derived by kernels. We remark that by using the relation (2.7) between $c_F(T_n)$ and $t_F(T_n)$, one can also extend the result in Theorem 2.15 for $t_F(T_n)$ from the Wiener class to $\mathbf{C}_{2\pi}$; see [6]. In the next section, we relate some of the circulant preconditioners discussed in Chapter 2 with well-known kernels in function theory.

3.2 Kernels of some circulant preconditioners

Let t_k be the Fourier coefficients of f as defined in (1.8). The jth partial sum of f is defined as

$$s_j[f](x) \equiv \sum_{k=-j}^{j} t_k e^{\mathbf{i}kx}, \qquad x \in \mathbb{R}. \tag{3.3}$$

Let us recall the relationship between the first column of a circulant matrix and its eigenvalues. Let C_n be a circulant matrix with the first column

$$(c_0, c_1, \ldots, c_{n-1})^T.$$

Then by (1.6), the eigenvalues of C_n can be written as follows:

$$\lambda_j(C_n) = (\Lambda_n)_{jj} = \sum_{k=0}^{n-1} c_k \zeta_j^k \tag{3.4}$$

with $\zeta_j \equiv e^{\frac{2\pi \mathbf{i} j}{n}}$, for $j = 0, \ldots, n-1$. Conversely, if the eigenvalues of C_n are given, then the first column of C_n can be obtained by using (1.5). Note that

$$\zeta_j^{n-k} = \zeta_j^{-k} = \bar{\zeta}_j^k, \qquad 0 \le j, k \le n-1. \tag{3.5}$$

3.2.1 Strang's circulant preconditioner $s(T_n(f))$

Given $T_n(f)$ as in (1.1), the corresponding Strang's preconditioner $\mathbf{s(T_n(f))}$ is defined by (2.1). Using (3.4), (3.5), and then (3.3), we see that the eigenvalues of $s(T_n(f))$ are equal to

$$\begin{aligned}
\lambda_j[s(T_n(f))] &= \sum_{k=0}^{m} t_k \zeta_j^k + \sum_{k=m+1}^{n-1} t_{k-n} \zeta_j^k \\
&= \sum_{k=0}^{m} t_k \zeta_j^k + \sum_{k=1}^{m} t_{-k} \zeta_j^{-k} \\
&= \sum_{k=0}^{m} t_k \zeta_j^k + \sum_{k=-m}^{-1} t_k \zeta_j^k \\
&= s_m[f]\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1.
\end{aligned}$$

Here we assume for simplicity that $n = 2m+1$. If $n = 2m$, then we define the $(m, 0)$th entry in $s(T_n(f))$ to be zero, and the equality still holds.

From Fourier analysis (see Zygmund [89, p. 49], for instance), the partial sum $s_m[f]$ defined in (3.3) is given by the convolution of f with the Dirichlet kernel $\mathcal{D}_m$, i.e.,

$$s_m[f](x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}_m(y) f(x-y) dy \equiv (\mathcal{D}_m * f)(x), \tag{3.6}$$

where

$$\mathcal{D}_k(x) = \frac{\sin(k + \frac{1}{2})x}{\sin(\frac{x}{2})} = \sum_{p=-k}^{k} e^{\mathbf{i}px}, \qquad k = 0, 1, \ldots.$$

Thus the eigenvalues of $s(T_n(f))$ can be expressed as

$$\lambda_j[s(T_n(f))] = (\mathcal{D}_m * f)\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1.$$

3.2.2 T. Chan's circulant preconditioner $c_F(T_n(f))$

Given $T_n(f)$ as in (1.1), the entries in the first column of the corresponding T. Chan's preconditioner $c_F(T_n(f))$ are given by

$$\big[c_F(T_n(f))\big]_{k0} = \frac{(n-k)t_k + k\bar{t}_{n-k}}{n}, \qquad 0 \le k \le n-1;$$

see (2.6). By (3.4) and (3.5) again, the eigenvalues of $c_F(T_n(f))$ are given by

$$\begin{aligned}\lambda_j[c_F(T_n(f))] &= \sum_{k=0}^{n-1} \frac{n-k}{n} t_k \zeta_j^k + \sum_{k=1}^{n-1} \frac{k}{n} \bar{t}_{n-k} \zeta_j^k \\ &= \sum_{k=0}^{n-1} \frac{n-k}{n} t_k \zeta_j^k + \sum_{k=1}^{n-1} \frac{n-k}{n} \bar{t}_k \bar{\zeta}_j^k \\ &= \sum_{k=0}^{n-1} \frac{n-k}{n} t_k \zeta_j^k + \sum_{k=-(n-1)}^{-1} \frac{n-|k|}{n} t_k \zeta_j^k \\ &= \frac{1}{n} \sum_{k=-(n-1)}^{n-1} (n-|k|) t_k \zeta_j^k, \qquad 0 \le j \le n-1.\end{aligned}$$

We note that this is a Cesàro summation process of order 1 for the Fourier series of f; see Zygmund [89, p. 76]. Using the definition of partial sum and after some rearrangements of the terms, we get

$$\lambda_j[c_F(T_n(f))] = \frac{1}{n} \sum_{k=0}^{n-1} s_k[f]\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1.$$

Thus the eigenvalues of $c_F(T_n(f))$ are just the values of the arithmetic mean of the first n partial sums of f at $2\pi j/n$. It is well-known that this arithmetic mean is given by the convolution of f with the Fejér kernel $\mathcal{F}_n$, i.e.,

$$\frac{1}{n} \sum_{k=0}^{n-1} s_k[f](x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{F}_n(y) f(x-y) dy \equiv (\mathcal{F}_n * f)(x), \tag{3.7}$$

where

$$\mathcal{F}_k(x) = \frac{1}{k} \left[\frac{\sin(\frac{kx}{2})}{\sin(\frac{x}{2})} \right]^2 = \sum_{p=-k+1}^{k-1} \frac{k-|p|}{k} e^{\mathbf{i}px}, \qquad k = 1, 2, \ldots; \tag{3.8}$$

see Zygmund [89, p. 88]. Thus the eigenvalues of $c_F(T_n(f))$ can also be expressed as

$$\lambda_j[c_F(T_n(f))] = (\mathcal{F}_n * f)\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1. \tag{3.9}$$

3.2.3 R. Chan's circulant preconditioner $r(T_n(f))$

Given $T_n(f)$ as in (1.1), R. Chan's circulant preconditioner $r(T_n(f))$ has the first column given by

$$\big[r(T_n(f))\big]_{k0} = \begin{cases} t_0, & k = 0, \\ t_k + \bar{t}_{n-k}, & 0 < k \le n-1; \end{cases}$$

see Section 2.4.2. Thus the eigenvalues of $r(T_n(f))$ are given by

$$\begin{aligned}\lambda_j[r(T_n(f))] &= t_0 + \sum_{k=1}^{n-1}(t_k + \bar{t}_{n-k})\zeta_j^k \\ &= t_0 + \sum_{k=1}^{n-1} t_k\zeta_j^k + \sum_{k=1}^{n-1} \bar{t}_k\bar{\zeta}_j^k \\ &= t_0 + \sum_{k=1}^{n-1} t_k\zeta_j^k + \sum_{k=-(n-1)}^{-1} t_k\zeta_j^k \\ &= s_{n-1}[f]\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1.\end{aligned}$$

Using (3.6), we have

$$\lambda_j[r(T_n(f))] = (\mathcal{D}_{n-1} * f)\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1. \tag{3.10}$$

3.2.4 Huckle's circulant preconditioner $h^{(p)}(T_n(f))$

Given any $0 < p < n$, Huckle's circulant preconditioner $h^{(p)}(T_n(f))$ is defined to be the circulant matrix with eigenvalues given by

$$\lambda_j[h^{(p)}(T_n(f))] = \frac{1}{p}\sum_{k=-p}^{p}(p - |k|)t_k\zeta_j^k, \qquad 0 \le j \le n-1;$$

see Section 2.4.1. The sum is also a Cesàro summation process of order 1 for the Fourier series of f. In fact, using (3.7) and after some simplifications, we have

$$\lambda_j[h^{(p)}(T_n(f))] = \frac{1}{p}\sum_{k=0}^{p-1} s_k[f]\Big(\frac{2\pi j}{n}\Big) = (\mathcal{F}_p * f)\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1.$$

3.2.5 Ku and Kuo's preconditioner $K_{(2)}$

One of the preconditioners proposed in Ku and Kuo [59] is the skew-circulant matrix which, using our notation in Section 2.4.2, can be written as

$$K_{(2)}(f) = 2T_n(f) - r(T_n(f)).$$

Notice that if Θ_n is the n-by-n diagonal matrix given by

$$\Theta_n = \mathrm{diag}\left(1, e^{\frac{\pi \mathrm{i}}{n}}, e^{\frac{2\pi \mathrm{i}}{n}}, \ldots, e^{\frac{(n-1)\pi \mathrm{i}}{n}}\right),$$

then $\Theta_n^* K_{(2)}(f)\Theta_n$ is a circulant matrix. Actually, this property holds for any skew-circulant matrix [37]. By (3.4) and (3.5) again, it is then straightforward to verify that

$$\lambda_j(K_{(2)}(f)) = \lambda_j(\Theta_n^* K_{(2)}(f)\Theta_n) = s_{n-1}[f]\Big(\frac{2\pi j}{n} - \frac{\pi}{n}\Big) = (\mathcal{D}_{n-1} * f)\Big(\frac{2\pi j}{n} - \frac{\pi}{n}\Big)$$

for $0 \le j \le n-1$. Comparing this with (3.10), we see that the eigenvalues of $K_{(2)}(f)$ and the eigenvalues of $r(T_n(f))$ are just the values of $\mathcal{D}_{n-1} * f$ sampled at different points in an interval of length 2π.

3.3 Preconditioners from kernels

In this section, we apply the idea explored in Section 3.2 to design other circulant preconditioners from kernels that are commonly used in function theory and signal processing. These kernels are listed in Table 3.1; see Hamming [44], Natanson [65, p. 58], and Walker [85, p. 88].

In the following, we will use the symbol $\mathcal{K}(x)$ to denote a generic kernel defined on $[-\pi, \pi]$. The notation $C_n(\mathcal{K} * f)$ denotes the circulant matrix with eigenvalues given by

$$\lambda_j(C_n(\mathcal{K} * f)) = (\mathcal{K} * f)\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1. \tag{3.11}$$

Using this notation, we can rewrite Strang's, T. Chan's, and R. Chan's circulant preconditioners as

$$s(T_n(f)) = C_n(\mathcal{D}_m * f), \quad c_F(T_n(f)) = C_n(\mathcal{F}_n * f), \quad r(T_n(f)) = C_n(\mathcal{D}_{n-1} * f),$$

respectively.

The first column of $C_n(\mathcal{K} * f)$ can be obtained by using (1.5) if we exactly know the values of

$$(\mathcal{K} * f)\Big(\frac{2\pi j}{n}\Big), \qquad 0 \le j \le n-1,$$

Table 3.1. *Some kernels and their definitions.*

Kernel	$\mathcal{K}(x)$
Modified Dirichlet	$\frac{1}{2}\left[\mathcal{D}_{n-1}(x) + \mathcal{D}_{n-2}(x)\right]$
de la Vallée Poussin	$2\mathcal{F}_{2\lfloor n/2 \rfloor}(x) - \mathcal{F}_{\lfloor n/2 \rfloor}(x)$
von Hann	$\frac{1}{4}\left[\mathcal{D}_{n-1}(x - \frac{\pi}{n}) + 2\mathcal{D}_{n-1}(x) + \mathcal{D}_{n-1}(x + \frac{\pi}{n})\right]$
Hamming	$0.23\left[\mathcal{D}_{n-1}(x - \frac{\pi}{n}) + \mathcal{D}_{n-1}(x + \frac{\pi}{n})\right] + 0.54\mathcal{D}_{n-1}(x)$
Bernstein	$\frac{1}{2}\left[\mathcal{D}_{n-1}(x) + \mathcal{D}_{n-1}(x + \frac{\pi}{n})\right]$

or otherwise by using the following construction process. Let us illustrate the process by using the de la Vallée Poussin's kernel which is defined as

$$\mathcal{K}(x) = 2\mathcal{F}_{2m}(x) - \mathcal{F}_m(x),$$

where $\mathcal{F}_k$ is the Fejér kernel and $m = \lfloor n/2 \rfloor$. For simplicity, let us consider the case where $n = 2m$. By using (3.7), (3.5), and then (3.3), we have

$$\begin{aligned}
(\mathcal{K} * f)\Big(\frac{2\pi j}{n}\Big) &= 2(\mathcal{F}_{2m} * f)\Big(\frac{2\pi j}{n}\Big) - (\mathcal{F}_m * f)\Big(\frac{2\pi j}{n}\Big) \\
&= 2\frac{s_0[f] + \cdots + s_{2m-1}[f]}{2m}\Big(\frac{2\pi j}{n}\Big) - \frac{s_0[f] + \cdots + s_{m-1}[f]}{m}\Big(\frac{2\pi j}{n}\Big) \\
&= \frac{1}{m}\{s_m[f] + \cdots + s_{2m-1}[f]\}\Big(\frac{2\pi j}{n}\Big) \\
&= s_m[f]\Big(\frac{2\pi j}{n}\Big) + \frac{2}{n}\left[\sum_{k=m+1}^{2m-1}(n-k)t_k\zeta_j^k + \sum_{k=m+1}^{2m-1}(n-k)\bar{t}_k\bar{\zeta}_j^k\right] \\
&= \sum_{k=0}^{m}\Big(t_k + \frac{2k}{n}\bar{t}_{n-k}\Big)\zeta_j^k + \sum_{k=m+1}^{2m-1}\Big[\frac{2(n-k)}{n}t_k + \bar{t}_{n-k}\Big]\zeta_j^k.
\end{aligned}$$

Hence the first column of $C_n(\mathcal{K} * f)$ is given by

$$[C_n(\mathcal{K} * f)]_{k0} = \begin{cases} t_k + \dfrac{2k}{n}\bar{t}_{n-k}, & 0 \le k \le m, \\ \dfrac{2(n-k)}{n}t_k + \bar{t}_{n-k}, & m < k \le n-1. \end{cases}$$

Table 3.2 lists the first column of the circulant preconditioners from the kernels in Table 3.1. Since the first entry $[C_n(\mathcal{K} * f)]_{00}$ is always equal to t_0, it is omitted from the table. Algorithm A.3 with `pchoice` equalling 4 to 8 generates the first columns and then the eigenvalues of these preconditioners.

3.4 Clustering properties

In this section, we discuss the convergence property of the preconditioned systems with circulant preconditioners derived from kernels. We will show that if the convolution product $\mathcal{K} * f$ tends to the generating function f uniformly, then the corresponding preconditioned matrix $C_n^{-1}T_n$ will have a clustered spectrum. From Fourier analysis (see Zygmund [89, p. 89]), we know that for the Fejér kernel $\mathcal{F}_n$, $\mathcal{F}_n * f$ tends to f uniformly on $[-\pi, \pi]$ for all f in $\mathbf{C}_{2\pi}$. Hence the preconditioned matrix $(c_F(T_n))^{-1}T_n(f)$ should have a clustered spectrum for all $f \in \mathbf{C}_{2\pi}$. This has already been proven in Corollary 3.2. For general $\mathcal{K}$, we start with the following lemma.

Lemma 3.3. *Let $f \in \mathbf{C}_{2\pi}$ and $\mathcal{K}$ be a kernel such that $\mathcal{K} * f$ tends to f uniformly on $[-\pi, \pi]$. If $C_n(\mathcal{K} * f)$ is the circulant matrix with eigenvalues given by* (3.11)*,*

Table 3.2. *The first column of circulant preconditioners from kernels.*

Kernel	$[C_n(\mathcal{K} * f)]_{k0},\ \ 1 \le k \le n-1$
Modified Dirichlet	$\begin{cases} t_1 + \frac{1}{2}\bar{t}_{n-1}, & k = 1, \\ t_k + \bar{t}_{n-k}, & 2 \le k \le n-2, \\ \frac{1}{2}t_{n-1} + \bar{t}_1, & k = n-1. \end{cases}$
de la Vallée Poussin	$\begin{cases} t_k + \frac{k}{m}\bar{t}_{2m-k}, & 1 \le k \le m, \\ \frac{2m-k}{m}t_k + \bar{t}_{2m-k}, & m < k < 2m,\ m = \lfloor n/2 \rfloor, \\ 0, & k = 2m. \end{cases}$
von Hann	$\cos^2\left(\frac{\pi k}{2n}\right)t_k + \cos^2\left(\frac{\pi(n-k)}{2n}\right)\bar{t}_{n-k}$
Hamming	$\left[0.54 + 0.46\cos^2\left(\frac{\pi k}{n}\right)\right] t_k + \left[0.54 + 0.46\cos^2\left(\frac{\pi(n-k)}{n}\right)\right] \bar{t}_{n-k}$
Bernstein	$\frac{1}{2}\left[(1 + e^{\frac{i\pi k}{n}})t_k + (1 - e^{\frac{i\pi k}{n}})\bar{t}_{n-k}\right]$

then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $n > N$, at most M eigenvalues of $T_n(f) - C_n(\mathcal{K} * f)$ have absolute values greater than ϵ.

Proof. We first write

$$T_n(f) - C_n(\mathcal{K} * f) = [T_n(f) - c_F((T_n(f))] + [c_F(T_n(f)) - C_n(\mathcal{K} * f)],$$

where $c_F(T_n(f)) = C_n(\mathcal{F}_n * f)$ is T. Chan's circulant preconditioner. In view of Theorem 3.1, it suffices to show that

$$\lim_{n\to\infty} \|c_F(T_n(f)) - C_n(\mathcal{K} * f)\|_2 = 0. \tag{3.12}$$

Since $c_F(T_n(f))$ and $C_n(\mathcal{K} * f)$ are both circulant matrices and hence can be diagonalized by the same Fourier matrix F_n, we see that (3.12) is equivalent to

$$\lim_{n\to\infty} \max_{0\le j\le n-1} |\lambda_j(C_n(\mathcal{F}_n * f)) - \lambda_j(C_n(\mathcal{K} * f))| = 0. \tag{3.13}$$

However, by (3.9) and (3.11), we have

$$\begin{aligned}&\max_{0\le j\le n-1}|\lambda_j(C_n(\mathcal{F}_n*f))-\lambda_j(C_n(\mathcal{K}*f))|\\&=\max_{0\le j\le n-1}\left|(\mathcal{F}_n*f)\Big(\frac{2\pi j}{n}\Big)-(\mathcal{K}*f)\Big(\frac{2\pi j}{n}\Big)\right|\\&\le\|\mathcal{F}_n*f-\mathcal{K}*f\|_\infty\le\|\mathcal{F}_n*f-f\|_\infty+\|f-\mathcal{K}*f\|_\infty.\end{aligned}$$

Since $\mathcal{F}_n*f$ and $\mathcal{K}*f$ both converge to f uniformly, (3.13) follows. □

Next we show that if f is positive, then $C_n(\mathcal{K}*f)$ is positive definite and uniformly invertible for large n.

Lemma 3.4. *Let $f\in\mathbf{C}_{2\pi}$ with the minimum value $f_{\min}>0$ and $\mathcal{K}$ be a kernel such that $\mathcal{K}*f$ tends to f uniformly on $[-\pi,\pi]$. If $C_n(\mathcal{K}*f)$ is the circulant matrix with eigenvalues given by* (3.11), *then for all n sufficiently large, we have*

$$\lambda_j(C_n(\mathcal{K}*f))\ge\frac{1}{2}f_{\min}>0,\qquad 0\le j\le n-1.$$

Proof. Since $\mathcal{K}*f$ converges to f uniformly and $f_{\min}>0$, there exists an $N>0$ such that for all $n>N$ and $0\le j\le n-1$,

$$\left|(f-\mathcal{K}*f)\Big(\frac{2\pi j}{n}\Big)\right|\le\|f-\mathcal{K}*f\|_\infty\le\frac{1}{2}f_{\min}.$$

Thus by (3.11), we have

$$\begin{aligned}\lambda_j(C_n(\mathcal{K}*f))&=(\mathcal{K}*f-f)\Big(\frac{2\pi j}{n}\Big)+f\Big(\frac{2\pi j}{n}\Big)\\&\ge f_{\min}-(f-\mathcal{K}*f)\Big(\frac{2\pi j}{n}\Big)\ge\frac{1}{2}f_{\min},\qquad 0\le j\le n-1.\end{aligned}$$ □

Combining Lemmas 3.3 and 3.4, we have the main theorem, namely that the spectrum of $C_n^{-1}T_n$ is clustered around one.

Theorem 3.5. *Let $f\in\mathbf{C}_{2\pi}$ be positive and $\mathcal{K}$ be a kernel such that $\mathcal{K}*f$ tends to f uniformly on $[-\pi,\pi]$. If $C_n(\mathcal{K}*f)$ is the circulant matrix with eigenvalues*

$$\lambda_j(C_n(\mathcal{K}*f))=(\mathcal{K}*f)\Big(\frac{2\pi j}{n}\Big),\qquad 0\le j\le n-1,$$

then for all $\epsilon>0$, there exist M and $N>0$ such that for all $n>N$, at most M eigenvalues of $I_n-C_n^{-1}(\mathcal{K}*f)T_n(f)$ have absolute values greater than ϵ.

Proof. We just note that

$$I_n-C_n^{-1}(\mathcal{K}*f)T_n(f)=C_n^{-1}(\mathcal{K}*f)[C_n(\mathcal{K}*f)-T_n(f)].$$ □

It follows clearly from Theorem 3.5 that the PCG method converges superlinearly. We remark that this unifying approach for the construction of circulant preconditioners transforms the problem of finding preconditioners to the problem of approximating the generating functions. It gives us a guideline for choosing a good preconditioner for a given generating function. According to Theorem 3.5, the circulant preconditioner generated by the delta function δ must be a good preconditioner since $\delta * f = f$. By (3.11), the preconditioner can be written as follows:

$$C_n(\delta * f) = F_n^* \mathrm{diag}\left(f(0), f\Big(\frac{2\pi}{n}\Big), \ldots, f\Big(\frac{2(n-1)\pi}{n}\Big) \right) F_n. \tag{3.14}$$

In Chapter 4, we will use this kernel approach to construct the best circulant preconditioner for ill-conditioned Toeplitz systems.

3.5 Examples

In this section, we test the convergence rate of the preconditioned systems with generating functions given by the Hardy–Littlewood series:

$$H_\alpha(x) = \sum_{k=1}^{\infty} \left(\frac{e^{\mathbf{i}k\log k}}{k^\alpha} e^{\mathbf{i}kx} + \frac{e^{-\mathbf{i}k\log k}}{k^\alpha} e^{-\mathbf{i}kx} \right);$$

see Zygmund [89, p. 197]. It converges uniformly to a function in $\mathbf{C}_{2\pi}$ when $\alpha > 0.5$. In the examples below, we investigate the convergence rate of the preconditioned systems for $\alpha = 1.0$ and 0.5.

We remark that in general, $H_\alpha(x)$ is not a positive function in $[-\pi, \pi]$. In fact, we find numerically that when $n = 512$, the minimum values of the partial sum $s_n[H_\alpha](x)$ are approximately equal to -4.146 and -6.492 for $\alpha = 1.0$ and 0.5, respectively. Thus, in the experiments, we choose the functions $H_1(x) + 4.2$ and $H_{0.5}(x) + 6.5$ as our generating functions. Eight different circulant preconditioners are tested. As before, the right-hand side $\mathbf{b}$ is the vector of all ones. Tables 3.3 and 3.4 show the number of iterations required for convergence. They can be reproduced by running A.1 with three parameters: `n`, the size of the system; `pchoice`, the choice of the preconditioner; and `fchoice`, 2 or 3 for Table 3.3 or 3.4, respectively.

Table 3.3. *Number of iterations for* $f(x) = H_1(x) + 4.2$.

Preconditioner used	n					
	32	64	128	256	512	1024
I	18	27	43	51	58	56
$C_n(\mathcal{D}_m * f)$	9	9	9	9	9	9
$C_n(\mathcal{F}_n * f)$	10	11	11	10	9	9
$C_n(\mathcal{D}_{n-1} * f)$	10	9	9	9	9	9
Modified Dirichlet	10	9	9	9	9	9
de la Vallée Poussin	9	9	9	9	9	9
von Hann	9	9	9	9	9	9
Bernstein	10	10	9	9	9	9
Hamming	9	9	9	9	9	9

Table 3.4. *Number of iterations for* $f(x) = H_{0.5}(x) + 6.5$.

Preconditioner used	n					
	32	64	128	256	512	1024
I	18	29	44	66	67	68
$C_n(\mathcal{D}_m * f)$	11	14	16	16	15	15
$C_n(\mathcal{F}_n * f)$	12	13	14	15	14	15
$C_n(\mathcal{D}_{n-1} * f)$	12	14	16	17	15	18
Modified Dirichlet	12	14	16	16	15	17
de la Vallée Poussin	11	14	15	16	15	15
von Hann	11	12	13	15	15	15
Bernstein	12	14	14	16	15	15
Hamming	11	13	14	16	15	15

From the tables, we see that as n increases, the number of iterations increases for the original matrix T_n, while it stays almost the same for the preconditioned matrices. Moreover, all preconditioned systems converge at the same rate for large n. We also see that the convergence rate depends on the degree of smoothness of the generating function. Finally, we notice that for small n, some of the preconditioners may have negative eigenvalues. However, it is interesting to note that the PCG method still converges in these cases.

Chapter 4

Ill-conditioned Toeplitz systems

In Sections 2.4.4 and 2.4.5, the band-Toeplitz preconditioner B_n and the $\{\omega\}$-circulant preconditioner P_n are proposed for solving ill-conditioned problems. It was proved in [28, 68] that they work well for some ill-conditioned Toeplitz systems with the generating function f having finitely many zeros. The basic idea behind these preconditioners is to find a function g that matches the zeros of f. However, the major drawback of these preconditioners is that they need f explicitly. For instance, to form P_n in (2.11), we need to know f in order to construct a Λ_n defined by (2.12). Similarly, in (3.14), δ is a good kernel to construct a good circulant preconditioner. However, we also need f explicitly. In contrast, to form Strang's preconditioner or T. Chan's preconditioner from a given Toeplitz matrix T_n, we need only the entries $\{t_j\}_{|j|<n}$ from the matrix. We do not need to know all the Fourier coefficients of f or the function f itself.

In this chapter, a family of new circulant preconditioners is derived for ill-conditioned Toeplitz systems with the generating function having a single zero. These preconditioners are called the best circulant preconditioners in [32]. The idea is to look for some sequences of trigonometric polynomials converging to the delta function δ. Then we use those polynomials as kernels to construct the preconditioners as we did in Chapter 3. More precisely, we will construct preconditioners by approximating f with the convolution product $\mathcal{K} * f$ that matches the zero of f and depends only on $\{t_j\}_{|j|<n}$.

4.1 Construction of preconditioner

Let $T_n(f)$ be an n-by-n Hermitian positive definite Toeplitz matrix with entries defined again by the Fourier coefficients of a function $f \in \mathbf{C}_{2\pi}^{+}$,

$$t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-\mathbf{i}kx} \, dx, \qquad k = 0, \pm 1, \pm 2, \ldots.$$

We recall that x_0 is a zero of f of order q if $f(x_0) = 0$ and q is the smallest positive even integer such that $f^{(q)}(x_0) \neq 0$ and $f^{(q+1)}(x)$ is continuous in a neighborhood of x_0. We remark that the condition number of $T_n(f)$ generated by such a function f grows like $O(n^q)$; see Theorem 1.13 and [71]. In this chapter, we will consider only f having a single zero. For the general case where f has more than one zero, we refer readers to [26].

In the following, we will use the generalized Jackson kernels

$$\mathcal{K}_{m,2r}(x) \equiv \frac{k_{m,2r}}{m^{2r-1}} \left[\frac{\sin(\frac{mx}{2})}{\sin(\frac{x}{2})} \right]^{2r}, \qquad r = 1, 2, \ldots, \tag{4.1}$$

to construct the circulant preconditioners. Here $k_{m,2r}$ is a normalization constant such that

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(x) dx = 1.$$

It is known that $k_{m,2r}$ is bounded above and below by constants independent of m; see [64, p. 57] or (4.10) below. Note that $\mathcal{K}_{m,2}(x)$ is the Fejér kernel $\mathcal{F}_m$ and $\mathcal{K}_{m,4}(x)$ is called the Jackson kernel; see [64].

For any m, the Fejér kernel $\mathcal{K}_{m,2}(x) = \mathcal{F}_m$ can be expressed as

$$\mathcal{K}_{m,2}(x) = \sum_{k=-m+1}^{m-1} b_k^{(m,2)} e^{\mathbf{i}kx},$$

where

$$b_k^{(m,2)} = \frac{m - |k|}{m}, \qquad |k| \le (m-1);$$

see (3.8). By (4.1), we see that $\mathcal{K}_{m,2r}(x)$ is the rth power of $\mathcal{K}_{m,2}(x)$ up to a scaling. Hence we have

$$\mathcal{K}_{m,2r}(x) = \sum_{k=-r(m-1)}^{r(m-1)} b_k^{(m,2r)} e^{\mathbf{i}kx}, \tag{4.2}$$

where the coefficients $b_k^{(m,2r)}$ can be obtained by convoluting the vector

$$\left(b_{-m+1}^{(m,2)}, \ldots, b_0^{(m,2)}, \ldots, b_{m-1}^{(m,2)} \right)^T = \left(\frac{1}{m}, \frac{2}{m}, \ldots, 1, \ldots, \frac{2}{m}, \frac{1}{m} \right)^T$$

with itself for $r-1$ times. This can be done by FFTs; see [75, pp. 294–296] and also Algorithm A.4 in the appendix. Note that the cost of computing the coefficients $\{b_k^{(m,2r)}\}$ for all $|k| \le r(m-1)$ is of order $O(rm \log m)$ operations.

We recall that the convolution product of two arbitrary functions

$$g(x) = \sum_{k=-\infty}^{\infty} b_k e^{\mathbf{i}kx} \quad \text{and} \quad h(x) = \sum_{k=-\infty}^{\infty} c_k e^{\mathbf{i}kx}$$

in $\mathbf{C}_{2\pi}$ can be written as

$$(g * h)(x) \equiv \frac{1}{2\pi} \int_{-\pi}^{\pi} g(t) h(x-t) dt = \sum_{k=-\infty}^{\infty} b_k c_k e^{\mathbf{i}kx}. \tag{4.3}$$

For a given n-by-n Toeplitz matrix $T_n(f)$, the proposed circulant preconditioner is $C_n(\mathcal{K}_{m,2r} * f)$, where $m = \lceil n/r \rceil$, i.e.,

$$r(m-1) < n \le rm. \tag{4.4}$$

By (4.2) and (4.3), since $f(x) = \sum_{k=-\infty}^{\infty} t_k e^{\mathbf{i}kx}$, the convolution product is given by

$$(\mathcal{K}_{m,2r} * f)(x) = \sum_{k=-r(m-1)}^{r(m-1)} t_k b_k^{(m,2r)} e^{\mathbf{i}kx} = \sum_{k=-n+1}^{n-1} d_k e^{\mathbf{i}kx}, \tag{4.5}$$

where

$$d_k = \begin{cases} t_k b_k^{(m,2r)}, & |k| \le r(m-1), \\ 0 & \text{otherwise.} \end{cases}$$

By (3.11), the eigenvalues of the preconditioner $C_n(\mathcal{K}_{m,2r} * f)$ are given by

$$\begin{aligned} \lambda_j(C_n(\mathcal{K}_{m,2r} * f)) = (\mathcal{K}_{m,2r} * f)\Big(\frac{2\pi j}{n}\Big) &= \sum_{k=-n+1}^{n-1} d_k e^{2\pi \mathbf{i} jk/n} \\ &= \sum_{k=0}^{n-1} (d_k + d_{-n+k}) e^{2\pi \mathbf{i} jk/n}, \qquad 0 \le j \le n-1. \end{aligned}$$

Using (1.5) and (1.6), the kth entry of the first column of the preconditioner is just $d_k + d_{-n+k}$. Recalling that the cost of computing all $b_k^{(m,2r)}$ is of order $O(rm \log m)$ operations, we see that the cost of constructing $C_n(\mathcal{K}_{m,2r} * f)$ is of $O(n \log n)$ operations and it requires only the entries $\{t_j\}_{|j|<n}$ from the given n-by-n Toeplitz matrix T_n. See Algorithm A.3 for `pchoice` equalling 9 to 11, where `b` in A.3 are computed by A.4 using (4.2). We remark that $\mathcal{K}_{m,2}$ is the Fejér kernel $\mathcal{F}_n$, and hence by (3.9), $C_n(\mathcal{K}_{m,2} * f)$ is just T. Chan's preconditioner.

4.2 Properties of generalized Jackson kernel

In this section, we study some properties of $\mathcal{K}_{m,2r}$ in order to see how good the approximation of f by $\mathcal{K}_{m,2r} * f$ will be. These properties are useful in the spectral analysis of the circulant preconditioners in Section 4.3. First, we claim that the preconditioners are positive definite.

Lemma 4.1. *The preconditioner $C_n(\mathcal{K}_{m,2r} * f)$ is positive definite for $f \in \mathbf{C}_{2\pi}^{+}$ and for all positive integers m, n, and r.*

Proof. Just note that by (4.1), $\mathcal{K}_{m,2r}(x)$ is positive except at $x = 2k\pi/m$, $k = \pm 1, \pm 2, \ldots, \pm(n-1)$, and $f \in \mathbf{C}_{2\pi}^{+}$ is nonnegative and not identically zero. Hence

$$(\mathcal{K}_{m,2r} * f)(x) > 0,$$

and the preconditioner $C_n(\mathcal{K}_{m,2r} * f)$ is positive definite by (3.11). □

For simplicity, we will use x to denote the function x defined on $\mathbb{R}$ in the following. For clarity, we will use $x_{2\pi}$ to denote the periodic extension of x on $[-\pi, \pi]$, i.e.,

$$x_{2\pi}(x) = \tilde{x} \qquad \text{if } \ x = \tilde{x} \ (\mathrm{mod}\ 2\pi) \ \text{ and } \ \tilde{x} \in [-\pi, \pi];$$

see Figure 4.1 below. It is clear that $x_{2\pi}^{2p} \in \mathbf{C}_{2\pi}^{+}$ for any integer p. We first show that $\mathcal{K}_{m,2r} * x_{2\pi}^{2p}$ matches the order of the zero of $x_{2\pi}^{2p}$ at $x = 0$ if $r > p$.

Lemma 4.2. *Let p and r be any integers with $r > p > 0$. Then*

$$\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(0) = \left(\mathcal{K}_{m,2r} * x^{2p}\right)(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t) t^{2p} dt = \frac{c_{p,2r}}{m^{2p}}, \tag{4.6}$$

where

$$\frac{2^{2p-1}}{2p+1}\left(\frac{2}{\pi}\right)^{4r} \le c_{p,2r} \le 2^{2p+1}\left(\frac{\pi}{2}\right)^{4r}. \tag{4.7}$$

Proof. The first two equalities in (4.6) are trivial by the definition of $x_{2\pi}$. For the last equality, since

$$\frac{x}{\pi} \le \sin\left(\frac{x}{2}\right) \le \frac{x}{2}, \qquad x \in [0, \pi],$$

we have by (4.1),

$$\begin{aligned}
\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t) t^{2p} dt &\le \frac{2\pi^{2r} k_{m,2r}}{m^{2r-1}} \int_0^{\pi} \frac{\sin^{2r}\left(\frac{mt}{2}\right)}{t^{2r-2p}} dt \\
&= \frac{2^{2p-2r+2}\pi^{2r} k_{m,2r}}{m^{2p}} \int_0^{\frac{m\pi}{2}} \frac{\sin^{2r} v}{v^{2r-2p}} dv \\
&\le \frac{2^{2p+2} k_{m,2r}}{m^{2p}} \left(\frac{\pi}{2}\right)^{2r} \left(\int_0^1 \frac{\sin^{2r} v}{v^{2r-2p}} dv + \int_1^{\infty} \frac{\sin^{2r} v}{v^{2r-2p}} dv\right) \\
&\le \frac{2^{2p+2} k_{m,2r}}{m^{2p}} \left(\frac{\pi}{2}\right)^{2r} \left(\int_0^1 v^{2p} dv + \int_1^{\infty} \frac{1}{v^{2r-2p}} dv\right) \\
&\le \frac{2^{2p+3} k_{m,2r}}{m^{2p}} \left(\frac{\pi}{2}\right)^{2r}. \qquad (4.8)
\end{aligned}$$

Similarly, we also have

$$\begin{aligned}
\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t) t^{2p} dt &\ge \frac{2^{2r+1} k_{m,2r}}{m^{2r-1}} \int_0^{\pi} \frac{\sin^{2r}\left(\frac{mt}{2}\right)}{t^{2r-2p}} dt \qquad (4.9) \\
&\ge \frac{2^{2p+2} k_{m,2r}}{m^{2p}} \int_0^1 \frac{\sin^{2r} v}{v^{2r-2p}} dv
\end{aligned}$$

$$\geq \frac{2^{2p+2}k_{m,2r}}{m^{2p}}\left(\frac{2}{\pi}\right)^{2r}\int_0^1 v^{2p}dv$$
$$= \frac{2^{2p+2}k_{m,2r}}{(2p+1)m^{2p}}\left(\frac{2}{\pi}\right)^{2r}.$$

By setting $p = 0$ in (4.8) and (4.9), we obtain

$$4\left(\frac{2}{\pi}\right)^{2r} k_{m,2r} \leq 2\pi = \int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)dt \leq 8\left(\frac{\pi}{2}\right)^{2r} k_{m,2r}.$$

Thus

$$\frac{2\pi}{8}\left(\frac{2}{\pi}\right)^{2r} \leq k_{m,2r} \leq \frac{2\pi}{4}\left(\frac{\pi}{2}\right)^{2r}. \tag{4.10}$$

Putting (4.10) back into (4.8) and (4.9), we have (4.7). □

We remark that using the same arguments as in (4.9), we can show that for $p \geq 1$,

$$\left(\mathcal{K}_{m,2} * x^{2p}\right)(0) \geq O\left(\frac{1}{m}\right); \tag{4.11}$$

i.e., T. Chan's preconditioner does not match the order of the zeros of x^{2p} at $x = 0$ when $p \geq 1$. We will see in Section 4.4 that T. Chan's preconditioner does not work for Toeplitz matrices generated by functions with zeros of order $2p$, where $p \geq 1$.

Next we estimate $(\mathcal{K}_{m,2r} * x_{2\pi}^{2p})(y)$ for $y \neq 0$. To this end, we first have to replace the function $x_{2\pi}^{2p}$ in the convolution product by x^{2p} defined on $\mathbb{R}$.

Lemma 4.3. *Let $p > 0$ be an integer. Then*

$$\pi^{2p} \leq \frac{\left[\mathcal{K}_{m,2r} * x^{2p}(x+2\pi)^{2p}\right](y)}{\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y)} \leq \left(\frac{5\pi}{2}\right)^{2p} \tag{4.12}$$

for any $y \in [-\pi, -\pi/2]$;

$$1 \leq \frac{\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y)}{\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y)} \leq 3^{2p} \tag{4.13}$$

for any $y \in [-\pi/2, \pi/2]$; and

$$\pi^{2p} \leq \frac{\left[\mathcal{K}_{m,2r} * x^{2p}(x-2\pi)^{2p}\right](y)}{\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y)} \leq \left(\frac{5\pi}{2}\right)^{2p} \tag{4.14}$$

for any $y \in [\pi/2, \pi]$.

Proof. To prove (4.12), we first claim that

$$\pi^{2p} \le \frac{(y-t)^{2p}(y+2\pi-t)^{2p}}{(y-t)_{2\pi}^{2p}} \le \left(\frac{5\pi}{2}\right)^{2p}, \tag{4.15}$$

where $t \in [-\pi, \pi]$ and $y \in [-\pi, -\pi/2]$. By the definition of $(y-t)_{2\pi}^{2p}$, we have (see Figure 4.1)

$$\frac{(y-t)^{2p}(y+2\pi-t)^{2p}}{(y-t)_{2\pi}^{2p}} = \begin{cases} (y+2\pi-t)^{2p}, & t \in [-\pi, y+\pi], \\ (y-t)^{2p}, & t \in [y+\pi, \pi]. \end{cases}$$

For $t \in [-\pi, y+\pi]$ and $y \in [-\pi, -\pi/2]$, we have

$$\pi^{2p} = (y+2\pi-(y+\pi))^{2p} \le (y+2\pi-t)^{2p} \le (y+3\pi)^{2p} \le \left(\frac{5\pi}{2}\right)^{2p}.$$

For $t \in [y+\pi, \pi]$ and $y \in [-\pi, -\pi/2]$, we have

$$\pi^{2p} = (y-(y+\pi))^{2p} \le (y-t)^{2p} \le (y-\pi)^{2p} \le (2\pi)^{2p}.$$

Thus we obtain (4.15).

We see that by using (4.15),

$$\begin{aligned} \left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y) &\equiv \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)(y-t)_{2\pi}^{2p} dt \\ &\le \frac{1}{\pi^{2p}}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)(y-t)^{2p}(y+2\pi-t)^{2p} dt\right] \\ &= \frac{1}{\pi^{2p}}\left[\mathcal{K}_{m,2r} * x^{2p}(x+2\pi)^{2p}\right](y). \end{aligned}$$

Similarly, we also have

$$\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y) \ge \left(\frac{2}{5\pi}\right)^{2p}\left[\mathcal{K}_{m,2r} * x^{2p}(x+2\pi)^{2p}\right](y).$$

Thus we obtain (4.12).

To prove (4.13), we just note that

$$1 \le \frac{(y-t)^{2p}}{(y-t)_{2\pi}^{2p}} \le 3^{2p},$$

where $t \in [-\pi, \pi]$ and $y \in [-\pi/2, \pi/2]$. As for (4.14), we have

$$\pi^{2p} \le \frac{(y-t)^{2p}(y-2\pi-t)^{2p}}{(y-t)_{2\pi}^{2p}} \le \left(\frac{5\pi}{2}\right)^{2p},$$

where $t \in [-\pi, \pi]$ and $y \in [\pi/2, \pi]$. □

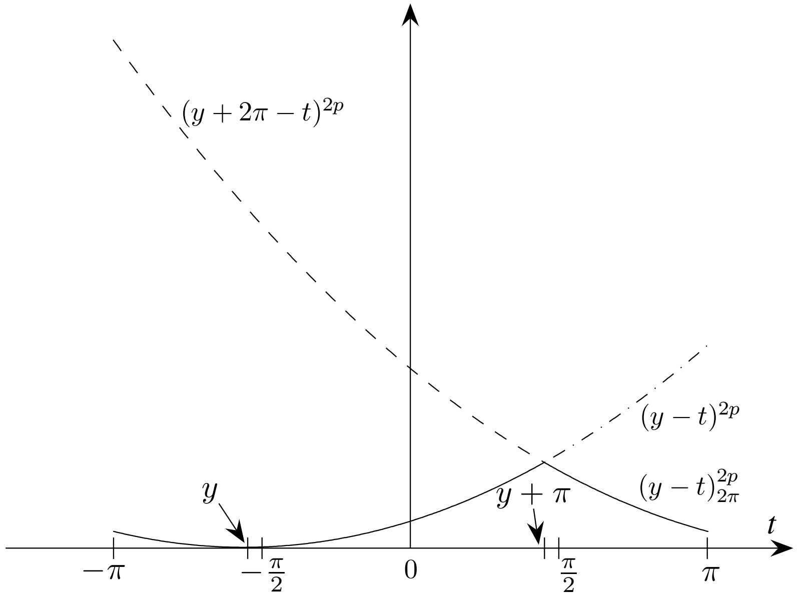

Figure 4.1. *The functions $(y-t)^{2p}_{2\pi}$, $(y-t)^{2p}$, and $(y+2\pi-t)^{2p}$.*

By using Lemmas 4.2 and 4.3, we can show that $\mathcal{K}_{m,2r} * x^{2p}_{2\pi}$ and $x^{2p}_{2\pi}$ are essentially the same away from the zero of $x^{2p}_{2\pi}$.

Theorem 4.4. *Let p and r be any integers with $r > p > 0$ and $m = \lceil n/r \rceil$. Then there exist α and β with $\beta > \alpha > 0$ independent of n such that for all sufficiently large n and $\pi/n \le |y| \le \pi$, we have*

$$\alpha \le \frac{\left(\mathcal{K}_{m,2r} * x^{2p}_{2\pi}\right)(y)}{y^{2p}_{2\pi}} \le \beta. \tag{4.16}$$

Proof. We see from Lemma 4.3 that for different values of y, $(\mathcal{K}_{m,2r} * x^{2p}_{2\pi})(y)$ can be replaced by different functions. Hence, we separate the proof for different ranges of values of y. We first consider $y \in [\pi/n, \pi/2]$. By the binomial expansion,

$$\begin{aligned}\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y) &\equiv \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)(y-t)^{2p}dt \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)\sum_{k=0}^{2p}\binom{2p}{k} y^{2p-k}(-t)^k dt.\end{aligned}$$

Note that $\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)t^k dt = 0$ for odd k. Thus

$$\frac{\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y)}{y_{2\pi}^{2p}} = \frac{\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y)}{y^{2p}} = \frac{1}{2\pi}\sum_{k=0}^{p}\binom{2p}{2k} y^{-2k}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)t^{2k}dt.$$

By (4.6), we then have

$$\frac{\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y)}{y_{2\pi}^{2p}} = \frac{1}{2\pi}\sum_{k=0}^{p}\binom{2p}{2k}\frac{c_{k,2r}}{y^{2k}m^{2k}}, \tag{4.17}$$

where by (4.7), $c_{k,2r}$ are bounded above and below by positive constants independent of m for $k = 0, \ldots, p$. By (4.4) and $\pi/n \le y$, we have

$$\frac{\pi}{r} \le \frac{\pi m}{n} \le ym. \tag{4.18}$$

Hence we obtain by (4.17) and (4.18),

$$c_{0,2r} \le \frac{\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y)}{y_{2\pi}^{2p}} \le \frac{1}{2\pi}\sum_{k=0}^{p}\left(\frac{r}{\pi}\right)^{2k}\binom{2p}{2k}c_{k,2r}.$$

Therefore, (4.16) follows for $y \in [\pi/n, \pi/2]$ by using (4.13). The case with $y \in [-\pi/2, -\pi/n]$ is similar to the case where $y \in [\pi/n, \pi/2]$.

Next we consider $y \in [\pi/2, \pi]$. Note that

$$\begin{aligned}\left[\mathcal{K}_{m,2r} * x^{2p}(x-2\pi)^{2p}\right](y) &\equiv \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)(t-y)^{2p}(t-y+2\pi)^{2p}dt \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)\left(y^{2p}(2\pi-y)^{2p}+q(t)\right)dt,\end{aligned}$$

where

$$q(t) = (t-y)^{2p}(t-y+2\pi)^{2p} - y^{2p}(2\pi-y)^{2p} \equiv \sum_{j=1}^{4p} q_j t^j$$

is a $4p$th degree polynomial without the constant term. We have by (4.6) again,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)q(t)dt = \sum_{j=1}^{2p} q_{2j}\frac{c_{2j,2r}}{m^{2j}}.$$

Thus by using the fact that $\frac{1}{2\pi}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)dt = 1$, we obtain

$$\left[\mathcal{K}_{m,2r} * x^{2p}(x-2\pi)^{2p}\right](y) = y^{2p}(2\pi-y)^{2p} + \sum_{j=1}^{2p} q_{2j}\frac{c_{2j,2r}}{m^{2j}}. \tag{4.19}$$

Since $(\pi/2)^{2p} \leq y_{2\pi}^{2p}$ for $y \in [\pi/2, \pi]$, we have

$$\begin{aligned}\frac{\left[\mathcal{K}_{m,2r} * x^{2p}(x-2\pi)^{2p}\right](y)}{y_{2\pi}^{2p}} &\leq (2\pi - y)^{2p} + \left(\frac{2}{\pi}\right)^{2p} \sum_{j=1}^{2p} |q_{2j}| c_{2j,2r} \\ &\leq \left(\frac{3\pi}{2}\right)^{2p} + \left(\frac{2}{\pi}\right)^{2p} \sum_{j=1}^{2p} |q_{2j}| c_{2j,2r}, \qquad (4.20)\end{aligned}$$

which is clearly bounded independent of n. For the lower bound, by using the fact that $\pi^{2p} \geq y_{2\pi}^{2p}$ for $y \in [\pi/2, \pi]$ in (4.19), we have

$$\begin{aligned}\frac{\left[\mathcal{K}_{m,2r} * x^{2p}(x-2\pi)^{2p}\right](y)}{y_{2\pi}^{2p}} &\geq (y - 2\pi)^{2p} + \frac{1}{\pi^{2p}} \sum_{j=1}^{2p} q_{2j} \frac{c_{2j,2r}}{m^{2j}} \\ &\geq \pi^{2p} + \frac{1}{\pi^{2p}} \sum_{j=1}^{2p} q_{2j} \frac{c_{2j,2r}}{m^{2j}}. \qquad (4.21)\end{aligned}$$

Clearly for sufficiently large n (and hence large m), the last expression is bounded uniformly from below by $\pi^{2p}/2$. Combining (4.20), (4.21), and (4.14), we see that (4.16) holds for $y \in [\pi/2, \pi]$ and for n sufficiently large. The case where $y \in [-\pi, -\pi/2]$ can be proved in a similar way. □

Using the fact that

$$\begin{aligned}\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)(y - z - t)_{2\pi}^{2p} dt \\ &= \left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y - z),\end{aligned}$$

we obtain the following corollary, which deals with functions having a zero at $z \neq 0$.

Corollary 4.5. *Let $z \in [-\pi, \pi]$. Let p and r be any integers with $r > p > 0$ and $m = \lceil n/r \rceil$. Then there exist α and β with $\beta > \alpha > 0$ independent of n such that for all sufficiently large n and $\pi/n \leq |y - z| \leq \pi$, we have*

$$\alpha \leq \frac{\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y)}{(y-z)_{2\pi}^{2p}} \leq \beta.$$

Now we can extend the results in Theorem 4.4 to any functions in $\mathbf{C}_{2\pi}^{+}$ with a single zero of order $2p$.

Theorem 4.6. *Let $f \in \mathbf{C}_{2\pi}^{+}$ with a zero of order $2p$ at $z \in [-\pi, \pi]$. Let $r > p$ be any integer and $m = \lceil n/r \rceil$. Then there exist α and β with $\beta > \alpha > 0$ independent of n such that for all sufficiently large n and $\pi/n \leq |y - z| \leq \pi$, we have*

$$\alpha \leq \frac{\left(\mathcal{K}_{m,2r} * f\right)(y)}{f(y)} \leq \beta.$$

Proof. By the definition of zeros, $f(x) = (x-z)_{2\pi}^{2p} g(x)$ for some positive continuous function $g(x)$ on $[-\pi, \pi]$. Write

$$\frac{(\mathcal{K}_{m,2r} * f)(y)}{f(y)} = \frac{\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p} g(x)\right](y)}{\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y)} \cdot \frac{\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y)}{(y-z)_{2\pi}^{2p}} \cdot \frac{1}{g(y)}.$$

Clearly the last factor is uniformly bounded above and below by positive constants. By Corollary 4.5, the same holds for the second factor when $\pi/n \le |y-z| \le \pi$. As for the first factor, by the mean value theorem for integrals, there exists a $\zeta \in [-\pi, \pi]$ such that

$$\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p} g(x)\right](y) = g(\zeta)\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y).$$

Hence for all $y \in [-\pi, \pi]$, we have

$$0 < g_{\min} \le \frac{\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p} g(x)\right](y)}{\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y)} \le g_{\max},$$

where $g_{\min}$ and $g_{\max}$ are the minimum and maximum values of g, respectively. Thus the theorem follows. □

Up to now, we have considered only the interval $\pi/n \le |y-z| \le \pi$. For $|y-z| \le \pi/n$, we can show that at the zero of f, the convolution product $\mathcal{K}_{m,2r} * f$ matches the order of the zero of f.

Theorem 4.7. *Let $f \in \mathbf{C}_{2\pi}^{+}$ with a zero of order $2p$ at $z \in [-\pi, \pi]$. Let $r > p$ be any integer and $m = \lceil n/r \rceil$. Then for any $|y-z| \le \pi/n$, we have*

$$(\mathcal{K}_{m,2r} * f)(y) = O\left(\frac{1}{n^{2p}}\right).$$

Proof. We first prove the theorem for $f(x) = x_{2\pi}^{2p}$. By the binomial theorem,

$$\begin{aligned}\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y) &\equiv \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)(y-t)^{2p} dt \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)\sum_{j=0}^{2p}\binom{2p}{j} y^{2p-j}(-t)^j dt.\end{aligned}$$

Since $\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)t^j dt = 0$ for odd j, we have

$$\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(t)\sum_{j=0}^{p}\binom{2p}{2j} y^{2p-2j} t^{2j} dt. \tag{4.22}$$

By using (4.22), (4.6), (4.7), and then (4.4), we have for $|y| \leq \pi/n$,

$$\begin{aligned}\left(\mathcal{K}_{m,2r} * x^{2p}\right)(y) &\leq \frac{1}{2\pi}\sum_{j=0}^{p}\binom{2p}{2j}\left(\frac{\pi}{n}\right)^{2p-2j}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)t^{2j}dt \\ &\leq \frac{1}{n^{2p}}\sum_{j=0}^{p}\binom{2p}{2j}r^{2j}\pi^{2p-2j}c_{j,2r} = O\left(\frac{1}{n^{2p}}\right).\end{aligned}$$

Hence by (4.13),

$$\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y) \leq O\left(\frac{1}{n^{2p}}\right).$$

Similarly, from (4.22), (4.6), (4.7), and then (4.4), we have

$$(\mathcal{K}_{m,2r} * x^{2p})(y) \geq \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathcal{K}_{m,2r}(t)t^{2p}dt = \frac{c_{p,2r}}{m^{2p}} = O\left(\frac{1}{n^{2p}}\right).$$

Hence by (4.13) again,

$$\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y) \geq O\left(\frac{1}{n^{2p}}\right).$$

Thus the theorem holds for $f(x) = x_{2\pi}^{2p}$.

In the general case where $f(x) = (x-z)_{2\pi}^{2p}g(x)$ for some positive function $g \in \mathbf{C}_{2\pi}$, by the mean value theorem for integrals, there exists a $\zeta \in [-\pi, \pi]$ such that

$$\begin{aligned}(\mathcal{K}_{m,2r} * f)(y) &= \left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}g(x)\right](y) \\ &= g(\zeta)\left[\mathcal{K}_{m,2r} * (x-z)_{2\pi}^{2p}\right](y) \\ &= g(\zeta)\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y-z).\end{aligned}$$

Hence

$$g_{\min}\cdot\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y-z) \leq (\mathcal{K}_{m,2r} * f)(y) \leq g_{\max}\cdot\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y-z)$$

for all $y \in [-\pi, \pi]$. From the first part of the proof, we already see that

$$\left(\mathcal{K}_{m,2r} * x_{2\pi}^{2p}\right)(y-z) = O\left(\frac{1}{n^{2p}}\right)$$

for all $|y-z| \leq \pi/n$. Hence the theorem follows. □

4.3 Spectral analysis

In this section, we study the spectrum of the preconditioned system.

4.3.1 Functions with a zero

In this subsection, we analyze the spectra of the preconditioned matrices when the generating function has a zero. We need the following lemma. Its proof is similar to that of Theorem 2.16. We therefore omit it.

Lemma 4.8. *If f and $g \in \mathbf{C}_{2\pi}^{+}$ are such that $0 < \alpha \leq f/g \leq \beta$ for some constants α and β, then for all n and all nonzero $\mathbf{v} \in \mathbb{C}^n$,*

$$\alpha \leq \frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* T_n(g)\mathbf{v}} \leq \beta.$$

The next theorem states that the spectra of the preconditioned matrices are essentially bounded.

Theorem 4.9. *Let $f \in \mathbf{C}_{2\pi}^{+}$ with a zero of order $2p$ at z. Let $r > p$ and $m = \lceil n/r \rceil$. Then there exist α and β with $\beta > \alpha > 0$ independent of n such that for all sufficiently large n, at most $2p+1$ eigenvalues of $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ are outside the interval $[\alpha, \beta]$.*

Proof. For any function $g \in \mathbf{C}_{2\pi}$, let $\tilde{C}_n(g)$ be the n-by-n circulant matrix with the jth eigenvalue given by

$$\lambda_j(\tilde{C}_n(g)) = \begin{cases} \dfrac{1}{n^{2p}} & \text{if } \left| \dfrac{2\pi j}{n} - z \right| < \dfrac{\pi}{n}, \\ g\left(\dfrac{2\pi j}{n}\right) & \text{otherwise} \end{cases} \tag{4.23}$$

for $j = 0, \ldots, n-1$. Since there is at most one j such that $|2\pi j/n - z| < \pi/n$, by (3.14), $\tilde{C}_n(g) - C_n(g)$ is a matrix of rank at most 1.

By the assumption, we have

$$f(x) = \sin^{2p}\left(\frac{x-z}{2}\right) g(x)$$

for some positive function g in $\mathbf{C}_{2\pi}$. We use the following decomposition of the Rayleigh quotient to prove the theorem:

$$\begin{aligned} \frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} &= \frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* T_n\left[\sin^{2p}\left(\frac{x-z}{2}\right)\right]\mathbf{v}} \cdot \frac{\mathbf{v}^* T_n\left[\sin^{2p}\left(\frac{x-z}{2}\right)\right]\mathbf{v}}{\mathbf{v}^* \tilde{C}_n\left[\sin^{2p}\left(\frac{x-z}{2}\right)\right]\mathbf{v}} \\ &\quad \cdot \frac{\mathbf{v}^* \tilde{C}_n\left[\sin^{2p}\left(\frac{x-z}{2}\right)\right]\mathbf{v}}{\mathbf{v}^* \tilde{C}_n(f)\mathbf{v}} \cdot \frac{\mathbf{v}^* \tilde{C}_n(f)\mathbf{v}}{\mathbf{v}^* \tilde{C}_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} \\ &\quad \cdot \frac{\mathbf{v}^* \tilde{C}_n(\mathcal{K}_{m,2r} * f)\mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} \end{aligned} \tag{4.24}$$

for all nonzero $\mathbf{v} \in \mathbb{C}^n$. We remark that by Lemma 4.8 and the definitions of C_n in (3.14) and $\tilde{C}_n$ in (4.23), all matrices in the factors in the right-hand side of (4.24) are positive definite.

As g is a positive function in $\mathbf{C}_{2\pi}$, by Lemma 4.8, the first factor in the right-hand side of (4.24) is uniformly bounded above and below. Similarly, by (4.23), the third factor is also uniformly bounded. The eigenvalues of the two circulant matrices in the fourth factor differ only when $|2\pi j/n - z| \geq \pi/n$. But by Theorem 4.6, the ratios of these eigenvalues are all uniformly bounded when n is large. The eigenvalues of the two circulant matrices in the last factor differ only when $|2\pi j/n - z| < \pi/n$. But by Theorem 4.7, their ratios are also uniformly bounded.

It remains to handle the second factor in (4.24). Defining $s_{2p}(x) \equiv \sin^{2p}(\frac{x-z}{2})$, we have

$$s_{2p}(x) = \frac{1}{2^p}[1 - \cos(x-z)]^p = \frac{1}{2^p}\Big(-\frac{1}{2}e^{\mathbf{i}z}e^{-\mathbf{i}x} + 1 - \frac{1}{2}e^{-\mathbf{i}z}e^{\mathbf{i}x}\Big)^p;$$

i.e., $s_{2p}(x)$ is a pth degree trigonometric polynomial in x. Note that for $n \geq 2p$,

$$(\mathcal{D}_{\lfloor n/2 \rfloor} * s_{2p})(y) = s_{2p}(y)$$

for all $y \in [-\pi, \pi]$. Therefore,

$$C_n[\mathcal{D}_{\lfloor n/2 \rfloor} * s_{2p}(x)] = C_n[s_{2p}(x)]$$

is Strang's circulant preconditioner for $T_n[s_{2p}(x)]$ when $n \geq 2p$; see Section 3.2.1. As $s_{2p}(x)$ is a pth degree trigonometric polynomial, $T_n[s_{2p}(x)]$ is a band-Toeplitz matrix with half bandwidth $p+1$. Therefore, when $n \geq 2p$, by the definition of Strang's preconditioner,

$$C_n\left[s_{2p}(x)\right] = T_n\left[s_{2p}(x)\right] + \begin{pmatrix} \mathbf{0} & \mathbf{0} & R_p \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ R_p^* & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{4.25}$$

where R_p is a p-by-p matrix. Thus

$$T_n[s_{2p}(x)] = \tilde{C}_n[s_{2p}(x)] + R_n,$$

where R_n is an n-by-n matrix with $\operatorname{rank}(R_n) \leq 2p+1$.

Putting this back into the numerator of the second factor in (4.24), we have for all nonzero $\mathbf{v} \in \mathbb{C}^n$,

$$\begin{aligned} &\frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} \\ &= \frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* T_n\left[s_{2p}(x)\right]\mathbf{v}} \cdot \frac{\mathbf{v}^* \tilde{C}_n\left[s_{2p}(x)\right]\mathbf{v}}{\mathbf{v}^* \tilde{C}_n(f)\mathbf{v}} \cdot \frac{\mathbf{v}^* \tilde{C}_n(f)\mathbf{v}}{\mathbf{v}^* \tilde{C}_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} \cdot \frac{\mathbf{v}^* \tilde{C}_n(\mathcal{K}_{m,2r} * f)\mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} \\ &\quad + \frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* T_n\left[s_{2p}(x)\right]\mathbf{v}} \cdot \frac{\mathbf{v}^* R_n \mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}}. \end{aligned}$$

Notice that for all sufficiently large n, except for the last factor, all factors above are uniformly bounded below and above by positive constants. We thus have

$$\frac{\mathbf{v}^* T_n(f)\mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} = \alpha(\mathbf{v}) + \beta(\mathbf{v}) \cdot \frac{\mathbf{v}^* R_n \mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}}$$

when n is large. Here

$$0 < \alpha_{\min} \le \alpha(\mathbf{v}) \le \alpha_{\max} < \infty, \qquad 0 < \beta_{\min} \le \beta(\mathbf{v}) \le \beta_{\max} < \infty.$$

Hence for large n and for all nonzero $\mathbf{v} \in \mathbb{C}^n$,

$$\frac{\mathbf{v}^* [T_n(f) - \beta_{\max} R_n]\mathbf{v}}{\mathbf{v}^* C_n(\mathcal{K}_{m,2r} * f)\mathbf{v}} \le \alpha_{\max}.$$

Let the number of positive eigenvalues of R_n be q. Then by Weyl's theorem, at most q eigenvalues of $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ are larger than $\alpha_{\max}$. Similarly, we can prove that at most $2p+1-q$ eigenvalues of $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ are less than $\alpha_{\min}$. Hence the theorem follows. □

Finally, we prove that all the eigenvalues of the preconditioned matrices are bounded from below by a positive constant independent of n.

Theorem 4.10. *Let $f \in \mathbf{C}_{2\pi}^{+}$ with a zero of order $2p$ at z. Let $r > p$ and $m = \lceil n/r \rceil$. Then there exists a constant $c > 0$ independent of n such that for all n sufficiently large, all eigenvalues of the preconditioned matrix $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ are larger than c.*

Proof. In view of the proof of Theorem 4.9, it suffices to obtain a lower bound of the second Rayleigh quotient in the right-hand side of (4.24). Equivalently, we have to get an upper bound of the spectral radius $\rho[T_n^{-1}[s_{2p}(x)]\tilde{C}_n[s_{2p}(x)]]$. We note that by (4.23),

$$\tilde{C}_n\left[s_{2p}(x)\right] = C_n\left[s_{2p}(x)\right] + E_n,$$

where E_n is either the zero matrix or is given by

$$F_n^* \mathrm{diag}\left(\ldots, 0, \frac{1}{n^{2p}} - s_{2p}\Big(\frac{2\pi j}{n}\Big), 0, \ldots\right) F_n$$

for some j such that $|2\pi j/n - z| < \pi/n$. Thus $\|E_n\|_2 = O(n^{-2p})$. By Theorem 1.13, $T_n^{-1}[s_{2p}(x)]$ is positive definite. Thus the matrix $T_n^{-1}[s_{2p}(x)]\tilde{C}_n[s_{2p}(x)]$ is similar to the symmetric matrix $T_n^{-1/2}\left[s_{2p}(x)\right]\tilde{C}_n\left[s_{2p}(x)\right]T_n^{-1/2}\left[s_{2p}(x)\right]$. Hence we have

$$\begin{aligned}
&\rho\left[T_n^{-1}\left[s_{2p}(x)\right]\tilde{C}_n\left[s_{2p}(x)\right]\right]\\
&= \rho\left[T_n^{-1/2}\left[s_{2p}(x)\right]\tilde{C}_n\left[s_{2p}(x)\right]T_n^{-1/2}\left[s_{2p}(x)\right]\right]\\
&\le \rho\left[T_n^{-1/2}\left[s_{2p}(x)\right]C_n\left[s_{2p}(x)\right]T_n^{-1/2}\left[s_{2p}(x)\right]\right]\\
&\quad + \rho\left[T_n^{-1/2}\left[s_{2p}(x)\right]E_n T_n^{-1/2}\left[s_{2p}(x)\right]\right]\\
&\le \rho\left[T_n^{-1}\left[s_{2p}(x)\right]C_n\left[s_{2p}(x)\right]\right] + \left\|T_n^{-1}\left[s_{2p}(x)\right]\right\|_2 \|E_n\|_2. \qquad (4.26)
\end{aligned}$$

By Theorem 1.13 again, we have

$$\left\|T_n^{-1}\left[s_{2p}(x)\right]\right\|_2 = O(n^{2p}).$$

Hence the last term in (4.26) is of $O(1)$.

It remains to estimate the first term in (4.26). According to (4.25), we partition

$$T_n^{-1}\left[s_{2p}(x)\right] = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12}^* & B_{22} & B_{23} \\ B_{13}^* & B_{23}^* & B_{33} \end{pmatrix},$$

where B_{11} and B_{33} are p-by-p matrices. We then have by (4.25),

$$\begin{aligned} \rho\left[T_n^{-1}\left[s_{2p}(x)\right]C_n\left[s_{2p}(x)\right]\right] &\le 1+\rho\left[\begin{pmatrix} B_{13}R_p^* & \mathbf{0} & B_{11}R_p \\ B_{23}R_p^* & \mathbf{0} & B_{12}^*R_p \\ B_{33}R_p^* & \mathbf{0} & B_{13}^*R_p \end{pmatrix}\right] \\ &= 1+\rho\left[\begin{pmatrix} B_{13}R_p^* & B_{11}R_p \\ B_{33}R_p^* & B_{13}^*R_p \end{pmatrix}\right], \end{aligned} \tag{4.27}$$

where the last equality follows because the 3-by-3 block matrix in the equation has vanishing central column blocks. In [4, Theorem 4.3], it has been shown that R_p, B_{11}, B_{13}, and B_{33} all have bounded $\|\cdot\|_1$ norms and $\|\cdot\|_\infty$ norms. Hence using the fact that $\rho[\cdot] \le \|\cdot\|_2 \le (\|\cdot\|_1 \cdot \|\cdot\|_\infty)^{1/2}$, we see that (4.27) is bounded and the theorem follows. □

By combining Theorems 4.9 and 4.10, the number of PCG iterations required for convergence is of $O(1)$. Since each PCG iteration requires $O(n\log n)$ operations and so does the construction of the preconditioner (see Section 4.1), the total complexity of the PCG method for solving Toeplitz systems generated by $f \in \mathbf{C}_{2\pi}^+$ is of $O(n\log n)$ operations.

4.3.2 Positive functions

In this subsection, we consider the case where the generating function is strictly positive. We note that by the Grenander–Szegö theorem, the spectrum of $T_n(f)$ is contained in $[f_{\min}, f_{\max}]$, where $f_{\min}$ and $f_{\max}$ are the minimum and maximum values of f. It is easy to see that

$$0 < f_{\min} \le \left(\mathcal{K}_{m,2r} * f\right)(y) \le f_{\max}.$$

Thus the whole spectrum of $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ is contained in

$$[f_{\min}/f_{\max},\ f_{\max}/f_{\min}];$$

i.e., the preconditioned system is also well-conditioned. We now show that its spectrum is clustered around 1.

Theorem 4.11. *Let $f \in \mathbf{C}_{2\pi}$ be positive. Then the spectrum of the preconditioned matrix $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ is clustered around 1 for sufficiently large n, where $m = \lceil n/r \rceil$.*

Proof. We first prove that $\mathcal{K}_{m,2r} * f$ converges to f uniformly on $[-\pi, \pi]$. For $\mu > 0$, let

$$\omega(f, \mu) \equiv \max_{x,|t| \leq \mu} |f(x) - f(x-t)|$$

be the modulus of continuity of f. It has the property that

$$\omega(f, \lambda\mu) \leq (\lambda + 1)\omega(f, \mu)$$

for $\lambda \geq 0$; see [64, p. 43]. By the uniform continuity of f, for each $\varepsilon > 0$, there exists an $\eta > 0$ such that $\omega(f, \eta) < \varepsilon$. Taking $n > 1/\eta$, we then have for all $y \in [-\pi, \pi]$,

$$\begin{aligned}
|f(y) - (\mathcal{K}_{m,2r} * f)(y)| &= \left| \frac{1}{2\pi} \int_{-\pi}^{\pi} [\mathcal{K}_{m,2r}(t)f(y) - \mathcal{K}_{m,2r}(t)f(y-t)]\, dt \right| \\
&\leq \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(y) - f(y-t)| \mathcal{K}_{m,2r}(t) dt \\
&\leq \frac{1}{2\pi} \int_{-\pi}^{\pi} \omega(f, |t|) \mathcal{K}_{m,2r}(t) dt \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} \omega\Big(f, n|t| \cdot \frac{1}{n}\Big) \mathcal{K}_{m,2r}(t) dt \\
&\leq \frac{1}{2\pi} \int_{-\pi}^{\pi} (n|t| + 1) \omega\Big(f, \frac{1}{n}\Big) \mathcal{K}_{m,2r}(t) dt \\
&= \omega\Big(f, \frac{1}{n}\Big)(c+1) \ \leq \ (c+1)\varepsilon,
\end{aligned}$$

where $c = \frac{n}{\pi} \int_0^{\pi} \mathcal{K}_{m,2r}(t) t dt$ is bounded by a constant independent of n; see (4.8) for $p = 1/2$. Therefore, $\mathcal{K}_{m,2r} * f$ converges uniformly to f. By Theorem 3.5, the spectrum of $C_n^{-1}(\mathcal{K}_{m,2r} * f)T_n(f)$ is clustered around 1 for sufficiently large n. □

We conclude immediately that when $f \in \mathbf{C}_{2\pi}$ is positive and the preconditioner $C_n(\mathcal{K}_{m,2r} * f)$ is used, the PCG method converges superlinearly; see Section 1.3.1.

4.4 Examples

In this section, we illustrate by eight numerical examples the effectiveness of the preconditioner $C_n(\mathcal{K}_{m,2r} * f)$ in solving Toeplitz systems $T_n(f)\mathbf{u} = \mathbf{b}$ and compare them with Strang's and T. Chan's circulant preconditioners. The last six examples are ill-conditioned Toeplitz systems where the condition numbers of the systems grow like $O(n^q)$ for some $q > 0$. They correspond to f having zeros of order q; see [71]. Because of the ill-conditioning, the CG method will converge slowly and the number of iterations required for convergence grows like $O(n^{q/2})$. However, we will see that using the preconditioner $C_n(\mathcal{K}_{m,2r} * f)$ with $2r > q$, the preconditioned system will converge linearly.

In the following, we set $m = \lceil n/r \rceil$. The right-hand side $\mathbf{b}$ is the vector of all ones. Since the functions are nonnegative, the $T_n(f)$ so generated are all positive definite; see Theorem 1.13. As mentioned in Section 4.1, the construction of the preconditioners for an n-by-n Toeplitz matrix requires only the n diagonal entries $\{t_j\}_{|j|<n}$ of the given Toeplitz matrix. No explicit knowledge of f is required. See A.3 for `pchoice` equalling 9 to 11. Note that `coef` in A.3 are computed in A.4 using (4.2).

Tables 4.1–4.4 show the number of iterations required for convergence for different preconditioners. They can be reproduced by running A.1 with three

Table 4.1. *Number of iterations for well-conditioned systems.*

	$x^4 + 1$						$\lvert x\rvert^3 + 0.01$					
n	32	64	128	256	512	1024	32	64	128	256	512	1024
I	19	36	55	66	70	71	20	52	130	272	395	431
$s(T_n)$	8	6	5	5	5	5	10	11	10	8	6	6
$c_F(T_n)$	7	7	6	6	6	5	13	15	18	15	12	10
$K_{m,4}$	6	5	5	5	5	5	9	8	6	6	6	6
$K_{m,6}$	6	5	5	5	5	5	9	8	7	7	6	7
$K_{m,8}$	6	6	5	5	5	5	10	9	7	6	7	6

Table 4.2. *Number of iterations for functions with order* 2 *zeros.*

	x^2						$x^2(\pi^4 - x^4)$					
n	32	64	128	256	512	1024	32	64	128	256	512	1024
I	17	38	82	177	371	765	16	32	64	128	256	512
$s(T_n)$	–	–	–	–	–	–	8	9	10	10	10	11
$c_F(T_n)$	10	12	14	17	22	28	9	12	14	16	21	25
$K_{m,4}$	7	8	8	8	9	9	7	7	9	9	9	11
$K_{m,6}$	7	8	9	9	9	9	8	9	9	9	10	10
$K_{m,8}$	8	9	9	9	9	9	8	9	9	10	10	10

Table 4.3. *Number of iterations for functions with order* 4 *zeros.*

	x^4						$x^4(\pi^2 - x^2)$					
n	32	64	128	256	512	1024	32	64	128	256	512	1024
I	30	106	414	1742	†	†	18	62	208	769	2962	†
$s(T_n)$	–	–	–	–	–	–	–	–	–	–	–	–
$c_F(T_n)$	16	25	39	82	211	547	14	21	32	53	139	336
$K_{m,4}$	11	13	16	18	20	24	12	13	16	19	21	25
$K_{m,6}$	13	14	17	18	19	22	13	14	16	19	21	23
$K_{m,8}$	13	15	17	19	22	22	14	14	16	18	21	25

Table 4.4. *Number of iterations for other functions.*

	$\lvert x\rvert^3$						$\sum_{\lvert k\rvert<1024} 1/(\lvert k\rvert+1)e^{\mathbf{i}kx} - 0.3862$					
n	32	64	128	256	512	1024	32	64	128	256	512	1024
I	22	59	168	499	1444	†	21	58	153	399	817	947
$s(T_n)$	–	–	–	–	–	–	–	–	–	–	–	–
$c_F(T_n)$	13	17	24	36	55	84	10	13	15	17	18	13
$K_{m,4}$	10	10	11	12	13	14	6	6	6	5	7	7
$K_{m,6}$	10	10	12	12	13	15	6	6	7	7	7	6
$K_{m,8}$	10	11	12	12	14	16	7	6	7	7	7	6

parameters: `n` equals the size of the system; `pchoice` equals $9, 10$, or 11 for the generalized Jackson kernels with $r = 2, 3$, or 4 respectively; and `fchoice` equals $4, 5, \ldots, 11$ for different generating functions. In the tables, as before, I denotes no preconditioner, $s(T_n)$ is Strang's preconditioner, $K_{m,2r}$ are the preconditioners from the generalized Jackson kernel $\mathcal{K}_{m,2r}$, and $c_F(T_n) = K_{m,2}$ is T. Chan's preconditioner. Iteration numbers more than 3,000 are denoted by "†". We note that $s(T_n)$ in general is not positive definite, as the Dirichlet kernel $\mathcal{D}_n$ is not positive; see Section 3.2. When some of its eigenvalues are negative, we denote the iteration number by "–", as the PCG method does not apply to nondefinite systems.

The first two test functions in Table 4.1 are positive functions and therefore correspond to well-conditioned systems. Notice that the iteration numbers for these original systems tend to a constant when n is large, indicating a linear convergence rate. In this case, we see that all preconditioners work well and the convergence is fast.

The two test functions in Table 4.2 are nonnegative functions with one or more zeros of order 2 on $[-\pi, \pi]$. Thus the condition numbers of the Toeplitz matrices are growing like $O(n^2)$, and hence the number of iterations required for convergence without using any preconditioners is increasing like $O(n)$. We see that for these functions, the number of iterations for convergence using T. Chan's preconditioner increases with n. This is to be expected from the fact that the order of $\mathcal{K}_{m,2} * x^2$ does not match that of x^2 at $x = 0$; see (4.11). However, we see that $K_{m,4}$, $K_{m,6}$, and $K_{m,8}$ all work very well, as predicted from the convergence analysis in Section 4.3.

When the order of the zero is 4, like the two test functions in Table 4.3, the condition numbers of the Toeplitz matrices will increase like $O(n^4)$ and the matrices will be very ill-conditioned even for moderate n. We see from the table that both Strang's and T. Chan's preconditioners fail. As predicted by the theory, $K_{m,4}$, $K_{m,6}$, and $K_{m,8}$ still work very well. The number of iterations required for convergence stays almost the same independent of n.

In Table 4.4, we test two functions that the theory does not cover. The first function is not differentiable at its zero. The second one is a function with slowly decaying Fourier coefficients. We found numerically that the minimum value of $\sum_{|k|<1024} \frac{1}{|k|+1} e^{\mathbf{i}kx}$ is approximately equal to 0.3862. Hence the second function is approximately zero at some points in $[-\pi, \pi]$. Table 4.4 shows that the $K_{m,2r}$ preconditioners still perform better than Strang's and T. Chan's preconditioners.

To further illustrate Theorems 4.9 and 4.10, we give in Figures 4.2 and 4.3 the spectra of the preconditioned matrices for all five preconditioners for $f(x) = x^2$ and x^4 when $n = 128$. We see that the spectra of the preconditioned matrices for $K_{m,6}$ and $K_{m,8}$ are in a small interval around 1, except for one to two large outliers, and that all the eigenvalues are well separated away from 0. We note that Strang's preconditioned matrices in both cases have negative eigenvalues and they are not depicted in the figures. Finally, we would like to mention that the general case where f has more than one zero is done in [26].

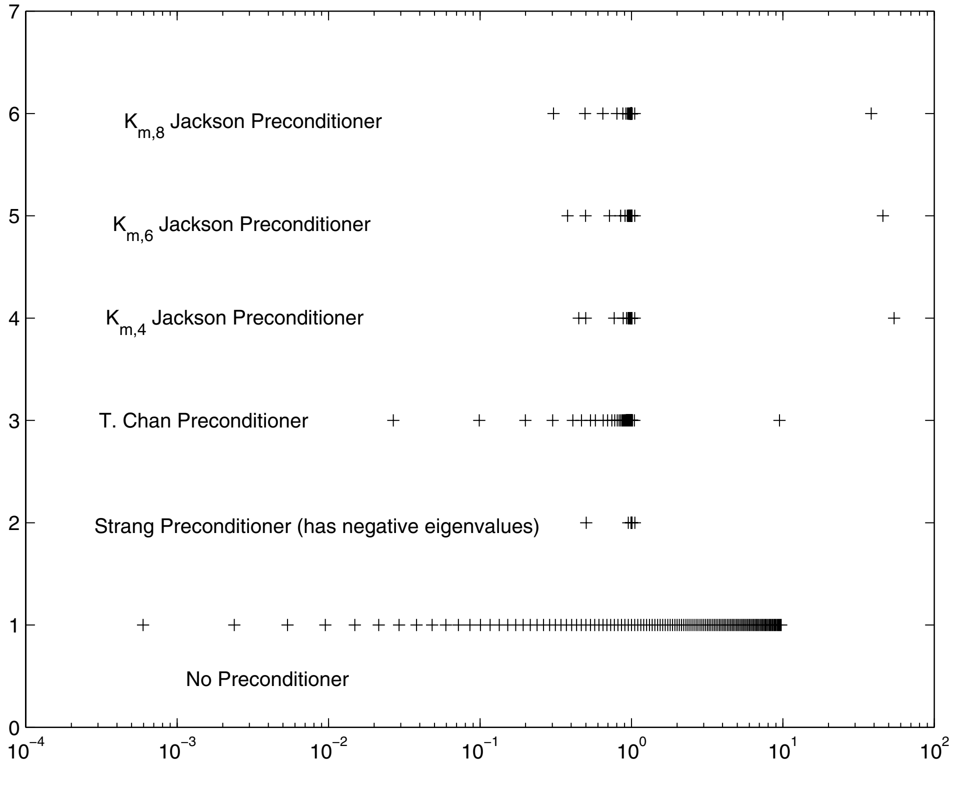

Figure 4.2. *Spectra of preconditioned matrices for $f(x) = x^2$ when $n = 128$.*

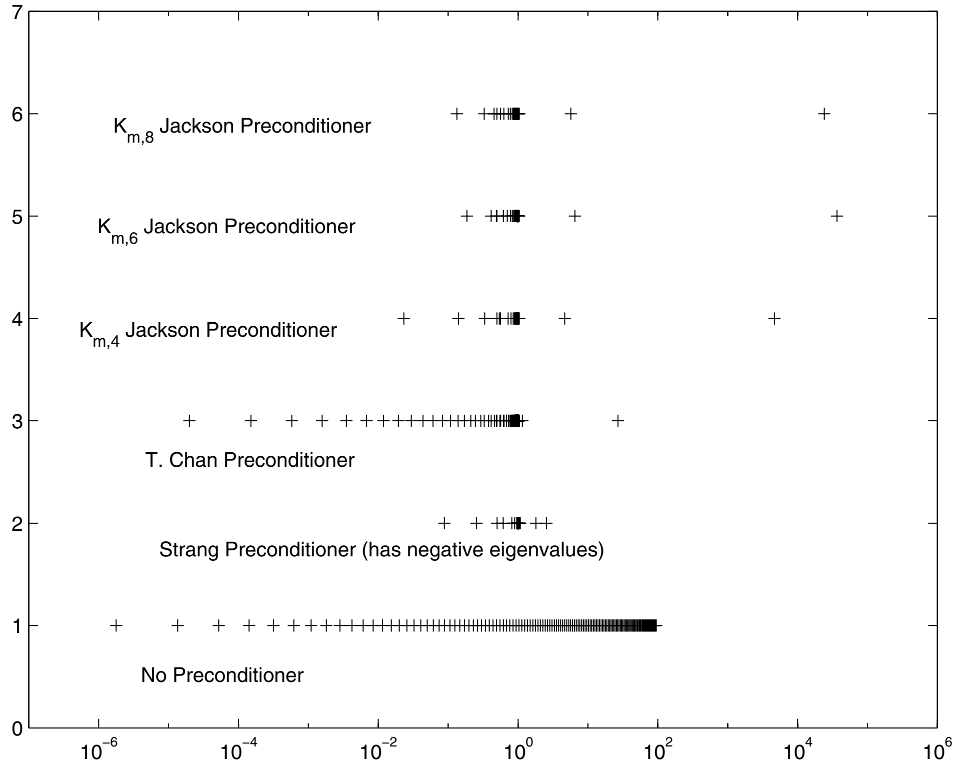

Figure 4.3. *Spectra of preconditioned matrices for $f(x) = x^4$ when $n = 128$.*

Chapter 5

Block Toeplitz systems

The m-by-m block Toeplitz matrix with n-by-n Toeplitz blocks is defined as follows:

$$T_{mn} = \begin{pmatrix} T_{(0)} & T_{(-1)} & \cdots & T_{(2-m)} & T_{(1-m)} \\ T_{(1)} & T_{(0)} & T_{(-1)} & \cdots & T_{(2-m)} \\ \vdots & T_{(1)} & T_{(0)} & \ddots & \vdots \\ T_{(m-2)} & \cdots & \ddots & \ddots & T_{(-1)} \\ T_{(m-1)} & T_{(m-2)} & \cdots & T_{(1)} & T_{(0)} \end{pmatrix}, \tag{5.1}$$

where the blocks $T_{(l)}$, for $|l| \leq m-1$, are themselves Toeplitz matrices of order n. Matrices of the form (5.1) are called BTTB matrices. We are interested in solving the BTTB system

$$T_{mn}\mathbf{u} = \mathbf{b}$$

by the PCG method. BTTB systems arise in a variety of applications in numerical differential equations [7, 19, 49, 55, 63], networks [36], and image processing [20, 24, 66]. In this chapter, several preconditioners that preserve the block structure of T_{mn} are constructed. We show that they are good preconditioners for solving some BTTB systems. Since the block preconditioners are defined not only for BTTB matrices but also for general matrices, we begin with the general case.

5.1 Operators for block matrices

In the following, we call c_U defined by (2.3) the point operator in order to distinguish it from the block operators that we now introduce. Let us begin by considering a general block matrix A_{mn} partitioned as follows:

$$A_{mn} = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,m} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,m} \\ \vdots & \ddots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \cdots & A_{m,m} \end{pmatrix}, \tag{5.2}$$

where $A_{i,j} \in \mathbb{C}^{n \times n}$.

5.1.1 Block operator $c_U^{(1)}$

In view of the point case, a natural choice of preconditioner for A_{mn} given by (5.2) is

$$\begin{pmatrix} c_U(A_{1,1}) & c_U(A_{1,2}) & \cdots & c_U(A_{1,m}) \\ c_U(A_{2,1}) & c_U(A_{2,2}) & \cdots & c_U(A_{2,m}) \\ \vdots & \ddots & \ddots & \vdots \\ c_U(A_{m,1}) & c_U(A_{m,2}) & \cdots & c_U(A_{m,m}) \end{pmatrix},$$

where the blocks $c_U(A_{i,j})$ are just the point approximations to $A_{i,j}$; see (2.3). In the following, we first study its spectral properties.

Let $\delta^{(1)}(A_{mn})$ be defined by

$$\delta^{(1)}(A_{mn}) \equiv \begin{pmatrix} \delta(A_{1,1}) & \delta(A_{1,2}) & \cdots & \delta(A_{1,m}) \\ \delta(A_{2,1}) & \delta(A_{2,2}) & \cdots & \delta(A_{2,m}) \\ \vdots & \ddots & \ddots & \vdots \\ \delta(A_{m,1}) & \delta(A_{m,2}) & \cdots & \delta(A_{m,m}) \end{pmatrix}, \tag{5.3}$$

where each block $\delta(A_{i,j})$, defined as in Section 2.2, is a diagonal matrix whose diagonal is equal to the diagonal of the matrix $A_{i,j}$. The following lemma gives the relation between the spectrum of A_{mn} and the spectrum of $\delta^{(1)}(A_{mn})$.

Lemma 5.1. *For any arbitrary $A_{mn} \in \mathbb{C}^{mn\times mn}$ partitioned as in* (5.2)*, we have*

$$\sigma_{\max}\big(\delta^{(1)}(A_{mn})\big) \leq \sigma_{\max}(A_{mn}), \tag{5.4}$$

where $\sigma_{\max}(\cdot)$ denotes the largest singular value. Furthermore, when A_{mn} is Hermitian, we have

$$\lambda_{\min}(A_{mn}) \leq \lambda_{\min}\big(\delta^{(1)}(A_{mn})\big) \leq \lambda_{\max}\big(\delta^{(1)}(A_{mn})\big) \leq \lambda_{\max}(A_{mn}), \tag{5.5}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalues, respectively.

Proof. Let

$$(A_{mn})_{i,j;k,l} = (A_{k,l})_{ij}$$

be the (i,j)th entry of the (k,l)th block of A_{mn}. Let P be the permutation matrix that satisfies

$$(P^*A_{mn}P)_{k,l;i,j} = (A_{mn})_{i,j;k,l}, \tag{5.6}$$

for $1 \leq i,j \leq n, 1 \leq k,l \leq m$, and let

$$B_{mn} \equiv P^*\delta^{(1)}(A_{mn})P.$$

Then B_{mn} is of the following form:

$$B_{mn} = \begin{pmatrix} B_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & B_{2,2} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & B_{n,n} \end{pmatrix}.$$

We know that B_{mn} and $\delta^{(1)}(A_{mn})$ have the same singular values and eigenvalues. For each k, since $B_{k,k} \in \mathbb{C}^{m\times m}$ is a principal submatrix of the matrix A_{mn}, it follows by Corollary 3.1.3 in [52, p. 149] that

$$\sigma_{\max}(B_{k,k}) \le \sigma_{\max}(A_{mn}).$$

Hence we have

$$\sigma_{\max}\bigl(\delta^{(1)}(A_{mn})\bigr) = \sigma_{\max}(B_{mn}) = \max_k \bigl(\sigma_{\max}(B_{k,k})\bigr) \le \sigma_{\max}(A_{mn}).$$

When A_{mn} is Hermitian, by Cauchy's interlace theorem, we then have

$$\begin{aligned} \lambda_{\min}(A_{mn}) &\le \min_k \bigl(\lambda_{\min}(B_{k,k})\bigr) = \lambda_{\min}\bigl(\delta^{(1)}(A_{mn})\bigr) \\ &\le \lambda_{\max}\bigl(\delta^{(1)}(A_{mn})\bigr) = \max_k \bigl(\lambda_{\max}(B_{k,k})\bigr) \le \lambda_{\max}(A_{mn}). \qquad \Box \end{aligned}$$

In the following, let $\mathscr{D}^{(1)}_{m,n}$ denote the set of all matrices of the form given by (5.3), i.e., $\mathscr{D}^{(1)}_{m,n}$ is the set of all m-by-m block matrices with n-by-n diagonal blocks, and let

$$\mathscr{M}^{(1)}_U \equiv \left\{ (I\otimes U)^* \Lambda^{(1)}_{mn} (I\otimes U) \mid \Lambda^{(1)}_{mn} \in \mathscr{D}^{(1)}_{m,n} \right\},$$

where I is the m-by-m identity matrix, U is any given n-by-n unitary matrix, and the symbol "$\otimes$" denotes the tensor product (Kronecker product). We recall that the tensor product of $A = (a_{ij}) \in \mathbb{C}^{p\times q}$ and $B \in \mathbb{C}^{r\times s}$ is defined as follows:

$$A\otimes B \equiv \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1q}B \\ a_{21}B & a_{22}B & \cdots & a_{2q}B \\ \vdots & \vdots & & \vdots \\ a_{p1}B & a_{p2}B & \cdots & a_{pq}B \end{pmatrix},$$

which is a pr-by-qs matrix. The basic properties of the tensor product can be found in [41, 55].

Similarly to the definition of the operator c_U, we now define an operator $c^{(1)}_U$ that maps every $A_{mn} \in \mathbb{C}^{mn\times mn}$ to the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over all $W_{mn} \in \mathscr{M}^{(1)}_U$. Some properties of $c^{(1)}_U$ are given in the following theorem.

Theorem 5.2. *For any arbitrary $A_{mn} \in \mathbb{C}^{mn\times mn}$ partitioned as in* (5.2)*, let $c^{(1)}_U(A_{mn})$ be the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over all $W_{mn} \in \mathscr{M}^{(1)}_U$. Then the following hold:*

(i) *$c^{(1)}_U(A_{mn})$ is uniquely determined by A_{mn} and is given by*

$$c^{(1)}_U(A_{mn}) = (I\otimes U)^* \delta^{(1)}\bigl[(I\otimes U) A_{mn} (I\otimes U)^*\bigr](I\otimes U). \qquad (5.7)$$

(ii) *$c_U^{(1)}(A_{mn})$ is also given by*

$$c_U^{(1)}(A_{mn}) = \begin{pmatrix} c_U(A_{1,1}) & c_U(A_{1,2}) & \cdots & c_U(A_{1,m}) \\ c_U(A_{2,1}) & c_U(A_{2,2}) & \cdots & c_U(A_{2,m}) \\ \vdots & \ddots & \ddots & \vdots \\ c_U(A_{m,1}) & c_U(A_{m,2}) & \cdots & c_U(A_{m,m}) \end{pmatrix}, \tag{5.8}$$

where c_U is the point operator defined by (2.3).

(iii) *We have*

$$\sigma_{\max}\big(c_U^{(1)}(A_{mn})\big) \le \sigma_{\max}(A_{mn}). \tag{5.9}$$

(iv) *If A_{mn} is Hermitian, then $c_U^{(1)}(A_{mn})$ is also Hermitian and*

$$\lambda_{\min}(A_{mn}) \le \lambda_{\min}\big(c_U^{(1)}(A_{mn})\big) \le \lambda_{\max}\big(c_U^{(1)}(A_{mn})\big) \le \lambda_{\max}(A_{mn}).$$

In particular, if A_{mn} is positive definite, then so is $c_U^{(1)}(A_{mn})$.

(v) *$c_U^{(1)}$ is a linear projection operator with the operator norms*

$$\|c_U^{(1)}\|_2 \equiv \sup_{\|A_{mn}\|_2=1} \|c_U^{(1)}(A_{mn})\|_2 = 1$$

and

$$\|c_U^{(1)}\|_{\mathscr{F}} \equiv \sup_{\|A_{mn}\|_{\mathscr{F}}=1} \|c_U^{(1)}(A_{mn})\|_{\mathscr{F}} = 1.$$

Proof.

(i) Let $W_{mn} \in \mathscr{M}_U^{(1)}$ be given by

$$W_{mn} = (I \otimes U)^* \Lambda_{mn}^{(1)} (I \otimes U),$$

where $\Lambda_{mn}^{(1)} \in \mathscr{D}_{m,n}^{(1)}$. Since the Frobenius norm is unitary invariant,

$$\begin{aligned} \|W_{mn} - A_{mn}\|_{\mathscr{F}} &= \|(I \otimes U)^* \Lambda_{mn}^{(1)} (I \otimes U) - A_{mn}\|_{\mathscr{F}} \\ &= \|\Lambda_{mn}^{(1)} - (I \otimes U) A_{mn} (I \otimes U)^*\|_{\mathscr{F}}. \end{aligned}$$

Thus, the minimizing problem $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over $\mathscr{M}_U^{(1)}$ is equivalent to the minimizing problem

$$\|\Lambda_{mn}^{(1)} - (I \otimes U) A_{mn} (I \otimes U)^*\|_{\mathscr{F}}$$

over $\mathscr{D}_{m,n}^{(1)}$. Note that $\Lambda_{mn}^{(1)}$ can affect only the diagonal of each block of

$$(I \otimes U) A_{mn} (I \otimes U)^*.$$

Therefore, the solution for the latter problem is

$$\Lambda_{mn}^{(1)} = \delta^{(1)}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big].$$

Hence

$$c_U^{(1)}(A_{mn}) = (I \otimes U)^* \delta^{(1)}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big](I \otimes U)$$

is the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$. Obviously, $c_U^{(1)}(A_{mn})$ is uniquely determined by A_{mn}.

(ii) Just note that

$$\begin{aligned}
&\delta^{(1)}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big] \\
&\quad = \begin{pmatrix} \delta(UA_{1,1}U^*) & \delta(UA_{1,2}U^*) & \cdots & \delta(UA_{1,m}U^*) \\ \delta(UA_{2,1}U^*) & \delta(UA_{2,2}U^*) & \cdots & \delta(UA_{2,m}U^*) \\ \vdots & \ddots & \ddots & \vdots \\ \delta(UA_{m,1}U^*) & \delta(UA_{m,2}U^*) & \cdots & \delta(UA_{m,m}U^*) \end{pmatrix}.
\end{aligned}$$

(iii) For any $A_{mn} \in \mathbb{C}^{mn \times mn}$, we have by (5.4) and (5.7),

$$\begin{aligned}
\sigma_{\max}\big(c_U^{(1)}(A_{mn})\big) &= \sigma_{\max}\big[\delta^{(1)}\big((I \otimes U)A_{mn}(I \otimes U)^*\big)\big] \\
&\leq \sigma_{\max}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big] = \sigma_{\max}(A_{mn}).
\end{aligned}$$

(iv) If A_{mn} is Hermitian, then it is clear that $c_U^{(1)}(A_{mn})$ is also Hermitian. Moreover, by (5.5) and (5.7), we have

$$\begin{aligned}
\lambda_{\min}(A_{mn}) &= \lambda_{\min}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big] \\
&\leq \lambda_{\min}\big[\delta^{(1)}\big((I \otimes U)A_{mn}(I \otimes U)^*\big)\big] \\
&= \lambda_{\min}\big(c_U^{(1)}(A_{mn})\big) \leq \lambda_{\max}\big(c_U^{(1)}(A_{mn})\big) \\
&= \lambda_{\max}\big[\delta^{(1)}\big((I \otimes U)A_{mn}(I \otimes U)^*\big)\big] \\
&\leq \lambda_{\max}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big] = \lambda_{\max}(A_{mn}).
\end{aligned}$$

(v) By (5.9), we have

$$\|c_U^{(1)}(A_{mn})\|_2 = \sigma_{\max}\big(c_U^{(1)}(A_{mn})\big) \leq \sigma_{\max}(A_{mn}) = \|A_{mn}\|_2.$$

However, for the identity matrix I_{mn}, we have

$$\|c_U^{(1)}(I_{mn})\|_2 = \|I_{mn}\|_2 = 1.$$

Hence $\|c_U^{(1)}\|_2 = 1$. For the Frobenius norm, since

$$\begin{aligned}
\|c_U^{(1)}(A_{mn})\|_{\mathscr{F}} &= \|\delta^{(1)}\big[(I \otimes U)A_{mn}(I \otimes U)^*\big]\|_{\mathscr{F}} \\
&\leq \|(I \otimes U)A_{mn}(I \otimes U)^*\|_{\mathscr{F}} = \|A_{mn}\|_{\mathscr{F}}
\end{aligned}$$

and

$$\Big\|c_U^{(1)}\Big(\frac{1}{\sqrt{mn}}I_{mn}\Big)\Big\|_{\mathscr{F}} = \frac{1}{\sqrt{mn}}\|I_{mn}\|_{\mathscr{F}} = 1,$$

it follows that $\|c_U^{(1)}\|_{\mathscr{F}} = 1$. □

5.1.2 Block operator $\tilde{c}_V^{(1)}$

For matrices A_{mn} partitioned as in (5.2), we can define another block approximation for them. Let $\tilde{\delta}^{(1)}(A_{mn})$ be defined by

$$\tilde{\delta}^{(1)}(A_{mn}) \equiv \begin{pmatrix} A_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & A_{2,2} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & A_{m,m} \end{pmatrix}. \tag{5.10}$$

In the following, we use $\tilde{\mathscr{D}}_{m,n}^{(1)}$ to denote the set of all matrices of the form given by (5.10); i.e., $\tilde{\mathscr{D}}_{m,n}^{(1)}$ is the set of all m-by-m block diagonal matrices with n-by-n blocks. Let

$$\tilde{\mathscr{M}}_V^{(1)} \equiv \left\{ (V \otimes I)^* \tilde{\Lambda}_{mn}^{(1)} (V \otimes I) \mid \tilde{\Lambda}_{mn}^{(1)} \in \tilde{\mathscr{D}}_{m,n}^{(1)} \right\},$$

where V is any given m-by-m unitary matrix and I is the n-by-n identity matrix.

We define an operator $\tilde{c}_V^{(1)}$ that maps every $A_{mn} \in \mathbb{C}^{mn \times mn}$ to the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over all $W_{mn} \in \tilde{\mathscr{M}}_V^{(1)}$. Similar to Theorem 5.2, we have the following theorem.

Theorem 5.3. *For any arbitrary $A_{mn} \in \mathbb{C}^{mn \times mn}$ partitioned as in* (5.2)*, let $\tilde{c}_V^{(1)}(A_{mn})$ be the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over all $W_{mn} \in \tilde{\mathscr{M}}_V^{(1)}$. Then the following hold:*

(i) *$\tilde{c}_V^{(1)}(A_{mn})$ is uniquely determined by A_{mn} and is given by*

$$\tilde{c}_V^{(1)}(A_{mn}) = (V \otimes I)^* \tilde{\delta}^{(1)} \big[(V \otimes I) A_{mn} (V \otimes I)^* \big] (V \otimes I). \tag{5.11}$$

(ii) *We have*

$$\sigma_{\max}\big(\tilde{c}_V^{(1)}(A_{mn})\big) \le \sigma_{\max}(A_{mn}).$$

(iii) *If A_{mn} is Hermitian, then $\tilde{c}_V^{(1)}(A_{mn})$ is also Hermitian and*

$$\lambda_{\min}(A_{mn}) \le \lambda_{\min}\big(\tilde{c}_V^{(1)}(A_{mn})\big) \le \lambda_{\max}\big(\tilde{c}_V^{(1)}(A_{mn})\big) \le \lambda_{\max}(A_{mn}).$$

(iv) *$\tilde{c}_V^{(1)}$ is a linear projection operator with the operator norms*

$$\|\tilde{c}_V^{(1)}\|_2 = \|\tilde{c}_V^{(1)}\|_{\mathscr{F}} = 1.$$

We omit the proof of Theorem 5.3 since it is quite similar to that of Theorem 5.2. The following theorem gives the relationship between $c_U^{(1)}$ and $\tilde{c}_V^{(1)}$.

Theorem 5.4. *Let U be any given unitary matrix and P be the permutation matrix defined as in* (5.6). *For any arbitrary matrix $A_{mn} \in \mathbb{C}^{mn\times mn}$ partitioned as in* (5.2), *we have*

$$\delta^{(1)}(A_{mn}) = P\tilde{\delta}^{(1)}(P^*A_{mn}P)P^*$$

and

$$c_U^{(1)}(A_{mn}) = P\tilde{c}_U^{(1)}(P^*A_{mn}P)P^*.$$

Proof. To prove the first equality, by the definition of $\tilde{\delta}^{(1)}$ and (5.6), we notice that

$$\begin{aligned}[\tilde{\delta}^{(1)}(P^*A_{mn}P)]_{k,l;i,j} &= \begin{cases}(P^*A_{mn}P)_{k,l;i,j}, & i=j,\\ 0, & i\neq j,\end{cases}\\ &= \begin{cases}(A_{mn})_{i,j;k,l}, & i=j,\\ 0, & i\neq j.\end{cases}\end{aligned}$$

Hence

$$[P\tilde{\delta}^{(1)}(P^*A_{mn}P)P^*]_{i,j;k,l} = [\tilde{\delta}^{(1)}(P^*A_{mn}P)]_{k,l;i,j} = \begin{cases}(A_{mn})_{i,j;k,l}, & i=j,\\ 0, & i\neq j,\end{cases}$$

which by definition is equal to $[\delta^{(1)}(A_{mn})]_{i,j;k,l}$. To prove the second equality, since $(I\otimes U)P = P(U\otimes I)$ for any matrix U, we have by (5.11) and (5.7),

$$\begin{aligned}P\tilde{c}_U^{(1)}(P^*A_{mn}P)P^* &= P(U\otimes I)^*\tilde{\delta}^{(1)}[(U\otimes I)P^*A_{mn}P(U\otimes I)^*](U\otimes I)P^*\\ &= (I\otimes U)^*P\tilde{\delta}^{(1)}[P^*(I\otimes U)A_{mn}(I\otimes U)^*P]P^*(I\otimes U)\\ &= (I\otimes U)^*\delta^{(1)}[(I\otimes U)A_{mn}(I\otimes U)^*](I\otimes U)\\ &= c_U^{(1)}(A_{mn}). \qquad \square\end{aligned}$$

5.1.3 Operator $c_{V,U}^{(2)}$

It is natural to consider the operator

$$c_{V,U}^{(2)} \equiv \tilde{c}_V^{(1)} \circ c_U^{(1)},$$

where "$\circ$" denotes the composite of operators. The following lemma is useful in deriving the properties of the operator $c_{V,U}^{(2)}$ in Theorem 5.6.

Lemma 5.5. *For any arbitrary matrix $A_{mn} \in \mathbb{C}^{mn\times mn}$ partitioned as in* (5.2), *we have*

$$(I\otimes U)^*\tilde{\delta}^{(1)}(A_{mn})(I\otimes U) = \tilde{\delta}^{(1)}[(I\otimes U)^*A_{mn}(I\otimes U)] \tag{5.12}$$

and

$$(V\otimes I)\delta^{(1)}(A_{mn})(V\otimes I)^* = \delta^{(1)}[(V\otimes I)A_{mn}(V\otimes I)^*]. \tag{5.13}$$

Furthermore,

$$\tilde{\delta}^{(1)}\circ\delta^{(1)}(A_{mn}) = \delta(A_{mn}) = \delta^{(1)}\circ\tilde{\delta}^{(1)}(A_{mn}). \tag{5.14}$$

The proof of Lemma 5.5 is straightforward; we therefore omit it. Let

$$\mathscr{M}_{V\otimes U} \equiv \{(V\otimes U)^*\Lambda_{mn}(V\otimes U) \mid \Lambda_{mn} \text{ is any } mn\text{-by-}mn \text{ diagonal matrix}\},$$

where V is any given m-by-m unitary matrix and U is any given n-by-n unitary matrix. We have the following theorem, which states that the operator $c^{(2)}_{V,U}$ is just a special case of the point operator.

Theorem 5.6. *For any arbitrary matrix $A_{mn} \in \mathbb{C}^{mn\times mn}$ partitioned as in* (5.2), *let $c_{V\otimes U}(A_{mn})$ be the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over all $W_{mn} \in \mathscr{M}_{V\otimes U}$, where $c_{V\otimes U}$ is the point operator defined by* (2.3). *Then the following hold:*

(i) *We have*

$$c^{(2)}_{V,U}(A_{mn}) = c_{V\otimes U}(A_{mn}).$$

(ii) *We have*

$$\sigma_{\max}\big(c^{(2)}_{V,U}(A_{mn})\big) \le \sigma_{\max}(A_{mn}).$$

(iii) *If A_{mn} is Hermitian, then $c^{(2)}_{V,U}(A_{mn})$ is also Hermitian and*

$$\lambda_{\min}(A_{mn}) \le \lambda_{\min}\big(c^{(2)}_{V,U}(A_{mn})\big) \le \lambda_{\max}\big(c^{(2)}_{V,U}(A_{mn})\big) \le \lambda_{\max}(A_{mn}).$$

(iv) *The operator $c^{(2)}_{V,U}$ has the operator norms*

$$\|c^{(2)}_{V,U}\|_2 = \|c^{(2)}_{V,U}\|_{\mathscr{F}} = 1.$$

Proof. Only (i) needs to be proved, and the others can be referred to Theorem 2.7. For any given A_{mn}, by the definitions of $c^{(1)}_U$ and $\tilde{c}^{(1)}_V$, we have

$$\begin{aligned} &c^{(2)}_{V,U}(A_{mn}) = \tilde{c}^{(1)}_V[c^{(1)}_U(A_{mn})] \\ &= (V\otimes I)^*\tilde{\delta}^{(1)}\big\{(V\otimes I)\big[(I\otimes U)^*\delta^{(1)}[(I\otimes U)A_{mn}(I\otimes U)^*](I\otimes U)\big](V\otimes I)^*\big\}(V\otimes I) \\ &= (V\otimes I)^*\tilde{\delta}^{(1)}\big\{(I\otimes U)^*(V\otimes I)\delta^{(1)}[(I\otimes U)A_{mn}(I\otimes U)^*](V\otimes I)^*(I\otimes U)\big\}(V\otimes I). \end{aligned}$$

Hence by (5.12)–(5.14), we have

$$\begin{aligned} c^{(2)}_{V,U}(A_{mn}) &= (V\otimes U)^*\tilde{\delta}^{(1)}\big\{\delta^{(1)}[(V\otimes U)A_{mn}(V\otimes U)^*]\big\}(V\otimes U) \\ &= (V\otimes U)^*\delta[(V\otimes U)A_{mn}(V\otimes U)^*](V\otimes U) = c_{V\otimes U}(A_{mn}). \qquad \square \end{aligned}$$

We remark that intuitively $c^{(1)}_U(A_{mn})$ and $\tilde{c}^{(1)}_V(A_{mn})$ resemble the diagonalization of A_{mn} along one specific direction. Hence $c^{(2)}_{V,U}(A_{mn})$ resembles the diagonalization of A_{mn} along both directions.

5.1.4 Three useful formulae

When U and V both are equal to the Fourier matrix F, we give three simple formulae for constructing $c_F^{(1)}(A_{mn})$, $\tilde{c}_F^{(1)}(A_{mn})$, and $c_{F,F}^{(2)}(A_{mn})$. These preconditioners will be used later to solve block Toeplitz systems. We have by (5.8),

$$c_F^{(1)}(A_{mn}) = \begin{pmatrix} c_F(A_{1,1}) & c_F(A_{1,2}) & \cdots & c_F(A_{1,m}) \\ c_F(A_{2,1}) & c_F(A_{2,2}) & \cdots & c_F(A_{2,m}) \\ \vdots & \ddots & \ddots & \vdots \\ c_F(A_{m,1}) & c_F(A_{m,2}) & \cdots & c_F(A_{m,m}) \end{pmatrix}, \tag{5.15}$$

where each block $c_F(A_{i,j})$ is T. Chan's circulant preconditioner for $A_{i,j}$. We remark that $c_F^{(1)}(A_{mn})$ is a block matrix with circulant blocks and is called a CB matrix [37].

Next we construct $\tilde{c}_F^{(1)}(A_{mn})$ by using Theorem 5.4. Let $A_{mn} = P^* B_{mn} P$ and partition B_{mn} into n^2 blocks with each block $B_{i,j} \in \mathbb{C}^{m\times m}$. Then by Theorem 5.4 and (5.15), we have

$$[\tilde{c}_F^{(1)}(A_{mn})]_{i,j;k,l} = [P^* c_F^{(1)}(B_{mn})P]_{i,j;k,l} = [c_F^{(1)}(B_{mn})]_{k,l;i,j} = (c_F(B_{i,j}))_{kl},$$

where $B_{i,j}$ is the (i,j)th block of the matrix B_{mn}. By (2.4), we see that the (k,l)th entry of the circulant matrix $c_F(B_{i,j})$ is given by

$$(c_F(B_{i,j}))_{kl} = \frac{1}{m} \sum_{p-q\equiv k-l(\bmod m)} (B_{i,j})_{pq}.$$

Since $(B_{i,j})_{pq} = (A_{p,q})_{ij}$, we have

$$[\tilde{c}_F^{(1)}(A_{mn})]_{i,j;k,l} = \frac{1}{m} \sum_{p-q\equiv k-l(\bmod m)} (A_{p,q})_{ij}$$

for $1 \le i,j \le n$ and $1 \le k,l \le m$. Thus, the (k,l)th block of $\tilde{c}_F^{(1)}(A_{mn})$ is given by

$$\frac{1}{m} \sum_{p-q\equiv k-l(\bmod m)} A_{p,q}.$$

Since it depends only on $k-l$ modulo m, we see that $\tilde{c}_F^{(1)}(A_{mn})$ is a block circulant matrix and is called a BC matrix [37]. By using Q defined as in (2.5), we further have

$$\tilde{c}_F^{(1)}(A_{mn}) = \frac{1}{m} \sum_{j=0}^{m-1} \left(Q^j \otimes \sum_{p-q\equiv j(\bmod m)} A_{p,q} \right). \tag{5.16}$$

Finally, by using (2.4), (5.16), and Theorem 5.6, one can easily obtain the following formula:

$$c_{F,F}^{(2)}(A_{mn}) = \frac{1}{mn} \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} \left(\sum_{p-q\equiv j(\bmod m)} \sum_{r-s\equiv k(\bmod n)} (A_{p,q})_{rs} \right) (Q^j \otimes Q^k). \tag{5.17}$$

We remark that $c^{(2)}_{F,F}(A_{mn})$ is a block circulant matrix with circulant blocks [37] and will be called a BCCB matrix. Actually, from Theorem 5.6, we know that $c^{(2)}_{F,F}(A_{mn}) = c_{F\otimes F}(A_{mn})$ is the minimizer of $\|W_{mn} - A_{mn}\|_{\mathscr{F}}$ over all $W_{mn} \in \mathscr{M}_{F\otimes F}$, where $\mathscr{M}_{F\otimes F}$ is the set of all BCCB matrices [37].

5.2 Operation cost for preconditioned system

In this section, we study the cost of solving $T_{mn}\mathbf{u} = \mathbf{b}$ by the PCG method with preconditioners $c^{(1)}_{F}(T_{mn})$ and $c^{(2)}_{F,F}(T_{mn})$, where T_{mn} is a BTTB matrix of the form given in (5.1). The analysis for $\tilde{c}^{(1)}_{F}(T_{mn})$ is similar. We first recall that in each iteration of the PCG method, we have to compute the matrix-vector multiplication $T_{mn}\mathbf{v}$ for some vector $\mathbf{v}$ and the product

$$\mathbf{y} = (c^{(1)}_{F}(T_{mn}))^{-1}\mathbf{d} \tag{5.18}$$

or

$$\mathbf{y} = (c^{(2)}_{F,F}(T_{mn}))^{-1}\mathbf{d} \tag{5.19}$$

for some vector $\mathbf{d}$; see Section 1.3.1 or [41, 56].

For the cost of computing $T_{mn}\mathbf{v}$, we recall that for any Toeplitz matrix T_n, the matrix-vector multiplication $T_n\mathbf{w}$ can be computed by FFTs by first embedding $T_n\mathbf{w}$ into a $2n$-by-$2n$ circulant matrix and extending $\mathbf{w}$ to a $2n$-vector by zeros; see (1.7). For the matrix-vector product $T_{mn}\mathbf{v}$, we can use the same trick. We first embed T_{mn} into a (blockwise) $2m$-by-$2m$ block circulant matrix where each block itself is a $2n$-by-$2n$ circulant matrix. Then we extend $\mathbf{v}$ to a $4mn$-vector by putting zeros in the appropriate places. Using $F_{2m} \otimes F_{2n}$ to diagonalize the $4mn$-by-$4mn$ BCCB matrix, $T_{mn}\mathbf{v}$ can be obtained in $O(mn \log mn)$ operations by using 2-dimensional FFTs. The MATLAB algorithm for doing this is in A.16. It requires the vector `tev` formed by the eigenvalues of the $4mn$-by-$4mn$ BCCB matrix which is generated in A.11.

5.2.1 Case of $c^{(1)}_{F}(T_{mn})$

For $c^{(1)}_{F}(T_{mn})$, by (5.15), its blocks are just $c_F(T_{(k)})$. Hence by (2.6) and (1.5), the diagonal $\delta(FT_{(k)}F^*)$ can be computed in $O(n \log n)$ operations. Therefore, in the initialization step, we need $O(mn \log n)$ operations to form

$$\Delta \equiv \delta^{(1)}\big((I \otimes F)T_{mn}(I \otimes F)^*\big).$$

This is done in A.13, where the input `t` is the first column of T_{mn} generated by A.9.

Once we obtain Δ, we can start the PCG iterations. Note that

$$P^*\Delta P = \begin{pmatrix} \tilde{T}_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \tilde{T}_{2,2} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \tilde{T}_{n,n} \end{pmatrix},$$

where P is defined as in (5.6) and

$$(\tilde{T}_{k,k})_{ij} = \big(\delta(FT_{i,j}F^*)\big)_{kk} = \big(\delta(FT_{(i-j)}F^*)\big)_{kk}$$

for $1 \le i, j \le m$ and $1 \le k \le n$. Hence the diagonal blocks $\tilde{T}_{k,k}$ are m-by-m Toeplitz matrices. Therefore, only $O(m \log^2 m)$ operations are required to compute $\tilde{T}_{k,k}^{-1}\mathbf{v}$ for any vector $\mathbf{v}$; see Ammar and Gragg [2]. Thus, (5.18) can be computed by

$$\begin{aligned}\mathbf{y} = \big(c_F^{(1)}(T_{mn})\big)^{-1}\mathbf{d} &= (I \otimes F^*)\Delta^{-1}(I \otimes F)\mathbf{d} \\ &= [(I \otimes F^*)P](P^*\Delta P)^{-1}[P^*(I \otimes F)]\mathbf{d}\end{aligned}$$

in $O(nm \log^2 m)$ operations; see A.15.

Recall that $T_{mn}\mathbf{v}$ can be obtained in $O(mn \log mn)$ operations. Thus we conclude that the cost per iteration is $O(nm \log^2 m + mn \log mn)$. The PCG algorithm is given in A.14. If $m > n$, one can use $\tilde{c}_F^{(1)}(T_{mn})$ as a preconditioner instead. We would like to mention that some of the block operations can be done in a parallel way. This can further reduce the cost per iteration.

5.2.2 Case of $c_{F,F}^{(2)}(T_{mn})$

We remark that $c_{F,F}^{(2)}(T_{mn})$ is a BCCB matrix and that for any BCCB matrix C_{mn}, it can be defined by its first column. Like (1.5), we have the following relation between the first column and the eigenvalues of C_{mn}:

$$(F_m \otimes F_n)C_{mn}\mathbf{e}_1 = \frac{1}{\sqrt{mn}}\Lambda_{mn}\mathbf{1}_{mn}, \tag{5.20}$$

where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{mn}$ is the first unit vector, $\mathbf{1}_{mn} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{mn}$, and Λ_{mn} is a diagonal matrix holding the eigenvalues of C_{mn}.

For $c_{F,F}^{(2)}(T_{mn})$, by using (5.17), it is not difficult to show that the kth entry in the jth block of the first column of the matrix is given by

$$c_k^{(j)} = \frac{1}{mn}\left[(m-j)(n-k)t_k^{(j)} + j(n-k)t_k^{(j-m)} + (m-j)kt_{k-n}^{(j)} + jkt_{k-n}^{(j-m)}\right]$$

for $0 \le j \le m-1$ and $0 \le k \le n-1$. Thus it requires only $O(mn)$ operations to compute the first column of $c_{F,F}^{(2)}(T_{mn})$. Using (5.20), we can compute the eigenvalues of $c_{F,F}^{(2)}(T_{mn})$ by using 2-dimensional FFTs in $O(mn \log mn)$ operations. This is done by A.17, where `t` is the first column of T_{mn} and `ev` is the vector holding the eigenvalues.

In each iteration of the PCG method, besides $O(mn \log mn)$ operations to compute $T_{mn}\mathbf{v}$, we also need to compute

$$\mathbf{y} = (c_{F,F}^{(2)}(T_{mn}))^{-1}\mathbf{d} = (F_m^* \otimes F_n^*)\Lambda_{mn}^{-1}(F_m \otimes F_n)\mathbf{d}$$

in (5.19). This can be done in $O(mn \log mn)$ operations by using 2-dimensional FFTs (see A.19), where the input `ev` is the vector holding the eigenvalues of

$c^{(2)}_{F,F}(T_{mn})$ computed by A.17. Thus the cost per iteration of the PCG method is $O(mn\log mn)$ operations, and it is given in A.18. Again some of the block operations can be done in a parallel way to reduce the cost.

For the implementation and cost of the PCG method for general block systems $A_{mn}\mathbf{u} = \mathbf{b}$, we refer readers to [21].

5.3 Convergence rate

In this section, we analyze the spectra of the preconditioned systems. Let the entries of T_{mn} be denoted by

$$(T_{mn})_{p,q;r,s} = t^{(r-s)}_{p-q}$$

for $1 \le p, q \le n$ and $1 \le r, s \le m$. Moreover, the matrix $T_{mn}(f)$ is associated with a generating function $f(x, y)$ as follows:

$$t^{(j)}_k(f) \equiv \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(x, y)e^{-\mathbf{i}(jx+ky)}dxdy. \tag{5.21}$$

We have the following important properties for any m and n:

(i) When f is real-valued, then $T_{mn}(f)$ are Hermitian, i.e.,

$$t^{(j)}_k(f) = \bar{t}^{(-j)}_{-k}(f).$$

(ii) When f is real-valued with $f(x, y) = f(-x, -y)$, then $T_{mn}(f)$ are real symmetric, i.e.,

$$t^{(j)}_k(f) = t^{(-j)}_{-k}(f).$$

(iii) When f is real-valued and even, i.e., $f(x, y) = f(|x|, |y|)$, then $T_{mn}(f)$ are level-2 symmetric, i.e.,

$$t^{(j)}_k(f) = t^{(|j|)}_{|k|}(f).$$

Let $\mathbf{C}_{2\pi\times 2\pi}$ denote the space of all 2π-periodic (in each direction) continuous real-valued functions $f(x, y)$. The following theorem gives the relation between the values of $f(x, y)$ and the eigenvalues of $T_{mn}(f)$. Although its proof is similar to that of Theorem 1.12, we give it here for completeness.

Theorem 5.7. *Let T_{mn} be a BTTB matrix with a generating function $f(x, y) \in \mathbf{C}_{2\pi\times 2\pi}$. Let $\lambda_{\min}(T_{mn})$ and $\lambda_{\max}(T_{mn})$ denote the smallest and largest eigenvalues of T_{mn}, respectively. Then we have*

$$f_{\min} \le \lambda_{\min}(T_{mn}) \le \lambda_{\max}(T_{mn}) \le f_{\max},$$

where $f_{\min}$ and $f_{\max}$ denote the minimum and maximum values of $f(x, y)$, respectively. In particular, if $f_{\min} > 0$, then T_{mn} is positive definite.

Proof. Let

$$\mathbf{v} = \left(v_0^{(0)}, v_1^{(0)}, \ldots, v_{n-1}^{(0)}, v_0^{(1)}, \ldots, v_0^{(m-1)}, \ldots, v_{n-1}^{(m-1)}\right)^T \in \mathbb{C}^{mn}.$$

Then we have

$$\mathbf{v}^* T_{mn} \mathbf{v} = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left| \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} v_k^{(j)} e^{-\mathbf{i}(jx+ky)} \right|^2 f(x,y) dx dy. \tag{5.22}$$

Since $f_{\min} \leq f(x,y) \leq f_{\max}$ for all x and y, we have by (5.22),

$$f_{\min} \leq \mathbf{v}^* T_{mn} \mathbf{v} \leq f_{\max},$$

provided that $\mathbf{v}$ satisfies the condition

$$\mathbf{v}^* \mathbf{v} = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left| \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} v_k^{(j)} e^{-\mathbf{i}(jx+ky)} \right|^2 dx dy = 1.$$

Therefore, we have by the Courant–Fischer minimax theorem,

$$f_{\min} \leq \lambda_{\min}(T_{mn}) \leq \lambda_{\max}(T_{mn}) \leq f_{\max}. \qquad \square$$

Now we are going to analyze the convergence rate of the PCG method. We consider $f(x,y)$ in the Wiener class, i.e.,

$$\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} |t_k^{(j)}(f)| < \infty.$$

We remark that the Wiener class is a proper subset of $\mathbf{C}_{2\pi \times 2\pi}$.

5.3.1 Convergence rate of $c_F^{(1)}(T_{mn})$

Let

$$s_F^{(1)}(T_{mn}) \equiv \begin{pmatrix} s(T_{(0)}) & s(T_{(-1)}) & \cdots & s(T_{(1-m)}) \\ s(T_{(1)}) & s(T_{(0)}) & \cdots & s(T_{(2-m)}) \\ \vdots & \ddots & \ddots & \vdots \\ s(T_{(m-1)}) & s(T_{(m-2)}) & \cdots & s(T_{(0)}) \end{pmatrix},$$

where $s(T_{(j)})$ is Strang's circulant preconditioner defined as in (2.1) for $T_{(j)}$, $|j| < m-1$. Consider

$$c_F^{(1)}(T_{mn}) - T_{mn} = c_F^{(1)}(T_{mn}) - s_F^{(1)}(T_{mn}) + s_F^{(1)}(T_{mn}) - T_{mn}. \tag{5.23}$$

By applying the technique used in the proofs of Theorem 2.2 and Lemma 2.8, for any $\epsilon > 0$, one can prove that

$$\lim_{n \to \infty} \rho \left[c_F^{(1)}(T_{mn}) - s_F^{(1)}(T_{mn}) \right] = 0, \tag{5.24}$$

where $\rho[\cdot]$ denotes the spectral radius and

$$s_F^{(1)}(T_{mn}) - T_{mn} = M_{mn} + L_{O(m)}, \tag{5.25}$$

where M_{mn} and $L_{O(m)}$ are Hermitian matrices with

$$\|M_{mn}\|_2 < \epsilon, \qquad \text{rank}\left(L_{O(m)}\right) \le O(m);$$

see [21, 55] for details. By (5.23), (5.24), (5.25), and Weyl's theorem, we immediately have the following theorem.

Theorem 5.8. *Let T_{mn} be defined by* (5.21) *with a generating function f in the Wiener class. Then for all $\epsilon > 0$, there exists an $N > 0$ such that for all $n > N$ and all $m > 0$, at most $O(m)$ eigenvalues of $c_F^{(1)}(T_{mn}) - T_{mn}$ have absolute values larger than ϵ.*

If $f_{\min} > 0$, by Theorem 5.7 and Theorem 5.2(iv), we then have

$$0 < f_{\min} \le \lambda_{\min}(T_{mn}) \le \lambda_{\min}\big(c_F^{(1)}(T_{mn})\big) \le \lambda_{\max}\big(c_F^{(1)}(T_{mn})\big) \le \lambda_{\max}(T_{mn}) \le f_{\max}.$$

Hence $\|\big(c_F^{(1)}(T_{mn})\big)^{-1}\|_2$ is uniformly bounded. By noting that

$$\big(c_F^{(1)}(T_{mn})\big)^{-1}T_{mn} = I - \big(c_F^{(1)}(T_{mn})\big)^{-1}\big(c_F^{(1)}(T_{mn}) - T_{mn}\big),$$

we have the following immediate corollary.

Corollary 5.9. *Let T_{mn} be defined by* (5.21) *with a positive generating function f in the Wiener class. Then for all $\epsilon > 0$, there exists an $N > 0$ such that for all $n > N$ and all $m > 0$, at most $O(m)$ eigenvalues of $\big(c_F^{(1)}(T_{mn})\big)^{-1}T_{mn} - I$ have absolute values larger than ϵ.*

As a consequence, the spectrum of $\big(c_F^{(1)}(T_{mn})\big)^{-1}T_{mn}$ is clustered around 1 except for at most $O(m)$ outlying eigenvalues which are also bounded. By Theorem 1.10 and Corollary 5.9, when the PCG method is used to solve $T_{mn}\mathbf{u} = \mathbf{b}$, the convergence rate will be fast. We recall that in Section 5.2.1, the algorithm requires $O(mn\log^2 m + mn\log n)$ operations in each iteration. Thus, the total complexity of the algorithm remains $O(mn\log^2 m + mn\log n)$.

5.3.2 Convergence rate of $c_{F,F}^{(2)}(T_{mn})$

For the convergence rate of the PCG method with $c_{F,F}^{(2)}(T_{mn})$, we note that

$$\begin{aligned}
c_{F,F}^{(2)}(T_{mn}) - T_{mn} &= c_{F,F}^{(2)}(T_{mn}) - \tilde{c}_F^{(1)}(T_{mn}) + \tilde{c}_F^{(1)}(T_{mn}) - T_{mn} \\
&= (\tilde{c}_F^{(1)} \circ c_F^{(1)})(T_{mn}) - \tilde{c}_F^{(1)}(T_{mn}) + \tilde{c}_F^{(1)}(T_{mn}) - T_{mn} \\
&= \tilde{c}_F^{(1)}\left(c_F^{(1)}(T_{mn}) - T_{mn}\right) + \tilde{c}_F^{(1)}(T_{mn}) - T_{mn} \\
&= \tilde{c}_F^{(1)}(M_{mn} + L_{O(m)}) + N_{mn} + L_{O(n)} \\
&= \tilde{c}_F^{(1)}(M_{mn}) + \tilde{c}_F^{(1)}(L_{O(m)}) + N_{mn} + L_{O(n)},
\end{aligned}$$

where M_{mn}, N_{mn}, $L_{O(m)}$, and $L_{O(n)}$ are defined similarly as in (5.25) with

$$\|M_{mn}\|_2 < \epsilon, \qquad \|N_{mn}\|_2 < \epsilon$$

and

$$\operatorname{rank}\left(L_{O(m)}\right) \leq O(m), \qquad \operatorname{rank}\left(L_{O(n)}\right) \leq O(n).$$

Note that

$$\|\tilde{c}_F^{(1)}(M_{mn})\|_2 \leq \|\tilde{c}_F^{(1)}\|_2 \|M_{mn}\|_2 \leq \|M_{mn}\|_2 < \epsilon.$$

Also, one can easily show that

$$\operatorname{rank}\big(\tilde{c}_F^{(1)}(L_{O(m)})\big) \leq O(m).$$

We therefore have the following theorem.

Theorem 5.10. *Let T_{mn} be defined by* (5.21) *with a generating function f in the Wiener class. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $m > M$ and all $n > N$, at most $O(m) + O(n)$ eigenvalues of $c_{F,F}^{(2)}(T_{mn}) - T_{mn}$ have absolute values larger than ϵ.*

When f is positive, by Theorem 5.7 and Theorem 5.6(iii), we have

$$0 < f_{\min} \leq \lambda_{\min}(T_{mn}) \leq \lambda_{\min}\big(c_{F,F}^{(2)}(T_{mn})\big) \leq \lambda_{\max}\big(c_{F,F}^{(2)}(T_{mn})\big) \leq \lambda_{\max}(T_{mn}) \leq f_{\max}.$$

Hence $\|\big(c_{F,F}^{(2)}(T_{mn})\big)^{-1}\|_2$ is uniformly bounded. By noting that

$$\big(c_{F,F}^{(2)}(T_{mn})\big)^{-1} T_{mn} = I - \big(c_{F,F}^{(2)}(T_{mn})\big)^{-1} \left(c_{F,F}^{(2)}(T_{mn}) - T_{mn}\right),$$

we then have the following immediate corollary.

Corollary 5.11. *Let T_{mn} be defined by* (5.21) *with a positive generating function f in the Wiener class. Then for all $\epsilon > 0$, there exist M and $N > 0$ such that for all $m > M$ and all $n > N$, at most $O(m) + O(n)$ eigenvalues of $\big(c_{F,F}^{(2)}(T_{mn})\big)^{-1} T_{mn} - I$ have absolute values larger than ϵ.*

As a consequence, the spectrum of $\big(c_{F,F}^{(2)}(T_{mn})\big)^{-1} T_{mn}$ is clustered around 1 except for at most $O(m) + O(n)$ outlying eigenvalues which are also bounded. By Theorem 1.10 and Corollary 5.11, when the PCG method is used to solve $T_{mn}\mathbf{u} = \mathbf{b}$, the convergence rate will be fast. We recall that in Section 5.2.2, the algorithm requires $O(mn \log mn)$ operations in each iteration step. Thus, the total complexity of the algorithm remains $O(mn \log mn)$. Finally, we would like to mention that in general, BCCB preconditioners for BTTB systems are not optimal in the sense that the spectra of preconditioned matrices are not clustered around 1 tightly; i.e., the number of outlying eigenvalues depends on m and n (see [73]).

5.4 Examples

In this section, we apply the PCG method to level-2 symmetric BTTB systems $T_{mn}\mathbf{u} = \mathbf{b}$ with the diagonals of the blocks $T_{(j)}$ in T_{mn} being given by $t_k^{(j)}$. Four different generating sequences were tested:

$$\text{(i)} \qquad t_k^{(j)} = \frac{1}{(|j|+1)(|k|+1)^{1+0.1\times(|j|+1)}}, \qquad j,k = 0, \pm 1, \pm 2, \ldots;$$

$$\text{(ii)} \qquad t_k^{(j)} = \frac{1}{(|j|+1)^{1.1}(|k|+1)^{1+0.1\times(|j|+1)}}, \qquad j,k = 0, \pm 1, \pm 2, \ldots;$$

$$\text{(iii)} \qquad t_k^{(j)} = \frac{1}{(|j|+1)^{1.1} + (|k|+1)^{1.1}}, \qquad j,k = 0, \pm 1, \pm 2, \ldots;$$

$$\text{(iv)} \qquad t_k^{(j)} = \frac{1}{(|j|+1)^{2.1} + (|k|+1)^{2.1}}, \qquad j,k = 0, \pm 1, \pm 2, \ldots.$$

The generating sequences (ii) and (iv) are absolutely summable, while (i) and (iii) are not. Tables 5.1 and 5.2 show the number of iterations required for convergence. The right-hand side vector $\mathbf{b}$ is chosen again to be the vector of all ones. In all cases, we see that as $m = n$ increases, the number of iterations remains roughly a constant for both $c_F^{(1)}(T_{mn})$ and $c_{F,F}^{(2)}(T_{mn})$.

Table 5.1. *Preconditioners used and number of iterations.*

		Sequence (i)			Sequence (ii)		
$n = m$	mn	I	$c_F^{(1)}(T_{mn})$	$c_{F,F}^{(2)}(T_{mn})$	I	$c_F^{(1)}(T_{mn})$	$c_{F,F}^{(2)}(T_{mn})$
8	64	15	6	7	15	5	7
16	256	28	6	8	27	6	8
32	1024	37	6	8	35	6	8
64	4096	45	7	9	41	7	9
128	16384	49	7	9	46	7	9
256	65536	51	7	9	47	7	9

Table 5.2. *Preconditioners used and number of iterations.*

		Sequence (iii)			Sequence (iv)		
$n = m$	mn	I	$c_F^{(1)}(T_{mn})$	$c_{F,F}^{(2)}(T_{mn})$	I	$c_F^{(1)}(T_{mn})$	$c_{F,F}^{(2)}(T_{mn})$
8	64	11	7	7	10	7	7
16	256	27	7	8	16	7	7
32	1024	43	8	8	23	8	8
64	4096	71	8	9	31	8	8
128	16384	104	8	9	36	8	8
256	65536	147	8	9	42	8	8

The MATLAB programs generating Tables 5.1 and 5.2 are given in A.8–A.19. To use them, one just has to run the main program A.8. It will prompt for the input of four parameters: `m` and `n`, the size of the BTTB matrix T_{mn}; `pchoice`, the choice of the preconditioner (either 0 for I, 1 for $c_F^{(1)}(T_{mn})$, or 2 for $c_{F,F}^{(2)}(T_{mn})$); and `fchoice`, the generating sequence used with 1 for (i), etc.

Appendix A

M-files used in the book

In this appendix, Professor Fu-Rong Lin has kindly provided the MATLAB programs used for the numerical examples in this book. The interested readers could study these codes and alter them as needed for their own purposes. For a general guide of MATLAB implementations, we refer readers to [48].

Though the programs are explained in the main text, for readers' convenience, we give some brief explanations here. The programs A.1–A.7 are used for solving Toeplitz systems in Sections 2.5, 3.5, and 4.4. To use them, one just has to run the main program A.1. It will prompt for the input parameters to run the tests in those sections. Program A.2 is used to generate the first column of Toeplitz matrices; and A.3 computes the eigenvalues of circulant matrices. The code of the PCG method is provided in A.5. In each iteration of the PCG method, we are required to solve the preconditioned system and compute the multiplication of a Toeplitz matrix with a vector. They are done by A.6 and A.7, respectively. Program A.4 computes the coefficients of the generalized Jackson kernel $\mathcal{K}_{m,2r}$ defined in (4.2).

The programs in A.8–A.19 are used for solving BTTB systems. To use them, run the main program A.8. The programs in A.9 and A.10 generate the first column and/or the first row of each block of BTTB matrices. The program in A.11 computes the eigenvalues of BCCB matrices. Programs A.12, A.14, and A.18 contain the codes of the PCG method with no preconditioner, the CB preconditioner $c_F^{(1)}(T_{mn})$, and the BCCB preconditioner $c_{F,F}^{(2)}(T_{mn})$, respectively. The programs for computing the eigenvalues of preconditioners $c_F^{(1)}(T_{mn})$ and $c_{F,F}^{(2)}(T_{mn})$ are in A.13 and A.17, respectively. In each iteration of the PCG method, we use A.15 and A.19 to solve the preconditioned system (5.18) and (5.19), respectively. Finally, A.16 multiplies a BTTB matrix with a vector.

Toeplitz Systems

A.1

```
% The main program for solving Toeplitz systems T\bu = {\bf b}.
clear           % release all variables from the MATLAB workspace
tol = 1.0e-7;   % the tolerance for the PCG method
it_max = 4000;  % the maximum number of iterations for the PCG method

n = input('input the size of the Toeplitz system n: ');
disp('choice of preconditioners: ');
disp('0: no preconditioner');
disp('1: T. Chan preconditioner');
disp('2: Strang preconditioner');
disp('3: R. Chan preconditioner');
disp('4: Modified Dirichlet kernel');
disp('5: de la Vallee Poussin kernel');
disp('6: von Hann kernel');
disp('7: Hamming kernel');
disp('8: Bernstein kernel');
disp('9: Generalized Jackson kernel, r=2');
disp('10: Generalized Jackson kernel, r=3');
disp('11: Generalized Jackson kernel, r=4');
disp('12: superoptimal preconditioner');

pchoice = input('input a preconditioner (pchoice): 0~12: ');

fprintf('\n');
disp('choice of entries for Toeplitz matrices: ');
disp('1: for results in Table 2.1');
disp('2: for results in Table 3.3');
disp('3: for results in Table 3.4');
disp('4: for results in Table 4.1 with f(x)=x^4+1');
disp('5: for results in Table 4.1 with f(x)=|x|^3+0.01');
disp('6: for results in Table 4.2 with f(x)=x^2');
disp('7: for results in Table 4.2 with f(x)=x^2(pi^4-x^4)');
disp('8: for results in Table 4.3 with f(x)=x^4');
disp('9: for results in Table 4.3 with f(x)=x^4(pi^2-x^2)');
disp('10: for results in Table 4.4 with f(x)=|x|^3');
disp('11: for results in Table 4.4 with f(x)=sigma-0.3862');

fchoice = input('input an example (fchoice): 1~11: ');

t = kern(n,fchoice); % t is the first column of the Toeplitz matrix T
b = ones(n,1);       % the right-hand side vector b
```

```
[gev,ev] = genevs(t,pchoice);      % compute the eigenvalues; see A.3
ig = zeros(n,1);                   % the initial guess

u = pcg(gev,ev,b,ig,tol, it_max);  % call the PCG method
```

A.2

```
function t = kern(n,fchoice)
% kern generates the first column of Toeplitz matrix T
%      depending on fchoice.
% n: the size of the Toeplitz matrix T;
% fchoice = 1:       example for Chapter 2;
% fchoice = 2 and 3: examples for Chapter 3;
% fchoice = 4 to 9:  examples for Chapter 4;
% t: the first column of the Toeplitz matrix.

t = zeros(1,n);

if fchoice == 1
    t(1)   = 2;
    t(2:n) = (1+sqrt(-1))./((2:n).^1.1);
elseif fchoice == 2
    i      = sqrt(-1);
    t(1)   = 4.2;
    k      = 1:n-1;
    t(2:n) = (exp(i*k.*log(k)))./k;
elseif fchoice == 3
    i      = sqrt(-1);
    t(1)   = 6.5;
    k      = 1:n-1;
    t(2:n) = (exp(i*k.*log(k)))./sqrt(k);
elseif fchoice == 4   % f(x) = x^4+1;
    pi2      = pi*pi;
    k        = (1:n-1).^2;
    t(1)     = pi2*pi2/5 + 1;
    t(2:n)   = (4*pi2-24./k)./k;
    t(2:2:n) = -t(2:2:n);
elseif fchoice == 5   % f(x) = |x|^3+0.01;
    k      = (1:n-1).^2;
    tp1    = ones(1,n-1); tp1(1:2:n-1) = -1;
    tp2    = zeros(1,n-1); tp2(1:2:n-1) = 12;
    t(1)   = 1/4*pi^3+0.01;
    t(2:n) = ((3*pi)*tp1)./k+(tp2/pi)./(k.^2);
    t(2:n) = (3*pi*tp1+(tp2/pi)./k)./k;
```

```
elseif fchoice == 6   % f(x) = x^2;
    t(1)     = pi*pi/3;
    t(2:n)   = 2./((1:n-1).^2);
    t(2:2:n) = -t(2:2:n);
elseif fchoice == 7   % f(x) = x^2(pi^4-x^4);
    pi2      = pi*pi; pi4 = pi2*pi2;
    tp       = 194.8181820680048-6*pi4;
    k        = (1:n-1).^2;
    t(1)     = -1/7*pi2*pi4+32.46969701133414*pi2;
    t(2:n)   = ((-720./k+pi2*120)./k+tp)./k;
    t(2:2:n) = -t(2:2:n);
elseif fchoice == 8   % f(x) = x^4;
    pi2      = pi*pi;
    k        = (1:n-1).^2;
    t(1)     = pi2*pi2/5;
    t(2:n)   = (4*pi2-24./k)./k;
    t(2:2:n) = -t(2:2:n);
elseif fchoice == 9   % f(x) = x^4(pi^2-x^2);
    pi2      = pi*pi; pi4 = pi2*pi2;
    k        = (1:n-1).^2;
    t(1)     = (-1/7*pi2+1.97392088021787)*pi4;
    c1       = 39.47841760435743*pi2-6*pi4;
    c2       = 120*pi2-236.8705056261446;
    t(2:n)   = ((-720./k+c2)./k+c1)./k;
    t(2:2:n) = -t(2:2:n);
elseif fchoice == 10  % f(x) = |x|^3;
    k      = (1:n-1).^2;
    tp1    = ones(1,n-1); tp1(1:2:n-1) = -1;
    tp2    = zeros(1,n-1); tp2(1:2:n-1) = 12;
    t(1)   = 1/4*pi^3;
    t(2:n) = ((3*pi)*tp1)./k+(tp2/pi)./(k.^2);
    t(2:n) = (3*pi*tp1+(tp2/pi)./k)./k;
elseif fchoice == 11  % f(x) = sigma-0.3862;
    t(1)   = 0.6138;
    m      = min(n,1024);
    t(2:m) = 1./(1+(1:m-1));
end
```

A.3

```
function [gev,ev] = genevs(t,pchoice)
% genevs computes the eigenvalues of the circulant matrix in which T
%     is embedded and the eigenvalues of circulant preconditioner C.
% t:       the first column of the Toeplitz matrix T;
% pchoice: choice of preconditioner;
% gev:     the eigenvalues of the circulant matrix in which
%               the Toeplitz matrix T is embedded;
% ev:      the eigenvalues of the circulant preconditioner C.

n   = length(t);
t1  = conj(t(n:-1:2));   % last column of T
gev = real(fft([t 0 t1].'));

if pchoice == 1       % T. Chan's preconditioner
    coef   = 1/n:1/n:1-1/n;
    ev     = fft([t(1) (1-coef).*t(2:n)+coef.*t1])';
elseif pchoice == 2   % Strang's preconditioner.
    ev     = fft([t(1:n/2) 0 conj(t(n/2:-1:2))].');
elseif pchoice == 3   % R. Chan's preconditioner
    ev     = fft([t(1) t(2:n)+t1].');
elseif pchoice == 4   % Modified Dirichlet kernel
    c    = [t(1) t(2:n)+t1].';
    c(2) = c(2)-0.5*t1(1);
    c(n) = c(n)-0.5*t(n);
    ev   = fft(c);
elseif pchoice == 5   % de la Vallee Poussin kernel
    c          = zeros(1,n);
    c(1)       = t(1);
    m          = floor(n/2);
    c(2:m+1)   = (1:m).*conj(t(2*m:-1:m+1))/m+t(2:m+1);
    c(m+2:2*m) = (m-1:-1:1).*t(m+2:2*m)/m+conj(t(m:-1:2));
    ev = fft(c)';
elseif pchoice == 6   % von Hann kernel
    tp   = pi/(2*n);
    coef = (cos(tp:tp:(n-1)*tp)).^2;
    c    = [t(1) coef.*t(2:n)+(1-coef).*t1];
    ev   = fft(c)';
elseif pchoice == 7   % Hamming kernel
    tp   = pi/(2*n);
    coef = (cos(tp:tp:(n-1)*tp)).^2;
    c    = [t(1) 0.46*(coef.*t(2:n)+(1-coef).*t1)+0.54*(t(2:n)+t1)];
    ev   = fft(c)';
elseif pchoice == 8   % Bernstein kernel
    tp   = pi/n;
    coef = 0.5*(1+exp((tp:tp:(n-1)*tp)*i));
    c    = [t(1) coef.*t(2:n)+(1-coef).*t1];
    ev   = fft(c)';
elseif pchoice == 9   % Generalized Jackson kernel: r = 2
    coef = convol(n,2);
    c    = [t(1)*coef(1) coef(2:n).*t(2:n)+coef(n:-1:2).*t1];
    ev   = fft(c)';
```

```
elseif pchoice == 10  % Generalized Jackson kernel: r = 3
    coef = convol(n,3);
    c    = [t(1)*coef(1) coef(2:n).*t(2:n)+coef(n:-1:2).*t1];
    ev   = fft(c)';
elseif pchoice == 11  % Generalized Jackson kernel: r = 4
    coef = convol(n,4);
    c    = [t(1)*coef(1) coef(2:n).*t(2:n)+coef(n:-1:2).*t1];
    ev   = fft(c)';
elseif pchoice == 12  % Superoptimal preconditioner
    h    = zeros(n,1);   % h: first column of the circulant part
    s    = zeros(n,1);   % s: first column of the skew-circulant part
    h(1) = 0.5*t(1);
    s(1) = h(1);
    t    = t.';
    t1   = t1.';

    h(2:n) = 0.5*(t(2:n) + t1);
    s(2:n) = t(2:n)-h(2:n);

    ev1  = fft(h);
    coef = (1:-2/n:2/n-1)';
    c    = coef.*s;
    % first column of T. Chan's preconditioner
    % for the skew-circulant part
    ev2  = fft(c);

    d   = (exp((0:1/n:1-1/n)*pi*i)).';
    s   = s.*d;
    sev = fft(s);   % eigenvalues of the skew-circulant part
    sev = sev.*conj(sev);
    s   = ifft(s);
    s   = conj(d).*s;
    h   = coef.*s;
    ev3 = fft(h);

    ev = (abs(ev1).^2 + 2*ev1.*ev2+ev3)./ev1;
elseif pchoice == 0
    ev = ones(n,1);
end
ev = real(ev);
```

A.4

```
function coef = convol(n,r)
% convol computes the coefficients for the generalized Jackson kernel.
%    K_{m,2r}, see (4.2).
% n:    the size of the Toeplitz matrix T;
% r:    2, 3, or 4;
% coef: the coefficients {b_k^(m,r):|k|<=r(m-1)} of
%           the generalized Jackson kernel K_{m,2r}.

m = floor(n/r);
a = 1:-1/m:1/m;

r0 = 1;
coef = [a(m:-1:2) a];
while r0 < r
   M = (2*r0+3)*m;
   b1 = zeros(M,1);
   c = zeros(M,1);
   c(1:m) = a;
   c(M:-1:M-m+2) = a(2:m);
   b1(m:m+2*r0*(m-1)) = coef;
   tp = ifft(fft(b1).*fft(c));
   coef = real(tp(1:2*(r0+1)*(m-1)+1));
   r0 = r0+1;
end
M = r*(m-1)+1;
coef = [coef(M:-1:1)' zeros(1,n-M)]';
coef = coef';
```

A.5

```
function u = pcg(gev,ev,b,ig,tol,it_max)
% pcg uses PCG to solve the Toeplitz system Tx=b with circulant
%      preconditioner C.
% gev:    the eigenvalues of the circulant matrix in which
%           the Toeplitz matrix T is embedded, see A.3;
% ev:     the eigenvalues of the circulant preconditioner C
%           which depend on the choice of preconditioner, see A.3;
% b:      the right-hand side vector b;
% ig:     the initial guess;
% tol:    the tolerance;
% it_max: the maximal number of iterations;
% u:      the output solution.

r    = b-tx(ig,gev);
u    = ig;
aa   = norm(r);
t1   = 1;
```

```
d    = zeros(length(b),1);
iter = 1;
e(1) = aa;
fprintf('\n at step   %1.0f, residual=%e', iter-1, e(iter));

while iter < it_max & e(iter)/e(1) > tol,
   z       = cinvx(r,ev);
   t1old   = t1;
   t1      = z'*r;
   beta    = t1/t1old;
   d       = z+beta*d;
   s       = tx(d,gev);
   suma    = d'*s;
   tau     = t1/suma;
   u       = u+tau*d;
   r       = r-tau*s;
   iter    = iter+1;
   e(iter) = sqrt(r'*r);
   fprintf('\n at step   %1.0f, relative residual = %e',...
           iter-1,e(iter)/e(1));
end
if (iter == it_max),
   fprintf('\n Maximum iterations reached');
end
```

A.6

```
function y = cinvx(d,ev)
% cinvx solves the preconditioning (circulant) system Cy=d.
% d:  the right-hand side vector;
% ev: the eigenvalues of the circulant matrix C;
% y:  the output vector.

y = ifft(fft(d)./ev);
if norm(imag(y)) < 1.0e-14   % check if v is real
   y = real(y);
end
```

A.7

```
function y = tx(v,gev)
% tx multiplies the Toeplitz matrix T with a vector v.
% v:   the vector to be applied;
% gev: the eigenvalues of the circulant matrix
%      in which T is embedded, see A.3;
% y: the result of the multiplication.

n = length(v);
y = zeros(2*n,1);
y(1:n) = v;

y = ifft(fft(y).*gev);
y = y(1:n);
if norm(imag(y)) < 1.0e-14   % check if y is real
   y = real(y);
end
```

BTTB Systems

A.8

```
% The main program for solving BTTB systems T_{mn}u = b.
clear    % release all variables from the MATLAB workspace
m = input('input the number of blocks m: ');
n = input('input the size of each block n: ');

disp('choice of preconditioners: ')
disp('0: No preconditioner');
disp('1: CB preconditioner');
disp('2: BCCB preconditioner');

pchoice = input('input a preconditioner (pchoice): 0, 1, 2: ');

fprintf('\n');
disp('choice of sequences: ');
disp('1: Sequence   (i) in Table 5.1');
disp('2: Sequence  (ii) in Table 5.1');
disp('3: Sequence (iii) in Table 5.2');
disp('4: Sequence  (iv) in Table 5.2');

fchoice = input('choice an example (fchoice): 1~4: ');

ig  = zeros(m*n,1);          % the initial guess
b   = ones(m*n,1);           % the right-hand side vector
t   = fcolrow(m,n,fchoice);  % the first columns and first rows
                             %    of each block of T_{mn}; see A.9.
tev = gentev(t);             % the eigenvalues of the BCCB matrix
                             %    in which T_{mn} is embedded; see A.11
```

```
tol = 1.e-7;
if pchoice == 0
   % call CG method without any preconditioner
   u = cg(b,ig,tev,tol);
end

if pchoice == 1
   ev = gen11ev(t);
   % call PCG method with the CB preconditioner
   u  = pcg11(b,ig,tev,ev,tol);
end

if pchoice == 2
   ev = gen12ev(t);
   % call PCG method with the BCCB preconditioner
   u  = pcg12(b,ig,tev,ev,tol);
end
```

A.9

```
function t = fcolrow(m,n,fchoice)
% fcolrow generates the first columns and first rows of each block of
%    the BTTB matrices T_{mn}. This program can also be used for
%    nonsymmetric BTTB matrices.
% m: the number of blocks;
% n: the size of each block;
% fchoice: the choice of generating function;
% t: (2n)-by-(2m) matrix consists of first columns and first rows
%        of the blocks of the BTTB matrix. Each column of t contains
%        the first column and first row of a Toeplitz block.

m1 = 2*m; n1 = 2*n;
t  = zeros(m1,n1);

for i = 0:m-1,
   for j = 0:n-1,
      t(i+1,j+1) = kern(-i,-j,fchoice);
   end
end

for i = m+1:m1-1,
   for j = 0:n-1,
      t(i+1,j+1) = kern(m1-i,-j,fchoice);
   end
end
```

```
for i = 0:m-1,
   for j = n+1:n1-1,
      t(i+1,j+1) = kern(-i,n1-j,fchoice);
   end
end

for i = m+1:m1-1,
   for j = n+1:n1-1,
      t(i+1,j+1) = kern(m1-i,n1-j,fchoice);
   end
end

t = t.';
```

A.10

```
function y = kern(k,j,fchoice)
% kern computes the (j,1) entry of the (k,1) block of the
%      BTTB matrix T_{mn}.
% fchoice: the choice of generating function;
% y: the output entry.

j = abs(j); k = abs(k); % for double symmetric BTTB matrix

if fchoice == 1,       % Sequence (i)
   y = 1./((j+1)*(k+1)^(1.1+0.1*j));
elseif fchoice == 2,   % Sequence (ii)
   y = 1./((j+1)^1.1*(k+1)^(1.1+0.1*j));
elseif fchoice == 3,   % Sequence (iii)
   y = 1./((j+1)^1.1+(k+1)^1.1);
elseif fchoice == 4,   % Sequence (iv)
   y = 1./((j+1)^2.1+(k+1)^2.1);
end
```

A.11

```
function tev = gentev(t)
% gentev computes eigenvalues of the BCCB matrix in which
%        the BTTB matrix T_{mn} is embedded;
% t:   the matrix generated by fcolrow.m, see A.9;
% tev: the output eigenvalues of the BCCB matrix.

tev = fft2(t);
```

A.12

```
function u = cg(b,ig,tev,tol)
% cg uses the CG method for solving T_{mn}u=b without any preconditioner.
% b:   the right-hand side vector;
% ig:  the initial guess for the CG method;
% tev: the eigenvalues generated by gentev.m, see A.11;
% tol: the tolerance;
% u:   the output solution.

mmax = 1000;  % the maximal number of iterations
u = ig;

r    = b-tx(tev,u);
e(1) = norm(r);
fprintf('\n Initial residual =   %e',e(1));
iter = 1;
t1   = 1.0;
d    = zeros(length(ig),1);
while iter < mmax & e(iter)/e(1) > tol,
   z       = r;
   t1old   = t1;
   t1      = z'*r;
   beta    = t1/t1old;
   d       = z+beta*d;
   s       = tx(tev,d);
   suma    = d'*s;
   tau     = t1/suma;
   u       = u+tau*d;
   r       = r-tau*s;
   iter    = iter+1;
   e(iter) = norm(r);
   fprintf('\n at step   %1.0f, relative residual = %e',...
            iter-1,e(iter)/e(1));
end

if (iter == mmax),
   fprintf('\n Maximum iterations reached');
end
```

A.13

```
function ev = genl1ev(t)
% ev computes eigenvalues of circulant blocks of the CB preconditioner.
% t:  the matrix generated by fcolrow.m, see A.9;
% ev: n-by-(2m) matrix, each column consists of eigenvalues of a
%     circulant block.

[n,m]  = size(t);
n      = n/2;
ev     = zeros(n,m);
v      = zeros(n,m);
v(1,:) = t(1,:);
for i = 2:n,
   v(i,:) = ((n-(i-1))*t(i,:)+(i-1)*t(n+i,:))/n;
end
ev = fft(v);
```

A.14

```
function u = pcgl1(b,ig,tev,ev,tol)
% pcgl1 uses the PCG method for solving T_{mn}u=b with the
%       CB preconditioner.
% b:   the right-hand side vector;
% ig:  the initial guess;
% tev: the matrix generated by gentev.m, see A.11;
% ev:  the matrix generated by genl1ev.m, see A.13;
% tol: the tolerance;
% u:   the output solution.

mmax = 1000;  % the maximal number of iterations
u = ig;

r = b-tx(tev,u);
e(1) = norm(r);
fprintf('\n Initial residual = %e',e(1));

iter = 1;
t1   = 1.0;
d    = zeros(length(ig),1);
while iter < mmax & e(iter)/e(1) > tol,
   z       = l1cinvx(ev,r);
   t1old   = t1;
   t1      = z'*r;
   beta    = t1/t1old;
   d       = z+beta*d;
   s       = tx(tev,d);
   suma    = d'*s;
   tau     = t1/suma;
   u       = u+tau*d;
   r       = r-tau*s;
   iter    = iter+1;
   e(iter) = norm(r);
   fprintf('\n at step   %1.0f, relative residual = %e',...
           iter-1,e(iter)/e(1));
end

if (iter == mmax),
   fprintf('\n Maximum iterations reached');
end
```

A.15

```
function y = l1cinvx(ev,d)
% l1cinvx solves the preconditioning system Cy=d with
%         CB preconditioner.
% ev: the matrix generated by genl1ev.m, see A.13;
% d:  the right-hand side vector;
% y:  the output solution.

[n,m] = size(ev);
m     = m/2;
rex   = reshape(d,n,m);
rex   = fft(rex);

for i = 1:n,
   A = toeplitz(ev(i,1:m),[ev(i,1),ev(i,2*m:-1:m+2)]);
   rex(i,:) = (A\((rex(i,:))'))';
% We may solve the Toeplitz systems by the PCG methods or
%    by fast direct methods
end

rex = ifft(rex);
y   = reshape(rex,m*n,1);
```

A.16

```
function y = tx(tev,v)
% tx multiplies the BTTB matrix T_{mn} with vector v.
% tev: the matrix generated by gentev.m, see A.11;
% y:   the output vector.

[n1,m1] = size(tev);
m       = m1/2;

n       = n1/2;
v       = reshape(v,n,m);
```

```
ev      = zeros(n1,m1);
ev(1:n,1:m) = v;

y   = fft2(ev);
y   = tev.*y;
y   = ifft2(y);
y   = y(1:n,1:m);
y   = reshape(y,m*n,1);
```

A.17

```
function ev = genl2ev(t)
% genl2ev computes eigenvalues of the BCCB preconditioner.
% t:  the matrix generated by fcolrow.m, see A.9;
% ev: all eigenvalues of the BCCB preconditioner.

[n,m]  = size(t);
n      = n/2;
v      = zeros(n,m);

v(1,:) = t(1,:);
for i = 2:n
   v(i,:) = ((n-(i-1))*t(i,:)+(i-1)*t(n+i,:))/n;
end
v = fft(v);  % CB eigenvalues.

v = v.';
m = m/2;
ev(1,:) = v(1,:);
for i = 2:m
   ev(i,:) = ((m-(i-1))*v(i,:)+(i-1)*v(m+i,:))/m;
end

if min(n,m) >= 2
   ev  = fft(ev);
end
ev = ev.';
```

A.18

```
function u = pcgl2(b,ig,tev,ev,tol)
% pcgl2 uses PCG method for solving T_{mn}u=b with the
%       BCCB preconditioner.
% b:   the right-hand side vector;
% ig:  the initial guess;
% tev: the matrix generated by gentev.m, see A.11;
% ev:  the matrix generated by genl2ev.m, see A.17;
% tol: the tolerance;
% u:   the output solution.

mmax = 1000;   % the maximal number of iterations.
u = ig;

r    = b-tx(tev,u);
e(1) = norm(r);
fprintf('\n Initial residual = %e',e(1));

iter = 1;
t1   = 1.0;
d    = zeros(length(ig),1);
while iter < mmax & e(iter)/e(1) > tol,
   z       = l2cinvx(ev,r);
   t1old   = t1;
   t1      = z'*r;
   beta    = t1/t1old;
   d       = z+beta*d;
   s       = tx(tev,d);
   suma    = d'*s;
   tau     = t1/suma;
   u       = u+tau*d;
   r       = r-tau*s;
   iter    = iter+1;
   e(iter) = norm(r);
   fprintf('\n at step   %1.0f, relative residual = %e',...
            iter-1,e(iter)/e(1));
end

if (iter == mmax),
   fprintf('\n Maximum iterations reached');
end
```

A.19

```
function y = l2cinvx(ev,d)
% l2cinvx solves the preconditioning system Cy=d for the
%         BCCB preconditioner.
% ev: the matrix generated by genl2ev.m, see A.17;
% d:  the right-hand side vector;
% y:  the output solution.
```

```
[n,m] = size(ev);
rex   = reshape(d,n,m);
rex   = fft2(rex);
rex   = rex./ev;
rex   = ifft2(rex);
y     = reshape(rex,n*m,1);
```

Bibliography

[1] S. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[2] G. S. Ammar and W. B. Gragg, *Superfast Solution of Real Positive Definite Toeplitz Systems*, SIAM J. Matrix Anal. Appl., Vol. 9 (1988), pp. 61–76.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1996.

[4] F. Di Benedetto, *Analysis of Preconditioning Techniques for Ill-Conditioned Toeplitz Matrices*, SIAM J. Sci. Comput., Vol. 16 (1994), pp. 682–697.

[5] F. Di Benedetto, C. Estatico, and S. Serra Capizzano, *Superoptimal Preconditioned Conjugate Gradient Iteration for Image Deblurring*, SIAM J. Sci. Comput., Vol. 26 (2005), pp. 1012–1035.

[6] F. Di Benedetto and S. Serra Capizzano, *A Note on the Superoptimal Matrix Algebra Operators*, Linear Multilinear Algebra, Vol. 50 (2002), pp. 343–372.

[7] D. Bertaccini, *A Circulant Preconditioner for the Systems of LMF-Based ODE Codes*, SIAM J. Sci. Comput., Vol. 22 (2000), pp. 767–786.

[8] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[9] D. Bini and F. Di Benedetto, *A New Preconditioner for the Parallel Solution of Positive Definite Toeplitz Systems*, in Proceedings of the 2nd AMC Symposium on Parallel Algorithms and Architectures, Crete, Greece, 1990, pp. 220–223.

[10] D. Bini and P. Favati, *On a Matrix Algebra Related to the Discrete Hartley Transform*, SIAM J. Matrix Anal. Appl., Vol. 14 (1993), pp. 500–507.

[11] R. Bitmead and B. Anderson, *Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations*, Linear Algebra Appl., Vol. 34 (1980), pp. 103–116.

[12] E. Boman and I. Koltracht, *Fast Transform Based Preconditioners for Toeplitz Equations*, SIAM J. Matrix Anal. Appl., Vol. 16 (1995), pp. 628–645.

[13] R. Brent, F. Gustavson, and D. Yun, *Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximants*, J. Algorithms, Vol. 1 (1980), pp. 259–295.

[14] E. Brigham, *The Fast Fourier Transform and Its Applications*, Prentice–Hall, Englewood Cliffs, NJ, 1988.

[15] J. R. Bunch, *Stability of Methods for Solving Toeplitz Systems of Equations*, SIAM J. Sci. Stat. Comput., Vol. 6 (1985), pp. 349–364.

[16] R. H. Chan, *The Spectrum of a Family of Circulant Preconditioned Toeplitz Systems*, SIAM J. Numer. Anal., Vol. 26 (1989), pp. 503–506.

[17] R. H. Chan, *Circulant Preconditioners for Hermitian Toeplitz Systems*, SIAM J. Matrix Anal. Appl., Vol. 10 (1989), pp. 542–550.

[18] R. H. Chan, *Toeplitz Preconditioners for Toeplitz Systems with Nonnegative Generating Functions*, IMA J. Numer. Anal., Vol. 11 (1991), pp. 333–345.

[19] R. H. Chan and T. F. Chan, *Circulant Preconditioners for Elliptic Problems*, Numer. Linear Algebra Appl., Vol. 1 (1992), pp. 77–101.

[20] R. H. Chan, T. F. Chan, and C.-K. Wong, *Cosine Transform Based Preconditioners for Total Variation Deblurring*, IEEE Trans. Image Process, Vol. 8 (1999), pp. 1472–1478.

[21] R. H. Chan and X.-Q. Jin, *A Family of Block Preconditioners for Block Systems*, SIAM J. Sci. Stat. Comput., Vol. 13 (1992), pp. 1218–1235.

[22] R. H. Chan, X.-Q. Jin, and M.-C. Yeung, *The Circulant Operator in the Banach Algebra of Matrices*, Linear Algebra Appl., Vol. 149 (1991), pp. 41–53.

[23] R. H. Chan, X.-Q. Jin, and M.-C. Yeung, *The Spectra of Super-Optimal Circulant Preconditioned Toeplitz Systems*, SIAM J. Numer. Anal., Vol. 28 (1991), pp. 871–879.

[24] R. H. Chan and M. K. Ng, *Conjugate Gradient Methods for Toeplitz Systems*, SIAM Rev., Vol. 38 (1996), pp. 427–482.

[25] R. H. Chan, M. K. Ng, and C.-K. Wong, *Sine Transform Based Preconditioners for Symmetric Toeplitz Systems*, Linear Algebra Appl., Vol. 232 (1996), pp. 237–259.

[26] R. H. Chan, M. K. Ng, and A. M. Yip, *The Best Circulant Preconditioners for Hermitian Toeplitz Matrices* II*: The Multiple Case*, Numer. Math., Vol. 92 (2002), pp. 17–40.

[27] R. H. Chan and G. Strang, *Toeplitz Equations by Conjugate Gradients with Circulant Preconditioner*, SIAM J. Sci. Stat. Comput., Vol. 10 (1989), pp. 104–119.

[28] R. H. Chan and P. T. P. Tang, *Fast Band-Toeplitz Preconditioners for Hermitian Toeplitz Systems*, SIAM J. Sci. Comput., Vol. 15 (1994), pp. 164–171.

[29] R. H. Chan and M.-C. Yeung, *Circulant Preconditioners for Toeplitz Matrices with Positive Continuous Generating Functions*, Math. Comp., Vol. 58 (1992), pp. 233–240.

[30] R. H. Chan and M.-C. Yeung, *Circulant Preconditioners Constructed from Kernels*, SIAM J. Numer. Anal., Vol. 29 (1992), pp. 1093–1103.

[31] R. H. Chan and M.-C. Yeung, *Jackson's Theorem and Circulant Preconditioned Toeplitz Systems*, J. Approx. Theory, Vol. 70 (1992), pp. 191–205.

[32] R. H. Chan, A. M. Yip, and M. K. Ng, *The Best Circulant Preconditioners for Hermitian Toeplitz Systems*, SIAM J. Numer. Anal., Vol. 38 (2000), pp. 876–896.

[33] T. F. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Stat. Comput., Vol. 9 (1988), pp. 766–771.

[34] T. F. Chan and P. Hansen, *A Look-Ahead Levinson Algorithm for General Toeplitz Systems*, IEEE Trans. Signal Process., Vol. 40 (1992), pp. 1079–1090.

[35] E. Cheney, *Introduction to Approximation Theory*, 2nd edition, AMS Chelsea, Providence, RI, 1998.

[36] W.-K. Ching, *Iterative Methods for Queuing and Manufacturing Systems*, Springer-Verlag, London, 2001.

[37] P. Davis, *Circulant Matrices*, 2nd edition, AMS Chelsea, Providence, RI, 1994.

[38] P. Delsarte, Y. Genin, and Y. Kamp, *A Generalization of the Levinson Algorithm for Hermitian Toeplitz Matrices with Any Rank Profile*, IEEE Trans. Acoust. Speech Signal Process., Vol. 33 (1985), pp. 964–971.

[39] R. Freund, *A Look-Ahead Bareiss Algorithm for General Toeplitz Matrices*, Numer. Math., Vol. 68 (1994), pp. 35–69.

[40] R. Freund and H.-Y. Zha, *Formally Biorthogonal Polynomials and a Look-Ahead Levinson Algorithm for General Toeplitz Systems*, Linear Algebra Appl., Vol. 188/189 (1993), pp. 255–303.

[41] G. Golub and C. van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, MD, 1996.

[42] M. Gover and S. Barnett, *Inversion of Toeplitz Matrices Which Are Not Strongly Nonsingular*, IMA J. Numer. Anal., Vol. 5 (1985), pp. 101–110.

[43] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*, 2nd edition, AMS Chelsea, Providence, RI, 2001.

[44] R. Hamming, *Digital Filters*, 3rd edition, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[45] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.

[46] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-Like Matrices and Operators*, Birkhäuser, Boston, 1984.

[47] W. Henkel, *An Extended Berlekamp-Massey Algorithm for the Inversion of Toeplitz Matrices*, IEEE Trans. Comm., Vol. 40 (1992), pp. 1557–1561.

[48] D. J. Higham and N. J. Higham, *MATLAB Guide*, SIAM, Philadelphia, 2000.

[49] S. Holmgren and K. Otto, *Iterative Solution Methods and Preconditioners for Block-Tridiagonal Systems of Equations*, SIAM J. Matrix Anal. Appl., Vol. 13 (1992), pp. 863–886.

[50] F. Hoog, *A New Algorithm for Solving Toeplitz Systems of Equations*, Linear Algebra Appl., Vol. 88/89 (1987), pp. 123–138.

[51] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[52] R. Horn and C. Johnson, *Topics on Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[53] T. Huckle, *Circulant and Skewcirculant Matrices for Solving Toeplitz Matrix Problems*, SIAM J. Matrix Anal. Appl., Vol. 13 (1992), pp. 767–777.

[54] X.-Q. Jin, *Hartley Preconditioners for Toeplitz Systems Generated by Positive Continuous Functions*, BIT, Vol. 34 (1994), pp. 367–371.

[55] X.-Q. Jin, *Developments and Applications of Block Toeplitz Iterative Solvers*, Kluwer Academic Publishers, Dordrecht, The Netherlands, and Science Press, Beijing, China, 2002.

[56] X.-Q. Jin and Y.-M. Wei, *Numerical Linear Algebra and Its Applications*, Science Press, Beijing, China, 2004.

[57] T. Kailath and A. H. Sayed, editors, *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, 1999.

[58] Y. Katznelson, *An Introduction to Harmonic Analysis*, Dover, New York, 1976.

[59] T.-K. Ku and C.-C. J. Kuo, *Design and Analysis of Toeplitz Preconditioners*, IEEE Trans. Signal Process., Vol. 40 (1992), pp. 129–141.

[60] T.-K. Ku and C.-C. J. Kuo, *Spectral Properties of Preconditioned Rational Toeplitz Matrices*, SIAM J. Matrix Anal. Appl., Vol. 14 (1993), pp. 146–165.

[61] T.-K. Ku and C.-C. J. Kuo, *Spectral Properties of Preconditioned Rational Toeplitz Matrices: The Nonsymmetric Case*, SIAM J. Matrix Anal. Appl., Vol. 14 (1993), pp. 521–544.

[62] N. Levinson. *The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction*, J. Math. Phys., Vol. 25 (1946), pp. 261–278.

[63] F.-R. Lin, X.-Q. Jin, and S.-L. Lei, *Strang-Type Preconditioners for Solving Linear Systems from Delay Differential Equations*, BIT, Vol. 43 (2003), pp. 136–149.

[64] G. Lorentz, *Approximation of Functions*, Holt, Rinehart and Winston, New York, 1966.

[65] I. Natanson, *Theory of Functions of a Real Variable*, Vol. II, Frederick Ungar, New York, 1967.

[66] M. K. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, Oxford, UK, 2004.

[67] J. Olkin, *Linear and Nonlinear Deconvolution Problems*, Ph.D. thesis, Rice University, Houston, TX, 1986.

[68] D. Potts and G. Steidl, *Preconditioners for Ill-Conditioned Toeplitz Matrices*, BIT, Vol. 39 (1999), pp. 513–533.

[69] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.

[70] S. Serra, *Optimal, Quasi-Optimal and Superlinear Band-Toeplitz Preconditioners for Asymptotically Ill-Conditioned Positive Definite Toeplitz Systems*, Math. Comp., Vol. 65 (1997), pp. 651–665.

[71] S. Serra, *On the Extreme Eigenvalues of Hermitian (Block) Toeplitz Matrices*, Linear Algebra Appl., Vol. 270 (1998), pp. 109–129.

[72] S. Serra Capizzano, *Toeplitz Preconditioners Constructed from Linear Approximation Processes*, SIAM J. Matrix Anal. Appl., Vol. 20 (1998), pp. 446–465.

[73] S. Serra Capizzano and E. E. Tyrtyshnikov, *Any Circulant-Like Preconditioner for Multilevel Matrices Is Not Superlinear*, SIAM J. Matrix Anal. Appl., Vol. 21 (1999), pp. 431–439.

[74] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Stud. Appl. Math., Vol. 74 (1986), pp. 171–176.

[75] G. Strang, *Introduction to Applied Mathematics*, Wellesley–Cambridge Press, Wellesley, MA, 1986.

[76] P. Swarztrauber, *Multiprocessor FFTs*, Parallel Comput., Vol. 5 (1987), pp. 197–210.

[77] D. Sweet, *The Use of Pivoting to Improve the Numerical Performance of Algorithms for Toeplitz Matrices*, SIAM J. Matrix Anal. Appl., Vol. 14 (1993), pp. 468–493.

[78] O. Toeplitz, *Zur Theorie der quadratischen und bilinearen Formen von unendlichvielen Veränderlichen.* I. *Teil: Theorie der L-Formen.*, Math. Ann., Vol. 70 (1911), pp. 351–376.

[79] L. N. Trefethen, *Approximation Theory and Numerical Linear Algebra*, in Algorithms for Approximation II, J. Mason and M. Cox, editors, Chapman and Hall, London, 1990, pp. 336–360.

[80] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[81] W. F. Trench, *An Algorithm for the Inversion of Finite Toeplitz Matrices*, SIAM J. Appl. Math., Vol. 12 (1964), pp. 515–522.

[82] E. E. Tyrtyshnikov, *Optimal and Superoptimal Circulant Preconditioners*, SIAM J. Matrix Anal. Appl., Vol. 13 (1992), pp. 459–473.

[83] E. E. Tyrtyshnikov, *A Unifying Approach to Some Old and New Theorems on Distribution and Clustering*, Linear Algebra Appl., Vol. 232 (1996), pp. 1–43.

[84] H. van der Vorst, *Preconditioning by Incomplete Decompositions*, Ph.D. thesis, Rijksuniversiteit te Utrecht, Utrecht, The Netherlands, 1982.

[85] J. Walker, *Fourier Analysis*, Oxford University Press, Oxford, UK, 1988.

[86] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

[87] C. Zarowski, *Schur Algorithms for Hermitian Toeplitz, and Hankel Matrices with Singular Leading Principal Submatrices*, IEEE Trans. Signal Process., Vol. 39 (1991), pp. 2464–2480.

[88] S. Zohar, *The Solution of a Toeplitz Set of Linear Equations*, J. Assoc. Comput. Mach., Vol. 21 (1974), pp. 272–276.

[89] A. Zygmund, *Trigonometric Series*, Vol. I, Cambridge University Press, Cambridge, UK, 1968.

Index