CIRCULANT PRECONDITIONERS FROM *B*-SPLINES AND THEIR APPLICATIONS

by

Tat-Ming Tso

Thesis
submitted in partial fulfillment of the
requirements for the degree of

MASTER OF PHILOSOPHY

THE CHINESE UNIVERSITY OF HONG KONG
(DIVISION OF MATHEMATICS)

JULY, 1997

Abstract of thesis entitled

CIRCULANT PRECONDITIONERS FROM B-SPLINES AND THEIR APPLICATIONS

submitted by

Tat-Ming Tso

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in July, 1997

The preconditioned conjugate gradient method is employed to solve Toeplitz system $A_n x = b$. Some well-known circulant preconditioners were developed in the past. The main aim of this thesis is to develop an $O(n \log n)$ method of constructing a family of circulant preconditioners from B-splines to solve the Toeplitz systems. We find that our preconditioner works better than some well-known circulant preconditioners, such as Strang's, T. Chan's and R. Chan's preconditioners, when solving the Toeplitz systems with the non-negative generating functions f with or without discontinuous points. Though the non-circulant band-Toeplitz preconditioners can work for the non-negative generating functions, the generating function f has to be given explicitly. The beauty of our preconditioner is that it can solve the Toeplitz systems without the knowledge of the underlying generating function.

DECLARATION

The author declares that this thesis represents his own work based on the idea suggested by Prof. Raymond H.F. Chan, the author's supervisor. All the work are done under the supervision of Prof. Raymond H.F. Chan during the period 1993-1997 for the degree of Master of Philosophy at The Chinese University of Hong Kong. The work submitted has not been previously included in a thesis, dissertation or report submitted to any institution for a degree, diploma or other qualification.

Tat-Ming Tso

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my supervisor, Prof. Raymond H.F. Chan, for his inspired guidance, constant encouragement and help throughout the period of my M.Phil. studies and in the preparation of this thesis. I would like to thank my colleagues Dr. K.P. Ng, Mr. C.K. Wong, Mr. M.C. Yeung, Mr. W.K. Ching, Dr. F.R. Lin, Mr. H.M. Zhou, Dr. H.W. Sun, Mr. W.F. Ng, Mr. C.P. Cheung, Mr. F. Chan and Mr. W.C. Tang for their many helpful discussions and encouragements.

CONTENTS

Page

Chapter 1 INTRODUCTION **1**

§1.1 Introduction 1

§1.2 Preconditioned Conjugate Gradient Method 3

§1.3 Outline of Thesis 3

Chapter 2 CIRCULANT AND NON-CIRCULANT PRECONDITIONERS **5**

§2.1 Circulant Matrix 5

§2.2 Circulant Preconditioners 6

§2.3 Circulant Preconditioners from Kernel Function 8

§2.4 Non-circulant Band-Toeplitz Preconditioners 9

Chapter 3 B-SPLINES **11**

§3.1 Introduction 11

§3.2 New Version of B-splines 15

Chapter 4 CIRCULANT PRECONDITIONERS CONSTRUCTED FROM B-SPLINES **24**

Chapter 5 NUMERICAL RESULTS AND CONCLUDING REMARKS **28**

Chapter 6 APPLICATIONS TO SIGNAL PROCESSING **37**

§6.1 Introduction 37

§6.2 Preconditioned regularized least squares 39

§6.3 Numerical Example 40

REFERENCES **43**

Chapter 1

INTRODUCTION

1.1 Introduction

In this thesis we discuss the solutions of a class of symmetric positive definite systems $A_n x = \mathrm{b}$ by the preconditioned conjugate gradient method, where A_n is an n-by-n Hermitian Toeplitz matrix (i.e. the entries of A_n are the same along its diagonals). Clearly, A_n has the following form :

$$A_n = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\ a_1 & a_0 & a_{-1} & & a_{2-n} \\ \vdots & a_1 & a_0 & \ddots & \vdots \\ a_{n-2} & & \ddots & \ddots & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix}. \tag{1.1}$$

In short, if $A_n = [a_{ij}]$, then $a_{ij} = a_{i-j}$.

Toeplitz systems appear in different kinds of applications in mathematics and engineering. In signal processing, solutions of Toeplitz systems are needed to obtain the filter coefficients in the design of recursive digital filters, see Chui and A. Chan [13] and Haykin [21]. Time series analysis involves solutions of Toeplitz systems for the unknown parameters of stationary auto-regressive models, see King et al. [24]. Solutions of partial differential equations, solutions of convolution-type integral equations, Padé approximations and minimum realization problems in control theory are involved in the applications of the Toeplitz systems, see Bunch [7] and the references therein. These applications give a strong motivation to mathematicians and engineers to develop fast algorithms to solve Toeplitz systems.

By applying the direct method such as the Gaussian elimination method to

By applying the direct method such as the Gaussian elimination method to solve the Toeplitz systems, the algorithm requires $O(n^3)$ operations. But since n-by-n Toeplitz matrices are determined by only $(2n-1)$ entries rather than n^2 entries, the solutions of Toeplitz systems are expected to be obtained in less than $O(n^3)$ operations. Other direct methods, for instance, that are based on the Levinson recursion formula are in constant use, see Levinson (1946) [26]. His algorithm requires $O(n^2)$ operations for solving an n-by-n Toeplitz system. Faster algorithms that require $O(n\log^2 n)$ have also been developed, see, for instance, Brent, Gustavson, and Yun (1980) [6], Bitmead and Anderson (1980) [4], and Ammar and Gragg (1988) [1]. The stability properties of these direct methods for symmetric positive definite matrices are discussed in Bunch [7].

In addition to the direct methods for solving Toeplitz systems, much attention has been focused on the iterative method such as the preconditioned conjugate gradient method recently, see the survey paper by Chan and Ng [10].

Let us begin by introducing the notation that will be used throughout the paper. Let $\mathbf{C}_{2\pi}$ be the set of all 2π-periodic continuous real-valued functions defined on $[-\pi,\pi]$. For all $f\in\mathbf{C}_{2\pi}$, let

$$a_k=\frac{1}{2\pi}\int_{-\pi}^{\pi}f(\theta)e^{-ik\theta}d\theta,\quad k=0,\pm1,\pm2,\cdots$$

be the Fourier coefficients of f. For all $n\geq 1$, let A_n be the n-by-n Toeplitz matrix with entries $a_{j,k}=a_{j-k}$, $0\leq j,k<n$. The function f is called the *generating function* of Toeplitz matrices A_n, see Grenander and Szegö [18]. Since f is a real-valued function, we have

$$a_{-k}=\bar{a}_k,\quad k=0,\pm1,\pm2,\cdots.$$

It follows that A_n are Hermitian Toeplitz matrices. Note that when f is an even function, the matrices A_n are real symmetric. The jth partial sum of f is defined as

$$s_j[f](\theta)\equiv\sum_{k=-j}^{j}a_ke^{ik\theta},\qquad\forall\theta\in\mathbb{R}.\tag{1.2}$$

In practical applications, the functions f are given. Examples of generating functions are the kernels of Wiener-Hopf equations, see Gohberg and Fel'dman

[16], the functions which give the amplitude characteristics of recursive digital filters, see Chui and A. Chan [13], the spectral density functions in stationary stochastic processes, see Grenander and Szegö [18], and the point-spread functions in image deblurring, see Jain [23].

1.2 Preconditioned Conjugate Gradient Method

As mentioned earlier, we want to solve the systems $A_n x = b$ by conjugate gradient method. The rate of convergence of the conjugate gradient method depends on the condition number of A_n see Axelsson and Barker [3]. In general, the smaller the condition number of A_n is, the faster the convergence will be. If the condition number of A_n is not small, we can use the method of preconditioning to increase the speed of the convergence rate. More precisely, instead of applying the conjugate gradient method to the system $A_n x =$ b, we apply the method to the transformed system $\hat{A}_n \hat{x} = \hat{b}$ where $\hat{A}_n = M_n^{-1/2} A_n M_n^{-1/2}$, $\hat{x} = M_n^{1/2} x$ and $\hat{b} = M_n^{-1/2} b$. The matrix M_n is called a preconditioner for A_n. It is hoped that the preconditioner should be chosen to minimize the condition number of $M_n^{-1} A_n$ and allow efficient computation of the product $M_n^{-1} v$ for any vector v. The preconditioner M_n for A_n can be viewed as an approximation to A_n that is easily invertible.

1.3 Outline of Thesis

The main aim of this paper is to propose a numerical procedure to construct a new family of circulant preconditioners from B-splines of different orders to solve the Toeplitz systems with non-negative generating function with or without discontinuous points. We will also apply the our circulant preconditioners to 1-dimensional Toeplitz least squares problems and numerical results show that they perform better than the T. Chan's circulant preconditioner.

The outline of the rest of the paper is as follows. In §2, some well-known circulant preconditioners for solving Toeplitz matrices are introduced and they can be expressed in terms of the convolution product of the generating function of the Toeplitz matrix with famous kernel functions such as the Dirichlet and the Fejér kernels. Secondly, we consider a non-circulant band-Toeplitz preconditioner for Toeplitz matrices which can work for the Toeplitz systems with non-negative functions. In §3, we briefly introduce B-splines and normalized B-splines. Futhermore, we construct a translated and scaled version of B-splines. From this new spline, we formulate the sequence of B_n^l and the Fourier transform $\hat{B}_n^l$ of B_n^l. Graphs of $\hat{B}_n^l$ are plotted. They show that $\hat{B}_n^l$ tends to the Dirac delta function δ as n and l becomes larger. In §4, we introduce the numerical procedure for constructing circulant preconditioners from the translated and scaled B-splines. In §5, numerical examples and concluding remarks are given. In §6, we consider the Toeplitz least squares problems and the method of regularization. We report on preliminary numerical experience with applying our preconditioners to 1-dimensional Toeplitz least squares problems.

Chapter 2

CIRCULANT AND NON-CIRCULANT PRECONDITIONERS

In this chapter, we consider solving Toeplitz systems by the preconditioned conjugate gradient method with circulant matrix as preconditioner. Let us begin by introducing the circulant matrix.

2.1 Circulant Matrix

A circulant matrix is a special kind of Toeplitz matrices. An n-by-n *circulant* matrix C_n is defined by

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

where $c_{-k} = c_{n-k}$ for $1 \leq k \leq n-1$. One of the beauties of a circulant matrix is that it can always be diagonalized by the Fourier matrix F_n, i.e.

$$C_n = F_n \Lambda_n F_n^*, \tag{2.1}$$

where the entries of F_n are given by

$$[F_n]_{j,k} = \frac{1}{\sqrt{n}} e^{2\pi ijk/n}, \quad 0 \leq j,k \leq n-1, \tag{2.2}$$

and Λ_n is a diagonal matrix holding the eigenvalues of C_n, see for instance Davis [14]. Therefore, for any vector y, when computing the products such as $C_n y$ and $C_n^{-1} y$, they can be expressed as $F_n \Lambda_n F_n^* y$ and $F_n \Lambda_n^{-1} F_n^* y$ respectively. Notice that they can be efficiently computed by Fast Fourier Transform in $O(n \log n)$ operations.

Another important property of a circulant matrix should be emphasized is the relationship between the first column of a circulant matrix and its eigenvalues. By (2.1) and (2.2), if e_1 and 1_n denote the first unit vector and the vector of all ones respectively, then we have

$$\sqrt{n} C_n e_1 = F_n \Lambda_n 1_n. \tag{2.3}$$

Hence if

$$C_n e_1 = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}, \tag{2.4}$$

then the eigenvalues of C_n are given by

$$\lambda_j(C_n) = (\Lambda_n)_{jj} = \sum_{k=0}^{n-1} c_k e^{2\pi ijk/n}, \quad 0 \leq j < n. \tag{2.5}$$

Conversely, if the eigenvalues of C_n are given by the right hand side of (2.5), then the first column of C_n is given by (2.4). We also remark that λ_j can be found in $O(n \log n)$ operations by taking the Fast Fourier Transform of the first column of C_n.

For an n-by-n Toeplitz matrix A_n, the product $A_n y$ can also be computed by Fast Fourier Transform by first embedding A_n into a $2n$-by-$2n$ circulant matrix:

$$\begin{pmatrix} A_n & * \\ * & A_n \end{pmatrix} \begin{pmatrix} y \\ 0 \end{pmatrix} = \begin{pmatrix} A_n y \\ 0 \end{pmatrix},$$

see Strang [31]. Thus we can carry out the multiplication as described above by the decompositon (2.1). It follows that this multiplication requires $O(2n \log 2n)$ operations.

2.2 Circulant Preconditioners

The idea of using the preconditioned conjugate gradient method with circulant preconditioners for solving positive definite Toeplitz systems was first proposed by Strang [31] in 1986. Instead of solving $A_n x = b$, we solve the preconditioned system $C_n^{-1} A_n x = C_n^{-1} b$ by the conjugate gradient method with C_n being a circulant preconditioner. Since then, many circulant matrices were developed as preconditioners for Toeplitz systems, see for instance [12, 22, 25]. In the following, we introduce some well-known circulant preconditioners such as Strang's, T. Chan's and R. Chan's circulant preconditioners. Befor we begin our discussion, let us recall the construction of Toeplitz matrices. We denote by $A_n[f]$ the n-by-n Toeplitz matrix with entries $a_{j,k} = a_{j-k}$, where for all integer k,

$$a_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \quad k = 0, \pm 1, \pm 2, \cdots$$

be the Fourier coefficients of f. The function f is called the *generating function* of Toeplitz matrix A_n.

2.2.1 Strang's Preconditioner $S_n[f]$.

Given $A_n[f]$, Strang's preconditioner $S_n[f]$ is defined to be the circulant matrix that copies the central diagonals of $A_n[f]$ and reflects them around to complete the circulant, see Strang [31]. More precisely, the kth entry in the first column of $S_n[f]$ is given by

$$(S_n[f])_{k0} = \begin{cases} a_k & 0 \le k \le \lfloor n/2 \rfloor, \\ a_{k-n} & \lfloor n/2 \rfloor < k < n. \end{cases}$$

2.2.2 T. Chan's Preconditioner $T_n[f]$.

Given $A_n[f]$, T. Chan's preconditioner $T_n[f]$ is defined to be the circulant matrix with diagonals that are arithmetic average of the diagonals of $A_n[f]$ (extended to length n by wrap-around when necessary), see T. Chan [12]. More precisely, the entries in the first column of $T_n[f]$ are given by

$$(T_n[f])_{k0} = \frac{1}{n}\{(n-k)a_k + k\bar{a}_{n-k}\}, \quad 0 \le k < n.$$

2.2.3 R. Chan's Preconditioner $R_n[f]$.

Given $A_n[f]$, the corresponding R. Chan's circulant preconditioner $R_n[f]$ has the first column given by

$$(R_n[f])_{k0} = \begin{cases} a_0 & k = 0, \\ a_k + \bar{a}_{n-k} & 0 < k < n, \end{cases}$$

see R. Chan [8].

The convergence results for these circulant preconditioners are all based on the regularity of the generating function f. A general result is that if f is a positive function in the Wiener class, then for large enough n, the preconditioned matrix has eigenvalues clustered around 1. In particular, the preconditiond conjugate gradient method applied to the preconditioned system converges superlinearly and the n-by-n Toeplitz system can be solved in $O(n \log n)$ operations. However, we remark that if f has a zero, then the result fails to hold and circulant preconditioned systems can converge at a very slow rate, see the numerical examples in §5.

2.3 Circulant Preconditioners from Kernel Function

Recently, R. Chan and Yeung [11] introduced a method of finding and analyzing circulant preconditioners for Toeplitz systems. Circulant preconditioners are considered as convolution product of the generating function with some kernels $\hat{C}_n$. For example, Strang's and T. Chan's circulant preconditioners are constructed by using the Dirichlet and Fejér kernels, respectively. Recall that the eigenvalues of Strang's preconditioner $S_n[f]$ are given by

$$\lambda_j(S_n[f]) = (\hat{D}_{\lfloor \frac{n}{2} \rfloor} * f)(\frac{2\pi j}{n}), \quad 0 \le j < n,$$

where the convolution of the Dirichlet kernel with f is given by

$$(\hat{D}_{\lfloor \frac{n}{2} \rfloor} * f)(\theta) \equiv \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{D}_{\lfloor \frac{n}{2} \rfloor}(\theta - \phi) f(\phi) d\phi$$

and

$$\hat{D}_k(\theta) = \frac{\sin(k+\frac{1}{2})\theta}{\sin\frac{1}{2}\theta}, \quad k = 1, 2, \ldots.$$

The eigenvalues of T. Chan's preconditioner $T_n[f]$ are given by

$$\lambda_j(T_n[f]) = (\hat{F}_n * f)(\frac{2\pi j}{n}), \quad 0 \le j < n, \tag{2.6}$$

where the Fejér kernels are given by

$$\hat{F}_k(\theta) = \frac{1}{k}\left(\frac{\sin\frac{k}{2}\theta}{\sin\frac{1}{2}\theta}\right)^2, \quad k = 1, 2, \cdots.$$

Similarly, the eigenvalues of R. Chan's preconditioner $R_n[f]$ is given by

$$\lambda_j(R_n[f]) = (\hat{D}_{n-1} * f)(\frac{2\pi j}{n}), \quad 0 \le j < n. \tag{2.7}$$

Most of the known circulant preconditioners, for instance the Huckle's preconditioner and the Ku and Kuo's preconditioner, can be derived easily from this approach. We can also apply the above idea to design other circulant preconditioners from kernels such as the von Hann kernel, Hamming kernel, and Bernstein kernel that are commonly used in function theory [33] and signal processing [20].

We note that the convolution products $\hat{C}_n * f$ of these kernels $\hat{C}_n$ with generating function f are just smooth approximations of the generating function f itself. This means that these circulant preconditioners are designed so that their eigenvalues approximate the value of f at $\frac{2\pi j}{n}$, $0 \le j < n$. Thus if $f(\frac{2\pi j}{n})$ can be computed efficiently, then the circulant preconditioners with eigenvalues given by $f(\frac{2\pi j}{n})$ is certainly a good choice. Its corresponding kernel is just the Dirac delta function δ. In other words, if the kernel $\hat{C}_n$ is close to the Dirac delta function δ, then the constructed circulant matrix will be a good preconditioner.

2.4 Non-circulant Band-Toeplitz Preconditioner

In section 2.2, we note that if the generating function f has a zero, the circulant preconditioned systems do not work well. The following theorem shows that when $f_{\min} = 0$, the condition number of Toeplitz matrix A is not uniformly bounded and A is ill-conditioned. Tyrtyshnikov has proved theoretically [32] that Strang's preconditioner will fail in this case.

Theorem 2.1 (R. Chan (1991) [8]) *Suppose that $f(\theta) - f_{\min}$ has a unique zero of order 2ν at $\theta = \theta_0$. Then for all $n > 0$, we have*

$$\lambda_{\min}(A) \leq d_1 f_{\min} + d_2 n^{-2\nu},$$

and

$$\kappa(A) \geq \frac{d_3 n^{2\nu}}{d_4 + f_{\min} n^{2\nu}},$$

where $\{d_i\}_{i=1}^4$ are some constants independent of n.

R. Chan in [8] resorted to using band Toeplitz matrices as preconditioners instead of finding other possible circulant preconditioners. He has used trigonometric polynomials of the form $(2-2\cos(\theta-\theta_0))^\nu$ of fixed degree to approximate the non-negative generating function f around the zeros θ_0 of f rather than by convolution products of f with some kernels. The power ν is the order of the zero θ_0 and is required to be even number. The linear convergence of the resulting preconditioners has been proved. The band-width of the preconditioners is $2\nu+1$ and its diagonals can be obtained by using Pascal's triangle. It is shown in [8] that the total number of operations per iteration is of order $O(n \log n)$ as ν is independent of n and the overall storage requirement in the preconditioned conjugate gradient method is about $(8+\nu)n$.

The advantage of using this band-Toeplitz preconditioners is that trigonometric polynomials can be chosen to match the zeros of f, so that the preconditioned method still works when f has zeros. However, the drawback of using these band-Toeplitz matrices as preconditioners is that the generating function f should be given explicitly, otherwise we cannot construct the band-Toeplitz preconditioners. In addition, when f is positive, these preconditioned systems converge much slower than those preconditioned systems by circulant preconditioners.

Chapter 3

B-SPLINES

3.1 Introduction

Let $\{u_i\}_{i=1}^m$ be a set of functions defined on the set $I = [a, b]$ and let $t_1, \ldots, t_m$ be points in I such that $t_1 < t_2 < \cdots < t_m$. Define

$$D\begin{pmatrix} t_1, & \ldots, & t_m \\ u_1, & \ldots, & u_m \end{pmatrix} = \det \begin{bmatrix} u_1(t_1) & u_2(t_1) & \cdots & u_m(t_1) \\ u_1(t_2) & u_2(t_2) & \cdots & u_m(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \cdots & u_m(t_m) \end{bmatrix}.$$

Given points $t_1, t_2, \ldots, t_{r+1}$ and a function f, we define its r-th order divided difference over the points t_1, t_2, $\ldots$, t_{r+1} by

$$[t_1, \quad \ldots \quad , t_{r+1}]\ f = \frac{D\begin{pmatrix} t_1, & t_2 & \ldots, & t_r & t_{r+1} \\ 1, & x, & \ldots, & x^{r-1}, & f \end{pmatrix}}{D\begin{pmatrix} t_1, & t_2 & \ldots, & t_r & t_{r+1} \\ 1, & x, & \ldots, & x^{r-1}, & x^r \end{pmatrix}}.$$

In the following, let us introduce the truncated power function:

$$(x-y)_+^j = (x-y)^j (x-y)_+^0, \quad j > 0$$

where

$$(x-y)_+^0 = \begin{cases} 0, & x < y, \\ 1, & x \geq y. \end{cases}$$

B-splines were first introduced by Schoenberg in [29]. In this section, let us begin by introducing the r-th order B-splines as appropriately scaled divided differences of the truncated power function.

Definition 3.1 *Let* $\cdots \leq y_{-1} \leq y_0 \leq y_1 \leq y_2 \leq \cdots$ *be a sequence of real numbers. Given integers* i *and* $m > 0$, *we define*

$$Q_i^m(x) = \begin{cases} (-1)^m[y_i, \ldots, y_{i+m}](x-y)_+^{m-1}, & \text{if} \quad y_i < y_{i+m}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

for all real x. *We call* $Q_i^m(x)$ *the mth order B-spline associated with the knots* y_i, $\ldots$, y_{i+m}.

For m=1, the B-spline associated with $y_i < y_{i+1}$ is particularly simple. It is the piecewise constant function

$$Q_i^1(x) = \begin{cases} \dfrac{1}{y_{i+1} - y_i}, & y_i \leq x < y_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 3.1 (Schumaker [30]) *Let* $m \geq 2$, *and suppose* $y_i < y_{i+m}$. *Then for all* $x \in \mathbf{R}$,

$$Q_i^m(x) = \frac{(x - y_i)Q_i^{m-1}(x) + (y_{i+m} - x)Q_{i+1}^{m-1}(x)}{(y_{i+m} - y_i)}.$$

This provides a recursion relation whereby B-splines of order m can be related to B-splines of order $m-1$.

So far we have said nothing about the size of B-splines. The B-splines $Q_i^m(x)$ introduced can have widely different sizes depending on the location of the knots. For example, in the interval $[y_i, y_{i+1}]$, the B-spline

$$Q_i^1(x) = \frac{1}{y_{i+1} - y_i}$$

can be extremely large or extremely small, depending on the spacing of the y_i's. For computational purposes it is not acceptable to deal with functions that are too small or too large. This suggests that we should introduce some normalization of the B-splines.

Definition 3.2 *Let*

$$N_i^m(x) = (y_{i+m} - y_i)Q_i^m(x)$$

where $Q_i^m(x)$ *is the B-spline defined in Definition (3.1). We call* N_i^m *the normalized B-spline associated with the knots* $y_i, \ldots, y_{i+m}$.

For m=1, the normalized B-spline associated with $y_i < y_{i+1}$ is given by

$$N_i^1(x) = \begin{cases} 1, & y_i \le x < y_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

For m=2, the normalized B-spline associated with $y_i < y_{i+2}$ is given by

$$N_i^2(x) = \begin{cases} \dfrac{x - y_i}{y_{i+1} - y_i}, & y_i \le x < y_{i+1}, \\ \\ \dfrac{y_{i+2} - x}{y_{i+2} - y_{i+1}}, & y_{i+1} \le x < y_{i+2}. \end{cases}$$

In many applications of splines, it suffices to work with equally spaced knots. This leads to simplifications in the theory as well as to substantial savings in computation. It follows that we want to discuss the B-splines with equally spaced knots.

We say that a set of knots $\{\cdots, y_i, y_{i+1}, \cdots\}$ is uniform with spacing h provided that $y_{i+1} - y_i = h$ for all i. For uniformly spaced knots it turns out that any B-spline can be obtained from one basic B-spline by translation and scaling. Let

$$Q^m(x) = \sum_{i=0}^{m} \frac{(-1)^i C_i^m (x-i)_+^{m-1}}{m!}$$

This is the usual B-splines associated with simple knots $0, 1, \cdots, m$. It belongs to $C^{m-2}(-\infty, \infty)$ **(Schumaker [30])**. Associated with Q^m, we also introduce the normalized version

$$N^m(x) = mQ^m(x). \tag{3.2}$$

Theorem 3.2 (Schumaker) [30]) *Suppose* $y_i, \ldots, y_{i+m}$ *are uniformly spaced knots with spacing h. Then*

$$Q_i^m(x) = \frac{1}{h} Q^m\left(\frac{x - y_i}{h}\right)$$

and

$$N_i^m(x) = N^m(\frac{x - y_i}{h})$$

.

For convenient reference, we now give the explicit formulae for the polynomial pieces of $N^m(x)$ for m=2, 3 and 4.

$$N^2(x) = \begin{cases} x, & 0 \le x < 1, \\ (2-x), & 1 \le x < 2. \end{cases}$$

$$N^3(x) = \begin{cases} \frac{x^2}{2}, & 0 \le x < 1, \\ \frac{(-2x^2 + 6x - 3)}{2}, & 1 \le x < 2, \\ \frac{(3-x)^2}{2}, & 2 \le x < 3. \end{cases}$$

$$N^4(x) = \begin{cases} \frac{x^3}{6}, & 0 \le x < 1, \\ \frac{(-3x^3 + 12x^2 - 12x + 4)}{6}, & 1 \le x < 2, \\ N^4(4-x), & 2 \le x < 4. \end{cases}$$

The normalized B-splines N^3 and N^4 are shown in Figure 3.1.

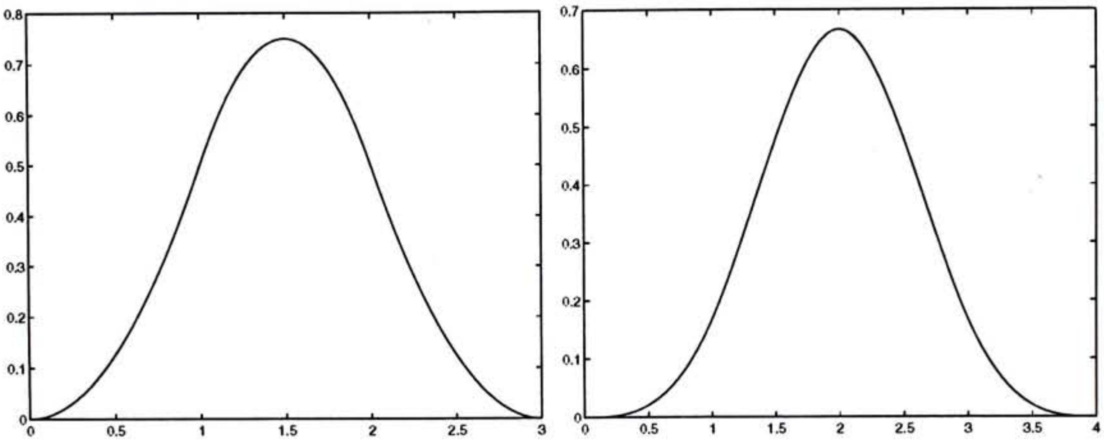

Figure 3.1 (left) The normalized B-spline N^3, (right)The normalized B-spline N^4.

Several of the formulae involving B-splines in the preceding discussion can be simplified in the case of equally spaced knots. For example, the basic recursion formula reads

$$Q^m(x) = \frac{xQ^{m-1}(x) + (m-x)Q^{m-1}(x-1)}{m}$$

or in terms of the normalized B-spline N^m,

$$N^m(x) = xQ^{m-1}(x) + (m-x)Q^{m-1}(x-1).$$

3.2 New Version of B-Splines

Before discussing further, we want to introduce a slightly translated and scaled version of the normalized B-splines N^m defined in (3.2)

$$B^m(x) = \frac{1}{\max(B^m)}[N^{m+1}(x + \frac{m+1}{2})], \quad \forall x \in \mathbf{R} \tag{3.3}$$

where $\max(B^m)$ is the maximum value of $B^m(x)$ for all $x \in \mathbf{R}$. This spline is symmetric about the origin, has support on $[-\frac{m+1}{2}, \frac{m+1}{2}]$ and the value of $B^m(0)$ for all m is scaled to one. For m odd, it has simple knots at the integers, while for even m, the knots are at the midpoints between the integers.

In the following, we give the explicit formulae for the translated and scaled version of the normalized B-splines N^m for m= 0, 1 and 2.

$$B^0(x) = \begin{cases} 1, & -1/2 \le x < 1/2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

$$B^1(x) = \begin{cases} 1+x, & -1 \le x < 0, \\ 1-x, & 0 \le x < 1. \end{cases} \tag{3.5}$$

$$B^2(x) = \begin{cases} \frac{2x^2}{3} + 2x + \frac{3}{2}, & -3/2 \le x < -1/2, \\ \frac{-4x^2}{3} + 1, & -1/2 \le x < 1/2, \\ \frac{2x^2}{3} - 2x + \frac{3}{2}, & 1/2 \le x < 3/2. \end{cases}$$

The corresponding graphs are shown in Figures 3.2 and 3.3.

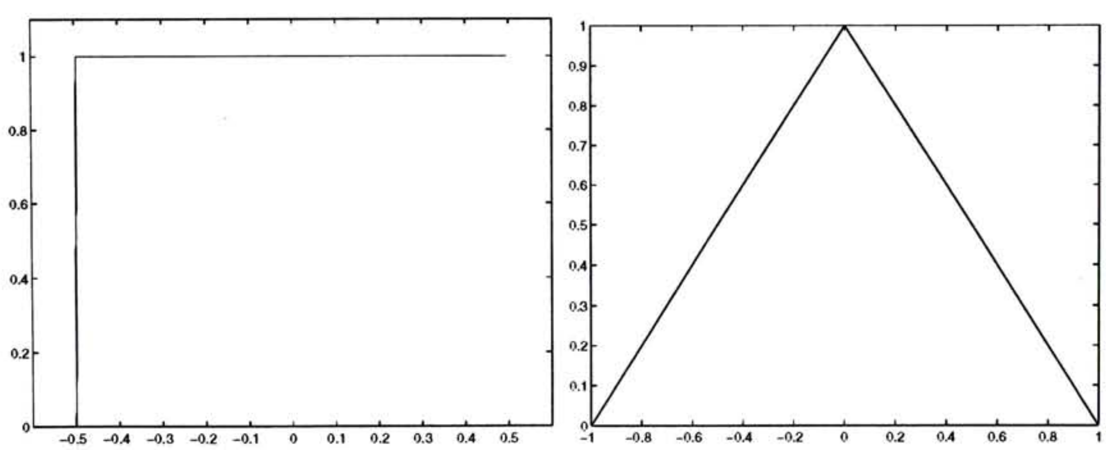

Figure 3.2 The translated and scaled B-spline B^0(left) and B^1(right).

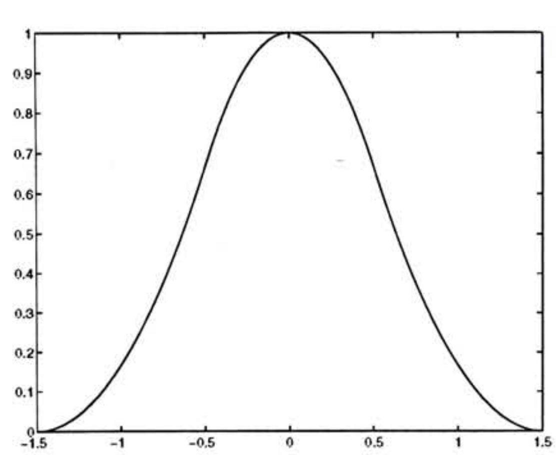

Figure 3.3 The translated and scaled B-spline B^2

In (3.4), we note that the support of $B^0(x)$ is from $-1/2$ to $1/2$. We divide the support into $2n$ equal partitions associated with $2n+1$ knots

$$x_{-n}, x_{-(n-1)}, \ldots, x_{-1}, x_0, x_1, \ldots, x_{n-1}, x_n.$$

By using the formula of $B^0(x)$ in (3.4), we have

$$B^0(x_i) = \begin{cases} 1, & i = 0, \pm 1, \pm 2, \ldots, \pm(n-1), n, \\ 0, & i = n. \end{cases}$$

For the convenience of constructing our preconditioners in the next chapter, we now introduce a new sequence B_n^0 from $B^0(x)$. This sequence is obtained by taking n equally spaced samples of the function $B^0(x)$. More precisely, $B_n^0 = \{b_k^0 = B^0(x_k)$; for $-n \le k \le n$ where $b_n^0 = b_{-n}^0 = 0\}$. In short, $B_n^0 = \{0, 1, 1, \ldots, 1, 1, 0\}$.

In addition, by using b_k^0 for $-n \le k \le n$ as the Fourier coefficients, we can find the Fourier series $\hat{B}_n^0(\theta)$ which containing these $2n+1$ coefficients.

In the following, we let

$$\hat{B}_n^0(\theta) \equiv \sum_{k=-n}^{n} b_k^0 e^{ik\theta}. \tag{3.6}$$

Then, we see that

$$\begin{aligned} \hat{B}_n^0(\theta) &= \sum_{k=-(n-1)}^{n-1} e^{ik\theta} \\ &= \sum_{k=-(n-1)}^{n-1} e^{ik\theta} \\ &= \frac{e^{-i(n-1)\theta}(1 - e^{i(2n-1)\theta})}{1 - e^{i\theta}} \\ &= \frac{e^{-i(n-1)\theta} - e^{in\theta}}{1 - e^{i\theta}} \\ &= \frac{e^{i(n-\frac{1}{2})\theta} - e^{-i(n-\frac{1}{2})\theta}}{e^{i\frac{1}{2}\theta} - e^{-i\frac{1}{2}\theta}} \\ &= \frac{\sin(n-\frac{1}{2})\theta}{\sin(\frac{1}{2}\theta)} \\ &= \hat{D}_{n-1}(\theta), \end{aligned}$$

where $\hat{D}_m(\theta)$ is the well-known Dirichlet kernel. This means that $\hat{B}_n^0(\theta)$ equals to the Dirichlet kernel $\hat{D}_{n-1}(\theta)$. In Figures 3.4 to 3.6, the graphs of $\hat{B}_n^0(\theta)$ for n=16, 32, 64, 128 and 256 are plotted. Notice that the graph of $\hat{B}_n^0(\theta)$ is almost like a Dirac delta function δ as n increases. However, we note that even for n = 256, the function $\hat{B}_n^0(\theta)$ still have significant ripples away from the origin.

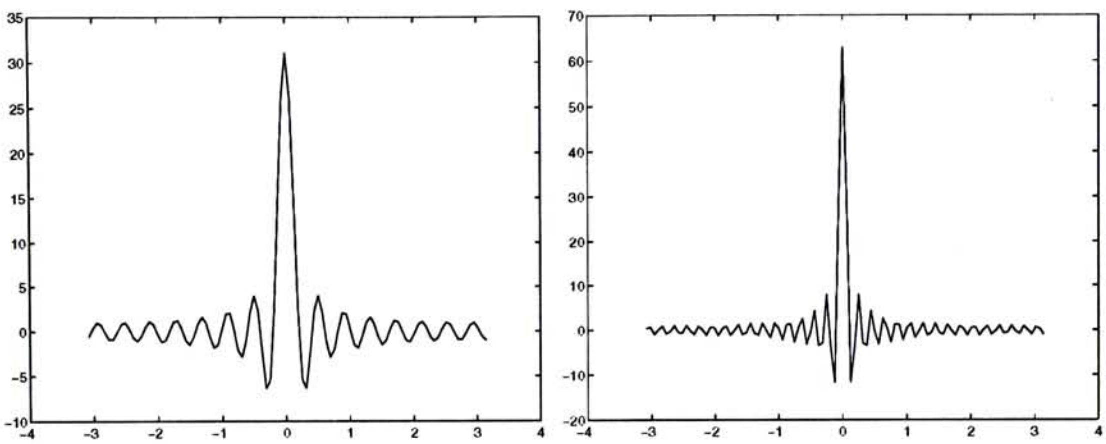

Figure 3.4 The graph of $\hat{B}_n^0(\theta)$ where n=16(left) and n=32(right).

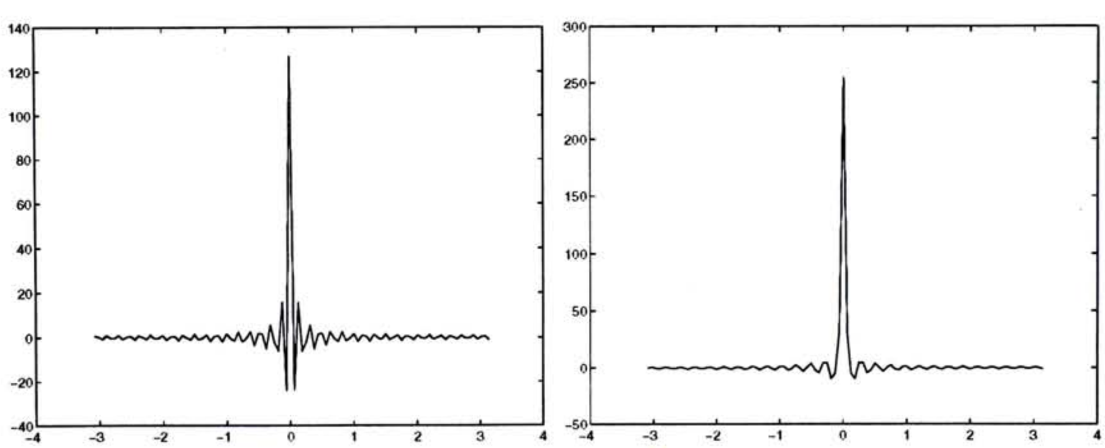

Figure 3.5 The graph of $\hat{B}_n^0(\theta)$ where n=64(left) and n=128(right).

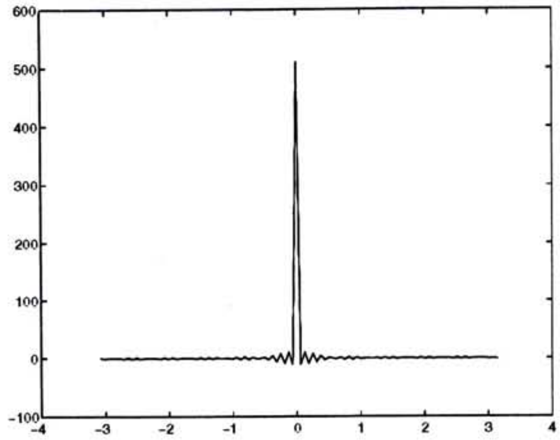

Figure 3.6 The graph of $\hat{B}_n^0(\theta)$ where n=256

Similarly, for $B^1(x)$, the support is from -1 to 1. We divide the support into $2n$ equal partitions associated with the $2n+1$ knots $x_{-n}, \ldots, x_0, \ldots, x_{n-1}, x_n$. By using the formula of $B^1(x)$ in (3.5), we have

$$B^1(x_j) = \begin{cases} \frac{n-|j|}{n}, & |j| \le n-1, \\ 0, & \text{otherwise.} \end{cases}$$

Hence again we take n evenly spaced samples of $B^1(x)$ and construct a new sequence $B_n^1 = \{b_k^1 = B^1(x_k);$ for $-n \le k \le n$ where $b_n^1 = b_{-n}^1 = 0\}$. In short, $B_n^1 = \{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1, \frac{n-1}{n}, \ldots, \frac{1}{n}, 0\}$.

From the sequence B_n^1, we form the Fourier series $\hat{B}_n^1(\theta)$ which will have the Fourier coefficients b_k^1 for $-n < k < n$, i.e.

$$\hat{B}_n^1(\theta) \equiv \sum_{k=-n}^{n} b_k^1 e^{ik\theta}.$$

Then we have

$$\begin{aligned} \hat{B}_n^1(\theta) &= \sum_{k=-(n-1)}^{n-1} b_k^1 e^{ik\theta} \\ &= \sum_{k=-(n-1)}^{n-1} \frac{n-|k|}{n} e^{ik\theta} \end{aligned}$$

$$= \hat{F}_n(\theta),$$

here $\hat{F}_m(\theta)$ is the well-known Fejér kernel. This means that $\hat{B}_n^1(\theta)$ equals to the Fejér kernel $\hat{F}_n(\theta)$. In Figures 3.7 to 3.9, the graphs of $\hat{B}_n^1(\theta)$ for n=16, 32, 64, 128 and 256 are plotted. Like the $\hat{B}_n^0(\theta)$, we can see that the graph of $\hat{B}_n^1(\theta)$ also converges to a Dirac delta function δ as n increases. However we see that the ripples away from the origin are much smaller than that of $\hat{B}_n^0(\theta)$.

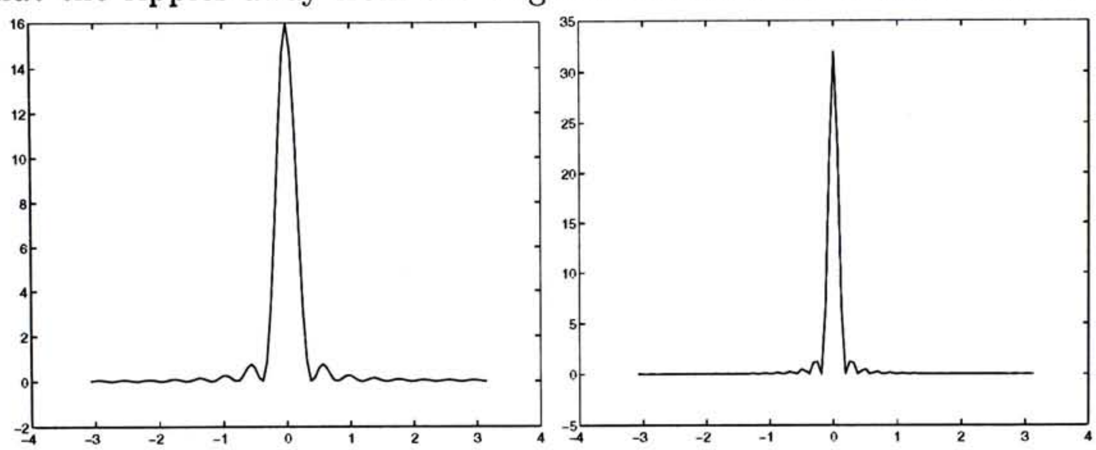

Figure 3.7 The graph of $\hat{B}_n^1(\theta)$ where n=16(left) and n=32(right).

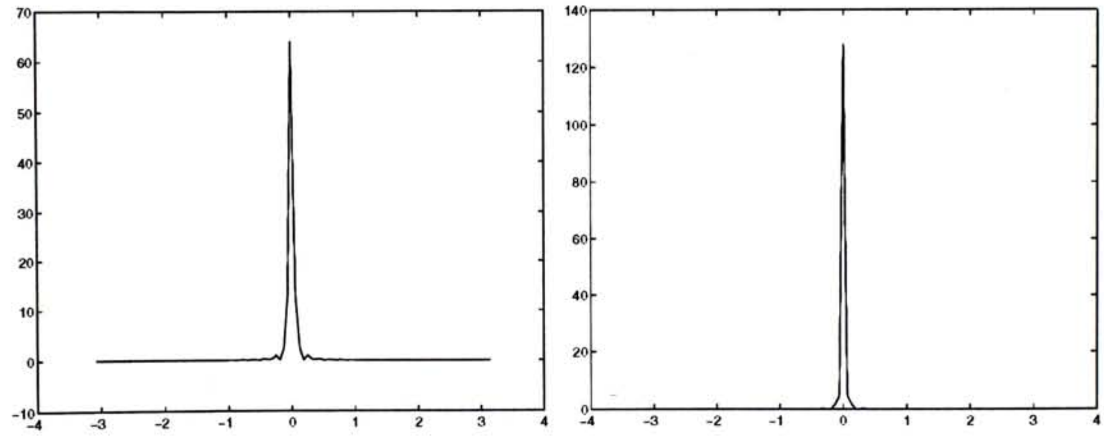

Figure 3.8 The graph of $\hat{B}_n^1(\theta)$ where n=64(left) and n=128(right).

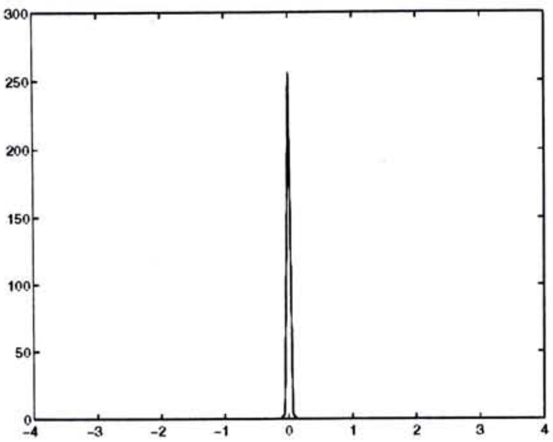

Figure 3.9 The graph of $\hat{B}_n^1(\theta)$ where n=256

Using the same idea presented above, we can construct B_n^l and $\hat{B}_n^l$. The graphs of $\hat{B}_n^l$ for $l = 2$ and 3 are given in Figures 3.10 to 3.15. Notice that these graphs show the same property as that of the $\hat{B}_n^0$ and $\hat{B}_n^1$. This means that when n becomes larger, the graphs of $\hat{B}_n^l$ is closer to the Dirac delta function δ. In addition, we see that the ripples away from the origin, which occured in $\hat{B}_n^0$ and $\hat{B}_n^1$, are nearly disappeared.

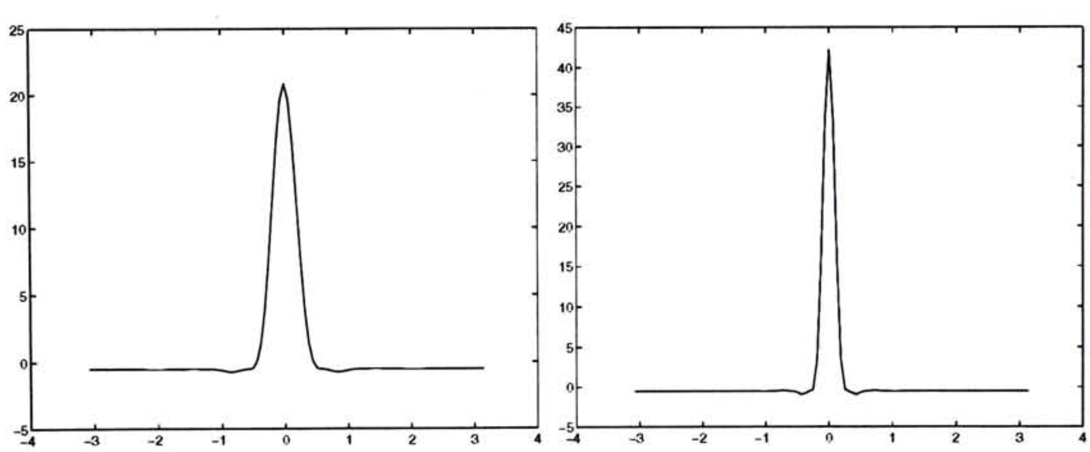

Figure 3.10 The graph of $\hat{B}_n^2(\theta)$ where n=16(left) and 32(right).

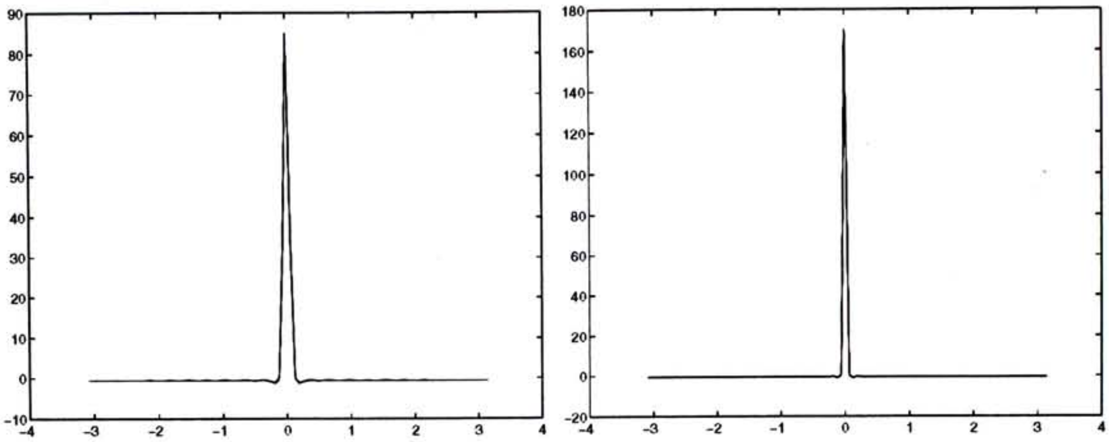

Figure 3.11 The graph of $\hat{B}_n^2(\theta)$ where n=64(left) and 128(right).

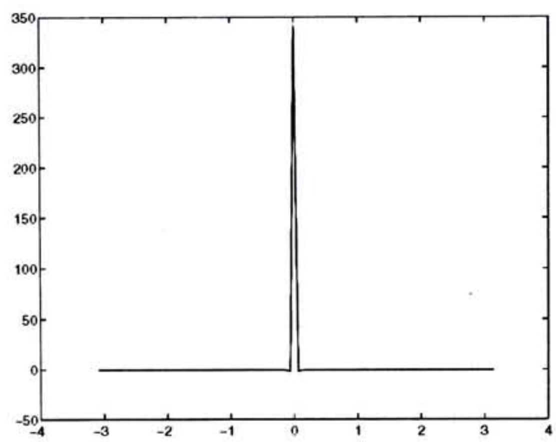

Figure 3.12 The graph of $\hat{B}_n^2(\theta)$ where n=256.

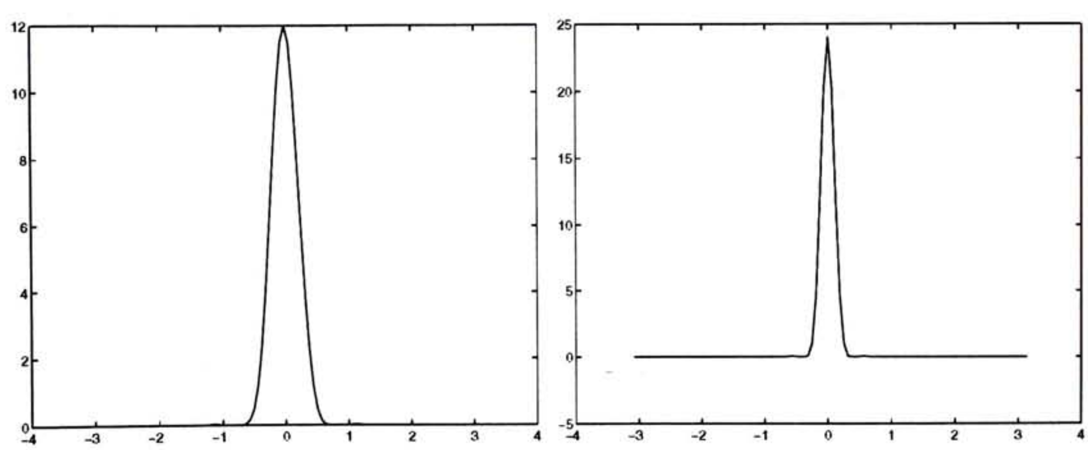

Figure 3.13 The graph of $\hat{B}_n^3(\theta)$ where n=16(left) and 32(right).

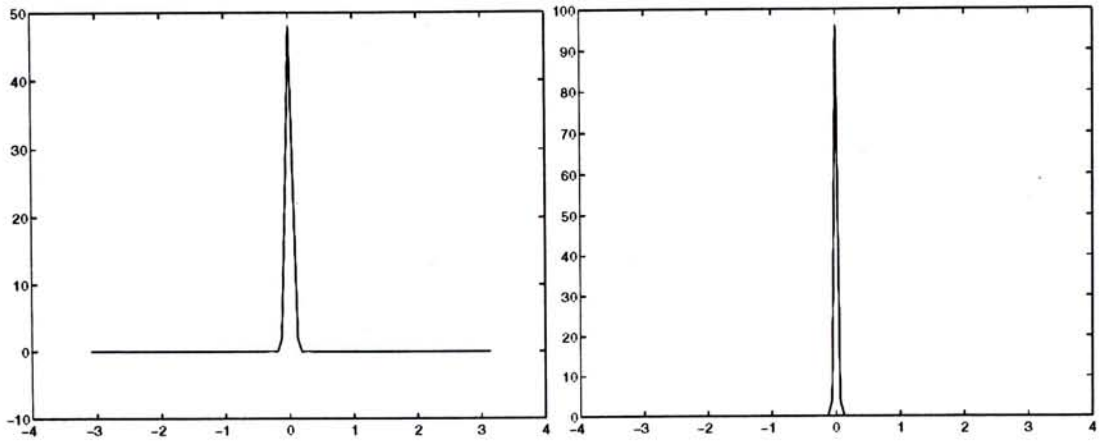

Figure 3.14 The graph of $\hat{B}_n^3(\theta)$ where n=64(left) and 128(right).

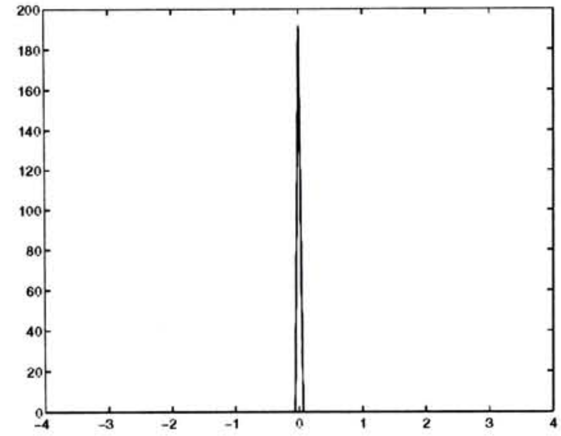

Figure 3.15 The graph of $\hat{B}_n^3(\theta)$ where n=256

Chapter 4

CIRCULANT PRECONDITIONERS CONSTRUCTED FROM B-SPLINES

In this section, we propose a numerical procedure to construct a family of circulant preconditioners for Toeplitz systems that are based on the translated and scaled B-splines introduced in the last section. According to the discussion in Chapter 2, the eigenvalues of the circulant preconditioner C_n can be derived from the convolution product of the generating function f and a kernel $\hat{C}_n$, such as $f * \hat{C}_n$. Using this idea, the first column of the circulant preconditioner can be obtained by using (2.3). For instance, we find that the eigenvalues of R. Chan's circulant preconditioner $R_n[f]$ are given by convolution product of the generating function f and the Dirichlet kernels in (2.7). It can be expressed as follows:

$$
\begin{aligned}
\lambda_j(R_n[f]) &= (f * \hat{D}_{n-1})(\frac{2\pi j}{n}), \quad 0 \le j < n \\
&= s_{n-1}[f](\frac{2\pi j}{n}) \\
&= a_0 + \sum_{k=1}^{n-1} a_k e^{2\pi ijk/n} + \sum_{k=1}^{n-1} \bar{a}_k \bar{e}^{2\pi ijk/n} \\
&= a_0 + \sum_{k=1}^{n-1} \{a_k + \bar{a}_{n-k}\} e^{2\pi ijk/n} \\
&= a_0 + \sum_{k=1}^{n-1} \{a_k + a_{-(n-k)}\} e^{2\pi ijk/n} \\
&= a_0 + \sum_{k=1}^{n-1} \{b_k^0 a_k + b_{-(n-k)}^0 \bar{a}_{n-k}\} e^{2\pi ijk/n}.
\end{aligned}
$$

As stated in (3.6), the Fourier series $\hat{B}_n^0$ contains the coefficients b_k^0, where $b_k^0 = 1$ for $|k| < n$. It is clear that we are able to construct the $(k+1)$-entry of the first column of the R. Chan's circulant preconditioner by adding the terms $b_k^0 a_k$

and $b^0_{-(n-k)}a_{-(n-k)}$ where b^0_k and $b^0_{-(n-k)}$ are the k-th and $-(n-k)$-th coefficients of $\hat{B}^0_n$.

That is to say, the first column of R. Chan's circulant preconditioner can be constructed as follows.

$$(R_n[f])_{k0} = \begin{cases} a_0 & k = 0, \\ \\ b^0_k a_k + b^0_{-(n-k)}\bar{a}_{n-k} & 0 < k < n, \end{cases}$$

Here we have related the entries of the first column of the R. Chan's circulant preconditioner to the Fourier coefficients of $\hat{B}^0_n$. More precisely, we can construct the eigenvalues of the R. Chan's circulant preconditioner by the convolution product of the generating function f and the kernel $\hat{B}^0_n$.

Similarly, the eigenvalues of T. Chan's circulant preconditioner $T_n[f]$, as in (2.6), are given by the convolution product of the generating function f and the Fejér kernel. The way of forming the eigenvalues of the $T_n[f]$ is as follows:

$$\begin{aligned}
\lambda_j(T_n[f]) &= (f * \hat{F}_n)(\frac{2\pi j}{n}), \quad 0 \le j < n \\
&= \frac{1}{n}\sum_{k=-(n-1)}^{n-1} (n - |k|)a_k e^{2\pi ijk/n} \\
&= \frac{1}{n}\sum_{k=1}^{n-1}(n-k)\{a_k e^{2\pi ijk/n} + \bar{a}_k \bar{e}^{2\pi ijk/n}\} + a_0 \\
&= \sum_{k=0}^{n-1}\frac{n-k}{n}a_k e^{2\pi ijk/n} + \sum_{k=1}^{n-1}\frac{n-k}{n}\bar{a}_k \bar{e}^{2\pi ijk/n} \\
&= \sum_{k=0}^{n-1}\frac{n-k}{n}a_k e^{2\pi ijk/n} + \sum_{k=1}^{n-1}\frac{k}{n}\bar{a}_{n-k} e^{2\pi ijk/n} \\
&= \sum_{k=0}^{n-1} b^1_k a_k e^{2\pi ijk/n} + \sum_{k=1}^{n-1} b^1_{-(n-k)}\bar{a}_{n-k} e^{2\pi ijk/n} \\
&= a_0 + \sum_{k=1}^{n-1}(b^1_k a_k + b^1_{-(n-k)}\bar{a}_{n-k})e^{2\pi ijk/n}.
\end{aligned}$$

Here, we note that the first column of T. Chan circulant preconditioner can be written as the sum of $b^1_k a_k$ and $b^1_{-(n-k)}\bar{a}_{n-k}$ where b^1_k and $b^1_{-(n-k)}$ are the k-th and $-(n-k)$-th coefficients of $\hat{B}^1_n$. Thus the first column of T. Chan's circulant preconditioner can also be related to the the Fourier coefficients of $\hat{B}_n^1$. We can use the following numerical procedure to construct the first column of the T. Chan's preconditioner.

$$(T_n[f])_{k0} = b_k^1 a_k + b_{-(n-k)}^1 \bar{a}_{n-k}, \quad 0 \le k < n.$$

As remarked in the last paragraph of Section 2.3 in Chapter 2, if we want to construct a good circulant preconditioner for the Toeplitz systems, the convolution product of the kernel and the generating function f should be a good approximation of f. From the previous section, we note that $\hat{B}_n^0$ and $\hat{B}_n^1$ are close to the delta function δ as n increases. Since $\hat{B}_n^2$ is closer to the delta function than $\hat{B}_n^0$ and $\hat{B}_n^1$, here we propose a numerical procedure to construct a new circulant preconditioner from B_n^2. Using the same technique as mentioned above, the first column of the circulant preconditioner from B_n^2 is obtained by

$$(B_n^2[f])_{k0} = b_k^2 a_k + b_{-(n-k)}^2 \bar{a}_{n-k}, \quad 0 \le k < n.$$

Hence the eigenvalues of circulant preconditioner constructed from $B_n^2[f]$ are given by

$$\lambda_j(B_n^2[f]) = \sum_{k=0}^{n-1} b_k^2 a_k e^{2\pi ijk/n} + \sum_{k=1}^{n-1} b_{-(n-k)}^2 \bar{a}_{n-k} e^{2\pi ijk/n} = (\hat{B}_n^2 * f)(\frac{2\pi j}{n}), \quad 0 \le j < n.$$

In the next chapter, we will implement our preconditioners to different types of generating function f. The results show that our preconditioners work better than the well-known circulant preconditioners such as Strang's, T. Chan's and R. Chan's preconditioners and the non-circulant band-toeplitz preconditioner.

We now consider the cost of solving Toeplitz systems $A_n[f]x = b$ by using the preconditioned conjugate gradient method with the circulant matrice B_n^l constructed from B-splines as our preconditioners. It is known that the cost per iteration in the preconditioned conjugate gradient method is about $5n$ multiplications and $5n$ additions plus the cost of computing the matrix-vector multiplications of $A_n y$ and $(B_n^l)^{-1} d$ for some vectors y and d, see Golub and van Loan [17]. The matrix-vector multiplication $A_n y$ is able to be computed by the Fast Fourier

Transform by first embedding A_n into $2n$-by-$2n$ circulant matrix, see Strang [31]. The cost is about $2n\log(2n)+2n$ operations. As mentioned in Chapter 2, circulant systems can be solved efficiently by the Fast Fourier Transform. Thus, the vector $(B_n^l)^{-1}d$ can be done by $O(n\log n)$ operations. Therefore, the cost per iteration of the preconditioned conjugate gradient method is of $O(n\log n)$. The cost of constructing the circulant matrix B_n^l is about $2n$ operations. It follows that the total number of operations per iteration is of $O(n\log n)$ operations. As for the storage of preconditioned conjugate gradient method, we need to store five n-vectors only.

Chapter 5

NUMERICAL RESULTS AND CONCLUDING REMARKS

In this chapter, we compare our preconditioners with Strang's circulant preconditioner, R. Chan's circulant preconditioner and T. Chan's circulant preconditioner. We test their performances on thirteen functions defined on $[-\pi, \pi]$. They are classified as four types. First one is the simple positive generating function:

a) $f = x^4 + 1$.

Second class is the positive generating functions with jumps:

a) $f = (x+\pi)^2 + 1$,

b) $f(x) = \begin{cases} \frac{0.9x}{\pi} + 10, & -\pi < x \le 0, \\ \frac{0.9x}{\pi} + 0.1, & 0 < x \le \pi. \end{cases}$

Third is the non-negative generating functions having no jumps:

a) $f = x^2$,

b) $f = |x|^3$,

c) $f = x^4$,

d) $f = 1 - \cos\theta$,

e) $f = |x^2(x^2-1)|$,

f) $f = \pi^2(x^2) - x^4$.

The last is the non-negative generating functions having jumps:

a) $f(x) = \begin{cases} x^2, & |x| \le \pi/2, \\ 1, & \pi/2 < |x| < \pi, \end{cases}$

b) $f = x(x+1)$,

c) $f(x) = \begin{cases} x^2, & x \le 0, \\ x, & x > 0, \end{cases}$

d) $f = (x+\pi)^2$.

The Toeplitz matrices A_n are formed by evaluating the Fourier coefficients of the test functions. In our tests, the vector of all ones is the right-hand side vector, the zero vector is the initial guess, and the stopping criterion is $\frac{\|r_q\|_2}{\|r_0\|_2} \leq 10^{-7}$, where r_q is the residual vector after q iterations. All computations are done by matlab on an IBM 43P workstation.

Tables 1-13 show the numbers of iterations required for convergence with different choices of preconditioners. In the tables, I denotes no preconditioner was used, B_n^0, B_n^1, B_n^2, B_n^3, B_n^4 and B_n^5 are respectively our preconditioners constructed from B-splines B_n^m for m=0, 1, 2, 3, 4 and 5. Actually B_n^0 and B_n^1 are respectively the same as the R. Chan's and T. Chan's circulant preconditioners.

From the numerical results, we see that in all tests, our preconditioners perform better than many other well-known preconditioners, such as Strang's, R. Chan's, T. Chan's circulant preconditioners when solving the Toeplitz systems with the non-negative generating function with or without jumps. When comparing the performance of our preconditioners and the band-Toeplitz preconditioner, we observe that the number of iterations required for convergence using our preconditioners is less than that using the non-circulant band-Toeplitz preconditioner in Table 4. However, in Table 6, we see that our preconditioners work pretty well for small n only. Though the non-circulant band-Toeplitz preconditioners can work for the non-negative generating functions, the generating function f has to be given explicitly. That is the reason why we do not use these band-Toeplitz preconditioners in practice. The beauty of our preconditioner is that it can solve the Toeplitz systems without the knowledge of the underlying generating function. Moreover, among all the preconditioners constructed from B-splines, the B_n^2 works the best almost in every case.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	8	20	37	56	67	70	71
Strang's	8	8	6	5	5	5	5
B_n^0(R.Chan)	6	5	5	5	5	5	5
B_n^1(T.Chan)	8	7	7	6	6	6	5
B_n^2	6	5	5	5	5	5	5
B_n^3	7	6	5	5	5	5	5
B_n^4	6	6	5	5	5	5	5
B_n^5	7	6	5	5	5	5	5

Table 1 . Number of Iterations for $f = x^4 + 1$.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	16	31	38	40	39	38	38
Strang's	10	14	17	19	19	19	19
B_n^0(R.Chan)	10	11	12	11	12	12	12
B_n^1(T.Chan)	10	11	11	12	11	12	12
B_n^2	7	8	9	9	9	10	10
B_n^3	8	8	9	9	10	10	11
B_n^4	8	8	9	9	10	10	11
B_n^5	8	9	9	10	10	10	11

Table 2. Number of Iterations for $f = (x + \pi)^2 + 1$.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	16	30	40	44	48	50	52
Strang's	16	20	23	27	31	39	40
B_n^0(R.Chan)	11	13	15	15	17	17	19
B_n^1(T.Chan)	11	13	15	15	17	17	19
B_n^2	9	12	13	14	14	14	15
B_n^3	10	11	13	14	15	15	16
B_n^4	10	11	13	14	15	16	17
B_n^5	10	12	14	15	15	16	17

Table 3. Number of Iterations for

$$f(x) = \begin{cases} \frac{0.9x}{\pi} + 10, & -\pi < x \le 0, \\ \\ \frac{0.9x}{\pi} + 0.1, & 0 < x \le \pi. \end{cases}$$

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	8	17	38	83	178	374	770
Strang's	7	7	7	7	8	8	8
B_n^0(R.Chan)	5	7	7	7	7	7	7
B_n^1(T.Chan)	8	10	12	14	18	22	28
B_n^2	6	6	8	8	8	8	8
B_n^3	7	7	8	8	8	9	9
B_n^4	7	7	8	8	9	9	9
B_n^5	7	7	8	9	9	9	9
Band-Toeplitz	8	10	11	12	12	12	12

Table 4. Number of Iterations for $f = x^2$.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	8	22	61	179	621	> 1000	> 1000
Strang's	8	10	13	16	20	39	75
B_n^0(R.Chan)	8	10	10	13	20	27	42
B_n^1(T.Chan)	8	13	17	25	37	101	198
B_n^2	8	9	10	10	13	14	15
B_n^3	8	10	10	11	13	15	16
B_n^4	8	10	11	11	13	15	16
B_n^5	9	10	10	11	14	15	16

Table 5. Number of Iterations for $f = |x|^3$.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	9	31	113	544	>1000	>1000	>1000
Strang's	8	14	21	36	121	406	>1000
B_n^0(R.Chan)	9	12	18	32	79	657	>1000
B_n^1(T.Chan)	9	16	26	65	177	484	>1000
B_n^2	9	12	13	15	22	30	49
B_n^3	9	12	15	18	23	39	68
B_n^4	9	12	15	17	21	31	48
B_n^5	9	12	15	17	22	30	55
Band-Toeplitz	8	15	20	24	27	29	30

Table 6. Number of Iterations for $f = x^4$.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	8	16	32	32	64	256	512
Strang's	*	*	*	*	*	*	*
B_n^0(R.Chan)	*	*	*	*	*	*	*
B_n^1(T.Chan)	7	8	10	13	15	19	25
B_n^2	6	6	6	7	7	7	7
B_n^3	6	6	6	8	8	8	8
B_n^4	6	6	7	8	8	8	8
B_n^5	6	7	7	8	8	8	8

Table 7. Number of Iterations for $f = 1 - \cos\theta$.

$*$ indicates 'Not Applicable' due to the singularities of the Strang's and R. Chan's circulant preconditioners.

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	10	30	90	252	600	>1000	>1000
Strang's	8	15	21	23	25	22	31
B_n^0(R.Chan)	9	15	17	15	17	22	19
B_n^1(T.Chan)	9	16	22	27	32	40	58
B_n^2	9	10	10	12	14	15	15
B_n^3	9	11	11	12	14	16	17
B_n^4	9	11	11	13	14	16	16
B_n^5	9	11	11	13	14	15	16

Table 8. Number of Iterations for $f = |x^2(x^2 - 1)|$

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	8	16	32	62	118	225	436
Strang's	7	7	9	9	9	10	11
B_n^0(R.Chan)	6	6	8	8	8	8	11
B_n^1(T.Chan)	7	9	10	13	15	20	24
B_n^2	7	7	8	8	9	9	10
B_n^3	7	7	9	9	9	9	12
B_n^4	7	8	9	9	9	9	12
B_n^5	8	8	9	9	9	10	12

Table 9. Number of Iterations for $f = \pi^2(x^2) - x^4$

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	7	11	20	41	86	178	372
Strang's	7	9	11	14	18	22	26
B_n^0(R.Chan)	7	9	11	12	15	21	25
B_n^1(T.Chan)	7	9	11	13	15	19	23
B_n^2	7	8	9	10	10	10	12
B_n^3	7	8	10	10	10	11	12
B_n^4	7	8	10	10	11	11	12
B_n^5	7	8	10	10	11	11	12

Table 10. Number of Iterations for

$$f(x) = \begin{cases} x^2, & |x| \leq \pi/2, \\ 1, & \pi/2 < |x| < \pi. \end{cases}$$

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	8	17	38	83	178	374	770
Strang's	7	7	7	7	8	8	8
B_n^0(R.Chan)	5	7	7	7	7	7	7
B_n^1(T.Chan)	8	10	12	14	18	22	28
B_n^2	6	6	8	8	8	8	8
B_n^3	7	7	8	8	8	9	9
B_n^4	7	7	8	8	9	9	9
B_n^5	7	7	8	9	9	9	9

Table 11. Number of Iterations for $f = x(x+1)$

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	16	36	71	134	257	486	929
Strang's	11	11	12	13	14	15	18
B_n^0(R.Chan)	10	11	11	12	13	15	18
B_n^1(T.Chan)	10	12	14	16	19	23	30
B_n^2	8	9	11	12	13	15	17
B_n^3	9	10	11	12	13	15	16
B_n^4	9	10	12	12	13	15	16
B_n^5	9	10	12	13	14	16	17

Table 12. Number of Iterations for

$$f(x) = \begin{cases} x^2, & x \le 0, \\ x, & x > 0. \end{cases}$$

Preconditioner Used	n						
	16	32	64	128	256	512	1024
I	16	34	76	167	352	678	>1000
Strang's	11	16	26	42	85	203	755
B_n^0(R.Chan)	11	15	19	28	43	67	113
B_n^1(T.Chan)	11	15	19	27	41	66	111
B_n^2	8	10	12	14	19	30	41
B_n^3	9	10	12	12	20	25	34
B_n^4	9	9	12	14	18	24	34
B_n^5	9	10	12	14	18	21	27

Table 13. Number of Iterations for $f = (x+\pi)^2$.

Chapter 6

APPLICATIONS TO SIGNAL PROCESSING

6.1 Introduction

In this chapter, we consider the least squares problem

$$\min_x \|b - Ax\|_2, \tag{6.1}$$

where A is a rectangular m-by-n Toeplitz matrix. Toeplitz least squares problems arise in a variety of applications, especially in signal and image processing, see [2] and [23]. In general the conjugate gradient method is always applied to the Hermitian positive definite systems. Though A is a rectangular m-by-n matrix, the conjugate gradient method can still be applied. Instead of applying the conjugate gradient method to the least squares problem, we use it to the normal equations in factored form,

$$A^*(b - Ax) = 0.$$

This can be solved by conjugate gradient method without explicitly forming the matrix A^*A, see Björck [5]. As mentioned in Chapter 1, to speed up the convergence of the conjugate gradient method, we can precondition the equation. Firstly, we transform (6.1) with a preconditioner, for instance C, and then we can use the conjugate gradient method to solve

$$\min \|b - AC^{-1}y\|_2,$$

and then set $x = C^{-1}y$. The preconditioner considered here is given by an n-by-n circulant matrix C, where C^*C is then a circulant matrix approximates A^*A.

For the purpose of constructing a preconditioner, we will extend the Toeplitz structure of the matrix A in (6.1) and, if necessary, padding zeros to the bottom left hand side. Without loss of generality, we suppose that $m = kn$ for some positive integer k. This padding is only for convenience in constructing the preconditioner and does not alter the original least squares problem. In the material to follow, we consider the case where k is a constant independent of n. More precisely, we consider kn-by-n matrices A of the form

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_k \end{bmatrix}, \tag{6.2}$$

where each square block A_j is a Toeplitz matrix. Notice that if A itself is a rectangular Toeplitz matrix, then each block A_j is necessarily Toeplitz.

Following [9, 27], for each block A_j, we can construct the circulant approximation C_j. Hence we define our preconditioner as a square circulant matrix C such that

$$C^*C = \sum_{j=1}^{k} C_j^* C_j.$$

Notice that each C_j is an n-by-n circulant matrix. Hence, they can all be diagonalized by the n-by-n discrete Fourier matrix F, i.e., $C_j = F^*\Lambda_j F$ where Λ_j is diagonal of the circulant matrix C_j. Thus, the spectrum of C_j, $j = 1, \cdots, k$, can be computed in $O(n \log n)$ operations by using FFT and we have

$$C^*C = F^* \sum_{j=1}^{k} (\Lambda_j^* \Lambda_j) F.$$

Clearly C^*C is a circulant matrix and its spectrum can be computed in $O(kn \log n)$ operations. The preconditioner is then given by

$$C = F^* (\sum_{j=1}^{k} \Lambda_j^* \Lambda_j)^{\frac{1}{2}} F.$$

R. Chan, Nagy, and Plemmons in [9] showed that total cost of operation in the preconditioned conjugate gradient method to the least squares problem per iteration is of order $O(m \log n)$.

6.2 Preconditioned Regularized Least Squares

In this section the least squares problems (6.1) with the ill-conditioned rectangular matrix A are considered. Such systems occur in many applications, such as signal and image restoration, see [2, 23]. The ill-conditioned nature of A arises from discretization of ill-posed problems in partial differential and integral equations. For example, the problem of estimating an original image from a blurred and noisy observed image is an important case of an *inverse problem*, and was first studied by Hadamard [19] in the inversion of certain integral equations. Since A is ill-conditioned, thus it will lead to extreme instability with respect to perturbations in b when solving $Ax = b$. The method of *regularization* can be used to achieve stability for these problems [5]. Stability is attained by introducing a stabilizing operator (called a regularization operator) which restricts the set of admissible solutions. Since this causes the regularized solution to be biased, a scalar (called a regularization parameter) is introduced to control the degree of bias. More specifically, the regularized solution is computed as

$$\min \left\| \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ \mu L \end{bmatrix} x(\mu) \right\|_2, \tag{6.3}$$

where μ is the regularization parameter and the $p \times n$ matrix L is the regularization operator.

Here the regularization operator L smooth the solution x to a certain degree. Choosing L as a kth difference operator matrix forces the solution to have a small kth derivative. The regularization parameter μ controls the degree of smoothness (*i.e.*, degree of bias) of the solution, and is usually small. Choosing the regularization parameter μ is not a trivial problem, we need to solve (6.3) for several values of μ [15] to determine the best one. Recent analytical methods for choosing an optimal parameter μ are discussed by Reeves and Mersereau [28].

Based on the discussion above, the regularization operator L is usually chosen to be the identity matrix or some discretization of a differentiation operator [15]. Thus L is typically a Toeplitz matrix. Hence, if A has the Toeplitz block form

(6.2), then the matrix

$$\tilde{A} = \begin{bmatrix} A \\ \mu L \end{bmatrix}$$

retains this structure, with the addition of one block (or two blocks if L is a difference operator with more rows than columns). Since $\tilde{A}$ has the block structure (6.2), we can form the circulant preconditioner C for $\tilde{A}$ and use the PCG algorithm for least squares problems to solve (6.3).

Notice that if L is chosen to be the identity matrix, then the circulant preconditioner for $\tilde{A}$ can be constructed by simply adding μ to each of the eigenvalues of the circulant preconditioner for A. In addition, the last block in $\tilde{A}$ (*i.e.*, μI) has singular values μ.

6.3 Numerical Example

Example : Here we consider an application to 1–dimensional signal reconstruction computations. In this example we construct a 100×100 Toeplitz matrix A, whose i, j entry is given by

$$a_{ij} = \begin{cases} 0 & \text{if } |i-j| > 8, \\ \frac{4}{51} g(0.15, x_i - x_j) & \text{otherwise,} \end{cases} \tag{6.4}$$

where

$$x_i = \frac{4i}{51}, \quad i = 1, 2, \ldots, 100,$$

and

$$g(\sigma, \gamma) = \frac{1}{2\sqrt{\pi}\sigma} \exp(-\frac{\gamma^2}{4\sigma^2}).$$

Matrices of this form occur in many signal restoration contexts as a "prototype problem" and are used to model certain degradations in a recorded signal [15, 23]. Due to the bandedness of A its generating function is in the Wiener class. The condition number of T is approximately 2.4×10^6.

Because of the ill–conditioning of A, the system $Ax = b$ will be very sensitive to any perturbations in b. To achieve stability we regularize the problem using the identity matrix as the regularization operator. Eldén [15] uses this approach to solve a linear system by direct methods with the same data matrix A defined in (6.4). To test our preconditioner we will fix $\mu = 0.01$, where μ is chosen based on some tests made by Eldén.

Let

$$\hat{A} = \begin{bmatrix} A \\ \mu I \end{bmatrix} \quad \text{and} \quad \hat{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Then $\hat{A}$ is simply a block Toeplitz matrix. Thus we can apply our preconditioners B_n^0 , B_n^1 and B_n^2 and the PCG algorithm to solve (6.3). The convergence results for solving $Ax = b$ and $\hat{A}x = \hat{b}$ with no preconditioner and $\hat{A}x = \hat{b}$ using B_n^0, B_n^1 and B_n^2 as preconditioners are shown in Table 14. The singular values of A and $\hat{A}(B_n^m)^{-1}$, for $m = 0$, 1 and 2 and the convergence history for solving $Ax = b$ and $\hat{A}x = \hat{b}$ using our preconditioners B_n^m , for $m = 0$, 1 and 2, are shown in Figures 6.1 and 6.2. These results indicate that the PCG algorithm with our preconditioner B_n^m may be an effective method for solving this regularized least squares problem.

n	$Ax = b$	$\hat{A}x = \hat{b}$	$\hat{A}(B_n^0)^{-1}x = \hat{b}$	$\hat{A}(B_n^1)^{-1}x = \hat{b}$	$\hat{A}(B_n^2)^{-1}x = \hat{b}$
100	> 100	54	8	13	8

Table 14. Numbers of iterations for convergence in Example .

This example illustrates the applicability of the circulant PCG method to regularized least squares problems. Recall that B_n^1 is the same as the T. Chan's circulant preconditioner, in Figures 6.1 and 6.2, we see that our preconditioners such as B_n^0 and B_n^2 perform better than B_n^1. The same as the previous numerical examples in §5, the performance of B_n^2 is the best.

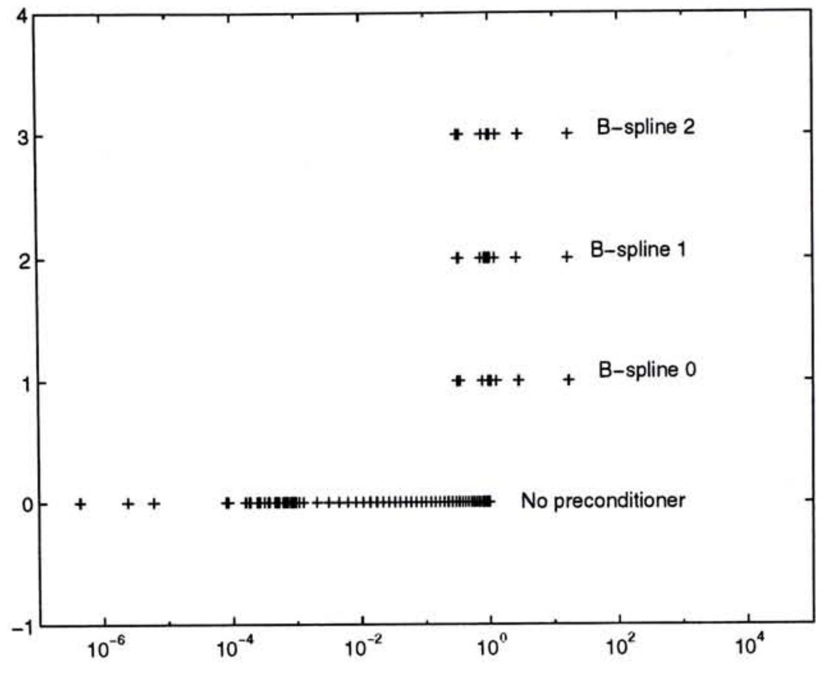

Figure 6.1: Singular values of A and $\hat{A}(B_n^m)^{-1}$ for $m = 0$, 1 and 2.

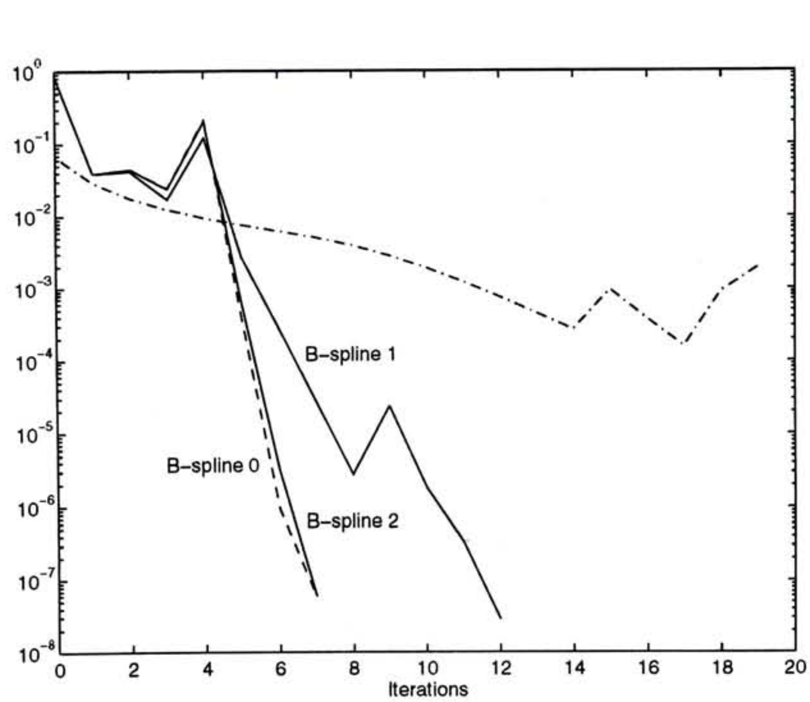

Figure 6.2: Convergence history for A and $\hat{A}(B_n^m)^{-1}$ for $m = 0$, 1 and 2.

References

[1] G. Ammar and W. Gragg, *Superfast Solution of Real Positive Definite Toeplitz Systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[2] H. Andrews and B. Hunt, *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[3] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems, Theory and Computation*, Academic Press, Orlando, FL, 1984.

[4] R. Bitmead and B. Anderson, *Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations*, Linear Algebra Appl., 34 (1980), pp. 103–116.

[5] Å. Björck, *Least Squares Methods*, in Handbook of Numerical Analysis, Vol. 1, P. Ciarlet and J. Lions, eds., Elsevier, Amsterdam, 1989.

[6] R. Brent, F. Gustavson, and D. Yun, *Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximants*, J. Algo., 1 (1980), pp. 259–295.

[7] J. Bunch, *Stability of Methods for Solving Toeplitz Systems of Equations*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 349–364.

[8] R. Chan, *Toeplitz Preconditioners for Toeplitz Systems with Nonnegative Generating Functions*, IMA J. Numer. Anal., 11 (1991), pp. 333–345.

[9] R. Chan, J. Nagy, and R. Plemmons, *Circulant Preconditioned Toeplitz Least Squares Iterations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 80–97.

[10] R. Chan and M. Ng, *Conjugate Gradient Methods for Toeplitz Systems*, SIAM Review, 38, No. 3 (1996), pp. 427-482.

[11] R. Chan and M. Yeung, *Circulant Preconditioners Constructed from Kernels*, SIAM J. Numer. Anal., 29 (1992), pp. 1093–1103.

[12] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 766–771.

[13] C. Chui and A. Chan, *Application of Approximation Theory Methods to Recursive Digital Filter Design*, IEEE Trans. Acoust., Speech, Signal Process., 30 (1982), pp. 18–24.

[14] P. Davis, *Circulant Matrices*, John Wiley & Sons, New York, 1979.

[15] L. Eldén, *An Algorithm for the Regularization of Ill-conditioned, Banded Least Squares Problems*, SIAM J. Sci. Statist. Computing, V5 (1984), pp. 237–254.

[16] I. Gohberg and I. Fel'dman, *Convolution Equations and Projection Methods for Their Solution*, Transl. Math. Monographs, Vol. 41, Amer. Math. Soc., Providence, RI, 1974.

[17] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[18] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*, 2nd ed., Chelsea Publishing, New York, 1984.

[19] J. Hadamard, *Lectures on the Cauchy Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, CT, 1923.

[20] R. Hamming, *Digital Filters*, 3rd ed., Prentice Hall, Englewood Cliffs, NJ, 1989.

[21] S. Haykin, *Adaptive Filter Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

[22] T. Huckle, *Circulant and Skew Circulant Matrices for Solving Toeplitz Matrix Problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 767–777.

[23] A. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[24] R. King et al., *Digital Filtering in One and Two Dimensions: Design and Applications*, Plenum Press, New York, 1989.

[25] T. Ku and C. Kuo, *Design and Analysis of Toeplitz Preconditioners*, IEEE Trans. Signal Process., 40 (1992), pp. 129–141.

[26] N. Levinson. *The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction*, J. Math. and Phys., 25 (1946), pp. 261–278.

[27] J. Nagy and R. Plemmons, *Some Fast Toeplitz Least Squares Algorithms*, in Proc. SPIE Conf. on Advanced Signal Processing Algorithms, Architectures, and Implementations II, Vol. 1566, (1991), pp. 35–46.

[28] S. Reeves and R. Mersereau, *Optimal Regularization Parameter Estimation for Image Restoration*, in Proc. SPIE Conf. on Image Processing Algorithms and Techniques II, Vol. 1452 (1991), pp. 127–138.

[29] I.J.Schoenberg, *Contributions to the Problem of Approximation of Equistant Data by Analytic Functions*, Quart. Appl. Math., 4 (1946), pp.45-99,112-141.

[30] Schumaker, Larry L., *Spline Functions*, John Wiley & Sons, New York, 1981.

[31] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[32] E. Tyrtyshnikov, *Circulant Preconditioners with Unbounded Inverses*, Linear Algebra Appl., 216 (1995), pp. 1–24.

[33] J. Walker, *Fourier Analysis*, Oxford University Press, New York, 1988.