



THE AUSTRALIAN NATIONAL UNIVERSITY

**TR-CS-97-12**

**Preconditioning of elliptic problems  
by approximation in the transform  
domain**

**Michael K. Ng**

**July 1997**

Joint Computer Science Technical Report Series

Department of Computer Science  
Faculty of Engineering and Information Technology

Computer Sciences Laboratory  
Research School of Information Sciences and Engineering

This technical report series is published jointly by the Department of Computer Science, Faculty of Engineering and Information Technology, and the Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University.

Please direct correspondence regarding this series to:

Technical Reports  
Department of Computer Science  
Faculty of Engineering and Information Technology  
The Australian National University  
Canberra ACT 0200  
Australia

or send email to:

`Technical.Reports@cs.anu.edu.au`

A list of technical reports, including some abstracts and copies of some full reports may be found at:

<http://cs.anu.edu.au/techreports/>

**Recent reports in this series:**

- TR-CS-97-11 Richard P. Brent, Richard E. Crandall, and Karl Dilcher. *Two new factors of Fermat numbers*. May 1997.
- TR-CS-97-10 Andrew Tridgell, Richard Brent, and Brendan McKay. *Parallel integer sorting*. May 1997.
- TR-CS-97-09 M. Manzur Murshed and Richard P. Brent. *Constant time algorithms for computing the contour of maximal elements on the Reconfigurable Mesh*. May 1997.
- TR-CS-97-08 Xun Qu, Jeffrey Xu Yu, and Richard P. Brent. *A mobile TCP socket*. April 1997.
- TR-CS-97-07 Richard P. Brent. *A fast vectorised implementation of Wallace's normal random number generator*. April 1997.
- TR-CS-97-06 M. Manzur Murshed and Richard P. Brent. *RMSIM: a serial simulator for reconfigurable mesh parallel computers*. April 1997.

# Preconditioning of Elliptic Problems by Approximation in the Transform Domain

Michael K. Ng \*

6 March 1996

Revised: February 13, 1997

## Abstract

Preconditioned conjugate gradient method is applied for solving linear systems  $Ax = b$  where the matrix  $A$  is the discretization matrix of second-order elliptic operators. In this paper, we consider the construction of the transform based preconditioner from the viewpoint of image compression. Given a smooth image, a major portion of the energy is concentrated in the low frequency regions after image transformation. We can view the matrix  $A$  as an image and construct the transformed based preconditioner by using the low frequency components of the transformed matrix. It is our hope that the smooth coefficients of the given elliptic operator can be approximated well by the low-rank matrix. Numerical results are reported to show the effectiveness of the preconditioning strategy. Some theoretical results about the properties of our proposed preconditioners and the condition number of the preconditioned matrices are discussed.

**Key Words.** Transform approximation, image compression, preconditioned conjugate gradient method, elliptic partial differential equation.

**AMS Subject Classifications.** 65F10, 65N22

---

\*ACSys, Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT 0200, Australia. E-mail: mng@cslab.anu.edu.au.

# 1 Introduction

In this paper, we study the numerical solution of second-order elliptic boundary value problems. Using finite differences or finite elements, such problems generally are reduced to linear systems of the form  $Ax = b$ , where  $A$  is a sparse matrix. We consider here symmetric and positive definite problems. One of the most popular iterative methods for solving such systems is the conjugate gradient (CG) method, see Axelsson and Barker [1, p.18]. In general, the convergence rate of the CG method depends on the condition number  $\kappa(A)$  of  $A$ . The smaller  $\kappa(A)$  is, the faster the convergence of the method will be. One way to speed up the convergence rate of the method is to precondition the system. Thus, instead of solving the original system  $Ax = b$ , we solve the preconditioned system  $M^{-1}Ax = M^{-1}b$ . The matrix  $M$ , called a preconditioner to the matrix  $A$ , is chosen with two criteria in mind:  $Mr = d$  is easy to solve for any vector  $d$ ; the spectrum of  $M^{-1}A$  is clustered and/or  $M^{-1}A$  is well-conditioned compared to  $A$ .

In [2]. R. Chan and T. Chan proposed using Fast Fourier Transforms (FFTs) based preconditioners for the numerical solutions of second-order elliptic equations. Their approach is to average the coefficients of  $A$  to form a circulant approximation. Part of their motivation is to exploit the fast inversion of circulant systems via the Fast Fourier Transforms (FFTs). They proved that circulant preconditioners can be chosen so that  $\kappa(M^{-1}A) = O(n)$ , where  $1/n$  is the mesh-size. Works in this direction can be found in Huckle [10] where skew circulant preconditioners are used for these discretization matrices. In [13, 14], the construction of circulant approximations on anisotropic elliptic problems and L-shaped domains of elliptic problems has been considered. Moreover, the idea of FFT-based preconditioners has been applied to systems arising from implicit time-marching methods for first-order hyperbolic equations, see Holmgren and Otto [8, 9].

One can view the circulant preconditioners as approximations to the original given equation with the given boundary conditions being replaced by periodic ones. It is thus natural to consider using other fast transform based preconditioners to precondition elliptic problems. Chan and Wong [5] proposed using sine transform based preconditioners for matrices  $A$ . For rectangular regions, the condition number of the sine transform based preconditioned system is proved to be of  $O(1)$ .

In this paper, we consider the construction of the transform based preconditioner from the viewpoint of image compression. In image processing, it is well known that for typical images, a major portion of the signal energy is concentrated in the low frequency regions after image transformation [11]. Thus it is already sufficient to reconstruct the image if only the low frequency regions of the compressed image are kept in the frequency domain. This leads us to approximate the transformed matrix  $SAS$  in the transform domain by the sum of a diagonal matrix and a low-rank matrix. Here  $S$  is the discrete sine transform matrix. The low-rank matrix corresponds to the low frequency components of the transformed matrix. It is our hope that the variations of the smooth coefficients of the given elliptic operator can be approximated well by the low-rank matrix in the transform domain. Numerical results in §4 are given to illustrate the effectiveness of the preconditioning technique. We will also see that the performance of our preconditioner is better than the FFT-based preconditioner and the sine transform based

preconditioner proposed in [2] and [5] respectively.

Let  $\ell$  be the rank of the low-rank matrix added in the transform domain. For the rectangular domain, both the construction of our proposed preconditioner  $M_\ell$  and the matrix-vector multiplication  $M_\ell^{-1}v$  for any vector  $v$  can be done in  $O((\ell^2 + 1)n^2 \log n) + O(\ell^2 n)$  operations. For the theoretical result, we will show that the condition number of the preconditioned system  $M_\ell^{-1}A$  is of order  $O(1)$ . Moreover, when  $\ell = 0$ , our proposed preconditioner  $M_\ell$  is just equal to the sine transform based preconditioner given in [5].

The outline of the paper is as follows. In §2, we construct our preconditioner using discrete sine transform matrix and the low-rank approximation. We also analyze the spectral condition number of the preconditioned system. In §3, we extend the construction of our preconditioner to irregular domains. Finally, numerical results are given in §4.

## 2 Preconditioners by Approximation in the Transform Domain

Consider the 2-dimensional elliptic problems

$$-(a(x, y)u_x)_x - (b(x, y)u_y)_y = f(x, y) \quad (1)$$

on the unit square  $[0, 1] \times [0, 1]$  with Dirichlet boundary condition. Assume that the coefficient functions  $a(x, y)$ ,  $b(x, y)$  satisfy

$$0 < c_{\min} \leq a(x, y), b(x, y) \leq c_{\max} \quad (2)$$

for some constants  $c_{\min}$  and  $c_{\max}$ . Let the domain be discretized by using a uniform grid with  $n$  internal grid points in each coordinate direction. With the usual 5-point centered differencing, the resulting discretization matrix  $A$  will be an  $n^2$ -by- $n^2$  symmetric positive definite matrix of the form

$$A = \begin{pmatrix} D_1 & A_2 & & & \\ A_2 & D_2 & A_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & A_n \\ & & & A_n & D_n \end{pmatrix}. \quad (3)$$

Here  $D_i$  are symmetric tridiagonal matrices for  $1 \leq i \leq n$  and  $A_i$  are diagonal matrices for  $2 \leq i \leq n$ .

The motivation for the construction of our preconditioners is given by the following result in the Fourier analysis. We first denote  $C^{(m)}$  to the set of functions that are  $m$ -times differentiable with its  $m$ th derivative in  $L^1[0, 2\pi]$  for  $m > 0$ . Moreover we let  $F_n$  be the discrete Fourier matrix with its entries given by

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

**Lemma 1** Given a periodic function  $a(s, t)$  defined on  $[0, 2\pi] \times [0, 2\pi]$ . Let  $A_n = [a_{j,k}]_{0 \leq j, k < n}$  with

$$a_{j,k} = a\left(\frac{2\pi j}{n}, \frac{2\pi k}{n}\right), \quad 0 \leq j, k < n.$$

If  $a(s, \cdot)$  and  $a(\cdot, t)$  are uniformly in  $C^{(m+1)}$  for  $0 \leq t \leq 2\pi$  and  $0 \leq s \leq 2\pi$  respectively with  $m > 0$ , then  $F_n^* A_n F_n$  can be approximated well by a matrix of low-rank.

**Proof:** The Fourier coefficients of  $a(s, t)$  are given by

$$f_{jk} = \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{2\pi} a(s, t) e^{-ijs} e^{-ikt} ds dt, \quad j, k = 0, \pm 1, \dots$$

They satisfy

$$|f_{jk}| \leq \frac{c}{\max\{|j| + 1, |k| + 1\}^{m+1}}, \quad \forall j, k,$$

for some constant  $c$  depending on  $a(s, t)$  (dependent on  $m$ ) but independent of  $j$  and  $k$ , see [12]. We let

$$b_{jk} = \frac{1}{n^2} \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} a\left(\frac{2\pi p}{n}, \frac{2\pi q}{n}\right) e^{-2\pi i p j / n} e^{-2\pi i q k / n},$$

Using the fact that  $a(s, t)$  are uniformly in  $C^{(m+1)}$  for  $0 \leq t \leq 2\pi$  and  $0 \leq s \leq 2\pi$  respectively with  $m > 0$ , we have

$$|b_{jk}| \leq \frac{c_1}{\max\{|j| + 1, |k| + 1\}^{m+1}}, \quad 0 \leq j, k < n. \quad (4)$$

Equating the entries of  $\frac{1}{n} F_n^* A_n F_n$  to  $b_{jk}$  and using (4), we obtain

$$\left| \left[ \frac{1}{n} F_n^* A_n F_n \right]_{j,k} \right| \leq \frac{c_2}{\max\{|j| + 1, |k| + 1\}^{m+1}}, \quad 0 \leq j, k < n.$$

Then we define the matrix  $L_n = [l_{j,k}]_{0 \leq j, k < n}$  by

$$[L_n]_{j,k} = \begin{cases} \left[ \frac{1}{n} F_n^* A_n F_n \right]_{j,k}, & 0 \leq j, k \leq \ell, \\ 0, & \ell < j, k < n. \end{cases}$$

Therefore, we have

$$\left\| \frac{1}{n} F_n^* A_n F_n - L_n \right\|_2 \leq \left\| \frac{1}{n} F_n^* A_n F_n - L_n \right\|_F \leq \sqrt{\sum_{j=\ell+1}^{n-1} \frac{c_2^2}{(j+1)^{2(m+1)}}} \leq \frac{c_3}{(\ell+1)^m}$$

where  $c_3$  is dependent on  $m$  but independent of  $\ell$  and  $n$ . Hence the result follows.  $\square$

It follows that the matrix  $A_n$  with the properties defined in Lemma 1 can be approximated well by a low-rank matrix in the transform domain. In image processing terminology, one

can view  $A_n$  as an image. In transform image coding, an image is transformed to a domain significantly different from the image spatial domain, and the transform coefficients are then coded. A large amount of energy is concentrated in a small fraction of the transform coefficients (low frequency components). Therefore, it is possible to code only a fraction of the transform coefficients without seriously affecting the image [11].

In the following, we will use the sine transform matrix to construct the preconditioner for the matrix  $A$  in (3).

## 2.1 Sine Transform Matrix

In this subsection we recall some of the results in approximating a given matrix by matrices that can be diagonalized by the discrete sine transform. Let  $S_n$  be the  $n$ -by- $n$  discrete sine transform matrix. Its  $(i, j)$ th entry is given by

$$\sqrt{\frac{2}{n+1}} \sin\left(\frac{\pi ij}{n+1}\right), \quad 1 \leq i, j \leq n.$$

We note that  $S_n$  is symmetric, orthogonal and the matrix-vector multiplication  $S_n v$  can be computed in  $O(n \log n)$  operations for any  $n$ -vector  $v$ , see Yip and Rao [15]. Given an  $n$ -by- $n$  matrix  $A_n$ . The minimizer  $s(A_n)$  of  $\|A_n - B\|_F$  over all matrices  $B$  that can be diagonalized by  $S_n$  is called the *optimal sine transform approximation* to  $A_n$ , see Chan et al. [4]. Here  $\|\cdot\|_F$  denotes the Frobenius norm of the matrix. The matrix  $s(A_n)$  is uniquely determined by  $A_n$  and is given by

$$s(A_n) = S_n \delta(S_n A_n S_n) S_n, \quad (5)$$

where  $\delta(S_n A_n S_n)$  denotes the diagonal matrix whose diagonal is equal to the diagonal of the matrix  $S_n A_n S_n$ , i.e.,

$$[\delta(S_n A_n S_n)]_{j,k} = \begin{cases} [S_n A_n S_n]_{j,j}, & j = k, \\ 0, & j \neq k. \end{cases}$$

In [4], we gave an approach of constructing  $s(A_n)$  with the cost  $O(n^2)$  operations. We remark that if  $A_n$  is a band matrix, then the cost can be reduced to  $O(n)$  operations.

It is well known that the discrete sine transform matrix  $S_n$  diagonalizes the set of tridiagonal, Toeplitz, symmetric matrices. By the definition of  $s(A_n)$  or (5), the optimal sine transform preconditioner gives exact approximation to all matrices in the set, in particular to the 1-dimensional discrete Laplacian:  $\text{tridiag}[-1, 2, -1]$ . Therefore, the system  $\text{tridiag}[-1, 2, -1]x = b$  can be solved in exactly one iteration by the PCG method with the optimal sine transform preconditioner.

## 2.2 Construction of Preconditioners

We note from (5) that the optimal sine transform based preconditioner is constructed by only extracting the information from the main diagonal of the transformed matrix  $S_n A_n S_n$ . From Lemma 1 and the transform coding in image processing, these lead us to use the sum of a

diagonal matrix and a low-rank matrix to construct the preconditioner for elliptic problems. We hope that the variation of the smooth coefficients of the given elliptic operator can be captured well by the low-rank approximation (low frequency components after image transformation) in the transform domain.

More precisely, given an  $n$ -by- $n$  matrix  $A_n$  and  $\ell \geq 0$ , we define

$$s_\ell(A_n) \equiv S_n \delta_\ell(S_n A_n S_n) S_n, \quad (6)$$

where  $\delta_\ell(S_n A_n S_n)$  is a matrix with its entries given by

$$[\delta_\ell(S_n A_n S_n)]_{j,k} = \begin{cases} [S_n A_n S_n]_{j,k}, & 0 \leq j, k \leq \ell, \\ [S_n A_n S_n]_{j,j}, & \ell + 1 \leq j < n, \\ 0, & \text{otherwise.} \end{cases}$$

We note that when  $\ell = 0$ ,  $\delta_\ell(\cdot) = \delta(\cdot)$  and  $s_\ell(\cdot) = s(\cdot)$  as in (5). Thus we see that our preconditioner is just a natural extension of the optimal sine transformed based preconditioner. However, we introduce the low-rank approximation in the preconditioner to deal with the variation of the coefficients of the given elliptic operator. The following Lemma gives the property of  $s_\ell(A_n)$ .

**Lemma 2** *If  $A_n$  is Hermitian, then  $s_\ell(A_n)$  is also Hermitian. Moreover, we have*

$$\lambda_{\min}(A_n) \leq \lambda_{\min}(s_\ell(A_n)) \leq \lambda_{\max}(s_\ell(A_n)) \leq \lambda_{\max}(A_n).$$

*In particular, if  $A_n$  is positive definite, then  $s_\ell(A_n)$  is also positive definite.*

**Proof:** It is easy to see that  $s_\ell(A_n)$  is Hermitian as  $S_n$  and  $S_n A_n S_n$  are. Let  $R_\ell$  be the  $(\ell + 1)$ -by- $(\ell + 1)$  principal submatrix of  $\delta_\ell(S_n A_n S_n)$ . We just note that

$$\begin{aligned} \lambda_{\min}(A_n) &= \lambda_{\min}(S_n A_n S_n) \\ &\leq \min\{\lambda_{\min}(R_\ell), \min_{\ell+1 \leq j < n} \{[S_n A_n S_n]_{j,j}\}\} \\ &= \lambda_{\min}(\delta_\ell(S_n A_n S_n)) \\ &= \lambda_{\min}(s_\ell(A_n)). \end{aligned}$$

The inequality for the supremum can be proved likewise.  $\square$

We construct a preconditioner  $M_\ell$  for the matrix  $A$  in (3) by taking the approximations  $s_\ell(\cdot)$  to each  $n$ -by- $n$  block of  $A$ :

$$M_\ell = \begin{pmatrix} s_\ell(D_1) & s_\ell(A_2) & & & & \\ s_\ell(A_2) & s_\ell(D_2) & s_\ell(A_3) & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & s_\ell(A_n) & \\ & & & s_\ell(A_n) & s_\ell(D_n) & \end{pmatrix}, \quad (7)$$

where  $s_\ell(\cdot)$  is the approximation given by (6).



That will take  $O(\ell^2 n + n^2)$  operations. Hence the cost of solving the linear system  $M_\ell x = b$  is  $O(n^2 \log n + \ell^2 n + n^2)$  operations.

We remark that the block Cholesky factorization of  $A$  can be written as

$$A = (\Sigma + L)\Sigma^{-1}(\Sigma + L^T), \quad (11)$$

where

$$L = \begin{pmatrix} 0 & & & & & \\ A_2 & 0 & & & & \\ & A_3 & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & A_n & 0 \end{pmatrix}$$

is the lower triangular matrix and  $\Sigma$  is a symmetric block diagonal matrix with diagonal blocks  $\Sigma_i$  satisfying

$$\begin{aligned} \Sigma_1 &= D_1, \\ \Sigma_i &= D_i - A_i \Sigma_{i-1}^{-1} A_i, \quad 2 \leq i \leq n. \end{aligned} \quad (12)$$

We see that the preconditioner  $M_\ell$  is also of this form and can be constructed by the method of block factorization.

**Lemma 4** *Let  $A$  be an  $n^2$ -by- $n^2$  symmetric positive definite matrix of the form given in (3). Let  $M_\ell$  be the preconditioner for  $A$ . Then the block factorization of  $M_\ell$  is given by*

$$(\Phi_\ell + \hat{L}_\ell)\Phi_\ell^{-1}(\Phi_\ell + \hat{L}_\ell^T) \quad (13)$$

where

$$\hat{L}_\ell \equiv \begin{pmatrix} 0 & & & & & \\ s_\ell(A_2) & 0 & & & & \\ & s_\ell(A_3) & \cdot & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & s_\ell(A_n) & 0 \end{pmatrix} \quad (14)$$

is a block lower triangular matrix and

$$\Phi_\ell \equiv \begin{pmatrix} \Phi_\ell^{(1)} & & & 0 \\ & \Phi_\ell^{(2)} & & \\ & & \ddots & \\ 0 & & & \Phi_\ell^{(n)} \end{pmatrix}$$

is a diagonal block matrix with diagonal blocks  $\Phi_\ell^{(i)}$  satisfying

$$\begin{aligned} \Phi_\ell^{(1)} &= s_\ell(D_1), \\ \Phi_\ell^{(i)} &= s_\ell(D_i) - s_\ell(A_i)(\Phi_\ell^{(i-1)})^{-1}s_\ell(A_i), \quad 2 \leq i \leq n. \end{aligned} \quad (15)$$

Next we show that the condition number of the preconditioned system  $\kappa(M_\ell^{-1}A)$  is bounded by a constant which is independent of the size of the matrix  $A$ .

**Theorem 1** *Let  $A$  be the 5-point discretization matrix of (1) on the unit square satisfying conditions (2). If  $M_\ell$  is the preconditioner defined in (7), we have*

$$\kappa(M_\ell^{-1}A) \leq \left(\frac{c_{\max}}{c_{\min}}\right)^2.$$

**Proof:** Under the assumption, we have

$$c_{\min}A_0 \leq A \leq c_{\max}A_0$$

where  $A_0$  is the 5-point discretization matrix of the Laplacian operator on the unit square. It suffices to show the following equality:

$$c_{\min}A_0 \leq M_\ell \leq c_{\max}A_0.$$

To prove the above inequality, we note that

$$\begin{aligned} c_{\min}\lambda_{\min}((I \otimes S_n)A_0(I \otimes S_n)) &\leq \lambda_{\min}((I \otimes S_n)A(I \otimes S_n)) \\ &\leq \lambda_{\min}(M_\ell) \\ &\leq \lambda_{\max}(M_\ell) \\ &\leq \lambda_{\max}((I \otimes S_n)A(I \otimes S_n)) \\ &\leq c_{\max}\lambda_{\max}((I \otimes S_n)A_0(I \otimes S_n)) \end{aligned}$$

The result follows.  $\square$

### 3 Irregular Domains

We consider irregular domains that are union of rectangular domains. In this case, the matrix  $A$  has the form given in (3) but the diagonal submatrices  $D_i$  of  $A$  are of different sizes and the submatrices  $A_i$  may not be square matrices. We use the result in Lemma 4 and derive the preconditioner for this case.

When  $A_i$  are square, the preconditioner can be constructed using the method in Lemma 4. Therefore, we only concentrate on the sub-block of  $A$  where the  $A_i$  are not square. Assume that only one of the sub-block of  $A$  is not square, let  $D_i$  and  $D_{i+1}$  be  $n_1$ -by- $n_1$  and  $n_2$ -by- $n_2$  symmetric tridiagonal matrices respectively and  $A_{i+1}$  is an  $n_2$ -by- $n_1$  matrix where

$$[A_{i+1}]_{j,k} = 0, \quad j \neq k.$$

For simplicity, we also assume that  $n_1 > n_2$ . The block factorization of  $A$  in this case still be of the form given by (11). Therefore, we construct our preconditioner in the form given by (13) except that the matrix  $\Phi_\ell^{(i)}$ , which are approximations to  $\Sigma_i$ , are defined as follows:

$$\begin{aligned}\Phi_\ell^{(1)} &= s_\ell(D_1), \\ \Phi_\ell^{(k)} &= s_\ell(D_k) - s_\ell(A_k J) s_\ell(J^T (\Phi_\ell^{(k-1)})^{-1} J) s_\ell(J^T A_k^T), \quad k = i + 1, \\ \Phi_\ell^{(k)} &= s_\ell(D_k) - s_\ell(A_k) s_\ell(\Phi_\ell^{(k-1)})^{-1} s_\ell(A_k^T), \quad k \neq i + 1\end{aligned}$$

where  $J$  is an  $n_1$ -by- $n_2$  matrix such that

$$[J]_{j,k} = \begin{cases} 1, & j = k, \\ 0, & \text{otherwise.} \end{cases}$$

We remark that constructing  $s_\ell(J^T (\Phi_\ell^{(i)})^{-1} J)$  requires  $O(n_1 n_2)$  operations. Therefore, for a given irregular domain which is the union of  $N$  rectangular regions, the construction cost of the preconditioner is bounded by  $O(N n^2) + O(n^2 \log n) + O(\ell^3 n)$  where  $n$  is the size of the largest diagonal block of  $A$ . For an L-shaped domain, the construction cost of the preconditioner is bounded by  $O(n^2 \log n) + O(\ell^3 n)$ . In solving the system with  $M_\ell x = b$ , we are required to multiply each  $\Phi_\ell^{(i)}$  and  $s_\ell(A_i)$  to some vector and to solve systems with coefficient matrices  $\Phi_\ell^{(i)}$ . By noting the forms of  $\Phi_\ell^{(i)}$  and  $s_\ell(A_i)$  in the transform domain, the system  $M_\ell x = b$  can be solved in  $O(n^2 \log n + \ell^3 n)$  operations.

## 4 Numerical Results

In this section, we compare the performance of our preconditioners with the MINV-type preconditioners in [6], the FFT-based preconditioners in [2]. The equations we tested are

$$(i) \quad \frac{\partial}{\partial x} [(1 + \epsilon e^{x+y}) \frac{\partial u}{\partial x}] + \frac{\partial}{\partial y} [(1 + \frac{\epsilon}{2} \sin(2\pi(x+y))) \frac{\partial u}{\partial y}] = f(x, y),$$

$$(ii) \quad \frac{\partial}{\partial x} [(1 + \epsilon e^{xy}) \frac{\partial u}{\partial x}] + \frac{\partial}{\partial y} [(1 + \epsilon(x^2 + y^2)) \frac{\partial u}{\partial y}] = f(x, y),$$

$$(iii) \quad \epsilon \frac{\partial}{\partial x} [(1 + e^{x+y}) \frac{\partial u}{\partial x}] + \frac{\partial}{\partial y} [(1 + \frac{1}{2} \sin(2\pi(x+y))) \frac{\partial u}{\partial y}] = f(x, y),$$

on the unit square. The  $\epsilon$  is a parameter controlling the variation of the coefficient function and the  $\varepsilon$  is a parameter controlling the ratio of anisotropy of elliptic problems. We discretize the equation using the standard 5-point scheme. The initial guess and the right hand side are chosen

to be random vectors and are the same for all methods. The PCG iterations are stopped when the residual vector  $r_k$  at the  $k$ -th iteration satisfies  $\|r_k\|_2/\|r_0\|_2 < 10^{-7}$ . In our experiments, we first symmetrically scale  $A$  by its diagonal before applying the preconditioner. This technique has also proven to be very useful when used in conjunction with other kinds of preconditioners. We apply diagonal scaling to all methods.

Tables 1–3 show the numbers of iterations required for convergence for equations (i), (ii) and (iii) respectively. In Tables 1–3, the notation  $I$  means no preconditioner is used, the symbol  $C$  denotes that the FFT-based block-factorization preconditioner is used and the parameter  $n$  is equal to  $1/h$  where  $h$  is the mesh-size. We remark that when  $\ell = 0$ , the preconditioner  $M_\ell$  is just equal to the sine transformed based preconditioner proposed by Chan and Wong [5]. We see from Tables 1–3 that the performance of our preconditioner  $M_{15}$  is better than the others, especially when  $n$  is large. In addition, Figures 1 and 2 show the eigenvalue distributions of the preconditioned systems for equations (i) with  $\varepsilon = 1$  and 2 respectively. We note that for the sine transformed based preconditioned systems and the MINV preconditioned systems, the eigenvalues are clustered around 1 and are in a relatively small interval. Therefore, by incorporating low-rank approximations in the sine transform based preconditioner, we can improve the performance of the PCG method.

In Table 4, we compare the time per PCG iteration for different types of preconditioners in the Ultra SPARC workstation. In order to exploit the Fast Fourier Transforms (FFTs) and the Fast Sine Transforms (FSTs) algorithms, we choose  $n = 2^m - 1$  for the sine transform based preconditioners and  $n = 2^m$  for the FFT-based preconditioners. We use the FFTs and FSTs programs obtained from the double precision version of FFTPACK in NETLIB, to implement the PCG method. We observed from Table 4 that the time per PCG step for the sine transformed based preconditioner is relatively expensive than that for MINV preconditioner. In Figure 2, we plot the time that required for solving equation (i) by the PCG method against the grid size  $n$ . We observe that the time achieved to reach convergence, with the sine transform based preconditioners and the MINV preconditioners are almost the same when the grid size  $n$  is large. However, the FST is highly parallelizable and therefore we expect that the method is efficient in the parallel computing environment.

To further illustrate the performance of the preconditioner, Table 5 shows the numbers of iterations required for convergence for equation (i) on the L-shaped domain  $[0, \frac{1}{2}] \times [0, 1] \cup [\frac{1}{2}, 1] \times [0, \frac{1}{2}]$ . We again see that the number of iterations of our proposed sine transformed based preconditioner is less than the others.

In summary, we propose sine transformed based block factorization preconditioners for elliptic problems from the viewpoint of image transform coding. Preliminary numerical results show that the preconditioning strategy may be an efficient and effective method. We also remark that the method can be extended to other transform based preconditioners like the cosine transform based preconditioner for queueing problem, see [3].

*Acknowledgment:* The author wishes to thank C.K. Wong at UCLA for his code and thank the referees for their valuable comments.

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	31	5	15	8	7	6	1	1
16	60	7	21	10	9	7	6	1
32	115	11	29	12	11	9	7	7
64	224	16	44	14	13	12	10	8
128	435	24	55	17	15	14	12	10

(a)

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	34	4	17	10	8	6	1	1
16	68	6	25	16	13	9	7	1
32	132	10	38	26	21	15	9	7
64	261	15	59	38	31	26	16	9
128	562	26	74	54	43	34	25	14

(b)

Table 1: Number of Iterations for Different Preconditioners when (a)  $\epsilon = 1$  and (b)  $\epsilon = 2$  in (i) on the Square Domain.

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	30	5	14	8	7	8	1	1
16	58	8	18	11	9	8	7	1
32	112	12	25	13	12	11	8	7
64	225	18	38	15	15	12	12	8
128	462	26	56	18	17	14	13	10

(a)

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	29	5	14	9	7	6	1	1
16	56	8	20	13	11	9	7	1
32	114	12	28	18	16	13	9	8
64	226	18	40	23	20	18	14	10
128	465	27	58	30	27	24	19	14

(b)

Table 2: Number of Iterations for Different Preconditioners when (a)  $\epsilon = 10$  and (b)  $\epsilon = 50$  in (ii) on the Square Domain.

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	37	3	6	6	5	4	1	1
16	82	4	9	8	6	5	4	1
32	160	6	13	11	8	7	5	4
64	311	9	18	13	11	8	7	5
128	609	13	25	16	13	11	10	7

(a)

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	42	2	13	9	7	6	1	1
16	87	3	13	10	8	6	4	1
32	198	4	14	11	8	7	4	3
64	396	6	15	13	9	8	5	4
128	821	8	16	15	11	9	6	5

(b)

Table 3: Number of Iterations for Different Preconditioners when (a)  $\varepsilon = 10$  and (b)  $\varepsilon = 0.001$  in (iii) on the Square Domain.

	$I$	MINV	$C$	$M_0$
$m$	$n = 2^m - 1$	$n = 2^m - 1$	$n = 2^m$	$n = 2^m - 1$
3	$2.03 \times 10^{-4}$	$3.58 \times 10^{-4}$	$9.79 \times 10^{-4}$	$5.58 \times 10^{-4}$
4	$5.61 \times 10^{-4}$	$1.16 \times 10^{-3}$	$3.36 \times 10^{-3}$	$1.98 \times 10^{-3}$
5	$2.56 \times 10^{-3}$	$5.02 \times 10^{-3}$	$1.70 \times 10^{-2}$	$1.01 \times 10^{-2}$
6	$1.15 \times 10^{-2}$	$2.22 \times 10^{-2}$	$7.23 \times 10^{-2}$	$4.11 \times 10^{-2}$
7	$5.40 \times 10^{-2}$	$8.78 \times 10^{-2}$	0.3318	0.1976
	$M_1$	$M_3$	$M_7$	$M_{15}$
$m$	$n = 2^m - 1$			
3	$6.01 \times 10^{-4}$	$6.11 \times 10^{-4}$	-----	-----
4	$2.02 \times 10^{-3}$	$2.15 \times 10^{-2}$	$2.20 \times 10^{-3}$	-----
5	$1.06 \times 10^{-2}$	$1.14 \times 10^{-2}$	$1.16 \times 10^{-2}$	$1.17 \times 10^{-2}$
6	$4.28 \times 10^{-2}$	$4.30 \times 10^{-2}$	$4.34 \times 10^{-2}$	$4.36 \times 10^{-2}$
7	0.2001	0.2012	0.2028	0.2035

Table 4: Times (in second) per PCG iteration for Different Preconditioners.

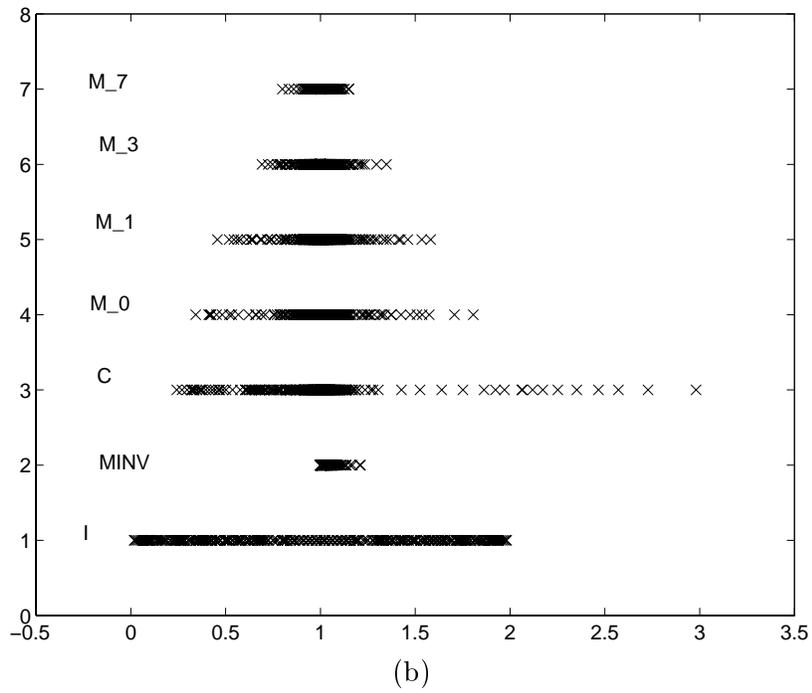
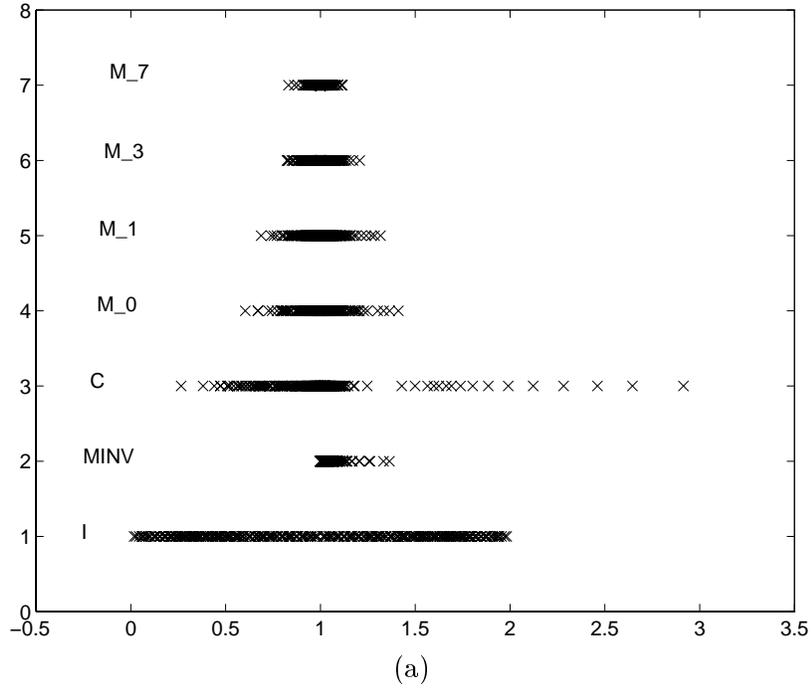


Figure 1: Spectra of the Preconditioned Systems for  $n = 16$  when  $\varepsilon = 1$  and (b)  $\varepsilon = 2$  in (i) on the Square Domain.

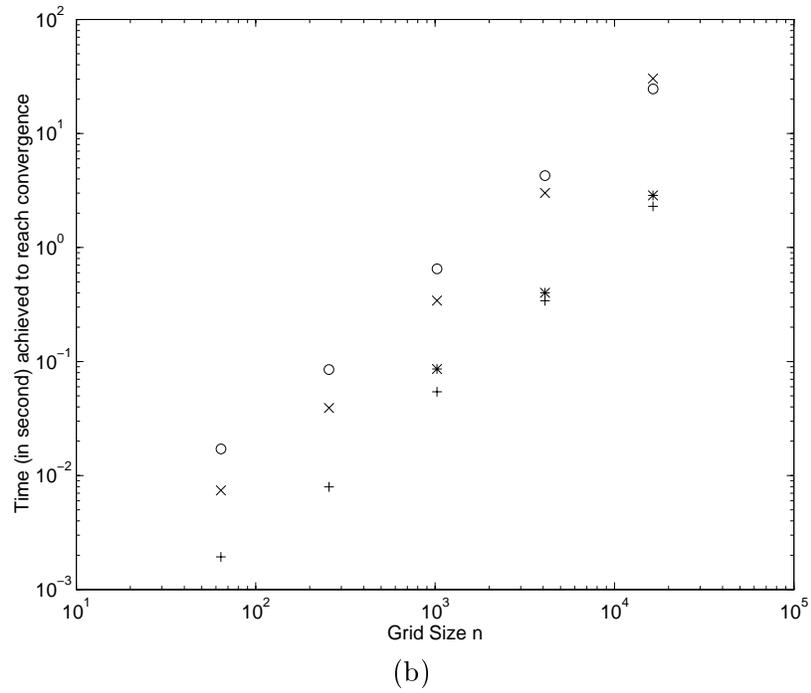
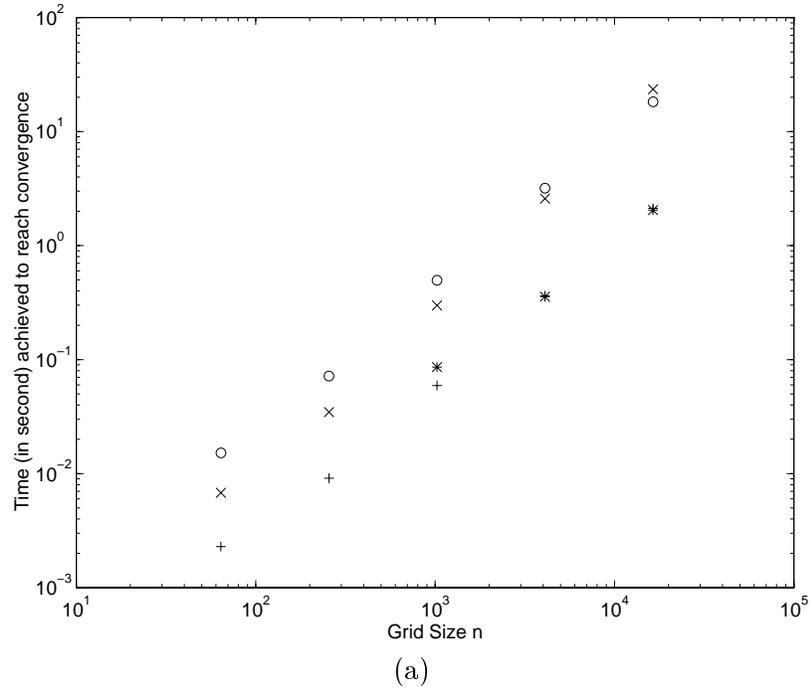


Figure 2: (a)  $\epsilon = 1$ , (b)  $\epsilon = 2$ . Here x denotes using  $I$ , + denotes using MINV, o denotes using  $C$ , \* denotes using  $M_{15}$ .

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	28	5	14	8	6	5	4	3
16	49	7	22	10	9	8	7	6
32	90	12	30	13	13	11	10	9
64	185	17	47	17	17	14	13	12
128	425	25	61	23	21	19	18	16

(a)

$n$	$I$	MINV	$C$	$M_\ell$				
				$\ell = 0$	$\ell = 1$	$\ell = 3$	$\ell = 7$	$\ell = 15$
8	31	4	17	10	9	7	5	3
16	55	6	26	16	15	11	9	6
32	104	11	40	29	25	20	13	10
64	207	16	62	43	37	27	20	16
128	478	27	79	58	46	36	28	21

(b)

Table 5: Number of Iterations for Different Preconditioners when (a)  $\epsilon = 1$  and (b)  $\epsilon = 2$  in (i) on the L-shaped Domain.

## References

- [1] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic Press, Orlando, FL., 1983.
- [2] R. Chan and T. Chan, *Circulant Preconditioners for Elliptic Problems* J. Numer. Linear Algebra Appls., 1 (1992), pp. 77–101.
- [3] R. Chan, W. Ching and C. Wong, *Optimal Trigonometric Preconditioners for Elliptic Queueing Problems*, SEA Bull. Math., 20, No. 3 (1996), pp. 117–pp.124.
- [4] R. Chan, M. Ng, and C. Wong, *Sine Transform Based Preconditioners for Symmetric Toeplitz Systems*, Linear Algebra Appls., 232 (1996), pp. 237–259.
- [5] R. Chan and C. Wong, *Sine Transform Based Preconditioners for Elliptic Problems*, Technical Report 93-12-4, Dept. of Math., The Chinese University of Hong Kong, (1993).
- [6] P. Concus, G. Golub, and G. Meurant, *Block Preconditioning for the Conjugate Gradient Method*, SIAM J. Sci. Statist. Comput., 6 (1985) pp. 220–252.
- [7] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
- [8] S. Holmgren and K. Otto, *Iterative Solution Methods and Preconditioners for Block-tridiagonal Systems of Equations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 863–886.
- [9] S. Holmgren and K. Otto, *Semicirculant Preconditioners for First-order Partial Differential Equations*, SIAM J. Sci. Comput., 15 (1994), pp. 385–407.
- [10] T. Huckle, *A Note on Skew Circulant Preconditioners for Elliptic Problems*, Numer. Algo., 2 (1992), pp. 279–286.
- [11] A. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, London, 1989.
- [12] Y. Katznelson, *An Introduction to Harmonic Analysis*, Second Edition, Dover, New York, 1974.
- [13] I. Lirkov, S. Margenov and P. Vassilevski, *Circulant Block-factorization Preconditioners for Elliptic Problems*, Computing, 53 (1994), pp. 59–74.
- [14] I. Lirkov, S. Margenov and L. Zikatanov, *Circulant Block-factorization Preconditioners for Anisotropic Elliptic Problems*, CAM. Rep. 95-39, UCLA, 1995.
- [15] P. Yip and K. Rao, *Fast Decimation-in-time Algorithms for a Family of Discrete Sine and Cosine Transforms*, Circuits Systems Signal Process., 3 (1984), pp. 387–408.