Optimal Trigonometric Preconditioners for Elliptic and Queueing Problems

Raymond H. Chan* Wai-Ki Ching† Chiu-Kwong Wong‡

March, 1994

Abstract

We consider using the preconditioned conjugate gradient (PCG) method to solve linear systems $A\mathbf{x} = \mathbf{b}$ arising from second-order elliptic problems and queueing problems. The preconditioners are matrices that can be diagonalized by either sine or cosine transform matrices. For 2-dimensional elliptic problems with slowly variating coefficients, the condition numbers of our preconditioned system is of order $O(1)$ whereas the system preconditioned by the MILU and MINV methods are of order $O(n)$, where n is the number of mesh points in one direction. For queueing problems, our method is also significantly faster than the point-SOR method.

1 Optimal Sine Transform Based Preconditioners

Let A be an n-by-n positive definite matrix. Consider solving the linear system $A\mathbf{x} = \mathbf{b}$ by the conjugate gradient (CG) method. If the condition number $\kappa(A)$ of A is close to 1, then the CG method converges very fast, see [1]. However, if $\kappa(A)$ is large, preconditioning is needed, i.e. we solve $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$. There are two criteria on choosing the preconditioner M. We want $M \approx A$ in some norm and yet M is easily invertible, see [1].

One approach to the second requirement is to use the circulant matrices as preconditioners. The reason is that all circulant matrices C can be diagonalized by Fourier matrix F. Hence C^{-1} can be easily invertible. Let us denote the class of circulant matrices by $\mathcal{F}$. For the first requirement that $C \approx A$ in some

*Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong.
Research supported in part by HKRGC grant no. CUHK 178/93E and CUHK DAG grant no. 220600280.
†Department of Mathematics, University of Hong Kong, Pokfulam Road, Hong Kong.
‡Department of Mathematics, University of Hong Kong, Pokfulam Road, Hong Kong.

norm, T. Chan [7] proposed circulant preconditioner that is the minimizer of the Frobenius norm

$$\|C - A\|_F \tag{1}$$

over $\mathcal{F}$. The minimizer, denoted by $c(A)$, is called the *optimal circulant preconditioner* for the matrix A. If the matrix A is the 1-dimensional discrete Laplacian with Dirichlet boundary conditions, we have $\kappa(A) = O(n^2)$ but $\kappa(c(A)^{-1}A) = O(n^{1.5})$. Here n is the number of mesh points used. Modifying $c(A)$ in the manner used in Modified ILU method can further reduce the condition number to $O(n)$, see [4].

In this paper, our preconditioners are matrices that can be diagonalized by either sine or cosine transform matrices. For simplicity, we concentrate on the sine transform matrix S first. One of our motivations for using S is that the transform matrices S and F possess basically the same properties. We have F is unitary and S is orthogonal. Both matrices have fast transform algorithm to perform the matrix-vector multiplication. Their main difference is that S diagonalizes the 1-dimensional discrete Laplacian with Dirichlet boundary condition while F diagonalizes the same operator but with periodic boundary condition.

However, from the results in [12], we know that for matrices arising from elliptic boundary value problem, a "good" preconditioner must keep the boundary condition of the given operator. Therefore, for Dirichlet problems, we consider preconditioners B such that $B = S\Lambda S$. Let $\mathcal{S}$ denote such class of matrices. Similar to (1), for any matrix A, we define the *optimal sine preconditioner* $s(A)$ to be the minimizer of the Frobenius norm $\|B - A\|_F$ over all $B \in \mathcal{S}$.

Since $\|\cdot\|_F$ is unitary invariant, we have $s(A) = S\delta(SAS)S$, where $\delta(B)$ is the diagonal matrix with diagonal entries given by the diagonal of B. However, constructing $s(A)$ by this formula requires $O(n^2 \log n)$ operations for general matrices A. whereas $c(A)$ can be obtained in only $O(n^2)$ operations. Thus we need a faster algorithm of computing $s(A)$.

2 Constructing $s(A)$

In forming $c(A)$, we used the fact that $\mathcal{F}$ has a basis $\{P_i\} = \{P^i\}$ where P is just the shift operator. To form $s(A)$, we first need a basis $\{Q_i\}$ of $\mathcal{S}$.

lemma [**2, Bini and Benedetto**]

$$[Q_i]_{h,k} = \begin{cases} 1 & \text{if } |h-k| = i-1, \\ -1 & \text{if } h+k = i-2, \\ -1 & \text{if } h+k = 2n-i+3, \\ 0 & \text{otherwise.} \end{cases}$$

Note that each basis matrix is a special sum of a Toeplitz matrix and Hankel matrix. Hence all elements in $\mathcal{S}$ are special Toeplitz-plus-Hankel matrices. To form $s(A)$ fast, we make use of the sparsity of $\{Q_i\}$ and obtained the following theorem.

theorem [**5, Chan, Ng and Wong**] The matrix $s(A)$ can be formed in $O(n^2)$ operations for general A and $O(n)$ operations if A is Toeplitz or banded. We remark that the operation counts for computing $s(A)$ are thus the same as that of $c(A)$.

3 Extension to 2-Dimensional Case

Consider the second order elliptic equation

$$-(a(x,y)u_x)_x - (b(x,y)u_y)_y = f(x,y) \quad \text{on } [0,1]^2$$

with Dirichlet boundary condition. Using the 5-point centered differencing gives

$$A = \begin{pmatrix} D_1 & A_2 & & & 0 \\ A_2 & D_2 & A_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & A_n \\ 0 & & & A_n & D_n \end{pmatrix}. \tag{2}$$

where D_i are symmetric tridiagonal matrices and A_i are diagonal matrices.

There are two approachs of constructing preconditioners for A. One can use the "block" approach introduced by T. Chan and Olkin [8] and take the optimal sine approximation of each block of A to obtain the preconditioner:

$$s_1(A) = \begin{pmatrix} s(D_1) & s(A_2) & & & 0 \\ s(A_2) & s(D_2) & s(A_3) & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & s(A_n) \\ 0 & & & s(A_n) & s(D_n) \end{pmatrix}.$$

We noted that for the 2-dimensional discrete Laplacian matrix A given in (2), $s_1(A) = A$ and we have an exact approximation. In general, we have the following theorem.

theorem [**6, Chan and Wong**] For the variable coefficient case,

$$\kappa(s_1(A)^{-1}A) \leq (\frac{c_{\max}}{c_{\min}})^2,$$

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

However, this "block" approach does not work for non-rectangular domains. For those domains, we need the "INV" approach introduced in [9]. We first note that the block Cholesky factorization of the matrix A in (2) is given by

$$A = (\Sigma + L)\Sigma^{-1}(\Sigma + L^t),$$

where

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

and $\Sigma = \text{diag}[\Sigma_1, \cdots, \Sigma_n]$ with

$$\Sigma_i \quad = \quad D_i - A_i \Sigma_{i-1}^{-1} A_i.$$

The preconditioner in [9], called the INV preconditioner, is obtained by approximating Σ_i by band matrices. Here we just use the optimal sine operator $s(\cdot)$ on Σ_i to obtain our approximation.

theorem [**6, Chan and Wong**] For rectangular domains, two approaches are equivalent.

For L,C, or T-shaped regions, the diagonal block D_i in (2) may have different sizes and the subdiagonal block A_i may not be square. However, we can still apply $s(\cdot)$ to the sub-block Σ_i.

For comparison, let us list in Table 1 the construction cost of different preconditioners and the cost per iteration of CG method if the preconditioner is used. In the table, n is number of mesh points in one direction, "No" means no preconditioner is used. For references on MINV and MILU preconditioners, see [9] and [10].

Construction Cost				Cost per Iteration			
No	$s_1(A)$	MINV	MILU	No	$s_1(A)$	MINV	MILU
0	$n^2 \log n$	n^2	n^2	n^2	$n^2 \log n$	n^2	n^2

Table 1: Construction Cost and Cost per Iteration.

4 Numerical Results for Elliptic Problems

Let us consider solving the second order elliptic problem

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

with Dirichlet boundary conditions. We note that the parameter ϵ controls the variations in the coefficient functions. The equation is discretized by standard five point scheme. Random initial guess and right hand side are used in the CG method, ut these vectors are kept the same for different preconditioners. The tolerance is 10^{-6} and again n is the number of mesh points in one direction. The numbers of iterations required for convergence for the case of rectangular domain and L-shaped domain are given in Tables 2 and 3 respectively.

ϵ	0.0				0.01			
n	No	$s_1(A)$	MINV	MILU	No	$s_1(A)$	MINV	MILU
4	9	1	4	7	12	3	4	7
8	23	1	5	9	25	3	5	9
16	43	1	7	13	47	3	7	13
32	84	1	11	19	90	3	11	20
64	165	1	16	28	186	3	16	28
128	318	1	24	41	363	3	24	41
ϵ	0.1				1.0			
n	No	$s_1(A)$	MINV	MILU	No	$s_1(A)$	MINV	MILU
4	13	3	3	7	15	5	3	6
8	26	4	5	9	29	7	4	9
16	46	5	8	13	54	9	6	14
32	97	5	11	20	107	11	10	20
64	189	5	16	28	209	12	15	28
128	379	5	24	41	419	13	22	41

Table 2: Numbers of Iterations for Rectangular Domain $[0,1]^2$.

ϵ	0.0				0.01			
n	No	$s_1(A)$	MINV	MILU	No	$s_1(A)$	MINV	MILU
8	21	3	4	9	22	3	4	9
16	42	3	6	12	40	3	6	12
32	77	4	9	17	80	4	9	17
64	155	4	14	25	155	4	14	25
128	306	4	21	36	311	4	21	36
ϵ	0.1				1.0			
n	No	$s_1(A)$	MINV	MILU	No	$s_1(A)$	MINV	MILU
8	22	4	4	9	24	7	4	9
16	42	5	6	12	45	9	6	13
32	82	5	9	18	86	10	8	18
64	162	6	14	25	169	12	12	26
128	322	7	21	36	338	14	19	37

Table 3: Numbers of Iterations for L-shaped Domain $[0,1]^2 \setminus [\frac{1}{2},1]^2$.

5 Extension to Queueing Problems

Consider a 2-queue Markovian network with overflow from queue 1 to queue 2 permitted only when queue 1 is full, see Figure 1. We are interested in finding the steady-state probability distribution vector, see [11].

Let us denote λ_i, μ_i, n_i and s_i to be the input rate, output rate of single server, buffer sizes and number of servers for queue i. If the traffic density

$$\frac{\lambda_i}{s_i\mu_i} = 1 + O(n_i^{-\alpha}),$$

for some positive α, then the queueing problem resembles a second order elliptic equation on a rectangle with oblique boundary condition on one side and Neumann boundary condition on the others, see [3].

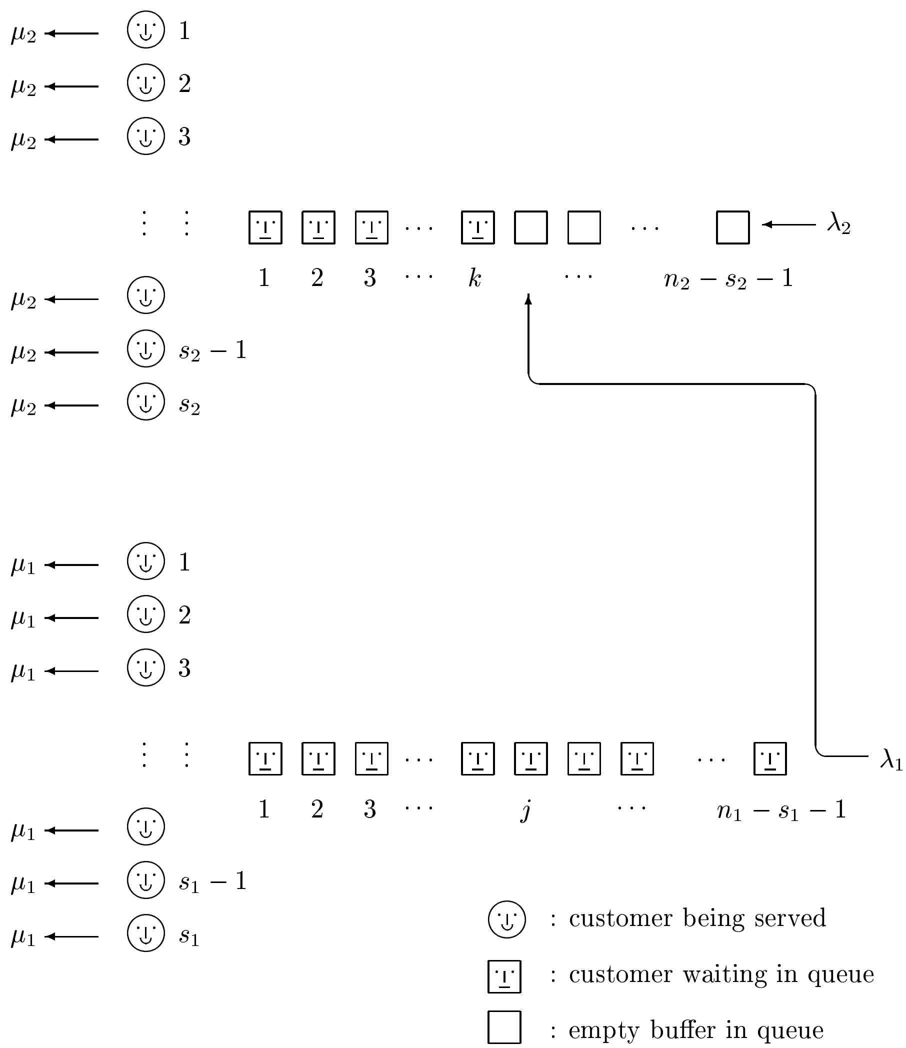

Figure 1. Two-queue Overflow Network.

The SOR method is the standard method in solving this problem, see [11]. However, in [3], CG method has also been considered, with the preconditioner being constructed by changing the oblique boundary condition to Neumann boundary condition. This preconditioner will be referred to as Neumann preconditioner in below. Since the 1-dimensional discrete Laplacian with Neumann boundary conditions can be diagonalized by cosine transform matrix, one naturally leads to consider using optimal cosine preconditioner to precondition such queueing system. The definition of such preconditioner is similar to that of optimal sine preconditioner and will be omitted here.

The following table compares the cost per iteration of the three different methods.

	$s = 1$			$s > 1$		
Cost per	Cosine	Neumann	PSOR	Cosine	Neumann	PSOR
Iteration	$n^2 \log n$	$n^2 \log n$	n^2	$n^2 \log n$	n^3	n^2

Table 4. Cost per Iteration.

The following table gives the number of iterations required for convergence when the tolerance is 10^{-6}. For the point-SOR method, optimal relaxation factor, obtained numerically to 4 significant digits, is used. The symbol $**$ signifies more than 1000 iterations. For the "Cosine" and "Neumann" preconditioners, since the problem is nonsymmetric, a generalized CG method, called the preconditioned conjugate gradient squared method is used, see [13].

$\alpha = 1$	$s = 1$			$s = 5$		
n	Cosine	Neumann	PSOR	Cosine	Neumann	PSOR
8	6	7	210	7	7	70
16	8	9	512	10	9	196
32	9	11	**	10	12	533
64	10	14	**	12	14	**
128	10	14	**	13	16	**
$\alpha = 2$	$s = 1$			$s = 5$		
n	Cosine	Neumann	PSOR	Cosine	Neumann	PSOR
8	6	7	113	7	7	60
16	8	9	262	10	9	182
32	9	11	581	10	12	476
64	11	14	**	12	14	**
128	10	17	**	13	16	**

Table 5: Number of Iterations for Convergence.

References

[1] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic Press, Orlando, Fl., 1983.

[2] Bini and Benedetto, *A New Preconditioner for The Parallel Solution of Positive Definite Toeplitz Systems*, In Proc. 2nd Annual SPAA, Crete, Greece, ACM Press, 220–223.

[3] R. Chan, *Iterative Methods for Overflow Queueing Models I*, Numer. Math., 51 (1987), 143–180.

[4] R. Chan and T. Chan, *Circulant Preconditioners for Elliptic Problems*, J. Numer. Linear Algebra Appls., 1 (1992), 77–101.

[5] R. Chan, K. Ng, and C. Wong, *Sine Transform Based Preconditioners for Symmetric Toeplitz Systems*, Lin. Algebra Appl., to appear.

[6] R. Chan and C. Wong, *Sine Transform Based Preconditioners for Elliptic Problems*, submitted.

[7] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Statist. Comput., V9 (1988), 766–771.

[8] T. Chan and J. Olkin, *Circulant Preconditioners for Toeplitz-block Matrices*, Numer. Algorithms, to appear.

[9] P. Concus, G. Golub, and G. Meurant, *Block Preconditioning for the Conjugate Gradient Method,* SIAM J. Sci. Statist. Comput., 6 (1985) 220–252.

[10] T. Dupont, R. Kendall and H. Rachford, *An Approximate Factorization Procedure for Solving Self-Adjoint Elliptic Difference Equations,* SIAM J. Numer. Anal., 5 (1968), 559–573.

[11] L. Kaufman, *Matrix Methods for Queueing Problems*, SIAM J. Sci. Statist. Comput., 4 (1982), 525–552.

[12] T. Manteuffel and S. Parter, *Preconditioning and Boundary Conditions*, SIAM J. Numer. Anal., 27 (1990), pp. 656–694.

[13] P. Sorenson, *A Fast Lanczos-type Solver for Non-symmetric Linear Systems,* SIAM J. Sci. Comput., 10 (1989), 36–52.