Abacus (Mathematics Science Series) Vol. 44, No 1, Aug. 2019

NUMERICAL EXPERIMENTS ON TOEPLITZ SYSTEMS OF LINEAR EQUATIONS USING THE PRECONDITIONED CONJUGATE GRADIENT METHOD

by

[1]C. C. Jibunoh (Jnr) and [2]M.N.O Ikhile

[1]Department of Mathematics, University of Benin, Benin City, Nigeria
Email: chike.jibunoh.edu@gmail.com
[2]Department of Mathematics, University of Benin, Benin City, Nigeria
Email: *mnoikhilo@yahoo.com*

Abstract

In this paper, we consider the conjugate gradient and the preconditioned conjugate gradient algorithms for solving Symmetric Positive Definite (SPD) Toeplitz systems. From these we develop what is defined as the C_i preconditioners which are modifications of the Ku and Kuo preconditioners. The C_i preconditioners are designed to speed up the iterative process in the earilier algorithms so as to obtain faster convergence. These C_i preconditioners also work satisfactorily for both the well and ill-conditioned Toeplitz systems and in many cases demonstrate superiority over the Strang and T.Chan's preconditioners.

Keywords: *Symmetric Positive Definite (SPD), Conjugate Gradient (CG), Preconditioned Conjugate Gradient (PCG), Krylov space, Matlab codes.*

1. Introduction

The name Toeplitz is in the memory of Otto Toeplitz's early work (Toeplitz, 1911) on bilinear forms related to Laurent series. Toeplitz matrices are matrices with constant diagonals. They often appear in the application of image processing, solutions of ordinary and partial differential systems of equations, numerical solutions of convolution-type integral equations, stationary autoregressive time series, signal processing and stochastic automata and neutral networks.

An n x n Toeplitz matrix is of the form; see (Chan and Jin, 2007),

$$T_n = \begin{bmatrix} t_0 & t_1 & \cdot\cdot & t_{n-2} & t_{n-1} \\ t_1 & t_0 & t_1 & \cdot & t_{n-2} \\ \vdots & t_1 & t_0 & \cdot & \cdot \\ t_{n-2} & \cdot & \cdot & \cdot & t_1 \\ t_{n-1} & t_{n-2} & \cdot & t_1 & t_0 \end{bmatrix} \qquad (1.1)$$

where $t_{ij} = t_{i-j}$ and T_n is constant along its diagonals. We are interested in solving the Toeplitz system $T_n x = b$, where b is a known vector and x is the unknown vector.

Toeplitz solvers are fast algorithm for solving Toeplitz systems. Most of the early works on Toeplitz solvers were based on direct methods. Direct methods based on Gaussian elimination method and Levinson recursion formula results in an algorithm of $O(n^3)$ and $O(n^2)$ complexity respectively. However, since the matrix is determined by only (2n-1) entries rather than n^2 entries, it is expected that a solution can be obtained in less than

$O(n^3)$ operations. As a result of this, iterative methods were developed and the conjugate gradient (CG) is one of the most popularly known iterative methods.
(Hestenes and Stiefel, 1952), developed one of the most powerful iterative methods for solving rapidly large linear systems of equations with symmetric positive definite (SPD) coefficient matrices i.e. the conjugate gradient method or simply CG algorithm.
(Strang, 1986) and (Olkin, 1986), proposed independently the use of the preconditioned conjugate gradient method (PCG) with circulant matrices as preconditioners to solve SPD Toeplitz systems. One of the main results of this iterative solver is that the complexity of solving a large class of n x n Toeplitz systems $T_n x = b$ is only of $O(n\log n)$ operations which is less than $O(n^2)$ and $O(n^3)$ operations of the direct methods.

In this paper, we shall consider the conjugate gradient and the preconditioned conjugate gradient algorithms and modify the Ku and Kuo's preconditioner, which we shall use to precondition the ill-conditioned SPD Toeplitz systems in order to speed up the convergence of the iterative process.

2. The Conjugate Gradient Method

A Krylov subspace iterative method, the CG, is one of the most important algorithm in solving large and sparse symmetric positive definite (SPD) matrices.
The rate of convergence of the method is in general good, but for ill-conditioned matrices it becomes problematic. In the following chapter we will show how CG can be improved for this kind of matrices.

The CG algorithm, applied to the system **A**x = **b**, starts with an initial guess of the solution x_0, with an initial residual r_0, and an initial search direction p_0 that is equal to initial residual $p_0 = r_0$.

The idea behind the conjugate gradient method is that the residual $r_k = b - Ax_k$ is orthogonal to the Krylov space generated by **b**, and therefore each residual is perpendicular to all the previous residuals. The residual is computed at each step and the solution at the next step is found using a search direction that is only a linear combination of the previous search directions, which for x_1 is just a combination between the previous and the current residual.

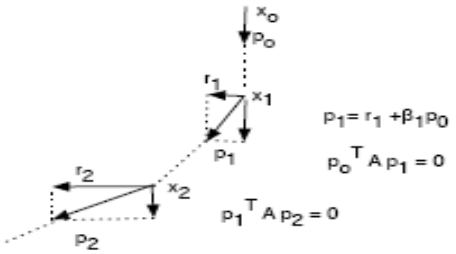

Figure 2.1: Searching Direction of the CG Algorithm. (ShewChuk, 1994)

Then, the solution x_k at step k, is just the previous iterate plus a constant times the last search direction. The immediate benefit of the search directions is that there is no need to store the previous search directions. And using the orthogonality of the residual to these previous search directions, the search is linearly independent of the previous directions. And for the solution at the next step, a new search direction is computed, as well as a new residual and new constant. The role of the constants is to give an optimal approximate solution as stated in (ShewChuk, 1994) and (Asby, 1996).

A more visual explanation of how the CG algorithm finds the approximate solution to the exact solution is given in Fig. 2.1.

The iterative formulas of CG as in (Trefethen and Bau, 1997) are given below:

— Approximate solution: $\mathrm{x}_{\mathrm{k}} = \mathrm{x}_{\mathrm{k-1}} + \alpha_{\mathrm{k}} \mathrm{p}_{\mathrm{k-1}}$

— Residual: $\mathrm{r}_{\mathrm{k}} = \mathrm{r}_{\mathrm{k-1}} - \alpha_{\mathrm{k}} \mathrm{Ap}_{\mathrm{k-1}}$

— Search direction: $\mathrm{p}_{\mathrm{k}} = \mathrm{r}_{\mathrm{k}} + \beta_{\mathrm{k}} \mathrm{p}_{\mathrm{k-1}}$

— Improvement at step k: $\beta_{\mathrm{k}} = \dfrac{\mathrm{r}_{\mathrm{k}}^{\mathrm{T}} \mathrm{r}_{\mathrm{k}}}{\mathrm{r}_{\mathrm{k-1}}^{\mathrm{T}} \mathrm{r}_{\mathrm{k-1}}}$

— Step length: $\alpha_{\mathrm{k}} = \dfrac{\mathrm{r}_{\mathrm{k-1}}^{\mathrm{T}} \mathrm{r}_{\mathrm{k-1}}}{\mathrm{p}_{\mathrm{k-1}}^{\mathrm{T}} \mathrm{Ap}_{\mathrm{k-1}}}$

2.1 The Rate of Convergence of the CG algorithm

When choosing an iterative method to solve a specific problem, it is important to take into consideration, the rate at which the method will converge to the exact solution. The fewer the iterations are needed to reach the solution, the higher the rate of convergence of the method.
An indicator of the difficulty of solving Ax = b with the conjugate gradient method is called the condition number of A. The condition number is defined as the product of the norm of A times the norm of the inverse of A: i.e,

$$\kappa(A) = \|A\| \left\|A^{-1}\right\|$$

Assuming that we know only the 2-norm condition number of a symmetric positive definite matrix, $\kappa(A) = \|A\|_2 \left\|A^{-1}\right\|_2$ we apply the CG to the matrix problem Ax = b and find that the A-norms of the errors satisfy the relationship:

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \frac{2}{\left[\left(\frac{\sqrt{k}+1}{\sqrt{k}-1}\right)^k + \left(\frac{\sqrt{k}+1}{\sqrt{k}-1}\right)^{-1}\right]} \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k \qquad (2.1)$$

If x_k is the k^{th} output of the CG, the above relationship is equivalent to

$$\|x - x_k\|_A \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k \|x - x_0\|_A \qquad (2.2)$$

It implies that

$$\|e_k\|_A = \min_{p \in p_k} \|p(A)e_0\|_A \leq \min_{p \in p_k} \max_{1 \leq i \leq n} |p(\lambda_i)| \|e_0\|_A$$

Hence,

$\dfrac{\|e_k\|_A}{\|e_k\|_A} \leq \min_{p \in p_k, p(0)=1} \max_{1 \leq i \leq n} |p(\lambda_i)|$, where λ_i were the eigenvalues of the symmetric positive definite

matrix A. For more details see (Caraba, 2008). Therefore the relation (2.1) implies that the convergence in the A-norm will be very fast if the condition number is one.
This condition number plays an important role in iterative methods. The closer its value to one the better the convergence of that method.

3. The Preconditioned Conjugate Gradient Method

The conjugate gradient method works very well on matrices that are well conditioned (i.e condition number is not too large). However, in real applications, most matrices are ill-conditioned (i.e the condition number is large), reducing the efficiency of the method. In this section we will discuss how the conjugate gradient method can be preconditioned to solve ill-conditioned large Toeplitz symmetric positive definite systems.

3.1 The Concept of Preconditioning

Preconditioning is an important technique to develop an efficient conjugate gradient method solver for scientific computing (Benzi, 2202). This technique comes into play when we want to solve large sparse systems with very large condition number.
In this paper, we shall apply this same technique to ill-conditioned and well-conditioned large- sparse Toeplitz linear systems.
By (2.2), the larger the condition number of SPD matrix A, the slower the conjugate gradient method will converge.
To fix ideas, suppose that M is a symmetric positive definite matrix such that either the condition number of $M^{-1}A$ is close to 1 or the eigenvalues of $M^{-1}A$ are clustered around 1. Then, by (2.13), the CG method, when applied to the preconditioned system
$M^{-1}Ax = M^{-1}b$,
will converge very fast. We will call M the preconditioner of the system Ax = b or of the matrix A.
The preconditioned conjugate gradient method takes the following form:

Algorithm 3.2

— $r_0 = b - Ax_0$
— $z_0 = M^{-1}r_0$
— $P_0 = z_0$
repeat
— $S_k = AP_k$
— $\alpha_k = \frac{r_k^T z_k}{P_k^T S_k}$
— $x_{k+1} = x_k + \alpha_k P_k$
— $r_{k+1} = r_k - \alpha_k S_k$
— If $\|r_{k+1}\| < 10^{-L};\ L >> 1$ then exit loop otherwise
— $z_{k+1} = M^{-1}r_{k+1}$
— $\beta_k = \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$
— $P_{k+1} = z_{k+1} + \beta_k P_k$
— k = k+1
end repeat

The result is x_{k+1}.

In algorithm 3.2, r_k denote the residual at the k^{th} step:

$r_k = b - Ax_k$ (3.1)

z_k is a vector for k^{th} step and P_k is the direction vector. We choose a Toeplitz matrix A which is positive definite and symmetric. The point x_0 is an initial guess or zero vector and x_{k+1} is the 'improved' approximate solution of the system. The matrix M^{-1} is the inverse of a matrix M which is the precondition for the system Ax = b.

The criteria for choosing a preconditioner M are:

i. M should be constructed within O(nlogn) operations
ii. Mv = y should be solved in O(nlogn) operations for any vector y.
iii. The spectrum of $M^{-1}A$ should be clustered and/or the condition number k of the preconditioned matrix should be close to 1.

The first two criteria are to keep the operation count per iteration within O(nlogn), as it is the count for the non-preconditioned system. The third criterion comes from the fact that the more well-conditioned or the more $M^{-1}A$ is clustered around the eigenvalues, the faster the convergence of the method will be.

3.2 Toeplitz Preconditioners

A Toeplitz preconditioner was proposed by Strang as in (Strang, 1986), and analyzed by (Chan and Strang, 1989). Strang preconditioner S is obtained by preserving the central half diagonals of A and using them to form a circulant matrix. Since S is circulant, the matrix-vector product $S^{-1}v$ can be conveniently computed via Fast Fourier Transform (FFT) with O(nlogn) operations. It has been shown in (Chan, 1989) – (Chan and Strang, 1989), that for a large class of matrices (called the Wiener Class), the spectrum of $S^{-1}A$ is clustered around 1 except a finite number of outliers.

In constructing Strang's preconditioner S, only half the elements of A is used. In order to use all elements of A, (Chan, 1988) proposed another Toeplitz preconditioner C. It is, by definition, the circulant matrix which minimizes the Frobenius norm $\|R - A\|_F$ over all circulant matrices R. This turns out to be a simple optimization problem, for which a closed-form solution exists. The elements of C can be computed directly from the elements of A by a simple formula. However, T. Chans's preconditioner C does not necessarily improve the convergence performance of the PCG method in comparism with Strang's preconditioner S.

Hence (Ku and Kuo, 1992) were motivated into seeking another direction to generalize Strang's preconditioner so that all elements of A can be effectively used. This led to a general approach for constructing Toeplitz preconditioners. They constructed a set of preconditioners called the k_i preconditioner, for solving a symmetric positive definite system of equations. These preconditioners k_i, i = 1, 2, 3, 4, use all elements of A.

3.3 The Construction of Toeplitz Preconditioners

Let A be an N x N SPD Toeplitz matrix, and $T_{N,1}$ be an N x N symmetric Toeplitz matrix approximating A. For example, we can choose $T_{N,1}$ = A or $T_{N,1}$ which minimizes the difference $T_{N,1} - A$ with respect to a certain norm. We define a 2N x 2N symmetric circulant matrix as

$$R_{2N} = \begin{bmatrix} T_{N,1} & T_{N,2} \\ T_{N,2} & T_{N,1} \end{bmatrix} \quad (3.2)$$

where

$$T_{N,1}=\begin{bmatrix} t_0 & t_1 & . & t_{n-2} & t_{n-1} \\ t_1 & t_0 & t_1 & . & t_{n-2} \\ . & t_1 & t_0 & . & . \\ t_{n-2} & . & . & . & t_1 \\ t_{n-1} & t_{n-2} & . & t_1 & t_0 \end{bmatrix} \quad (3.3)$$

$$T_{N,2}=\begin{bmatrix} c & t_{n-1} & . & t_2 & t_1 \\ t_{n-1} & c & t_{n-1} & . & t_2 \\ . & t_{n-1} & c & . & . \\ t_2 & . & . & . & t_{n-1} \\ t_1 & t_2 & . & t_{n-1} & c \end{bmatrix}$$ and where $T_{N,2}$ is determined by elements of $T_{N,1}$ as, (3.4)

with a constant c. Now, let us consider the following augmented system:

$$\begin{bmatrix} T_{N,1} & T_{N,2} \\ T_{N,2} & T_{N,1} \end{bmatrix}\begin{bmatrix} x \\ x \end{bmatrix}=\begin{bmatrix} b \\ b \end{bmatrix} \quad (3.5)$$

This equation (3.5) can be embedded by a circular convolution between two 2N-periodic sequences, whose periods are

$$t_0, t_1, ..., t_{N-2}, t_{N-1}, c, t_{N-1}, t_{N-2}, ..., t_1 \quad (3.6)$$

and

$$x_1, x_2, ...x_{N-2}, x_N, x_1, x_2, ..., x_{N-1}, x_N \quad (3.7)$$

The output sequence is also 2N-periodic, whose period is

$$b_1, b_2, ..., b_{N-2}, b_N, b_1, b_2, ..., b_{N-1}, b_N \quad (3.8)$$

The solution of (3.5) for x corresponds to a circular de-convolution problem and can be computed via FFT with O(NlogN) operations. Since (3.8) is equivalent to $(T_{N,1}+T_{N,2})x = b$ we can compute $(T_{N,1}+T_{N,2})^{-1}b$ efficiently and use

$$M_1 = T_{N,1} + T_{N,2}, \quad (3.9)$$

as a preconditioner for A. Various preconditioners are constructed in a similar way by assuming different periodicities for x and b, such as negative periodicity, even periodicity and odd periodicity. The corresponding augmented systems and preconditioners can be written as follows:

$$\begin{bmatrix} T_{N,1} & T_{N,2} \\ T_{N,2} & T_{N,1} \end{bmatrix}\begin{bmatrix} x \\ -x \end{bmatrix}=\begin{bmatrix} b \\ -b \end{bmatrix}, \quad \text{hence}$$

$$M_2 = T_{N,1} - T_{N,2} \quad (3.10)$$

Here

$$\begin{bmatrix} T_{N,1} & T_{N,2} \\ T_{N,2} & T_{N,1} \end{bmatrix}\begin{bmatrix} x \\ jx \end{bmatrix}=\begin{bmatrix} b \\ jb \end{bmatrix}, \quad \text{and}$$

$$M_3 = T_{N,1} + jT_{N,2} \quad (3.11)$$

$$\begin{bmatrix} T_{N,1} & T_{N,2} \\ T_{N,2} & T_{N,1} \end{bmatrix} \begin{bmatrix} x \\ -jx \end{bmatrix} = \begin{bmatrix} b \\ -jb \end{bmatrix} \quad \text{and}$$

$$M_4 = T_{N,1} - jT_{N,2} \tag{3.12}$$

where j is the N x N symmetric elementary matrix which has, by definition, ones along the secondary diagonal and zeros elsewhere (equivalently, $j_{ij} = 1, \text{if } i + j = N + 1$ and $j_{ij} = 0$ if $i + j \neq N + 1$), i.e. of the form

$$j = \begin{bmatrix} 0 & 0 & \dots & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & 1 & 0 & \dots \\ 1 & \dots & 0 & 0 \end{bmatrix} \tag{3.13}$$

3.4 Development of the C_i Preconditioners

To choose the appropriate constant c, in (3.4) several factors should be considered. Since the convergence rate of the PCG method depends on the eigenvalue distribution of the preconditioned matrix $M^{-1}A$, and according to (Ku and Kuo, 1992), if λ is an eigenvalue of the preconditioner $M_i, i = 1, 2, 3, 4$, it is also an eigenvalue of the matrix R_{2N}. Therefore to guarantee the positive definiteness of M_i, we require that

$$\sum_{n=-(N-1)}^{N-1} t_n e^{-\frac{i\pi kn}{N}} + (-1)^k \mathrm{c} > 0 \tag{3.14}$$

Since we want the norm of the matrix A to be as small as possible, c should be a small number. For sufficiently large N, we can adopt the simple rule, namely, if the behavior of the sequence $\{t_n\}$ is known, then c = t_N. Otherwise, c = 0.

Consider the k_i preconditioner where c = 0 rather than c = 1 as in (Ku and Kuo, 1992), since we have said previously that c should be a small number. We shall refer to the resultant preconditioner as C_i.

For example

$$A = \begin{bmatrix} 42 & 30 & 20 & 10 \\ 30 & 42 & 30 & 20 \\ 20 & 30 & 42 & 30 \\ 10 & 20 & 30 & 42 \end{bmatrix}$$

will be used to illustrate the construction procedure given in section 3.3

We use (3.9)-(3.12) to construct the preconditioners C_i. Although there exist many choices to select from, we see that $T_{N,1} = A$ seems trivial. For this choice, all elements of A are used in a straightforward way, and we call the resulting preconditioners C_i. Letting c = 0 in (3.4) we obtain:

$$T_{N,1} = \begin{bmatrix} 42 & 30 & 20 & 10 \\ 30 & 42 & 30 & 20 \\ 20 & 30 & 42 & 30 \\ 10 & 20 & 30 & 42 \end{bmatrix} \qquad T_{N,2} = \begin{bmatrix} 0 & 10 & 20 & 30 \\ 10 & 0 & 10 & 20 \\ 20 & 10 & 0 & 10 \\ 30 & 20 & 10 & 0 \end{bmatrix}$$

$$\mathrm{C}_1 = T_{N,1} + T_{N,2} = \begin{bmatrix} 42 & 40 & 30 & 40 \\ 40 & 42 & 40 & 30 \\ 30 & 40 & 42 & 40 \\ 40 & 30 & 40 & 42 \end{bmatrix} \tag{3.15}$$

$$C_2 = T_{N,1} - T_{N,2} = \begin{bmatrix} 42 & 20 & 0 & -20 \\ 20 & 42 & 40 & 0 \\ 0 & 20 & 42 & 40 \\ -20 & 0 & 20 & 42 \end{bmatrix} \tag{3.16}$$

$$C_3 = T_{N,1} + JT_{N,2} = \begin{bmatrix} 42 & 30 & 20 & 10 \\ 30 & 42 & 30 & 20 \\ 20 & 30 & 42 & 30 \\ 10 & 20 & 30 & 42 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times T_{N,2} = \begin{bmatrix} 72 & 50 & 30 & 10 \\ 50 & 52 & 30 & 30 \\ 30 & 30 & 52 & 50 \\ 10 & 30 & 50 & 72 \end{bmatrix} \tag{3.18}$$

$$C_4 = T_{N,1} - JT_{N,2} = \begin{bmatrix} 42 & 30 & 20 & 10 \\ 30 & 42 & 30 & 20 \\ 20 & 30 & 42 & 30 \\ 10 & 20 & 30 & 42 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \mathrm{T}_{N,2} = \begin{bmatrix} 12 & 10 & 10 & 10 \\ 10 & 32 & 30 & 10 \\ 10 & 30 & 32 & 10 \\ 10 & 10 & 10 & 12 \end{bmatrix} \tag{3.19}$$

4. Numerical Experiments

In this section the Strang's preconditioner S(A), Chan's preconditioner T(A) , the C_i preconditioners constructed in section 3 will be applied to a well and ill-conditioned symmetric Positive definite (SPD) Toeplitz system and compare their convergence rate with that of the unpreconditioned (I) system.

We will illustrate these convergence rate on a SPD Toeplitz system of size n = 32, 64, 128, 256, 512, generated from different functions, solved in (Chan and Jin, 2007), where the right hand side vector b, is a vector of all ones, i.e b = (1,1, . . . , 1)

All computational results and experiments presented in this work have been done using Matlab 7.7.0 (2008b).

We represent the preconditioners which reached maximum iteration without converging to a solution with "†" and "—" for those ones that stagnated.

Then, we compare the unpreconditioned system, and the PCG using the preconditioners mentioned above. The results are presented in Tables 4.1 – 4.2 for different sizes of n mentioned here.

Table 4.1.

Number of iterations for well-conditioned SPD Toeplitz systems.

P / ***n***	*f(x) = x^4 + 1*					*f(x) = $\|x\|^3$ + 0.01*				
	32	64	128	256	512	32	64	128	256	512
I	19	36	54	66	70	20	51	130	271	393

S(A)	7	6	5	5	4	9	9	8	7	5
T(A)	7	7	6	6	6	13	15	19	15	12
K_1	10	10	10	10	10	17	27	49	80	87
K_2	15	27	58	124	287	20	44	131	419	973
K_3	13	17	28	49	84	20	42	124	392	1234
K_4	13	22	43	89	196	18	36	104	407	1201
C_1	5	5	5	5	5	8	9	8	7	7
C_2	5	5	5	5	5	8	8	7	7	7
C_3	5	5	5	5	5	9	10	9	7	7
C_4	5	5	5	5	5	7	7	6	6	6

Table 4.2.

Number of iteration for ill-conditioned SPD Toeplitz systems

P / ***n***	$f(x) = x^2$					$f(x) = x^4$					$f(x) = x^4(\pi - x^2)$				
	32	64	128	256	512	32	64	128	256	512	32	64	128	256	512
I	16	37	82	176	370	31	118	508	†	†	18	66	233	951	†
S(A)	6	6	6	7	7	15	18	25	61	—	12	15	20	32	—
T(A)	10	12	14	17	22	16	26	43	—	—	14	21	33	83	—
K_1	15	26	43	78	144	17	32	104	—	—	14	22	42	155	—
K_2	20	17	136	496	2628	21	64	292	—	—	13	27	94	436	—
K_3	19	46	156	735	†	21	57	311	—	—	15	28	82	517	†
K_4	18	38	118	662	†	18	54	300	—	—	12	24	79	—	†
C_1	7	7	7	7	7	12	16	29	67	—	11	15	20	27	—
C_2	7	7	7	7	7	11	17	21	32	—	11	15	24	49	—
C_3	6	8	8	8	8	12	19	30	61	—	12	17	24	39	—
C_4	6	6	6	6	6	11	15	21	41	—	10	14	19	33	—

Clearly, from Tables 4.1 and 4.2, we can see that the C_i, $i = 1(1)4$ preconditioners maintained a lesser number of iterations than the Strang's, T. Chans' and the Ku and Kuo preconditioners respectively.

5. Conclusion

In this paper, we presented the conjugate gradient and the preconditioned conjugate gradient algorithms for solving symmetric positive definite (SPD) Toeplitz systems. We considered the K_i preconditioners when c = 0, which we called C_i preconditioners. These preconditioners are applied to speed up the CG iterative process and we have also shown that it worked well for both the well and ill-conditioned Toeplitz systems. In the present cases it is superior to the Strang and T.Chan's preconditioners.

A good preconditioner must be constructed easily and the norm of M_i^{-1}A must be small. It must be able to reduce the condition number of the matrix A to a smaller condition number and also the condition number of $M^{-1}A$ should be close to one, in order to reduce the condition number of the matrix.

The C_i's preconditioner which is K_i when c = 0 in (3.4) satisfies all these requirements. The preconditioner improves the K_i preconditioners which may be evident from the result of the numerical experiments presented in Tables 4.1 and 4.2.

REFERENCES

Asby S., Barth T., Mantueffel T. and Saylor P. (1996). A tutorial on iterative methods for Linear Algebraic systems, University of Illinois.

Benzi M. Preconditioning Techniques for Large Linear Systems: A survey (2002). *Journal of Computational physics*. 182: pp.418-477.

Brenner S. C. and Sung L.Y. Numerical Linear Algebra, Math 526: Lecture Notes, *Department of Mathematics of South Carolina*, SC, 29208.

Caraba E. (2008) Preconditioned Conjugate Gradient Algorithm, undergraduate honors thesis, under the direction of Dr. Susanne Brenner and Dr. Li-Yeung Sung. *Department of Mathematics, Louisana State University*, pp.34.

Chan R.H. (1989). Circulant Preconditioners for Hermitian Toeplitz system. *SIAM J. Matrix Anal. Appl*., Vol. 10, pp. 542 – 550.

Chan.R.H and Jin X.Q. (2007). An Introduction to iterative Toeplitz Solvers., pp85 – 92.

Chan R.H. and Strang G. (1989). Toeplitz equations by Conjugate Gradients with circulant preconditioner. *SIAM J.Sci. Stat. Comput.*, Vol. 10, pp. 104 – 119.

Chan T.F. (1988). An Optimal Circulant Preconditioner for Toeplitz systems. *SIAM J.Sci. Stat. Comput.*, Vol. 9, pp.766 – 771.

Davis P. (1994). Circulant Matrices, 2nd edition, *Ams Chelsea Providence*, RI.

Golub G.H. and Van Loan C.F. (1996). Matrix Computations. 3rd ed., *The Johns Hopkins University Press*.

Hestenes M.R. and Stiefel E. (1952). Methods of Conjugate Gradient for solving linear systems. *Journal of Research of the National Bureau of standards*, Vol. 49, No. 6.

Ku T.K. and Kuo C.C.J. (1992). Design and Analysis of Toeplitz preconditioners. *Journal of IEEE transactions on signal processing*, Vol., 40.

Micheal K. NG. (2004). Iterative Methods for Toeplitz Systems. *Oxford University Press*, Oxford, UK.

ShewChuk J.R. (1994). *An* Introduction to the Conjugate Gradient Method without the Agonizing pain.

Strang G. (1986). A proposal for Toeplitz matrix calculations, *Stud. Appl.*, Vol., 74, pp.171 -176.

Trefethen L.N. and Bau D. (1997). Numerical Linear Algebra. *SIAM*, Philadephia, pp.89 – 97; 293-303, 313 – 321.

Toeplitz O. (1911). The Theory of Quadratic and Bilinear Forms of Infinitely Variable Part 1: Theory of L- Forms, *Math. Ann.* Vol. 70 pp. 351-376.

Olkin J. (1986). Linear and non-linear Deconvolution problems. *Ph.D Thesis, Rice University, Houston*, TX.