Preconditioned Conjugate Gradient Method with Shifted Multi-Step Circulant and Skew-Circulant Splitting Preconditioner

Yi Yin, Nian-Ci Wu, and Chengzhi Liu

***Abstract*—In this paper, we are concerned with the solution of Hermitian positive definite Toeplitz linear systems by employing the preconditioned conjugate gradient (CG) method. The shifted multi-step circulant and skew-circulant splitting (CSCS) preconditioner P_m is designed based on the shifted classical CSCS iterative methods. We show that if the CSCS is P-regular, the spectrum of the preconditioned matrix is clustered around 1 for moderate m. Theoretical and experimental results show that our preconditioner can accelerate the convergence of CG for solving large Hermitian positive definite Toeplitz linear systems.**

***Index Terms*—Hermitian Toeplitz matrices, CSCS iteration, P-regular, the shifted multi-step CSCS preconditioner, CG.**

I. Introduction

CONSIDER the solution to a large linear system of equations

$$T\mathbf{x} = \mathbf{b}, \tag{1}$$

where the coefficient matrix $T = [t_{kj}]_{k,j=1}^{n} \in \mathbb{C}^{n\times n}$ is a Hermitian positive definite Toeplitz matrix with $t_{kj} = t_{k-j}$ and the right-hand side $\mathbf{b} \in \mathbb{C}^n$. The Toeplitz system in Equation (1) has gained mainstream interest in many areas such as [1], [2], [3], [4], [5].

Recently, there has been a lot of literature about constructing, analyzing, and implementing various direct and iterative methods to solve (1). For example, Krylov subspace based techniques such as conjugate gradient (CG) for Hermitian positive definite Toeplitz systems [1]; the Hermitian and skew-Hermitian splitting (HSS) method and the accelerated HSS iterative method for non-Hermitian positive definite Toeplitz systems [6], [7]; the superfast structured Toeplitz solutions for general Toeplitz systems [4], [5]; the circulant and skew-circulant splitting (CSCS) iterative method for positive definite Toeplitz systems [8], [9]; the accelerated CSCS [10] and shifted CSCS iterative methods [11] for Hermitian Toeplitz systems; the augmented Kaczmarz algorithm for inconsistent systems [12]; and the local circulant and residue splitting iterative method for Toeplitz-structured saddle point problems [13]. Despite the fact that the iterative solver is a preferred choice for solving large sparse linear systems, its convergence rate may be very slow or even divergent.

As stated in [14], iterative methods combined with preconditioning techniques could significantly speed up the convergence rate. Especially, a careful and problem-dependent choice of preconditioner can make the Krylov subspace iterative methods more reliable. The main task in preconditioning is to design a preconditioner P that is as close as possible to the coefficient matrix T. Thus, we can replace the system (1) with the preconditioned system $P^{-1}T\mathbf{x} = P^{-1}\mathbf{b}$. Very often, the preconditioned matrix $P^{-1}T$ is well conditioned or has a clustered spectrum, and P^{-1} can be obtained easily. Research on preconditioning technique has been a continuous hot spot in numerical algebra. Many preconditioners have been proposed based on the special structure of the coefficient matrix T such as [15], [16], [17], [18]. For more details about this research area, we refer the reader to read the surveys [14], [19], [20], [21].

Of all the preconditioning techniques, the polynomial preconditioner in [22] is of popular and interesting, which has fueled new research such as [23], [24], [25]. The primary challenge in constructing polynomial preconditioners lies in the efficient splitting of the coefficient matrix and the selection of an appropriate iterative solver.

By utilizing the shifted CSCS iterative method introduced in [11], we propose a shifted multi-step CSCS preconditioner and employ the preconditioned conjugate gradient (PCG) method to solve (1). This approach is hereafter abbreviated as CSCS(m)-CG, where m is the order of the polynomial. This proposed method profits from the fact that the circulant matrices and skew-circulant matrices can be diagonalized by the discrete Fourier matrices. Therefore, any n-vector multiplication with the Toeplitz matrix T can be computed with $\mathcal{O}(n\log n)$ arithmetic operations by using the fast Fourier transform (FFT). Under certain assumptions about the generation function of T, we will show that the eigenvalues of the preconditioned matrix are clustered around 1 for moderate m. Hence we can expect that the CSCS(m)-CG method has a good performance in solving (1).

The remainder of the paper is organized as follows. After reviewing the shifted CSCS iterative method in Section II, we introduce the shifted multi-step CSCS preconditioners and give efficient implementations of the CSCS(m)-CG method in Section III. In Section IV, we establish the estimators of the lower and upper bounds of the spectrum of the preconditioned matrix. Numerical experiments are given to confirm the effectiveness of our preconditioners in Section V. Finally, we end this work with some conclusions in Section VI.

Manuscript received November 4, 2024; revised February 10, 2025.

This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2023JJ50080.

Y. Yin is a lecturer of Department of Basic Education, Hunan College of Information, Changsha 410200, P. R. China (e-mail: yin51348878yi@163.com).

N.-C. Wu is a lecturer of School of Mathematics and Statistics, South-Central Minzu University, Wuhan 430074, P. R. China (e-mail: nianciwu@scuec.edu.cn).

C. Z. Liu is an associate professor of School of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi 417000, P. R. China (corresponding author to provide e-mail: it-rocket@163.com).

II. The shifted CSCS iteration

Assume that the generating function of T in (1) is defined by $f(x) = \sum_{-\infty}^{+\infty} t_k e^{-\mathrm{i}kx}$ with $\mathrm{i} \equiv \sqrt{-1}$ and $x \in [-\pi, \pi]$ for $k = 0, \pm 1, \cdots, \pm(n-1)$. It is verified that $t_{-k} = \mathsf{conj}(t_k)$[1] for all integers and T is a Hermitian matrix, see [9]. After that we can split T according to

$$T = C_\beta - S_\gamma, \tag{2}$$

where $C_\beta = Circ(c_0, \cdots, c_{n-1})$ is a Hermitian circulant matrix and $S_\gamma = Skewcirc(s_0, \cdots, s_{n-1})$ is a Hermitian skew circulant matrix with $c_0 = \beta$, $s_0 = \gamma$, $c_k = (t_{n-k} + t_{-k})/2$ and $s_k = (t_{n-k} - t_{-k})/2$ for $k = 1, \cdots, n-1$. For more details, we refer to [9], [10], [11]. As described in [10, Theorem 2.1], if $f(x) = \sum_{-\infty}^{+\infty} t_k e^{-\mathrm{i}kx}$ is in the Wiener class where $\sum_{-\infty}^{+\infty} |t_k| < \infty$ and satisfies the condition $f(x) \geq \delta > 0$ for any x, we know that the circulant matrix C_{t_0} is uniformly positive and bounded for sufficiently large n. It implies that the assumption that C_β is positive definite is reasonable.

Due to the promising performance and elegant mathematical properties of CSCS in (2), Liu et al proposed the shifted CSCS iterative method for solving the Toeplitz systems in (1) [11]. The algorithm framework is detailed below.

The shifted CSCS iterative method: Given an initial guess $\mathbf{x}^{(0)}$, until iterative sequence $\{\mathbf{x}^{(k)}\}$ converges, compute

$$\hat{C}_\beta \mathbf{x}^{(k+1)} = \hat{S}_\gamma \mathbf{x}^{(k)} + \mathbf{b}, \quad k = 0, 1, 2, \cdots \tag{3}$$

where $\hat{C}_\beta = \alpha I + C_\beta$, $\hat{S}_\gamma = \alpha I + S_\gamma$, and α is the shift parameter.

It is shown in [11] that for $j = 1, \cdots, n$, there exists a shift α such that $\lambda_j(\hat{C}_\beta^H + \hat{S}_\gamma) = 2\alpha + \lambda_j(C_\beta + S_\gamma) > 0$, where $\lambda_j(M)$ means the jth eigenvalue of the matrix M. By Hermann Wey's theorem, i.e., Theorem 4.3.1 in [26], it is verified that $\hat{C}_\beta^H + \hat{S}_\gamma$ is positive definite. Thus $T = \hat{C}_\beta - \hat{S}_\gamma$ is a P-regular splitting and the spectral radius of iterative matrix $\hat{C}_\beta^{-1}\hat{S}_\gamma$ will be less than 1 when T is Hermitian positive definite, see [27, Theorem 1.2].

III. The proposed method

A. Definition of preconditioners

Unless otherwise specified, we take $\beta = t_0$ and $\gamma = 0$ to make the circulant part dominant. For simplicity, we denote $\hat{C}_{t_0}$ and $\hat{S}_0$ as C and S, respectively. Since $T = C - S$ is a P-regular splitting, the inverse of T can be written as

$$T^{-1} = (I - C^{-1}S)^{-1}C^{-1}.$$

Replacing the first factor of the right-hand term above with its Taylor expansion, we have

$$\begin{aligned} T^{-1} &= (I-G)^{-1}C^{-1} = (I + G + G^2 + \cdots)C^{-1} \\ &= (I + G + G^2 + \cdots + G^{m-1})C^{-1} \\ &\quad + G^m(I-G)^{-1}C^{-1} \\ &= p_{m-1}(G)C^{-1} + G^m T^{-1}, \end{aligned} \tag{4}$$

[1]The operation $\mathsf{conj}(\mathbf{v})$ means the conjugate transpose of vector $\mathbf{v}$.

where $G = C^{-1}S$, $p_{m-1}(G) = \sum_{k=0}^{m-1} G^k$. If the term $G^m T^{-1}$ in (4) is of small norm, the matrix

$$P_m^{-1} = p_{m-1}(G)C^{-1} \tag{5}$$

is close to T^{-1} and the matrix P_m serves as an effective preconditioner for (1). Therefore we can replace the system (1) with the preconditioned system

$$P_m^{-1}T\mathbf{x} = P_m^{-1}\mathbf{b},$$

which can be solved by employing the state of the art iterative method such as the CG method. Accordingly, we refer to the matrix P_m as the shifted multi-step CSCS preconditioner.

Remark 1: If the skew-circulant part S dominates, then we can use an alternative preconditioning strategy. Such a polynomial preconditioner can also be derived from multi-splitting methods or the two-stage iteration methods of trivial outer splittings such as discussed in [24], [25], [28], [29].

Theorem 1: Let T be a Hermitian positive definite matrix. The shifted multi-step CSCS preconditioner P_m is a Hermitian positive definite matrix.

Proof: Since C and S are Hermitian, $G^j C^{-1}$ is Hermitian for all $j = 0, 1, \ldots, m-1$. It follows immediately that P_m^{-1} is Hermitian according to (5). According to

$$(I - T^{1/2}C^{-1}T^{1/2})^H = I - T^{1/2}C^{-1}T^{1/2},$$

we have

$$G = I - C^{-1}T = T^{-1/2}(I - T^{1/2}C^{-1}T^{1/2})T^{1/2},$$

which implies that all eigenvalues of G are real. In addition,

$$T^{1/2}P_m^{-1}T^{1/2} = T^{1/2}(P_m^{-1}T)T^{-1/2} = T^{1/2}(I - G^m)T^{-1/2},$$

and $\lambda_\ell(I - G^m) > 0$ for $\ell = 1, 2, \cdots, n$ since the spectral radius $\rho(G) = \rho(C^{-1}S) < 1$, we know that P_m^{-1} is positive definite and the result follows. ■

Remark 2: If the preconditioner P_m in (5) is Hermitian positive definite, then the corresponding multi-step preconditioner is also known as the validity multi-step preconditioner in [23].

B. Efficient implementations

When the CSCS(m)-CG method is employed to solve the linear system (1), the main computational cost is the matrix-vector product $\mathbf{y} = T\mathbf{p}$ and the calculation of $P_m^{-1}\mathbf{r}$, which is equivalent to solving the generalized residual equation $P_m\mathbf{z} = \mathbf{r}$, where $\mathbf{p}$, $\mathbf{r}$ and $\mathbf{z}$ are the search direction, the current residual and the generalized residual, respectively.

We first discuss the calculation of the matrix-vector product $T\mathbf{p}$. Since the circulant matrix C can be diagonalized by the discrete Fourier matrix F_n, and the skew-circulant matrix S can be diagonalized by the diagonal-scaled discrete Fourier matrix $\hat{F}_n$, i.e.,

$$C = F_n^* \Lambda F_n \quad \text{and} \quad S = \hat{F}_n^* \Sigma \hat{F}_n,$$

where Λ and Σ are diagonal matrices holding the eigenvalues of C and S, respectively, the discrete Fourier matrix $F_n = \frac{1}{\sqrt{n}}\left[e^{-\mathrm{i}2\pi kj/n}\right]_{k,j=0}^{n-1}$ and $\hat{F}_n = F_n\Omega$ with $\Omega = diag(\omega)$ and the n-vector $\omega^T = \left[(-1)^{-(j-1)/n}\right]_{j=0}^{n-1}$. For more details, we refer to Definition 2.1 in [1].

Note that the first column of F_n is given by $F_n(:,1) = \frac{1}{\sqrt{n}}\mathbf{e}$ with n-vector $\mathbf{e}$ being a vector with all entries equal to 1. It will take only $\mathcal{O}(n\log n)$ operations to compute all the eigenvalues of matrix C. The details are as follows. Let the vector $\boldsymbol{\lambda}$ hold the eigenvalues of C. We have

$$\boldsymbol{\lambda} = \Lambda\mathbf{e} = \sqrt{n}F_nC\mathbf{e}_1 = \mathtt{fft}(C(:,1)),$$

where $\mathbf{e}_1$ is the first column of an identity matrix with order n, and the operation $\mathtt{fft}(\mathbf{v})$ is a command that computes the product $\sqrt{n}F_n\mathbf{v}$ for any n-vector $\mathbf{v}$ in MATLAB.

Similarly, the vector $\boldsymbol{\sigma}$ is comprised of the eigenvalues of S and can be obtained using

$$\boldsymbol{\sigma} = \mathtt{fft}(\omega .* S(:,1)),$$

where the operation '.*' is the entrywise multiplication.

By embedding T into a $2n \times 2n$ circulant matrix, $\mathbf{y} = T\mathbf{p}$ can be computed by the FFT with $\mathcal{O}(n\log n)$ arithmetic operations. We briefly called as $\mathbf{y} = \mathtt{HToepvec}(\mathbf{t},\mathbf{p})$, where $\mathbf{t}$ the first column of Hermitian Toeplitz matrix T. Reader can refer to for more details about the fast calculation of $T\mathbf{p}$.

Next, we discuss the calculation of $P_m^{-1}\mathbf{r}$. From (5), we get $\mathbf{z} = P_m^{-1}\mathbf{r}$. It is worth specifying that P_m^{-1} does not need to be formed explicitly since $P_m^{-1}\mathbf{r}$ can be evaluated via a series of matrix-vector products, i.e., to solve the $P_m\mathbf{z} = \mathbf{r}$, we need to perform m steps

$$C\mathbf{z}^{(j)} = S\mathbf{z}^{(j-1)} + \mathbf{r}, \quad j = 1,\cdots,m,$$

of the shifted classical CSCS iteration in (3). Thus

$$\mathbf{z}^{(m)} = G^m\mathbf{z}^{(0)} + (I + G + G^2 + \cdots + G^{m-1})C^{-1}\mathbf{r}. \quad (6)$$

If we select $\mathbf{z}^{(0)} = \mathbf{0}$ in (6), then

$$\mathbf{z}^{(m)} = (I + G + G^2 + \cdots + G^{m-1})C^{-1}\mathbf{r} = \mathbf{z}. \quad (7)$$

The computation of $P_m^{-1}\mathbf{r}$ can be formulated as the following subroutine $\mathbf{z} = \mathsf{InvPvec}(\boldsymbol{\lambda},\boldsymbol{\sigma},\mathbf{r},m)$.

Subroutine $\mathbf{z} = \mathsf{InvPvec}(\boldsymbol{\lambda},\boldsymbol{\sigma},\mathbf{r},m)$

Input two eigenvalue vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$ of C and S, respectively, the current residual $\mathbf{r}$, the order m of P_m.

1: Compute $\mathbf{z}^{(0,1)} = \mathtt{fft}(\mathbf{r})./\boldsymbol{\lambda}$, $\mathbf{z}^{(0)} = \mathtt{ifft}(\mathbf{z}^{(0,1)})$.
2: For $j = 1 : m$
(a) Update $\mathbf{r}$:= $S\mathbf{z}^{(j-1)} + \mathbf{r}$ by
$\mathbf{s}_1 = \mathtt{fft}(\boldsymbol{\omega} .* \mathbf{z}^{(j-1)})$, $\mathbf{s}_2 = \boldsymbol{\sigma} .* \mathbf{s}_1$, $\mathbf{s} = conj(\boldsymbol{\omega}) .* \mathtt{ifft}(\mathbf{s}_2)$, $\mathbf{r} = \mathbf{s} + \mathbf{r}$.
(b) Solve the equation $C\mathbf{z}^{(j)} = \mathbf{r}$ by
$\mathbf{z}^{(j,1)} = \mathtt{fft}(\mathbf{r})./\boldsymbol{\lambda}$, $\mathbf{z}^{(j)} := \mathtt{ifft}(\mathbf{z}^{(j,1)})$.
3: EndFor
4: Set $\mathbf{z} := \mathbf{z}^{(m)}$ and output $\mathbf{z}$.

When employing the CSCS(m)-CG method to solve the Hermitian positive definite Toeplitz linear systems, it is quite natural to allow the usage of the aforementioned FFT. Therefore, the total arithmetic operations at each step of the CSCS(m)-CG method is $\mathcal{O}(n\log n)$.

The CSCS(m)-CG method.

Input the first column $\mathbf{t}$ of T, the right-hand side $\mathbf{b}$, the stopping tolerance TOL, the maximum number ℓ of iterations, two eigenvalue vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$ of C and S, respectively, the order m of P_m.

1: Call $\mathbf{y} = \mathrm{HToepvec}(\mathbf{t},\mathbf{x}^{(0)})$ and compute $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{y}$.
2: Call $\mathbf{z}^{(0)} = \mathrm{InvPvec}(\boldsymbol{\lambda},\boldsymbol{\sigma},\mathbf{r}^{(0)},m)$ and assign $\mathbf{p}^{(0)} = \mathbf{z}^{(0)}$.
3: For $k = 0, 1, \cdots, \ell$
If $\|\mathbf{r}^{(k)}\|_2/\|\mathbf{r}^{(0)}\|_2 >$ TOL
(a) Call $\mathbf{y} = \mathrm{HToepvec}(\mathbf{t},\mathbf{p}^{(k)})$ and compute $u_k = (\mathbf{r}^{(k)},\mathbf{z}^{(k)})/(\mathbf{y},\mathbf{p}^{(k)})$.
(b) Compute $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + u_k\mathbf{p}^{(k)}$ and $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - u_k\mathbf{y}$.
(c) Call $\mathbf{z}^{(k+1)} = \mathrm{InvPvec}(\boldsymbol{\lambda},\boldsymbol{\sigma},\mathbf{r}^{(k+1)},m)$.
(d) Compute $v_k = (\mathbf{r}^{(k+1)},\mathbf{z}^{(k+1)})/(\mathbf{r}^{(k)},\mathbf{z}^{(k)})$ and $\mathbf{p}^{(k+1)} = \mathbf{z}^{(k+1)} + v_k\mathbf{p}^{(k)}$.
Else goto 5.
4: EndFor
5: Set $\mathbf{x} := \mathbf{x}^{(k)}$ and output $\mathbf{x}$.

Remark 3: It is well known that FFT is highly parallelizable which has been efficiently implemented on multiprocessors. Moreover, the CG method is also easily parallelizable. Thus, the CSCS(m)-CG method is well-suited for parallel computing. For more references on parallel computing, see [21], [30].

IV. Convergence analysis

In this section, our main result is that the spectra of these preconditioned matrices $P_m^{-1}T$ are clustered around 1. Hence the CSCS(m)-CG method when applied to solve the linear systems (1) will converge sufficiently fast. Given an $n \times n$ matrix M, we denote the eigenvalues of M arranged in nonincreasing order, i.e., $\lambda_1(M) \le \lambda_2(M) \le \cdots \le \lambda_n(M)$.

Lemma 1: [26, Lemma 5.6.10] Let $A \in \mathbb{C}^{n\times n}$ and $\varepsilon > 0$ be given, there is a matrix norm $\|\cdot\|_\varepsilon$ such that $\rho(A) \le \|A\|_\varepsilon \le \rho(A) + \varepsilon$.

Theorem 2: If $T = C - S$ is the P-regular splitting and the preconditioner P_m is defined by (5), then we have the following bounds

$$1 - \|G\|_\varepsilon^m \le \lambda_j(\hat{T}) \le 1 + \|G\|_\varepsilon^m, \ j = 1,\cdots,n, \quad (8)$$

where the preconditioned matrix $\hat{T} = P_m^{-1}T$ and $\|\cdot\|_\varepsilon$ is the matrix norm in Lemma 1.

Proof: It suffices to show that $\lambda_1(\hat{T}) \ge 1 - \|G\|_\varepsilon^m$ and $\lambda_n(\hat{T}) \le 1 + \|G\|_\varepsilon^m$. According to the above hypothesis, we can get $\rho(G) < 1$ easily. From (5), some algebraic manipulations give

$$\hat{T} = p_{m-1}(G)(I - G) = I - G^m,$$

thus

$$\begin{aligned}\lambda_1(\hat{T}) &= \frac{1}{\rho(\hat{T}^{-1})} = \frac{1}{\rho([I-G^m]^{-1})} \\ &\ge \frac{1}{\|(I-G^m)^{-1}\|_\varepsilon} = \frac{1}{\|I + G^m + (G^m)^2 + \cdots\|_\varepsilon} \\ &\ge \frac{1}{1 + \|G^m\|_\varepsilon + \|G^m\|_\varepsilon^2 + \cdots} \\ &= 1 - \|G^m\|_\varepsilon \ge 1 - \|G\|_\varepsilon^m.\end{aligned}$$

The right inequality in (8) can be obtained by the submultiplicativity of matrix norm immediately. ■

Corollary 1: If the CSCS is a convergent splitting, then the eigenvalues of the preconditioned matrix $\hat{T} = P_m^{-1}T$ have the following bounds

$$1 - \|G\|_\varepsilon^m \leq \lambda_j(\hat{T}) \leq 1 + \|G\|_\varepsilon^m,\ j = 1, \cdots, n,$$

where $\|\cdot\|_\varepsilon$ is the matrix norm in Lemma 1.

Theorem 2 and Corollary 1 show that when the CSCS is P-regular or a convergent splitting, the eigenvalues of the preconditioned matrix $P_m^{-1}T$ will be located in a circle centered at (1, 0) with radius $\rho(G^m) = \rho(G)^m$, it can be easily deduced that the convergence rate of the CSCS(m)-CG method is linear, see for instance [2, Theorem 1.10]. Thus, the CSCS(m)-CG method will converge in a constant number of iterations and the total complexity of solving the Toeplitz system is $\mathcal{O}(n \log n)$.

According to (4) and (5), we quantify the deviation and obtain $\|T^{-1} - P_m^{-1}\|_\varepsilon = \|G^m T^{-1}\|_\varepsilon = O(\|G\|_\varepsilon^m)$. The preconditioning effect of P_m is partially determined by the spectral radius of the matrix $G = (\alpha I + C_\beta)^{-1}(\alpha I + S_\gamma)$. The determination of an optimal parameter $\alpha = \arg\min(\rho(G))$ is the key issue for improving computing efficiency. Unfortunately, it is difficult to determine α theoretically, which strongly depend on the specific structure and properties of the coefficient matrix T when solving the linear systems (1) by using the PCG method. Here, we will not touch this topic further. However, numerous numerical tests show that $\alpha \approx -[\lambda_1(C_\beta) + \lambda_1(S_\gamma)]/2$ is an appropriate choice, which is similar to the conclusion in [11].

Remark 4: We remark here that the selection of m will affect the performance of the preconditioner P_m. According to (7), the larger m is, the more costly in solving $P_m\mathbf{z} = \mathbf{r}$. On the other hand, since $\rho(G) < 1$, it follows that the larger m is, the closer the preconditioner P_m^{-1} is to T^{-1}, the more clustered around 1 the spectra of $P_m^{-1}T$ are, and the faster the convergence rate of the CSCS(m)-CG method will be. Therefore, we need to strike a balance between the computational complexity and convergence rate. To reduce the amount of calculation, it is not necessary for m to be too large in actual implementations. Experimentally, we found that $m = 3$ or 4 is a suitable choice.

V. Numerical examples

In this section, we illustrate the effectiveness of the proposed CSCS(m)-CG method for solving three kinds of Hermitian positive definite Toeplitz systems. All numerical tests were performed on a Founder desktop PC with Intel(R) Core(TM) i5-7500 CPU 3.40 GHz using MATLAB with machine precision 10^{-16}. The right-hand side $\mathbf{b} = \mathbf{e}$, the initial guess $\mathbf{x}^{(0)}$ is set to zero vector. The algorithm terminates when the current iteration satisfies the condition TOL$= ||\mathbf{r}^{(k)}||_2/||\mathbf{r}^{(0)}||_2 \leq 10^{-12}$, where $\mathbf{r}^{(k)}$ is the residual vector at the kth iteration.

Example 1: ([10], [18], [9]) Consider the Toeplitz matrix T defined by

$$t_k = \begin{cases} (1+\sqrt{-1})/(1+k)^p, & k > 0, \\ 2, & k = 0, \\ \mathsf{conj}(t_k), & k < 0. \end{cases} \tag{9}$$

Note that the sequence $\{t_k\}_{k=0}^{n-1}$ in Example 1 is absolutely summable when $p > 1$, and the generating function $f(x) = 2\sum_{k=0}^{+\infty}(\sin(kx) + \cos(kx))/(1+k)^p$ with $x \in [0,\ 2\pi]$ is continuous and belongs to the Wiener class. It can be shown that the Toeplitz matrix T is Hermitian positive definite. In our tests, we took $p = 1.1$, which is the same as in [18].

In Figure 1, we display the spectral distributions of the preconditioned matrices $P_m^{-1}T$ in Example 1 with different m and α when $n = 2000$ (left), $n = 4000$ (middle), and $n = 6000$ (right), respectively. It is evident from Figure 1 that the spectral distributions of the preconditioned matrices become increasingly clustered around 1 as m increases. These results are entirely consistent with the previous description in Section III.

To further illustrate the effectiveness of the proposed method, we list in Tables I-III the spectral radii of the iteration matrices, the iteration counts, and the elapsed CPU time obtained by implementing the CSCS(m)-CG method. The numerical results obtained by the CG method without using a preconditioner are also listed for comparison. In Tables I-III and the subsequent tables, the iteration number, elapsed CPU time, and the spectral radius of iteration matrices defined in (3) are denoted by Iter, Time, and $\rho(G(\alpha))$, respectively.

TABLE I
Numerical results of Example 1 when n = 2000, $\alpha = 0.6$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.738	25	0.023	41	0.046
2	0.544	17	0.013		
3	0.401	13	0.010		
4	0.296	12	0.010		

TABLE II
Numerical results of Example 1 when n = 4000, $\alpha = 0.8$.

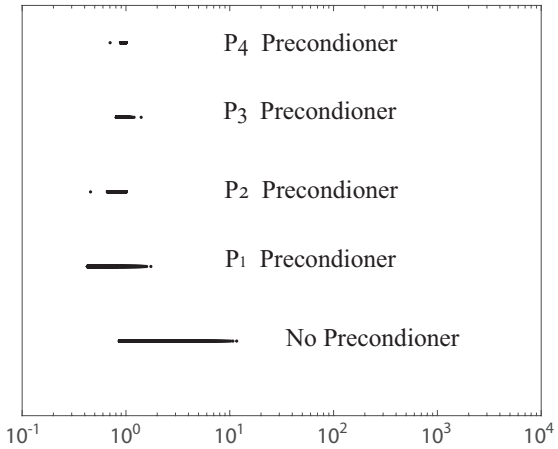
m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.684	25	0.038	41	0.159
2	0.467	15	0.021		
3	0.319	13	0.019		
4	0.218	10	0.017		

TABLE III
Numerical results of Example 1 when n = 6000, $\alpha = 1.0$.

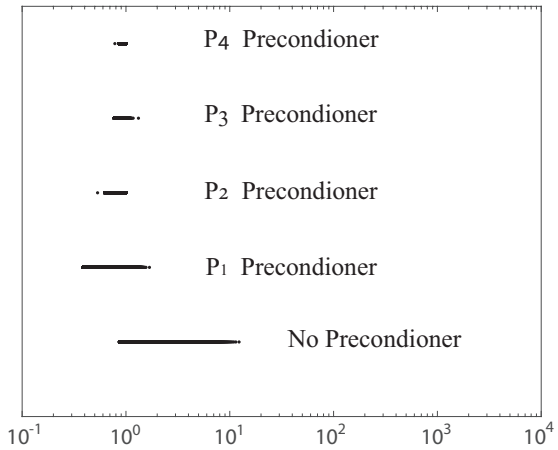
m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.644	25	0.051	42	0.351
2	0.414	14	0.029		
3	0.267	13	0.028		
4	0.172	9	0.027		

From Tables I-III, we can observe that the iteration count of the CSCS(m)-CG method becomes smaller as m increases. Generally, the larger m may require more computational time at each iteration, however, the total computation can be greatly reduced due to fewer iterations. This means that the shifted multi-step CSCS preconditioners P_m can accelerate the convergence of the CG methods significantly.

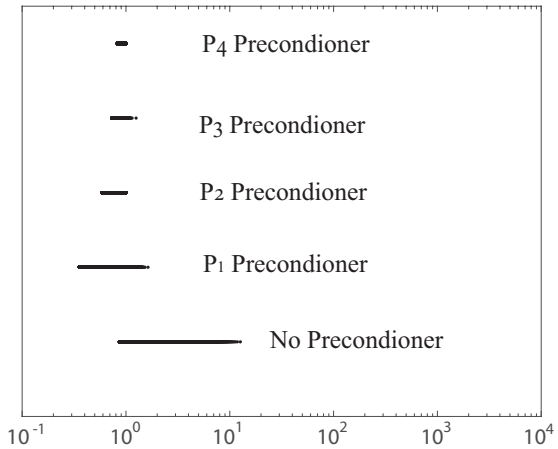
Example 2: ([10]) Consider the Toeplitz matrix T defined



(a) n = 2000, $\alpha = 0.6$



(b) n = 4000, $\alpha = 0.8$



(c) n = 6000, $\alpha = 1.0$

Fig. 1. Spectral distributions of preconditioned matrices for Example 1.

by

$$t_k = \begin{cases} -\sqrt{-1}(\mu - \upsilon)(1 + (-1)^k)/\pi k, & k > 0, \\ \mu + \upsilon, & k = 0, \\ \mathsf{conj}(t_k), & k < 0, \end{cases} \tag{10}$$

where μ and υ are the maximum and minimum values of the generating function, respectively.

In this example, the generating function is given by

$$f(x) = \begin{cases} \frac{\mu-\upsilon}{\pi}x + \mu & , \quad -\pi \le x < 0, \\ \frac{\mu-\upsilon}{\pi}x + \upsilon & , \quad 0 < x \le \pi, \end{cases} \tag{11}$$

which implies that T is Hermitian positive definite. The computational results with $\mu = 10$, $\upsilon = 0.5$ are listed in Tables IV-VI for $n = 2000$, 4000, and 6000, respectively. These results further demonstrate that the iteration counts and the computing times are smaller when the P_m preconditioner is used in the CG method.

TABLE IV
NUMERICAL RESULTS OF EXAMPLE 2 WHEN n = 2000, $\alpha = -1.0$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.815	42	0.034	60	0.061
2	0.664	30	0.022		
3	0.542	23	0.018		
4	0.442	20	0.019		

TABLE V
NUMERICAL RESULTS OF EXAMPLE 2 WHEN n = 4000, $\alpha = -0.9$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.811	43	0.055	61	0.132
2	0.658	29	0.039		
3	0.534	24	0.036		
4	0.433	20	0.036		

TABLE VI
NUMERICAL RESULTS OF EXAMPLE 2 WHEN n = 6000, $\alpha = -0.8$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.809	43	0.076	61	0.151
2	0.655	29	0.055		
3	0.530	23	0.052		
4	0.429	20	0.048		

Example 3: Consider the discretization of the heat equation

$$\frac{\partial u}{\partial t} - \tilde{\alpha}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{12}$$

with the homogeneous boundary conditions using the centered and the upwind difference schemes.

For the mathematical treatment it is sufficient to consider the case where $\tilde{\alpha} = 1$. This leads to the symmetric Toeplitz matrix $T = \mathrm{tridiag}(-1,\ 2{+}h^2/\tau, -1) \in \mathbb{R}^{n\times n}$, where h and τ are the space and time steps, respectively. The generating function of this matrix is given by $f(x) = 4\sin^2(x/2) + h^2/\tau$.

In fact, the eigenvalues of T are $\lambda_j(T) = 4\sin^2[\pi(j+1)/(n+1)] + h^2/\tau$ for $j = 0, 1, \cdots, n-1$. Clearly, T is ill-conditioned for larger n and smaller h^2/τ, which implies that the CG method for solving (1) will converge slowly. However, from the numerical results presented in Tables VII-IX for various h^2/τ when $n = 2000$, we can see that the preconditioner P_m significantly accelerates the convergence of the CG method.

In our numerical tests, all the evidences from Tables I-IX and Figure 1 show that as the degree m increases, the

TABLE VII
NUMERICAL RESULTS OF EXAMPLE 3 WHEN $\alpha = -0.4$, $h^2/\tau = 1\times10^{-2}$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.984	263	0.112	272	0.652
2	0.967	207	0.132		
3	0.952	175	0.125		
4	0.936	152	0.127		

TABLE VIII
NUMERICAL RESULTS OF EXAMPLE 3 WHEN $\alpha = -0.4$, $h^2/\tau = 5 \times 10^{-2}$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.923	110	0.053	119	0.327
2	0.852	85	0.059		
3	0.787	69	0.050		
4	0.726	56	0.046		

TABLE IX
NUMERICAL RESULTS OF EXAMPLE 3 WHEN $\alpha = -0.4$, $h^2/\tau = 1\times10^{-1}$.

m	CSCS(m)-CG			CG	
	$\rho(G(\alpha))^m$	Iter	Time(s)	Iter	Time(s)
1	0.857	75	0.042	83	0.255
2	0.735	56	0.045		
3	0.630	44	0.032		
4	0.540	36	0.030		

more clustered the eigenvalues of the preconditioned matrix are, and the faster the convergence rate of the CSCS(m)-CG method will be. Although the reduction in terms of iteration count compensate for the additional work per iteration when the larger m are selected, the number of operations at each iteration increases rapidly. In summary, from the numerical results of the preceding three examples we found that $m = 3$ or 4 could be appropriately used in the PCG method with the shifted multi-step CSCS preconditioner P_m in terms of both iteration number and computational cost. Theoretically, one should choose the optimal m, however, it is not easy to get the optimal m which reaches a tradeoff between the computational complexity and convergence rate. We will continue to further in-depth study from the viewpoint of both theory and computations for this problem.

VI. CONCLUSION

In this paper, we investigate the solution of Hermitian positive definite Toeplitz systems, with a particular focus on the role of preconditioning in enhancing the performance of Krylov subspace iterative methods. Based on the shifted CSCS iterative method, we introduce the shifted multi-step CSCS preconditioner. We rigorously prove that if the shifted CSCS splitting is P-regular, the spectrum of the preconditioned coefficient matrix clusters around 1 for moderate values of m, leading to improved convergence properties. The numerical results validate the effectiveness of our proposed method, demonstrating significant improvements in both convergence rate and computational efficiency. These findings underscore the potential of the shifted multi-step CSCS preconditioner as a powerful tool for solving large-scale Hermitian positive definite Toeplitz systems.

REFERENCES

[1] T. Chan and M. K. Ng, "Conjugate gradient methods for Toeplitz systems," SIAM Review, vol. 38, no. 3, pp. 427-482, 1996.
[2] T. Chan and X. Jin, "An introduction to iterative Toeplitz solvers," SIAM, 2007.
[3] D. Kressner, "Structured eigenvalue problems and Toeplitz matrices," SIAM Journal on Matrix Analysis and Applications, vol. 39, pp. 958-987, 2018.
[4] J. Xia, "Superfast multifrontal Toeplitz solvers by randomized sampling," SIAM Journal on Matrix Analysis and Applications, vol. 33, pp. 837-858, 2012.
[5] J. Xia, "Efficient structured solvers for Toeplitz systems via randomized approaches," SIAM Journal on Matrix Analysis and Applications, vol. 35, pp. 636-659, 2014.
[6] H. Fang, "Hermitian and skew-Hermitian splitting methods for Toeplitz systems," Journal of Computational Mathematics, vol. 28, pp. 188-201, 2010.
[7] G. Gu and J. Tian, "Accelerated HSS methods for Toeplitz matrices," Journal of Numerical Analysis, vol. 27, pp. 245-260, 2009.
[8] M. K. Ng, "Circulant and skew-circulant splitting iterative methods for Toeplitz systems," Journal of Computational and Applied Mathematics, vol. 328, no. 1, pp. 101-108, 2003.
[9] M. K. Ng, "Iterative methods for Toeplitz systems," Oxford University Press, 2003.
[10] S. Akhondi, "Accelerated CSCS iterative methods for Toeplitz systems," Journal of Computational Methods, vol. 49, pp. 159-172, 2012.
[11] X. Liu, "Shifted classical CSCS iterative methods for Toeplitz systems," Numerical Linear Algebra with Applications, vol. 23, pp. 780-798, 2016.
[12] Z. Wang, "Augmented Kaczmarz algorithm for inconsistent systems," Numerical Algorithms, 2025 (to appear).
[13] Y. Zhu, "Local circulant and residue splitting iterative method for Toeplitz-structured saddle point problems," Journal of Computational Mathematics, vol. 38, pp. 245-260, 2020.
[14] M. Benzi, "Preconditioning techniques for large linear systems," Journal of Computational Physics, vol. 182, pp. 418-477, 2002.
[15] Z. Bai, "Generalized preconditioners for Toeplitz systems," Mathematics of Computation, vol. 66, pp. 603-620, 1997.
[16] Z. Bai, "Preconditioning strategies for Toeplitz systems: A survey," Applied Numerical Mathematics, vol. 56, pp. 1481-1496, 2006.
[17] M. Benzi, "Preconditioners based on circulant matrices," Numerical Linear Algebra with Applications, vol. 11, pp. 1-22, 2004.
[18] T. Chan, "An optimal circulant preconditioner for Toeplitz systems," SIAM Journal on Scientific and Statistical Computing, vol. 10, pp. 104-119, 1989.
[19] O. Axelsson, "Iterative solution methods," Cambridge University Press, 1985.
[20] A. M. Bruaset, "A survey of preconditioned iterative methods," Longman Scientific and Technical, 1995.
[21] Y. Saad, "Iterative methods for sparse linear systems," SIAM, 2000.
[22] R. Adams, "A polynomial preconditioner for conjugate gradient iterations," SIAM Journal on Scientific and Statistical Computing, vol. 6, pp. 452-463, 1985.
[23] W. Cao, "Polynomial preconditioners for Toeplitz systems," Journal of Computational Mathematics, vol. 20, pp. 281-294, 2002.
[24] A. Hadjidimos, "Multistep preconditioners for iterative solvers," Numerische Mathematik, vol. 73, pp. 133-147, 1996.
[25] T. Huang, "Optimal polynomial preconditioners for Toeplitz systems," Numerical Linear Algebra with Applications, vol. 22, pp. 131-154, 2015.
[26] R. A. Horn and C. R. Johnson, "Matrix Analysis," Cambridge University Press, Cambridge, UK, 1985.
[27] R. Nabben, "A note on comparison theorems for splittings and multisplittings of Hermitian positive definite matrices," Linear Algebra and its Applications, vol. 233, pp. 67–80, 1996.
[28] Z. Bai, "Multisplitting methods for Toeplitz systems," SIAM Journal on Scientific Computing, vol. 25, pp. 495-512, 2004.
[29] T. Huang, "Two-stage iterative methods for Toeplitz systems," Linear Algebra and its Applications, vol. 185, pp. 121-138, 1993.
[30] K. Asanovic, R. Bodik, J. Demmel, et al., "A view of the parallel computing landscape," Communications of the ACM, vol. 52, pp. 56-67, 2009.