Shift-Invert Lanczos Method for the Symmetric Positive Semidefinite Toeplitz Matrix Exponential

Hong-Kui Pang and Hai-Wei Sun*,†

Department of Mathematics, University of Macau, Macao, China

SUMMARY

The Lanczos method with shift-invert technique is exploited to approximate the symmetric positive semidefinite Toeplitz matrix exponential. The complexity is lowered by the Gohberg-Semencul formula and the fast Fourier transform. Application to the numerical solution of an integral equation is studied. Numerical experiments are carried out to demonstrate the effectiveness of the proposed method.

KEY WORDS: Toeplitz; matrix exponential; Krylov subspace; Lanczos method; shift-invert; Gohberg-Semencul formula

1. INTRODUCTION

An n-by-n matrix T_n is said to be *Toeplitz* if $[T_n]_{j,k} = t_{j-k}$ for $1 \le j, k \le n$. Toeplitz matrices occur in a variety of applications in mathematics and engineering; see [1, 2] and the references therein. In this paper we study the approximation to the product of the symmetric Toeplitz matrix exponential (TME) with a vector

$$y(\tau) = e^{-\tau T_n} r, \tag{1}$$

where T_n is symmetric positive semidefinite Toeplitz matrix, $\tau > 0$ is a scaling factor, and r is a given vector. The TME is involved in a number of applications. For option pricing in jump-diffusion models, the TME is employed to calculate the solution of a partial integro-differential equation [3]. The TME can also be applied to numerically solve the Volterra-Wiener-Hopf equation; see [4, 5] and Example 3 in Section 4 for instance.

*Correspondence to: Department of Mathematics, University of Macau, Macao, China.
†E-mail: HSun@umac.mo

Contract/grant sponsor: FDCT of Macao; contract/grant number: 033/2009/A

Contract/grant sponsor: University of Macau; contract/grant number: UL020/08-Y3/MAT/JXQ01/FST, RG057/09-10S/SHW/FST

Classical methods for computing an n-by-n dense matrix exponential, such as the matrix decomposition method or the scaling and squaring method [6], need $\mathcal{O}(n^3)$ complexity. Over the last twenty years, Krylov subspace methods have been intensively investigated for large and sparse matrices, due to their efficiency and easy implementation [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. The main idea is to project the matrix exponential onto a small Krylov subspace and compute the resulting matrix exponential. This approach is achieved by the Lanczos process for symmetric matrices or the Arnoldi process for nonsymmetric matrices, while both require only the matrix-vector multiplications. Furthermore, the shift-invert technique can be employed to speed up the Lanczos or Arnoldi process [14, 17].

In general, Toeplitz matrices are dense. Nevertheless, the Toeplitz matrix-vector multiplication can be computed by the fast Fourier transform (FFT) with $\mathcal{O}(n \log n)$ complexity [1, 2]. Furthermore, the inverse of Toeplitz matrix can be explicitly expressed by the Gohberg-Semencul formula (GSF) [18, 19]. Those properties can be used to speed up the approximation to the TME. In this paper, we extend the shift-invert Lanczos method [17] to compute the symmetric positive semidefinite TME. By the Toeplitz structure and the GSF, the computational cost of approximation to the symmetric TME is reduced to $\mathcal{O}(n \log n)$ operations. As an application, we employ our method to a TME which arises in the numerical solution of the Volterra-Wiener-Hopf equation [4]. We remark that it is not necessary to assume the matrix to be positive semidefinite, which is postulated throughout this paper. For the indefinite case, one can easily transform it to the positive semidefinite case by performing a suitable shift and then multiply the result with a corresponding factor [17].

The paper is organized as follows. In Section 2, we introduce some background on Toeplitz matrices. In Section 3, the shift-invert Lanczos method is extended to approximate the symmetric TME. Numerical results and applications to the Volterra-Wiener-Hopf equation are reported in Section 4 to demonstrate the effectiveness of the proposed method. Finally, concluding remarks are given in Section 5.

2. TOEPLITZ MATRICES AND SOME PROPERTIES

We define a Toeplitz matrix C_n ($[C_n]_{j,k} = c_{j-k}$) as a *circulant* matrix if $c_k = c_{k-n}$ for $1 \le k \le n-1$. A circulant matrix can be diagonalized by the Fourier matrix F_n; i.e.,

$$C_n = F_n^* \Lambda_n F_n, \tag{2}$$

where the entries of F_n are given by

$$[F_n]_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi \mathbf{i} jk}{n}}, \qquad \mathbf{i} \equiv \sqrt{-1}, \qquad 0 \le j,\, k \le n-1,$$

and Λ_n is a diagonal matrix holding the eigenvalues of C_n. We note that Λ_n can be obtained in $\mathcal{O}(n \log n)$ operations by taking the FFT of the first column of C_n [1, 2]. Once Λ_n is obtained, the product $C_n r$ or $C_n^{-1} r$ for any vector r can be computed by two n-length FFTs with $\mathcal{O}(n \log n)$ complexity.

A Toeplitz matrix S_n ($[S_n]_{j,k} = s_{j-k}$) is *skew-circulant* if $s_k = -s_{k-n}$ for $1 \le k \le n-1$. Analogous to (2), a skew-circulant matrix has the spectral decomposition [2],

$$S_n = \Omega^* F_n^* \Lambda_n F_n \Omega, \tag{3}$$

where $\Omega = \mathrm{diag}(1, e^{-\mathbf{i}\pi/n}, \ldots, e^{-\mathbf{i}(n-1)\pi/n})$. Therefore, the products of $S_n r$ and $S_n^{-1} r$ also can be computed by two n-length FFTs with $\mathcal{O}(n \log n)$ complexity.

Notice that a Toeplitz matrix T_n can be embedded into a $2n$-by-$2n$ circulant matrix; i.e.,

$$\begin{bmatrix} T_n & \times \\ \times & T_n \end{bmatrix} \begin{bmatrix} r \\ 0 \end{bmatrix} = \begin{bmatrix} T_n r \\ \dagger \end{bmatrix}.$$

Therefore, the multiplication of $T_n r$ can be done by two $2n$-length FFTs, or roughly four n-length FFTs, with $\mathcal{O}(n \log n)$ complexity provided that the spectra of the embedded circulant matrix are already obtained; see more details in [1, 2].

The celebrated GSF for the inverse of a symmetric positive definite Toeplitz matrix T_n is formulated as [18]:

$$T_n^{-1} = \frac{1}{l_1} \left(L_n L_n^\intercal - \hat{L}_n \hat{L}_n^\intercal \right), \tag{4}$$

where both L_n and $\hat{L}_n$ are lower triangular Toeplitz matrices given by

$$L_n = \begin{bmatrix} l_1 & 0 & \cdots & 0 \\ l_2 & l_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_n & \cdots & l_2 & l_1 \end{bmatrix} \quad \text{and} \quad \hat{L}_n = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ l_n & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_2 & \cdots & l_n & 0 \end{bmatrix},$$

with $l = (l_1, l_2, \ldots, l_n)^\intercal$ being the first column of T_n^{-1}. Thus l is the solution of the linear system

$$T_n l = e_1 \equiv (1, 0, \ldots, 0)^\intercal. \tag{5}$$

We note that $l_1 > 0$ always holds due to the symmetric positive definiteness of T_n [19]. Once l in (5) is obtained, according to (4), the multiplication of $T_n^{-1} r$ can be done in $\mathcal{O}(n \log n)$ operations with four n-length Toeplitz matrix-vector products, or roughly sixteen n-length FFTs. In the following, we present a strategy to reduce the cost of computing $T_n^{-1} r$ to only four n-length FFTs, or roughly one n-length Toeplitz matrix-vector products [20].

Let

$$J_n = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \iddots & \vdots \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

be the n-by-n anti-identity matrix. Because of the displacement structure of Toeplitz matrices [19], we have

$$J_n T_n^{-1} J_n = T_n^{-1}, \quad J_n L_n^\intercal J_n = L_n, \quad J_n \hat{L}_n^\intercal J_n = \hat{L}_n.$$

It follows that

$$T_n^{-1} = \frac{1}{l_1} \left(L_n L_n^\intercal - \hat{L}_n \hat{L}_n^\intercal \right) = \frac{1}{l_1} \left(L_n^\intercal L_n - \hat{L}_n^\intercal \hat{L}_n \right).$$

Then we have

$$\begin{aligned} & L_n L_n^\intercal - \hat{L}_n \hat{L}_n^\intercal \\ = \; & \frac{1}{2} \left[(L_n + \hat{L}_n^\intercal)(L_n^\intercal - \hat{L}_n) + (L_n^\intercal + \hat{L}_n)(L_n - \hat{L}_n^\intercal) \right] \\ = \; & \frac{1}{2} \left[(L_n + \hat{L}_n^\intercal)(L_n^\intercal - \hat{L}_n) + J_n (L_n + \hat{L}_n^\intercal)(L_n^\intercal - \hat{L}_n) J_n \right]. \end{aligned}$$

Therefore

$$T_n^{-1}r = Re(z) + J_n Im(z), \tag{6}$$

where

$$z = \frac{1}{2l_1}\left[(L_n + \hat{L}_n^{\intercal})(L_n^{\intercal} - \hat{L}_n)\right](r + \mathbf{i}J_n r), \tag{7}$$

and $Re(z)$ and $Im(z)$ represent the real and imaginary parts of z respectively. We remark that in (7), $L_n + \hat{L}_n^{\intercal}$ is circulant, and $L_n^{\intercal} - \hat{L}_n$ is skew-circulant. Thus the cost for computing $T_n^{-1}r$ is almost the same as computing one circulant and one skew-circulant matrix-vector product, or roughly four n-length FFTs.

In order to construct T_n^{-1} by the GSF, we have to solve the Toeplitz system (5). There are many methods for solving the Toeplitz systems. For instance, the superfast direct methods with $\mathcal{O}(n\log^2 n)$ complexity [21, 22] can be employed to solve the Toeplitz systems. Alternatively, a large class of Toeplitz systems can be iteratively solved in $\mathcal{O}(n\log n)$ [1, 2]. In this paper, we exploit the preconditioned conjugate gradient (PCG) method with the Strang's circulant preconditioner [23] to solve (5). The Strang's circulant preconditioner $s(T_n)$ of T_n is defined as a circulant matrix obtained by copying the central diagonals of T_n and bringing them around to complete the circulant requirement. More precisely, the diagonals s_k of $s(T_n)$ are given by

$$s_k = \begin{cases} t_{n+k}, & -n+1 \le k < -\lfloor n/2 \rfloor, \\ t_k, & -\lfloor n/2 \rfloor \le k \le \lfloor n/2 \rfloor, \\ t_{k-n}, & \lfloor n/2 \rfloor < k \le n-1, \end{cases}$$

where t_k are the diagonals of T_n and $\lfloor n/2 \rfloor$ denotes the largest integer which does not exceed $n/2$. The PCG with the Strang's preconditioner has been widely studied for a large class of Toeplitz systems; see [1, 2] for more details.

3. SHIFT-INVERT LANCZOS METHOD

3.1. Lanczos method

We briefly introduce the standard Lanczos method for approximating the vector $y(\tau) = e^{-\tau T_n}r$. An orthogonal basis of an m-dimensional Krylov subspace

$$\mathcal{K}_m(T_n, r) \equiv \text{span}\ \{r, T_n r, T_n^2 r, \ldots, T_n^{m-1} r\}$$

is constructed by the Lanczos process for a real symmetric matrix T_n as bellow:

Algorithm 1: Lanczos algorithm

1. *Initialize*: Compute $r_1 = r/\left\|r\right\|_2$
2. *Iterate*: For $j = 1, 2, \ldots, m$ do:
   - $d_{j,j} := r_j^{\intercal} T_n r_j$
   - $\hat{r}_{j+1} := T_n r_j - d_{j,j} r_j - d_{j-1,j} r_{j-1}$
   - $d_{j+1,j} := \left\|\hat{r}_{j+1}\right\|_2$
   - $d_{j,j+1} := d_{j+1,j}$
   - $r_{j+1} := \hat{r}_{j+1}/d_{j+1,j}$

From Algorithm 1 we obtain the following relation [16]

$$T_n R_m = R_m D_m + d_{m+1,m} r_{m+1} e_m^\intercal, \tag{8}$$

where $R_m = [r_1, r_2, \ldots, r_m]$ is the n-by-m matrix containing the orthonormal basis of $\mathcal{K}_m(T_n, r)$, D_m is the m-by-m symmetric tri-diagonal matrix which consists of the coefficients $d_{j,k}$, and e_m is the mth column of identity matrix of size m. We note that $R_m e_1 = r_1$ and $D_m = R_m^\intercal T_n R_m$. Therefore, D_m represents the projection of the linear transformation T_n onto the subspace $\mathcal{K}_m(T_n, r)$. Thus, we have the following approximation[16]

$$e^{-\tau T_n} r \approx \beta R_m e^{-\tau D_m} e_1, \qquad \beta = \|r\|_2 .$$

When $m \ll n$, the large matrix exponential $e^{-\tau T_n}$ is replaced by a small matrix exponential $e^{-\tau D_m}$. Furthermore, the small $e^{-\tau D_m}$ can be quickly evaluated by the high-order rational Chebyshev approximations [24, 10] or the scaling and squaring algorithm with Padé approximation [25].

3.2. Shift-invert technique

However, it is shown in [11, 16] that when $\|\tau T_n\|_2$ becomes larger, more iterations in the Lanczos process may be needed to achieve a given accuracy for the approximation to the vector $e^{-\tau T_n} r$. We note a fact that the exponential function is quickly decaying. Hence the vector $e^{-\tau T_n} r$ is mostly determined by the smallest eigenvalues of T_n and their corresponding invariant subspaces. In order to guarantee the fast Lanczos process, van den Eshof and Hochbruck [17] exploited the shift-invert technique which is usually employed to compute the small eigenvalues [26].

Let I_n be the identity matrix of size n and σ denote the shift parameter. The shift-invert Lanczos process [17] is presented as follows.

Algorithm 2: shift-invert Lanczos algorithm

1. *Initialize*: Compute $r_1 = r/\|r\|_2$
2. *Iterate*: For $j = 1, 2, \ldots, m$ do:
   - $d_{j,j} := r_j^\intercal (I_n + \sigma T_n)^{-1} r_j$
   - $\hat{r}_{j+1} := (I_n + \sigma T_n)^{-1} r_j - d_{j,j} r_j - d_{j-1,j} r_{j-1}$
   - $d_{j+1,j} := \|\hat{r}_{j+1}\|_2$
   - $d_{j,j+1} := d_{j+1,j}$
   - $r_{j+1} := \hat{r}_{j+1}/d_{j+1,j}$

Analogous to (8), we have

$$(I_n + \sigma T_n)^{-1} R_m = R_m D_m + d_{m+1,m} r_{m+1} e_m^\intercal, \qquad R_m^\intercal R_m = I_m. \tag{9}$$

Therefore, the approximation to $e^{-\tau T_n} r$ is given by

$$e^{-\tau T_n} r \approx \beta R_m e^{-\tau [\frac{1}{\sigma}(D_m^{-1} - I_m)]} e_1 \equiv \beta R_m g(D_m) e_1, \quad \beta = \|r\|_2, \tag{10}$$

where $g(x) = e^{-\frac{\tau}{\sigma}(x^{-1}-1)}$ with $x \in (0, 1]$ and $g(0) = 0$. The error bound of the approximation formula (10) has been estimated in [17].

Theorem 1. [17] *Let μ be a nonnegative number such that $T_n - \mu I_n$ is positive semi-definite. Then*

$$\|\beta R_m g(D_m)e_1 - y(\tau)\| \leq 2\beta e^{-\tau\mu} E_{m-1}^{m-1}(\widetilde{\sigma}),$$

where $\widetilde{\sigma} = \frac{\sigma}{\tau(1+\sigma\mu)}$ and

$$E_{m-1}^{m-1}(\widetilde{\sigma}) \equiv \inf_{q \in \prod_{m-1}^{m-1}} \sup_{t \geq 0} |q(t) - e^{-t}|$$

with $\prod_k^j \equiv \{p(t)(1+\widetilde{\sigma}t)^{-k} | p \in P_j\}$, in which P_j denotes the set of all polynomials of degree $j-1$ or less.

According to Theorem 1, we note that the error of the approximation by the shift-invert Lanczos method is independent of $\|\tau T_n\|_2$. Only the smallest eigenvalue of T_n plays a modest role in the form of μ. This is a very attractive advantage. In addition, a priori error bound and the optimal choice of the shift parameter σ is achieved by evaluating the quantity $E_{m-1}^{m-1}(\widetilde{\sigma})$.

Nevertheless, the evaluation of the quantity $E_{m-1}^{m-1}(\widetilde{\sigma})$ is not easy. In the absence of insightful analytical estimates, the authors in [17] locate the optimal value σ by numerically estimating the quantity $E_{m-1}^{m-1}(\widetilde{\sigma})$ with $\tau = 1$ and $\mu = 0$. The actual values are tabulated as shown in Table I. We remark that for the case $\tau \neq 1$, one can choose σ as $\sigma_{opt}\tau$, where σ_{opt} is guided by Table I according to the required accuracy. For example, if we are interested in an accuracy of about 10^{-4}, we can consider Table I and decide to choose $\sigma = 0.19\tau$.

Table I. Tabulated values in [17] of the shift-invert parameter σ_{opt}, and the corresponding value $E_j^j(\sigma_{opt})$.

j	$E_j^j(\sigma_{opt})$	σ_{opt}	j	$E_j^j(\sigma_{opt})$	σ_{opt}
1	6.7e-02	1.73e-00	11	4.0e-06	9.90e-02
2	2.0e-02	4.93e-01	12	1.6e-06	1.19e-01
3	7.3e-03	2.64e-01	13	6.1e-07	1.00e-01
4	3.1e-03	1.75e-01	14	2.5e-07	8.64e-02
5	1.4e-03	1.30e-01	15	1.0e-07	7.54e-02
6	4.0e-04	1.91e-01	16	4.0e-08	8.67e-02
7	1.6e-04	1.44e-01	17	1.6e-08	7.63e-02
8	6.5e-05	1.90e-01	18	6.6e-09	6.78e-02
9	2.4e-05	1.47e-01	19	2.7e-09	7.62e-02
10	9.7e-06	1.19e-01	20	1.1e-09	6.82e-02

Popolizio and Simoncini [15] also discussed the selection of the optimal shift-invert parameter σ with a completely different strategy and got somehow similar results. In the numerical tests in Section 4, we choose the optimal parameter $\sigma = \sigma_{opt}\tau$, according to Table I.

3.3. Implementation for the symmetric TME

In the standard Lanczos process, only matrix-vector products $T_n r_j$ for $j = 1, \ldots, m$ are involved. Given a Toeplitz matrix T_n, these products can be done by FFTs with $\mathcal{O}(n \log n)$ complexity; see Section 2. Nevertheless, in the shift-invert Lanczos process, we have to deal with the matrix-vector multiplication $(I_n + \sigma T_n)^{-1} r_j$ at each step. In [17], an inner-outer iteration

scheme is exploited in the shift-invert Lanczos process, where the product $(I_n + \sigma T_n)^{-1} r_j$ is implemented by an inner iteration. Therefore, the stopping criterion of the inner iteration needs to be prescribed. In order to speed up the convergence of the inner iteration, a suitable preconditioner should be constructed. Both choosing an appropriate stopping criterion and constructing a good preconditioner in the inner iteration are not trivial. However, if T_n is a symmetric positive semidefinite Toeplitz matrix, then $I_n + \sigma T_n$ is symmetric positive definite for $\sigma > 0$, and the GSF introduced in Section 2 provides an explicit representation of the inverse of Toeplitz matrix $I_n + \sigma T_n$. Therefore, all products $(I_n + \sigma T_n)^{-1} r_j$ can be carried out by (6) and (7) with $\mathcal{O}(n \log n)$ complexity. The shift-invert Lanczos method with the GSF for the symmetric TME is given as below.

Algorithm 3: shift-invert Lanczos algorithm for symmetric TME

1. Select an optimal shift parameter σ from Table I
2. Solve the symmetric positive definite system $(I_n + \sigma T_n) l = e_1$
3. Perform shift-invert Lanczos algorithm in which each multiplication $(I_n + \sigma T_n)^{-1} r_j$ is computed through GSF (6) and (7) by FFT
4. Compute the approximation $y_m(\tau) = \beta R_m e^{-\frac{\tau}{\sigma}(D_m^{-1} - I_m)} e_1$

In step 2 of Algorithm 3, we solve the linear Toeplitz system

$$(I_n + \sigma T_n) l = e_1 \tag{11}$$

by the PCG method with Strang's preconditioner, which requires $\mathcal{O}(n \log n)$ complexity [1, 2]. Once l is obtained, the products $(I_n + \sigma T_n)^{-1} r_j$ for $j = 1, \ldots, m$ in step 3 can be computed exactly through (6) and (7) by FFT with $\mathcal{O}(n \log n)$ complexity. We remark that for each iteration of shift-invert Lanczos process, the computational cost is almost the same as the standard one; see Section 2 for details. In step 4, we need to compute the vector $e^{-\frac{\tau}{\sigma}(D_m^{-1} - I_m)} e_1$. The calculation of the matrix exponential by some classical methods such as scaling and squaring method [25] needs $\mathcal{O}(m^3)$ operations. Recall that the matrix D_m generated in the Lanczos process is a symmetric positive definite tri-diagonal matrix. Therefore, we can exploit the following partial fraction expansion of the Chebyshev rational approximation to reduce the computational cost:

$$q(z) = \omega_0 + \sum_{j=1}^{k} \frac{\omega_j}{z - \lambda_j}, \tag{12}$$

where λ_j's are the poles of q and ω_j's are the corresponding coefficients [24, 10]. This approximation is greatly enough to obtain a good working accuracy only for $k = 14$ [24]. By the partial fraction expansion of the Chebyshev rational approximation, the calculation in step 4 becomes

$$\begin{aligned} e^{-\frac{\tau}{\sigma}(D_m^{-1} - I_m)} e_1 &\approx \omega_0 e_1 + \sum_{j=1}^{k} \omega_j \left[\frac{\tau}{\sigma}(D_m^{-1} - I_m) - \lambda_j I_m \right]^{-1} e_1 \\ &= \omega_0 e_1 + \sum_{j=1}^{k} \sigma \omega_j \left[\tau I_m - (\tau + \sigma \lambda_j) D_m \right]^{-1} D_m e_1. \end{aligned}$$

We note that D_m is a symmetric tri-diagonal matrix. Hence $\tau I_m - (\tau + \sigma\lambda_j)D_m$ is tri-diagonal as well. Thus the cost of computing each term in the above equation is $\mathcal{O}(m)$, which implies that the complexity of approximating the vector $\hat{y}(\tau) = e^{-\frac{\tau}{\sigma}(D_m^{-1} - I_m)}e_1$ is about $\mathcal{O}(km)$. After obtaining $\hat{y}(\tau)$, we calculate the vector $\beta R_m \hat{y}$ in $\mathcal{O}(mn)$ operations. In summary, the total computational cost in Algorithm 3 is of $\mathcal{O}(mn \log n)$. We remark that m is in general much smaller than n; see [15, 17] for instance.

4. NUMERICAL RESULTS

In this section we demonstrate the behavior of the shift-invert Lanczos algorithm for approximating the vector $y(\tau) = e^{-\tau T_n} r$ with T_n being the symmetric positive semidefinite Toeplitz matrix. All numerical experiments are tested by running MATLAB R2009b on a Pentium(R) D 3.40GHz, 3.39GHz with 504MbRAM. In our experiments, we take the MATLAB command "expm" as the true value of $y(\tau)$ except for Example 2, where the true solution is analytically given. In all tables, "n" denotes the matrix size, "tol" represents the tolerance of the relative error $\|y(\tau) - y_m(\tau)\|_2 / \|y(\tau)\|_2 < \text{tol}$, where $y_m(\tau)$ is the numerical approximation to $y(\tau)$. Symbols "Stand" and "SI" refer to the standard Lanczos method and the shift-invert Lanczos method, respectively. The shift parameter $\sigma = \sigma_{opt}\tau$, where σ_{pot} is chosen from Table I according to the required accuracy. The partial fraction expansion of the Chebyshev rational approximation (12) is used to compute the small projection matrix exponential in both methods.

Example 1. We consider a symmetric positive definite Toeplitz matrix T_n whose diagonals t_k are given by [2]

$$t_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} x^4 e^{-\mathbf{i}kx}\mathrm{d}x, \quad k = 0, \pm 1, \pm 2, \ldots, \pm(n-1).$$

The vector r is chosen to be all ones. We use the standard Lanczos method and the shift-invert Lanczos method to approximate the vector $e^{-\tau T_n} r$, respectively. Note that $\|T_n\|_2$ keeps almost unchanged regardless of the matrix size n in this example. Hence we can fix $n = 1024$ and vary the value τ. The number of iterations for various τ and final accuracies "tol" are reported in Table II.

Table II. Iteration numbers of shift-invert Lanczos method and standard Lanczos method for $n = 1024$ in Example 1.

τ	tol $= 10^{-4}$		tol $= 10^{-7}$		tol $= 10^{-9}$	
	SI	Stand	SI	Stand	SI	Stand
1	6	21	13	33	17	40
10	7	67	14	102	19	120
100	7	213	14	319	19	375
1000	7	668	14	976	19	1132

From Table II we see that the number of iterations by the shift-invert Lanczos method is much smaller than the one by the standard Lanczos method, especially for the large τ; i.e., the

large $\|\tau T_n\|_2$. Moreover, for the shift-invert Lanczos method, the iteration numbers stay almost unchanged. This fact illuminates that the convergence of the shift-invert Lanczos method is actually independent of $\|\tau T_n\|_2$, while the standard Lanczos method needs more iterations as $\|\tau T_n\|_2$ increases.

As a comparison, we compute the vector $\hat{y}(\tau) = e^{-\frac{\tau}{\sigma}(D_m^{-1} - I_m)} e_1$ by the MATLAB command "expm". Note that the complexity by "expm" is of order $\mathcal{O}(m^3)$, while that by the Chebyshev rational approximation is $\mathcal{O}(m)$. Numerical results are displayed in Table III, where "direc" means that $\hat{y}(\tau)$ is computed directly by the MATLAB command "expm" and "Chev" means that by the Chebyshev rational approximation. From Table III, we see that the CPU time by the Chebyshev rational approximation is less than the one by the MATLAB command "expm" directly. The advantage by the Chebyshev rational approximation is even more evident for the big size m.

Table III. Comparison of iteration numbers and CPU times (in parentheses, unit second) by the Chebyshev rational approximation and by the MATLAB command to compute the small projection matrix exponential in Example 1.

τ	$\text{tol} = 10^{-9}$			
	SI(direc)	SI(Chev)	Stand(direc)	Stand(Chev)
1	17(0.0010)	17(0.0005)	40(0.0016)	40(0.00063)
10	19(0.0011)	19(0.0005)	120(0.0085)	120(0.0010)
100	19(0.0011)	19(0.0005)	375(8.6224)	375(0.0023)
1000	19(0.0011)	19(0.0005)	1132(182.0566)	1132(0.0068)

Example 2. We consider a heat equation which is an example in [27]. Assume an iron bar, of length $50cm$, with specific heat $c = 0.437J/(gK)$, density $\rho = 7.88g/cm^3$, and thermal conductivity $\kappa = 0.836W/(cmK)$, is insulated except at the end and has the initial temperature

$$\psi(x) = 5 - \frac{1}{5}\,|x - 25|\,,$$

where $\psi(x)$ is given in degree Celsius. We also assume that, at time $t = 0$, the ends of the bar are placed in an ice bath (0 degrees Celsius). Now we compute the temperature distribution after 60 and 300 seconds. This problem satisfies the heat equation

$$\rho c \frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \qquad 0 \le x \le 50, \quad t > 0, \tag{13}$$

subject to the initial and boundary conditions

$$u(x, 0) = \psi(x), \quad 0 \le x \le 50,$$

$$u(0, t) = 0, \quad u(50, t) = 0, \quad t > 0.$$

The analytical solution of (13) is given by [27]:

$$u(x, t) = \sum_{j=1}^{\infty} a_j(t) \sin\left(\frac{j\pi x}{50}\right), \quad a_j(t) = \frac{40 \sin(j\pi/2)}{\pi^2 j^2} \exp\left(-\frac{\kappa j^2 \pi^2}{50^2 \rho c} t\right). \tag{14}$$

Numerically solving (13) by cental difference method leads to a matrix exponential problem

$$\hat{u}(t) = e^{-tT_n}u_0,$$

where $\hat{u}(t) = (\hat{u}_1(t), \hat{u}_2(t), \ldots, \hat{u}_n(t))^\intercal$ is an approximation to the exact solution $u(x,t)$ at grid points $x_j, j = 1, 2, \ldots, n$, T_n is a symmetric positive definite Toeplitz matrix, and $u_0 = (\psi(x_1), \psi(x_2), \ldots, \psi(x_n))^\intercal$ is the initial vector. We approximate $\hat{u}(t)$ at time τ by the standard Lanczos method and the shift-invert Lanczos method, respectively. In our tests, the iteration is stopped when the relative error between the iteration solution $\hat{u}_m$ and the true solution u no longer decreases. Here we take the first 150 terms of the series solution (14) to be the true solution u for comparison. Table IV shows the iteration numbers and CPU times (in parentheses, unit second) of the shift-invert Lanczos method and the standard Lanczos method for different numbers of spacial grid points n and times τ (unit second). The column labeled "err" stands for the relative error of $\|u - \hat{u}_m\|_2 / \|u\|_2 \leq$ err, which is a little different from "tol" in Example 1.

Table IV. Iteration numbers and CPU times (in parentheses, unit second) for the shift-invert Lanczos method and the standard Lanczos method in Example 2.

n	$\tau = 60$			$\tau = 300$		
	err	SI	Stand	err	SI	Stand
128	7.88e-05	9(0.0036)	55(0.0101)	6.71e-05	9(0.0036)	64(0.0115)
256	1.97e-05	11(0.0051)	115(0.0228)	1.68e-05	9(0.0042)	128(0.0257)
512	4.92e-06	13(0.0098)	242(0.0789)	4.19e-06	9(0.0075)	255(0.0833)
1024	1.23e-06	13(0.0165)	479(0.2662)	1.05e-06	9(0.0132)	509(0.2843)
2048	3.08e-07	14(0.0493)	980(1.4535)	2.62e-07	9(0.0309)	1020(1.4205)
4096	7.69e-08	16(0.0658)	1993(2.7482)	6.54e-08	10(0.0361)	2041(3.3108)
8192	1.92e-08	16(0.1735)	> 3500	1.67e-08	10(0.1134)	> 4000

From Table IV we see that the shift-invert Lanczos method needs fewer iterations and CPU times to reach the final required accuracies than those of standard Lanczos method. In particular, for large spatial grid numbers, the standard Lanczos method becomes unacceptable for its very great iteration numbers, while the shift-invert Lanczos method still works well.

Example 3. We consider the Volterra-Wiener-Hopf integral equation of the second kind [4] as an application of the proposed algorithm. The integral equation is given as follows,

$$\begin{cases} u(x,t) = f(x,t) + \lambda \int_0^t \int_0^\infty K_0(|x-\xi|)u(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta, \\ 0 < x < \infty, 0 \leq t < \infty, \lambda < 0, \end{cases} \tag{15}$$

where λ is a constant, $f(x,t)$ is a given function, and $K_0(x) = \int_0^\infty \frac{\cos\xi}{\sqrt{x^2+\xi^2}}\mathrm{d}\xi$ is the Macdonald function [4], or the modified Bessel function of the second kind [28], which has the property that $K_0(x) \simeq \ln(2/\gamma x)$ (γ is the Euler constant) in the neighborhood $x \to 0$, and $K_0(x) = \sqrt{\pi/2x}$, $(x \to \infty)$.

To solve (15) numerically, we discretize the x-domain with a uniform mesh size Δx and the grid number n. Then the domain of x is truncated to be $(0, \Delta x \cdot n]$. Correspondingly, the infinity domain of integral in the right hand side of (15) is also truncated to be $(0, \Delta x \cdot n]$. By exploiting the rectangle rule with the same uniform mesh size $\Delta\xi = \Delta x$ for the truncated integral, we obtain the following equation

$$\hat{u}(t) = \overline{f}(t) + \lambda \int_0^t T_n \hat{u}(\eta) \mathrm{d}\eta, \tag{16}$$

where $\hat{u}(t)$, analogous to the notation in Example 2, is an approximation to the exact solution $u(x,t)$ at the grid points $j\Delta x$, $j = 1, \ldots, n$, $\overline{f}(t) = (f(\Delta x, t), \ldots, f(n\Delta x, t))^\intercal$, and T_n comes from the discretization of truncated integral with diagonals

$$t_k = \begin{cases} K_0(k\Delta x), & k = \pm 1, \pm 2, \ldots, \pm(n-1), \\ \dfrac{1}{\Delta x} \displaystyle\int_0^{\Delta x} \ln(2/\gamma x) \mathrm{d}x, & k = 0. \end{cases}$$

We note that T_n is a symmetric positive definite Toeplitz matrix [4, 29]. In our tests, we take $\lambda = -10$ and $f(x,t) = 10x^2 e^{-x/2}$ for simplicity. Therefore the approximation solution $\hat{u}(t)$ is expressed as

$$\hat{u}(t) = e^{t\lambda T_n} \overline{f}_0,$$

where $\overline{f}_0 = \overline{f}(0)$ is the initial vector.

In the following, we approximate $e^{t\lambda T_n}\overline{f}_0$ at the time τ by the shift-invert Lanczos method and the standard Lanczos method, respectively. In all experiments, we fix the mesh size $\Delta x = 0.01$. The iteration numbers for various dimensions n, times τ and final tolerances are summarized in Table V. From Table V, we see that the iteration numbers of the shift-invert Lanczos method are smaller than those of the standard Lanczos method. In addition, we also take into account the whole computational costs, measured in the number of n-length FFTs, in Figure 1 for $n = 512$ and $\tau = 20$. All the numerical results in Table V and Figure 1 show that the shift-invert Lanczos method outperforms the standard one.

We now illustrate that the proposed shift-invert Lanczos method by the GSF through (6) and (7) to handle the inverse really reduces the computational cost compared with other implementations, such as the Cholesky factorization. In Algorithm 2, the products $(I_n + \sigma T_n)^{-1} r_j$ must be computed in order to implement the shift-invert Lanczos process. If the Cholesky factorization is used to decompose the matrix $I_n + \sigma T_n$, then the computational cost is $\mathcal{O}(n^3)$. Having the factorization in hand, we compute $(I_n + \sigma T_n)^{-1} v_j$ at each step of the Lanczos process by solving two triangular systems, which require $\mathcal{O}(n^2)$ operations. However, by the GSF to calculate the products $(I_n + \sigma T_n)^{-1} r_j$, we need to solve a Toeplitz system (11) once and for all, which can be done by the PCG method in $\mathcal{O}(n \log n)$ complexity. Then what remains is to compute $(I_n + \sigma T_n)^{-1} r_j$ at each step of the Lanczos process by FFT through (6) and (7), where the complexity is about four n-length FFTs. The CPU times of the standard Lanczos method, and shift-invert Lanczos method with GSF and Cholesky factorization are reported in Table VI. The mark "LLT" indicates the Cholesky factorization. From Table VI, we see that the shift-invert Lanczos method with GSF is less time-consuming than both the standard Lanczos and the shift-invert Lanczos method with Cholesky factorization.

Table V. Iteration numbers of the standard Lanczos method and shift-invert Lanczos method for various dimensions, time τ and final tolerances in Example 3.

n	τ	tol $= 10^{-4}$		tol $= 10^{-6}$	
		SI	Stand	SI	Stand
256	10	13	25	17	30
	20	13	33	18	39
	30	13	39	18	46
512	10	13	33	18	38
	20	13	43	18	50
	30	13	51	19	60
1024	10	13	40	18	47
	20	13	54	18	63
	30	13	64	19	73
2048	10	13	48	18	57
	20	14	65	19	77
	30	14	78	19	94

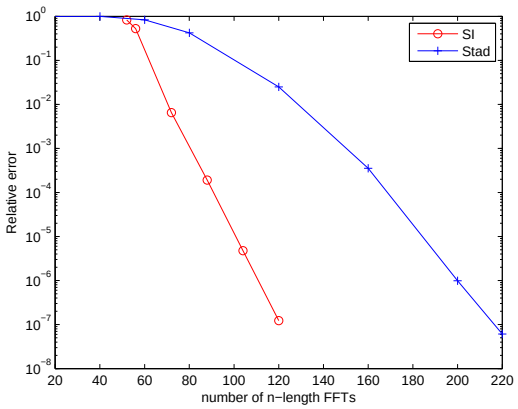

Figure 1. Number of n-length FFTs versus relative error for $n = 512$, $\tau = 20$, and $\Delta x = 0.01$ in Example 3.

5. Concluding remarks

In this paper we employ the shift-invert Lanczos algorithm to approximate the symmetric positive semidefinite TME and apply the proposed method to solve the Volterra-Wiener-Hopf integral equation (15). The GSF (4) is exploited such that we can avoid using the inner iteration as [17] for implementing the shift-invert Lanczos process. Moreover, the complexity is reduced to $\mathcal{O}(n \log n)$ by the Toeplitz properties. Numerical results are performed to show the efficiency of the proposed method. We remark that the formulae (6) and (7) is no longer suitable for the nonsymmetric case. More discussions for the nonsymmetric TME are studied in [30].

Table VI. CPU times (in seconds) of standard Lanczos method, shift-invert Lanczos method with GSF or Cholesky decomposition for $\tau = 20$ and $\Delta x = 0.01$ in Example 3.

n	tol = 10^{-4}			tol = 10^{-6}		
	Stand	SI		Stand	SI	
		GSF	LLT		GSF	LLT
256	0.0063	0.0058	0.0254	0.0077	0.0068	0.0332
512	0.0107	0.0083	0.1272	0.0127	0.0101	0.1603
1024	0.0276	0.0148	0.6571	0.0357	0.0178	0.8091
2048	0.0524	0.0352	2.9190	0.0682	0.0413	3.5170

ACKNOWLEDGEMENT

The authors would like to thank our colleagues Spike T. Lee and Xin Liu for their helpful discussions. We are also grateful to the anonymous referees for the useful suggestions and comments that improved the presentation of this paper.

REFERENCES

1. Chan R, Jin X. *An Introduction to Iterative Toeplitz Solvers*. SIAM: Philadelphia, 2007.
2. Chan R, Ng M. Conjugate gradient methods for Toeplitz systems. *SIAM Review* 1996; **38**:427–482.
3. Tangman DY, Gopaul A, Bhuruth M. Exponential time integration and Chebychev discretisation schemes for fast pricing of options. *Applied Numerical Mathematics* 2008; **58**:1309–1319.
4. Abdou MA, Badr AA. On a method for solving an integral equation in the displacement contact problem. *Applied Mathematics and Computation* 2002; **127**:65–78.
5. Gohberg I, Hanke M, Koltracht I. Fast preconditioned conjugate gradient algorithms for Wiener-Hopf integral equations. *SIAM Journal on Numerical Analysis* 1994; **31**:429–443.
6. Moler C, Van Loan C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* 2003; **45**:3–49.
7. Bergamaschi L, Vianello M. Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numerical Linear Algebra with Applications* 2000; **7**:27–45.
8. Eiermann M, Ernst O. A restarted Krylov subspace method for the evaluation of matrix functions. *SIAM Journal on Numerical Analysis* 2006; **44**:2481–2504.
9. Frommer A, Simoncini V. Stopping criteria for rational matrix functions of Hermitian and symmetric matrices. *SIAM Journal on Scientific Computing* 2008; **30**:1387–1412.
10. Gallopoulos E, Saad Y. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM journal on scientific and statistical computing* 1992; **13**:1236–1264.
11. Hochbruck M, Lubich C. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis* 1997; **34**:1922–1925.
12. Lopez L, Simoncini V. Analysis of projection methods for rational function approximation to the matrix exponential. *SIAM Journal on Numerical Analysis* 2006; **44**:613–635.
13. Moret I. On RD-rational Krylov approximations to the core-functions of exponential integrators. *Numerical Linear Algebra with Applications* 2007; **14**:445–457.
14. Moret I, Novati P. RD-rational approximations of the matrix exponential. *BIT* 2004; **44**:595–615.
15. Popolizio M, Simoncini V. Acceleration techniques for approximating the matrix exponential operator. *SIAM Journal on Matrix Analysis and Applications* 2008; **30**:657–683.
16. Saad Y. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis* 1992; **29**:209–228.
17. Van Den Eshof J, Hochbruck M. Preconditioning Lanczos approximations to the matrix exponential. *SIAM Journal on Scientific Computing* 2006; **27**:1438–1457.
18. Gohberg I, Semencul A. On the inversion of finite Toeplitz matrices and their continous analogs. *Istoriko-Matematicheskie Issledovaniya* 1972; **2**:201–233.

19. Heinig G, Rost L. *Algebraic Methods for Toeplitz-like Matrices and Operators*. Birkhäuser Verlag, Basel: Switzerland, 1984.
20. Ng M, Sun H, Jin X. Recursive-based PCG methods for Toeplitz systems with nonnegative generating functions. *SIAM Journal on Scientific Computing* 2003; **24**:1507–1529.
21. Ammar G, Gragg W. Superfast solution of real positive definite Toeplitz systems. *SIAM Journal on Matrix Analysis and Applications* 1988; **9**:61–76.
22. Kailath T, Sayed AH, eds. *Fast Reliable Algorithms for Matrices with Structure*. SIAM: Philadelphia, 1999.
23. Strang G. A proposal for Toeplitz matrix calculations. *Studies in Applied Mathematics* 1986; **74**:171–176.
24. Carpenter A, Ruttan A, Varga R. Extened numerical computations on the $\frac{1}{9}$ conjecture in rational approximation theory, in *Rational Approximation and Interpolation, Lecture Notes in Mathematics. Graves-Morris P, Saff E, Varga R, eds.*, Springer-Verlag: Berlin, 1984, **1105**:383-411.
25. Higham NJ. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications* 2005; **26**:1179–1193.
26. Bai Z, Demmel J, Dongarra J, Ruhe A, Van Der Vorst HA. Eds. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM: Philadelphia, 2000.
27. Gockenbach M. *Partial Differential Equations-Analytial and Numerical Methods*. SIAM: Philadelphia, 2002.
28. Gradchtein IC, Reguk IM. *Tables of Integrals, Summation, Series and Derivatives*. Fizmatgiz: Moscow, 1971.
29. Betaman G, Ergelyi A. *Higher Transcedental Functions*. Nauka: Moscow, 1973.
30. Lee S, Pang H, Sun H. Shift-invert Arnoldi approximation to the Toeplitz matrix exponential. *SIAM Journal on Scientific Computing* 2010; **32**:774–792.