



---

*Research article*

## Algorithms for solving a class of real quasi-symmetric Toeplitz linear systems and its applications

Xing Zhang<sup>1,2</sup>, Xiaoyu Jiang<sup>1,\*</sup>, Zhaolin Jiang<sup>3,\*</sup> and Heejung Byun<sup>2</sup>

<sup>1</sup> School of Information Science and Technology, Linyi University, Linyi 276000, China

<sup>2</sup> College of Information Technology, The University of Suwon, Hwaseong-si 445-743, Republic of Korea

<sup>3</sup> School of Mathematics and Statistics, Linyi University, Linyi 276000, China

\* **Correspondence:** Email: [jxy19890422@sina.com](mailto:jxy19890422@sina.com), [jzh1208@sina.com](mailto:jzh1208@sina.com).

**Abstract:** In this paper, fast numerical methods for solving the real quasi-symmetric Toeplitz linear system are studied in two stages. First, based on an order-reduction algorithm and the factorization of Toeplitz matrix inversion, a sequence of linear systems with a constant symmetric Toeplitz matrix are solved. Second, two new fast algorithms are employed to solve the real quasi-symmetric Toeplitz linear system. Furthermore, we show a fast algorithm for quasi-symmetric Toeplitz matrix-vector multiplication. In addition, the stability analysis of the splitting symmetric Toeplitz inversion is discussed. In mathematical or engineering problems, the proposed algorithms are extraordinarily effective for solving a sequence of linear systems with a constant symmetric Toeplitz matrix. Fast matrix-vector multiplication and a quasi-symmetric Toeplitz linear solver are proven to be suitable for image encryption and decryption.

**Keywords:** image; encryption; decryption; decomposition; matrix inversion; quasi-symmetric Toeplitz matrix

---

### 1. Introduction

There have been extensive studies and applications of Toeplitz and quasi-Toeplitz matrices recently. Toeplitz or quasi-Toeplitz matrices are adopted as a core tool in the quantum magnetic field [1], generation of random bits [2], quantum key distribution [3–5], the evaluation of the short-time rate of change in the trend of CO<sub>2</sub> data [6], the deconvolution of hemodynamic responses [7] and the scattering and radiation on thin wire models [8]. Some studies have utilized Toeplitz or quasi-Toeplitz matrices to solve the ordinary and partial differential equations [9, 10], also applying them to solve the fractional differential equations [11–16].

A large number of linear systems with Toeplitz coefficient matrices need to be solved. Recently, Liu et al. developed fast solvers for Toeplitz linear systems [17–19]. The inversion of the Toeplitz matrix is represented as a combination of circulant and skew-circulant matrices in [20,21] so that the solution can be gained directly by using the fast Fourier transform (FFT) and the inverse FFT (IFFT). Here, we turn to express the Toeplitz inversion by using the sum products of the skew-imaginary circulant [22] and skew-circulant matrices, where the skew-imaginary circulant matrix is an  $\omega$ -circulant matrix with  $\omega = -i$ ,  $i = \sqrt{-1}$ .

Regarding the design of fast algorithms for a quasi-Toeplitz linear system, the authors of [23] presented new efficient algorithms to solve the CUPL-Toeplitz linear system [24–26] by first splitting the CUPL-Toeplitz matrix into a Toeplitz matrix and a low-rank matrix. In [27], Zhang et al. studied order-reduction algorithms to further reduce the computation complexity. The tridiagonal quasi-Toeplitz linear system is solved in [28].

We are concerned with a class of quasi-symmetric Toeplitz linear systems. The authors of [18] showed the trigonometric transformation splitting method to solve the real symmetric Toeplitz linear system. Different from the iterative method, we adopt the direct method based on the symmetric Toeplitz inversion representation, which is simplified due to the symmetry. We also show the numerical stability of the splitting symmetric Toeplitz inversion inspired by previous studies [26,29,30].

In this paper, we study an  $n \times n$  real symmetric Toeplitz matrix with perturbations in the first and last columns  $P = (p_{j,k})_{j,k=1}^n \in \mathbb{R}^{n \times n}$ :

$$p_{j,k} = \begin{cases} \varsigma_1 + t_1, & j = 2 \text{ and } k = 1, \\ \varsigma_2 + t_1, & j = n - 1 \text{ and } k = n, \\ t_{|j-k|}, & \text{otherwise.} \end{cases} \quad (1.1)$$

Obviously,

$$P = A + \alpha e_1^T + \beta e_n^T, \quad (1.2)$$

where  $\alpha = [0, \varsigma_1, 0, \dots, 0]^T$ ,  $e_1 = [1, 0, \dots, 0]^T$  is the first unit vector,  $\beta = [0, \dots, 0, \varsigma_2, 0]^T$ ,  $e_n = [0, \dots, 0, 1]^T$  is the last unit vector and  $A = (t_{|j-k|})_{j,k=1}^n$  is a nonsingular symmetric Toeplitz matrix.

An outline of this paper is as follows. Section 2 solves a sequence of linear systems with the same symmetric Toeplitz coefficient matrix. Section 3 presents the solvers for the quasi-symmetric Toeplitz linear system. A discussion of the quasi-symmetric Toeplitz matrix-vector multiplication is presented in Section 4. In Section 5, the stability analysis of the splitting inverse of the symmetric Toeplitz matrix is shown. Different numerical examples are given in Section 6. In Section 7, images are encrypted and decrypted by using the proposed quasi-symmetric Toeplitz matrix. Finally, we express our conclusions in Section 8.

## 2. Solving symmetric Toeplitz linear systems with the same coefficient matrix

The starting point of the algorithms derived in this paper is the following inversion formula of Toeplitz matrices:

**Lemma 1.** [31, p. 738] Let  $A = (t_{j-k})_{j,k=1}^n \in \mathbb{R}^{n \times n}$  be a Toeplitz matrix. If the vectors  $x = [x_1, x_2, \dots, x_n]^T$ ,  $x_1 \neq 0$  and  $y = [y_1, y_2, \dots, y_n]^T$  are the solutions of the linear systems

$$Ax = e_1 \quad \text{and} \quad Ay = e_n. \quad (2.1)$$

Then  $A$  is invertible and

$$A^{-1} = -\frac{1}{x_1(i+1)}(S_{I1}S_1 - S_{I2}S_2), \quad (2.2)$$

where  $S_{I1}$  and  $S_{I2}$  are skew-imaginary circulant matrices [22] holding the first columns as  $x = [x_1, x_2, \dots, x_n]^T$  and  $\tilde{y} = [iy_n, y_1, y_2, \dots, y_{n-1}]^T$  respectively;  $S_1$  and  $S_2$  are skew-circulant matrices with the first columns  $\hat{y} = [-y_n, y_1, y_2, \dots, y_{n-1}]^T$  and  $x = [x_1, x_2, \dots, x_n]^T$  respectively.

Note that we are concerned with a symmetric Toeplitz matrix  $A$ ; then, the special structure means that the solutions  $x$  and  $y$  in Eq (2.1) satisfy  $y = Jx$ , where the matrix  $J$  is an anti-identity matrix of order  $n$ .

If  $Ax = e_1$  has the solution  $x = [x_1, x_2, \dots, x_n]^T$  and  $x_1 \neq 0$ , then the symmetric Toeplitz matrix  $A$  is invertible and Eq (2.2) can be rewritten as

$$A^{-1} = \frac{1}{x_1(i+1)}(S_I S^T + iS_I^* S), \quad (2.3)$$

where  $S_I$  and  $S$  have the same first column  $x = [x_1, x_2, \dots, x_n]^T$  and the symbol  $S_I^*$  denotes the conjugate transpose of  $S_I$ .

Performing the diagonalization scheme on the skew-imaginary circulant matrix, that is,  $S_I = \Omega_n^* F_n^* \Lambda_{S_I} F_n \Omega_n$ , we can obtain

$$A^{-1} = \frac{1}{x_1(i+1)} \Omega_n^* F_n^* (\Lambda_{S_I} F_n \Omega_n S^T + i \Lambda_{S_I}^* F_n \Omega_n S), \quad (2.4)$$

where  $F_n = (F_{j,k})_{j,k=1}^n$ ,  $F_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i(j-1)(k-1)}{n}}$ ,  $1 \leq j, k \leq n$ ,  $\Omega_n = \text{diag}(1, e^{\frac{-3i\pi}{2n}}, \dots, e^{\frac{-3i(n-1)\pi}{2n}})$  and  $\Lambda_{S_I}$  is a diagonal matrix containing the eigenvalues of  $S_I$ .

Equation (2.4) shows a new decomposition of the symmetric Toeplitz inversion. By taking full account of the structural characteristics of the symmetric matrix, we solve only one system  $Ax = e_1$  instead of two linear systems.

In the case of solving a sequence of linear systems with the same coefficient matrix, it is effective to solve  $Ax = e_1$  first and then use the new decomposition of the inverse of the symmetric Toeplitz matrix for calculation. Therefore, Algorithm 1 is proposed for the computation of  $x$  and the eigenvalues of  $S_I$ . By applying Eq (2.4), an order-reduction algorithm [27] and FFT and IFFT operations, Algorithm 2 realizes the product of  $A^{-1}$  and  $v$  in the real number fields.

---

**Algorithm 1:** Compute  $x$  and  $\Lambda_{S_I}$

---

Step 1: Solve  $Ax = e_1$  by using any symmetric Toeplitz linear solver

Step 2:  $S_I$  and  $S$  have the same first column:  $x = [x_1, x_2, \dots, x_n]^T$

Step 3: The entries of  $\Lambda_{S_I}$  are computed from  $\sqrt{n}F_n\Omega_n x$  by applying an FFT

---

---

**Algorithm 2:** An algorithm for  $z = A^{-1}v$  in the real number field

---

Step 1: Calculate the vectors  $v_{S^T} = S^T v$  and  $v_S = S v$  by using Algorithm 1 in [27]

Step 2: Calculate  $\tilde{v} = \Lambda_{S_I} F_n \Omega_n v_{S^T} + i \Lambda_{S_I}^* F_n \Omega_n v_S$  by applying the FFT

Step 3: Calculate  $z = \frac{1}{x_1(i+1)} \Omega_n^* F_n^* \tilde{v}$  by applying Eq (2.4) and IFFT

---

In the proposed algorithms, the diagonal matrix can be represented by a vector containing the diagonal entries; then, the multiplication of the diagonal matrix and the vector is implemented by using a dot product, thus reducing the storage and computational complexity.

To solve a sequence of linear systems with a constant symmetric Toeplitz coefficient matrix, Algorithm 1 only needs to be computed once. The first step in Algorithm 1 is to solve the symmetric Toeplitz linear system  $Ax = e_1$ . Any symmetric Toeplitz linear solver can be used, such as the generalized minimal residual (GMRES) method, simplified quasi-minimal residual method, conjugate gradient (CG) method, etc. Since  $A$  is set as a symmetric positive definite matrix in our numerical experiments in this paper, we choose a the preconditioned CG (PCG) method with the Strang circulant preconditioner [32] to solve it. The workload of Algorithm 1 is considered to be  $O(n \log_2 n)$ .

One FFT or IFFT needs  $5n \log_2 n + O(n)$  real arithmetic operations [33, p. 75], and one run of Algorithm 1 in [27] needs  $\frac{15n}{2} \log_2 n + O(n)$ . Algorithm 2 mainly consists of two FFTs, one IFFT and two runs of Algorithm 1 in [27]. In the first step of Algorithm 2, it can save two FFTs with the length  $\frac{n}{2}$  when calculating  $S^T v$  and  $S v$  simultaneously. Finally, we regard the workload of Algorithm 2 as  $25n \log_2 n + O(n)$ . The application of the order-reduction algorithm reduces the entire computational complexity.

A sequence of Toeplitz linear systems with the same coefficient matrix is solved in [20, 21, 34]. The method of using the splitting Toeplitz matrix inversion greatly saves the computational time. This reminds us that in a mathematical or engineering problem, we may be faced with thousands of linear systems with the same symmetric Toeplitz matrix. By applying Algorithm 2, it will be more efficient to solve these symmetric Toeplitz linear systems.

### 3. Quasi-symmetric Toeplitz linear solvers

In this section, we solve a real quasi-symmetric Toeplitz linear system

$$Pa = b, \quad (3.1)$$

where  $a = [a_1, a_2, \dots, a_n]^T$  is an unknown vector and  $b = [b_1, b_2, \dots, b_n]^T$  is known. The coefficient matrix has a decomposition shown in Eq (1.2). We substitute Eq (1.2) into Eq (3.1) and multiply both sides of the equation by  $A^{-1}$  from the left; given that  $I_n$  is an identity matrix, we have

$$(I_n + A^{-1} \alpha e_1^T + A^{-1} \beta e_n^T) a = A^{-1} b. \quad (3.2)$$

Let  $A^{-1} \alpha = \mu$ ,  $A^{-1} \beta = \nu$  and  $A^{-1} b = \eta$ , where  $\mu = [\mu_1, \mu_2, \dots, \mu_n]^T$ ,  $\nu = [\nu_1, \nu_2, \dots, \nu_n]^T$  and  $\eta = [\eta_1, \eta_2, \dots, \eta_n]^T$ . The solution of Eq (3.2) can be

$$a = (I_n + \mu e_1^T + \nu e_n^T)^{-1} \eta. \quad (3.3)$$

Therefore, solving  $Pa = b$  is the same as calculating Eq (3.3). Based on the Sherman-Morrison-Woodbury (SMW) formula [35],  $(I_n + \mu e_1^T + \nu e_n^T)^{-1}$  can be represented as

$$(I_n + \mu e_1^T + \nu e_n^T)^{-1} = I_n - C, \quad (3.4)$$

where  $C = (c_{j,k})_{j,k=1}^n$  and

$$c_{j,k} = \begin{cases} \frac{(1+\nu_n)\mu_j - \mu_n\nu_j}{(1+\mu_1)(1+\nu_n) - \mu_n\nu_1}, & k = 1, \\ \frac{(1+\mu_1)\nu_j - \nu_1\mu_j}{(1+\mu_1)(1+\nu_n) - \mu_n\nu_1}, & k = n, \\ 0, & \text{otherwise.} \end{cases} \quad (3.5)$$

According to Eqs (3.3)–(3.5), we have

$$a = (I_n - C)\eta = (\eta_j - c_{j,1}\eta_1 - c_{j,n}\eta_n)_{j=1}^n. \quad (3.6)$$

To solve the linear system  $Pa = b$ , we must solve the symmetric Toeplitz systems  $A\mu = \alpha$ ,  $Av = \beta$  and  $A\eta = b$  at first. By using Algorithms 1 and 2 and Eqs (3.5) and (3.6), we obtain the following:

---

**Algorithm 3:** An algorithm to solve the real quasi-symmetric Toeplitz system  $Pa = b$

---

Step 1: Run Algorithm 1

Step 2: Calculate  $\mu = A^{-1}\alpha$ ,  $\nu = A^{-1}\beta$  and  $\eta = A^{-1}b$  by using Algorithm 2

Step 3: Calculate  $c_{j,k}$  by using Eq (3.5)

Step 4: Calculate  $a$  by using Eq (3.6)

---

Similarly, if we multiply Eq (1.2) by  $A^{-1}$  from the right, we can get  $P = (I_n + \alpha e_1^T A^{-1} + \beta e_n^T A^{-1})A$  and substitute it into Eq (3.1); then,

$$(I_n + \alpha e_1^T A^{-1} + \beta e_n^T A^{-1})Aa = b. \quad (3.7)$$

Denote  $e_1^T A^{-1} = \rho^T$ ,  $e_n^T A^{-1} = \sigma^T$  and  $Aa = \psi$ , where  $\rho = [\rho_1, \rho_2, \dots, \rho_n]^T$ ,  $\sigma = [\sigma_1, \sigma_2, \dots, \sigma_n]^T$  and  $\psi = [\psi_1, \psi_2, \dots, \psi_n]^T$ . Then, Eq (3.7) can be rewritten as

$$\psi = (I_n + \alpha\rho^T + \beta\sigma^T)^{-1}b. \quad (3.8)$$

The SMW formula is applied to Eq (3.8); we can write

$$(I_n + \alpha\rho^T + \beta\sigma^T)^{-1} = I_n - D, \quad (3.9)$$

where  $D = (d_{j,k})_{j,k=1}^n$  and

$$d_{j,k} = \begin{cases} \frac{(1+\sigma_{n-1}\varsigma_2)\varsigma_1\rho_k - \rho_{n-1}\varsigma_1\varsigma_2\sigma_k}{(1+\rho_2\varsigma_1)(1+\sigma_{n-1}\varsigma_2) - \sigma_2\rho_{n-1}\varsigma_1\varsigma_2}, & j = 2, \\ \frac{(1+\rho_2\varsigma_1)\varsigma_2\sigma_k - \sigma_2\varsigma_1\varsigma_2\rho_k}{(1+\rho_2\varsigma_1)(1+\sigma_{n-1}\varsigma_2) - \sigma_2\rho_{n-1}\varsigma_1\varsigma_2}, & j = n - 1, \\ 0, & \text{otherwise.} \end{cases} \quad (3.10)$$

From Eqs (3.8)–(3.10), we can get

$$\psi = (I_n - D)b = b - [0, \sum_{k=1}^n d_{2,k}b_k, 0, \dots, 0, \sum_{k=1}^n d_{n-1,k}b_k, 0]^T. \quad (3.11)$$

Since  $A$  is a symmetric Toeplitz matrix, if  $e_1^T A^{-1} = \rho^T$  and  $e_n^T A^{-1} = \sigma^T$ , then  $A\rho = e_1$  and  $A\sigma = e_n$ . According to Eqs (3.10) and (3.11) and the proposed algorithms in Section 2, we propose another algorithm for solving  $Pa = b$ .

---

**Algorithm 4:** An algorithm to solve the real quasi-symmetric Toeplitz system  $Pa = b$

---

Step 1: Run Algorithm 1;  $x$  equals to  $\rho$  because  $A\rho = e_1$

Step 2: Calculate  $\sigma = A^{-1}e_n$  according to Algorithm 2

Step 3: Calculate  $\psi$  by using Eqs (3.10) and (3.11)

Step 4: Calculate  $a = A^{-1}\psi$  according to Algorithm 2

---

From the above analysis, it is quite evident that the main work of Algorithms 3 and 4 is solving three symmetric Toeplitz linear systems. Moreover, we can omit one calculation of  $A\rho = e_1$  in Algorithm 4.

A greater number of symmetric Toeplitz linear systems will need to be solved if there are more perturbations in the coefficient matrix of the quasi-symmetric Toeplitz linear system. It is therefore meaningful to decompose the inverse of the symmetric Toeplitz matrix, as it will save computational time, especially in the case of high-order systems.

#### 4. Fast algorithm for the product of the quasi-symmetric Toeplitz matrix and vector

In terms of quasi-symmetric Toeplitz matrix-vector multiplication, the symmetric Toeplitz matrix  $A$  can be seen as the sum of the circulant and skew-circulant matrices [36], that is,

$$Pv = (A + \alpha e_1^T + \beta e_n^T)v = (C + S + \alpha e_1^T + \beta e_n^T)v, \quad (4.1)$$

where  $v = (v_1, v_2, \dots, v_n)$  is the vector and the circulant matrix  $C$  and the skew-circulant matrix  $S$  are all symmetric.

The first columns of circulant and skew-circulant matrices need to be computed in the splitting step. Considering the symmetry of  $n$ -th order  $A$ , the first columns of  $C$  and  $S$  can be obtained by applying the first  $\frac{n}{2} + 1$  ( $n$  is even) or  $\frac{n+1}{2}$  ( $n$  is odd) elements instead of  $n$  elements, which can reduce the computation complexity.

Based on the diagonalization scheme of the circulant matrix,  $C$  has the spectral decomposition  $C = F_n^* \Delta_C F_n$ , where  $F_n = (F_{j,k})_{j,k=1}^n$ ,  $F_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i(j-1)(k-1)}{n}}$ ,  $1 \leq j, k \leq n$ , and  $\Delta_C$  is a diagonal matrix containing the eigenvalues of  $C$ . In [27], an order-reduction algorithm for the multiplication of the real skew-circulant matrix and vector is proposed. So the product of the quasi-symmetric Toeplitz matrix and vector can be expressed as

$$Pv = F_n^* \Delta_C F_n v + Sv + \alpha v_1 + \beta v_n, \quad (4.2)$$

By applying Eq (4.2), the order-reduction algorithm for real skew-circulant matrix-vector multiplication, the FFT and IFFT operations, we give Algorithm 5 for  $Pv$  in the real number field as follows.

**Algorithm 5:** An algorithm for  $Pv$  based on the order-reduction algorithm in the real number field

- Step 1: Split the  $n \times n$  symmetric  $A$  into the symmetric  $C$  and symmetric  $S$ , where  $c_k$  and  $s_k$  are the first columns of  $C$  and  $S$ , respectively
- Step 2: The entries of  $\Delta_C$  are computed from  $\sqrt{n}F_n c_k$  by applying an FFT
- Step 3: Calculate  $z_1 = F_n^* \Delta_C F_n v$  by applying an FFT, IFFT
- Step 4: Calculate  $z_2 = S v$  by applying Algorithm 1 in [27]
- Step 5: Calculate  $Pv = z_1 + z_2 + \alpha v_1 + \beta v_n$

Because the complexity of one FFT or IFFT is  $5n \log_2 n + O(n)$  [33, p. 75], Algorithm 1 in [27] needs  $\frac{15n}{2} \log_2 n + O(n)$  real arithmetic operations. The complexity of Algorithm 5 is  $\frac{45n}{2} \log_2 n + O(n)$ , and it consists of one order-reduction algorithm, two FFTs and one IFFT.

### 5. Stability analysis

We find that the inverse factorization of the symmetric Toeplitz matrix is critical for the solution of quasi-symmetric Toeplitz linear equations. The following is the error analysis of Eq (2.3) in terms of the 1-norm,  $\infty$ -norm and 2-norm, respectively. Assume that  $\hat{x} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n]^T$  is the numerical solution of  $Ax = e_1$ . If  $\hat{x}_1 \neq 0$ , we denote

$$\hat{A}^{-1} = \frac{1}{\hat{x}_1(i+1)} (\hat{S}_I \hat{S}^T + i \hat{S}_I^* \hat{S}) \tag{5.1}$$

as a perturbation of  $A^{-1}$ , where the forms of  $\hat{S}_I$  and  $\hat{S}$  are the same as those in Eq (2.3) and have corresponding perturbations.

**Theorem 1.** Let  $\epsilon > 0$ ; if  $x_1 \neq 0$  and  $\hat{x}_1 \neq 0$ , suppose that the relative error of  $\frac{1}{x_1}$  is  $\hat{\epsilon} = \frac{|1/x_1 - 1/\hat{x}_1|}{|1/x_1|}$  and

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \epsilon; \tag{5.2}$$

we get

$$\|A^{-1} - \hat{A}^{-1}\|_1 \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \|x\|_1^2 \} \tag{5.3}$$

and

$$\frac{\|A^{-1} - \hat{A}^{-1}\|_1}{\|A^{-1}\|_1} \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \|x\|_1 \}. \tag{5.4}$$

*Proof.* From the representation of  $A^{-1}$  and  $\hat{A}^{-1}$  and Eqs (2.3) and (5.1), we can get

$$\begin{aligned} \|A^{-1} - \hat{A}^{-1}\|_1 &= \left\| \frac{1}{x_1(i+1)} (S_I S^T + i S_I^* S) - \frac{1}{\hat{x}_1(i+1)} (\hat{S}_I \hat{S}^T + i \hat{S}_I^* \hat{S}) \right\|_1 \\ &= \frac{1}{\sqrt{2}} \left\| \frac{1}{x_1} S_I S^T + \frac{i}{x_1} S_I^* S - \frac{1}{\hat{x}_1} \hat{S}_I \hat{S}^T - \frac{i}{\hat{x}_1} \hat{S}_I^* \hat{S} \right\|_1 \\ &\leq \frac{1}{\sqrt{2}} \left\| \left( \frac{1}{x_1} S_I \right) S^T - \left( \frac{1}{\hat{x}_1} \hat{S}_I \right) \hat{S}^T \right\|_1 + \frac{1}{\sqrt{2}} \left\| S_I^* \left( \frac{1}{x_1} S \right) - \hat{S}_I^* \left( \frac{1}{\hat{x}_1} \hat{S} \right) \right\|_1. \end{aligned} \tag{5.5}$$

On the one hand, we can get

$$\begin{aligned}
 \left\| \left( \frac{1}{x_1} S_I \right) S^T - \left( \frac{1}{\hat{x}_1} \hat{S}_I \right) \hat{S}^T \right\|_1 &= \left\| \left( \frac{1}{x_1} S_I \right) S^T - \left( \frac{1}{x_1} S_I \right) \hat{S}^T + \left( \frac{1}{x_1} S_I \right) \hat{S}^T - \left( \frac{1}{\hat{x}_1} \hat{S}_I \right) \hat{S}^T \right\|_1 \\
 &= \left\| \left( \frac{1}{x_1} S_I \right) (S^T - \hat{S}^T) + \left( \frac{1}{x_1} S_I - \frac{1}{\hat{x}_1} \hat{S}_I \right) \hat{S}^T \right\|_1 \\
 &\leq \left| \frac{1}{x_1} \right| \cdot \|S_I\|_1 \cdot \|S^T - \hat{S}^T\|_1 + \left\| \frac{1}{x_1} S_I - \frac{1}{\hat{x}_1} \hat{S}_I \right\|_1 \cdot \|\hat{S}^T\|_1.
 \end{aligned} \tag{5.6}$$

According to the structural characteristics of  $S_I$  and  $S$  and Eq (5.11), we note that

$$\|S_I\|_1 = \|x\|_1, \Delta \|S^T - \hat{S}^T\|_1 = \|x - \hat{x}\|_1 \leq \epsilon \|x\|_1, \Delta \|\hat{S}^T\|_1 = \|\hat{x}\|_1 \leq (1 + \epsilon) \|x\|_1. \tag{5.7}$$

$\hat{\epsilon} = \frac{|1/x_1 - 1/\hat{x}_1|}{|1/x_1|}$  is the relative error; then,

$$\begin{aligned}
 \left\| \frac{1}{x_1} S_I - \frac{1}{\hat{x}_1} \hat{S}_I \right\|_1 &= \left\| \frac{1}{x_1} x - \frac{1}{\hat{x}_1} \hat{x} \right\|_1 \\
 &= \left| \frac{1}{x_1} \right| \cdot \|x - \hat{x} + (1 - \frac{x_1}{\hat{x}_1}) \hat{x}\|_1 \\
 &\leq \left| \frac{1}{x_1} \right| (\|x - \hat{x}\|_1 + \hat{\epsilon} \|\hat{x}\|_1) \\
 &\leq \left| \frac{1}{x_1} \right| [\epsilon + (1 + \epsilon) \hat{\epsilon}] \cdot \|x\|_1.
 \end{aligned} \tag{5.8}$$

Combining Eqs (5.6)–(5.8), we thus obtain

$$\begin{aligned}
 \left\| \left( \frac{1}{x_1} S_I \right) S^T - \left( \frac{1}{\hat{x}_1} \hat{S}_I \right) \hat{S}^T \right\|_1 &\leq \left| \frac{1}{x_1} \right| \cdot \|x\|_1^2 \cdot \epsilon + \left| \frac{1}{x_1} \right| [\epsilon + (1 + \epsilon) \hat{\epsilon}] (1 + \epsilon) \cdot \|x\|_1^2 \\
 &= \left| \frac{1}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon) \hat{\epsilon}] (1 + \epsilon) \} \|x\|_1^2.
 \end{aligned} \tag{5.9}$$

Similarly, on the other hand,

$$\|S_I^* \left( \frac{1}{x_1} S \right) - \hat{S}_I^* \left( \frac{1}{\hat{x}_1} \hat{S} \right)\|_1 \leq \left| \frac{1}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon) \hat{\epsilon}] (1 + \epsilon) \} \|x\|_1^2. \tag{5.10}$$

Based on Eqs (5.5), (5.9) and (5.10), we can write

$$\|A^{-1} - \hat{A}^{-1}\|_1 \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon) \hat{\epsilon}] (1 + \epsilon) \} \|x\|_1^2.$$

By  $Ax = e_1$ , it is proven that  $\|x\|_1 \leq \|A^{-1}\|_1$ , so

$$\frac{\|A^{-1} - \hat{A}^{-1}\|_1}{\|A^{-1}\|_1} \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon) \hat{\epsilon}] (1 + \epsilon) \} \|x\|_1.$$



**Theorem 2.** Let  $\epsilon > 0$ ; if  $x_1 \neq 0$ ,  $\hat{x}_1 \neq 0$  and

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \epsilon, \quad (5.11)$$

then

$$\|A^{-1} - \hat{A}^{-1}\|_\infty \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \} \|x\|_1^2 \quad (5.12)$$

and

$$\frac{\|A^{-1} - \hat{A}^{-1}\|_\infty}{\|A^{-1}\|_\infty} \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \} \|x\|_1, \quad (5.13)$$

where  $\hat{\epsilon} = \frac{|1/x_1 - 1/\hat{x}_1|}{|1/x_1|}$  is the relative error of  $\frac{1}{x_1}$ .

*Proof.* Similar to Theorem 1, this proof is based on the same conditions:

$$\begin{aligned} \|x\|_1 &\leq \|A^{-1}\|_\infty, \quad \|S_I\|_\infty = \|x\|_1, \quad \|\hat{S}^T\|_\infty = \|\hat{x}\|_1 \leq (1 + \epsilon)\|x\|_1, \\ \|S^T - \hat{S}^T\|_\infty &= \|x - \hat{x}\|_1 \leq \epsilon\|x\|_1 \end{aligned}$$

and

$$\left\| \frac{1}{x_1} S_I - \frac{1}{\hat{x}_1} \hat{S}_I \right\|_\infty = \left\| \frac{1}{x_1} x - \frac{1}{\hat{x}_1} \hat{x} \right\|_1 \leq \left| \frac{1}{x_1} \right| \{ \epsilon + (1 + \epsilon)\hat{\epsilon} \} \cdot \|x\|_1.$$

**Theorem 3.** Under the assumptions and concepts of the previous two theorems, the upper bound of the 2-norm is said to be

$$\|A^{-1} - \hat{A}^{-1}\|_2 \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \} \|x\|_1^2 \quad (5.14)$$

and

$$\frac{\|A^{-1} - \hat{A}^{-1}\|_2}{\|A^{-1}\|_2} \leq \left| \frac{\sqrt{2n}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \} \|x\|_1. \quad (5.15)$$

*Proof.* As we all know,  $\|A^{-1} - \hat{A}^{-1}\|_2^2 \leq \|A^{-1} - \hat{A}^{-1}\|_1 \cdot \|A^{-1} - \hat{A}^{-1}\|_\infty$ ; from Eqs (5.3) and (5.12), we have

$$\|A^{-1} - \hat{A}^{-1}\|_2 \leq \left| \frac{\sqrt{2}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \} \|x\|_1^2.$$

Recall that

$$\|x\|_1 \leq \sqrt{n}\|x\|_2 = \sqrt{n}\|A^{-1}e_1\|_2 \leq \sqrt{n}\|A^{-1}\|_2\|e_1\|_2 = \sqrt{n}\|A^{-1}\|_2. \quad (5.16)$$

From Eqs (5.14) and (5.16), we can get

$$\frac{\|A^{-1} - \hat{A}^{-1}\|_2}{\|A^{-1}\|_2} \leq \left| \frac{\sqrt{2n}}{x_1} \right| \{ \epsilon + [\epsilon + (1 + \epsilon)\hat{\epsilon}](1 + \epsilon) \} \|x\|_1.$$

We show the stability analysis of inverse factorization of the symmetric Toeplitz matrix in this section. The absolute and relative perturbation upper bounds are shown in Eqs (5.3) and (5.4), (5.12) and (5.13), (5.14) and (5.15), as far as  $Ax = e_1$  is solvable.

## 6. Numerical simulations

In this section, we give some examples for the comparison of different algorithms. The first two examples present a comparison of different quasi-symmetric Toeplitz linear solvers. Examples 3 and 4 are devoted to the fast algorithm for the quasi-symmetric Toeplitz matrix-vector multiplication.

These experiments were done by using MATLAB (R2022a) on a laptop with the following specifications: 16 GB RAM, AMD Ryzen 7 5800H CPU 3.20 GHz. In the following tables, the calculation time is in seconds, “ $n$ ” denotes the matrix order and “—” indicates that it is out of MATLAB’s memory.

**Example 1.** An  $n \times n$  quasi-symmetric Toeplitz linear system  $Pa = b$  is considered in this example. According to Eq (1.2), the first column of  $A$  is  $(t_i)_{i=1}^n = \frac{1}{i}$ , and  $\varsigma_1$  and  $\varsigma_2$  are random values in the range of  $(0, 1)$ . Assume that  $a_{exact} = [1, 1, \dots, 1]^T$  is the exact solution, so the vector  $b$  is  $b = Pa_{exact}$ .

**Table 1.** Error and CPU time for Example 1 by different methods.

$n$	Back-slash		Alg <sub>Huang</sub> I		Algorithm 3		Alg <sub>Huang</sub> II		Algorithm 4	
	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
$2^{12}$	1.6531e-13	0.3730	9.7445e-05	0.0055	6.0940e-07	0.0042	7.1225e-05	0.0053	5.6413e-07	0.0042
$2^{13}$	3.6759e-13	2.1042	6.7327e-06	0.0110	2.4744e-06	0.0077	8.6325e-06	0.0105	1.7807e-06	0.0072
$2^{14}$	6.9422e-13	14.6981	1.0160e-05	0.0193	4.5065e-06	0.0149	1.2015e-05	0.0187	6.9125e-06	0.0145
$2^{15}$	—	—	1.7506e-05	0.0543	9.6450e-06	0.0374	1.7114e-05	0.0526	1.3520e-05	0.0368
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2^{23}$	—	—	3.3928e-04	22.3273	2.3274e-05	17.2194	6.6408e-04	20.4695	2.5144e-05	12.1401
$2^{24}$	—	—	5.0053e-04	40.9616	2.4198e-05	30.3279	4.4434e-04	38.0793	3.3228e-05	23.7662

A comparison of the error and time required to solve a quasi-symmetric Toeplitz linear system is presented in Table 1. The derived two algorithms for solving  $Pa = b$  are shown in the table as Algorithms 3 and 4. Based on Algorithm 3, Alg<sub>Huang</sub> I replaces Steps 1 and 2 with Huang’s algorithm [20, 21] for solving three symmetric Toeplitz systems. Similarly, Alg<sub>Huang</sub> II was executed based on Algorithm 4. The back-slash method means solving  $Pa = b$  directly by using the back-slash operator in MATLAB. “Error” was set to be  $\|\frac{a-a_{exact}}{a_{exact}}\|_{\infty}$ , and the stopping criterion for those algorithms, besides Back-slash, was set to be  $1 \times 10^{-7}$ .

Algorithms 3 and 4 have significant superiority over the other methods. They can solve high-order quasi-symmetric Toeplitz linear systems. However, Back-slash cannot work when the matrix order is above  $2^{14}$ . The proposed Algorithms 3 and 4 have the shortest computational time. They perform better than Alg<sub>Huang</sub> I and Alg<sub>Huang</sub> II in high-order linear systems due to the application of the order-reduction algorithm.

**Example 2.** Consider another quasi-symmetric Toeplitz linear system  $Pa = b$ , the first column of  $A$  and that  $\varsigma_1$  and  $\varsigma_2$  in Eq (1.2) are generated from an open interval  $(0, 1)$ . The sum of the elements in the first column was added to the diagonal entries of  $A$  to keep  $A$  as a diagonally dominant matrix. The exact solution of the system was set to be  $a_{exact} = [1, 1, \dots, 1]^T$ .

Table 2 shows a comparison of different methods for solving  $Pa = b$  for Example 2. Alg<sub>PGMRES</sub> refers to solving the linear system by using the GMRES method with the preconditioner. Since  $P$  is a nonsymmetric matrix, the GMRES method can be utilized to solve  $Pa = b$ . In order to accelerate the convergence rate, the Strang circulant preconditioner [32] of the symmetric Toeplitz matrix  $A$  was established.

**Table 2.** Error and CPU time for Example 2 by different methods.

$n$	Back-slash		Alg <sub>PGMRES</sub>		Algorithm 3		Algorithm 4	
	Error	Time	Error	Time	Error	Time	Error	Time
$2^{12}$	2.2649e-14	0.3658	2.7463e-07	0.0015	8.2994e-09	0.0051	4.2296e-09	0.0051
$2^{13}$	3.1530e-14	2.8237	6.8151e-08	0.0026	1.1341e-09	0.0088	6.1199e-10	0.0083
$2^{14}$	4.7296e-14	20.7760	2.0729e-08	0.0043	4.9841e-08	0.0150	9.4785e-08	0.0136
$2^{15}$	—	—	1.1276e-08	0.0091	1.2811e-08	0.0316	1.2697e-08	0.0301
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2^{23}$	—	—	1.7832e-08	2.7828	3.2381e-08	12.1427	3.8938e-08	7.1750
$2^{24}$	—	—	7.2746e-09	5.9250	1.0605e-08	24.9603	2.4069e-08	14.3739

According to theoretical analysis, the complexity of Algorithm 4 is lower than that of Algorithm 3. The results show that Algorithm 3 takes more time than Algorithm 4, which is consistent with the theoretical analysis. Alg<sub>PGMRES</sub> performs better than the other methods. However, compared with the algorithm for solving symmetric systems, the convergence analysis of the GMRES algorithm is very difficult, and there is no clear conclusion at present.

**Example 3.** Consider the multiplication of  $Pv$ . The vector  $v$  is Gaussian distributed. The quasi-symmetric Toeplitz matrix  $P$  is decomposed as Eq (1.2), where the first column of  $A$  is  $(t_i)_{i=1}^n = \frac{1}{i}$  and  $\zeta_1$  and  $\zeta_2$  are random values in the range  $(0, 1)$ .

**Table 3.** Computational time for  $Pv$  in Example 3.

$n$	$Pv$ -direct	$Pv$ -splitting	Algorithm 5
$2^{12}$	0.0052	5.9079e-04	6.0490e-04
$2^{13}$	0.0231	9.5132e-04	9.3785e-04
$2^{14}$	0.0924	0.0020	0.0017
$2^{15}$	—	0.0034	0.0032
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2^{22}$	—	0.5931	0.5177
$2^{23}$	—	1.2814	1.1760
$2^{24}$	—	2.4970	2.2437

In Table 3,  $Pv$ -direct denotes calculation of the quasi-symmetric Toeplitz matrix-vector multiplication via MATLAB.  $Pv$ -splitting refers to splitting  $P$  according to Eq (4.1) first, and then utilizing the diagonalization scheme of the circulant matrix and skew-circulant matrix for fast calculation. Algorithm 5 is the proposed quasi-symmetric Toeplitz matrix-vector multiplication. Obviously, Algorithm 5 consumed the least amount of time.

**Example 4.** Consider the multiplication of  $Pv$ , the first column of  $A$ ,  $\zeta_1$  and  $\zeta_2$  in Eq (1.2), and that the vector  $v$  are random values in the range  $(0, 1)$ . The sum of the elements in the first column was added to the diagonal entries of  $A$ .

Table 4 shows another example for the quasi-symmetric Toeplitz matrix-vector multiplication. As the matrix order increases, the efficiency of Algorithm 5 becomes increases. Our proposed algorithm is suitable for high-order matrix-vector multiplication. However, the  $Pv$ -direct method not only takes the longest time, but it also cannot work when the matrix order is above  $2^{14}$ .

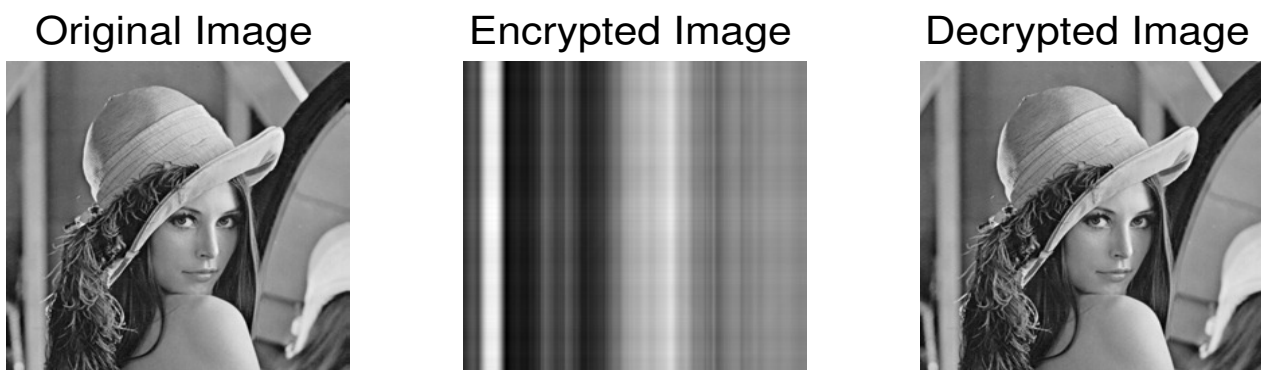
**Table 4.** Computational time for  $Pv$  in Example 4.

$n$	$Pv$ -direct	$Pv$ -splitting	Algorithm 5
$2^{12}$	0.0040	5.7744e-04	5.6916e-04
$2^{13}$	0.0191	9.7499e-04	9.3755e-04
$2^{14}$	0.0624	0.0018	0.0017
$2^{15}$	—	0.0034	0.0031
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2^{22}$	—	0.5712	0.5041
$2^{23}$	—	1.2831	1.1746
$2^{24}$	—	2.5153	2.2225

## 7. Applications

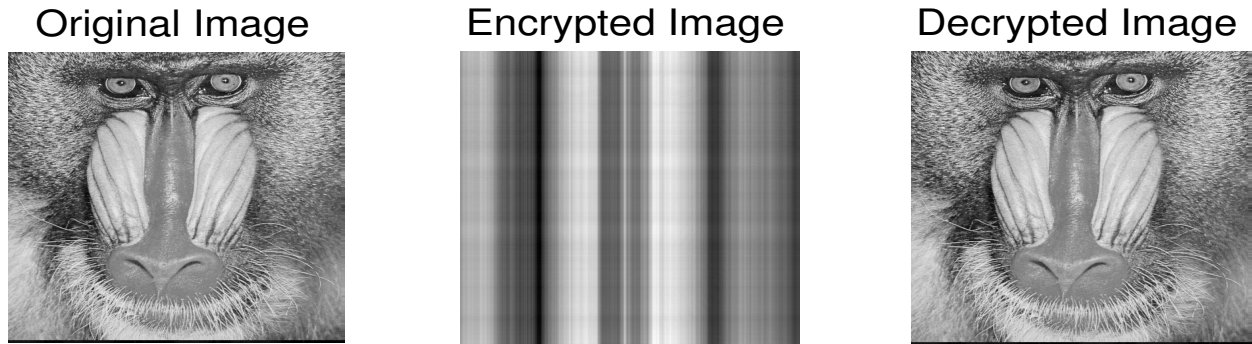
In the past decades, image encryption and decryption have been widely researched in the area of information security. The results show that the proposed algorithms can be used to encrypt and decrypt images.

**Example 1.** Based on Eq (1.2), we consider a quasi-symmetric Toeplitz matrix  $P$ , where the entries in the first column of  $A$  and  $\varsigma_1$  and  $\varsigma_2$  are random values in the range  $(0, 1)$ . To keep  $P$  as a diagonally dominant matrix, we added a parameter to the diagonal entries of  $P$ . Images were encrypted and decrypted by left-multiplying by  $P^2$  and  $P^{-2}$ , respectively.

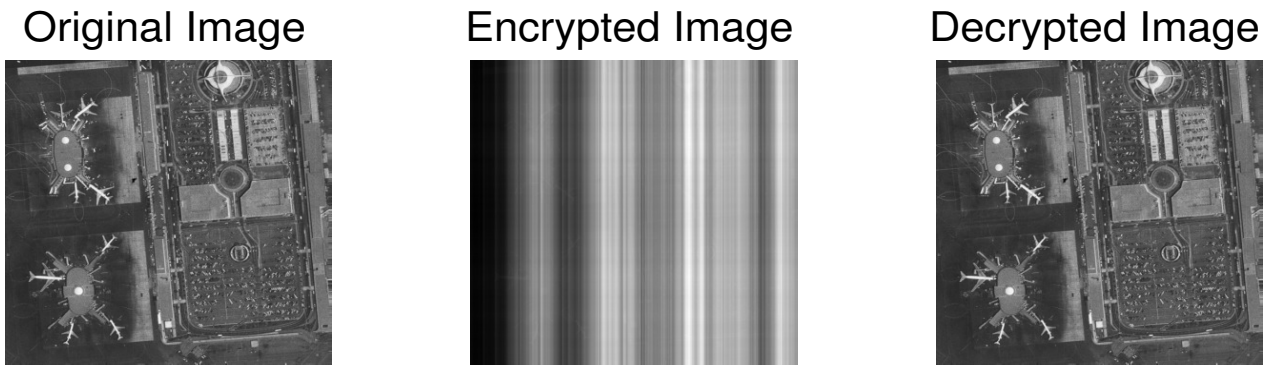


**Figure 1.**  $256 \times 256$  grayscale image for Example 1.

Figures 1– 3 present the effects of image encryption and decryption. The famous figure “Lenna” is taken as an example. Let the original image matrix be  $X = [x_1, x_2, \dots, x_{256}]$ ; the encrypted image matrix can be obtained by matrix-vector multiplication via Algorithm 5, that is,  $Y = [y_1, y_2, \dots, y_{256}] = P(P[x_1, x_2, \dots, x_{256}])$ . The decrypted image matrix can be computed by  $\hat{X} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{256}] = P^{-1}(P^{-1}[y_1, y_2, \dots, y_{256}])$ . Algorithm 4 was performed for image decryption because of its lower complexity than Algorithm 3.



**Figure 2.**  $512 \times 512$  grayscale image for Example 1.



**Figure 3.**  $1024 \times 1024$  grayscale image for Example 1.

In the process of image encryption and decryption, it is necessary to do multiple matrix-vector multiplications with the same quasi-symmetric Toeplitz matrix and solve linear systems with the same coefficient matrix. Some calculations can be done only once, like Steps 1 and 2 of Algorithm 5 in image encryption, as well as Steps 1 and 2 of Algorithm 4 in image decryption. So, the computational time will be saved.

It is obvious that our proposed algorithms can realize efficient encryption and decryption for different pixel-sized images.

## 8. Conclusions

Algorithms for solving a class of real quasi-symmetric Toeplitz linear systems are shown in this paper. We have proposed methods for solving a sequence of linear systems with the constant symmetric Toeplitz matrix based on the decomposition of Toeplitz matrix inversion. By splitting the coefficient matrix into a symmetric Toeplitz matrix plus two low-rank matrices and combining it with the SMW formula, two fast algorithms with  $O(n \log_2 n)$  complexity have been given to solve the real quasi-symmetric Toeplitz linear system. Quasi-symmetric Toeplitz matrix-vector multiplication in the real number field has been presented. Structural perturbation analysis of the inverse factorization of symmetric Toeplitz was also analyzed. Numerical simulations and applications have shown that the

proposed algorithms are accurate and efficient.

### Acknowledgements

The research was supported by the National Natural Science Foundation of China (Grant No. 12101284) and Natural Science Foundation of Shandong Province (Grant No. ZR2020MA051).

### Conflict of interest

The authors declare that there is no conflict of interest.

### References

1. M. M. Rams, M. Zwolak, B. Damski, A quantum phase transition in a quantum external field: superposing two magnetic phases, *Sci. Rep.*, **2** (2012), 655. <https://doi.org/10.1038/srep00655>
2. P. A. Papakonstantinou, D. P. Woodruff, G. Yang, True randomness from big data, *Sci. Rep.*, **6** (2016), 33740. <https://doi.org/10.1038/srep33740>
3. B. Y. Tang, B. Liu, Y. P. Zhai, C. Q. Wu, W. R. Yu, High-speed and large-scale privacy amplification scheme for quantum key distribution, *Sci. Rep.*, **9** (2019), 15733. <https://doi.org/10.1038/s41598-019-50290-1>
4. X. B. Wang, J. T. Wang, J. Q. Qin, C. Jiang, Z. W. Yu, Guessing probability in quantum key distribution, *npj Quantum Inf.*, **6** (2020), 45. <https://doi.org/10.1038/s41534-020-0267-3>
5. Y. A. Chen, Q. Zhang, T. Y. Chen, W. Q. Cai, S. K. Liao, J. Zhang, et al., An integrated space-to-ground quantum communication network over 4600 kilometres, *Nature*, **589** (2021), 214–219. <https://doi.org/10.1038/s41586-020-03093-8>
6. S. Nordebo, M. F. Naeem, P. Tans, Estimating the short-time rate of change in the trend of the keeling curve, *Sci. Rep.*, **10** (2020), 21222. <https://doi.org/10.1038/s41598-020-77921-2>
7. A. Machado, Z. Cai, T. Vincent, G. Pellegrino, J. M. Lina, E. Kobayashi, et al., Deconvolution of hemodynamic responses along the cortical surface using personalized functional near infrared spectroscopy, *Sci. Rep.*, **11** (2021), 5964. <https://doi.org/10.1038/s41598-021-85386-0>
8. X. M. Gu, T. Z. Huang, X. L. Zhao, H. B. Li, L. Li, Fast iterative solvers for numerical simulations of scattering and radiation on thin wires, *J. Electromagn. Waves Appl.*, **29** (2015), 1281–1296. <https://doi.org/10.1080/09205071.2015.1042559>
9. Z. Z. Bai, Y. M. Huang, M. K. Ng, On preconditioned iterative methods for certain time-dependent partial differential equations, *SIAM J. Numer. Anal.*, **47** (2009), 1019–1037. <https://doi.org/10.1137/080718176>
10. Z. Z. Bai, R. H. Chan, Z. R. Ren, On sinc discretization and banded preconditioning for linear third-order ordinary differential equations, *Numer. Linear Algebra Appl.*, **18** (2011), 471–497. <https://doi.org/10.1002/nla.738>
11. X. M. Gu, Y. L. Zhao, X. L. Zhao, B. Carpentieri, Y. Y. Huang, A note on parallel preconditioning for the all-at-once solution of Riesz fractional diffusion equations, *Numer. Math. Theory Methods Appl.*, **14** (2021), 893–919. <https://doi.org/10.48550/arXiv.2003.07020>

12. Y. L. Zhao, X. M. Gu, A. Ostermann, A preconditioning technique for an all-at-once system from Volterra subdiffusion equations with graded time steps, *J. Sci. Comput.*, **88** (2021), 11. <https://doi.org/10.1007/s10915-021-01527-7>
13. Z. Z. Bai, K. Y. Lu, Optimal rotated block-diagonal preconditioning for discretized optimal control problems constrained with fractional time-dependent diffusive equations, *Appl. Numer. Math.*, **163** (2021), 126–146. <https://doi.org/10.1016/j.apnum.2021.01.011>
14. Z. Z. Bai, K. Y. Lu, Fast matrix splitting preconditioners for higher dimensional spatial fractional diffusion equations, *J. Comput. Phys.*, **404** (2020), 1. <https://doi.org/10.1016/j.jcp.2019.109117>
15. W. H. Luo, X. M. Gu, Y. Liu, M. Jing, A Lagrange-quadratic spline optimal collocation method for the time tempered fractional diffusion equation, *Math. Comput. Simul.*, **182** (2021), 1–24. <https://doi.org/10.1016/j.matcom.2020.10.016>
16. M. Li, X. M. Gu, C. M. Huang, M. F. Fei, G. Y. Zhang, A fast linearized conservative finite element method for the strongly coupled nonlinear fractional Schrödinger equations, *J. Comput. Phys.*, **358** (2018), 256–282. <https://doi.org/10.1016/j.jcp.2017.12.044>
17. Z. Y. Liu, X. R. Qin, N. C. Wu, Y. L. Zhang, The shifted classical circulant and skew circulant splitting iterative methods for Toeplitz matrices, *Can. Math. Bull.*, **60** (2017), 807–815. <https://doi.org/10.4153/CMB-2016-077-5>
18. Z. Y. Liu, N. C. Wu, X. R. Qin, Y. L. Zhang, Trigonometric transform splitting methods for real symmetric Toeplitz systems, *Comput. Math. Appl.*, **75** (2018), 2782–2794. <https://doi.org/10.1016/j.camwa.2018.01.008>
19. Z. Y. Liu, S. H. Chen, W. J. Xu, Y. L. Zhang, The eigen-structures of real (skew) circulant matrices with some applications, *Comput. Appl. Math.*, **38** (2019), 178. <https://doi.org/10.1007/s40314-019-0971-9>
20. S. L. Lei, Y. C. Huang, Fast algorithms for high-order numerical methods for space fractional diffusion equations, *Int. J. Comput. Math.*, **94** (2016), 1062–1078. <http://dx.doi.org/10.1080/00207160.2016.1149579>
21. Y. C. Huang, S. L. Lei, A fast numerical method for block lower triangular Toeplitz with dense Toeplitz blocks system with applications to time-space fractional diffusion equations, *Numerical Algorithms*, **76** (2017), 605–616. <https://doi.org/10.1007/s11075-017-0272-6>
22. Z. L. Jiang, T. T. Xu, Norm estimates of  $\omega$ -circulant operator matrices and isomorphic operators for  $\omega$ -circulant algebra, *Sci. China Math.*, **59** (2016), 351–366. <https://doi.org/10.1007/s11425-015-5051-z>
23. Y. R. Fu, X. Y. Jiang, Z. L. Jiang, S. Jhang, Fast algorithms for finding the solution of CUPL-Toeplitz linear system from Markov chain, *Appl. Math. Comput.*, **396** (2021), 125859. <https://doi.org/10.1016/j.amc.2020.125859>
24. X. Y. Jiang, K. Hong, Skew cyclic displacements and inversions of two innovative patterned matrices, *Appl. Math. Comput.*, **308** (2017), 174–184. <https://doi.org/10.1016/j.amc.2017.03.024>
25. Y. P. Zheng, S. Shon, J. Kim, Cyclic displacements and decompositions of inverse matrices for CUPL Toeplitz matrices, *J. Math. Anal. Appl.*, **455** (2017), 727–741. <https://doi.org/10.1016/j.jmaa.2017.06.016>

26. Z. L. Jiang, X. T. Chen, J. M. Wang, The explicit inverses of CUPL-Toeplitz and CUPL-Hankel matrices, *East Asian J. Appl. Math.*, **7** (2017), 38–54. <https://doi.org/10.4208/eajam.070816.191016a>
27. X. Zhang, X. Y. Jiang, Z. L. Jiang, H. Byun, An improvement of methods for solving the CUPL-Toeplitz linear system, *Appl. Math. Comput.*, **421** (2022), 126932. <https://doi.org/10.1016/j.amc.2022.126932>
28. L. Du, T. Sogabe, S. L. Zhang, A fast algorithm for solving tridiagonal quasi-Toeplitz linear systems, *Appl. Math. Lett.*, **75** (2018), 74–81. <https://doi.org/10.1016/j.aml.2017.06.016>
29. P. P. Xie, Y. M. Wei, The stability of formulae of the Gohberg-Semencul-Trench type for Moore-Penrose and group inverses of Toeplitz matrices, *Linear Algebra Appl.*, **498** (2016), 117–135. <https://doi.org/10.1016/j.laa.2015.01.029>
30. J. Wu, X. M. Gu, Y. L. Zhao, Y. Y. Huang, B. Carpentieri, A note on the structured perturbation analysis for the inversion formula of Toeplitz matrices, *Jpn. J. Ind. Appl. Math.*, **40** (2023), 645–663. <https://doi.org/10.1007/s13160-022-00543-w>
31. I. Gohberg, V. Olshevsky, Circulants, displacements and decompositions of matrices, *Integr. Equations Oper. Theory*, **15** (1992), 730–743. <https://doi.org/10.1007/BF01200697>
32. R. Chan, X. Q. Jin, *An Introduction to Iterative Toeplitz Solvers*, SIAM, Philadelphia, 2007. <https://doi.org/10.1137/1.9780898718850>
33. R. E. Blahut, *Fast Algorithms for Signal Processing*, Cambridge University Press, New York, 2010. <https://doi.org/10.1017/CBO9780511760921>
34. H. Y. Jian, T. Z. Huang, X. M. Gu, X. L. Zhao, Y. L. Zhao, Fast implicit integration factor method for nonlinear space Riesz fractional reaction–diffusion equations, *J. Comput. Appl. Math.*, **378** (2020), 112935. <https://doi.org/10.1016/j.cam.2020.112935>
35. M. Batista, A. A. Karawia, The use of the Sherman-Morrison-Woodbury formula to solve cyclic block tri-diagonal and cyclic block penta-diagonal linear systems of equations, *Appl. Math. Comput.*, **210** (2009), 558–563. <https://doi.org/10.1016/j.amc.2009.01.003>
36. M. K. Ng, Circulant and skew-circulant splitting methods for Toeplitz systems, *J. Comput. Appl. Math.*, **159** (2003), 101–108. [https://doi.org/10.1016/S0377-0427\(03\)00562-4](https://doi.org/10.1016/S0377-0427(03)00562-4)



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)