Fast Approximate Inversion of A Block Triangular Toeplitz Matrix with Applications to Fractional Sub-Diffusion Equations *

Xin Lu † Hong-Kui Pang ‡ Hai-Wei Sun §

Abstract

A fast approximate inversion method is proposed for the block lower triangular Toeplitz with tri-diagonal blocks (BL3TB) matrix. The BL3TB matrix is approximated by a block ϵ-circulant matrix, which can be efficiently inverted using the fast Fourier transforms. The error estimation is given to show the high accuracy of the approximation. In applications, the proposed method is employed to solve the fractional sub-diffusion equation whose discretized matrix by a finite difference method is a BL3TB matrix. Numerical experiments are carried out to demonstrate the efficiency of the proposed method.

Key words: Block triangular Toeplitz matrix; Block ϵ-circulant matrix; Fourier transform; Fractional sub-diffusion equations.

Mathematics Subject Classification: 65L05; 65N22; 65F10; 65F15

1 Introduction

We study the linear system

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{1.1}$$

*The research was supported by research grants MYRG102(Y1-L3)-FST13-SHW from University of Macau and 105/2012/A3 from FDCT of Macao.

†Department of Mathematics, University of Macau, Macao (luxin1024@126.com).

‡School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou, 221116, Jiangsu, China (panghongkui@163.com). This author was supported by the National Natural Science Foundation of China under grant 11201192, the Natural Science Foundation of Jiangsu Province under grant BK2012577, and the Natural Science Foundation for Colleges and Universities in Jiangsu Province under grant 12KJB110004.

§Department of Mathematics, University of Macau, Macao (HSun@umac.mo).

where $\mathbf{A}$ is a block lower triangular Toeplitz with tri-diagonal blocks (BL3TB) matrix of the form

$$\mathbf{A} = \begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & \ddots & \ddots & \\ A_{n-1} & \dots & A_1 & A_0 \end{bmatrix}, \tag{1.2}$$

in which all A_j, $j = 0, \dots, n-1$, are m-by-m tri-diagonal matrices, $\mathbf{u}$ is the unknown vector, and $\mathbf{b}$ is the right hand side.

Toeplitz matrices emerge from numerous topics such as signal and image processing, numerical solutions of partial differential equations and integral equations, as well as queueing networks; see [8, 9] and the references therein. Among them, the triangular Toeplitz matrix plays a key role in the displacement representation of general Toeplitz matrices, which is fundamental in the study of structured matrices and polynomial computations; see [5, 6, 17, 21]. In particular, the coefficient matrices of fractional sub-diffusion equations discretized by finite difference schemes can be written as BL3TB matrices; see [13, 15, 31] and Section 4.

The forward substitution method [16] can be straightly applied to solve the scalar triangular Toeplitz matrices ($\mathbf{A}$ in (1.2) with $m = 1$) with $\mathcal{O}(n^2)$ arithmetic operations. By making use of the Toeplitz structure and the fast Fourier transform (FFT), an exact inversion method, which is based on the divide-and-conquer strategy, is employed to invert the scalar triangular Toeplitz matrices [14], whose complexity is of $\mathcal{O}(n \log n)$. The forward substitution method can be extended to solve the BL3TB linear system (1.1), named as the block forward substitution method [16], with $\mathcal{O}(mn^2)$ complexity and $\mathcal{O}(mn)$ storage requirement. The divide-and-conquer method also can be extended to solve the BL3TB linear system (1.1). Nevertheless, its computational cost is of $\mathcal{O}(m^2 n \log n + m^3 n)$ and storage requirement is of $\mathcal{O}(m^2 n)$ [5] since the inverse of a tri-diagonal matrix is usually dense. Therefore, the divide-and-conquer method may not be better than the block forward substitution method for solving the BL3TB system if the block size m is large.

Another type of inversion method for the scalar triangular Toeplitz matrix is the approximate inversion method, which was firstly proposed by Bini [4]. The approximate inversion method to invert the scalar triangular Toeplitz matrix also requires $\mathcal{O}(n \log n)$ operations with better parallel performance [4, 19]. To our acknowledge, the approximate inversion method has never been extended to treat the block triangular Toeplitz matrix in the literature.

The main aim of this paper is to propose an approximate inversion method, which is developed from Bini's method [4], to solve the BL3TB linear system (1.1). More precisely, the BL3TB matrix $\mathbf{A}$ in (1.2) is firstly approximated by a block ϵ-circulant matrix $\mathbf{A}_\epsilon$. Using the Fourier matrix and diagonal matrix together, $\mathbf{A}_\epsilon$ can be block-diagonalized in $\mathcal{O}(mn \log n)$ operations. Furthermore, this block diagonal matrix also preserves the tri-diagonal block structure

and hence can be solved in $\mathcal{O}(mn)$ operations. Therefore, the total computational complexity for inverting $\mathbf{A}_\epsilon$ is only of $\mathcal{O}(mn \log n)$ arithmetic operations with $\mathcal{O}(mn)$ storage requirement. Theoretically, a sufficient condition is given to guarantee the invertibility of $\mathbf{A}_\epsilon$, and the error estimation between $\mathbf{A}^{-1}$ and $\mathbf{A}_\epsilon^{-1}$ is obtained to ensure the high accurate approximation.

In applications, the proposed algorithm is employed to solve fractional sub-diffusion equations, which arise in research topics including modeling chaotic dynamics of classical conservative systems [27], groundwater contaminant transport [2, 3], turbulent flow [7, 25], biology [20], finance [23], image processing [1], and physics [26]. Resulting matrices of fractional sub-diffusion equations discretized by finite difference schemes [13, 15, 31] are shown to be BL3TB; see Section 4. Employing the traditional time-marching algorithm (the same as the block forward substitution method) to solve the resulting systems step by step requires $\mathcal{O}(mn^2)$ operations. Nevertheless, utilizing the proposed method, the computational complexity can be reduced to only $\mathcal{O}(mn \log n)$ arithmetic operations. Numerical experiments are given to demonstrate the efficiency of the proposed method.

The rest of the paper is organized as follows. In Section 2, we propose a fast approximate inversion algorithm for the BL3TB matrix. Some properties of the BL3TB matrix and the error estimation are given in Section 3. In Section 4, we show the coefficient matrices of fractional sub-diffusion equations discretized by finite difference schemes are actually BL3TB and hence the proposed method can be exploited to solve them efficiently. Numerical results are reported in Section 5. At the end, concluding remarks are given in Section 6.

2 Approximate inversion method

Let $H_z = \left[h_{i,j}^{(z)}\right]_{i,j=1}^{n}$ be a matrix whose entries are all zero except that $h_{i,i-1}^{(z)} = 1$ for $i = 2, 3, \ldots, n$ and $h_{1,n}^{(z)} = z$ with a scalar z. It is easy to verify that for $k = 0, 1, \ldots, n$

$$H_z^k = \begin{bmatrix} 0 & zI_k \\ I_{n-k} & 0 \end{bmatrix}, \tag{2.1}$$

where I_k is the identity matrix of order k. In particular,

$$H_0^k = \begin{bmatrix} 0 & 0 \\ I_{n-k} & 0 \end{bmatrix}.$$

Therefore, the BL3TB matrix $\mathbf{A}$ in (1.2) can be written as

$$\mathbf{A} = \sum_{j=0}^{n-1} H_0^j \otimes A_j, \tag{2.2}$$

where " $\otimes$ " denotes the Kronecker tensor product.

We follow Bini's initial idea [4] to replace H_0 by H_ϵ with $\epsilon > 0$ in (2.2). As a consequence, the matrix $\mathbf{A}$ is approximated by

$$\mathbf{A}_\epsilon \equiv \sum_{j=0}^{n-1} H_\epsilon^j \otimes A_j = \begin{bmatrix} A_0 & \epsilon A_{n-1} & \dots & \epsilon A_2 & \epsilon A_1 \\ A_1 & A_0 & \epsilon A_{n-1} & \dots & \epsilon A_2 \\ \vdots & A_1 & A_0 & \ddots & \vdots \\ A_{n-2} & \dots & \ddots & \ddots & \epsilon A_{n-1} \\ A_{n-1} & A_{n-2} & \dots & A_1 & A_0 \end{bmatrix}, \tag{2.3}$$

where $\mathbf{A}_\epsilon$ with $\epsilon > 0$ is *a block ϵ-circulant matrix*; see [5].

Let $D_\delta = \mathrm{diag}(1, \delta, \dots, \delta^{n-1})$ with $\delta = \sqrt[n]{\epsilon}$ be a diagonal matrix and the n-by-n Fourier matrix be given by

$$F_n = \frac{1}{\sqrt{n}} \left[\omega^{(i-1)(j-1)}\right]_{i,j=1}^{n}, \qquad \omega = \exp\left(\frac{2\pi \mathbf{i}}{n}\right), \qquad \mathbf{i} \equiv \sqrt{-1}.$$

We note that the block ϵ-circulant matrix can be simultaneously block diagonalized by means of a combination of F_n and D_δ. The following theorem gives the exact formula.

Theorem 1 (see [5, Theorem 2.10]) *Let $\mathbf{A}_\epsilon$ be the block ϵ-circulant matrix given by* (2.3). *Then*

$$\mathbf{A}_\epsilon = \left[(D_\delta^{-1} F_n^*) \otimes I_m\right] \mathrm{diag}\left(\Lambda_0, \Lambda_1, \dots, \Lambda_{n-1}\right) \left[(F_n D_\delta) \otimes I_m\right], \tag{2.4}$$

where Λ_k, $k = 0, 1, \dots, n-1$, are $m \times m$ matrices and satisfy that

$$\begin{bmatrix} \Lambda_0 \\ \Lambda_1 \\ \vdots \\ \Lambda_{n-1} \end{bmatrix} = \left[(\sqrt{n} F_n D_\delta) \otimes I_m\right] \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{n-1} \end{bmatrix}. \tag{2.5}$$

From (2.5), it is easy to check that

$$\Lambda_k = \sum_{j=0}^{n-1} \delta^j \omega^{kj} A_j, \quad k = 0, 1, \dots, n-1. \tag{2.6}$$

Note that A_j are $m \times m$ tri-diagonal matrices for $j = 0, 1, \dots, n-1$, therefore Λ_k are also $m \times m$ tri-diagonal matrices for $k = 0, 1, \dots, n-1$. Hence there are only $\mathcal{O}(m)$ nonzero entries (of three diagonals) in each Λ_k. Furthermore, using the FFT, all $m \times m$ tri-diagonal matrices Λ_k can be constructed in $\mathcal{O}(mn \log n)$ arithmetic operations.

Now we consider the BL3TB linear system (1.1). Note that the matrix $\mathbf{A}_\epsilon$ can be arbitrarily close to $\mathbf{A}$ as the small positive number ϵ tends to zero. Thus, if both $\mathbf{A}$ and $\mathbf{A}_\epsilon$ are nonsingular,

it is reasonable to regard $\mathbf{A}_\epsilon^{-1}$ as a good approximation of $\mathbf{A}^{-1}$ when ϵ is small enough. We emphasize that in practical computation, the value of ϵ cannot be too small since we have to invert the diagonal matrix D_δ where ϵ is involved. An appropriate ϵ will be chosen in Section 5 as in [19].

From (2.4), we immediately have

$$\mathbf{A}_\epsilon^{-1} = \left[(D_\delta^{-1}F_n^*) \otimes I_m\right] \operatorname{diag}\left(\Lambda_0^{-1}, \Lambda_1^{-1}, \ldots, \Lambda_{n-1}^{-1}\right) \left[(F_n D_\delta) \otimes I_m\right], \tag{2.7}$$

which shows that $\mathbf{A}_\epsilon^{-1}$ is also a block ϵ-circulant matrix. Therefore, the solution of (1.1) can be approximated as

$$\mathbf{u} = \mathbf{A}^{-1}\mathbf{b} \approx \mathbf{A}_\epsilon^{-1}\mathbf{b} \equiv \mathbf{u}_\epsilon. \tag{2.8}$$

In practical implementation, using (2.7), the algorithm to obtain the approximate solution $\mathbf{u}_\epsilon = \mathbf{A}_\epsilon^{-1}\mathbf{b}$ in (2.8) can be written as follows.

Algorithm 1: Approximate inversion method for solving the BL3TB system

1. Input: ϵ, $\mathbf{b}$, and tri-diagonal matrices $A_0, A_1, \ldots, A_{n-1}$
2. Compute $\delta = \sqrt[n]{\epsilon}$ and $D_\delta = \operatorname{diag}(1, \delta, \ldots, \delta^{n-1})$
3. Compute tri-diagonal matrices Λ_k for $k = 0, 1, \ldots, n-1$ by using (2.5)
4. Compute $\mathbf{b}_\epsilon = \left[(F_n D_\delta) \otimes I_m\right] \mathbf{b}$
5. Solve $\operatorname{diag}(\Lambda_0, \ldots, \Lambda_{n-1})\widetilde{\mathbf{b}}_\epsilon = \mathbf{b}_\epsilon$
6. Compute $\mathbf{u}_\epsilon = \left[(D_\delta^{-1}F_n^*) \otimes I_m\right] \widetilde{\mathbf{b}}_\epsilon$

Thus the approximate solution $\mathbf{u}_\epsilon$ can be computed in $\mathcal{O}(mn \log n)$ operations with only $\mathcal{O}(mn)$ storage requirement.

From Algorithm 1, we see that the approximate inversion method (2.7) and (2.8) for solving the BL3TB linear system (1.1) is easy to carry out. Nevertheless, from the theoretical point of view, two important issues in the proposed method should be studied: the invertibility of $\mathbf{A}_\epsilon$ and the error estimation between $\mathbf{u}$ and $\mathbf{u}_\epsilon$. Those will be investigated in the following section.

3 Invertibility and error estimation

In this section, we study the invertibility of $\mathbf{A}_\epsilon$ and estimate the difference between $\mathbf{A}_\epsilon^{-1}$ and $\mathbf{A}^{-1}$. All theoretical analyses are available to general block lower triangular Toeplitz matrices, not only to BL3TB matrices.

We first introduce some concepts for supplementing our work.

Definition 1 (see [24]) *Suppose that $B = [b_{i,j}]_{m\times m}$ and $C = [c_{i,j}]_{m\times m}$ are $m \times m$ real matrices.*

i) *B is nonnegative; i.e., $B \geq 0 \iff b_{i,j} \geq 0$ for $1 \leq i, j \leq m$;*

ii) $B \geq C \iff B - C \geq 0$;

iii) $|B| = [|b_{i,j}|]_{m\times m}$, *where* $|\cdot|$ *denotes the absolute value;*

iv) $C = \alpha I_m - B$ *is a nonsingular M-matrix if* $B \geq 0$ *and* $\rho(B) < \alpha$, *where* $\rho(B)$ *is the spectral radius of* B.

In addition, we also need definitions of *matrix power series* and *matrix polynomial*, which are analogous to [5] and will come into use afterward.

Definition 2 (see [5]) *Regardless the convergence, a matrix power series is defined as*

$$\Theta(z) \equiv \sum_{j=0}^{\infty} A_j z^j, \quad z \in \mathbb{C},$$

provided that $\{A_j\}_{j=0}^{\infty}$ *is a sequence of* $m \times m$ *matrices. Moreover, a matrix polynomial is given by*

$$\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j.$$

Obviously, a matrix polynomial is a special matrix power series with all $A_j = 0$ for $j \geq n$.

In order to investigate the convergence of a matrix power series, we introduce a Wiener algebra which plays an important role in our study. The *Wiener algebra* is denoted by the set $\mathcal{W}$ of matrix power series $\Theta(z) = \sum_{j=0}^{\infty} A_j z^j$ in which $\sum_{j=0}^{\infty} |A_j| < \infty$.

There is a link between the matrix power series and the block lower triangular Toeplitz matrices. We call $\Theta(z) = \sum_{j=0}^{+\infty} A_j z^j$ the *associate matrix power series* of a block lower triangular Toeplitz matrix $\mathbf{A}$, denoted by $\mathcal{T}_n[\Theta(z)]$, if its first block column is a sequence of $m \times m$ matrices $\{A_i\}_{i=0}^{n-1}$. More precisely,

$$\mathbf{A} = \begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & \ddots & \ddots & \\ A_{n-1} & \dots & A_1 & A_0 \end{bmatrix} = \mathcal{T}_n[\Theta(z)] = \mathcal{T}_n \left[\sum_{j=0}^{+\infty} A_j z^j \right]. \tag{3.1}$$

Using (2.2) and (3.1), we have

$$\mathbf{A} = \mathcal{T}_n \left[\sum_{j=0}^{+\infty} A_j z^j \right] = \mathcal{T}_n \left[\sum_{j=0}^{n-1} A_j z^j \right] = \sum_{j=0}^{n-1} H_0^j \otimes A_j, \tag{3.2}$$

where H_0 is defined in (2.1). The corresponding block ϵ-circulant matrix of $\mathbf{A}$, as in (2.3), is denoted by

$$\mathbf{A}_\epsilon \equiv \mathcal{C}_\epsilon[\mathbf{A}] = \mathcal{C}_\epsilon \left[\sum_{j=0}^{n-1} H_0^j \otimes A_j \right] = \sum_{j=0}^{n-1} H_\epsilon^j \otimes A_j. \tag{3.3}$$

3.1 Invertibility of block ϵ-circulant matrix

From (3.1), it is obvious that the matrix $\mathbf{A}$ is nonsingular if and only if A_0 is nonsingular. Nevertheless, it is uncertain if $\mathbf{A}_\epsilon$ in (3.3) is still nonsingular even if $\mathbf{A}$ is nonsingular. In order to guarantee the invertibility of $\mathbf{A}_\epsilon$, we verify the following Theorem.

Theorem 2 *Let the matrix polynomial* $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ *and* $\mathbf{A} = \mathcal{T}_n[\Theta_n(z)]$. *If* $\Theta_n(z)$ *is nonsingular for* $|z| \leq 1$, *then the block* ϵ*-circulant matrix* $\mathbf{A}_\epsilon = \mathcal{C}_\epsilon[\mathbf{A}]$ *given by* (3.3) *is invertible for* $0 < \epsilon < 1$, *and*

$$\mathbf{A}_\epsilon^{-1} = \left[(D_\delta^{-1} F_n^*) \otimes I_m\right] \operatorname{diag}\left([\Theta_n(\delta)]^{-1}, [\Theta_n(\delta\omega)]^{-1}, \ldots, [\Theta_n(\delta\omega^{n-1})]^{-1}\right) \left[(F_n D_\delta) \otimes I_m\right], \tag{3.4}$$

where $D_\delta = \operatorname{diag}(1, \delta, \ldots, \delta^{n-1})$, $\delta = \sqrt[n]{\epsilon}$, *and* $\omega = \exp\left(\frac{2\pi \mathbf{i}}{n}\right)$.

Proof: From (2.4), it is obvious that the matrix $\mathbf{A}_\epsilon$ is nonsingular if and only if all Λ_k are nonsingular for $k = 0, 1, \ldots, n-1$.

Using (2.6), for each k, we have

$$\Lambda_k = \sum_{j=0}^{n-1} \delta^j \omega^{kj} A_j = \Theta_n(\delta\omega^k).$$

Note that $|\delta\omega^k| \leq 1$ and hence $\Theta_n(\delta\omega^k)$ is nonsingular. As a consequence, all Λ_k are nonsingular. Thus $\mathbf{A}_\epsilon$ is invertible and (3.4) holds. □

Theorem 2 gives a sufficient condition which guarantees the invertibility of $\mathbf{A}_\epsilon$. Nevertheless, it is not easy to check if a given matrix polynomial $\Theta_n(z)$ is nonsingular for all $|z| \leq 1$. The following lemma provides an equivalent condition which concerns the invertibility of the matrix power series and may be easier to test.

Lemma 3 (see [5, Theorem 3.2]) *Suppose that the matrix power series* $\Theta(z) \in \mathcal{W}$. *Then* $\Theta(z)$ *is invertible with* $[\Theta(z)]^{-1} \in \mathcal{W}$ *if and only if* $\Theta(z)$ *is nonsingular for* $|z| \leq 1$.

In general, the inverse of an invertible matrix power series $\Theta(z) \in \mathcal{W}$ may not belong to $\mathcal{W}$ anymore. Lemma 3 gives a sufficient and necessary condition which ensures $[\Theta(z)]^{-1} \in \mathcal{W}$. We comment that the matrix polynomial $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ is a special matrix power series which belongs to $\mathcal{W}$. Therefore, by Lemma 3 and Theorem 2, the following corollary is immediately obtained.

Corollary 4 *If the matrix polynomial* $\Theta_n(z)$ *is invertible with* $[\Theta_n(z)]^{-1}$ *in* $\mathcal{W}$ *and* $\mathbf{A} = \mathcal{T}_n[\Theta_n(z)]$, *then* $\mathbf{A}_\epsilon = \mathcal{C}_\epsilon[\mathbf{A}]$ *given by* (3.3) *is nonsingular for* $0 < \epsilon < 1$ *and* (3.4) *holds.*

Assume that the matrix polynomial $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ is invertible with $[\Theta_n(z)]^{-1} \in \mathcal{W}$. Then its inverse can be expressed as a matrix power series

$$[\Theta_n(z)]^{-1} = \sum_{j=0}^{+\infty} B_j z^j, \tag{3.5}$$

where $\{B_j\}_{j=0}^{\infty}$ is a sequence of $m \times m$ matrices which satisfies the following recursive formula,

$$\begin{cases} B_0 = A_0^{-1}, \\ B_k = -A_0^{-1} \sum_{j=1}^{k} A_j B_{k-j}, \quad \text{for} \quad k \geq 1. \end{cases} \tag{3.6}$$

Let

$$\mathbf{B} = \mathcal{T}_n \left[[\Theta_n(z)]^{-1} \right] = \mathcal{T}_n \left[\sum_{j=0}^{\infty} B_j z^j \right] = \begin{bmatrix} B_0 & & & \\ B_1 & B_0 & & \\ \vdots & \ddots & \ddots & \\ B_{n-1} & \dots & B_1 & B_0 \end{bmatrix} = \sum_{j=0}^{n-1} H_0^j \otimes B_j. \tag{3.7}$$

It is easy to check that $\mathbf{B} = \mathbf{A}^{-1}$ by (3.6) and (3.7).

Under the assumption that $\Theta_n(z)$ is invertible with $[\Theta_n(z)]^{-1} \in \mathcal{W}$, we know that $\mathbf{A}_\epsilon$ is invertible for $0 < \epsilon < 1$. From (3.4) and Theorem 1, it is obvious that $\mathbf{A}_\epsilon^{-1}$ is also a block ϵ-circulant matrix. Suppose that $\mathbf{A}_\epsilon^{-1}$ has the form

$$\mathbf{A}_\epsilon^{-1} \equiv \begin{bmatrix} B_0^{(\epsilon)} & \epsilon B_{n-1}^{(\epsilon)} & \dots & \epsilon B_2^{(\epsilon)} & \epsilon B_1^{(\epsilon)} \\ B_1^{(\epsilon)} & B_0^{(\epsilon)} & \epsilon B_{n-1}^{(\epsilon)} & \dots & \epsilon B_2^{(\epsilon)} \\ \vdots & B_1^{(\epsilon)} & B_0^{(\epsilon)} & \ddots & \vdots \\ B_{n-2}^{(\epsilon)} & \dots & \ddots & \ddots & \epsilon B_{n-1}^{(\epsilon)} \\ B_{n-1}^{(\epsilon)} & B_{n-2}^{(\epsilon)} & \dots & B_1^{(\epsilon)} & B_0^{(\epsilon)} \end{bmatrix} = \sum_{j=0}^{n-1} H_\epsilon^k \otimes B_j^{(\epsilon)}. \tag{3.8}$$

The following theorem provides an exact expression for each block $B_k^{(\epsilon)}$ of $\mathbf{A}_\epsilon^{-1}$, $k = 0, 1, \dots, n-1$.

Theorem 5 *Assume that* $\mathbf{A} = \mathcal{T}_n[\Theta_n(z)]$, *where* $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ *is invertible with* $\Phi(z) = [\Theta_n(z)]^{-1} = \sum_{j=0}^{+\infty} B_j z^j \in \mathcal{W}$. *Then* $\mathbf{A}_\epsilon$ *is nonsingular for* $0 < \epsilon < 1$ *and* $\mathbf{A}_\epsilon^{-1}$ *in* (3.8) *satisfies*

$$B_k^{(\epsilon)} = \sum_{j=0}^{+\infty} \epsilon^j B_{k+jn}, \quad k = 0, 1, \dots, n-1. \tag{3.9}$$

Proof: It follows immediately from Corollary 4 that $\mathbf{A}_\epsilon$ is invertible for $0 < \epsilon < 1$ and $\mathbf{A}_\epsilon^{-1}$ has the decomposition given by (3.4).

According to Corollary 4, (3.4), and analogous to (2.5), we have

$$\begin{bmatrix} \Phi(\delta) \\ \Phi(\delta\omega) \\ \vdots \\ \Phi(\delta\omega^{n-1}) \end{bmatrix} = \begin{bmatrix} \Lambda_0^{-1} \\ \Lambda_1^{-1} \\ \vdots \\ \Lambda_{n-1}^{-1} \end{bmatrix} = \left[\sqrt{n}(F_n D_\delta) \otimes I_m\right] \begin{bmatrix} B_0^{(\epsilon)} \\ B_1^{(\epsilon)} \\ \vdots \\ B_{n-1}^{(\epsilon)} \end{bmatrix}, \tag{3.10}$$

where $D_\delta = \mathrm{diag}(1, \delta, \ldots, \delta^{n-1})$, $\delta = \sqrt[n]{\epsilon}$, and $\omega = \exp\left(\frac{2\pi\mathbf{i}}{n}\right)$, and hence

$$\begin{bmatrix} B_0^{(\epsilon)} \\ B_1^{(\epsilon)} \\ \vdots \\ B_{n-1}^{(\epsilon)} \end{bmatrix} = \left[\frac{1}{\sqrt{n}}(D_\delta^{-1} F_n^*) \otimes I_m\right] \begin{bmatrix} \Phi(\delta) \\ \Phi(\delta\omega) \\ \vdots \\ \Phi(\delta\omega^{n-1}) \end{bmatrix}.$$

Thus, using the fact that

$$\frac{1}{n}\sum_{i=0}^{n-1} \omega^{ij} = \begin{cases} 1, & j \bmod n = 0, \\ 0, & \text{otherwise}, \end{cases}$$

for $k = 0, 1, \ldots, n-1$, we obtain

$$\begin{aligned} B_k^{(\epsilon)} &= \frac{1}{n}\sum_{i=0}^{n-1} \delta^{-k}\omega^{-ki}\Phi(\delta\omega^i) \\ &= \frac{1}{n}\sum_{i=0}^{n-1} \delta^{-k}\omega^{-ki}\sum_{j=0}^{+\infty} B_j(\delta\omega^i)^j \\ &= \sum_{j=0}^{+\infty} \delta^{j-k}\left(\frac{1}{n}\sum_{i=0}^{n-1}\omega^{(j-k)i}\right) B_j \\ &= \sum_{j=0}^{+\infty} \epsilon^j B_{k+jn}. \end{aligned}$$

□

In the following, Theorem 5 will be utilized to estimate the error between $\mathbf{A}^{-1}$ and $\mathbf{A}_\epsilon^{-1}$.

3.2 Error estimation

First, we introduce the following lemma which is useful in our investigation.

Lemma 6 *Let C_j be $m \times m$ matrices for $j = 0, 1, \ldots, n-1$. Then*

i) $\left\| \sum_{j=0}^{n-1} H_0^j \otimes C_j \right\|_\infty = \left\| \sum_{j=0}^{n-1} |C_j| \right\|_\infty$;

ii) $\left\| \sum_{j=0}^{n-1} (H_\epsilon^j - H_0^j) \otimes C_j \right\|_\infty = \epsilon \left\| \sum_{j=1}^{n-1} |C_j| \right\|_\infty$, *with* $\epsilon > 0$.

Proof: i) By (2.1), we have

$$\sum_{j=0}^{n-1} H_0^j \otimes C_j = \begin{bmatrix} C_0 & & & \\ C_1 & C_0 & & \\ \vdots & \ddots & \ddots & \\ C_{n-1} & \dots & C_1 & C_0 \end{bmatrix}.$$

Thus, **i)** holds by taking the maximum matrix norm on both sides of the above equation.

ii) By (2.3),

$$\sum_{j=0}^{n-1} (H_\epsilon^j - H_0^j) \otimes C_j = \begin{bmatrix} 0 & \epsilon C_{n-1} & \dots & \epsilon C_1 \\ & \ddots & \ddots & \vdots \\ & & \ddots & \epsilon C_{n-1} \\ & & & 0 \end{bmatrix}.$$

Therefore, analogous to **i)**, we obtain **ii)** by taking the maximum matrix norm on both sides of the above equation.

□

Theorem 7 *Assume that* $\mathbf{A} = \mathcal{T}_n[\Theta_n(z)]$ *and* $\mathbf{A}_\epsilon = \mathcal{C}_\epsilon[\mathbf{A}]$, *where* $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ *is invertible with* $[\Theta_n(z)]^{-1} = \sum_{j=0}^{+\infty} B_j z^j \in \mathcal{W}$. *Then*

$$\frac{\left\| \mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1} \right\|_\infty}{\|\mathbf{A}^{-1}\|_\infty} \leq \left[1 + (1+\epsilon) \frac{\|M_1\|_\infty}{\|M_0\|_\infty} \right] \epsilon = \mathcal{O}(\epsilon), \tag{3.11}$$

where $M_0 = \sum_{j=0}^{n-1} |B_j|$ *and* $M_1 = \sum_{j=n}^{+\infty} |B_j|$.

Proof: From (3.7) and (3.8), we have

$$\begin{aligned} \mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1} &= \sum_{j=0}^{n-1} H_\epsilon^j \otimes B_j^{(\epsilon)} - \sum_{j=0}^{n-1} H_0^j \otimes B_j \\ &= \sum_{j=0}^{n-1} H_0^j \otimes (B_j^{(\epsilon)} - B_j) + \sum_{j=0}^{n-1} (H_\epsilon^j - H_0^j) \otimes B_j^{(\epsilon)}. \end{aligned}$$

Then, by Lemma 6,

$$\begin{aligned}\left\|\mathbf{A}_\epsilon^{-1}-\mathbf{A}^{-1}\right\|_\infty &\leq \left\|\sum_{j=0}^{n-1} H_0^j\otimes(B_j^{(\epsilon)}-B_j)\right\|_\infty + \left\|\sum_{j=0}^{n-1}(H_\epsilon^j-H_0^j)\otimes B_j^{(\epsilon)}\right\|_\infty \\ &= \left\|\sum_{j=0}^{n-1}|B_j^{(\epsilon)}-B_j|\right\|_\infty + \epsilon\left\|\sum_{j=1}^{n-1}|B_j^{(\epsilon)}|\right\|_\infty . \end{aligned} \tag{3.12}$$

According to (3.9) in Theorem 5, the first term of (3.12) satisfies

$$\begin{aligned}\left\|\sum_{j=0}^{n-1}|B_j^{(\epsilon)}-B_j|\right\|_\infty &\leq \left\|\sum_{j=0}^{n-1}\epsilon\sum_{k=1}^{+\infty}\epsilon^{k-1}|B_{j+kn}|\right\|_\infty \\ &\leq \epsilon\left\|\sum_{j=0}^{n-1}\sum_{k=1}^{+\infty}|B_{j+kn}|\right\|_\infty \\ &= \epsilon\left\|\sum_{j=n}^{+\infty}|B_j|\right\|_\infty = \epsilon\left\|M_1\right\|_\infty . \end{aligned} \tag{3.13}$$

Moreover, for the second term of (3.12),

$$\left\|\sum_{j=1}^{n-1}|B_j^{(\epsilon)}|\right\|_\infty \leq \left\|\sum_{j=0}^{n-1}|B_j^{(\epsilon)}-B_j|\right\|_\infty + \left\|\sum_{j=0}^{n-1}|B_j|\right\|_\infty \leq \epsilon\left\|M_1\right\|_\infty + \left\|M_0\right\|_\infty . \tag{3.14}$$

Substituting inequalities (3.13) and (3.14) into (3.12), we obtain

$$\left\|\mathbf{A}_\epsilon^{-1}-\mathbf{A}^{-1}\right\|_\infty \leq \epsilon\big[\left\|M_0\right\|_\infty + (1+\epsilon)\left\|M_1\right\|_\infty\big]. \tag{3.15}$$

It follows from (3.7) and Lemma 6 again that

$$\left\|\mathbf{A}^{-1}\right\|_\infty = \left\|\sum_{j=0}^{n-1} H_0^j\otimes B_j\right\|_\infty = \left\|\sum_{j=0}^{n-1}|B_j|\right\|_\infty = \left\|M_0\right\|_\infty . \tag{3.16}$$

Thus, by (3.15) and (3.16), we conclude that

$$\frac{\left\|\mathbf{A}_\epsilon^{-1}-\mathbf{A}^{-1}\right\|_\infty}{\left\|\mathbf{A}^{-1}\right\|_\infty} \leq \left[1+(1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right]\epsilon = \mathcal{O}(\epsilon).$$

The proof is completed. □

By the above theorem, we further provide the error estimation between the approximate solution $\mathbf{u}_\epsilon$ and the exact solution $\mathbf{u}$.

Corollary 8 *Under the assumption of Theorem 7, we have*

$$\frac{\|\mathbf{u}_\epsilon - \mathbf{u}\|_\infty}{\|\mathbf{u}\|_\infty} \le \epsilon \left[1 + (1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right] \kappa(\mathbf{A}),$$

where $\sum_{j=0}^{n-1} |B_j| = M_0$, $\sum_{j=n}^{+\infty} |B_j| = M_1$, *and* $\kappa(\mathbf{A}) = \left\|\mathbf{A}^{-1}\right\|_\infty \|\mathbf{A}\|_\infty$ *is the condition number of* $\mathbf{A}$.

Proof: It is easy to check that

$$\begin{aligned} \|\mathbf{u}_\epsilon - \mathbf{u}\|_\infty &= \left\|\mathbf{A}_\epsilon^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{b}\right\|_\infty \\ &\le \left\|\mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1}\right\|_\infty \|\mathbf{b}\|_\infty \\ &= \left\|\mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1}\right\|_\infty \|\mathbf{A}\mathbf{u}\|_\infty \\ &\le \left\|\mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1}\right\|_\infty \|\mathbf{A}\|_\infty \|\mathbf{u}\|_\infty . \end{aligned}$$

By Theorem 7, we have

$$\begin{aligned} \frac{\|\mathbf{u}_\epsilon - \mathbf{u}\|_\infty}{\|\mathbf{u}\|_\infty} &\le \left\|\mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1}\right\|_\infty \|\mathbf{A}\|_\infty \\ &\le \epsilon \left[1 + (1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right] \left\|\mathbf{A}^{-1}\right\|_\infty \|\mathbf{A}\|_\infty \\ &= \epsilon \left[1 + (1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right] \kappa(\mathbf{A}). \end{aligned}$$

□

Generally speaking, the assumption in Theorem 7 is not easy to check. In the following, we will give another condition that may be easier to verify in practice. We first introduce a lemma below.

Lemma 9 (see [5, Theorem 3.4]) *Let* $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ *with* $A_j \le 0$ *for* $j \ge 1$, *and* $\Theta_n(1) = \sum_{j=0}^{n-1} A_j$ *be a nonsingular M-matrix. Then* $\Theta_n(z)$ *is invertible and* $[\Theta_n(z)]^{-1} = \sum_{j=0}^{+\infty} B_j z^j \in \mathcal{W}$ *has nonnegative block coefficients.*

Combined Theorem 7, Corollary 8, and Lemma 9 together, the following corollary is immediately obtained.

Corollary 10 *Let* $\mathbf{A} = \mathcal{T}_n[\Theta_n(z)]$, *where* $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$, *in which* $A_j \le 0$ *for* $j \ge 1$ *and* $\Theta_n(1) = \sum_{j=0}^{n-1} A_j$ *is a nonsingular M-matrix. Then*

i) *the block* ϵ*-circulant matrix* $\mathbf{A}_\epsilon = \mathcal{C}_\epsilon[\mathbf{A}]$ *defined by* (3.3) *is invertible for* $0 < \epsilon < 1$;

ii)

$$\frac{\left\|\mathbf{A}_\epsilon^{-1}-\mathbf{A}^{-1}\right\|_\infty}{\left\|\mathbf{A}^{-1}\right\|_\infty} \leq \left[1+(1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right]\epsilon;$$

iii)

$$\frac{\|\mathbf{u}_\epsilon-\mathbf{u}\|_\infty}{\|\mathbf{u}\|_\infty} \leq \epsilon\left[1+(1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right]\kappa(\mathbf{A}),$$

where $M_0=\sum_{j=0}^{n-1}B_j$ *and* $M_1=\sum_{j=n}^{+\infty}B_j$ *with* $[\Theta_n(z)]^{-1}=\sum_{j=0}^{+\infty}B_jz^j\in\mathcal{W}$.

4 Applications in fractional sub-diffusion equations

In this section, we study the fractional sub-diffusion equation [10, 11, 13, 15, 18, 30, 31]:

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = {}_0\mathcal{D}_t^{1-\gamma}\left[K_\gamma\dfrac{\partial^2u(x,t)}{\partial x^2}\right]+f(x,t), \quad x\in(a,b), \quad t\in(0,T], \\ u(a,t)=\phi_1(t), \quad u(b,t)=\phi_2(t), \quad t\in[0,T], \\ u(x,0)=\psi(x), \quad x\in[a,b], \end{cases} \tag{4.1}$$

where $0<\gamma<1$, $K_\gamma>0$ is the generalized diffusion constant, $f(x,t)$, $\phi_1(t)$, $\phi_2(t)$, and $\psi(x)$ are known sufficiently smooth functions, and ${}_0\mathcal{D}_t^{1-\gamma}u$ is the Riemann-Liouville fractional derivative [22] of order $1-\gamma$ defined by

$${}_0\mathcal{D}_t^{1-\gamma}u(x,t)=\frac{1}{\Gamma(\gamma)}\frac{\partial}{\partial t}\int_0^t\frac{u(x,\tau)}{(t-\tau)^{1-\gamma}}\mathrm{d}\tau.$$

The fractional sub-diffusion equation (4.1) has attracted considerable interest in recent years, and many numerical methods have been proposed to solve it. Among them, finite difference methods are often utilized to discretize this equation; see [10, 11, 13, 15, 18, 30, 31]. More precisely, let $\Delta x=(b-a)/(m+1)$ and $\Delta t=T/n$ with mesh points $(x_i,t_k)=(a+i\Delta x,k\Delta t)$ for $i=0,1,\ldots,m+1$ and $k=0,1,\ldots,n$, respectively, and the corresponding approximate solutions be $u_i^k\approx u(x_i,t_k)$. Then the resulting finite difference schemes of the equation (4.1) discretized by the compact finite difference methods can be written as the following form:

$$\begin{cases} \alpha_i^{(0)}u_{i-1}^1+\beta_i^{(0)}u_i^1+\eta_i^{(0)}u_{i+1}^1=\tilde{b}_i^1, \quad 1\leq i\leq m, \\ \alpha_i^{(0)}u_{i-1}^k+\beta_i^{(0)}u_i^k+\eta_i^{(0)}u_{i+1}^k+\sum_{j=1}^{k-1}\left(\alpha_i^{(k-j)}u_{i-1}^j+\beta_i^{(k-j)}u_i^j+\eta_i^{(k-j)}u_{i+1}^j\right)=\tilde{b}_i^k, \\ \qquad 1\leq i\leq m,\ 2\leq k\leq n, \\ u_0^k=\phi_1(t_k),\ u_{m+1}^k=\phi_2(t_k), \quad 1\leq k\leq n, \\ u_i^0=\psi(x_i), \quad 0\leq i\leq m+1, \end{cases} \tag{4.2}$$

where the coefficients $\alpha_i^{(k)}$, $\beta_i^{(k)}$, and $\eta_i^{(k)}$ are determined by which finite difference methods are used, and the right-hand-side $\tilde{b}_i^k$ contain the nonhomogeneous term and the initial condition.

Let $\mathbf{u}^k = \left[u_1^k, u_2^k, \ldots, u_m^k\right]^\intercal$. Then (4.2) can be expressed as the matrix form

$$\begin{cases} A_0\mathbf{u}^1 = \mathbf{b}^1, \\ A_0\mathbf{u}^k + \sum_{j=1}^{k-1} A_{k-j}\mathbf{u}^j = \mathbf{b}^k, \quad k = 2, 3, \ldots, n, \end{cases} \tag{4.3}$$

where

$$A_j = \begin{bmatrix} \beta_1^{(j)} & \eta_1^{(j)} & & \\ \alpha_2^{(j)} & \beta_2^{(j)} & \ddots & \\ & \ddots & \ddots & \eta_{m-1}^{(j)} \\ & & \alpha_m^{(j)} & \beta_m^{(j)} \end{bmatrix}, \quad \text{for} \quad j = 0, 1, \ldots, n-1, \tag{4.4}$$

and $\mathbf{b}^k \in \mathbb{R}^m$ consist of the vector $\left[\tilde{b}_1^k, \tilde{b}_2^k, \ldots, \tilde{b}_m^k\right]^\intercal$ and boundary conditions.

In general, the linear system (4.3) is solved step by step using the time-marching method with $\mathcal{O}(mn^2)$ operations and $\mathcal{O}(mn)$ storage requirement [10, 13, 15, 31], as all A_j are tri-diagonal matrices. When the step number n is large, the time-marching method is very time consuming; see numerical results in Section 5.

We note that (4.3) can be further rewritten as a BL3TB linear system (1.1) with

$$\mathbf{A} = \begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & \ddots & \ddots & \\ A_{n-1} & \ldots & A_1 & A_0 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \vdots \\ \mathbf{u}^n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \vdots \\ \mathbf{b}^n \end{bmatrix}. \tag{4.5}$$

Therefore, the proposed approximate inversion method described in Section 2 can be employed to solve (4.5). The computational complexity to solve this BL3TB system by the proposed inversion method is of $\mathcal{O}(mn \log n)$ which is much less than $\mathcal{O}(mn^2)$ operations by the traditional time-marching scheme (the same as the block forward substitution method).

Theoretically, we also need to verify if the BL3TB matrix produced by the finite difference method satisfies the condition that guarantees the high accurate approximation by the proposed method. In the following, as an example, we will study the BL3TB matrix resulted from the compact finite difference scheme in [15].

In [15], Gao and Sun proposed a compact finite difference scheme for solving the fractional sub-diffusion equation (4.1). The accuracy by their method is of $\mathcal{O}(\Delta t^{2-\gamma} + \Delta x^4)$. The

coefficient matrices in (4.3) by Gao-Sun's scheme are as follows,

$$A_0 = \begin{bmatrix} 10\Delta x^2 + 24\mu & \Delta x^2 - 12\mu & & \\ \Delta x^2 - 12\mu & 10\Delta x^2 + 24\mu & \ddots & \\ & \ddots & \ddots & \Delta x^2 - 12\mu \\ & & \Delta x^2 - 12\mu & 10\Delta x^2 + 24\mu \end{bmatrix} \in \mathbb{R}^{m\times m} \tag{4.6}$$

and

$$A_j = \Delta x^2(q_j - q_{j-1}) \begin{bmatrix} 10 & 1 & & \\ 1 & 10 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 10 \end{bmatrix} \in \mathbb{R}^{m\times m}, \quad 1 \le j \le n-1, \tag{4.7}$$

where $\mu = K_\gamma \Delta t^\gamma \Gamma(2-\gamma)$ and $q_j = (j+1)^{1-\gamma} - j^{1-\gamma}$; see [15] for more details of the discretization.

We first introduce a lemma about the properties of q_j.

Lemma 11 (see [29]) *Let $0 < \gamma < 1$, $q_j = (j+1)^{1-\gamma} - j^{1-\gamma}$, $j = 0, 1, \ldots$. Then,*

i) $1 = q_0 > q_1 > q_2 > \cdots > q_j \to 0$, *as $j \to +\infty$;*

ii) $q_j < \frac{1-\gamma}{j^\gamma}$.

With the help of Lemma 11, we prove the following theorem.

Theorem 12 *Let $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ with A_j given by (4.6) and (4.7). If $\Delta x^2 \le 12K_\gamma\Gamma(1-\gamma)(T-\Delta t)^\gamma$, then $A_j \le 0$ for $j \ge 1$, and $\Theta_n(1) = \sum_{j=0}^{n-1} A_j$ is a nonsingular M-matrix.*

Proof: According to (4.7) and **i)** in Lemma 11, it is easily obtained that $A_j \le 0$ for $j \ge 1$.

Using (4.6) and (4.7), we have

$$\Theta_n(1) = \sum_{j=0}^{n-1} A_j = (24\mu + 10\Delta x^2 q_{n-1}) I_m - (12\mu - \Delta x^2 q_{n-1}) C,$$

where I_m is the identity matrix and

$$C = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix}.$$

Note that

$$\rho\big((12\mu - \Delta x^2 q_{n-1})C\big) \leq |12\mu - \Delta x^2 q_{n-1}| \cdot \|C\|_\infty = |24\mu - 2\Delta x^2 q_{n-1}| < 24\mu + 10\Delta x^2 q_{n-1}.$$

Then by **iv)** of Definition 1, to render $\Theta_n(1)$ being a nonsingular M-matrix, we only need to verify that $12\mu - \Delta x^2 q_{n-1} \geq 0$.

Recall that $\mu = K_\gamma \Gamma(2-\gamma)\Delta t^\gamma$, it follows from **ii)** of Lemma 11 that

$$\begin{aligned} 12\mu - \Delta x^2 q_{n-1} &\geq 12K_\gamma\Gamma(2-\gamma)\Delta t^\gamma - \Delta x^2 \frac{1-\gamma}{(n-1)^\gamma} \\ &= 12K_\gamma(1-\gamma)\Gamma(1-\gamma)\Delta t^\gamma - \Delta x^2 \frac{1-\gamma}{(T-\Delta t)^\gamma}\Delta t^\gamma \\ &= (1-\gamma)\Delta t^\gamma \left(12K_\gamma\Gamma(1-\gamma) - \frac{\Delta x^2}{(T-\Delta t)^\gamma}\right). \end{aligned}$$

Thus, by the assumption that $\Delta x^2 \leq 12K_\gamma\Gamma(1-\gamma)(T-\Delta t)^\gamma$, we have $12\mu - \Delta x^2 q_{n-1} \geq 0$. The proof is completed. □

By Theorem 12 and Corollary 10, we immediately obtain that if $\Delta x^2 \leq 12K_\gamma\Gamma(1-\gamma)(T-\Delta t)^\gamma$, then all the results of Corollary 10 hold for the BL3TB linear system produced by Gao-Sun's scheme. We note that the condition $\Delta x^2 \leq 12K_\gamma\Gamma(1-\gamma)(T-\Delta t)^\gamma$ is weak in practice since Δx is usually very small. Therefore, the proposed approximate inversion method works very well for the corresponding BL3TB system.

5 Numerical experiments

In this section, we employ the approximate inversion method in Section 2 to solve the linear system (1.1). All numerical experiments are tested by running MATLAB R2010a on a PC with the configuration: Intel(R) Core(TM) CPU i7-2600 3.40 GHz and 16 GB of memory.

Example 1

In this example, we consider $\Theta(z) = \sum_{j=0}^{+\infty} A_j z^j$, where $A_0 = 5I_m - L$, $A_k = -\frac{1}{2^k}L$, $k = 1, 2, \ldots$, and

$$L \equiv \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \eta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \eta_{m-1} & \alpha_m \end{bmatrix}$$

in which $[\alpha_1, \alpha_2, \ldots, \alpha_m]^\intercal$, $[\beta_1, \beta_2, \ldots, \beta_{m-1}]^\intercal$, and $[\eta_1, \eta_2, \ldots, \eta_{m-1}]^\intercal$ are random vectors with entries in $[0, 1]$. Let $\mathbf{A} = \mathcal{T}_n\left[\sum_{j=0}^{+\infty} A_j z^j\right]$. It is not difficult to check that this matrix satisfies

the assumption of Corollary 10. We choose the right hand side $\mathbf{b}$ to be the vector such that the exact solution of the system (1.1) is $\mathbf{u} = [1, 1, \ldots, 1]^\intercal$.

Three methods are applied to solve this example, namely our proposed approximate inversion method (AIM), the divide-and-conquer method (DACM) [5], and the block forward substitution method (BFSM) [16]. In practical computations, we fix $n = 512$ and vary m. Numerical results are reported in Table 1. The symbol "Error_j" in the table refers to the relative maximum error between the exact solution $\mathbf{u}$ and the approximate solution $\mathbf{u}_{\epsilon_j}$ obtained by the approximate inversion method with $\epsilon_j = 0.5 \times 10^{-j}$, i.e.,

$$\text{Error}_\text{j} = \frac{\left\|\mathbf{u}_{\epsilon_j} - \mathbf{u}\right\|_\infty}{\|\mathbf{u}\|_\infty}, \quad j = 4, 6, \ldots, 12,$$

"CPU" means the elapsed time of the computation with unit second. The AIM has the same CPU time for different ϵ when m and n are fixed. Thus, we only list the CPU time for $\epsilon_{12} = 0.5 \times 10^{-12}$ in Table 1. Since the divide-and-conquer method and the block forward substitution method are both direct methods, in Table 1 we only report the CPU times consumed by them.

Table 1: Comparison of CPU times in seconds by the AIM, the DACM, and the BFSM for Example 1 when $n = 512$.

m	AIM						DACM	BFSM
	Error_4	Error_6	Error_8	Error_{10}	Error_{12}	CPU	CPU	CPU
4	1.848e-5	1.848e-7	1.915e-8	2.633e-6	2.126e-4	0.003	0.018	0.021
8	3.028e-5	3.028e-7	1.915e-8	2.633e-6	2.651e-4	0.005	0.031	0.033
16	3.280e-5	3.280e-7	2.814e-8	3.160e-6	2.611e-4	0.009	0.090	0.047
32	4.380e-5	4.380e-7	3.322e-8	3.599e-6	2.651e-4	0.016	0.446	0.085
64	4.160e-5	4.160e-7	3.891e-8	3.659e-6	3.059e-4	0.033	2.051	0.178
128	4.371e-5	4.371e-7	4.352e-8	3.599e-6	3.702e-4	0.071	9.311	0.394
256	3.899e-5	3.899e-7	5.091e-8	3.599e-6	3.529e-4	0.153	53.188	0.995

From Table 1, we see that when m is far less than n, the divide-and-conquer method is faster than the block forward substitution method. However, as m increases, the former becomes slower than the latter. Among them, our proposed approximate inversion method always spends less CPU time than other two methods. When ϵ is larger than $0.5e-8$, it is also observed in Table 1 that our method yields the error of $\mathcal{O}(\epsilon)$ which is agreement with our theoretical analysis. The relative errors increase gradually when ϵ is smaller than $0.5e-8$. Thus, numerically we cannot set ϵ too small, otherwise the computation of D_δ^{-1} will bring in very large rounding error. In the following example, we choose $\epsilon = 0.5e-8$ which is the same as that in [19].

Example 2 (see [10, 13, 15])
In this example, we consider a fractional sub-diffusion equation (4.1) whose data are given as follows: $a = 0$, $b = 1$, $T = 1$, $K_\gamma = 1$, $\gamma = 0.75$, and

$$f(x,t) = e^x \left[(1+\gamma)t^\gamma - \frac{\Gamma(2+\gamma)}{\Gamma(1+2\gamma)} t^{2\gamma} \right].$$

The initial condition is chosen as $\psi(x) = 0$, and the boundary conditions are given by $\phi_1(t) = t^{1+\gamma}$ and $\phi_2(t) = et^{1+\gamma}$. The exact solution is $u(x,t) = e^x t^{1+\gamma}$.

In [15], this fractional sub-diffusion equation is discretized by Gao-Sun's scheme and the time-marching algorithm (the block forward substitution method) is employed to solve the resulting system. In the numerical tests, we also carry out the proposed approximate inversion method for the resulting system to show the effectiveness of the method. Since the divide-and-conquer method is more expensive than the approximate inversion method, as shown in Example 1, it is not necessary to report its numerical results.

In the following tables, "$E_\infty(\Delta t, \Delta x)$" denotes the relative maximum error between the exact solution $\mathbf{U}^n$ and the approximate solution $\mathbf{u}^n$ at the last time step; i.e.,

$$E_\infty(\Delta t, \Delta x) = \frac{\|\mathbf{U}^n - \mathbf{u}^n\|_\infty}{\|\mathbf{U}^n\|_\infty}.$$

The symbol "Order" denotes the convergence order which is equal to $\log_2 \frac{E_\infty(\Delta t,\Delta x)}{E_\infty(\Delta t/2,\Delta x)}$ for the temporal direction or $\log_2 \frac{E_\infty(\Delta t,\Delta x)}{E_\infty(\Delta t,\Delta x/2)}$ for the spatial direction, respectively. Theoretically, it is easy to check that the assumption $\Delta x^2 \le 12K_\gamma\Gamma(1-\gamma)(T-\Delta t)^\gamma$ in Theorem 12 always holds for the numerical tests below. Therefore, the high accurate approximation by our proposed method is guaranteed.

In Table 2, we fix the spatial grid number $m+1 = 200$ and vary the temporal time step n, as in [15]. We observe that the consumed CPU time by the approximate inversion method is much less than that by the time-marching method. On the other hand, the approximate inversion method can preserve almost the same accuracy as that by the time-marching method, which demonstrates the efficiency of the proposed method.

Note that the convergence order by Gao-Sun's scheme in temporal direction is of $\mathcal{O}(\Delta t^{2-\gamma})$ which is much lower than the convergence order of $\mathcal{O}(\Delta x^4)$ in spatial direction. To balance the errors arising from the temporal discretization and the spatial discretization, it often requires a smaller temporal step size in the numerical experiment, i.e., a larger n should be chosen. Therefore, in Table 3, analogous to [15], we fix $n = 200,000$ and vary m in spatial direction. We also see that the errors by the proposed approximate inversion method are almost the same as that by the time-marching method using in [15]. Moreover, the consumed CPU time by the approximate inversion method is much less than that by the the time-marching method. In particular, when $n = 200,000$ and $m+1 = 16$, the approximate inversion method spends less

Table 2: Comparison of relative errors, convergence orders and CPU times in seconds by the AIM and BFSM in temporal direction for Example 2 when $m+1=200$.

n	AIM			BFSM		
	$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU
100	1.141e-4	-	0.037	1.140e-4	-	0.049
200	4.805e-5	1.2477	0.069	4.792e-5	1.2503	0.119
400	2.017e-5	1.2523	0.118	2.015e-5	1.2498	0.426
800	8.323e-6	1.2770	0.233	8.472e-6	1.2500	1.965
1600	3.377e-6	1.3014	0.457	3.562e-6	1.2500	8.630

that 5 seconds of CPU time, while the time-marching method needs more than 10,599 seconds (almost 3 hours).

Table 3: Comparison of relative errors, convergence orders and CPU times in seconds by the AIM and BFSM in spatial direction for Example 2 when $n=200,000$.

$m+1$	AIM			BFSM		
	$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU
4	1.060e-6	-	1.344	1.060e-6	-	2,273.180
8	5.823e-8	4.1862	2.348	5.842e-8	4.1815	5,037.054
16	4.587e-9	3.6661	4.785	4.292e-9	3.7667	10,599.835

6 Concluding remarks

In this paper, we propose an approximate inversion method to solve the BL3TB linear system. The computational complexity of the proposed method is of $\mathcal{O}(mn\log n)$ operations with $\mathcal{O}(mn)$ storage requirement, which is much cheaper than $\mathcal{O}(mn^2)$ operations of the block forward substitution method. Theoretically, a sufficient condition is given to show the high accuracy of the approximation. In applications, the approximate inversion method is applied to solve the fractional sub-diffusion equation. Numerical results exemplify that our method is vastly superior to the popular used time-marching method.

We remark that the proposed method is fundamentally a one-step approximate method. If the approximate solution $\mathbf{u}_\epsilon$ is not accurate enough to the exact solution $\mathbf{u}$, then the iteration methods, such as matrix splitting iteration methods or Krylov subspace methods, could be exploited to improve the accuracy of the approximation. By Theorem 7 we know that $\mathbf{A}_\epsilon^{-1}$

can be very close to $\mathbf{A}^{-1}$ under certain conditions. Therefore $\mathbf{A}_\epsilon$ could be a good choice as the splitting matrix in the splitting iteration method or a preconditioner for the Krylov subspace method. Moreover, in the application, we only discuss Gao-Sun's scheme [15] and give the theoretical analysis for the resulting BL3TB matrix. Regardless the theoretical analysis, besides Gao-Sun's scheme, our approximate inversion method also can be implemented for other schemes, such as those proposed in [10, 11, 12, 13, 18, 28, 30, 31]. In the future, we will extend our approximate inversion method to the multidimensional fractional order partial differential equations [29], which will be more challenging.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] Bai J, Feng X. Fractional-order anisotropic diffusion for image denoising. *IEEE Transactions on Image Processing* 2007; **16**:2492–2502.

[2] Benson D, Wheatcraft S, Meerschaert M. Application of a fractional advection-dispersion equation. *Water Resources Research* 2000; **36**:1403–1412.

[3] Benson D, Wheatcraft S, Meerschaert M. The fractional-order governing equation of Lévy motion. *Water Resources Research* 2000; **36**:1413–1423.

[4] Bini D. Parallel solution of certain Toeplitz linear systems. *SIAM Journal on Computing* 1984; **13**:268–276.

[5] Bini D, Latouche G, Meini B. *Numerical Methods for Structured Markov Chains*. Oxford University Press: New York, 2005.

[6] Bini D, Pan V. Polynomial division and its computational complexity. *Journal of Complexity* 1986; **2**:179–203.

[7] Carreras B, Lynch V, Zaslavsky G. Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence models. *Physics of Plasmas* 2001; **8**:5096–5103.

[8] Chan R, Jin X. *An Introduction to Iterative Toeplitz Solvers*. SIAM: Philadelphia, 2007.

[9] Chan R, Ng M. Conjugate gradient methods for Toeplitz systems. *SIAM Review* 1996; **38**:427–482.

[10] Chen C, Liu F, Anh V, Turner I. Numerical schemes with high spatial accuracy for a variable-order anomalous subdiffusion equation. *SIAM Journal on Scientific Computing* 2010; **32**:1740–1760.

[11] Chen C, Liu F, Turner I, Anh V. A Fourier method for the fractional diffusion equation describing sub-diffusion. *Journal of Computational Physics* 2007; **227**:886–897.

[12] Chen C, Liu F, Turner I, Anh V, Chen Y. Numerical approximation for a variable-order nonlinear reaction-subdiffusion equation. *Numerical Algorithm* 2013; **63**:265–290.

[13] Cui M. Compact finite difference method for the fractional diffusion equation. *Journal of Computational Physics* 2009; **228**:7792–7804.

[14] Commenges D, Monsion M. Fast inversion of triangular Toeplitz matrices. *IEEE Transactions on Automatic Control* 1984; **29**:250–251.

[15] Gao G, Sun Z. A compact finite difference scheme for the fractional sub-diffusion equations. *Journal of Computational Physics* 2011; **230**:586–595.

[16] Golub G, Loan C. *Matrix Computations: 3rd edition*. Johns Hopkins University Press: Baltimore, 1996.

[17] Kailath T, Kung S, Morf M. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications* 1979; **68**:395–407.

[18] Langlands T, Henry B. The accuracy and stability of an implicit solution method for the fractional diffusion equation. *Journal of Computational Physics* 2005; **205**:719–736.

[19] Lin F, Ching W, Ng M. Fast inversion of triangular Toeplitz matrices. *Theoretical Computer Science* 2004; **315**:511–523.

[20] Magin R. *Fractional Calculus in Bioengineering*. Begell House Publishers, 2006.

[21] Pan V. *Structured Matrices and Polynomials: Unified Superfast Algorithms*. Springer: New York, 2001.

[22] Podlubny I. *Fractional Differential Equations*. Academic Press: New York, 1999.

[23] Raberto M, Scalas E, Mainardi F. Waiting-times and returns in high-frequency financial data: an empirical study. *Physica A* 2002; **314**:749–755.

[24] Saad Y. *Iterative Methods for Sparse Linear Systems: 2nd edition*. SIAM: Philadelphia, 2003.

[25] Shlesinger M, West B, Klafter J. Lévy dynamics of enhanced diffusion: application to turbulence. *Physical Review Letters* 1987; **58**:1100–1103.

[26] Sokolov I, Klafter J, Blumen A. Fractional kinetics. *Physics Today* 2002; **55**:48–54.

[27] Zaslavsky G, Stevens D, Weitzner H. Self-similar transport in incomplete chaos. *Physical Review E* 1993; **48**:1683–1694.

[28] Zeng F, Li C, Liu F. High-order explicit-implicit numerical methods for nonlinear anomalous diffusion equations. *The European Physical Journal Special Topics* 2013; **222**: 1885–1900.

[29] Zhang Y, Sun Z. Alternating direction implicit schemes for the two-dimensional fractional sub-diffusion equation. *Journal of Computational Physics* 2011; **230**:8713–8728.

[30] Zhang Y, Sun Z, Wu H. Error estimates of Crank-Nicolson-type difference schemes for the sub-diffusion equation. *SIAM Journal on Numerical Analysis* 2011; **49**:2302–2322.

[31] Zhuang P, Liu F, Anh V, Turner I. New solution and analytical techniques of the implicit numerical method for the anomalous subdiffusion equation. *SIAM Journal on Numerical Analysis* 2008; **46**:1079–1095.