Received: 20 July 2016 | Revised: 31 August 2017 | Accepted: 12 October 2017
DOI: 10.1002/nla.2132

RESEARCH ARTICLE

WILEY

Approximate inversion method for time-fractional subdiffusion equations

Xin Lu[1] | Hong-Kui Pang[2] | Hai-Wei Sun[3] | Seak-Weng Vong[3]

[1]School of Mathematics and Big Data, Foshan University, Foshan, Guangdong 528000, China

[2]School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou, Jiangsu 221116, China

[3]Department of Mathematics, University of Macau, Macao

Correspondence
Hai-Wei Sun, Department of Mathematics, University of Macau, Macao.
Email: HSun@umac.mo

Funding information
University of Macau, Grant/Award Number: MYRG2016-00063-FST and MYRG2015-00064-FST; FDCT of Macao, Grant/Award Number: 054/2015/A2 and 010/2015/A; National Natural Science Foundation of China, Grant/Award Number: 11271238; National Natural Science Foundation of China, Grant/Award Number: 11771189; Natural Science Foundation of Jiangsu Province, Grant/Award Number: BK20171162; Jiangsu Key Laboratory of Education Big Data Science and Engineering; Priority Academic Program Development of Jiangsu Higher Education Institutions

JEL Codes: 65L05; 65N22; 65F10; 65F15

Summary

The finite-difference method applied to the time-fractional subdiffusion equation usually leads to a large-scale linear system with a block lower triangular Toeplitz coefficient matrix. The approximate inversion method is employed to solve this system. A sufficient condition is proved to guarantee the high accuracy of the approximate inversion method for solving the block lower triangular Toeplitz systems, which are easy to verify in practice and have a wide range of applications. The applications of this sufficient condition to several existing finite-difference schemes are investigated. Numerical experiments are presented to verify the validity of theoretical results.

KEYWORDS

approximate inversion method, block lower triangular Toeplitz matrix, fast Fourier transforms, matrix polynomial, time-fractional subdiffusion equations

1 | INTRODUCTION

In this paper, we consider the fast solving of the following time-fractional subdiffusion equation[1]:

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

where $0 < \gamma < 1$, $k(x) > 0$, $f(x,t)$, $\phi_1(t)$, $\phi_2(t)$, and $\psi(x)$ are known sufficiently smooth functions, and ${}_0^C\mathcal{D}_t^\gamma u(x,t)$ is the Caputo fractional derivative[2] of order γ defined by

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

The time-fractional diffusion equation has attracted considerable attention over the past 2 decades, and a variety of stable and accurate numerical methods have been developed. For instance, Langlands et al.[3] investigated the time-fractional diffusion equation with constant coefficient $k(x) = \kappa$ and constructed an implicit finite difference using the $L1$ scheme to approximate the fractional derivative. Chen et al.[4] presented a Fourier method and gave the global accuracy and stability analysis of the difference scheme. Zhuang et al.[5] proposed a novel unconditional stable scheme based on the numerical integration. Cui[6] proposed a fourth-order compact difference scheme to increase the spatial accuracy, whereas the Grünwald–Letnikov difference approximation is used in the time direction.[2] Gao et al.[7] applied the compact difference scheme with the $L1$ approximation to discretize the time-fractional subdiffusion equation and analyzed the stability and convergence of the proposed scheme by the energy method. For the case of variable diffusion coefficient $k(x)$, Zhao et al.[1] designed a compact difference scheme with fourth order in space and $(2-\gamma)$th order in time to solve Equation 1. More works on the numerical solution of time-fractional diffusion equations can be found in previous works[8-14] and references therein. Notice that all the numerical methods mentioned above are time-marching schemes, which solve the fractional diffusion equations step by step. Due to the nonlocal property of the time-fractional derivative,[15] the solutions of all previous time levels should be utilized to compute the solution in the current time step. As a consequence, the total cost of the time-marching method for solving the time-fractional diffusion equation with n time steps is of $\mathcal{O}(mn^2)$, and the storage requirement is of $\mathcal{O}(mn)$, where m refers to the spatial grid number. It is easy to see that solving the time-fractional diffusion equation by the time-marching method costs n times as much as solving an integer-order diffusion equation.

Note that if the numerical solutions in all time steps are stacked in one vector, then the finite-difference discretization of Equation 1 can lead to the following block lower triangular Toeplitz (BLTT) system[16,17]:

$$ \mathbf{A}\mathbf{u} = \mathbf{b} \tag{2} $$

with

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

where $A_j \in \mathbb{C}^{m\times m}$ is a tridiagonal matrix for $j = 0, 1, 2, \ldots, n-1$, $\mathbf{u} \in \mathbb{C}^{mn}$ is the unknown vector, and $\mathbf{b} \in \mathbb{C}^{mn}$ is the right hand side; see Section 2 for more details. With this formation, the commonly used time-marching method is actually the block forward substitution (BFS) method.[18]

We remark that there exist several other methods for dealing with the BLTT system (2). One of them is the block divide-and-conquer method,[19] which is designed by making use of the BLTT structure of the coefficient matrix and the fast Fourier transform. It is an exact inversion method, and the complexity is of $\mathcal{O}(m^2 n\log n + m^3 n)$ and the storage requirement is of $\mathcal{O}(m^2 n)$. Obviously, when m is large, the block divide-and-conquer method is not competitive with the BFS method from both the computational cost and the storage requirement. A fast direct method,[16] which combines the BFS method with the divide-and-conquer strategy, was recently proposed to solve Equation 2. The computational cost of this method is of $\mathcal{O}(mn\log^2 n)$ and the storage requirement is of $\mathcal{O}(mn)$.

Most recently, another type of method, named the approximate inversion method (AIM), has been proposed by Lu et al.[17] to fast solve the BLTT system (2). Based on the BLTT structure, the authors approximated the coefficient matrix of Equation 2 by the block ϵ-circulant matrix, which can be block-diagonalized by the Fourier matrix and diagonal matrix in $\mathcal{O}(mn\log n)$ operations. Moreover, the resulting block diagonal matrix possesses the tridiagonal block structure. As a consequence, the total computational cost by the AIM is of $\mathcal{O}(mn\log n)$ and the storage requirement is of $\mathcal{O}(mn)$. In addition, a sufficient condition has been given in the work by Lu et al.[17] to guarantee the invertibility of the corresponding block ϵ-circulant matrix, and the error estimation of the method has been provided. Nevertheless, the sufficient condition given in the work by Lu et al.[17] is not easy to verify in practice, and the range of its application is limited. In this paper, we revisit the AIM and provide a new sufficient condition, which is easier to check and can be applied to several existing numerical schemes.

It is worth mentioning that McLean[20] has also proposed a fast method with $\mathcal{O}(mn \log n)$ complexity in a completely different way to solve the time-fractional diffusion equation. In his work,[20] the time-fractional diffusion equation is discretized by a piecewise-constant, discontinuous Galerkin method in time combined with a continuous, piecewise-linear finite element method in space. Following the panel clustering technique for boundary element methods,[21] the author designed a fast summation algorithm to deal with the discretization equations using $\mathcal{O}(mn \log n)$ operations in total and $\mathcal{O}(m \log n)$ active memory locations during each time step. Nevertheless, this fast algorithm solves the solutions of n time steps one by one and involves many techniques for implementation. In comparison, our method obtains the solutions of n time steps all at once and can be applied more easily.

The remainder of the paper is organized as follows. Section 2 is devoted to state the finite-difference schemes for solving the fractional diffusion in Equation 1, which can lead to a common coefficient matrix structure—the BLTT structure. In Section 3, we turn to the AIM and establish a sufficient condition to guarantee the efficiency and robustness of the algorithms. The applications of the new sufficient condition to several finite-difference schemes are studied in Section 4. In Section 5, the numerical experiments are presented to illustrate the correction of our results. Finally, we give the concluding remarks in Section 6.

2 | FINITE-DIFFERENCE DISCRETIZATION AND THE BLTT SYSTEM

In this section, we discretize the time-fractional subdiffusion in Equation 1 by finite-difference schemes and reveal the BLTT structure of the resulting linear system.

Let m and n be two positive integers, and $\Delta x = (b-a)/(m+1)$ and $\Delta t = T/n$ be the sizes of spatial grid and time step, respectively. We define the mesh points $(x_j, t_k) = (a + j\Delta x, k\Delta t)$ for $j = 0, 1, \ldots, m+1$ and $k = 0, 1, \ldots, n$, and the corresponding approximate solutions $u_j^k \approx u(x_j, t_k)$ of Equation 1. When a certain finite-difference scheme (for example, see previous works[1,6,7]) is applied to the spatial derivative of the Equation 1, one can obtain the following semi-discretized system of ordinary differential equation:

$$\begin{cases} B\,{}_0^C\mathcal{D}_t^\gamma \hat{\mathbf{u}}(t) + C\hat{\mathbf{u}}(t) = \hat{\mathbf{f}}(t), & t \in (0, T], \\ u(x, 0) = \psi(x), & x \in [a, b], \end{cases} \tag{3}$$

where B is usually a real matrix with its numerical range lying in the open right half-plane, C is generally a real symmetric positive definite matrix, $\hat{\mathbf{u}}(t) = [u(x_1, t), u(x_2, t), \ldots, u(x_m, t)]^\intercal$, and $\hat{\mathbf{f}}(t) \in \mathbb{R}^m$ consists of the vector $[f(x_1, t), f(x_2, t), \ldots, f(x_m, t)]^\intercal$ and boundary conditions.

For the temporal fractional derivative ${}_0^C\mathcal{D}_t^\gamma u(x_i, t_k)$, the popularly used finite-difference formula is the $L1$ approximation,[1,7] as follows:

$${}_0^C\mathcal{D}_t^\gamma u(x_i, t_k) = \sum_{j=0}^{k-1} p_j u_i^{(k-j)} + \mathcal{O}\left(\Delta t^{2-\gamma}\right), \tag{4}$$

where p_j are the coefficients for $j = 0, 1, 2, \ldots$. Let $\mathbf{u}^k = [u_1^k, u_2^k, \ldots, u_m^k]^\intercal$. By substituting Equation 4 into the semi-discretized ordinary differential equation system (3) and omitting the small term, we can obtain the following finite-difference scheme in matrix form:

$$\begin{cases} A_0\mathbf{u}^1 = \mathbf{b}^1, \\ A_0\mathbf{u}^k + \displaystyle\sum_{j=1}^{k-1} A_{k-j}\mathbf{u}^j = \mathbf{b}^k, & k = 2, 3, \ldots, n, \end{cases} \tag{5}$$

where

$$A_j = p_j B + q_j C, \quad \text{for} \quad j = 0, 1, \ldots, n-1, \tag{6}$$

$\mathbf{b}^k = \hat{\mathbf{f}}(t_k) \in \mathbb{R}^m$, and p_k and q_k are some real numbers; see Sections 3 and 4 for the properties of p_k and q_k.

If we stack all $\mathbf{u}^k$ for $k = 1, 2, \ldots, n$ in one vector, then the iteration scheme (5) can be further rewritten as a large linear system, as follows:

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{7}$$

in which $\mathbf{A}$ is a BLTT matrix of the following form:

$$\mathbf{A} = \begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & \ddots & \ddots & \\ A_{n-1} & \cdots & A_1 & A_0 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \vdots \\ \mathbf{u}^n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \vdots \\ \mathbf{b}^n \end{bmatrix}. \tag{8}$$

We remark that applying other finite-difference scheme to the temporal fractional derivative and carrying out in a similar way as above can also lead to the BLTT linear system (7). By solving the system (7), we obtain the numerical solution of Equation 1. In the next section, we focus on fast solving Equation 7 by the AIM.

3 | DISCUSSION ON THE AIM

In this section, we first give a review of the AIM recently developed by Lu et al.[17] and then provide a new sufficient condition for it to guarantee the stability and high accuracy of solving the BLTT systems (7).

3.1 | The AIM

Let the matrix $\mathbf{A}$ in Equation 8 be approximated by

$$\mathbf{A}_\epsilon \equiv \begin{bmatrix} A_0 & \epsilon A_{n-1} & \cdots & \epsilon A_2 & \epsilon A_1 \\ A_1 & A_0 & \epsilon A_{n-1} & \cdots & \epsilon A_2 \\ \vdots & A_1 & A_0 & \ddots & \vdots \\ A_{n-2} & \cdots & \ddots & \ddots & \epsilon A_{n-1} \\ A_{n-1} & A_{n-2} & \cdots & A_1 & A_0 \end{bmatrix} \tag{9}$$

with $\epsilon > 0$, which is a block ϵ-circulant matrix and can be block-diagonalized (see previous works[17,19]) as

$$\mathbf{A}_\epsilon = \left[\left(D_\delta^{-1} F_n^*\right) \otimes I_m\right] \operatorname{diag}\left(\Lambda_0, \Lambda_1, \ldots, \Lambda_{n-1}\right) \left[(F_n D_\delta) \otimes I_m\right]. \tag{10}$$

Here, the symbol " $\otimes$ " is the Kronecker tensor product, $D_\delta = \operatorname{diag}\left(1, \delta, \ldots, \delta^{n-1}\right)$ with $\delta = \sqrt[n]{\epsilon}$ is a diagonal matrix, and F_n is the n-by-n Fourier matrix given by

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

and the diagonal blocks Λ_k, $k = 0, 1, \ldots, n-1$, are $m \times m$ matrices and satisfy the following:

$$\begin{bmatrix} \Lambda_0 \\ \Lambda_1 \\ \vdots \\ \Lambda_{n-1} \end{bmatrix} = \left[\sqrt{n}(F_n D_\delta) \otimes I_m\right] \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{n-1} \end{bmatrix}. \tag{11}$$

From Equation 10, we immediately have

$$\mathbf{A}_\epsilon^{-1} = \left[\left(D_\delta^{-1} F_n^*\right) \otimes I_m\right] \operatorname{diag}\left(\Lambda_0^{-1}, \Lambda_1^{-1}, \ldots, \Lambda_{n-1}^{-1}\right) \left[(F_n D_\delta) \otimes I_m\right], \tag{12}$$

provided that Λ_k for $k = 0, 1, \ldots, n-1$ are invertible. The expression (12) implies that $\mathbf{A}_\epsilon^{-1}$ is also a block ϵ-circulant matrix. Then, replacing the coefficient matrix of Equation 7 by the block ϵ-circulant matrix gives the approximate solution as follows:

$$\mathbf{u} = \mathbf{A}^{-1}\mathbf{b} \approx \mathbf{A}_\epsilon^{-1}\mathbf{b} \equiv \mathbf{u}_\epsilon. \tag{13}$$

By Equations 12 and 13, we see that the AIM can be implemented easily by fast Fourier transforms, and the computational complexity is of $\mathcal{O}(mn \log n)$. Lu et al.[17] also considered the the invertibility of $\mathbf{A}_\epsilon$ and the error estimation for the AIM. However, the conditions provided by Lu et al.[17] are usually difficult to check, and the range of application is limited. In the next subsection, we will provide a new sufficient condition, which is easier to check and has a wider range of applications.

3.2 | A sufficient condition

In order to discuss the sufficient condition for guaranteeing the invertibility of $\mathbf{A}_\epsilon$ and the approximation accuracy of the AIM, we need the following definitions.

Definition 1. (see work by Horn et al.[22])
Let $B = [b_{i,j}]_{m\times m}$ be an $m \times m$ complex matrix. The numerical range of B is defined as follows:

$$W(B) \equiv \left\{ \mathbf{w}^* B \mathbf{w} \,:\, \|\mathbf{w}\|_2 = 1, \mathbf{w} \in \mathbb{C}^m \right\},$$

where $\|\cdot\|$ denotes the 2-norm.

Definition 2. (see work by Bini et al.[19])
The Wiener algebra is denoted by the set $\mathcal{W}$ of matrix power series $\Theta(z) = \sum_{j=0}^{\infty} A_j z^j$ with $z \in \mathbb{C}$ such that

$$\sum_{j=0}^{\infty} |A_j| < \infty,$$

where $|A_j|$ is a matrix whose elements are absolute values of elements of A_j.

Following the work by Lu et al.,[17] we call $\Theta(z) = \sum_{j=0}^{\infty} A_j z^j$ the associate matrix power series of the BLTT matrix $\mathbf{A}$ in Equation 8. The link between $\Theta(z)$ and $\mathbf{A}$ is defined by the following:

$$\mathbf{A} = \begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & \ddots & \ddots & \\ A_{n-1} & \cdots & A_1 & A_0 \end{bmatrix} = \mathcal{T}_n[\Theta(z)] = \mathcal{T}_n\left[\sum_{j=0}^{\infty} A_j z^j\right] = \mathcal{T}_n\left[\sum_{j=0}^{n-1} A_j z^j\right]. \tag{14}$$

It was shown by Lu et al.[17] that the nonsingularity of the matrix polynomial $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ with $[\Theta_n(z)]^{-1} \in \mathcal{W}$ can guarantee the invertibility of $\mathbf{A}_\epsilon$ and the accuracy of the AIM. Notice that a matrix polynomial is a special matrix power series with all $A_j = 0$ for $j \geq n$. The theorem below gives the necessary and sufficient condition for the nonsingularity of $\Theta_n(z)$ with $[\Theta_n(z)]^{-1} \in \mathcal{W}$.

Theorem 1. (see theorem 3.2 in the work by Bini et al.[19])
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 $\det(\Theta(z)) \neq 0$ for $|z| \leq 1$.

Unfortunately, this theorem is not easy to verify in practice. Next, we provide a sufficient condition such that $\Theta_n(z)$ is nonsingular with $[\Theta_n(z)]^{-1} \in \mathcal{W}$. We start with the following lemma.

Lemma 1. (see work by Horn et al.[22])
The eigenvalues of a matrix $A \in \mathbb{C}^{m\times m}$ lie inside its numerical range, that is,

$$\sigma(A) \subseteq W(A),$$

where $\sigma(A)$ denotes the spectrum of A.

Based on Lemma 1, we can prove the following main theorem.

Theorem 2. *Suppose that all the coefficient matrices $A_j \in \mathbb{R}^{m\times m}$ in Equation 14 have the following structure:*

$$A_j = p_j B + q_j C, \quad j = 0, 1, \ldots, n-1, \tag{15}$$

and satisfy the following conditions:

1. *$p_0 > 0 > p_j$ with $j \geq 1$, $\sum_{j=0}^{n-1} p_j > 0$, and $q_0 \geq 0 = q_j$ with $j \geq 1$;*
2. *C is a real symmetric positive definite matrix, and the numerical range of the real matrix B is located in the right half-plane, that is,*

$$W(B) \subseteq \{z \,:\, Re(z) > 0,\ z \in \mathbb{C}\}. \tag{16}$$

Then, the matrix polynomial $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ is invertible with $[\Theta_n(z)]^{-1} \in \mathcal{W}$.

Proof. Let $\mathbf{w} \in \mathbb{C}^m$ with $\|\mathbf{w}\|_2 = 1$. According to the conditions above, we obtain the following:

$$|\mathbf{w}^* A_0 \mathbf{w}| = |p_0 \mathbf{w}^* B \mathbf{w} + q_0 \mathbf{w}^* C \mathbf{w}| \geq p_0 \left|\mathbf{w}^* B \mathbf{w}\right| > \sum_{j=1}^{n-1} \left|p_j\right| \left|\mathbf{w}^* B \mathbf{w}\right|. \tag{17}$$

Then, for any unit vector $\mathbf{w} \in \mathbb{C}^m$ and $|z| \le 1$, we have

$$\begin{aligned}\left|\mathbf{w}^* \left(\sum_{j=1}^{n-1} A_j z^j\right)\mathbf{w}\right| &= \left|\sum_{j=1}^{n-1} p_j z^j \mathbf{w}^* B\mathbf{w}\right| \\ &\le \sum_{j=1}^{n-1} |p_j|\, |\mathbf{w}^* B\mathbf{w}| \\ &< |\mathbf{w}^* A_0 \mathbf{w}|,\end{aligned}$$

where the last inequality follows from Equation 17. Consequently,

$$\mathbf{w}^*\Theta_n(z)\mathbf{w} = \mathbf{w}^* A_0\mathbf{w} + \mathbf{w}^*\left(\sum_{j=1}^{n-1} A_j z^j\right)\mathbf{w} \ne 0.$$

By Lemma 1, we derive that all the eigenvalues of $\Theta_n(z)$ are not equal to zero for $|z| \le 1$, which means that $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ is nonsingular for $|z| \le 1$. Therefore, the results of the theorem are obtained from Theorem 1. □

According to the above theorem and Lu et al.'s[17] theorem 7 and corollary 8, we immediately obtain that the matrix $\mathbf{A}_\epsilon$ is nonsingular and the high accuracy of the AIM is guaranteed if the conditions of Theorem 2 hold. Moreover, precisely, we have the following results.

Theorem 3. *Let $\mathbf{A} = \mathcal{T}_n[\Theta_n(z)]$ with $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$. Assume that A_j satisfy the conditions of Theorem 2. Then, we have the following:*

1. *The block ϵ-circulant matrix $\mathbf{A}_\epsilon$ defined by Equation 9 is nonsingular for $0 < \epsilon < 1$;*
2. $$\frac{\|\mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1}\|_\infty}{\|\mathbf{A}^{-1}\|_\infty} \le \left[1 + (1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right]\epsilon = \mathcal{O}(\epsilon);$$
3. $$\frac{\|\mathbf{u}_\epsilon - \mathbf{u}\|_\infty}{\|\mathbf{u}\|_\infty} \le \epsilon\left[1 + (1+\epsilon)\frac{\|M_1\|_\infty}{\|M_0\|_\infty}\right]\kappa(\mathbf{A}),$$

where $M_0 = \sum_{j=0}^{n-1} B_j$ and $M_1 = \sum_{j=n}^{+\infty} B_j$ with $[\Theta_n(z)]^{-1} = \sum_{j=0}^{+\infty} B_j z^j \in \mathcal{W}$, and $\kappa(\mathbf{A}) = \|\mathbf{A}^{-1}\|_\infty \|\mathbf{A}\|_\infty$ is the condition number of $\mathbf{A}$.

4 | APPLICATIONS FOR SEVERAL FINITE-DIFFERENCE SCHEMES

In this section, the finite-difference schemes proposed by Gao et al.[7] and Zhao et al.[1] are investigated. We show that the sufficient conditions given in Theorem 2 are suitable for the BLTT matrices resulting from these finite-difference schemes, and then, the AIM can be employed to solve the corresponding BLTT systems efficiently.

4.1 | Gao–Sun's scheme

Gao et al.[7] proposed a compact finite-difference scheme with fourth order in space and $(2-\gamma)$th order in time to discretize the time-fractional subdiffusion Equation 1 with constant diffusion coefficient $k(x) = k_\gamma > 0$. The resulting linear systems can be rewritten into a BLTT system (7), and the blocks of the coefficient matrix have the following structure:

$$A_0 = \Delta x^2 a_0^{(1-\gamma)} Q + 12\rho P, \quad A_j = \Delta x^2\left(a_j^{(1-\gamma)} - a_{j-1}^{(1-\gamma)}\right)Q, \quad 1 \le j \le n-1, \tag{18}$$

where

$$P = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{m\times m}, \quad Q = \begin{bmatrix} 10 & 1 & & \\ 1 & 10 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 10 \end{bmatrix} \in \mathbb{R}^{m\times m}, \tag{19}$$

$\rho = k_\gamma \Delta t^\gamma \Gamma(2-\gamma) > 0$, and $a_j^{(\gamma)} = (j+1)^\gamma - j^\gamma$ for $j = 0, 1, \ldots$; see work by Gao et al.[7] for more details of discretization.

The coefficients $a_j^{(\gamma)}$ has the following proposition.

Lemma 2. (see work by Zhuang et al.[5] and Gao et al.[7])
Let $0 < \gamma < 1$, $a_j^{(\gamma)} = (j+1)^\gamma - j^\gamma$, $j = 0, 1, \dots$. *Then,* $1 = a_0^{(\gamma)} > a_1^{(\gamma)} > a_2^{(\gamma)} > \cdots > a_j^{(\gamma)} \to 0$, *as* $j \to +\infty$.

In Gao–Sun's scheme, the matrices Q and P are corresponding to B and C in Equation 15, respectively, which are both real symmetric positive definite matrices. In addition, the coefficients are as follows:

$$\begin{aligned} p_0 &= \Delta x^2 a_0^{(1-\gamma)} > 0 > p_j = \Delta x^2 \left(a_j^{(1-\gamma)} - a_{j-1}^{(1-\gamma)} \right), \quad j \geq 1; \\ \sum_{j=0}^{n-1} p_j &= \Delta x^2 a_{n-1}^{(1-\gamma)} > 0; \\ q_0 &= 12\rho > 0 = q_j, \quad j \geq 1. \end{aligned}$$

Thus, the BLTT matrix resulting from Gao–Sun's scheme satisfies the conditions of Theorem 2.

4.2 | Zhao–Xu's scheme

Zhao et al.[1] proposed a compact finite-difference scheme with a convergence order of $\mathcal{O}(\Delta t^{2-\gamma} + \Delta x^4)$ for solving the time-fractional subdiffusion Equation 1 with the variable diffusion coefficient $k(x)$. The blocks of the BLTT matrix resulting from Zhao–Xu's scheme have the following structures:

$$A_0 = a_0^{(1-\gamma)} \Delta x^2 (2Q + \Delta x R) + 24\zeta S, \tag{20}$$

and

$$A_j = \Delta x^2 \left(a_j^{(1-\gamma)} - a_{j-1}^{(1-\gamma)} \right) (2Q + \Delta x R), \quad 1 \leq j \leq n-1, \tag{21}$$

where $\zeta = \Delta t^\gamma \Gamma(2-\gamma) > 0$, the matrix Q is the same as that in Equation 19,

$$R = \begin{bmatrix} 0 & -\left(\frac{k'}{k}\right)_2 & & \\ \left(\frac{k'}{k}\right)_1 & 0 & \ddots & \\ & \ddots & \ddots & -\left(\frac{k'}{k}\right)_m \\ & & \left(\frac{k'}{k}\right)_{m-1} & 0 \end{bmatrix} \in \mathbb{R}^{m \times m},$$

and

$$S = \begin{bmatrix} \varphi_{1+\frac{1}{2}} + \varphi_{1-\frac{1}{2}} & -\varphi_{1+\frac{1}{2}} & & \\ -\varphi_{1+\frac{1}{2}} & \varphi_{2+\frac{1}{2}} + \varphi_{2-\frac{1}{2}} & \ddots & \\ & \ddots & \ddots & -\varphi_{(m-1)+\frac{1}{2}} \\ & & -\varphi_{(m-1)+\frac{1}{2}} & \varphi_{m+\frac{1}{2}} + \varphi_{m-\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{m \times m},$$

in which $\left(\frac{k'}{k}\right)_i = \frac{k'(x_i)}{k(x_i)}$ with $\left| \left(\frac{k'(x)}{k(x)}\right)' \right| \leq c_2$, $\varphi_i = \varphi(x_i)$ with $c_1 \leq \varphi(x) = k(x) - \frac{\Delta x^2}{12}\left(\frac{(k'(x))^2}{k(x)} - \frac{1}{2}k''(x) \right) \leq c_2$, and c_1, c_2 are two positive constants; see the work by Zhao et al.[1] for more details of discretization.

In this scheme, the matrices $2Q + \Delta x R$ and S are corresponding to B and C in Equation 15, respectively. Because the coefficients $a_j^{(1-\gamma)}$ have the properties as that in Lemma 2, we have

$$\begin{aligned} p_0 &= \Delta x^2 a_0^{(1-\gamma)} > 0 > p_j = \Delta x^2 \left(a_j^{(1-\gamma)} - a_{j-1}^{(1-\gamma)} \right), \quad j \geq 1; \\ \sum_{j=0}^{n-1} p_j &= \Delta x^2 a_{n-1}^{(1-\gamma)} > 0; \\ q_0 &= 24\zeta > 0 = q_j, \quad j \geq 1. \end{aligned}$$

Next, we verify that S is a positive definite matrix, and the numerical range of $2Q + \Delta x R$ lies in the right half-plane. Noticing that $0 < c_1 \le \varphi(x) \le c_2$, for any vector $\mathbf{w} = [w_1, w_2, \dots, w_m]^\intercal \in \mathbb{C}^m$ with w_j being not a constant, we have

$$\begin{aligned}
\mathbf{w}^* S \mathbf{w} &= \sum_{j=1}^{m} \left(\varphi_{j+\frac{1}{2}} + \varphi_{j-\frac{1}{2}} \right) |w_j|^2 - 2 \sum_{j=1}^{m-1} \varphi_{j+\frac{1}{2}} \mathrm{Re}\left(\bar{w}_j w_{j+1}\right) \\
&= \sum_{j=1}^{m} \varphi_{j+\frac{1}{2}} |w_j|^2 - 2 \sum_{j=1}^{m-1} \varphi_{j+\frac{1}{2}} \mathrm{Re}\left(\bar{w}_j w_{j+1}\right) + \sum_{j=0}^{m-1} \varphi_{j+\frac{1}{2}} |w_{j+1}|^2 \\
&\ge \sum_{j=1}^{m-1} \varphi_{j+\frac{1}{2}} |w_j|^2 - 2 \sum_{j=1}^{m-1} \varphi_{j+\frac{1}{2}} |w_j||w_{j+1}| + \sum_{j=1}^{m-1} \varphi_{j+\frac{1}{2}} |w_{j+1}|^2 \\
&= \sum_{j=1}^{m-1} \varphi_{j+\frac{1}{2}} \left(|w_j| - |w_{j+1}| \right)^2 \\
&\ge c_1 \sum_{j=1}^{m-1} \left(|w_j| - |w_{j+1}| \right)^2 \\
&> 0.
\end{aligned}$$

If $\mathbf{w} = w[1, 1, \dots, 1]^\intercal \in \mathbb{C}^m$ with $w \neq 0$, then we have

$$\mathbf{w}^* S \mathbf{w} = w \left(\varphi_{1-\frac{1}{2}} + \varphi_{m+\frac{1}{2}} \right) > 0.$$

Therefore, the matrix S is a positive definite matrix.

On the other hand, for all unit vectors $\mathbf{w} = [w_1, w_2, \dots, w_m]^\intercal \in \mathbb{C}^m$, noticing that

$$\left| \left(\frac{k'(x)}{k(x)} \right)' \right| \le c_2,$$

we obtain that

$$\begin{aligned}
\mathrm{Re}\left(\mathbf{w}^*(2Q + \Delta x R)\mathbf{w}\right) &= 2\mathrm{Re}(\mathbf{w}^* Q \mathbf{w}) - \Delta x \mathrm{Re} \left(\sum_{j=2}^{m} \left(\frac{k'}{k} \right)_j w_j \bar{w}_{j-1} - \sum_{j=1}^{m-1} \left(\frac{k'}{k} \right)_j w_j \bar{w}_{j+1} \right) \\
&= 2\mathrm{Re}(\mathbf{w}^* Q \mathbf{w}) - \Delta x \left(\sum_{j=1}^{m-1} \left(\left(\frac{k'}{k} \right)_{j+1} - \left(\frac{k'}{k} \right)_j \right) \mathrm{Re}(\bar{w}_j w_{j+1}) \right) \\
&= 2\mathrm{Re}(\mathbf{w}^* Q \mathbf{w}) - \Delta x^2 \left(\sum_{j=1}^{m-1} \left(\left(\frac{k'}{k} \right)' (\theta_j) \right) \mathrm{Re}(\bar{w}_j w_{j+1}) \right) \\
&\ge 2\mathrm{Re}(\mathbf{w}^* Q \mathbf{w}) - \Delta x^2 \left(\sum_{j=1}^{m-1} \left| \left(\frac{k'}{k} \right)' (\theta_j) \right| \left| \mathrm{Re}(\bar{w}_j w_{j+1}) \right| \right) \\
&\ge 16 \, \|\mathbf{w}\|_2^2 - \Delta x^2 c_2 \, \|\mathbf{w}\|_2^2 \\
&= 16 - \Delta x^2 c_2 > 0,
\end{aligned}$$

provided that $\Delta x < \frac{4}{\sqrt{c_2}}$, where $\theta_j \in [x_j, x_{j+1}]$ for $j = 1, 2, \dots, m-1$. As a result, when Δx is sufficiently small, we have

$$W(2Q + \Delta x R) \subseteq \{ z : \mathrm{Re}(z) > 0, \ z \in \mathbb{C} \}.$$

Hence, the BLTT matrix arising from Zhao–Xu's scheme also satisfies the conditions of Theorem 2.

In view of the properties of matrices A_j in both Gao–Sun's scheme and Zhao–Xu's scheme, we immediately obtain that all the results of Theorem 3 hold for the BLTT matrices produced by the two finite-difference schemes above. Therefore, the fast approximate inversion method given in Section 1 can work very well for the corresponding BLTT systems. Some remarks are given as follows.

Remark 1. **(a)** For Gao–Sun's scheme, it requires the condition $\Delta x^2 \le 12 k_\gamma \Gamma(1-\gamma)(T - \Delta t)^\gamma$ in the work by Lu et al.[17] for the invertibility of $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ in the Wiener algebra $\mathcal{W}$. However, here, we do not need any condition for Δx to guarantee the invertibility of $\Theta_n(z)$ in $\mathcal{W}$. **(b)** For Zhao–Xu's scheme, the spatial step Δx should satisfy the condition ($\Delta x < \frac{4}{\sqrt{c_2}}$) in order to guarantee the invertibility of $\Theta_n(z) = \sum_{j=0}^{n-1} A_j z^j$ in $\mathcal{W}$. We note that this condition is weak and can be easily satisfied in practice.

5 | NUMERICAL EXPERIMENTS

In this section, we employ the AIM presented in Section 3 to solve the linear system (7). In all numerical tests, we choose $\epsilon = 0.5 \times 10^{-8}$ for AIM, which is the same as that in the work by Lu et al.[17] All numerical experiments are tested by running MATLAB R2010a on a PC with the following configuration: Intel(R) Core(TM) CPU i7-2600 3.40 GHz and 16 GB of memory.

In the following tables, "$E_\infty(\Delta t, \Delta x)$" denotes the maximum norm error of the numerical solution at the last time step, that is,

$$E_\infty(\Delta t, \Delta x) = \max_{0 \le j \le m+1} \left| u(x_j, t_n) - u_j^n \right|,$$

where $u(x_j, t_n)$ represents the exact solution at the grid point (x_j, t_n), and u_j^n is the numerical solution with the mesh step sizes Δx and Δt. The symbol "Order" stands for 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. The column "CPU" displays the total CPU time with unit second for solving the corresponding linear systems.

Example 1. (see works by Chen et al.,[4] Cui,[6] and Gao et al.[7])
Consider the following fractional subdiffusion equation with constant coefficient:

$$\begin{cases} {}_0^C\mathcal{D}_t^\gamma u(x,t) = \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t), & x \in (0,1), \quad t \in (0,1], \\ u(0,t) = t^{1+\gamma}, \quad u(1,t) = e t^{1+\gamma}, & t \in (0,1], \\ u(x,0) = 0, \quad x \in [0,1], \end{cases}$$

where $f(x,t) = e^x[\Gamma(2+\gamma)t - t^{1+\gamma}]$. The exact solution is $u(x,t) = e^x t^{1+\gamma}$.

We discretize this equation by Gao–Sun's scheme and obtain the BLTT system (7) with blocks of the coefficient matrix given by Equations 18 and 19. For highlighting the accuracy and efficiency of our method, we also apply the block forward substitution method (BFSM) to solve the resulting system. Following the work of Gao et al.,[7] we first fix the spatial grid number $m + 1 = 200$ sufficiently large and let the time grid number n vary from 400 to 1600, and then fix the time grid number $n = 200,000$ large enough and vary the spatial grid number $m + 1$ from 4 to 16. The maximum norm error of the solution at time $t = 1$, the convergence order, and the CPU time consumed by the methods AIM and BFSM for solving the resulting BLTT linear systems are reported in Tables 1 and 2.

TABLE 1 Comparison of errors and CPU times in seconds by the AIM and BFSM in temporal direction for Example 1 when $m + 1 = 200$

γ	n	AIM			BFSM		
		$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU
	400	8.738E−5	–	0.084	8.739E−5	–	0.352
$\frac{4}{5}$	800	3.803E−5	1.200	0.183	3.804E−5	1.200	1.666
	1600	1.635E−5	1.218	0.377	1.656E−5	1.200	7.410

Note. AIM = approximate inversion method.

TABLE 2 Comparison of errors and CPU times in seconds by the AIM and BFSM in spatial direction for Example 1 when $n = 200,000$

γ	$m+1$	AIM			BFSM		
		$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU
	4	2.988E−6	–	0.920	2.992E−6	–	2,238.459
$\frac{1}{2}$	8	1.831E−7	4.029	1.935	1.869E−7	4.001	5,035.569
	16	1.136E−8	4.011	3.977	1.139E−8	4.036	10,675.189

Note. AIM = approximate inversion method.

It is clearly seen that the consumed CPU time by the AIM is much less than that by the BFSM. Moreover, the AIM can preserve the same convergence order as that by the BFSM either in temporal direction or in spatial direction, which demonstrates the efficiency of the proposed method. Especially, for the case $n = 200{,}000$ and $m + 1 = 16$, the AIM spends less than 4 s of CPU time, whereas the BFSM needs almost 3 hrs. These brilliant performances illustrate the great competitiveness of our proposed method.

Example 2. (see the work of Zhao et al.[1])
In this example, we consider the following fractional subdiffusion equation with variable coefficient:

$$\begin{cases} {}_0^C\mathcal{D}_t^{\gamma} u(x,t) = \frac{\partial}{\partial x}\left((x^2+1)\frac{\partial u(x,t)}{\partial x}\right) + f(x,t), & x \in (0,1), \quad t \in (0,1], \\ u(0,t) = 0, \quad u(1,t) = 0, & t \in (0,1], \\ u(x,0) = 0, \quad x \in [0,1], \end{cases}$$

where $f(x,t) = \left[\frac{\Gamma(4)}{\Gamma(4-\gamma)} t^{3-\gamma} + \pi^2 t^3 (x^2+1)\right] \sin(\pi x) - 2\pi x t^3 \cos(\pi x)$. The exact solution is $u(x,t) = t^3 \sin(\pi x)$.

Discretization of this equation by Zhao–Xu's scheme leads to the BLTT system (7), the blocks of whose coefficient matrix have the structures as in Equations 20 and 21. As done in Example 1, we first fix the spatial grid number $m+1$ sufficiently large and vary the time grid number n, and then fix the time grid number n large enough and change the spatial grid number $m+1$. Tables 3 and 4 present the numerical results for a variety of γ. Note that the method proposed by Ke et al.,[16] which combines the BFS method with the divide-and-conquer strategy, can also be used to solve the BLTT linear system.

TABLE 3 Comparison of errors and CPU times in seconds by the AIM, BFSM, and DC-BFS method in temporal direction for Example 2 when $m + 1 = 200$

γ	n	AIM			BFSM			DC-BFS		
		$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU
	400	3.108E−6	–	0.151	3.106E−6	–	0.505	3.106E−6	–	0.260
$\frac{1}{3}$	800	9.983E−7	1.638	0.283	9.944E−7	1.643	2.120	9.944E−7	1.643	0.522
	1600	3.178E−7	1.651	0.576	3.173E−7	1.648	9.134	3.173E−7	1.648	1.168
	400	1.214E−5	–	0.132	1.213E−5	–	0.495	1.213E−5	–	0.262
$\frac{1}{2}$	800	4.307E−6	1.495	0.281	4.316E−6	1.491	2.111	4.316E−6	1.491	0.558
	1600	1.504E−6	1.518	0.571	1.533E−6	1.493	9.123	1.533E−6	1.493	1.135
	400	4.421E−5	–	0.131	4.422E−5	–	0.491	4.422E−5	–	0.260
$\frac{2}{3}$	800	1.758E−5	1.330	0.281	1.759E−5	1.330	2.132	1.759E−5	1.330	0.542
	1600	6.984E−6	1.332	0.573	6.991E−6	1.331	9.066	6.991E−6	1.331	1.145

Note. AIM = approximate inversion method.

TABLE 4 Comparison of errors and CPU times in seconds by the AIM, BFSM, and DC-BFS method in spatial direction for Example 2 when $n = 200{,}000$

γ	$m+1$	AIM			BFSM			DC-BFS		
		$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU	$E_\infty(\Delta t, \Delta x)$	Order	CPU
	4	1.026E−3	–	1.490	1.026E−3	–	2,359.820	1.026E−3	–	117.317
$\frac{1}{3}$	8	6.697E−5	3.937	3.030	6.697E−5	3.937	5,240.231	6.697E−5	3.937	250.113
	16	4.226E−6	3.986	5.995	4.226E−6	3.986	11,030.314	4.226E−6	3.986	471.304
	4	1.013E−3	–	1.475	1.013E−3	–	2,344.101	1.013E−3	–	119.764
$\frac{1}{2}$	8	6.590E−5	3.942	3.002	6.590E−5	3.942	5,248.869	6.590E−5	3.942	254.708
	16	4.165E−6	3.984	5.822	4.165E−6	3.984	10,830.941	4.165E−6	3.984	473.802
	4	9.976E−4	–	1.496	9.976E−4	–	2,351.658	9.976E−4	–	120.261
$\frac{2}{3}$	8	6.468E−5	3.947	3.035	6.468E−5	3.947	5,250.092	6.468E−5	3.947	255.133
	16	4.104E−6	3.978	5.860	4.103E−6	3.979	10,761.447	4.103E−6	3.979	476.475

Note. AIM = approximate inversion method.

The cost of this method is typically $\mathcal{O}(mn\log^2 n)$ flops. For the purpose of comparison, we also perform this method to the resulting linear system. The numerical results are also listed in Tables 3 and 4, where the method is denoted by "DC-BFS".

From Tables 3 and 4, we once again observe that the AIM solves the fractional subdiffusion equation as precisely as the two methods BFSM and DC-BFS do. Moreover, the AIM still shows a great advantage in the elapsed CPU times, even compared with the fast method DC-BFS. This advantage is rather significant especially when the time grid number n is large.

6 | CONCLUDING REMARKS

The finite-difference approximation to the fractional subdiffusion in Equation 1 usually leads to the linear system with the BLTT coefficient matrix. A fast AIM was presented by Lu et al.[17] to solve the resulting linear system. In addition, a sufficient condition for guaranteeing the invertibility of the corresponding block ϵ-circulant matrix was discussed, and the error estimation of the method was considered. However, the condition used by Lu et al.[17] is generally not easy to check. Based on the necessary and sufficient condition of the invertibility of the associate matrix power series of the BLTT coefficient matrix, in the present work, we provide a new sufficient condition for the BLTT coefficient matrix to ensure the invertibility of the block ϵ-circulant matrix and the high accuracy of the method. Compared with the condition in the work by Lu et al.,[17] the new sufficient condition is easier to verify in practice and has a wider range of applications. Two existing finite-difference schemes, which are used to deal with the constant diffusion coefficient case and variable diffusion coefficient case, are investigated carefully. Numerical experiments are displayed to exemplify our results.

We note that the AIM is an approximation method and seems to be a lossy method. Nevertheless, the accuracy of the method is shown to be $\mathcal{O}(\epsilon)$ with $0 < \epsilon < 1$; see Theorem 3. Numerically, the value of ϵ can be chosen as half of the roundoff error, for example, $\epsilon = 0.5 \times 10^{-8}$, which is enough for many applications. In this view point, the AIM can be regarded as a rough lossless method. Furthermore, higher accuracy (almost to the machine accuracy) can be achieved by performing the AIM with several different values of ϵ and then taking a linear convex combination of these solutions; see the work of Bini et al.[23] for more details. We point out that the AIM can also be applied to the multidimensional version of Equation 1. In fact, a recent paper[24] has extended the AIM, combined with the multigrid method, to efficiently solve the two-dimensional fractional diffusion equation. Finally, we remark that the AIM here cannot be extended directly to the time-dependent diffusion coefficient case, that is, $k = k(x, t)$, because the resulting matrix $\mathbf{A}$ in this situation will not be in the format of Equation 8. For the time-dependent diffusion coefficient case, one could employ the method proposed by Ke et al.,[16] whose complexity is of $\mathcal{O}(mn\log^2 n)$ operations and the storage requirement is of $\mathcal{O}(mn)$.

ACKNOWLEDGEMENTS

We are very grateful to the anonymous referees for their invaluable comments and very detailed suggestions that have greatly improved the presentation of this paper. The research was supported by research grants MYRG2016-00063-FST from the University of Macau and 054/2015/A2 from FDCT of Macao and the National Natural Science Foundation of China under grant 11271238. Hong-Kui Pang was supported by the National Natural Science Foundation of China under grant 11771189, the Natural Science Foundation of Jiangsu Province under grant BK20171162, the Jiangsu Key Laboratory of Education Big Data Science and Engineering, and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. Seak-Weng Vong was supported by research grants MYRG2015-00064-FST from the University of Macau and 010/2015/A from FDCT of Macao.

ORCID

Hai-Wei Sun http://orcid.org/0000-0002-5507-6083

REFERENCES

1. Zhao X, Xu Q. Efficient numerical schemes for fractional sub-diffusion equation with the spatially variable coefficient. Appl Math Model. 2014;38:3848–3859.
2. Podlubny I. Fractional Differential Equations. New York: Academic Press; 1999.
3. Langlands T, Henry B. The accuracy and stability of an implicit solution method for the fractional diffusion equation. J Comput Phys. 2005;205:719–736.

4. Chen C, Liu F, Turner I, Anh V. A fourier method for the fractional diffusion equation describing sub-diffusion. J Comput Phys. 2007;227:886–897.
5. Zhuang P, Liu F, Anh V, Turner I. New solution and analytical techniques of the implicit numerical method for the anomalous subdiffusion equation. SIAM J Numer Anal. 2008;46:1079–1095.
6. Cui M. Compact finite difference method for the fractional diffusion equation. J Comput Phys. 2009;228:7792–7804.
7. Gao G, Sun Z. A compact finite difference scheme for the fractional sub-diffusion equations. J Comput Phys. 2011;230:586–595.
8. Jiang Y, Ma J. High-order finite element methods for time-fractional partial differential equations. J Comput Appl Math. 2011;235:3285–3290.
9. Lin Y, Xu C. Finite difference/spectral approximations for the time-fractional diffusion equation. J Comput Phys. 2007;225:1533–1552.
10. Murio D. Implicit finite difference approximation for time fractional diffusion equations. Comput Math Appl. 2008;56:1138–1145.
11. Mustapha K. An implicit finite-difference time-stepping method for a sub-diffusion equation, with spatial discretization by finite elements. IMA J Numer Anal. 2011;31:719–739.
12. Yuste S, Acedo L. An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations. SIAM J Numer Anal. 2005;42:1862–1874.
13. Zhang Y, Sun Z, Wu H. Error estimates of Crank-Nicolson-type difference schemes for the sub-diffusion equation. SIAM J Numer Anal. 2011;49:2302–2322.
14. Zeng F, Li C, Liu F, Turner I. The use of finite difference/element approaches for solving the time-fractional subdiffusion equation. SIAM J Sci Comput. 2013;35:2976–3000.
15. Diethelm K. The analysis of Fractional Differential Equations. Berlin: Springer; 2010.
16. Ke R, Ng MK, Sun H. A fast direct method for block triangular Toeplitz-like with tri-diagonal block systems from time-fractional partial differential equations. J Comput Phys. 2015;303:203–211.
17. Lu X, Pang H, Sun H. Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations. Numer Linear Algebra Appl. 2015;22:866–882.
18. Golub G, Loan C. Matrix Computations. 4th ed. Baltimore: Johns Hopkins University Press; 2013.
19. Bini D, Latouche G, Meini B. Numerical Methods for Structured Markov Chains. New York: Oxford University Press Inc; 2005.
20. McLean W. Fast summation by interval clustering for an evolution equation with memory. SIAM J Sci Comput. 2012;34:A3039–A3056.
21. Hackbushch W, Nowak ZP. On the fast matrix multiplication in the boundary element method by panel clustering. Numer Math. 1989;54:463–491.
22. Horn R, Johnson C. Topics in Matrix Analysis. Cambridge: Cambridge University Press; 1991.
23. Bini D, Dendievel S, Latouche G, Meini B. Computing the exponential of large block-triangular block-Toeplitz matrices encountered in fluid queues. Linear Algebra Appl. 2016;502:387–419.
24. Lin X, Lu X, Ng MK, Sun H. A fast accurate approximation method with multigrid solver for two-dimensional fractional sub-diffusion equation. J Comput Phys. 2016;323:204–218.

How to cite this article: Lu X, Pang H-K, Sun H-W, Vong S-W. Approximate inversion method for time-fractional sub-diffusion equations. *Numer Linear Algebra Appl*. 2017;e2132. https://doi.org/10.1002/nla.2132