Journal of Computational and Applied Mathematics 277 (2015) 73–86

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Strang-type preconditioners for solving fractional diffusion equations by boundary value methods

Xian-Ming Gu [a,b], Ting-Zhu Huang [a,*], Xi-Le Zhao [a], Hou-Biao Li [a], Liang Li [a]

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, PR China

[b] Institute of Mathematics and Computing Science, University of Groningen, Nijenborgh 9, P.O. Box 407, 9700 AK Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 7 August 2013
Received in revised form 17 March 2014

MSC:
65F12
65L05
65N22

Keywords:
Fractional diffusion equations
Shifted Grünwald formula
BVM
GMRES method
Block-circulant preconditioner
Fast Fourier transform

ABSTRACT

The finite difference scheme with the shifted Grünwald formula is employed to semi-discrete the fractional diffusion equations. This spatial discretization can reduce to the large system of ordinary differential equations (ODEs) with initial values. Recently, the boundary value method (BVM) was developed as a popular algorithm for solving the large systems of ODEs. This method requires the solutions of one or more nonsymmetric and large-scale linear systems. In this paper, the GMRES method with the block circulant preconditioner is proposed to solve relevant linear systems. Some conclusions about the convergence analysis and spectrum of the preconditioned matrices are also drawn if the diffusion coefficients are constant. Finally, extensive numerical experiments are reported to show the performance of our method for solving the fractional diffusion equations.

1. Introduction

During recent years, the concept of fractional derivatives, and their applications to modeling anomalous diffusion phenomena are widely recognized by engineers and mathematicians. Fractional diffusion equations (FDEs) are useful for applications in which a cloud of particles spreads faster than predicted by the classical equation. FDEs arise in research topics including modeling chaotic dynamics of classical conservative systems [1], turbulent flow [2,3], groundwater contaminant transport [4,5], and applications in biology [6], finance [7,8], image processing [9,10], hydrology [11] and other physics issues [12]. For example, anomalous diffusion is a possible mechanism underlying plasma transport in magnetically confined plasmas, and the fractional order space derivative operators can be used to model such transport mechanism. As the closed-form analytical solutions of the FDEs are unavailable in most situations, so numerical solutions for FDEs become main ways and then have been developed intensively, such as (compact) finite difference method [13–19], finite element method [20–23], discontinuous Galerkin method [24,25] and other numerical methods [26–31].

However, due to the nonlocality of fractional differential operator, it had proved that a naive discretization of the FDE, even though implicit, leads to unconditionally unstable [16,17]. Moreover, most numerical methods for FDEs are liable to generate full coefficient matrices, which typically require computational cost of $\mathcal{O}(N^3)$ and storage of $\mathcal{O}(N^2)$, where N is the number of grid points [32]. It is rather different from the classical second-order diffusion equations which usually yield

* Corresponding author. Tel.: +86 28 61831608.
E-mail addresses: guxianming@live.cn (X.-M. Gu), tingzhuhuang@126.com, tzhuang@uestc.edu.cn (T.-Z. Huang).

http://dx.doi.org/10.1016/j.cam.2014.08.011
0377-0427/

sparse coefficient matrices with $\mathcal{O}(N)$ nonzero entries and can be solved very efficiently via fast iterative methods with $\mathcal{O}(N)$ complexity.

In order to handle the difficulty of the stability, Meerschaet and Tadjeran [16,17] proposed a shifted Grünwald discretization to approximate FDEs. Their method has been proven to be unconditionally stable. Later, Wang, et al. [32] discovered that the full coefficient matrix via Meerschaet–Tadjeran's method possesses a Toeplitz-like structure. More precisely, such a full matrix can be written as the sum of diagonal-multiply-Toeplitz matrices. Thus the storage requirement is significantly reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. It is well known that the matrix–vector multiplication for the Toeplitz matrix can be calculated by using the fast Fourier transform (FFT) with $\mathcal{O}(N \log N)$ operations [33,34]. With this advantage, Wang and Wang [27] employed the conjugate gradient normal residual (CGNR) method to solve the discretized system of the FDE by Meerschaet–Tadjeran's method. Due to the Toeplitz-like structure, the cost per iteration by using the CGNR method is of $\mathcal{O}(N \log N)$. The convergence of the CGNR method is fast with smaller diffusion coefficients [27] (in that case the discretized system is well-conditioned). Nevertheless, if the diffusion coefficient functions are not small, the resulting system will become ill-conditioned and hence the CGNR method converges very slowly. To overtake this shortcoming, Pang and Sun [35] proposed a multigrid method to solve the discretized system of the FDE by using the Meerschaet–Tadjeran's method. With the damped-Jacobi method as the smoother, the multigrid algorithm can preserve the computational cost per iteration as $O(N \log N)$ operations. Numerical results showed that their multigrid method converges very fast, even for the ill-conditioned systems. However, from the theoretical point of view, the linear convergence of their multigrid method, despite a very simple case (both diffusion coefficients are equal and constant), has not been proven, see [35] for details. Recently, Lei and Sun [36] proposed a robust CGNR method with the circulant preconditioner to solve FDEs by Meerschaet–Tadjeran's method under the conditions that the diffusion coefficients are constant and the ratio is bounded away from zero. The convergence analysis of their method can be archived more easily than the multigrid method does.

In this paper, we firstly induce the FDEs to be a system of ODEs via using the spatial semi-discretizing method. Then the BVM is employed to solve the ODEs system. Meanwhile, we apply the GMRES [37] with the block-circulant type preconditioners to solve linear systems arising from the application of BVM, which is a fairly new method based on the linear multistep formulae to solve ODEs. The BVMs are unconditionally stable and are high-accuracy schemes for solving initial value problems (IVPs) based on the linear multistep formulas [38,39]. Unlike Runge–Kutta or other initial value methods (IVMs), BVMs achieve the advantage of both good stability and high-order accuracy [39,40]. The main purpose of this paper is to investigate the effectiveness of preconditioning on the speed of the resulting iterative processes of BVMs for solving FDEs.

The rest of present paper is organized as follows. In Section 2, the background of the spatial discretization for the FDE to reduce the system of ODEs is reviewed. Then we introduce that how to result in the linear systems by block-BVMs. In Section 3, we construct the block circulant-type preconditioner and BCCB preconditioner [41]. Then the invertibility of two different kinds of preconditioners and the convergence rate and computational cost of the preconditioned GMRES method are also studied. In Section 4, extensive numerical results are reported to display the performance of the proposed method.

2. Semi-discretization of FDEs and boundary value methods

In this paper, we study an initial–boundary value problem of the FDEs as follows,

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = d_+(x,t)\dfrac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} + d_-(x,t)\dfrac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} + f(x,t), \\ x \in (x_L, x_R), \quad t \in (t_0, T], \\ u(x_L,t) = u(x_R,t) = 0, \quad 0 \le t \le T, \\ u(x,t_0) = u_0(x), \quad x \in [x_L, x_R], \end{cases} \tag{1}$$

where $\alpha \in (1,2)$ is the order of the fractional derivative, $f(x,t)$ is the source (or sinks) term, and diffusion coefficient functions $d_\pm(x,t)$ are nonnegative; i.e., $d_\pm(x,t) \ge 0$, $d_+(x,t) + d_-(x,t) \ne 0$. The function $u(x,t)$ can be interpreted as representing the concentration of a particle plume undergoing anomalous diffusion.

2.1. Semi-discretization of FDEs via finite difference method

Meerschaert and Tadjeran [17] have shown that using the shifted Grünwald formula to approximate the two-sided fractional derivatives of order $\alpha \in (1,2)$ leads to stable numerical schemes. We begin this method, it is known that the left-sided and the right-sided fractional derivatives $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ and $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$ are defined in the Grünwald–Letnikov form [42]

$$\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} = \lim_{\Delta x \to 0^+} \frac{1}{\Delta x^\alpha} \sum_{k=0}^{\lfloor (x-x_L)/\Delta x \rfloor} g_k^{(\alpha)} u(x - k\Delta x, t),$$

$$\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} = \lim_{\Delta x \to 0^+} \frac{1}{\Delta x^\alpha} \sum_{k=0}^{\lfloor (x_R-x)/\Delta x \rfloor} g_k^{(\alpha)} u(x + k\Delta x, t),$$

where $\lfloor\cdot\rfloor$ denotes the floor function, and $g_k^{(\alpha)}$ is the alternating fractional binomial coefficient given as

$$\begin{cases} g_0^{(\alpha)} = 1, \\ g_k^{(\alpha)} = \dfrac{(-1)^k}{k!}\alpha(\alpha-1)\cdots(\alpha-k+1), \quad k = 1, 2, 3, \ldots, \end{cases} \tag{2}$$

which can be evaluated by applying the recurrence relation

$$g_{k+1}^{(\alpha)} = \Big(1 - \frac{\alpha+1}{k+1}\Big)g_k^{(\alpha)}, \quad k = 0, 1, 2, \ldots.$$

Let N be a positive integer and $\Delta x = \frac{x_R - x_L}{N+1}$ be the size of spatial grid. We define a spatial partition $x_i = x_L + i\Delta x$ for $i = 0, 1, \ldots, N+1$. Let $u_i = u(x_i, t)$, $d_{\pm,i} = d_\pm(x_i, t)$, and $f_i = f(x_i, t)$. The shifted Grünwald approximation in [16,17] is written as follows,

$$\frac{\partial^\alpha u(x_i, t)}{\partial_+ x^\alpha} = \frac{1}{\Delta x^\alpha}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1} + \mathcal{O}(\Delta x),$$
$$\frac{\partial^\alpha u(x_i, t)}{\partial_- x^\alpha} = \frac{1}{\Delta x^\alpha}\sum_{k=0}^{N-i+2} g_k^{(\alpha)} u_{i+k-1} + \mathcal{O}(\Delta x),$$

where $g_k^{(\alpha)}$ is defined in (2), and the spatial semi-discretization, i.e., its corresponding system of ODEs for the (1) is given as follows,

$$\begin{cases} \dfrac{d\boldsymbol{u}(t)}{dt} = J_N\boldsymbol{u}(t) + \boldsymbol{f}(t), \quad t \in (t_0, T], \\ \boldsymbol{u}(t_0) = [u_0(x_1), u_0(x_2), \ldots, u_0(x_N)]^{\mathrm{T}} = \boldsymbol{u}_0, \end{cases} \tag{3}$$

where $\boldsymbol{u}(t) = [u_1, u_2, \ldots, u_N]^{\mathrm{T}}$, $\boldsymbol{f}(t) = [f_1, f_2, \ldots, f_N]^{\mathrm{T}}$, $\Delta x^\alpha = \frac{(x_R - x_L)^\alpha}{(N+1)^\alpha}$, and J_N is the coefficient matrix with an appropriate size and can be given in the following

$$J_N = \frac{1}{\Delta x^\alpha}\Big[D_+ G_\alpha + D_- G_\alpha^{\mathrm{T}}\Big], \tag{4}$$

with $D_\pm = \mathrm{diag}(d_{\pm,1}, d_{\pm,2}, \ldots, d_{\pm,N})$ and

$$G_\alpha = \begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 \\ \vdots & g_2^{(\alpha)} & g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{N-1}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_N^{(\alpha)} & g_{N-1}^{(\alpha)} & \cdots & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}_{N\times N}. \tag{5}$$

It is obvious that G_α is a Toeplitz matrix (see [34,43]). Therefore, it can be stored with $N+1$ entries [32]. Furthermore, the matrix–vector multiplication for the Toeplitz-like matrix J_N in (4) can be computed in $\mathcal{O}(N\log N)$ operations by the FFT; see [33,35]. The alternating fractional binomial coefficient $g_k^{(\alpha)}$ have some useful properties, that are observed in [16,17,32], and are summarized in the following proposition.

Proposition 1 ([36]). *Let $1 < \alpha < 2$ and $g_k^{(\alpha)}$ be defined in (2). We have*

$$\begin{cases} g_0^{(\alpha)} = 1, \quad g_1^{(\alpha)} = -\alpha < 0, \quad g_2^{(\alpha)} > g_3^{(\alpha)} > \cdots > 0, \\ \displaystyle\sum_{k=0}^{\infty} g_k^{(\alpha)} = 0, \quad \sum_{k=0}^{n} g_k^{(\alpha)} < 0, \quad \forall n \geq 1. \end{cases} \tag{6}$$

Also, we give the following conclusion which is very helpful for theoretical analysis,

Proposition 2. *Let $1 < \alpha < 2$ and $g_k^{(\alpha)}$ be defined in (2). All eigenvalues of J_N fall inside the open disc*

$$\{z \in \mathbb{C} : \ |z - \gamma_i| < -\gamma_i\}, \quad i = 1, \ldots, N,$$

where $\gamma_i = (r_{+,i} + r_{-,i})g_1^{(\alpha)} < 0$ are constants.

Proof. Here the entries of the matrix J_N are given by

$$p_{ij} = \begin{cases} (r_{+,i} + r_{-,i})g_1^{(\alpha)}, & j = i, \\ r_{+,i}g_2^{(\alpha)} + r_{-,i}g_0^{(\alpha)}, & j = i-1, \\ r_{+,i}g_0^{(\alpha)} + r_{-,i}g_2^{(\alpha)}, & j = i+1, \\ r_{+,i}g_{i-j+1}^{(\alpha)}, & j < i-1, \\ r_{-,i}g_{j-i+1}^{(\alpha)}, & j > i+1, \end{cases} \tag{7}$$

where $r_{\pm,i} = \frac{d_{\pm,i}}{\Delta x^\alpha} \geq 0$. It is not hard to find that $p_{ij} \leq 0$ for all $i \neq j$, then all the Gershgorin discs of the matrix are centered at $\gamma_i = (r_{+,i} + r_{-,i})g_1^{(\alpha)} < 0$ with radius

$$\begin{aligned} R_i &= \sum_{j=1, j\neq i}^{N} |p_{ij}| = r_{+,i} \sum_{k=0, k\neq 1}^{i} g_k^{(\alpha)} + r_{-,i} \sum_{k=0, k\neq 1}^{N-i+1} g_k^{(\alpha)} \quad (r_{+,i} + r_{-,i} \neq 0) \\ &< (r_{+,i} + r_{-,i}) \sum_{k=0, k\neq 1}^{\infty} g_k^{(\alpha)} = -(r_{+,i} + r_{-,i})g_1^{(\alpha)} = -\gamma_i, \end{aligned}$$

by applying the properties of the sequence $g_k^{(\alpha)}$; see Proposition 1.

Remark 1. It is worth noting that:

(i) The real parts of all eigenvalues of the matrix J_N are strictly negative for all N.

(ii) The absolute values of all eigenvalues of the matrix J_N are bounded above by $\max_{1\leq j\leq N}\{2|\gamma_i|\}$.

2.2. Boundary value methods

Next, we consider a class of robust numerical methods called the BVMs for solving the system of ODEs, see [38,44]. Using the μ-step block-BVM over a uniform mesh $h = (T - t_0)/s$ for the discretization of Eq. (3), we obtain

$$\sum_{i=-\nu}^{\mu-\nu} \alpha_{i+\nu} \boldsymbol{u}_{n+i} = h \sum_{i=-\nu}^{\mu-\nu} \beta_{i+\nu} \boldsymbol{g}_{n+i}, \quad n = \nu, \ldots, s - \mu + \nu. \tag{8}$$

Here $\boldsymbol{u}_n$ is the discrete approximation to $\boldsymbol{u}(t_n)$, $\boldsymbol{g}_n = J_N \boldsymbol{u}_n + \boldsymbol{f}_n$, and $\boldsymbol{f}_n = \boldsymbol{f}(t_n)$. Also, Eq. (8) requires ν initial conditions and $\mu - \nu$ final conditions which are provided by the following $\mu - 1$ additional equations:

$$\sum_{i=0}^{\mu} \alpha_i^{(j)} \boldsymbol{u}_i = h \sum_{i=0}^{\mu} \beta_i^{(j)} \boldsymbol{g}_i, \quad j = 1, 2, \ldots, \nu - 1, \tag{9}$$

and

$$\sum_{i=0}^{\mu} \alpha_{\mu-i}^{(j)} \boldsymbol{u}_{n-i} = h \sum_{i=0}^{\mu} \beta_{\mu-i}^{(j)} \boldsymbol{g}_{n-i}, \quad j = s - \mu + \nu + 1, \ldots, s. \tag{10}$$

The coefficients $\{\alpha_k^{(j)}\}$ and $\{\beta_k^{(j)}\}$ in Eqs. (9) and (10) should be selected such that truncation errors in these $\mu - 1$ equations are of the same order as that in Eq. (8). We combine Eqs. (8)–(10) and the initial condition $\boldsymbol{u}(t_0) = \boldsymbol{u}_0$, a discrete system of linear equations for Eq. (3) is obtained by the following block matrix form,

$$M\boldsymbol{u} \equiv (A \otimes I_N - hB \otimes J_N)\boldsymbol{u} = \boldsymbol{e}_1 \otimes \boldsymbol{u}_0 + h(B \otimes I_N)\boldsymbol{f}. \tag{11}$$

Here $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{s+1}$ and

$$\boldsymbol{u} = (\boldsymbol{u}_0^T, \ldots, \boldsymbol{u}_s^T)^T \in \mathbb{R}^{(s+1)N}, \qquad \boldsymbol{f} = (\boldsymbol{f}_0^T, \ldots, \boldsymbol{f}_s^T)^T \in \mathbb{R}^{(s+1)N}.$$

In (11), the matrix $A \in \mathbb{R}^{(s+1)\times(s+1)}$ is given by:

$$A = \begin{bmatrix}
1 & \cdots & 0 & & & & & \\
\alpha_0^{(1)} & \cdots & \alpha_\mu^{(1)} & & & & & \\
\vdots & \vdots & \vdots & & & & & \\
\alpha_0^{(\nu-1)} & \cdots & \alpha_\mu^{(\nu-1)} & & & & & \\
\alpha_0 & \cdots & \alpha_\mu & & & & & \\
 & \alpha_0 & \cdots & \alpha_\mu & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & \ddots & \ddots & \ddots & & \\
 & & & & \alpha_0 & \cdots & \alpha_\mu & \\
 & & & & \alpha_0^{(s-\mu+\nu+1)} & \cdots & \alpha_\mu^{(s-\mu+\nu+1)} & \\
 & & & & \vdots & \vdots & \vdots & \\
 & & & & \alpha_0^{(s)} & \cdots & \alpha_\mu^{(s)} &
\end{bmatrix},$$

and $B \in \mathbb{R}^{(s+1)\times(s+1)}$ is defined similarly by using β's instead of α's in A and the first row of B is zeros.

Usually the resulting linear system (11) is large and ill-conditioned, and solving it is a core problem in the application of BVMs. If a direct method is employed to solve the system (11), the operation cost can be very high for practical application. Therefore interest has been turned to iterative solvers, such as GMRES method. As we know that a clustered spectrum often translates in rapid convergence of GMRES method [45], so we use the GMRES method for solving the resulting linear system (11). In order to accelerate the convergence of GMRES iterations, we construct some block circulant-type preconditioners.

3. Construction of preconditioners and convergence analysis

In this section, we will show how to construct the block circulant-type preconditioners for accelerating the iterative solver and show that these preconditioners are invertible if an A_{ν_1,ν_2}-stable BVM is used. Meanwhile, some theoretical analyses on both the convergence rate of iterative solver and operation cost for each iteration are also investigated.

3.1. Construction of preconditioners

To mimic the terminology of [46] and neglect the perturbations in the upper left and low right corners of A and B, we give the first preconditioner for Eq. (11):

$$S = s(A) \otimes I_N - hs(B) \otimes J_N, \tag{12}$$

where

$$s(A) = \begin{bmatrix}
\alpha_\nu & \cdots & \alpha_\mu & & & \alpha_0 & \ldots & \alpha_{\nu-1} \\
\vdots & \ddots & & \ddots & & & \ddots & \vdots \\
\alpha_0 & & \ddots & & \ddots & & & \alpha_0 \\
 & \ddots & & \ddots & & \ddots & 0 & \\
 & & \ddots & & \ddots & & \ddots & \\
 & 0 & & \ddots & & \ddots & & \\
\alpha_\mu & & & & \ddots & & \ddots & \alpha_\mu \\
\vdots & \ddots & & & & \ddots & & \vdots \\
\alpha_{\nu+1} & \cdots & \alpha_\mu & & & \alpha_0 & \cdots & \alpha_\nu
\end{bmatrix}$$

and $s(B)$ is defined similarly by using $\{\beta_i\}_{i=0}^{\mu}$ instead of $\{\alpha_i\}_{i=0}^{\mu}$ in $s(A)$. The $\{\alpha_i\}_{i=0}^{\mu}$ and $\{\beta_i\}_{i=0}^{\mu}$ here are the coefficients in Eq. (8). We note that $s(A)$ and $s(B)$ are the generalized Strang-type circulant preconditioners of A and B respectively, see [34].

Moreover, we also can propose the Strang-type BCCB preconditioner, which can be constructed as follow for solving Eq. (11)

$$S^{(2)} = s(A) \otimes I_N - hs(B) \otimes s(J_N) \tag{13}$$

with J_N being mentioned in (4). Here we define the $s(J_N)$ as following matrix form

$$s(J_N) = \frac{1}{\Delta x^{\alpha}}\left(\overline{d}_{+}s(G_{\alpha}) + \overline{d}_{-}s(G_{\alpha}^{\mathrm{T}})\right) \tag{14}$$

with $\overline{d}_{\pm} = \frac{1}{N}\sum_{i=1}^{N} d_{\pm,i}$. More precisely, the first columns of $s(G_\alpha)$ and $s(G_\alpha^{\mathrm{T}})$ are given by

$$\begin{bmatrix} g_1^{(\alpha)} \\ \vdots \\ g_{\lfloor\frac{N+1}{2}\rfloor}^{(\alpha)} \\ 0 \\ \vdots \\ 0 \\ g_0^{(\alpha)} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} g_1^{(\alpha)} \\ g_0^{(\alpha)} \\ 0 \\ \vdots \\ 0 \\ g_{\lfloor\frac{N+1}{2}\rfloor}^{(\alpha)} \\ g_0^{(\alpha)} \end{bmatrix}.$$

As we know, Lei and Sun [36] proposed Strang circulant preconditioner to approximate the coefficient matrix with structure as the sum of diagonal-multiply-Toeplitz matrices. The convergent behavior of this method are very efficient and robust in numerical experiments. So we take the similar strategy to construct the preconditioner $S^{(2)}$. The advantage of BCCB preconditioners is that the operation cost in each iteration of the GMRES method for the preconditioned systems is much less than that required by using any block-circulant preconditioners.

Next, we will demonstrate that the preconditioner S is invertible provided that the given BVM is stable and the eigenvalues of J_N are in the negative half of the complex plane $\mathbb{C}$. Also the invertibility of the preconditioner $S^{(2)}$ will be analyzed and improved. The stability of a BVM is related to two characteristic polynomials of degree μ, defined as follows:

$$\rho(z) = z^{\nu}\sum_{j=-\nu}^{\mu-\nu}\alpha_{j+\nu}z^{j} \quad \text{and} \quad \sigma(z) = z^{\nu}\sum_{j=-\nu}^{\mu-\nu}\beta_{j+\nu}z^{j}. \tag{15}$$

Definition 1 ([38, p. 101]). Consider a BVM with the characteristic polynomials $\rho(z)$ and $\sigma(z)$ given by (12). The region

$$\mathcal{D}_{\nu,\mu-\nu} = \left\{q \in \mathbb{C} : \rho(z) - q\sigma(z) \text{ has } \nu \text{ zeros inside } |z| = 1 \text{ and } \mu-\nu \text{ zeros outside } |z| = 1\right\}$$

is called the region of $A_{\nu,\mu-\nu}$-stability of the given BVM. Moreover, the BVM is said to be $A_{\nu,\mu-\nu}$-stable if

$$\mathbb{C}^{-} \equiv \{q \in \mathbb{C} : \ \mathrm{Re}(q) < 0\} \subseteq \mathcal{D}_{\nu,\mu-\nu}.$$

Theorem 1 ([46]). *If the BVM for (3) is $A_{\nu,\mu-\nu}$-stable and $h\lambda_k(J_N) \in \mathcal{D}_{\nu,\mu-\nu}$ where $\lambda_k(J_N)$ $(k = 1, \ldots, N)$ are the eigenvalues of J_N, then the preconditioner S in (12) is invertible.*

In particular, we have

Corollary 1 ([46]). *If the BVM for (3) is $A_{\nu,\mu-\nu}$-stable and $h\lambda_k(J_N) \in \mathbb{C}^{-}$, then the preconditioner S is invertible.*

In fact, we note that the eigenvalues of J_N are indeed in the negative half of the complex plane $\mathbb{C}^{-}$ via Proposition 2 and Remark 1. So if we add the condition that the given BVM is stable, we can immediately conclude that the preconditioner S is invertible. It implies that this preconditioner can be expected to be robust and efficient.

Similar to Theorem 1, we can show that if the BVM for (4) is $A_{\nu,\mu-\nu}$-stable and the eigenvalues of $s(J_N)$ satisfy

$$\lambda_k(s(J_N)) \in \mathbb{C}^{-}$$

for $k = 1, \ldots, N$, then the preconditioner $S^{(2)}$ is invertible.

However, for some special FDE problems, the matrix J_N is usually the complete Toeplitz-like structure, but $s(J_N)$ may be singular. Note that the eigenvalues of $S^{(2)}$ are defined by

$$\lambda_{jk}(S^{(2)}) = \phi_j - h\psi_j\lambda_k(s(J_N)), \quad j = 0, \ldots, s, \ k = 1, \ldots, N, \tag{16}$$

where ϕ_j and ψ_j are the eigenvalues of $s(A)$ and $s(B)$ respectively. When some eigenvalues of $s(J_N)$ are zero, then some eigenvalues of $S^{(2)}$ is the same as the eigenvalues of the matrix $s(A)$. It is well-known that the eigenvalues of the circulant matrix $s(A)$ can be expressed as the following sum, see [47],

$$\phi_j = \sum_{r=-\nu}^{\mu-\nu}\alpha_{r+\nu}\omega^{rj}, \quad \omega = e^{2\pi\mathbf{i}/(s+1)}, \ j = 0, \ldots, s,$$

where $\alpha_0, \ldots, \alpha_n$ are the coefficients of the first characteristic polynomial in (15).

From the characteristic polynomials defined in (15), the coefficients must satisfy the consistent conditions,

$$\rho(1) = 0 \quad \text{and} \quad \rho'(1) = \sigma(1).$$

Thus, we have

$$\phi_0 = \rho(1) = 0$$

for any consistent BVM. From (16), we know that $S^{(2)}$ is singular when some eigenvalues of $s(J_N)$ are zero. In this case, we move the zero eigenvalue of $s(A)$ to a nonzero value. More precisely, we change the matrix $s(A) = F\text{diag}(\phi_0, \ldots, \phi_s)F^*$ to

$$\widetilde{s}(A) \equiv F\text{diag}(\widetilde{\phi}_0, \ldots, \phi_s)F^*,$$

where $\widetilde{\phi}_0 \equiv \text{Re}(\phi_s)$ and F is the Fourier matrix. Define

$$\widetilde{S}^{(2)} \equiv \widetilde{s}(A) \otimes I_N - hs(B) \otimes s(J_N), \tag{17}$$

we can also prove that $\widetilde{S}^{(2)}$ is invertible, see [41] for details.

From the conclusions of [36], we can obtain the following theorem,

Theorem 2. *All eigenvalues of circulant matrices $s(G_\alpha)$ and $s(G_\alpha^{\mathrm{T}})$ fall inside the open disc*

$$\{z \in \mathbb{C} : \ |z + \alpha| < \alpha\}.$$

By Theorem 2, we can find that the parts of all eigenvalues of $s(G_\alpha)$ and $s(G_\alpha^{\mathrm{T}})$ are strictly negative for all N. Moreover, we know that $\overline{d}_\pm \geq 0,\ \overline{d}_+ + \overline{d}_- \neq 0$. So we can conclude that

$$\text{Re}(\lambda_k(s(J_N))) = \frac{1}{\Delta x^\alpha}\left(\overline{d}_+\text{Re}(s(G_\alpha)) + \overline{d}_-\text{Re}(s(G_\alpha^{\mathrm{T}}))\right) < 0.$$

It means that the eigenvalues of $s(J_N)$ are in the negative half of the complex plane $\mathbb{C}^-$ and then both the preconditioners $S^{(2)}$ and $\widetilde{S}^{(2)}$ are invertible provided that the given BVM is stable, refer to [41, Theorem 2, p. 32].

3.2. Convergence rate and operation cost

Following the conclusions in [46], we have the following theorems for the convergence rates.

Theorem 3 ([46])**.** *We have*

$$S^{-1}M = I + L$$

where I is the identity matrix and the rank of L is at most $2N\mu$. Therefore, when the GMRES method is applied to solve $S^{-1}M\boldsymbol{y} = S^{-1}\boldsymbol{b}$, the method will converge in at most $2N\mu + 1$ iterations in exact arithmetic.

Lei and Jin [41] proved that when J_N is a Toeplitz matrix in the *Wiener class* [33,34], the preconditioned matrix $(\widetilde{S}^{(2)})^{-1}M$ can be written as the sum of the identity matrix, a matrix with rank $\mathcal{O}(N)$, a matrix with rank $\mathcal{O}(s)$ and a matrix with small norm. Now we are going to analyze the spectrum of preconditioned matrix $(\widetilde{S}^{(2)})^{-1}M$. In fact, for Eq. (3), when we take

$$d_{+,i} = d_+ \geq 0, \qquad d_{-,i} = d_- \geq 0 \quad \text{and} \quad d_+ + d_- \neq 0, \tag{18}$$

for all $i = 1, \ldots, N$. Then we obtain a nonsymmetric Toeplitz matrix as follows,

$$J_N = \frac{1}{\Delta x^\alpha}\left(d_+G_\alpha + d_-G_\alpha^{\mathrm{T}}\right) = \frac{T_N}{\Delta x^\alpha} = \frac{[t_{j-k}]_{N\times N}}{\Delta x^\alpha}, \tag{19}$$

where

$$T_N = \begin{bmatrix} d_+g_1^{(\alpha)}+d_-g_1^{(\alpha)} & d_+g_0^{(\alpha)}+d_-g_2^{(\alpha)} & d_-g_3^{(\alpha)} & \cdots & d_-g_2^{(\alpha)} \\ d_+g_2^{(\alpha)}+d_-g_0^{(\alpha)} & d_+g_1^{(\alpha)}+d_-g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ d_+g_3^{(\alpha)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & d_+g_0^{(\alpha)}+d_-g_2^{(\alpha)} \\ d_+g_N^{(\alpha)} & \cdots & \cdots & d_+g_2^{(\alpha)}+d_-g_0^{(\alpha)} & d_+g_1^{(\alpha)}+d_-g_1^{(\alpha)} \end{bmatrix}.$$

We introduce the *generating function* of the sequence of Toeplitz matrices $\{T_N\}_{N=1}^{\infty}$ [33]:

$$p(\theta) = \sum_{k=-\infty}^{\infty} t_k e^{\mathbf{i}k\theta}, \tag{20}$$

where t_k is the kth diagonal of T_N. The generating function $p(\theta)$ is in the Wiener class if and only if

$$\sum_{k=-\infty}^{\infty} |t_k| < \infty.$$

For T_N defined in (19), we have

$$p(\theta) = \sum_{k=-\infty}^{\infty} t_k e^{\mathbf{i}k\theta} = \sum_{k=-1}^{\infty} g_{k+1}^{(\alpha)}(d_+ e^{\mathbf{i}k\theta} + d_- e^{\mathbf{i}k\theta}) \tag{21}$$

and obtain the following theorem.

Theorem 4. *Let p be the generating function of $\{T_N\}_{N=1}^{\infty}$, we conclude that p is in the Wiener class.*

Proof. By using the properties of the sequence $\{g_{k+1}^{(\alpha)}\}_{k=0}^{\infty}$ given in (6), we have

$$\begin{aligned}\sum_{k=-\infty}^{\infty} |t_k| &= (d_+ + d_-) \sum_{k=-1}^{\infty} |g_{k+1}^{(\alpha)}| \\ &= (d_+ + d_-)\Big(-2g_1^{(\alpha)} + \sum_{k=0}^{\infty} g_k^{(\alpha)}\Big) \\ &= 2\alpha(d_+ + d_-) < \infty.\end{aligned}$$

Thus p is in the Wiener class.

Moreover, let

$$E \equiv M - \widetilde{S}^{(2)}, \qquad E_1 \equiv M - S, \qquad E_2 \equiv S - \widetilde{S}^{(2)}.$$

Then $E = E_1 + E_2$. For E_1, we have the following lemma.

Lemma 1 (*[48]*)**.** *We have*

$$\text{rank} \le 2N\mu = \mathcal{O}(N),$$

where μ is given by the BVMs used for (3).

For the matrix E_2, by Eqs. (12) and (17), we have

$$\begin{aligned}E_2 &= (s(A) - \widetilde{s}(A)) \otimes I_N - hs(B) \otimes (J_N - s(J_N)) \\ &= L_A \otimes I_N - hs(B) \otimes L_J,\end{aligned}$$

where $L_A = s(A) - \widetilde{s}(A)$ and $L_J = J_N - s(J_N)$. Applying the conclusion of [41], we have

$$\text{rank}(L_A \otimes I_N) \le N = \mathcal{O}(N). \tag{22}$$

From the Eqs. (14), (18) and (19), we can obtain

$$s(J_N) = \frac{s(T_N)}{\Delta x^{\alpha}} \quad \text{with } s(T_N) = d_+ s(G_\alpha) + d_- s(G_\alpha^{\mathrm{T}}).$$

For the term $s(B) \otimes L_J$ in E_2, since T_N is a Toeplitz in the Winer class (see Theorem 4), the matrix $\Delta x^{\alpha} L_J = (T_N - s(T_N))$ can be expressed as a sum of a matrix with low rank and a matrix with small norm, see [41]. More precisely, for any given $\varepsilon > 0$, there exists a constant $C(\varepsilon)$ such that

$$(T_N - s(T_N)) = \Delta x^{\alpha} L_J = U + V \quad \text{with rank } (U) \le C(\varepsilon) \quad \text{and} \quad \|V\|_2 \le \varepsilon, \tag{23}$$

when N is sufficiently large. Then we have

$$s(B) \otimes L_J = s(B) \otimes U' + s(B) \otimes (V/\Delta x^{\alpha}), \quad U' = U/\Delta x^{\alpha}, \tag{24}$$

with

$$\text{rank}(s(B) \otimes U') \le s \cdot C(\varepsilon) = \mathcal{O}(s). \tag{25}$$

For $\|s(B) \otimes (V/\Delta x^{\alpha})\|_2$, we note that

$$\|s(B)\|_1 = N_1 < \infty, \qquad \|s(B)\|_\infty = N_2 < \infty, \qquad \|s(B)\|_2 = N_3 < \infty, \tag{26}$$

where N_1 and N_2 are two constants independent of the size of matrices, and $N_3 = \sqrt{N_1 N_2}$, see [41]. Furthermore, we have by (23) and (26)

$$\|s(B) \otimes (V/\Delta x^{\alpha})\|_2 = \|s(B)\|_2 \|(V/\Delta x^{\alpha})\|_2 \leq \frac{\varepsilon N_3}{\Delta x^{\alpha}}. \tag{27}$$

By using (22), (24), (25) and (27), we know that for any $\varepsilon > 0$, the matrix E_2 can be decomposed as

$$E_2 = L_{\mathcal{O}(N)} + hL_{\mathcal{O}(s)} + hW \tag{28}$$

with $\operatorname{rank}(L_{\mathcal{O}(N)}) = \mathcal{O}(N)$, $\operatorname{rank}(L_{\mathcal{O}(s)}) = \mathcal{O}(s)$, $\|W\|_2 \leq \varepsilon/\Delta x^{\alpha}$. In conclusion, we then have the following theorem for the spectrum of $(\widetilde{S}^{(2)})^{-1}M$.

Theorem 5. *Under the conditions of (18) and Theorem 4, then the preconditioned matrix $(\widetilde{S}^{(2)})^{-1}M$ can be written as the sum of the identity matrix, a matrix with rank $\mathcal{O}(N)$, a matrix with rank $\mathcal{O}(s)$ and a matrix with the norm $\frac{\varepsilon}{\Delta x^{\alpha}}$ ($\varepsilon > 0$ is usually small).*

Proof. The proof of this theorem is greatly similar to that of [41, Theorem 3], we omit here.

As a consequence, the most of eigenvalues of $(\widetilde{S}^{(2)})^{-1}M$ is clustered at 1 (also see Fig. 1). Moreover, the GMRES method, when applied to solve the preconditioned linear systems

$$(\widetilde{S}^{(2)})^{-1}M\boldsymbol{y} = (\widetilde{S}^{(2)})^{-1}\boldsymbol{b}$$

will converge fast. Therefore, a detailed analysis for the convergence rate could be carried out in future work.

Regarding the operation cost per iteration, the main work in each iteration for the GMRES method is the matrix–vector multiplication

$$S^{-1}M\boldsymbol{z} = (s(A) \otimes I_N - hs(B) \otimes J_N)^{-1}(A \otimes I_N - hB \otimes J_n)\boldsymbol{z}$$

where $\boldsymbol{z}$ is a vector, see for instant Saad [37]. Since A and B are band matrices and J_N is a full matrix, the matrix–vector multiplication $M\boldsymbol{z} = (A \otimes I_N - hB \otimes J_n)\boldsymbol{z}$ can be implemented not slowly.

To calculate $S^{-1}M\boldsymbol{z}$, since $s(A)$ and $s(B)$ are circulant matrices, we have the following decompositions via the FFTs

$$s(A) = F\Lambda_A F^* \quad \text{and} \quad s(B) = F\Lambda_B F^*,$$

where Λ_A and Λ_B are diagonal matrices containing the eigenvalues of $s(A)$ and $s(B)$ respectively, see [47]. It follows that

$$S^{-1}(M\boldsymbol{z}) = (F^* \otimes I_N)(\Lambda_A \otimes I_N - h\Lambda_B \otimes J_N)^{-1}(F \otimes I_N)(M\boldsymbol{z}).$$

This multiplication can be achieved by using FFTs and solving s linear systems of order N, refer to [48]. It follows that the total number of operations per iteration is $\mathcal{O}(Ns\log s + sN\zeta)$, where ζ is the number of nonzeros of J_N. For comparing the computational cost of the method with direct solver for the linear systems (11), refer to [48]. However, in the case of numerical method for FDEs, the coefficient matrix J_N is full, it means that ζ is much large. We need to take much time to solve s (Toeplitz-like) linear systems of order N, this shortage will keep the preconditioner S from becoming the efficient one. In order to overcome this shortage, we propose the preconditioners $S^{(2)}$ and $\widetilde{S}^{(2)}$. For simplicity, we assume that $s + 1 = N$ in the following analysis of the operation cost of preconditioners $S^{(2)}$ and $\widetilde{S}^{(2)}$. Regarding the operation cost in each iteration of the GMRES method, the main work is the matrix–vector multiplication

$$(\widetilde{S}^{(2)})^{-1}M\boldsymbol{v} \equiv (\widetilde{s}(A) \otimes I_N - hs(B) \otimes s(J_N))^{-1}M\boldsymbol{v},$$

where $\boldsymbol{v}$ is a vector. Since $(\widetilde{S}^{(2)})^{-1}$ can be diagonalized by exploiting the 2-dimensional Fourier matrix, i.e.,

$$(\widetilde{S}^{(2)})^{-1}M\boldsymbol{v} \equiv (F_{s+1} \otimes F_N)(\Lambda_A \otimes I_N - h\Lambda_B \otimes \Lambda_{J_N})^{-1}(F_{s+1}^* \otimes F_N^*)(M\boldsymbol{v}),$$

where $s(J_N) = F_N^* \Lambda_{J_N} F_N$ and Λ_{J_N} is a diagonal matrix holding the eigenvalues of $s(J_N)$. The matrix–vector multiplication $(\widetilde{S}^{(2)})^{-1}M\boldsymbol{v}$ can be implemented within $\mathcal{O}(N^2 \log N)$ operations via FFTs. For the Strang-type block-circulant preconditioner S defined as the form (12), in each iteration, there are N Toeplitz-like systems of order N needed to be solved. Thus, the complexity in each iteration of the preconditioners $S^{(2)}$ and $\widetilde{S}^{(2)}$ is much lower.

4. Numerical experiments

In this section, we solve two different FDE problems (1) numerically by employing the BVM and the GMRES method together with the circulant-type preconditioners in Sections 2–3. We also compare the Strang-type BCCB preconditioners $S^{(2)}$ and $\widetilde{S}^{(2)}$ with the Strang-type block-circulant preconditioner S. The number of iterations required for convergence and CPU time of those methods are reported. In these examples, the BVM we used here is the fifth order GAM which has $\mu = 4$. Its formulae and the additional initial and final conditions can be found in Ref. [38].

All experiments are performed in MATLAB 2011b and all the computations are run on an Inter(R) Pentium(R) CPU 2.80 GHz PC with 3.85G available memory. We use the MATLAB-provided M-file 'gmres' (see MATLAB on-line documentation) to solve the preconditioned systems. We use donations "Iters" and "CPU" to represent the number of iterations and

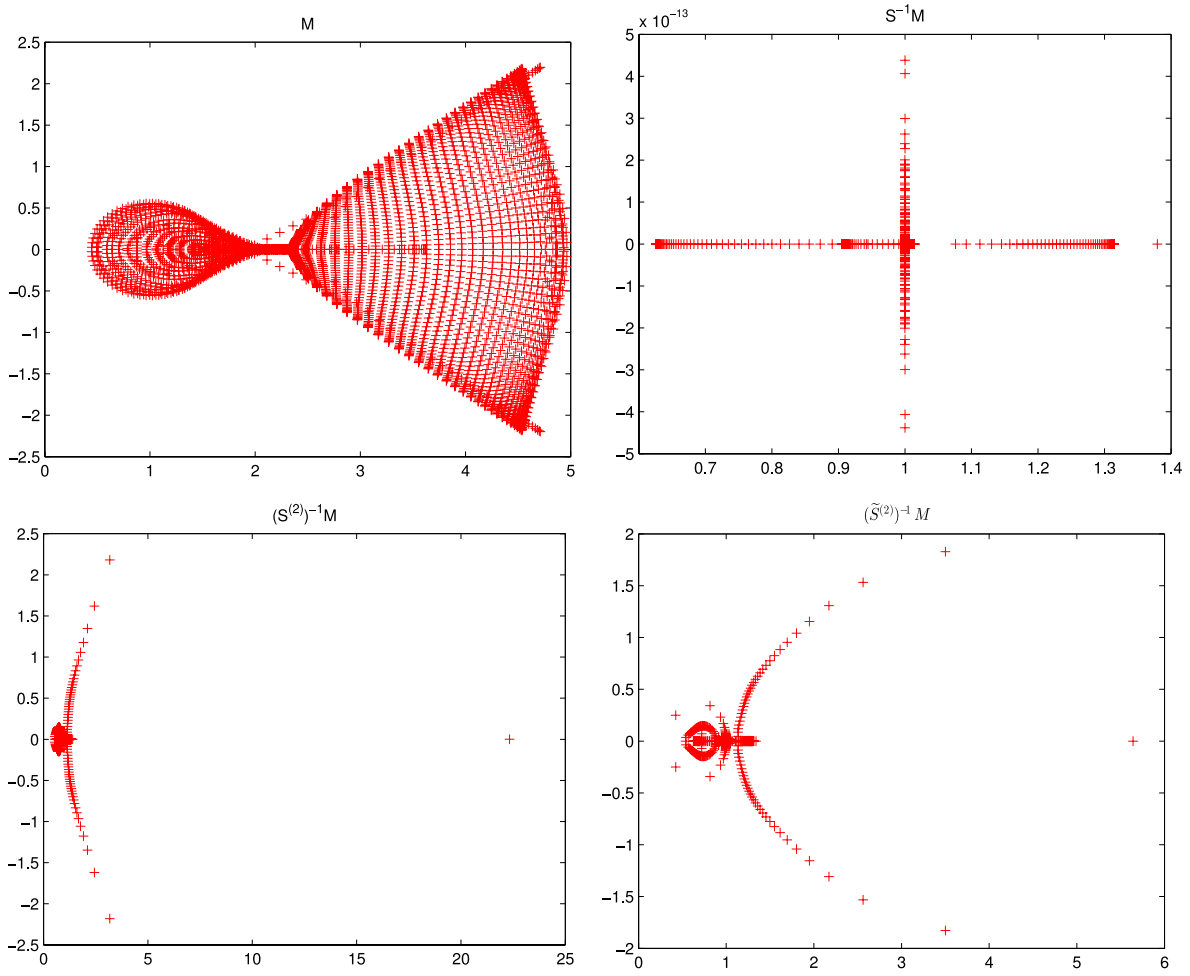

Fig. 1. The spectra of matrix M and different preconditioned matrices with $N = 48,\ s = 64$ for the Example 1.

CPU elapsed time (mean value from ten times repeated experiments) of implementing GMRES(20) solver, respectively. In our tests, the initial guess is the zero vector and stopping criterion in the GMRES method is

$$\frac{\|\boldsymbol{r}_q\|_2}{\|\boldsymbol{r}_0\|_2} < 10^{-8},$$

where $\boldsymbol{r}_q$ is the residual after the qth iterations.

Example 1. In this example, we solve the initial–boundary value problem of FDE (1) with source term $f(x, t) \equiv 0$, for the order of fractional derivatives $\alpha = 1.2$ and 1.5. The spatial domain is $[x_L, x_R] = [0, 2]$ and the time interval is $[t_0, T] = [0, 1]$. The initial condition $u(x, 0)$ is the following Gaussian pulse

$$u(x, 0) = \exp\Big(-\frac{(x - x_c)^2}{2\xi^2}\Big), \quad x_c = 1.2,\ \xi = 0.08,$$

and the diffusion coefficients

$$d_+(x, t) \equiv 0.6, \quad \text{and} \quad d_-(x, t) \equiv 0.5.$$

Tables 1 and 2 list the number of iterations required for convergence of the GMRES method with different preconditioners and their corresponding CPU time. In the tables, I means that no preconditioner is used, and S, $S^{(2)}$ and $\widetilde{S}^{(2)}$ denote the Strang-type block-circulant preconditioners, Strang-type and modified Strang-type BCCB preconditioners respectively, see (12), (13) and (17).

For Example 1, the number of iterations of both $S^{(2)}$ and $\widetilde{S}^{(2)}$ are larger than those of Strang-type block-circulant preconditioner S. But the operation cost per iteration of both $S^{(2)}$ and $\widetilde{S}^{(2)}$ is less than those of S. As we can see from Table 1–Table 2, the

Table 1
The number of iterations and CPU time (s) of GMRES(20) solver for Example 1 with $\alpha = 1.2$.

N	s	I		S		$S^{(2)}$		$\widetilde{S}^{(2)}$	
		Iters	CPU	Iters	CPU	Iters	CPU	Iters	CPU
24	16	82	0.0781	9	0.0310	15	0.0234	17	0.0263
	32	130	0.1714	8	0.0475	15	0.0348	17	0.0359
	64	265	0.5934	8	0.0935	15	0.0521	17	0.0588
	128	457	2.1064	7	0.2291	14	0.0989	17	0.1142
48	16	174	0.1562	9	0.0783	19	0.0308	20	0.0336
	32	198	0.3122	8	0.1249	18	0.0442	20	0.0485
	64	260	0.8279	8	0.2988	17	0.1148	20	0.1455
	128	460	2.5592	7	0.4213	17	0.1363	20	0.1948
96	16	234	0.2654	9	0.2811	23	0.0457	26	0.0532
	32	262	0.6425	8	0.4654	23	0.1172	26	0.1314
	64	339	1.3739	7	0.7948	21	0.1713	26	0.1901
	128	393	2.8865	7	1.4978	21	0.2927	26	0.3248

Table 2
The number of iterations and CPU time (s) of GMRES(20) solver for Example 1 with $\alpha = 1.5$.

N	s	I		S		$S^{(2)}$		$\widetilde{S}^{(2)}$	
		Iters	CPU	Iters	CPU	Iters	CPU	Iters	CPU
24	16	141	0.1208	10	0.0328	19	0.0271	27	0.0341
	32	201	0.2607	9	0.0558	19	0.0407	27	0.0522
	64	237	0.5416	8	0.0977	19	0.0668	28	0.0918
	128	378	1.8091	8	0.2038	18	0.1632	28	0.2056
48	16	259	0.2327	10	0.0782	25	0.0375	34	0.0466
	32	282	0.4308	9	0.1357	25	0.0602	36	0.0803
	64	313	0.9682	8	0.2703	25	0.1498	37	0.1902
	128	431	2.3788	8	0.4262	24	0.2326	37	0.3042
96	16	394	0.4289	10	0.2802	23	0.0574	47	0.0807
	32	428	0.9987	9	0.4838	23	0.1453	49	0.1898
	64	532	2.0718	8	0.8568	23	0.2386	51	0.3118
	128	632	4.6136	8	1.4922	32	0.3788	51	0.5459

CPU time of $S^{(2)}$ is less than those of the others especially when N and s are large. Moreover, the matrix J_N is ill-conditioned when N is large. The performance of $S^{(2)}$ is the best in terms of the CPU time. We strongly suggest that the preconditioner $S^{(2)}$ is a good choice and we do not need to formulate the complete matrix J_N in order to save storage. Especially, when J_N is the Toeplitz-like structure ($d_{\pm}(x,t) \neq const$). In order to further illustrate the effectiveness of the block-circulant preconditioners, we list the spectra of the original matrix M and the preconditioned matrices $S^{-1}M$, $(S^{(2)})^{-1}M$, $(\widetilde{S}^{(2)})^{-1}M$ in Fig. 1.

Example 2. In this example, we study the case for which the source term $f(x,t) \neq 0$. We solve the FDE problem (1) of order $\alpha = 1.5$ and $\alpha = 1.8$, respectively. The spatial domain is $[x_L, x_R] = [0, 2]$ and the time interval is $[t_0, T] = [0, 1]$. The left and right diffusion coefficients are

$$d_+(x,t) = \Gamma(3-\alpha)x^{\alpha} \quad \text{and} \quad d_-(x,t) = \Gamma(3-\alpha)(2-x)^{\alpha},$$

respectively. The source term is

$$f(x,t) = -32e^{-t}\left\{x^2 + \frac{1}{8}(2-x)^2(8+x^2) - \frac{3}{3-\alpha}[x^3 + (2-x)^3] + \frac{3}{(4-\alpha)(3-\alpha)}[x^4 + (2-x)^4]\right\},$$

and the initial condition is

$$u(x,0) = 4x^2(2-x)^2, \quad \text{and} \quad u(0,t) = u(1,t) = 0.$$

The exact solution of this problem is

$$u(x,t) = 4e^{-t}x^2(2-x)^2,$$

for any $\alpha \in (1,2)$. With the exact solution, we calculate the exact error of the numerical solution under the infinity norm.

Tables 3 and 4 list the number of iterations required for convergence of the GMRES method with different preconditioners and their corresponding CPU time for numerically solving Example 2, also including the corresponding numerical error. In the tables, I means that no preconditioner is used, and $S^{(2)}$ and $\widetilde{S}^{(2)}$ denote the Strang-type and modified Strang-type BCCB preconditioners respectively, see (13) and (17). Here, the symbol "†" means that the unpreconditioned GMRES method needs the many iterations to converge for desired error tolerance. We do not give the specific number of iterations. From the tables, we note that the number of iterations and CPU time consuming for convergence with the Strang-type

Table 3
The number of iterations and CPU time (s) of GMRES(20) solver for Example 2 with $\alpha = 1.5$.

N	s	Error	I		$S^{(2)}$		$\widetilde{S}^{(2)}$	
			Iters	CPU	Iters	CPU	Iters	CPU
32	32	4.3401e−2	165	0.2529	28	0.0564	39	0.0764
	64	4.0506e−2	200	0.4996	28	0.0959	40	0.1412
	96	3.9623e−2	264	1.0185	29	0.1576	40	0.2348
	128	3.9178e−2	272	1.5268	29	0.2328	40	0.2986
64	32	2.4662e−2	367	0.6218	37	0.0989	56	0.1245
	64	2.1559e−2	401	1.3488	37	0.1984	57	0.2811
	96	2.0593e−2	460	2.4864	37	0.2526	57	0.3904
	128	2.0151e−2	479	3.0765	37	0.3542	57	0.4991
128	32	1.6068e−2	809	2.0743	51	0.1881	73	0.2868
	64	1.2287e−2	815	3.8276	51	0.3925	76	0.4846
	96	1.1230e−2	862	5.8157	51	0.5188	76	0.6874
	128	1.0737e−2	872	7.4864	52	0.6459	77	0.8921
256	32	1.3177e−2	2365	9.0548	65	0.4119	95	0.5356
	64	8.0302e−3	2147	14.8762	67	0.6845	98	0.9321
	96	6.7131e−3	1934	19.6118	67	0.9813	100	1.3878

Table 4
The number of iterations and CPU time (s) of GMRES(20) solver for Example 2 with $\alpha = 1.8$.

N	s	Error	I		$S^{(2)}$		$\widetilde{S}^{(2)}$	
			Iters	CPU	Iters	CPU	Iters	CPU
32	32	3.7348e−2	381	0.5025	38	0.0822	91	0.1567
	64	3.1716e−2	416	0.9716	37	0.1227	95	0.2967
	96	2.9804e−2	480	1.7664	38	0.1972	96	0.4914
	128	2.8864e−2	499	2.3748	38	0.2436	97	0.6187
64	32	2.3108e−2	875	1.3624	54	0.1218	125	0.2521
	64	1.7520e−2	1027	3.4325	54	0.2418	131	0.5632
	96	1.5622e−2	1133	5.3047	54	0.3449	135	0.7984
	128	1.4666e−2	1154	6.9526	55	0.4834	136	1.0614
128	32	1.6867e−2	3438	8.1957	75	0.2498	167	0.5462
	64	1.1300e−2	3300	14.1056	72	0.4847	180	1.0761
	96	9.4078e−3	3203	19.7183	73	0.6708	186	1.5910
	128	8.4549e−3	3021	24.2746	74	0.8265	188	2.0574
256	32	1.3982e−2	†	35.2787	95	0.5292	224	1.1217
	64	8.4238e−3	†	68.0410	98	0.9326	245	2.1688
	96	6.5356e−3	†	101.1632	102	1.4383	251	3.4024

and modified Strang-type BCCB preconditioners are both much less than those with no preconditioner. Moreover, the accelerated performance of $S^{(2)}$ outperforms that of $\widetilde{S}^{(2)}$ considerably in aspects of both iteration steps and CPU time consuming. Meanwhile, for fixed spatial grid size N, the numerical accuracy ("Error") can be improved by changing the time-step size s. It suggests that our method is more flexible than classical implicity difference scheme for numerically solving the FDEs.

Next, let us take a simple comparison in terms of the proposed method and the method in [36] (denoted as Lei–Sun's method) for solving Example 2 numerically. As seen from Table 5,[1] the numerical accuracy by proposed method is slightly lower than that by Lei–Sun's method. It can be explained by saying that the spatial discretized accuracy of both method are 1-order. We try to improve the accuracy of numerical solution via the high order time-step discretized scheme and it is not very successful. Our proposed method can be recommended as an alternative since they take less CPU time to converge in the case of small discretized size.

However, except the cases of $N = 128$, our proposed method converges slightly faster than the method in [36] in terms of the CPU time. Here it should point out that when the spatial discretized size N becomes increasingly large, the proposed method will still requires much CPU time than Lei–Sun's method. In future work, the high-order spatial discretized scheme [29] should be exploited to improve the global numerical accuracy of the proposed method with comparing to the traditional time-step scheme.

The behavior of the numerical and exact solutions of this problem with different values of $\alpha = 1.5$ and $\alpha = 1.8$ are given in Fig. 2. Where in Fig. 2, the numerical results at discretized size $N = 256$, $s = 96$ for different values of $\alpha = 1.5$ and $\alpha = 1.8$ and the corresponding exact solutions are plotted. From these figures, we can conclude that the numerical results obtained by using the proposed method are in excellent agreement with the exact solution and the numerical solutions are reliable. Meanwhile, our proposed method can be provided for a different and interesting numerical aspect for solving the FDEs.

[1] Here the "Iters_2" denotes the average number of iterations required via the Lei-Sun's method and it is different from the definition of "Iters" in present paper. So it cannot conclude that the recent method [36] outperforms the proposed method considerably in aspects of only iteration steps.

Table 5
Comparison results of the proposed method and Lei–Sun's method for Example 2 with $\alpha = 1.5$.

$N+1$	s	Proposed method			Lei–Sun's method		
		Error	Iters	CPU	Error	Iters_2	CPU
32	32	4.4521e−2	27	6.5674e−2	4.2309e−2	11	1.5937e−1
	64	4.1793e−2	28	1.2101e−1	4.0654e−2	10	1.7560e−1
	96	4.0870e−2	28	1.9217e−1	4.0101e−2	9	2.3569e−1
	128	4.0405e−2	28	2.5233e−1	3.9825e−2	9	3.0965e−1
64	32	2.4937e−2	37	9.5482e−2	2.2529e−2	13	1.8001e−1
	64	2.1854e−2	37	1.8389e−1	2.0763e−2	12	2.1845e−1
	96	2.0907e−2	37	2.7206e−1	2.0202e−2	11	2.9512e−1
	128	2.0454e−2	37	3.6357e−1	1.9922e−2	11	3.9658e−1
128	32	1.6128e−2	50	2.6879e−1	1.3186e−2	16	2.5619e−1
	64	1.2358e−2	51	4.6902e−1	1.1164e−2	14	3.1798e−1
	96	1.1301e−2	51	6.5108e−1	1.0566e−2	13	4.6739e−1
	128	1.0811e−2	52	8.5112e−1	1.0282e−2	13	5.8103e−1

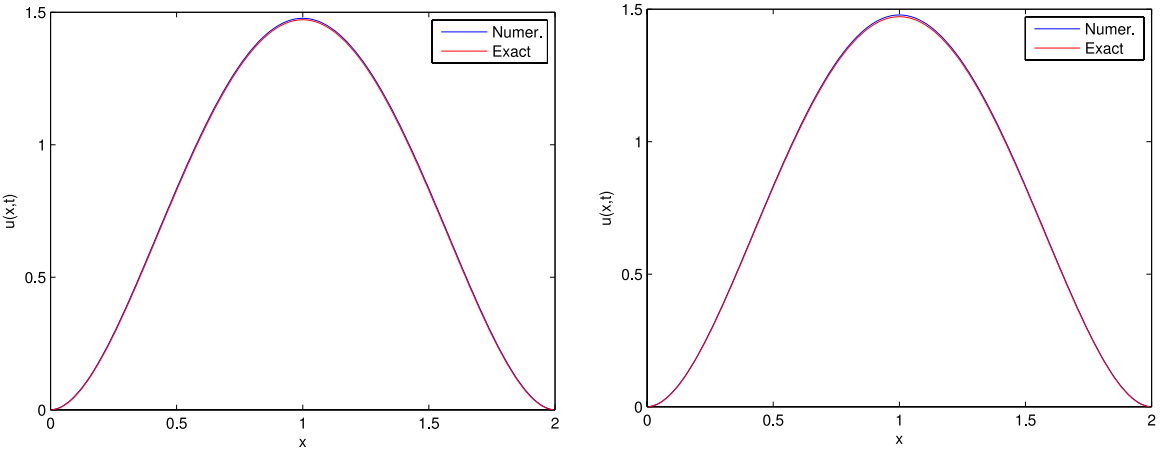

Fig. 2. Comparison of the numerical and exact solutions using the discretized size $N = 256$, $s = 96$ with different α for Example 2; Left: $\alpha = 1.5$; Right: $\alpha = 1.8$.

Acknowledgments

The authors would like to thank Prof. Hai-Wei Sun, Dr. Siu-Long Lei and Prof. Fu-Rong Lin for their helpful discussions. We are also grateful to the anonymous referees and editor Prof. Michael Ng for their useful suggestions and comments that improved the presentation of this paper. This research is supported by 973 Program (2013CB329404), NSFC (61370147, 61170311, 11101071, 61402082, 11301057, and 61170309), Chinese Universities Specialized Research Fund for the Doctoral Program (20110185110020), the Fundamental Research Funds for the Central Universities (ZYGX2013J106).

References

[1] G.M. Zaslavsky, D. Stevens, H. Weitzner, Self-similar transport in incomplete chaos, Phys. Rev. E 48 (1993) 1683–1694.
[2] B.A. Carreras, V.E. Lynch, G.M. Zaslavsky, Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence models, Phys. Plasma 8 (2001) 5096–5103.
[3] M.F. Shlesinger, B.J. West, J. Klafter, Lévy dynamics of enhanced diffusion: application to turbulence, Phys. Rev. Lett. 58 (1987) 1100–1103.
[4] D.A. Benson, S.W. Wheatcraft, M.M. Meerschaert, Application of a fractional advection–dispersion equation, Water Resour. Res. 36 (2000) 1403–1413.
[5] D.A. Benson, S.W. Wheatcraft, M.M. Meerschaert, The fractional-order governing equation of Lévy motion, Water Resour. Res. 36 (2000) 1413–1423.
[6] R.L. Magin, Fractional Calculus in Bioengineering, Begell House Publishers, Connecticut, USA, 2006.
[7] E. Orsingher, L. Beghin, Fractional diffusion equations and processes with randomly varying time, Ann. Probab. 37 (2009) 206–249.
[8] M. Raberto, E. Scalas, F. Mainardi, Waiting-times and returns in high-frequency financial data: an empirical study, Physica A 314 (2002) 749–755.
[9] J. Bai, X.-C. Feng, Fractional-order anisotropic diffusion for image denoising, IEEE Trans. Image Process. 16 (2007) 2492–2502.
[10] J.M. Blackledge, Diffusion and fractional diffusion based image processing, in: Wen Tang, John Collomosse (Eds.), EG UK Theory and Practice of Computer Graphics, Cardiff, 2009, pp. 233–240.
[11] B. Baeumer, D.A. Benson, M.M. Meerschaert, S.W. Wheatcraft, Subordinated advection–dispersion equation for contaminant transport, Water Resour. Res. 37 (2001) 1543–1550.
[12] I.M. Sokolov, J. Klafter, A. Blumen, Fractional kinetics, Phys. Today 55 (2002) 48–54.
[13] T.A.M. Langlands, B.I. Henry, The accuracy and stability of an implicit solution method for the fractional diffusion equation, J. Comput. Phys. 205 (2005) 719–736.
[14] F. Liu, V.V. Anh, I.W. Turner, Numerical solution of the space fractional Fokker–Planck equation, J. Comput. Appl. Math. 166 (2004) 209–219.
[15] M.M. Meerschaert, H.P. Scheffler, C. Tadjeran, Finite difference methods for two-dimensional fractional dispersion equation, J. Comput. Phys. 211 (2006) 249–261.

[16] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for fractional advection–dispersion flow equations, J. Comput. Appl. Math. 172 (2004) 65–77.
[17] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math. 56 (2006) 80–90.
[18] D.A. Murio, Implicit finite difference approximation for time fractional diffusion equations, Comput. Math. Appl. 56 (2008) 1138–1145.
[19] L.-J. Su, W.-Q. Wang, Z.-X. Yang, Finite difference approximations for the fractional advection–diffusion equation, Phys. Lett. A 373 (2009) 4405–4408.
[20] W.-H. Deng, Finite element method for the space and time fractional Fokker–Planck equation, SIAM J. Numer. Anal. 47 (2008) 204–226.
[21] V.J. Ervin, N. Heuer, J.P. Roop, Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation, SIAM J. Numer. Anal. 45 (2007) 572–591.
[22] K. Burrage, N. Hale, D. Kay, An efficient implicit FEM scheme for fractional-in-space reaction–diffusion equations, SIAM J. Sci. Comput. 34 (2012) 2145–2172.
[23] J.P. Roop, Computational aspects of FEM approximation of fractional advection dispersion equations on bounded domains in $\mathbb{R}^2$, J. Comput. Appl. Math. 193 (2006) 243–268.
[24] X. Ji, H.-Z. Tang, High-order accurate Runge–Kutta (local) discontinuous Galerkin methods for one-and two-dimensional fractional diffusion equations, Numer. Math. Theory Methods Appl. 5 (2012) 333–358.
[25] K. Mustapha, W. McLean, Piecewise-linear, discontinuous Galerkin method for a fractional diffusion equation, Numer. Algorithms 56 (2011) 159–184.
[26] C. Piret, E. Hanert, A radial basis functions method for fractional diffusion equations, J. Comput. Phys. 238 (2013) 71–81.
[27] K.-X. Wang, H. Wang, A fast characteristic finite difference method for fractional advection–diffusion equations, Adv. Water Resour. 34 (2011) 810–816.
[28] C.-M. Chen, F. Liu, I. Turner, V. Anh, A Fourier method for the fractional diffusion equation describing sub-diffusion, J. Comput. Phys. 227 (2007) 886–897.
[29] H. Zhou, W.-Y. Tian, W.-H. Deng, Quasi-compact finite difference schemes for space fractional diffusion equations, J. Sci. Comput. 56 (2013) 45–66.
[30] A. Mohebbi, M. Abbaszade, M. Dehghan, A high-order and unconditionally stable scheme for the modified anomalous fractional sub-diffusion equation with a nonlinear source term, J. Comput. Phys. 240 (2013) 36–48.
[31] K. Mustapha, An implicit finite difference time-stepping method for a sub-diffusion equation, with spatial discretization by finite elements, IMA J. Numer. Anal. 31 (2011) 719–739.
[32] H. Wang, K.-X. Wang, T. Sircar, A direct $\mathcal{O}(N \log^2 N)$ finite difference method for fractional diffusion equations, J. Comput. Phys. 229 (2010) 8095–8104.
[33] R. Chan, X.-Q. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, USA, 2007.
[34] R. Chan, M. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev. 38 (1996) 427–482.
[35] H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. 231 (2012) 693–703.
[36] S.-L. Lei, H.-W. Sun, A circulant preconditioner for fractional diffusion equations, J. Comput. Phys. 242 (2013) 715–725.
[37] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[38] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, Netherlands, 1998.
[39] P. Amodio, F. Mazzia, D. Trigiante, Stability of some boundary value methods for the solution of initial value problems, BIT 33 (1993) 434–451.
[40] L. Brugnano, D. Trigiante, Boundary value methods: the third way between linear multistep and Runge–Kutta methods, Comput. Math. Appl. 36 (1998) 269–284.
[41] S.-L. Lei, X.-Q. Jin, BCCB preconditioners for systems of BVM-based numerical integrators, Numer. Linear Algebra Appl. 11 (2004) 25–40.
[42] I. Podlubny, Fractional Differential Equations, Academic Press, New York, USA, 1999.
[43] R. Chan, G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner, SIAM J. Sci. Stat. Comput. 10 (1989) 104–119.
[44] A.O.H. Axelsson, J.G. Verwer, Boundary value techniques for initial value problems in ordinary differential equations, Math. Comp. 45 (1985) 153–171.
[45] M. Benzi, G.H. Golub, A preconditioner for generalized saddle point problems, SIAM J. Matrix Anal. Appl. 26 (2004) 20–41.
[46] R.H. Chan, X.-Q. Jin, Y.-H. Tam, Strang-type preconditioners for solving system of ODEs by boundary value methods, Electron. J. Math. Phys. Sci. 1 (2002) 14–46.
[47] P. Davis, Circulant Matrices, Wiley, New York, USA, 1979.
[48] R. Chan, M. Ng, X.-Q. Jin, Strang-type preconditioner for systems of LMF-based ODE codes, IMA J. Numer. Anal. 21 (2001) 451–462.