

RESEARCH

Open Access



Fast permutation preconditioning for fractional diffusion equations

Sheng-Feng Wang^{1*}, Ting-Zhu Huang¹, Xian-Ming Gu¹ and Wei-Hua Luo²

*Correspondence:

yihe1001@163.com

¹ School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People's Republic of China
Full list of author information is available at the end of the article

Abstract

In this paper, an implicit finite difference scheme with the shifted Grünwald formula, which is unconditionally stable, is used to discretize the fractional diffusion equations with constant diffusion coefficients. The coefficient matrix possesses the Toeplitz structure and the fast Toeplitz matrix-vector product can be utilized to reduce the computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$, where N is the number of grid points. Two preconditioned iterative methods, named bi-conjugate gradient method for Toeplitz matrix and bi-conjugate residual method for Toeplitz matrix, are proposed to solve the relevant discretized systems. Finally, numerical experiments are reported to show the effectiveness of our preconditioners.

Keywords: Fractional diffusion equations, Shifted Grünwald formula, Toeplitz matrix, BiCGT method, BiCRT method, Fast Fourier transforms

Mathematics Subject Classification: 65F10, 65L12, 65L20, 65T50, 26A33

Background

In the last few decades, many anomalous diffusion phenomena have been found in the real world, which lead to the generation of the fractional diffusion equations (FDEs). The FDEs emerge from numerous research fields such as modeling chaotic dynamics of classical conservative systems (Zaslavsky et al. 1993), groundwater contaminant transport (Benson et al. 2000a, b), turbulent flow (Carreras et al. 2001; Shlesinger et al. 1987), and applications in physics (Sokolov et al. 2002), finance (Raberto et al. 2002), biology (Magin 2006), hydrology (Baeumer et al. 2001) and image processing (Blackledge 2009; Bai and Feng 2007). Usually, it is unavailable to access the closed-form analytical solutions of the FDEs (Alquran et al. 2015; Allan and Al-Khaled 2006; Sababheh et al. 2003). Therefore, many numerical approaches for the FDEs have been proposed and developed intensively in the last decade, for instance Zhang et al. (2010), Ervin et al. (2007), Langlands and Henry (2005), Liu et al. (2004), Meerschaert and Tadjeran (2004, 2006), Tian et al. (2015), Gu et al. (2015). However, even if the discretized approach of the FDEs is implicit, it still can result in unconditionally unstable (Meerschaert and Tadjeran 2004, 2006) because of the nonlocality of the fractional differential operators.

In order to overcome the difficulty of the stability, (Meerschaert and Tadjeran 2004, 2006) put forward a shifted Grünwald discretization to approximate FDEs with a left-sided

fractional derivative and the FDEs with two-sided fractional derivatives, respectively, and their method has been proved to be unconditionally stable. However, it is worth noting that most of the numerical methods for FDEs tend to generate full coefficient matrices, which require computational cost of $\mathcal{O}(N^3)$ and storage of $\mathcal{O}(N^2)$, there is no doubt that it will certainly increase the computational work; see, e.g., Wang et al. (2010) for a discussion of these issues.

Recently, there is some progress on fast numerical solutions of FDEs. Wang et al. (2010) discovered that the full coefficient matrix generated by Meerschaert and Tadjeran (2006) method has a good feature, i.e., it can be written as a sum of diagonal-multiply-Toeplitz matrices. Thus the storage requirement is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. As we know, the Toeplitz matrix-vector product (MVP) can be computed in $\mathcal{O}(N \log N)$ operations by the fast Fourier transforms (FFTs) (Chan and Ng 1996). Fast methods Lei and Sun (2013), Popolizio (2015), Gu et al. (2015, 2015) have been developed to solve FDEs with the shifted Grünwald formula. Wang and Wang (2011) proposed a conjugate gradient normal residual (CGNR) to solve the discretized system by Meerschaert and Tadjeran's method with the computational cost of $\mathcal{O}(N \log N)$. The preconditioned CGNR with a circulant preconditioner is proposed by Lei and Sun (2013), to solve FDEs by Meerschaert and Tadjeran's method with constant diffusion coefficients.

In this paper, we employ the implicit finite difference method to discretize the FDEs and the problem is transformed to solve a linear nonsymmetric Toeplitz system in each time step. Since, the Bi-Conjugate Gradient (BiCG) (Saad 2003, pp. 234–236) and Bi-Conjugate Residual (BiCR) (Sogabe et al. 2009) can be regarded as two effective methods for solving the nonsymmetric system. There is no doubt that the two methods with Toeplitz fast MVP can be used to solve such discretized Toeplitz linear systems. However, from Sogabe et al. (2005), Pestana and Wathen (2015), if the two iterative methods are employed directly, then we indeed fail to make full use of Toeplitz structure of the discretized system, it also means that their computational cost fail to attain optimality. Hence, it is still worth finding more effective methods to reduce the computational complexity. Recently, in Sogabe et al. (2005), Pestana and Wathen (2015), a permutation matrix P was introduced to transform the nonsymmetric matrix into a symmetric one so as to improve the performance of iterative methods. In view of this point, we re-explain the ideas in Sogabe et al. (2005), Pestana and Wathen (2015) as a kind of preconditioning techniques for solving the discretized system of the FDEs by the method of Meerschaert and Tadjeran. More precisely, we do equivalent transformation for the original discretized system, left multiplying by a permutation matrix (Pestana and Wathen 2015) at the same time, then we obtain a new symmetric linear system with the coefficient matrix being a Hankle matrix, which has the same solution with the original discretized system. As we know, the symmetric linear systems are usually simpler to be solved than the nonsymmetric cases. Conjugate Gradient (CG) and Conjugate Residual (CR) are two effective methods for solving symmetric linear system. In this paper, we extend CG and CR to BiCGT and BiCRT, respectively, which are proposed to solve the equivalent equation. The numerical results show that both BiCGT and BiCRT are more competitive than CGNR.

The paper is organized as follows. In Sect. 2, we briefly introduce the discretization of FDEs by finite difference method. In Sect. 3, we construct the permutation preconditioner and propose BiCGT and BiCRT to solve the equivalent system of linear equations. In Sect. 4, numerical results are reported to illustrate the efficiency of the proposed methods. Concluding remarks are given in Sect. 5.

Discretization of FDEs by finite difference method

In this section, we are interested in solving an initial-boundary value problem of the following FDEs,

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = d_1 \frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} + d_2 \frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} + f(x,t), & x \in (x_L, x_R), t \in (0, T], \\ u(x_L, t) = u(x_R, t) = 0, & 0 \leq t \leq T, \\ u(x, 0) = u_0(x), & x \in [x_L, x_R], \end{cases} \tag{1}$$

where $\alpha \in (1, 2)$ is the order of the space fractional derivative, $f(x, t)$ is the source term, and the diffusion coefficients satisfying $d_1 \geq 0, d_2 \geq 0$, and $d_1 + d_2 \neq 0$. In this paper, we use the Grünwald-Letnikov form Podlubny (1999) to define the left-sided and the right-sided fractional derivatives $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ and $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$:

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

where $\lfloor \cdot \rfloor$ denotes the floor function, and the Grünwald coefficients $g_k^{(\alpha)}$ are defined as follows

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

which can be evaluated by the recurrence relation

$$g_{k+1}^{(\alpha)} = \left(1 - \frac{\alpha + 1}{k + 1} \right) g_k^{(\alpha)}, \quad k = 0, 1, 2, \dots$$

In order to derive the proposed scheme, let $h = \frac{x_R - x_L}{N+1}$ and $\Delta t = T/M$ be the sizes of spatial grid and time step, respectively (N, M are positive integers). Define $x_i = x_L + ih$ ($i = 0, 1, \dots, N + 1$) and the temporal partition $t_m = m\Delta t$ ($m = 0, 1, \dots, M$). Let $u_i^{(m)} = u(x_i, t_m), f_i^{(m)} = f(x_i, t_m)$. We employ the shifted Grünwald approximation (Meerschaert and Tadjeran 2004, 2006):

$$\begin{aligned} \frac{\partial^\alpha u(x_i, t_m)}{\partial_+ x^\alpha} &= \frac{1}{h^\alpha} \sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^{(m)} + \mathcal{O}(h), \\ \frac{\partial^\alpha u(x_i, t_m)}{\partial_- x^\alpha} &= \frac{1}{h^\alpha} \sum_{k=0}^{N-i+2} g_k^{(\alpha)} u_{i+k-1}^{(m)} + \mathcal{O}(h), \end{aligned}$$

where $g_k^{(\alpha)}$ is defined in Eq. (2). Then the corresponding finite difference scheme

$$\frac{u_i^{(m)} - u_i^{(m-1)}}{\Delta t} = \frac{d_1}{h^\alpha} \sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^{(m)} + \frac{d_2}{h^\alpha} \sum_{k=0}^{N-i+2} g_k^{(\alpha)} u_{i+k-1}^{(m)} + f_i^{(m)} \tag{3}$$

is unconditionally stable, see Meerschaert and Tadjeran (2004, 2006) for details.

Let $u^{(m)} = [u_1^{(m)}, u_2^{(m)}, \dots, u_N^{(m)}]^T \in \mathbb{R}^N, f^{(m)} = [f_1^{(m)}, f_2^{(m)}, \dots, f_N^{(m)}]^T \in \mathbb{R}^N$. Then we can rewrite (3) into the matrix form

$$\left(\frac{h^\alpha}{\Delta t}I - A^{(m)}\right)u^{(m)} = \frac{h^\alpha}{\Delta t}u^{(m-1)} + h^\alpha f^{(m)}, \tag{4}$$

with

$$A^{(m)} = d_1G_\alpha + d_2G_\alpha^T,$$

where

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

We can note that G_α is a nonsymmetric Toeplitz matrix, thus it can be stored with $N + 1$ entries (Wang et al. 2010). The Toeplitz matrix-vector product (MVP) can be computed in $\mathcal{O}(N \log N)$ complexities with the aid of FFTs (Pang and Sun 2012).

Define $v_{N,M} = \frac{h^\alpha}{\Delta t} = (x_R - x_L)^\alpha T^{-1} \frac{M}{(N+1)^\alpha}$, which is related to the number of time steps and grid points. The above linear system (4) can be rewritten in the following matrix form

$$M^{(m)}u^{(m)} = b^{(m-1)}, \tag{5}$$

where

$$M^{(m)} = \frac{h^\alpha}{\Delta t}I - A^{(m)} = v_{N,M}I - (d_1G_\alpha + d_2G_\alpha^T),$$

and the right hand vector

$$b^{(m-1)} = v_{N,M}(u^{(m-1)} + \Delta t f^{(m)}).$$

In order to illustrate the convergence and stability of the implicit difference scheme (3), we note that $g_k^{(\alpha)}$ satisfy the following proposition.

Proposition 1 (Meerschaert and Tadjeran 2004; 2006; Wang et al. 2010) *Let $1 < \alpha < 2$ and $g_k^{(\alpha)}$ be defined in (2). Then we have*

$$\begin{cases} g_0^{(\alpha)} = 1, g_1^{(\alpha)} = -\alpha < 0, g_2^{(\alpha)} > g_3^{(\alpha)} > \cdots > 0, \\ \sum_{j=0}^\infty g_j^{(\alpha)} = 0, \sum_{j=0}^n g_j^{(\alpha)} < 0, \quad \text{for } n \geq 1, \\ g_j^{(\alpha)} = \mathcal{O}(j^{-(\alpha+1)}). \end{cases}$$

Since $|g_1^{(\alpha)}| > \sum_{j=0, j \neq 1}^n g_k^{(\alpha)}$, $M^{(m)} = v_{N,M}I - A^{(m)}$ is a strongly diagonally dominant and nonsingular Toeplitz matrix, and thus the scheme (3) is monotone; refer to Wang et al. (2010).

The BiCGT method and the BiCRT method

We will show how to construct the permutation preconditioners for accelerating the iterative solver and describe the derivation process of BiCGT and BiCRT. Furthermore, an analysis of computational cost for each iteration step is also proposed. In the linear system (5), $M^{(m)}$ is a nonsymmetric Toeplitz matrix. As previously mentioned, we cannot exploit BiCG and BiCR for resulting linear systems (5) directly, otherwise it is not possible to take advantage of the Toeplitz structure of coefficient matrix. So we need to modify and improve the BiCG and BiCR methods particularly for solving (5). Recently, Sogabe et al. (2005), proposed a preconditioner of permutation matrix for improving the performance of the Krylov subspace method, which is used to solve a nonsymmetric Toeplitz linear system. Later, Pestana and Wathen rigorously establish a circulant preconditioned MINRES method (Paige and Saunders 1975) for nonsymmetric Toeplitz systems. Inspired by their pioneer work, we construct a preconditioner, which is a permutation matrix P (Sogabe et al. 2005) with the form of

$$P = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{N \times N}$$

We would like to solve the Toeplitz system (5) by the CG-like method. This goal can be achieved with little additional computing cost, since we can get the equivalent system:

$$(PM^{(m)})u^{(m)} = Pb^{(m-1)}, \tag{6}$$

which can be regarded as a left preconditioning technique (Saad 2003) and also has the same solution with (5). Define $\tilde{M}^{(m)} = PM^{(m)}$, $\tilde{b}^{(m-1)} = Pb^{(m-1)}$, then (6) can be rewritten into

$$\tilde{M}^{(m)}u^{(m)} = \tilde{b}^{(m-1)}, \tag{7}$$

an equivalent statement is that $M^{(m)}$ is self-adjoint with respect to the bilinear form defined by P (Paige and Saunders 1975). P is symmetric positive definite, a nonsymmetric Toeplitz matrix is exactly changed into a symmetric matrix $\tilde{M}^{(m)}$, so that (7) can be solved by the modified BiCG, where the additional operations for dual systems have been eliminated.

Firstly, we consider CG and BiCG (Saad 2003). Then we followed the philosophy behind the derivation of iterative method in Sogabe et al. (2005 Algorithm1), i.e., in the CG Algorithm, we replace A and b with $\tilde{A} = PM$ and $\tilde{f} = Pf$, respectively. Then we get the following new algorithm:

Algorithm 1 BiCGT for $\tilde{M}^{(m)}u^{(m)} = \tilde{b}^{(m-1)}$ with preconditioner P

1. u_0 is an initial guess, $r_0 = P(b - M^{(m)}u_0)$, $\beta_{-1} = 0$.
 2. For $n = 0, 1, \dots$, until $\|r_n\| \leq \epsilon \|b\|$, Do
 3. $p_n = r_n + \beta_{n-1}p_{n-1}$
 4. $q_n = M^{(m)}p_n$
 5. $\alpha_n = (r_n, r_n)/(p_n, Pq_n)$
 6. $u_{n+1} = u_n + \alpha_n p_n$
 7. $r_{n+1} = r_n - \alpha_n Pq_n$
 8. $\beta_n = (r_{n+1}, r_{n+1})/(r_n, r_n)$
 9. End do.
-

In BiCGT, we only need one MVP, i.e. $M^{(m)}p_n$, and two inner products, three vector additions/subtractions per iteration. The rewritten algorithm is more effective than CGNR, because P multiply an arbitrary vector is to reorder the vector in its reversed order (Sogabe et al. 2005). Therefore, it can greatly reduce the required number of MVPs.

Algorithm 2 BiCRT $\tilde{M}^{(m)}u^{(m)} = \tilde{b}^{(m-1)}$ with preconditioner P

1. u_0 is an initial guess, $r_0 = P(b - M^{(m)}u_0)$, $\beta_{-1} = 0$, $p_0 = 0$, $ap_0 = 0$.
 2. For $n = 0, 1, \dots$, until $\|r_n\| \leq \epsilon \|b\|$, Do
 3. $p_n = r_n + \beta_{n-1}p_{n-1}$
 4. $q_n = M^{(m)}r_n$
 5. $ap_n = Pq_n + \beta_{n-1}ap_{n-1}$
 6. $\alpha_n = (r_n, Pq_n)/(ap_n, ap_n)$
 7. $u_{n+1} = u_n + \alpha_n p_n$
 8. $r_{n+1} = r_n - \alpha_n ap_n$
 9. $\beta_{n+1} = (r_{n+1}, Pq_{n+1})/(r_n, Pq_n)$
 10. End do.
-

In a similar way, we can get the algorithm of BiCRT for symmetric linear system (7), BiCRT reduces to CR if we get rid of the permutation preconditioner P , and the algorithm of BiCRT is presented as blew.

If we employ BiCG or BiCR directly, it is impossible to minimize the residual vector in some special conditions. However, BiCRT could realize this goal to some extent. The approximation u_{n+1} is generated from u_n by moving from u_n in a certain direction p_n to a minimum point of the residual function $E(u) = \|\tilde{M}^{(m)}u - \tilde{b}^{(m-1)}\|_2$, $u \in \mathbb{R}^N$. In other words, for $u_{n+1} = u_n + \alpha_n p_n$, α_n is chosen to minimize $E(u)$.

It is useful to consider the computational cost. We give a table to illustrate the computational cost of BiCG, BiCR, BiCGT, BiCRT and CGNR. ‘‘AXPY’’ denotes addition of scaled vectors, and ‘‘1 + 1’’ denotes 1 product with the matrix and 1 with its transpose. From Table 1, it is remarkable that BiCG and BiCR, BiCGT and BiCRT require almost the same memory and computational cost in each iteration step. More precisely, BiCGT is the best method to solve the above system in terms of computational cost (i.e. AXPYs and MVPs). For BiCRT, the number of AXPYs is one more than BiCGT, and for CGNR, the number of MVPs is one more than BiCGT. As we know, the computational complexity of one Toeplitz MVP is $\mathcal{O}(N \log N)$ by FFTs, but one AXPY can be computed in $\mathcal{O}(N)$ complexity. So BiCRT is more efficient than CGNR from this perspective.

Table 1 Summary of algorithmic cost per iteration step

Method	Dot product	AXPY	MVP
BiCG	2	5	1+1
BiCR	2	6	1+1
BiCGT	2	3	1
BiCRT	2	4	1
CGNR	2	3	2

Numerical results

We solve the FDEs (1) numerically by the implicit finite difference method given in Sect. 2. After the finite difference discretization and the equivalent transformation, the symmetric linear system (7) is solved by BiCGT (Algorithm 1), BiCRT (Algorithm 2), and CGNR (Wang and Wang 2011), respectively. All the MVPs $\tilde{M}^{(m)}u^{(m)}$ are done by FFTs in $\mathcal{O}(N \log N)$ operations (Lei and Sun 2013) and the initial guess is chosen to be zero vector at each time step. The stopping criterion of BiCGT, BiCRT and CGNR is set to be

$$\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} < 10^{-7},$$

where $r^{(k)}$ is the residual vector of the linear system after k iterations and $r^{(0)}$ is the initial residual vector.

In the following tables, “ N ” denotes the number of spatial grid points, “ M ” denotes the number of time steps, CPU(s) denotes the total CPU time (in seconds) for solving the whole discretized system. “Error” denotes the infinity norm of the difference between the true solution and the approximation at the last time step. “Iter” denotes the average number of iterations over all time discretized level for solving the FDEs, i.e., $\text{Iter} = \frac{1}{M} \sum_{m=1}^M \text{Iter}(m)$, where $\text{Iter}(m)$ is the number of iterations required for solving the linear system (7) in the m th time discretized level. All experiments are run in MATLAB R2010a on a PC with the following configuration: Windows 7 (32 bit), Iter(R) Core(TM) i3-2130 CPU 3.40 GHz and 4 GB RAM.

Example 1 We consider FDEs (1) on space interval $[x_L, x_R] = [0, 1]$ and time interval $[0, T] = [0, 1]$ with diffusion coefficients $d_1 = 0.8, d_2 = 0.2$, initial condition $u_0(x) = \sin(1)x^3(1-x)^3$, and source term

$$f(x, t) = \cos(t+1)x^3(1-x)^3 - \sin(t+1) \left\{ \frac{\Gamma(4)}{\Gamma(4-\alpha)} [d_1x^{3-\alpha} + d_2(1-x)^{3-\alpha}] - 3 \frac{\Gamma(5)}{\Gamma(5-\alpha)} [d_1x^{4-\alpha} + d_2(1-x)^{4-\alpha}] + 3 \frac{\Gamma(6)}{\Gamma(6-\alpha)} [d_1x^{5-\alpha} + d_2(1-x)^{5-\alpha}] - \frac{\Gamma(7)}{\Gamma(7-\alpha)} [d_1x^{6-\alpha} + d_2(1-x)^{6-\alpha}] \right\}.$$

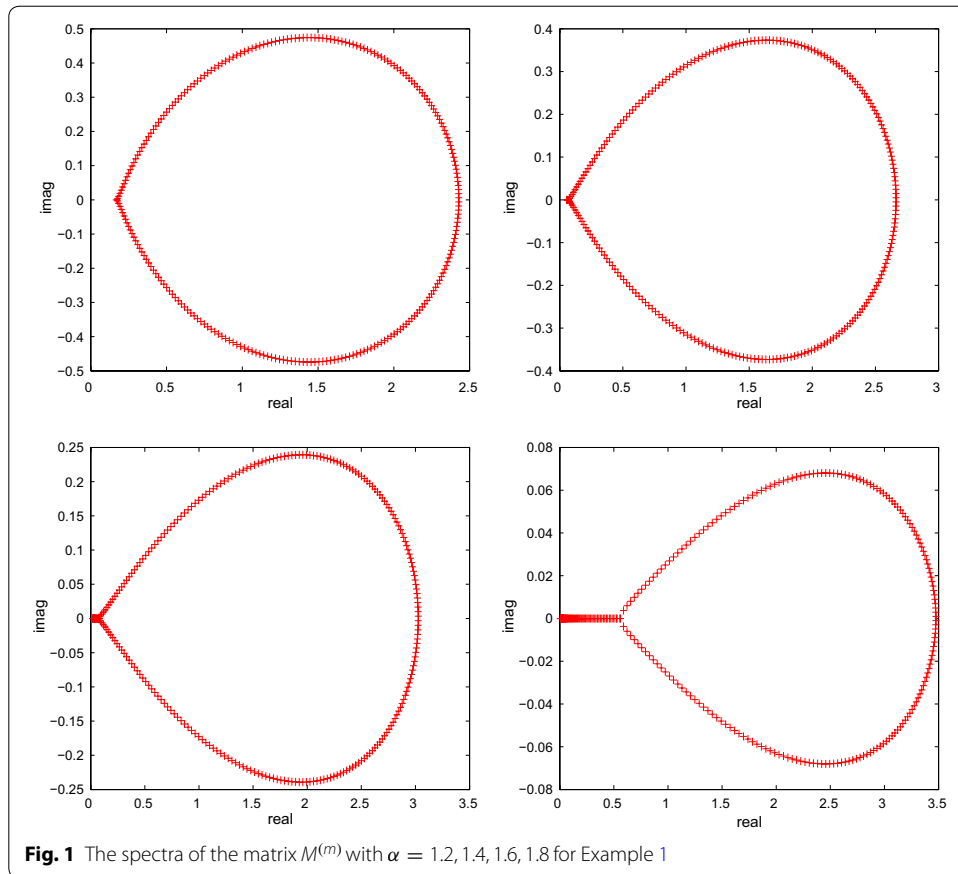
The exact solution is $u(x, t) = \sin(t + 1)x^3(1 - x)^3$. For the finite difference discretization, the space step and time step are taken to be $h = 1/(N + 1)$, $\Delta t = 2h$, i.e., $N + 1 = 2M$.

The numerical results are listed in Table 2, as for comparisons, we also carry out CGNR without preconditioner. From Table 2, it is remarkable that the error is improved for CGNR, BiCGT and BiCRT as the increasing of α . However, BiCGT and BiCRT are more effective than CGNR in terms of CPU time. More precisely, the performance of BiCGT is the best in terms of the CPU time except the cases of $\alpha = 1.4, N = 255$, $\alpha = 1.4, N = 511$, $\alpha = 1.4, N = 1023$, and $\alpha = 1.8, N = 1023$. In addition, the average number of iterations of BiCGT is less than that by CGNR and BiCRT sometimes. For instance, look at these cases in the numerical results at discretized size $N = 127, M = 64$, $N = 255, M = 128$, and $N = 511, M = 256$ for $\alpha = 1.8$. BiCGT and BiCRT have faster convergence speed, less computational time expenditure than CGNR. Meanwhile, we can see that the CPU time increases as the order α of the time derivative increases. To explain this phenomenon, we list the spectra of the original matrix $M^{(m)}$ with different α in Fig. 1. As we can see from the figure, most of eigenvalues is close to zero with the increasing of the value of α , it means that the coefficient matrix become increasingly ill-conditioned, it also implies that the linear systems will be difficult to solve.

Example 2 Consider FDEs (1) on space interval $[x_L, x_R] = [0, 1]$ and time interval $[0, T] = [0, 1]$ with diffusion coefficients $d_1 = 0.8$, $d_2 = 0.2$, initial condition $u_0(x) = x^2(1 - x)^2$, source term

Table 2 Comparisons for solving Example 1 by different methods with $\alpha = 1.4, 1.5$ and 1.8 at $t = 1$

α	$N + 1$	CGNR			BiCGT			BiCRT		
		CPU(s)	Error	Iter	CPU(s)	Error	Iter	CPU(s)	Error	Iter
1.4	64	0.129	3.7873e-4	69.6	0.101	3.7873e-4	63.6	0.103	3.7873e-4	63.7
	128	1.038	1.9163e-4	163.1	0.632	1.9163e-4	144.0	0.646	1.9163e-4	143.7
	256	4.828	9.6389e-5	272.0	3.963	9.6389e-5	354.0	3.898	9.6390e-5	338.7
	512	28.391	4.8338e-5	375.8	25.382	4.8338e-5	590.5	23.993	4.8342e-5	546.4
	1024	136.318	2.4205e-5	476.0	129.990	2.4206e-5	838.5	114.42	2.4212e-5	726.3
1.5	64	0.140	2.7756e-4	75.7	0.107	2.7756e-4	65.4	0.150	2.7756e-4	65.4
	128	1.206	1.4046e-4	188.2	0.760	1.4046e-4	162.6	0.844	1.4046e-4	162.2
	256	7.339	7.0658e-5	410.5	4.888	7.0654e-5	421.0	5.662	7.0654e-5	418.9
	512	51.154	3.5436e-5	681.4	42.872	3.5435e-5	955.4	45.933	3.5439e-5	904.0
	1024	289.79	1.7747e-5	1015.0	267.921	1.7744e-5	1657.2	277.638	1.7751e-5	1464.7
1.8	64	0.186	8.0708e-5	103.8	0.131	8.0708e-5	85.2	0.140	8.0708e-5	85.0
	128	1.852	4.0979e-5	292.8	1.042	4.0979e-5	234.3	1.100	4.0978e-5	234.0
	256	14.583	2.0659e-5	841.5	8.570	2.0659e-5	746.9	8.629	2.0659e-5	729.4
	512	182.390	1.0375e-5	2472.5	96.153	1.0374e-5	2205.7	96.526	1.0374e-5	2164.3
	1024	1708.245	5.2057e-6	6053.9	1075.756	5.1984e-6	6778.4	1058.220	5.1985e-6	6604.0



$$f(x, t) = -e^{-t} \left\{ x^2(1-x)^2 + \frac{\Gamma(3)}{\Gamma(3-\alpha)} [d_1 x^{2-\alpha} + d_2(1-x)^{2-\alpha}] - 2 \frac{\Gamma(4)}{\Gamma(4-\alpha)} [d_1 x^{3-\alpha} + d_2(1-x)^{3-\alpha}] + \frac{\Gamma(5)}{\Gamma(5-\alpha)} [d_1 x^{4-\alpha} + d_2(1-x)^{4-\alpha}] \right\}.$$

The exact solution of this example is $u(x, t) = e^{-t} x^2(1-x)^2$. For the finite difference discretization, the space step and time step are taken to be $h = 1/(N + 1)$, $\Delta t = 2h$ and $N + 1 = 2M$, respectively.

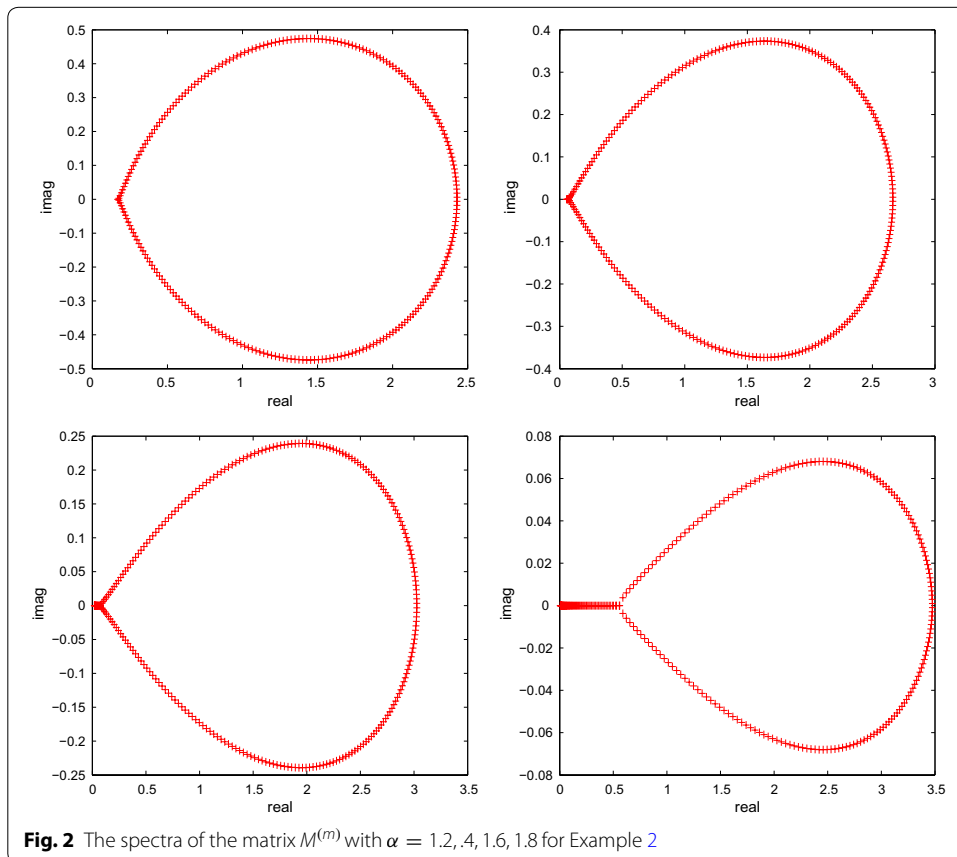
Table 3 shows the numerical results for solving Example 2 by different methods. The error is decreased for those methods as the increasing of α , and the accuracy is almost the same as Example 1. Similar to Example 1, BiCGT and BiCRT are more effective than CGNR in terms of CPU time elapsed. Besides, the CPU time increases as the order α of the time derivative increases is similar to Example 1, and the reason is the same as Example 1. We also list the spectra of the matrix $M^{(m)}$ with different α in Fig. 2.

Concluding remarks

Two new iterative methods, named BiCGT and BiCRT, are presented to solve the resulting linear system of the FDEs (1), which are discretized by the implicit finite difference method. Namely, with the help of the permutation matrix P , we transform the difficult nonsymmetric linear systems into the symmetric cases, which are often simpler to be

Table 3 Comparisons for solving Example 2 by different methods with $\alpha = 1.4, 1.5$ and 1.8 at $t = 1$

α	$N + 1$	CGNR			BiCGT			BiCRT		
		CPU(s)	Error	Iter	CPU(s)	Error	Iter	CPU(s)	Error	Iter
1.4	64	0.131	4.0801e-4	70	0.101	4.0801e-4	64.1	0.103	4.0801e-4	64.2
	128	1.007	2.0542e-4	157.3	0.671	2.0542e-4	151.9	0.682	2.0542e-4	151.7
	256	4.764	1.0283e-4	274.0	3.985	1.0283e-4	353.7	3.977	1.0283e-4	349.0
	512	30.947	5.1397e-5	408	27.848	5.1395e-5	643.2	26.268	5.1398e-5	601.4
1.5	64	0.147	2.7980e-4	76.3	0.100	2.7980e-4	66.9	0.111	2.7980e-4	66.6
	128	1.286	1.4222e-4	190.0	0.745	1.4222e-4	171.5	0.791	1.4222e-4	171.0
	256	7.410	7.1577e-5	397.5	4.806	7.1582e-5	427.2	4.962	7.1582e-5	426.0
	512	56.933	3.5883e-5	713.8	43.722	3.5877e-5	1012.6	42.825	3.5883e-5	972.4
1.8	64	0.182	9.8722e-5	105.5	0.144	9.8722e-5	91.3	0.237	9.8722e-5	90.2
	128	1.890	4.6159e-5	298.2	1.088	4.6159e-5	241.3	1.592	4.6159e-5	240.1
	256	15.027	2.2331e-5	858.4	8.798	2.2331e-5	758.0	10.687	2.2331e-5	746.2
	512	189.009	1.0989e-5	2503.8	105.062	1.0989e-5	2332.5	133.211	2.2331e-5	2283.1
	1024	1733.04	5.4529e-6	6104.3	1158.655	5.4529e-6	7074.2	1007.543	5.4529e-6	6900.6



solved. The computational complexity can be reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$ by utilizing FFTs. Numerical experiments illustrate the effectiveness of the proposed methods. In our future work, we will apply BiCGT and BiCRT to solve other (two dimensional) fractional differential equations (Wang and Basu 2012), such as fractional advection-diffusion equations; and we will investigate and develop some suitable preconditioning, see e.g. Lei and Sun (2013), Gu et al. (2015) to further accelerate our proposed methods.

Authors' contributions

SFW, TZH, XMG and WHL designed the study and numerical experiments, carried out the analysis and contributed to writing the paper. All authors have read and approved the final manuscript.

Author details

¹ School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People's Republic of China. ² Data Recovery Key Laboratory of Sichuan Province, Neijiang Normal University, Neijiang 641100, Sichuan, People's Republic of China.

Acknowledgements

This research is supported by NSFC (Nos. 61370147, and 61402082), the Fundamental Research Funds for the Central Universities (No. ZYGX2013Z005). We are grateful to the anonymous referees and editor Dr. Max Haring for their useful suggestions and comments that improved the presentation of this paper.

Competing interests

All authors declare that they have no competing interests.

Received: 29 March 2016 Accepted: 5 July 2016

Published online: 19 July 2016

References

- Allan FM, Al-Khaled K (2006) An approximation of the analytic solution of the shock wave equation. *J Comput Appl Math* 192:301–309
- Alquran M, Al-Khaled K, Sardar T, Chattopadhyay J (2015) Revisited Fisher's equation in a new outlook: a fractional derivative approach. *Phys A* 438:81–93
- Baeumer B, Benson DA, Meerschaert MM, Wheatcraft SW (2001) Subordinated advection-dispersion equation for contaminant transport. *Water Resour Res* 37:1543–1550
- Bai J, Feng X-C (2007) Fractional-order anisotropic diffusion for image denoising. *IEEE Trans Image Process* 16:2492–2502
- Benson D, Wheatcraft SW, Meerschaert MM (2000a) Application of a fractional advection-dispersion equation. *Water Resour Res* 36:1403–1413
- Benson D, Wheatcraft SW, Meerschaert MM (2000b) The fractional-order governing equation of Lévy motion. *Water Resour Res* 36:1413–1423
- Blackledge JM (2009) Diffusion and fractional diffusion based image processing. In: Tang W, Collomosse J (eds) *EG UK theory and practice of computer graphics*. Cardiff, 233–240. <http://arrow.dit.ie/engscheart/158/>
- Carreras BA, Lynch VE, Zaslavsky GM (2001) Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence models. *Phys Plasma* 8:5096–5103
- Chan R, Ng M (1996) Conjugate gradient methods for Toeplitz systems. *SIAM Rev* 38:427–482
- Ervin VJ, Heuer N, Roop JP (2007) Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation. *SIAM J Numer Anal* 45:572–591
- Gu X-M, Huang T-Z, Li H-B, Li L, Luo W-H (2015) On k -step CSCS-based polynomial preconditioners for Toeplitz linear systems with application to fractional diffusion equations. *Appl Math Lett* 42:53–58
- Gu X-M, Huang T-Z, Zhao X-L, Li H-B, Li L (2015) Strang-type preconditioners for solving fractional diffusion equations by boundary value methods. *J Comput Appl Math* 277:73–86
- Langlands TAM, Henry BI (2005) The accuracy and stability of an implicit solution method for the fractional diffusion equation. *J Comput Phys* 205:719–736
- Lei S-L, Sun H-W (2013) A circulant preconditioner for fractional diffusion equations. *J Comput Phys* 242:715–725
- Liu F, Anh V, Turner I (2004) Numerical solution of the space fractional Fokker-Planck equation. *J Comput Appl Math* 166:209–219
- Magin RL (2006) *Fractional calculus in bioengineering*. Begell House Publishers, Connecticut
- Meerschaert MM, Tadjeran C (2004) Finite difference approximations for fractional advection-dispersion flow equations. *J Comput Appl Math* 172:65–77
- Meerschaert MM, Tadjeran C (2006) Finite difference approximations for two-sided space-fractional partial differential equations. *Appl Numer Math* 56:80–90
- Paige CC, Saunders MA (1975) Solution of sparse indefinite systems of linear equations. *SIAM J Numer Anal* 12:617–629
- Pang H-K, Sun H-W (2012) Multigrid method for fractional diffusion equations. *J Comput Phys* 231:693–703
- Pestana J, Wathen AJ (2015) A preconditioned MINRES method for nonsymmetric Toeplitz matrices. *SIAM J Matrix Anal Appl* 36:273–288
- Podlubny I (1999) *Fractional differential equations*. Academic Press, New York

- Popolizio M (2015) A matrix approach for partial differential equations with Riesz space fractional derivatives. *Eur Phys J. Spec Top* 222:1975–1985
- Raberto M, Scalas E, Mainardi F (2002) Waiting-times and returns in high-frequency financial data: an empirical study. *Phys A* 314:749–755
- Saad Y (2003) Iterative methods for sparse linear systems, 2nd edn. SIAM, Philadelphia
- Sababheh MS, Nusayr AM, Al-Khaled K, (2003) Some convergence results on sinc interpolation. *J Inequal Pure Appl Math.* 4, Art. 32. <http://www.emis.de/journals/JIPAM/article270.html>
- Shlesinger MF, West BJ, Klafter J (1987) Lévy dynamics of enhanced diffusion: application to turbulence. *Phys Rev Lett* 58:1100–1103
- Sogabe T, Sugihara M, Zhang S-L (2009) An extension of the conjugate residual method to nonsymmetric linear systems. *J Comput Appl Math* 226:103–113
- Sogabe T, Zheng B, Hashimoto K, Zhang S-L (2005) A preconditioner of permutation matrix for solving nonsymmetric Toeplitz systems. *Trans Jpn Soc Ind Appl Math* 15 (2):159–168. <http://ci.nii.ac.jp/naid/10016594462> (in Japanese)
- Sokolov IM, Klafter J, Blumen A (2002) Fractional kinetics. *Phys Today* 55:48–54
- Tian WY, Zhou H, Deng W (2015) A class of second order difference approximations for solving space fractional diffusion equations. *Math Comp* 84:1703–1727
- Wang H, Basu TS (2012) A fast finite difference method for two-dimensional space-fractional diffusion equations. *SIAM J Sci Comput* 34:A2444–A2458
- Wang K, Wang H (2011) A fast characteristic finite difference method for fractional advection-diffusion equations. *Adv Water Res* 34:810–816
- Wang H, Wang K, Sircar T (2010) A direct $\mathcal{O}(N \log^{\epsilon} N)$ finite difference method for fractional diffusion equations. *J Comput Phys* 229:8095–8104
- Zaslavsky GM, Stevens D, Weitzner H (1993) Self-similar transport in incomplete chaos. *Phys Rev E* 48:1683–1694. doi:10.1103/PhysRevE.48.1683
- Zhang H, Liu F, Anh V (2010) Galerkin finite element approximation of symmetric space-fractional partial differential equations. *Appl Math Comput* 217:2534–2545

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
