A Circulant Preconditioner for Fractional Diffusion Equations☆

Siu-Long Lei[a], Hai-Wei Sun[a,*]

[a]*Department of Mathematics, University of Macau, Macao, China*

Abstract

The implicit finite difference scheme with the shifted Grünwald formula, which is unconditionally stable, is employed to discretize fractional diffusion equations. The resulting systems are Toeplitz-like and then the fast Fourier transform can be used to reduce the computational cost of the matrix-vector multiplication. The preconditioned conjugate gradient normal residual method with a circulant preconditioner is proposed to solve the discretized linear systems. The spectrum of the preconditioned matrix is proven to be clustered around 1 if diffusion coefficients are constant; hence the convergence rate of the proposed iterative algorithm is superlinear. Numerical experiments are carried out to demonstrate that our circulant preconditioner works very well, even though for cases of variable diffusion coefficients.

Keywords: fractional diffusion equations, shifted Grünwald discretization, Toeplitz, circulant preconditioner, fast Fourier transform, CGNR method
2000 MSC: 65F10, 65L12, 65L20, 65T50, 26A33

1. Introduction

In the last decade or so, the catholicity of anomalous diffusion phenomena in the real world has led to the fractional diffusion equation (FDE). FDEs arise in research topics including modeling chaotic dynamics of classical conservative systems [33], groundwater contaminant transport [3, 4], turbulent flow [6, 25], and applications in biology [16], finance [24], image processing [1], and physics [26]. As there are very few cases of FDEs in which the closed-form analytical solutions are available, numerical solutions for FDEs become major ways and then have been developed intensively [5, 11, 12, 14, 15, 17, 18, 19, 20, 27, 28, 29].

Nevertheless, since the fractional differential operator is nonlocal, it was shown that a naive discretization of the FDE, even though implicit, leads to unconditionally unstable [18, 19]. Moreover, most numerical methods for FDEs tend to generate full coefficient

☆This work was partially supported by the research grant MYRG038(Y1-L1)-FST12-LSL and MYRG140(Y3-L2)-FST11-SHW from University of Macau, and the research grant 105/2012/A3 from FDCT of Macao.

*Corresponding author
Email addresses: sllei@umac.mo (Siu-Long Lei), HSun@umac.mo (Hai-Wei Sun)

matrices, which require computational cost of $\mathcal{O}(N^3)$ and storage of $\mathcal{O}(N^2)$, where N is the number of grid points [31]. It is quite different from second-order diffusion equations which usually yield sparse coefficient matrices with $\mathcal{O}(N)$ nonzero entries and can be solved very efficiently by fast iterative methods with $\mathcal{O}(N)$ complexity.

To overcome the difficulty of the stability, Meerschaet and Tadjeran [18, 19] proposed a shifted Grünwald discretization to approximate FDEs. Their method has been proven to be unconditionally stable. Later, Wang, Wang, and Sircar [31] discovered that the full coefficient matrix by the Meerschaet-Tadjeran's method holds 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 computed by the fast Fourier transform (FFT) with $\mathcal{O}(N \log N)$ operations [8, 21]. With this advantage, Wang and Wang [32] employed the conjugate gradient normal residual (CGNR) method to solve the discretized system of the FDE by the Meerschaet-Tadjeran's method. Thanks to the Toeplitz-like structure, the cost per iteration by the CGNR method is of $\mathcal{O}(N \log N)$. The convergence of the CGNR method is fast with smaller diffusion coefficients [32] (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 [22] proposed a multigrid method to solve the discretized system of the FDE by the Meerschaet-Tadjeran's method. With the damped-Jacobi method as the smoother, the multigrid algorithm can preserve the computational cost per iteration as $\mathcal{O}(N \log N)$ 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 [22] for details.

In the literature on Toeplitz or Toeplitz-like systems, circulant preconditioners always played important roles. In fact, circulant preconditioners have been theoretically and numerically studied with numerous applications for over twenty years; see for instances [7, 8, 21]. Nevertheless, to our knowledge, circulant preconditioners were never particularly applied to the Toeplitz-like matrix with structure as the sum of diagonal-multiply-Toeplitz matrices before. In this paper, we propose the preconditioned CGNR (PCGNR) method with a circulant preconditioner to solve such Toeplitz-like systems. The proposed preconditioner is invertible and its spectrum is theoretically proven to be clustered around 1 under some conditions. Thus the superlinear convergence rate of the PCGNR method is obtained. As the computational cost per iteration is of $\mathcal{O}(N \log N)$, the total complexity of the PCGNR method at each time step retains $\mathcal{O}(N \log N)$ operations.

The paper is organized as follows. In Section 2, the background of the discretization for the FDE is reviewed. In Section 3, the PCGNR method with a circulant preconditioner is proposed to solve the discretized linear system. The convergence rate of the PCGNR method is studied in Section 4. In Section 5, numerical results are reported to demonstrate the efficiency of the proposed method. Concluding remarks are given in Section 6.

2. FDE and finite difference discretization

In this paper, we study an initial-boundary value problem of the FDE:

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

where $\alpha \in (1,2)$ is the order of the fractional derivative, $f(x,t)$ is the source term, and diffusion coefficient functions $d_\pm(x,t)$ are nonnegative; i.e., $d_\pm(x,t) \ge 0$. 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 [23]

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

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

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

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

which can be evaluated by the recurrence relation

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

Let N and M be positive integers, and $\Delta x = \frac{x_R - x_L}{N+1}$ and $\Delta t = T/M$ be the sizes of spatial grid and time step, respectively. We define a spatial and temporal partition $x_i = x_L + i\Delta x$ for $i = 0,1,\ldots,N+1$ and $t_m = m\Delta t$ for $m = 0,1,\ldots,M$. Let $u_i^{(m)} = u(x_i,t_m)$, $d_{\pm,i}^{(m)} = d_\pm(x_i,t_m)$, and $f_i^{(m)} = f(x_i,t_m)$. The shifted Grünwald approximation in [18, 19] is as follows,

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

where $g_k^{(\alpha)}$ is defined in (2), and the corresponding implicit finite difference scheme

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

is unconditionally stable. The alternating fractional binomial coefficient $g_k^{(\alpha)}$ have some useful properties, that are observed in [18, 19, 31], and are summarized in the following proposition.

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

$$\begin{cases} g_0^{(\alpha)} = 1, \quad g_1^{(\alpha)} = -\alpha < 0, \quad g_2^{(\alpha)} > g_3^{(\alpha)} > \cdots > 0, \\ \sum\limits_{k=0}^{\infty} g_k^{(\alpha)} = 0, \quad \sum\limits_{k=0}^{n} g_k^{(\alpha)} < 0, \ \forall\, n \geq 1. \end{cases} \tag{4}$$

Let $u^{(m)} = \left[u_1^{(m)}, u_2^{(m)}, \cdots, u_N^{(m)}\right]^\intercal$, $f^{(m)} = \left[f_1^{(m)}, f_2^{(m)}, \cdots, f_N^{(m)}\right]^\intercal$, and I be the identity matrix with an appropriate size. Then the numerical scheme (3) can be written in the following matrix form

$$\left(\frac{\Delta x^\alpha}{\Delta t} I + A^{(m)}\right) u^{(m)} = \frac{\Delta x^\alpha}{\Delta t} u^{(m-1)} + \Delta x^\alpha f^{(m)}, \tag{5}$$

with

$$A^{(m)} = D_+^{(m)} G_\alpha + D_-^{(m)} G_\alpha^\intercal, \tag{6}$$

where $D_\pm^{(m)} = \mathrm{diag}(d_{\pm,1}^{(m)}, \cdots, d_{\pm,N}^{(m)})$ and

$$G_\alpha = -\begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 \\ \vdots & g_2^{(\alpha)} & g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{N-1}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_N^{(\alpha)} & g_{N-1}^{(\alpha)} & \cdots & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}_{N\times N}. \tag{7}$$

It is obvious that G_α is a Toeplitz matrix (see [8, 9]). Therefore, it can be stored with $N+1$ entries [31]. Furthermore, the matrix-vector multiplication for the Toeplitz-like matrix $A^{(m)}$ in (6) can be obtained in $\mathcal{O}(N \log N)$ operations by the FFT; see [8, 22].

Denote

$$\nu_{N,M} = \frac{\Delta x^\alpha}{\Delta t} = (x_R - x_L)^\alpha T^{-1} \frac{M}{(N+1)^\alpha} \tag{8}$$

which is related to the numbers of time steps and grid points. The linear system (5) can be written as

$$\mathcal{M}^{(m)} u^{(m)} = b^{(m-1)}, \tag{9}$$

where

$$\mathcal{M}^{(m)} = \nu_{N,M} I + A^{(m)} = \nu_{N,M} I + D_+^{(m)} G_\alpha + D_-^{(m)} G_\alpha^\intercal \tag{10}$$

and

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

The coefficient matrix $\mathcal{M}^{(m)}$ in (10) is a strictly diagonally dominant M-matrix (see [31]) and therefore it is nonsingular.

3. PCGNR method with circulant preconditioner

The conjugate gradient (CG) method is a popular and effective iterative method for solving symmetric positive definite linear systems. Furthermore, for the nonsymmetric linear system (9), the CG method can also be applied to its equivalent normalized linear system

$$(\mathcal{M}^{(m)})^\intercal \mathcal{M}^{(m)} u^{(m)} = (\mathcal{M}^{(m)})^\intercal b^{(m-1)},$$

which is called the CGNR method [2]. The algorithm of the CGNR method is given in Algorithm 1 below.

Algorithm 1. CGNR for $\mathcal{M}^{(m)} u^{(m)} = b^{(m-1)}$

1. Given initial guess $u_0^{(m)}$
2. $r_0 = b^{(m-1)} - \mathcal{M}^{(m)} u_0^{(m)}$
3. $z_0 = (\mathcal{M}^{(m)})^\intercal r_0$, $p_0 = z_0$
4. For $i = 0, 1, \ldots$, until convergence, Do
5. $\quad w_i = \mathcal{M}^{(m)} p_i$
6. $\quad \alpha_i = ||z_i||_2^2 / ||w_i||_2^2$
7. $\quad u_{i+1}^{(m)} = u_i^{(m)} + \alpha_i p_i$
8. $\quad r_{i+1} = r_i - \alpha_i w_i$
9. $\quad z_{i+1} = (\mathcal{M}^{(m)})^\intercal r_{i+1}$
10. $\quad \beta_i = ||z_{i+1}||_2^2 / ||z_i||_2^2$
11. $\quad p_{i+1} = z_{i+1} + \beta_i p_i$
12. EndDo

In [32], the CGNR method is employed to solve (5). However, the drawback of the CGNR method is at its slow convergence rate, due to the fact that the condition number of the coefficient matrix $(\mathcal{M}^{(m)})^\intercal \mathcal{M}^{(m)}$ is usually large. In fact, the numerical results given in Section 5 show that the CGNR method really has a poor convergence rate for ill-conditioned systems. Therefore, preconditioner technique could be exploited to accelerate the convergence rate of the CGNR method.

When the CG method, with a matrix $\mathcal{P}^{(m)}$, is employed to solve the following normalized preconditioned system

$$\left[(\mathcal{P}^{(m)})^{-1} \mathcal{M}^{(m)} \right]^\intercal \left[(\mathcal{P}^{(m)})^{-1} \mathcal{M}^{(m)} \right] u^{(m)} = \left[(\mathcal{P}^{(m)})^{-1} \mathcal{M}^{(m)} \right]^\intercal (\mathcal{P}^{(m)})^{-1} b^{(m-1)},$$

it is equivalent to the original system $\mathcal{M}^{(m)}u^{(m)} = b^{(m-1)}$ mathematically. This method is called the PCGNR method with a preconditioner $\mathcal{P}^{(m)}$. The algorithm of the PCGNR method is given in Algorithm 2 below, which is the direct application of the Algorithm 1 to the system $(\mathcal{P}^{(m)})^{-1}\mathcal{M}^{(m)}u^{(m)} = (\mathcal{P}^{(m)})^{-1}b^{(m-1)}$.

Algorithm 2. PCGNR for $\mathcal{M}^{(m)}u^{(m)} = b^{(m-1)}$ with preconditioner $\mathcal{P}^{(m)}$

1. Given initial guess $u_0^{(m)}$
2. $r_0 = (\mathcal{P}^{(m)})^{-1}(b^{(m-1)} - \mathcal{M}^{(m)}u_0^{(m)})$
3. $z_0 = \left[(\mathcal{P}^{(m)})^{-1}\mathcal{M}^{(m)}\right]^{\intercal} r_0,\ p_0 = z_0$
4. For $i = 0, 1, \ldots,$ until convergence, Do
5. $\quad w_i = \mathcal{M}^{(m)}p_i$
6. $\quad w_i = (\mathcal{P}^{(m)})^{-1}w_i$
7. $\quad \alpha_i = ||z_i||_2^2/||w_i||_2^2$
8. $\quad u_{i+1}^{(m)} = u_i^{(m)} + \alpha_i p_i$
9. $\quad r_{i+1} = r_i - \alpha_i w_i$
10. $\quad z_{i+1} = (\mathcal{P}^{(m)})^{-\intercal} r_{i+1}$
11. $\quad z_{i+1} = (\mathcal{M}^{(m)})^{\intercal} z_{i+1}$
12. $\quad \beta_i = ||z_{i+1}||_2^2/||z_i||_2^2$
13. $\quad p_{i+1} = z_{i+1} + \beta_i p_i$
14. EndDo

Now we propose a circulant preconditioner, which is generated from the Strang's circulant preconditioner [9], in the PCGNR method to solve (9). The Strang's circulant matrix $s(B) = [s_{j-k}]_{0\le j,k<N}$ for a real Toeplitz matrix $B = [b_{j-k}]_{0\le j,k<N}$ is obtained by copying the central diagonals of B and bringing them around to complete the circulant requirement [8, 13]. More precisely, the diagonals of $s(B)$ are given by

$$s_j = \begin{cases} b_j, & 0 \le j < N/2, \\ 0, & j = N/2 \text{ if } N \text{ is even}, \\ b_{j-N}, & N/2 < j < N, \\ s_{j+N}, & 0 < -j < N. \end{cases}$$

Recall the formula (10) that

$$\mathcal{M}^{(m)} = \nu_{N,M} I + A^{(m)} = \nu_{N,M} I + D_+^{(m)} G_\alpha + D_-^{(m)} G_\alpha^{\intercal},$$

where

$$D_\pm^{(m)} = \mathrm{diag}(d_{\pm,1}^{(m)}, \cdots, d_{\pm,N}^{(m)}),$$

and G_α is the Toeplitz matrix given in (7). Then our circulant preconditioner is defined as

$$\mathcal{S}^{(m)} = \nu_{N,M} I + \overline{d}_+^{(m)} s(G_\alpha) + \overline{d}_-^{(m)} s(G_\alpha^{\intercal}), \tag{11}$$

where

$$\overline{d}_{\pm}^{(m)} = \frac{1}{N}\sum_{i=1}^{N} d_{\pm,i}^{(m)}.$$

More precisely, the first columns of both $s(G_\alpha)$ and $s(G_\alpha^\intercal)$ are given by

$$-\begin{bmatrix} g_1^{(\alpha)} \\ \vdots \\ g_{\lfloor\frac{N+1}{2}\rfloor}^{(\alpha)} \\ 0 \\ \vdots \\ 0 \\ g_0^{(\alpha)} \end{bmatrix} \quad \text{and} \quad -\begin{bmatrix} g_1^{(\alpha)} \\ g_0^{(\alpha)} \\ 0 \\ \vdots \\ 0 \\ g_{\lfloor\frac{N+1}{2}\rfloor}^{(\alpha)} \\ \vdots \\ g_2^{(\alpha)} \end{bmatrix},$$

respectively.

We know that a circulant matrix C can be diagonalized by the Fourier matrix F [9]; i.e.,

$$C = F^*\Lambda F,$$

where the entries of F are given by

$$F_{j,k} = \frac{1}{\sqrt{N}} e^{2\pi \mathbf{i} jk/N}, \qquad 0 \le j,k \le N-1,$$

with the imaginary unit $\mathbf{i}$, and Λ is a diagonal matrix holding the eigenvalues of C.

Let $s(G_\alpha) = F^*\Lambda_\alpha F$. Then $s(G_\alpha^\intercal) = F^*\bar{\Lambda}_\alpha F$, where $\bar{\Lambda}_\alpha$ is the complex conjugate of Λ_α. The following lemma is the key to prove the invertibility of $\mathcal{S}^{(m)}$ in (11).

Lemma 1. *All eigenvalues of $s(G_\alpha)$ and $s(G_\alpha^\intercal)$ fall inside the open disc*

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

Proof: All the Gershgorin disc of the circulant matrices $s(G_\alpha)$ and $s(G_\alpha^\intercal)$ are centered at $-g_1^{(\alpha)} = \alpha$ with radius

$$r_N = g_0^{(\alpha)} + \sum_{k=2}^{\lfloor\frac{N+1}{2}\rfloor} g_k^{(\alpha)} < \sum_{k=0,k\neq 1}^{\infty} g_k^{(\alpha)} = -g_1^{(\alpha)} = \alpha,$$

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

Remark 1. *It is worth to note that:*

1. *The real parts of all eigenvalues of $s(G_\alpha)$ and $s(G_\alpha^\intercal)$ are strictly positive for all N.*

2. *The absolute values of all eigenvalues of $s(G_\alpha)$ and $s(G_\alpha^\intercal)$ are bounded above by $2\alpha < 4$ for all N.*

Decompose the circulant matrix $\mathcal{S}^{(m)} = F^*\Lambda_s F$ with the diagonal matrix $\Lambda_s = \nu_{N,M} I + \overline{d}_+^{(m)}\Lambda_\alpha + \overline{d}_-^{(m)}\bar{\Lambda}_\alpha$. Then $\mathcal{S}^{(m)}$ is invertible if all diagonal entries of Λ_s are nonzero.

Lemma 2. *Let $1 < \alpha < 2$. The preconditioner $\mathcal{S}^{(m)}$ in (11) is invertible and*

$$||(\mathcal{S}^{(m)})^{-1}||_2 \leq \frac{1}{\nu_{N,M}}. \tag{12}$$

Proof: By Part 1 of Remark 1, we have $\mathrm{Re}([\Lambda_\alpha]_{k,k}) > 0$. Noting that $\nu_{N,M} > 0$ and $\overline{d}_\pm^{(m)} \geq 0$, we have

$$\begin{aligned} |[\Lambda_s]_{k,k}| &\geq \mathrm{Re}([\Lambda_s]_{k,k}) \\ &= \nu_{N,M} + \overline{d}_+^{(m)}\mathrm{Re}([\Lambda_\alpha]_{k,k}) + \overline{d}_-^{(m)}\mathrm{Re}([\bar{\Lambda}_\alpha]_{k,k}) \\ &\geq \nu_{N,M} > 0 \end{aligned}$$

for each $k = 1, \ldots, N$. Therefore, $\mathcal{S}^{(m)}$ is invertible. Furthermore, we have

$$||(\mathcal{S}^{(m)})^{-1}||_2 = \frac{1}{\min\limits_{1\leq k\leq N} |[\Lambda_s]_{k,k}|} \leq \frac{1}{\nu_{N,M}}.$$

□

The computational cost per iteration of the PCGNR method with circulant preconditioners is $\mathcal{O}(N \log N)$ complexity, as the main workload is the matrix-vector multiplication. Therefore, if the iterative method is of linear or superlinear convergence, the total cost for solving (9) at each time step by the proposed method will keep $\mathcal{O}(N \log N)$ operations.

Before to analyze the convergence of the proposed method, we prove the following theorem which gives a bound of $||\mathcal{S}^{(m)}||_2$. This theorem will be exploited to prove the superlinear convergence rate of the PCGNR method in next section.

Lemma 3. *Let $1 < \alpha < 2$ and $0 \leq d_\pm(x,t) \leq \widetilde{d}$. We have*

$$||\mathcal{S}^{(m)}||_2 \leq \nu_{N,M} + 8\widetilde{d}. \tag{13}$$

Proof: For any $k = 1, \ldots, N$, by Part 2 of Remark 1,

$$\begin{aligned} |[\Lambda_s]_{k,k}| &= |\nu_{N,M} + \overline{d}_+^{(m)}[\Lambda_\alpha]_{k,k} + \overline{d}_-^{(m)}[\bar{\Lambda}_\alpha]_{k,k}| \\ &\leq \nu_{N,M} + \overline{d}_+^{(m)}|[\Lambda_\alpha]_{k,k}| + \overline{d}_-^{(m)}|[\bar{\Lambda}_\alpha]_{k,k}| \\ &< \nu_{N,M} + 4\overline{d}_+^{(m)} + 4\overline{d}_-^{(m)} \\ &\leq \nu_{N,M} + 8\widetilde{d}. \end{aligned}$$

Then

$$||\mathcal{S}^{(m)}||_2 = \max_{1\le k\le N} |[\Lambda_s]_{k,k}| < \nu_{N,M} + 8\widetilde{d}.$$

□

4. Spectrum of the preconditioned matrix

In this section, we study the convergence rate of the PCGNR method with the proposed circulant preconditioner $\mathcal{S}^{(m)}$ in (11) for the linear system (9). For convenience of our investigation, we assume that the left and the right diffusion coefficient functions $d_\pm(x,t)$ of the FDE (1) are two nonnegative constants and M is properly chosen, depending on N, such that $\nu_{N,M}$ in (8) is bounded away from 0; i.e., there exist two real numbers $\check{\nu}$ and $\hat{\nu}$ such that

$$0 < \check{\nu} \le \nu_{N,M} \le \hat{\nu}, \qquad \forall N. \tag{14}$$

By this assumption, for all $i = 1, \ldots, N$,

$$d_{+,i}^{(m)} = d_+ \ge 0, \quad d_{-,i}^{(m)} = d_- \ge 0, \quad \text{and} \quad d_+ + d_- \ne 0. \tag{15}$$

Throughout this section, we add a subscript N to each matrix to denote the matrix size. Under the assumption in (15), the matrix $A^{(m)}$ defined in (6), $\mathcal{M}^{(m)}$ defined in (10), and $\mathcal{S}^{(m)}$ defined in (11) are independent of m, and we therefore simply denote them as A_N, $\mathcal{M}_N$, and $\mathcal{S}_N$, respectively. Now the coefficient matrix $\mathcal{M}^{(m)}$ in (10) becomes

$$\mathcal{M}_N = \nu_{N,M} I + A_N, \tag{16}$$

which is a nonsymmetric Toeplitz matrix, where

$$\begin{aligned} A_N &= d_+ G_\alpha + d_- G_\alpha^{\mathsf{T}} \\ &= -\begin{bmatrix} d_+ g_1^{(\alpha)} + d_- g_1^{(\alpha)} & d_+ g_0^{(\alpha)} + d_- g_2^{(\alpha)} & d_- g_3^{(\alpha)} & \cdots & d_- g_N^{(\alpha)} \\ d_+ g_2^{(\alpha)} + d_- g_0^{(\alpha)} & d_+ g_1^{(\alpha)} + d_- g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ d_+ g_3^{(\alpha)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & d_+ g_0^{(\alpha)} + d_- g_2^{(\alpha)} \\ d_+ g_N^{(\alpha)} & \cdots & \cdots & d_+ g_2^{(\alpha)} + d_- g_0^{(\alpha)} & d_+ g_1^{(\alpha)} + d_- g_1^{(\alpha)} \end{bmatrix} \\ &\equiv [a_{j-k}]_{N\times N}. \end{aligned} \tag{17}$$

To study the spectrum of the preconditioned matrix

$$\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)^{\mathsf{T}} \left(\mathcal{S}_N^{-1}\mathcal{M}_N\right),$$

we first introduce the *generating function* of the sequence of Toeplitz matrices $\{A_N\}_{N=1}^{\infty}$ [8]:

$$p(\theta) = \sum_{k=-\infty}^{\infty} a_k e^{\mathbf{i}k\theta}, \tag{18}$$

where a_k is the k-th diagonal of A_N. The generating function $p(\theta)$ is in the *Wiener class* [8, 9] if and only if

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

For A_N defined in (17), we have

$$p(\theta) = \sum_{k=-\infty}^{\infty} a_k e^{\mathbf{i}k\theta} = -\sum_{k=-1}^{\infty} g_{k+1}^{(\alpha)}(d_+ e^{\mathbf{i}k\theta} + d_- e^{-\mathbf{i}k\theta}). \tag{19}$$

We remark that $\nu_{N,M} + p(\theta)$ cannot be a generating function of $\mathcal{M}_N$ in (16) since $\nu_{N,M}$ is dependent on N.

Lemma 4. *Let p and q be the generating functions of $\{A_N\}_{N=1}^{\infty}$ and $\{A_N + A_N^{\intercal}\}_{N=1}^{\infty}$, respectively, we have*

1. *p is in the Wiener class;*
2. *q is real-valued and nonnegative.*

Proof:

1. By the properties of the sequence $\{g_k^{(\alpha)}\}_{k=0}^{\infty}$ given in (4), we have

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

   Thus p is in the Wiener class.

2. From (4) and the definition of generating function in (18), we have

$$\begin{aligned}
& q(\theta) = \sum_{k=-\infty}^{\infty} (a_k + a_{-k}) e^{\mathbf{i}k\theta} \\
= \; & -(d_+ + d_-) \left[2g_1^{(\alpha)} + \left(g_0^{(\alpha)} + g_2^{(\alpha)} \right) \left(e^{\mathbf{i}\theta} + e^{-\mathbf{i}\theta} \right) + \sum_{k=2}^{\infty} g_{k+1}^{(\alpha)} \left(e^{\mathbf{i}k\theta} + e^{-\mathbf{i}k\theta} \right) \right] \\
= \; & -(d_+ + d_-) \left[2g_1^{(\alpha)} + 2 \left(g_0^{(\alpha)} + g_2^{(\alpha)} \right) \cos\theta + 2 \sum_{k=2}^{\infty} g_{k+1}^{(\alpha)} \cos k\theta \right] \\
\geq \; & -2(d_+ + d_-) \sum_{k=-1}^{\infty} g_{k+1}^{(\alpha)} = 0.
\end{aligned}$$

□

From Part 2 of Lemma 4, and according to the Grenander-Szegö Theorem (see Theorem 1.12 in [7]), the matrix $A_N + A_N^\intercal$ is symmetric positive semidefinite. Furthermore, by noting that

$$\mathcal{M}_N \mathcal{M}_N^\intercal = \nu_{N,M}^2 I + \nu_{N,M} \left(A_N + A_N^\intercal\right) + A_N A_N^\intercal,$$

we conclude that the smallest singular value of the matrix $\mathcal{M}_N$ is bounded below by $\nu_{N,M}$; i.e.,

$$\sigma_{\min}(\mathcal{M}_N) \geq \nu_{N,M}. \tag{20}$$

By Part 1 of Lemma 4, we have the following lemma.

Lemma 5. (see [8]) *If $p(\theta)$, the generating function of A_N, is in the Wiener class, then for any $\epsilon > 0$, there exist N' and $M' > 0$, such that for all $N > N'$,*

$$A_N - s(A_N) = U_N + V_N$$

where

$$\text{rank}(U_N) \leq M'$$

and

$$||V_N||_2 < \epsilon.$$

Now we consider the spectrum of $\mathcal{S}_N^{-1}\mathcal{M}_N - I$. By Lemma 5, we have

$$\mathcal{S}_N^{-1}\mathcal{M}_N - I = \mathcal{S}_N^{-1}(\mathcal{M}_N - \mathcal{S}_N) = \mathcal{S}_N^{-1}U_N - \mathcal{S}_N^{-1}V_N.$$

Note that $\text{rank}(\mathcal{S}_N^{-1}U_N) \leq \text{rank}(U_N) \leq M'$ and

$$||\mathcal{S}_N^{-1}V_N||_2 \leq ||\mathcal{S}_N^{-1}||_2||V_N||_2 < \frac{\epsilon}{\nu_{N,M}} \leq \frac{\epsilon}{\check{\nu}},$$

by the assumption (14) and the inequality (12) in Lemma 2. Therefore, the following corollary is resulted.

Corollary 1. *If $\nu_{N,M}$ satisfies the assumption (14), for any $\epsilon > 0$, there exists N'' and $M'' > 0$ such that, for any $N > N''$,*

$$\mathcal{S}_N^{-1}\mathcal{M}_N - I = \widetilde{U}_N + \widetilde{V}_N, \tag{21}$$

where $\text{rank}(\widetilde{U}_N) \leq M''$ *and* $||\widetilde{V}_N||_2 < \epsilon$.

Now we show that the spectrum of the normalized preconditioned matrix

$$\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)^\intercal \left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)$$

is clustered around 1.

Theorem 1. *If $\nu_{N,M}$ satisfies the assumption (14), for any $0 < \epsilon < 1$, there exists N'' and $M'' > 0$ such that, for all $N > N''$, at most $2M''$ eigenvalues of the matrix*

$$\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)^{\intercal}\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right) - I$$

have absolute values larger than 3ϵ.

Proof: By (21), we have

$$\begin{aligned} & \left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)^{\intercal}\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right) \\ = \; & (I + \widetilde{U}_N + \widetilde{V}_N)^{\intercal}(I + \widetilde{U}_N + \widetilde{V}_N) \\ = \; & I + \widehat{U}_N + \widehat{V}_N, \end{aligned}$$

where

$$\widehat{U}_N = \widetilde{U}_N^{\intercal}(I + \widetilde{U}_N + \widetilde{V}_N) + (I + \widetilde{V}_N^{\intercal})\widetilde{U}_N$$

and

$$\widehat{V}_N = \widetilde{V}_N + \widetilde{V}_N^{\intercal} + \widetilde{V}_N^{\intercal}\widetilde{V}_N.$$

Then by Corollary 1, we see that $\mathrm{rank}(\widehat{U}_N) \leq 2M''$ and $||\widehat{V}_N||_2 < 3\epsilon$. Thus, we have

$$\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)^{\intercal}\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right) - I = \widehat{U}_N + \widehat{V}_N,$$

where both $\widehat{U}_N$ and $\widehat{V}_N$ are real and symmetric. By the well-known Weyl's theorem, we conclude that at most $2M''$ eigenvalues of the matrix

$$\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right)^{\intercal}\left(\mathcal{S}_N^{-1}\mathcal{M}_N\right) - I$$

have absolute values larger than 3ϵ. □

For the superlinear convergence of the PCGNR method, according to the corollary 1.11 in [7], it is further required that the smallest singular value of the matrix $\mathcal{S}_N^{-1}\mathcal{M}_N$ is uniformly bounded away from zero. In fact, using (13), (20), and the assumption (14), we have

$$\sigma_{\min}(\mathcal{S}_N^{-1}\mathcal{M}_N) \geq \frac{\sigma_{\min}(\mathcal{M}_N)}{||\mathcal{S}_N||_2} \geq \frac{\nu_{N,M}}{\nu_{N,M} + 8\widetilde{d}} \geq \frac{\hat{\nu}}{\check{\nu} + 8\widetilde{d}} > 0,$$

where $\widetilde{d} = \max\{d_+, d_-\}$. Therefore, the PCGNR method for solving (9) converges superlinearly.

5. Numerical results

In this section, we solve the FDE problem (1) numerically by the implicit finite difference method given in Section 2. After the finite difference discretization, at each time step, the nonsymmetric linear system $\mathcal{M}^{(m)}u^{(m)} = b^{(m-1)}$ is solved by the CGNR method (Algorithm 1), the PCGNR method (Algorithm 2), and the multigrid method [22], respectively. Number of iterations required for convergence and CPU time of those methods are reported. The stopping criterion of those methods is

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

where $r^{(k)}$ is the residual vector of the linear system after k iterations, and the initial guess at each time step is chosen as the zero vector. In the following tables, "N" denotes the number of spatial grid points, "M" denotes the number of time steps, "Error" denotes the error between the true solution and the approximation under the infinity norm, "CPU(s)" denotes the total CPU time in seconds for solving the whole FDE problem, and "Iter" denotes the average number of iterations required for solving the FDE problem; i.e.,

$$\text{Iter} = \frac{1}{M}\sum_{m=1}^{M} \text{Iter}(m),$$

where Iter(m) denotes the number of iterations required for solving (9). For the PCGNR method, besides the proposed circulant preconditioner $\mathcal{S}^{(m)}$ in (11), we also test the T. Chan's preconditioner [10] which can be written as

$$c(\mathcal{M}^{(m)}) = \nu_{N,M} I + \overline{d}_{+}^{(m)} c(G_\alpha) + \overline{d}_{-}^{(m)} c(G_\alpha^{\mathsf{T}}),$$

where $c(B)$ denotes the T. Chan's preconditioner for an arbitrary matrix B. More precisely, the first columns of the circulant matrices $c(G_\alpha)$ and $c(G_\alpha^{\mathsf{T}})$ are given as

$$-\frac{1}{N}\begin{bmatrix} N g_1^{(\alpha)} \\ (N-1) g_2^{(\alpha)} \\ \vdots \\ 2 g_{N-1}^{(\alpha)} \\ g_N^{(\alpha)} + (N-1) g_0^{(\alpha)} \end{bmatrix} \quad \text{and} \quad -\frac{1}{N}\begin{bmatrix} N g_1^{(\alpha)} \\ (N-1) g_0^{(\alpha)} + g_N^{(\alpha)} \\ 2 g_{N-1}^{(\alpha)} \\ \vdots \\ (N-1) g_2^{(\alpha)} \end{bmatrix},$$

respectively. We remark that the superlinear convergence of the PCGNR method with the preconditioner $c(\mathcal{M}^{(m)})$, as that with $\mathcal{S}^{(m)}$ in Section 4, can be analogously proven if the assumptions (14) and (15) hold. In the following tables, we denote "$\mathcal{I}$" as the CGNR method, "$\mathcal{C}$" as the PCGNR method with the T. Chan's circulant preconditioner $c(\mathcal{M}^{(m)})$, "$\mathcal{S}$" as the PCGNR method with the proposed circulant preconditioner $\mathcal{S}^{(m)}$, and "MGM" as the multigrid method in [22]. In Table 2, "Error" denotes the error between the exact

solution and the numerical solution obtained by the PCGNR method with the proposed preconditioner $S^{(m)}$. All numerical experiments are run in MATLAB 7.12 (R2011a) on a Dell workstation Precision T7500 with the configuration: Intel(R) Xeon(R) CPU X5690 3.47GHz and 12 GB RAM.

Example 1. In this example, we solve the initial-boundary value problem of FDE (1) with source term $f(x,t) \equiv 0$, for order of fractional derivatives $\alpha = 1.2$, 1.5 and 1.8. The spatial domain is $[x_L, x_R] = [0, 2]$ and the time interval is $[0, T] = [0, 1]$. The initial condition $u(x, 0)$ is the following Gaussian pulse

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

and the diffusion coefficients are

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

For satisfying (14), we choose $M = \lfloor (2N+2)^\alpha/2 \rfloor$ such that $\Delta t \approx 2\Delta x^\alpha$. Thus, the super-linear convergence rate by the PCGNR method with the proposed circulant preconditioner is guaranteed as both (14) and (15) hold. We remark that the linear convergence rate by the multigrid method in [22] has not been proven theoretically.

The numerical results of Example 1 are shown in Table 1. It shows that both the average number of iterations and the CPU time by the PCGNR method with circulant preconditioners are much less than those by the CGNR method. It also shows that the CPU times by the PCGNR methods are less than that by the multigrid method.

Example 2. In this example, we study the case for which the diffusion coefficients are variable. We solve the FDE problem (1) of order α with the following data. The spatial domain is $[x_L, x_R] = [0, 2]$ and the time interval is $[0, T] = [0, 1]$. The left and right diffusion coefficients are

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

respectively. The source term is

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

and the initial condition is

$$u(x,0) = 4x^2(2-x)^2.$$

The exact solution of this problem is

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

Table 1: Comparisons for solving Example 1 by the CGNR method, the PCGNR method with preconditioners $c(\mathcal{M}^{(m)})$ and $\mathcal{S}^{(m)}$, and the multigrid method for $\alpha = 1.2$, 1.5, and 1.8. Here $\Delta t \approx 2\Delta x^{\alpha}$.

			$\mathcal{I}$		$\mathcal{C}$		$\mathcal{S}$		MGM	
α	$N+1$	M	Iter	CPU(s)	Iter	CPU(s)	Iter	CPU(s)	Iter	CPU(s)
1.2	2^6	32	37.6	0.10	6.0	0.05	5.8	0.04	10.1	0.30
	2^7	74	34.4	0.36	6.0	0.12	5.3	0.11	8.8	0.65
	2^8	169	31.4	0.96	5.0	0.28	5.0	0.28	7.3	1.56
	2^9	388	28.5	4.08	5.0	1.29	5.0	1.27	6.3	3.93
	2^{10}	891	25.7	15.22	5.0	4.49	5.0	4.44	5.5	10.54
1.5	2^6	91	40.9	0.30	6.0	0.11	5.6	0.10	7.0	0.52
	2^7	256	39.2	1.40	6.0	0.43	5.2	0.37	6.1	1.59
	2^8	724	35.8	4.70	5.4	1.28	5.0	1.20	5.6	5.25
	2^9	2048	32.3	24.21	5.0	6.74	5.0	6.73	5.1	16.91
	2^{10}	5793	29.0	110.65	5.0	28.92	5.0	28.93	4.8	60.20
1.8	2^6	256	42.6	0.85	7.0	0.31	5.8	0.26	6.1	1.25
	2^7	891	41.0	5.01	6.0	1.45	5.5	1.36	6.0	5.50
	2^8	3104	36.3	20.59	6.0	5.92	5.3	5.38	5.0	19.93
	2^9	10809	31.8	126.14	5.2	36.55	5.1	35.88	5.0	88.35
	2^{10}	37641	27.5	685.80	5.0	187.06	5.0	187.07	4.1	339.35

for any $\alpha \in (1, 2)$. With the exact solution, we calculate the exact error of the numerical solution. In the following tests, we consider $\alpha = 1.2$, 1.5, and 1.8. We remark that the same example in [19, 22, 31] is only considered for $\alpha = 1.8$. In the numerical test, we naturally choose $\Delta t = \Delta x$.

Table 2 reports the history for numerically solving Example 2. Note that the average number of iterations for the CGNR method is increasing as N is increasing rapidly, while the average numbers of iterations by the PCGNR methods almost keep constant, and therefore their CPU times are much less than that by the CGNR method. We remark that both diffusion coefficients are not constant and $\nu_{N,M}$ tends to 0. Therefore, both assumptions (14) and (15) are not satisfied and hence the superlinear convergence rate is not theoretically guaranteed. However, from the point of view on CPU time, the performance of the proposed circulant preconditioner still works well and is the best among all the methods.

6. Concluding Remarks

In this paper, we have employed the PCGNR method with a circulant preconditioner to solve the discretized linear systems of (1). Theoretically, we prove the superlinear convergence rate of the proposed method under the conditions that the diffusion coefficients $d_{\pm}(x, t)$ are constant and the ratio $\frac{\Delta x^{\alpha}}{\Delta t}$ is bounded away from zero. Numerical tests have shown the efficiency of the PCGNR method. Moreover, for variable diffusion coefficients in the FDE (1), the proposed algorithm still works very well.

In our future consideration, more effective preconditioners will be developed to solve the two or more dimensional FDE problem [30] and the steady-state fractional partial differential

Table 2: Comparisons for solving Example 2 by the CGNR method, the PCGNR method with preconditioners $c(\mathcal{M}^{(m)})$ and $\mathcal{S}^{(m)}$, and the multigrid method for $\alpha = 1.2$, 1.5 and 1.8.

				$\mathcal{I}$		$\mathcal{C}$		$\mathcal{S}$		MGM	
α	$N+1$	M	Error	Iter	CPU(s)	Iter	CPU(s)	Iter	CPU(s)	Iter	CPU(s)
1.2	2^6	2^5	3.1501e-2	33.8	0.08	8.0	0.05	8.0	0.05	8.0	0.19
	2^7	2^6	1.5983e-2	65.5	0.53	8.0	0.12	8.0	0.12	7.1	0.41
	2^8	2^7	8.0488e-3	82.0	1.70	8.0	0.29	7.0	0.26	7.1	1.03
	2^9	2^8	4.0377e-3	89.9	7.92	8.0	1.18	8.0	1.18	7.0	2.60
	2^{10}	2^9	2.0214e-3	96.0	30.90	7.0	3.32	8.0	3.73	7.0	6.91
1.5	2^6	2^5	2.2529e-2	46.6	0.11	10.0	0.06	8.0	0.05	8.0	0.19
	2^7	2^6	1.1164e-2	111.6	0.87	10.4	0.16	9.0	0.14	7.0	0.41
	2^8	2^7	5.5563e-3	264.6	5.34	10.9	0.38	9.3	0.33	7.0	1.03
	2^9	2^8	2.7721e-3	568.3	48.74	9.9	1.44	9.9	1.44	7.0	2.62
	2^{10}	2^9	1.3838e-3	903.7	288.51	11.0	5.01	10.0	4.52	6.0	5.93
1.8	2^6	2^5	1.7434e-2	70.6	0.16	16.0	0.08	13.0	0.07	7.0	0.16
	2^7	2^6	8.3524e-3	202.0	1.58	18.0	0.26	14.0	0.21	7.0	0.40
	2^8	2^7	4.0838e-3	587.2	11.68	18.9	0.63	14.0	0.48	7.0	1.01
	2^9	2^8	2.0186e-3	1703.0	146.05	21.0	2.89	14.0	1.97	7.0	2.60
	2^{10}	2^9	1.0035e-3	4900.0	1541.66	20.0	8.66	13.0	5.78	6.0	5.92

equations.

Acknowledgements

The authors would like to thank Dr. Hong-kui Pang for his helpful discussions. We are also grateful to the anonymous referees for the useful suggestions and comments that improved the presentation of this paper.

References

[1] J. Bai and X. Feng, *Fractional-order anisotropic diffusion for image denoising*, IEEE Tran. Image Proc., 16 (2007), pp. 2492–2502.

[2] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1994.

[3] D. Benson, S. W. Wheatcraft, and M. M. Meerschaert, *Application of a fractional advection-dispersion equation*, Water Resour. Res. 36 (2000), pp.1403–1413.

[4] D. Benson, S. W. Wheatcraft, and M. M. Meerschaert, *The fractional-order governing equation of Lévy motion*, Water Resour. Res. 36 (2000), pp. 1413–1423.

[5] B. Beumer, M. Kovács, and M. M. Meerschaert, *Numerical solutions for fractional reaction-diffusion equations*, Comput. Math. Appl., 55 (2008), pp. 2212–2226.

[6] B. A. Carreras, V. E. Lynch, and G. M. Zaslavsky, *Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence models*, Phys. Plasma 8 (2001), pp. 5096–5103.

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

[8] R. Chan and M. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

[9] R. Chan and G. Strang, *Toeplitz equations by conjugate gradients with circulant preconditioner*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 104–119.

[10] T. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.

[11] W. Deng, *Finite element method for the space and time fractional Fokker-Planck equation*, SIAM J. Numer. Anal., 47 (2008), pp. 204–226.

[12] V. J. Ervin, N. Heuer, and J. P. Roop, *Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation*, SIAM J. Numer. Anal., 45 (2007), pp. 572–591.

[13] X. Jin, *Preconditioning Techniques for Teoplitz Systems*, Higher Education Press, Beijing, 2010.

[14] T. A. M. Langlands and B. I. Henry, *The accuracy and stability of an implicit solution method for the fractional diffusion equation*, J. Comput. Phys. 205 (2005), pp. 719–736.

[15] F. Liu, V. Anh, and I. Turner, *Numerical solution of the space fractional Fokker-Planck equation*, J. Comput. Appl. Math., 166 (2004), pp. 209–219.

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

[17] M. M. Meerschaert, H. P. Scheffler, and C. Tadjeran, *Finite difference methods for two-dimensional fractional dispersion equation*, J. Comput. Phys., 211 (2006), pp. 249–261.

[18] M. M. Meerschaert and C. Tadjeran, *Finite difference approximations for fractional advection-dispersion flow equations*, J. Comput. Appl. Math., 172 (2004), pp. 65–77.

[19] M. M. Meerschaert and C. Tadjeran, *Finite difference approximations for two-sided space-fractional partial differential equations*, Appl. Numer. Math., 56 (2006), pp. 80–90.

[20] D. A. Murio, *Implicit finite difference approximation for time fractional diffusion equations*, Comput. Math. Appl., 56 (2008), pp. 1138–1145.

[21] M. Ng, *Iterative Methods for Toeplitz Systems*, Oxford, 2004.

[22] H. Pang and H. Sun, *Multigrid method for fractional diffusion equations*, J. Comput. Phys., 231 (2012), pp. 693–703.

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

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

[25] M. F. Shlesinger, B. J. West, and J. Klafter, *Lévy dynamics of enhanced diffusion: application to turbulence*, Phys. Rev. Lett., 58 (1987), pp. 1100–1103.

[26] I. M. Sokolov, J. Klafter, and A. Blumen, *Fractional kinetics*, Phys. Today Nov., (2002), pp. 28–53.

[27] E. Sousa, *Finite difference approximates for a fractional advection diffusion problem*, J. Comput. Phys., 228 (2009), pp. 4038–4054.

[28] L. Su, W. Wang, and Z. Yang, *Finite difference approximations for the fractional advection-diffusion equation*, Phys. Lett. A, 373 (2009), pp. 4405–4408.

[29] C. Tadjeran, M. M. Meerschaert, and H. P. Scheffler, *A second-order accurate numerical approximation for the fractional diffusion equation*, J. Comput. Phys., 213 (2006), pp. 205–213.

[30] H. Wang and K. Wang, *An $\mathcal{O}(N \log^2 N)$ alternating-direction finite difference method for two-dimensional fractional diffusion equations*, J. Comput. Phys., 230 (2011). pp. 7830–7839.

[31] H. Wang, K. Wang, and T. Sircar, *A direct $O(N \log^2 N)$ finite difference method for fractional diffusion equations*, J. Comput. Phys., 229 (2010), pp. 8095–8104.

[32] K. Wang and H. Wang, *A fast characteristic finite difference method for fractional advection-diffusion equations*, Adv. Water Resour., 34 (2011), pp. 810–816.

[33] G. M. Zaslavsky, D. Stevens, and H. Weitzner, *Self-similar transport in incomplete chaos*, Phys. Rev. E, 48 (1993), pp. 1683–1694.