Journal of Computational Physics 336 (2017) 69–86

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A multigrid method for linear systems arising from time-dependent two-dimensional space-fractional diffusion equations ☆

Xue-lei Lin [a], Michael K. Ng [b], Hai-Wei Sun [a,*]

[a] Department of Mathematics, University of Macau, Macao
[b] Department of Mathematics, Hong Kong Baptist University, Hong Kong

ARTICLE INFO

Article history:
Received 21 September 2016
Received in revised form 6 December 2016
Accepted 2 February 2017
Available online 6 February 2017

Keywords:
Fractional diffusion equation
Multigrid method
Banded-splitting smoother
Non-rectangular domain

ABSTRACT

In this paper, we study a V-cycle multigrid method for linear systems arising from time-dependent two-dimensional space-fractional diffusion equations. The coefficient matrices of the linear systems are structured such that their matrix-vector multiplications can be computed efficiently. The main advantage using the multigrid method is to handle the space-fractional diffusion equations on non-rectangular domains, and to solve the linear systems with non-constant coefficients more effectively. The main idea of the proposed multigrid method is to employ two banded splitting iteration schemes as pre-smoother and post-smoother. The pre-smoother and the post-smoother take banded splitting of the coefficient matrix under the x-dominant ordering and the y-dominant ordering, respectively. We prove the convergence of the proposed two banded splitting iteration schemes in the sense of infinity norm. Results of numerical experiments for time-dependent two-dimensional space-fractional diffusion equations on rectangular, L-shape and U-shape domains are reported to demonstrate that both computational time and iteration number required by the proposed method are significantly smaller than those of the other tested methods.

1. Introduction

In this paper, we consider the initial–boundary value problem of the two-dimensional space-fractional diffusion equation (SFDE):

$$\frac{\partial u(x,y,t)}{\partial t} = d_+(x,y,t)\,{}_aD_x^{\alpha}u(x,y,t) + d_-(x,y,t)\,{}_xD_{b(y)}^{\alpha}u(x,y,t) + e_+(x,y,t)\,{}_cD_y^{\beta}u(x,y,t) + e_-(x,y,t)\,{}_yD_{d(x)}^{\beta}u(x,y,t) + f(x,y,t), \quad (x,y,t)\in\Omega\times(0,T], \tag{1.1}$$

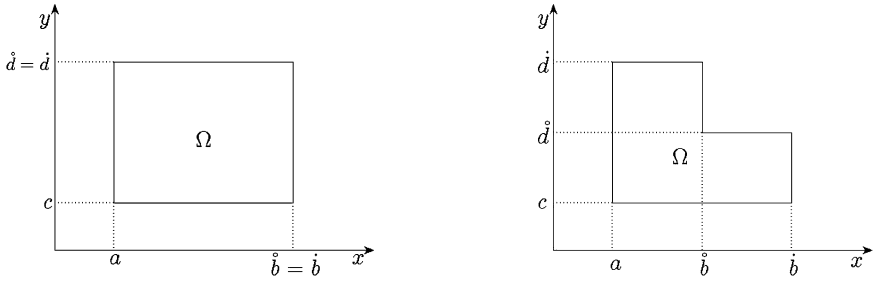
$$u(x,y,t) = 0, \quad (x,y,t)\in\partial\Omega\times(0,T], \tag{1.2}$$

$$u(x,y,0) = \psi(x,y), \quad (x,y)\in\bar{\Omega}, \tag{1.3}$$

☆ This research was supported by research grants MYRG2016-00063-FST from University of Macau, 054/2015/A2 from FDCT of Macao, HKRGC GRF HKBU12301214, HKBU FRG2/15-16/064, and 11271238 from NSF of China.

* Corresponding author.
E-mail address: HSun@umac.mo (H.-W. Sun).

http://dx.doi.org/10.1016/j.jcp.2017.02.008
0021-9991/



(a) $\mathring{b} = \dot{b}$ and $\mathring{d} = \dot{d}$ (b) $\mathring{b} \neq \dot{b}$ and $\mathring{d} \neq \dot{d}$

Fig. 1. Shapes of Ω in different cases.

where $\alpha, \beta \in (1,2)$, $d_\pm(x,y,t)$ and $e_\pm(x,y,t)$ are all nonnegative coefficients over the domain $\Omega \times (0,T]$ holding $d_+ + d_- > 0$ and $e_+ + e_- > 0$, $f(x,y,t)$ is the source term, ψ is a given initial condition, $\bar{\Omega} = [a,\dot{b}] \times [c,\dot{d}]) \setminus ([\mathring{b},\dot{b}] \times [\mathring{d},\dot{d}])$, $\partial\Omega$ denotes boundary of $\bar{\Omega}$, Ω denotes interior of $\bar{\Omega}$, $\dot{d} \geq \mathring{d} > c$, $\dot{b} \geq \mathring{b} > a$,

$$b(y) = \begin{cases} \dot{b}, & y \in (c,\mathring{d}), \\ \mathring{b}, & y \in [\mathring{d},\dot{d}), \end{cases} \qquad d(x) = \begin{cases} \dot{d}, & x \in (a,\mathring{b}), \\ \mathring{d}, & x \in [\mathring{b},\dot{b}). \end{cases}$$

Fig. 1(a) shows that Ω is a rectangular domain in the case of $\dot{b} = \mathring{b}$ and $\dot{d} = \mathring{d}$. Fig. 1(b) shows that Ω is an L-shape domain in the case of $\dot{b} \neq \mathring{b}$ and $\dot{d} \neq \mathring{d}$.

For a function $v(x)$ with compact support on an interval $[x_L, x_R]$, the left-sided and the right-sided Riemann–Liouville fractional derivatives of $v(x)$ are defined respectively by

$$ {}_{x_L}D_x^{\gamma} v(x) := \frac{1}{\Gamma(2-\gamma)} \frac{d^2}{dx^2} \int_{x_L}^{x} \frac{v(\xi)}{(x-\xi)^{\gamma-1}} d\xi, $$

$$ {}_{x}D_{x_R}^{\gamma} v(x) := \frac{1}{\Gamma(2-\gamma)} \frac{d^2}{dx^2} \int_{x}^{x_R} \frac{v(\xi)}{(\xi-x)^{\gamma-1}} d\xi. $$

The SFDE is a class of fractional differential equations which has been widely and successfully used in modeling of anomalous diffusive systems, unification of diffusion, description of fractional random walk and wave propagation phenomenon in the last few decades [1,2,8,13,16,17]. Since analytical solutions of SFDEs are often inaccessible, many numerical schemes have been proposed to solve the SFDEs [7,10–12,18,20]. Nevertheless, the fractional differential operators are non-local, for which discretization schemes tend to generate dense coefficient matrices. Hence, direct solvers like Gaussian elimination for solving the dense linear systems resulting from discretization of the SFDEs require $\mathcal{O}(NM^6)$ operations and $\mathcal{O}(NM^2 + M^4)$ storage, provided that N and M^2 are the numbers of temporal-grid points and spatial-grid points, respectively. Such an expensive complexity motivates us to develop fast algorithms for solving discretized SFDEs.

For uniform-grid discretization of the SFDE on rectangular domain, the associated dense coefficient matrix in each temporal step usually possesses block Toeplitz-like (BTL) structure whose matrix-vector multiplication can be fast computed via the fast Fourier transformation (FFT) with only $\mathcal{O}(M^2 \log M)$ operations and $\mathcal{O}(M^2)$ storage. For the BTL linear system, there are a series of fast solvers proposed to solve it. For example, the generalized minimum residual (GMRES) methods with row approximation preconditioner in [15] and banded preconditioner in [9] are both efficient ones for solving the BTL linear systems resulting from uniform-grid discretization of the SFDEs even in the case of oscillating coefficients.

For uniform-grid discretization of the SFDE on non-rectangular domain (e.g., L-shape domain), the associated dense coefficient matrix in each temporal step is no longer BTL but a block matrix with each block being BTL. A matrix with such a structure still allows a fast matrix-vector multiplication whose operations cost and storage requirement are still of $\mathcal{O}(M^2 \log M)$ and $\mathcal{O}(M^2)$, respectively. Although the above mentioned row approximation preconditioner and banded preconditioner can be extended to solving such linear systems by some variance, the GMRES method with these extended preconditioners converges more slowly for these non-BTL linear systems than it does for the BTL linear systems.

The main aim of this paper is to study a V-cycle multigrid method for solving the linear systems arising from time-dependent two-dimensional SFDEs. The main advantage of using multigrid method is to handle SFDEs on non-rectangular domains and to solve SFDEs with oscillating coefficients more efficiently than the above mentioned solvers. In [14], Pang and Sun proposed a multigrid method with the damped Jacobi smoother to solve one-dimensional discretized SFDEs. However, it keeps uncertain if their method can be extended to handle the two-dimensional SFDEs, specially to the non-rectangular domains. In this paper, we propose a V-cycle multigrid method with banded splitting iteration schemes to solve the two-dimensional SFDEs in non-rectangular domains. To our knowledge, although multigrid methods for solving integer-order partial differential equations on non-rectangle domains have been thoroughly investigated (see, for instance, [21]), the counterpart for SFDEs with non-constant coefficients has never been studied before. In the proposed V-cycle multigrid method, the pre-smoothing iteration takes a banded splitting of the coefficient matrix under x-dominant ordering, while y-dominant ordering for the post-smoother. Thus, in each pre-smoothing iteration, it requires to solve a banded linear system while in each post-smoothing iteration, it requires to solve a permuted banded linear system. That means, the operations cost and the storage requirement of each pre- or post-smoothing iteration are of $\mathcal{O}(M^2 \log M)$ and $\mathcal{O}(M^2)$, respectively, which are actually dominated by operations cost and storage requirement of one matrix-vector multiplication of the coefficient matrix. Theoretically, we show the convergence of the proposed two banded splitting iteration schemes in the sense of infinity norm. Results of numerical experiments for time-dependent two-dimensional SFDEs on rectangular, L-shape and U-shape domains are reported to demonstrate that both computational time and iteration number required by the proposed multigrid method are significantly smaller than those of the other tested methods.

The rest of this paper is organized as follows. In Section 2, we study coefficient matrices arising from the uniform-grid discretizations of time-dependent two-dimensional SFDEs on rectangular and non-rectangular domains. In Section 3, we present a multigrid method using the proposed banded smoother for solving the linear systems deriving from Section 2. In Section 4, we prove the convergence of the proposed two banded splitting iteration schemes. In Section 5, numerical results are reported to show the effectiveness of the proposed multigrid method. Finally, some concluding remarks are given in Section 6.

2. Discretized time-dependent two-dimensional SFDEs

2.1. Discretized SFDE on rectangular domain

In this subsection, we discuss the discrete form of the SFDE (1.1)–(1.3) on rectangular domain (i.e., $\dot{b} = \mathring{b}$ and $\dot{d} = \mathring{d}$).

Let N and M be positive integers. Denote by $\tau = T/N$, $h_x = (\mathring{b} - a)/(M+1)$ and $h_y = (\mathring{d} - c)/(M+1)$, the temporal step, spatial step in x direction and spatial step in y direction, respectively. Define the temporal girds, spatial grids in x-direction and spatial grids in y-direction, by $\{t_n = n\tau | 0 \le n \le N\}$, $\{x_i = a + ih_x | 0 \le i \le M+1\}$ and $\{y_j = c + jh_y | 0 \le j \le M+1\}$, respectively. Then, the vectors consisting of spatial-grid points with x-dominant ordering and y-dominant ordering are defined respectively by

$$\mathscr{P}_{R,x,M} = (P_{11}, P_{21}, \ldots, P_{M1}, P_{12}, P_{22}, \ldots, P_{M2}, \ldots\ldots, P_{1M}, P_{2M}, \ldots, P_{MM})^{\mathrm{T}} \in \mathbb{T}^{M^2 \times 1}, \tag{2.1}$$

$$\mathscr{P}_{R,y,M} = (P_{11}, P_{12}, \ldots, P_{1M}, P_{21}, P_{22}, \ldots, P_{2M}, \ldots\ldots, P_{M1}, P_{M2}, \ldots, P_{MM})^{\mathrm{T}} \in \mathbb{T}^{M^2 \times 1}, \tag{2.2}$$

where $\mathbb{T}^{m \times n}$ denotes the set of all $m \times n$ matrices with entries belonging to the two-dimensional Euclidean space, P_{ij} denotes the point (x_i, y_j) for $0 \le i, j \le M+1$.

Let $v(z)$ be a smooth function with compact support on $\in [z_L, z_R]$. Let $\gamma \in (1, 2)$. For simplicity, we assume the approximation of the fractional derivatives ${}_{z_L}D_z^{\gamma} v(z)$ and ${}_zD_{z_R}^{\gamma} v(z)$ to be following shifted numerical-integration formulas

$${}_{x_L}D_x^{\gamma} v(z)|_{z=z_L+ih} \approx \tilde{v}_{+,i} := -\frac{1}{h^{\gamma}} \sum_{j=1}^{i+1} g_{i-j+1}^{(\gamma)} v(z_L + jh), \quad 1 \le i \le K, \tag{2.3}$$

$${}_xD_{x_R}^{\gamma} v(z)|_{z=z_L+ih} \approx \tilde{v}_{-,i} := -\frac{1}{h^{\gamma}} \sum_{j=i-1}^{K} g_{j-i+1}^{(\gamma)} v(z_L + jh), \ 1 \le i \le K, \tag{2.4}$$

where $h = (z_R - z_L)/(K+1)$, $g_i^{(\gamma)} (0 \le i \le K)$ are real numbers varying from different discretization schemes. Actually, there are a series of discretization schemes fitting in the forms of (2.3) and (2.4); see for instance [12,18,20]. Denote $\tilde{\mathbf{v}}_{\pm} = (\tilde{v}_{\pm,1}, \tilde{v}_{\pm,2}, \ldots, \tilde{v}_{\pm,K})^{\mathrm{T}}$, $\mathbf{v} = (v_1, v_2, \ldots, v_K)^{\mathrm{T}}$. Then, matrical forms of (2.3) and (2.4) are

$$\tilde{\mathbf{v}}_{+} = -\frac{1}{h^{\gamma}} \mathbf{G}_{\gamma,K} \mathbf{v} \quad \text{and} \quad \tilde{\mathbf{v}}_{-} = -\frac{1}{h^{\gamma}} \mathbf{G}_{\gamma,K}^{\mathrm{T}} \mathbf{v},$$

respectively, provided that $\mathbf{G}_{\gamma,K}$ is a Toeplitz matrix [5] with its first column and its first row being

$$(g_1^{(\gamma)}, g_2^{(\gamma)}, \ldots, g_K^{(\gamma)})^{\mathrm{T}} \in \mathbb{R}^{K \times 1} \quad \text{and} \quad (g_1^{(\gamma)}, g_0^{(\gamma)}, 0, \ldots, 0) \in \mathbb{R}^{1 \times K},$$

respectively, where $\mathbb{R}^{m \times n}$ denotes the set of all $m \times n$ real matrices. Since $\mathbf{G}_{\gamma,K}$ is a Toeplitz matrix, its matrix-vector multiplication can be fast computed via using FFT with only $\mathcal{O}(K)$ storage and only $\mathcal{O}(K \log K)$ operations [5]. Moreover, the forward difference is used to approximate the temporal derivative $\frac{\partial u}{\partial t}$ throughout this paper. Then, we obtain an implicit finite difference discretization of the SFDE (1.1)–(1.3) on the uniform grids, $\mathscr{P}_{R,x,M}$, in the case of Ω being a rectangle as follows

$$\tau^{-1}\left(\mathbf{u}^{n+1}-\mathbf{u}^{n}\right)=-\left(h_x^{-\alpha}\mathbf{B}_{R,x}^{(n+1)}+h_y^{-\beta}\mathbf{B}_{R,y}^{(n+1)}\right)\mathbf{u}^{n+1}+\mathbf{f}^{n+1},\quad 0\le n\le N-1, \tag{2.5}$$

where $\mathbf{u}^n=u(\mathscr{P}_{R,x,M},t_n)$, $\mathbf{f}^n=f(\mathscr{P}_{R,x,M},t_n)$,

$$\mathbf{B}_{R,x}^{(n)}=\mathbf{D}_{R,+}^{(n)}\left(\mathbf{I}_M\otimes\mathbf{G}_{\alpha,M}\right)+\mathbf{D}_{R,-}^{(n)}(\mathbf{I}_M\otimes\mathbf{G}_{\alpha,M}^{\mathrm{T}}),\quad \mathbf{D}_{R,\pm}^{(n)}=\mathrm{diag}(d_\pm(\mathscr{P}_{R,x,M},t_n)),$$
$$\mathbf{B}_{R,y}^{(n)}=\mathbf{E}_{R,+}^{(n)}\left(\mathbf{G}_{\beta,M}\otimes\mathbf{I}_M\right)+\mathbf{E}_{R,-}^{(n)}(\mathbf{G}_{\beta,M}^{\mathrm{T}}\otimes\mathbf{I}_M),\quad \mathbf{E}_{R,\pm}^{(n)}=\mathrm{diag}(e_\pm(\mathscr{P}_{R,x,M},t_n)),$$

$\mathbf{I}_k$ denotes $k\times k$ identity matrix, "$\otimes$" denotes the Kronecker product.

The resulting task from (2.5) is to recursively solve

$$\left(\mathbf{I}_{M^2}+\eta_x\mathbf{B}_{R,x}^{(n+1)}+\eta_y\mathbf{B}_{R,y}^{(n+1)}\right)\mathbf{u}^{n+1}=\mathbf{u}^n+\tau\mathbf{f}^{n+1},\quad 0\le n\le N-1, \tag{2.6}$$

where $\eta_x=\tau h_x^{-\alpha}$, $\eta_y=\tau h_y^{-\beta}$. Since the coefficient matrices in (2.6) are all BTL, their matrix-vector multiplications can be fast computed with only $\mathcal{O}(M^2)$ storage and only $\mathcal{O}(M^2\log M)$ operations by utilizing FFT and properties of Kronecker product.

2.2. Discretized SFDE on L-shape domain

In this subsection, we discuss the discrete form of the SFDE (1.1)–(1.3) on L-shape domain (i.e., $\dot{b}\neq\mathring{b}$ and $\dot{d}\neq\mathring{d}$). For simplicity, we assume $\mathring{b}-a=\dot{b}-\mathring{b}$ and $\mathring{d}-c=\dot{d}-\mathring{d}$ throughout the rest of this paper. This subsection actually gives an insight into the structure of coefficient matrices of the discretized SFDE on L-shape domain, which is quite different from the classical BTL one resulting from the uniform-grid discretization of the SFDEs on rectangular domain.

Let N and M be positive integers. Denote by $\tau=T/N$, $h_x=(\mathring{b}-a)/(M+1)$ and $h_y=(\mathring{d}-c)/(M+1)$, the temporal step, spatial step in x-direction and spatial step in y-direction, respectively. The corresponding spatial-grid points in x-direction and spatial-grid points in y-direction are defined by $\{x_i=a+ih_x|0\le i\le\dot{M}+1\}$ and $\{y_j=c+jh_y|0\le j\le\dot{M}+1\}$, respectively, where $\dot{M}+1=2(M+1)$. Also, the vectors consisting of spatial-grid points with x-dominant ordering and y-dominant ordering are respectively defined by

$$\mathscr{P}_{L,x,M}=\left(\{P_{i1}\}_{i=1}^{m_1},\{P_{i2}\}_{i=1}^{m_2},\ldots,\{P_{i\dot{M}}\}_{i=1}^{m_{\dot{M}}}\right)\in\mathbb{T}^{\ddot{M}\times1}, \tag{2.7}$$
$$\mathscr{P}_{L,y,M}=\left(\{P_{1j}\}_{j=1}^{m_1},\{P_{2j}\}_{j=1}^{m_2},\ldots,\{P_{\dot{M}j}\}_{j=1}^{m_{\dot{M}}}\right)\in\mathbb{T}^{\ddot{M}\times1}, \tag{2.8}$$

where P_{ij} denotes the point (x_i,y_j) for $0\le i,j\le\dot{M}+1$, respectively, $\ddot{M}=2\dot{M}M-M^2$,

$$m_i=\begin{cases}\dot{M}, & 1\le i\le M,\\ M, & M<i\le\dot{M}.\end{cases}$$

By (2.3)–(2.4) and forward difference approximation of $\frac{\partial u}{\partial t}$, we obtain an implicit finite difference discretization of the SFDE (1.1)–(1.3) on the uniform grids, $\mathscr{P}_{L,x,M}$, in the case of Ω being an L-shape domain as follows

$$\tau^{-1}(\mathbf{u}^{n+1}-\mathbf{u}^n)=-\left(h_x^{-\alpha}\mathbf{B}_{L,x}^{(n+1)}+h_y^{-\beta}\mathbf{B}_{L,y}^{(n+1)}\right)\mathbf{u}^{n+1}+\mathbf{f}^{n+1},\qquad 0\le n\le N-1, \tag{2.9}$$

where $\mathbf{u}^n=u(\mathscr{P}_{L,x,M},t_n)$, $\mathbf{f}^n=f(\mathscr{P}_{L,x,M},t_n)$,

$$\mathbf{B}_{L,x}^{(n)}=\mathbf{D}_{L,+}^{(n)}\hat{\mathbf{B}}_{\alpha,M}+\mathbf{D}_{L,-}^{(n)}\hat{\mathbf{B}}_{\alpha,M}^{\mathrm{T}},\qquad \mathbf{D}_{L,\pm}^{(n)}=\mathrm{diag}(d_\pm(\mathscr{P}_{L,x,M},t_n)),$$
$$\mathbf{B}_{L,y}^{(n)}=\mathbf{E}_{L,+}^{(n)}\check{\mathbf{B}}_{\beta,M}+\mathbf{E}_{L,-}^{(n)}\check{\mathbf{B}}_{\beta,M}^{\mathrm{T}},\qquad \mathbf{E}_{L,\pm}^{(n)}=\mathrm{diag}(e_\pm(\mathscr{P}_{L,x,M},t_n)),$$
$$\hat{\mathbf{B}}_{\alpha,M}=\begin{bmatrix}\mathbf{I}_M\otimes\mathbf{G}_{\alpha,\dot{M}} & \\ & \mathbf{I}_{\bar{M}}\otimes\mathbf{G}_{\alpha,M}\end{bmatrix},\qquad \check{\mathbf{B}}_{\beta,M}=\begin{bmatrix}\mathbf{G}_{\beta,\dot{M}}^{(l,u)}\otimes\mathbf{I}_{\dot{M}} & \mathbf{G}_{\beta,\dot{M}}^{(r,u)}\otimes\tilde{\mathbf{I}}_M\\ \mathbf{G}_{\beta,\dot{M}}^{(l,d)}\otimes\tilde{\mathbf{I}}_M^{\mathrm{T}} & \mathbf{G}_{\beta,\dot{M}}^{(r,d)}\otimes\mathbf{I}_M\end{bmatrix},$$

$\tilde{\mathbf{I}}_M=[\mathbf{I}_M\quad\mathbf{O}_{M\times\bar{M}}]^{\mathrm{T}}$, $\bar{M}=\dot{M}-M$, $\mathbf{O}_{m\times n}$ denotes $m\times n$ zero matrix, $\mathbf{G}_{\beta,\dot{M}}^{(l,u)}\in\mathbb{R}^{M\times M}$, $\mathbf{G}_{\beta,\dot{M}}^{(r,u)}\in\mathbb{R}^{M\times\bar{M}}$, $\mathbf{G}_{\beta,\dot{M}}^{(l,d)}\in\mathbb{R}^{\bar{M}\times M}$ and $\mathbf{G}_{\beta,\dot{M}}^{(r,d)}\in\mathbb{R}^{\bar{M}\times\bar{M}}$ denote the partitions of the matrix $\mathbf{G}_{\beta,\dot{M}}$ such that

$$\mathbf{G}_{\beta,\dot{M}}=\begin{bmatrix}\mathbf{G}_{\beta,\dot{M}}^{(l,u)} & \mathbf{G}_{\beta,\dot{M}}^{(r,u)}\\ \mathbf{G}_{\beta,\dot{M}}^{(l,d)} & \mathbf{G}_{\beta,\dot{M}}^{(r,d)}\end{bmatrix}. \tag{2.10}$$

The resulting task from (2.9) is to recursively solve

$$\left(\mathbf{I}_{\ddot{M}}+\eta_x\mathbf{B}_{L,x}^{(n+1)}+\eta_y\mathbf{B}_{L,y}^{(n+1)}\right)\mathbf{u}^{n+1}=\mathbf{u}^n+\tau\mathbf{f}^{n+1},\quad 0\le n\le N-1, \tag{2.11}$$

where $\eta_x = \tau h_x^{-\alpha}$, $\eta_y = \tau h_y^{-\beta}$. Note that the coefficient matrices in (2.11) are all block matrices with each block being BTL. Thus, matrix-vector multiplications of coefficient matrices in (2.11) can also be fast computed with only $\mathcal{O}(M^2)$ storage and only $\mathcal{O}(M^2 \log M)$ operations.

3. Multigrid method

In this section, we propose a multigrid method (MGM) with two banded splitting iteration schemes as pre-smoother and post-smoother to solve the linear systems in (2.6) and (2.11). For convenience of statement, linear systems in (2.6) and (2.11) can be respectively simplified as

$$\mathbf{A}_R \mathbf{u}_R = \mathbf{b}_R, \tag{3.1}$$

$$\mathbf{A}_L \mathbf{u}_L = \mathbf{b}_L, \tag{3.2}$$

where $\mathbf{b}_R$ and $\mathbf{b}_L$ denote some given right hand sides resulting from (2.6) and (2.11), respectively,

$$\mathbf{A}_R = \mathbf{I}_{M^2} + \eta_x \mathbf{B}_{R,x} + \eta_y \mathbf{B}_{R,y}, \qquad \mathbf{A}_L = \mathbf{I}_{\ddot{M}} + \eta_x \mathbf{B}_{L,x} + \eta_y \mathbf{B}_{L,y},$$

$\mathbf{B}_{R,x}$ and $\mathbf{B}_{R,y}$ denote $\mathbf{B}_{R,x}^{(n)}$ and $\mathbf{B}_{R,y}^{(n)}$ in (2.6) for some n, respectively, $\mathbf{B}_{L,x}$ and $\mathbf{B}_{L,y}$ denote $\mathbf{B}_{L,x}^{(n)}$ and $\mathbf{B}_{L,y}^{(n)}$ in (2.11) for some n, respectively. For simplicity, in the rest of this section, we use $\mathbf{A}$ to denote $\mathbf{A}_R$ or $\mathbf{A}_L$. And we also use

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{3.3}$$

to denote the linear system (3.1) or (3.2).

Define a sequence of spatial-grids sizes such that $M_i = 2^i - 1,\ 2 \le i \le l$. Let $M = M_l$ for some $l > 2$. The corresponding x-direction spatial step and y-direction spatial step are given by $h_{x,i} = (\mathring{b} - a)/(M_i + 1)$ and $h_{y,i} = (\mathring{d} - c)/(M_i + 1)$, respectively, for $2 \le i \le l$. Let $\mathbf{A}_i$ denote $\mathbf{A}$ with $M = M_i$ for $2 \le i \le l$. For convenience, we assume $\mathbf{A}_i$ is of size $K_i \times K_i$ for $2 \le i \le l$. Denote by $\mathscr{S}_i$ and $\tilde{\mathscr{S}_i}$, the pre-smoothing iteration and the post-smoothing iteration at ith grid, respectively, for $3 \le i \le l$. Moreover, denote by $\mathbf{I}_{i+1}^{i} \in \mathbb{R}^{K_i \times K_{i+1}}$ and $\mathbf{I}_i^{i+1} \in \mathbb{R}^{K_{i+1} \times K_i}$, the restriction operator and the interpolation operator between ith and $(i+1)$th grids. Then, one iteration of V-cycle MGM for solving (3.3) is given by

---

Algorithm 1 One iteration of V-cycle MGM.

Set : $\mathbf{f}^h = \mathbf{b}$;
function $\mathbf{u}^h = \text{MGM}(i, \mathbf{u}_0, \mathbf{f}^h, \nu)$
if $i == 2$ **then**
  $\mathbf{u}^h = \mathbf{A}_i^{-1}\mathbf{f}^h$;
  return $\mathbf{u}^h$;
else
  iterate $\mathbf{u}^h = \mathscr{S}_i(\mathbf{u}^h, \mathbf{f}^h)$ ν times with initial guess $\mathbf{u}_0$; *%pre-smoothing iteration*
  $\mathbf{e} = \text{MGM}(i-1, \mathbf{0}, \mathbf{I}_i^{i-1}(\mathbf{f}^h - \mathbf{A}_i\mathbf{u}^h), \nu)$; % $\mathbf{0}$ *denotes zero initial guess*
  $\mathbf{u}^h = \mathbf{u}^h + \mathbf{I}_{i-1}^{i}\mathbf{e}$;*%correction*
  iterate $\mathbf{u}^h = \tilde{\mathscr{S}_i}(\mathbf{u}^h, \mathbf{f}^h)$ ν times; *%post-smoothing iteration*
  return $\mathbf{u}^h$;
end if
end

Of particular interest in this paper is to propose two banded splitting iteration schemes as $\mathscr{S}_i$ and $\tilde{\mathscr{S}_i}$. Denote $\mathbf{A}_i = [a_{jk}^{(i)}]_{j,k=1}^{K_i}$ for $3 \le i \le l$. Split $\mathbf{A}_i$ as $\mathbf{A}_i = \mathbf{D}_i - \mathbf{R}_i$ for $3 \le i \le l$, where the banded matrix $\mathbf{D}_i = [d_{jk}^{(i)}]_{j,k=1}^{K_i}$ with bandwidth ω is the banded truncation of $\mathbf{A}_i$ such that

$$d_{jk}^{(i)} = \begin{cases} a_{jk}^{(i)}, & |j-k| \le \omega, \\ 0, & |j-k| > \omega \end{cases}.$$

Here, the bandwidth ω is a sufficiently small positive constant integer. Then, for a linear system

$$\mathbf{A}_i \mathbf{x} = \mathbf{y}, \tag{3.4}$$

with a randomly given right hand side $\mathbf{y} \in \mathbb{R}^{K_i \times 1}$, one possible splitting is that

$$\mathbf{D}_i \mathbf{x} = \mathbf{R}_i \mathbf{x} + \mathbf{y},$$

which induces a banded iteration scheme for (3.4) as pre-smoother $\mathscr{S}_i$ such that

$$\mathbf{x}^{k+1} = \mathscr{S}_i(\mathbf{x}^k, \mathbf{y}) := \mathbf{D}_i^{-1}(\mathbf{R}_i\mathbf{x}^k + \mathbf{y}) = \mathbf{x}^k + \mathbf{D}_i^{-1}(\mathbf{y} - \mathbf{A}_i\mathbf{x}^k), \quad 3 \le i \le l, \tag{3.5}$$

with an initial guess $\mathbf{x}^k$.

From (2.6) and (2.11), we note that the discretization of both ${}_aD_x^\alpha$ and ${}_xD_{b(y)}^\alpha$ are localized in the block diagonal of $\mathbf{A}_i$. Thus, the banded matrix $\mathbf{D}_i$ characterizes the x-direction fractional derivatives, ${}_aD_x^\alpha$ and ${}_xD_{b(y)}^\alpha$ well. However, the discretization of both ${}_cD_y^\beta$ and ${}_yD_{d(x)}^\beta$ are dispersedly distributed in $\mathbf{A}_i$, for which the banded matrix $\mathbf{D}_i$ is insufficient to characterize. In order to remedy this situation, in the following, we introduce another banded iteration scheme as $\tilde{\mathscr{S}}_i$.

We study the linear systems (3.3) under a permuted ordering. Define permutation matrices $\mathbf{P}_{R,i}$ and $\mathbf{P}_{L,i}$ such that

$$\mathscr{P}_{R,y,M_i} = \mathbf{P}_{R,i}\mathscr{P}_{R,x,M_i}, \qquad \mathscr{P}_{L,y,M_i} = \mathbf{P}_{L,i}\mathscr{P}_{L,x,M_i}, \qquad 3 \le i \le l.$$

Also, we use $\mathbf{P}_i$ to denote $\mathbf{P}_{R,i}$ or $\mathbf{P}_{L,i}$. Then, it is easy to see that $\mathbf{P}_i$ is just a matrix transforming vectors from x-dominant ordering to y-dominant ordering. Let $\tilde{\mathbf{A}}_i = \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^{\mathrm{T}}$. Denote $\tilde{\mathbf{A}}_i = [\tilde{a}_{jk}^{(i)}]_{j,k=1}^{K_i}$. Similarly, split $\tilde{\mathbf{A}}_i$ as $\tilde{\mathbf{A}}_i = \tilde{\mathbf{D}}_i - \tilde{\mathbf{R}}_i$ for $3 \le i \le l$, where the banded matrix $\tilde{\mathbf{D}}_i = [\tilde{d}_{jk}^{(i)}]_{j,k=1}^{K_i}$ with bandwidth ω is the banded truncation of $\tilde{\mathbf{A}}_i$ such that

$$\tilde{d}_{jk}^{(i)} = \begin{cases} \tilde{a}_{jk}^{(i)}, & |j-k| \le \omega, \\ 0, & |j-k| > \omega. \end{cases}$$

Since $\tilde{\mathbf{A}}_i$ is the coefficient matrix of the discretized SFDE under the y-dominant ordering, similar to the discussion above, $\tilde{\mathbf{D}}_i$ characterizes the discretization of both ${}_cD_y^\beta$ and ${}_yD_{d(x)}^\beta$ well. On the other hand, linear system (3.4) is actually equivalent to

$$\tilde{\mathbf{A}}_i\tilde{\mathbf{x}} = \tilde{\mathbf{y}}, \quad 3 \le i \le l,$$

with $\tilde{\mathbf{x}} = \mathbf{P}_i\mathbf{x}$, $\tilde{\mathbf{y}} = \mathbf{P}_i\mathbf{y}$. Then, we obtain another splitting as follows

$$\tilde{\mathbf{D}}_i\tilde{\mathbf{x}} = \tilde{\mathbf{R}}_i\tilde{\mathbf{x}} + \tilde{\mathbf{y}}, \quad 3 \le i \le l,$$

which induces another banded splitting iteration scheme for (3.4) as post-smoother $\tilde{\mathscr{S}}_i$ such that

$$\begin{aligned}\mathbf{x}^{k+1} = \tilde{\mathscr{S}}_i(\mathbf{x}^k,\mathbf{y}) := \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{D}}_i^{-1}\left(\tilde{\mathbf{R}}_i\mathbf{P}_i\mathbf{x}^k + \mathbf{P}_i\mathbf{y}\right) &= \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{D}}_i^{-1}(\tilde{\mathbf{D}}_i - \tilde{\mathbf{A}}_i)\mathbf{P}_i\mathbf{x}^k + \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{D}}_i^{-1}\mathbf{P}_i\mathbf{y} \\ &= \mathbf{x}^k + \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{D}}_i^{-1}\mathbf{P}_i(\mathbf{y} - \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{A}}_i\mathbf{P}_i\mathbf{x}^k) \\ &= \mathbf{x}^k + \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{D}}_i^{-1}\mathbf{P}_i(\mathbf{y} - \mathbf{A}_i\mathbf{x}^k), \quad 3 \le i \le l, \end{aligned} \tag{3.6}$$

with an initial guess $\mathbf{x}^k$. Here, the construction of $\tilde{\mathbf{D}}_i$ requires only $\mathcal{O}(\omega K_i)$ operations and only $\mathcal{O}(\omega K_i)$ storage through the relationship $\tilde{\mathbf{A}}_i = \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^{\mathrm{T}}$ for $3 \le i \le l$, which is proportional to the number of unknowns in (3.4).

Remark. The positivity of the bandwidth ω is of great importance. Actually, if $\omega = 0$, then both $\mathscr{S}_i$ and $\tilde{\mathscr{S}}_i$ are exactly the classical Jacobi iteration. It is well known that MGM with Jacobi smoother usually converges very slowly or sometimes even diverges, which is also shown in numerical experiments in Section 5. What makes a difference is that when $\omega > 0$, the performance of MGM using the proposed banded smoother is significantly better than that of the MGM with the Jacobi smoother.

For the choice of restriction operator and interpolation operator, we refer to the typical piecewise linear restriction and piecewise linear interpolation. On the rectangular domain, the restriction ${}_R\mathbf{I}_{i+1}^i$ and the interpolation ${}_R\mathbf{I}_i^{i+1}$ are defined by

$${}_R\mathbf{I}_{i+1}^i = \mathbf{J}_{M_i} \otimes \mathbf{J}_{M_i}, \qquad {}_R\mathbf{I}_i^{i+1} = 4({}_R\mathbf{I}_{i+1}^i)^{\mathrm{T}}, \qquad 2 \le i \le l-1, \tag{3.7}$$

provided that for any positive integer k,

$$\mathbf{J}_k = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & \\ & & 1 & 2 & 1 & & \\ & & & & \ddots & & \\ & & & & 1 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{k\times(2k+1)}. \tag{3.8}$$

On the L-shape domain, the restriction ${}_L\mathbf{I}_{i+1}^i$ and the interpolation ${}_L\mathbf{I}_i^{i+1}$ are defined by

$${}_L\mathbf{I}_{i+1}^i = \begin{bmatrix} \mathbf{J}_{M_i} \otimes \hat{\mathbf{J}}_i & \\ & \check{\mathbf{J}}_i \otimes \mathbf{J}_{M_i} \end{bmatrix}, \quad {}_L\mathbf{I}_i^{i+1} = \begin{bmatrix} 4\mathbf{J}_{M_i}^{\mathrm{T}} \otimes \mathbf{J}_{M_{i+1}}^{\mathrm{T}} & \mathbf{J}_i^{(r,u)} \\ & \mathbf{J}_i^{(r,d)} \end{bmatrix}, \quad 2 \le i \le l-1, \tag{3.9}$$

where $\hat{\mathbf{J}}_i = \mathrm{blockdiag}(\mathbf{J}_{M_i}, 1, \mathbf{J}_{M_i})$, $\check{\mathbf{J}}_i = \mathrm{blockdiag}(1, \mathbf{J}_{M_i})$,

$$\mathbf{J}_i^{(r,d)} = \begin{bmatrix} 2 & \mathbf{0}_{1\times M_i} \\ \mathbf{e}_{M_{i+1},1} & 4\mathbf{J}_{M_i}^{\mathrm{T}} \end{bmatrix} \otimes \mathbf{J}_{M_i}^{\mathrm{T}}, \quad \mathbf{J}_i^{(r,u)} = \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \bar{\mathbf{J}}_i & \mathbf{O} \end{bmatrix}, \quad \bar{\mathbf{J}}_i = \begin{bmatrix} \mathbf{J}_{M_i}^{\mathrm{T}} \\ \mathbf{O}_{(M_{i+1}+1)\times M_i} \end{bmatrix},$$

where $\mathbf{e}_{M_{i+1},1}$ denotes the first column of $\mathbf{I}_{M_{i+1}}$, $\mathbf{O}$ denotes a zero matrix with proper size.

With components defined above, we furthermore define MGM(ν) iteration for solving the linear system (3.3) as follows:

Algorithm 2 MGM(ν) iteration.

Set : $\mathbf{r}_0 = \mathbf{b}$;
do
$\mathbf{u}_k = \text{MGM}(l, \mathbf{u}_{k-1}, \mathbf{b}, \nu)$; %$k \geq 1$, $\mathbf{u}_0$ *is an initial guess*
$\mathbf{r}_k = \mathbf{y} - \mathbf{A}\mathbf{u}_k$; *%compute current residual*
until $\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq 10^{-7}$ *%stopping criterion*

3.1. The complexity

We consider estimation of the complexity of Algorithm 1. Denote by $\mathcal{J}(M)$ and $\mathcal{R}(M)$, storage and operations required by Algorithm 1, respectively. Let J_i and $R_i(\nu)$ denote the storage and the operations required by Algorithm 1 at ith grid, respectively, for $2 \leq i \leq l$. Throughout the rest of this subsection, we denote by c, some positive constant independent of ν, $M_i (2 \leq i \leq l)$ and l.

For banded matrices, it is well known that the matrix-vector multiplications $\mathbf{D}_i^{-1}\mathbf{x}$ and $\tilde{\mathbf{D}}_i^{-1}\mathbf{x}$ can be fast computed with $\mathcal{O}(\omega^2 K_i)$ operations and $\mathcal{O}(\omega K_i)$ storage via LU factorization for a randomly given vector $\mathbf{x}$. Moreover, since $\mathbf{A}_i$ has the same structure as $\mathbf{A}$, it is BTL on rectangular domain while it is a block matrix with each block being BTL on L-shape domain. Thus, matrix-vector multiplication of $\mathbf{A}_i$ requires only $\mathcal{O}(K_i \log K_i)$ operations and only $\mathcal{O}(K_i)$ storage. Also, the permutation transformations in (3.6) require only $\mathcal{O}(K_i)$ trivial read–write operations and only $\mathcal{O}(K_i)$ storage. Moreover, by (2.1) and (2.7), we see that K_i is actually of $\mathcal{O}(M_i^2)$. Hence, we conclude that one iteration of both (3.5) and (3.6) require only $\mathcal{O}\left(M_i^2 \log M_i\right)$ operations and only $\mathcal{O}\left(M_i^2\right)$ storage. Also, since the restriction $\mathbf{I}_{i+1}^{i}$ and the interpolation $\mathbf{I}_i^{i+1}$ defined in both (3.7) and (3.9) are sparse matrices with $\mathcal{O}(M_i^2)$ non-zero elements, the operations cost and the storage requirement at ith grid are dominated by those of smoothing iteration at ith grid in Algorithm 1 when $i > 2$. Note also that we solve the linear system at the coarsest grid directly. Hence, we conclude that

$$J_2 \leq cM_2^4, \quad R_2(\nu) \leq cM_2^6, \quad J_i \leq cM_i^2, \quad R_i(\nu) \leq c\nu M_i^2 \log M_i, \quad 3 \leq i \leq l. \tag{3.10}$$

In addition, by nothing it holds that

$$M_i = 2^i - 1 < \frac{(2^{i+1} - 1)}{2} = \frac{M_{i+1}}{2} < \cdots < \frac{M_l}{2^{l-i}}. \tag{3.11}$$

As a result, (3.11) and (3.10) induce that

$$\mathcal{J}(M) = \sum_{i=2}^{l} J_i \leq cM_2^4 + c\sum_{i=3}^{l} M_i^2 \leq cM^2 \sum_{i=2}^{l} \frac{1}{2^{2(l-i)}} = \mathcal{O}(M^2),$$

$$\mathcal{R}(M) = \sum_{i=2}^{l} R_i(\nu) \leq cM_2^6 + c\nu \sum_{i=3}^{l} M_i^2 \log M_i \leq c\nu M^2 \log M \sum_{i=2}^{l} \frac{1}{2^{2(l-i)}} = \mathcal{O}(\nu M^2 \log M).$$

Hence, we conclude that storage requirement and the operations cost of Algorithm 1 are of $\mathcal{O}\left(M^2\right)$ and $\mathcal{O}(M^2 \log M)$ for fixed ν, respectively.

4. Convergence analysis

In this section, we firstly prove the convergence property of the pre-smoothing iteration (3.5) and the post-smoothing iteration (3.6). And then, we numerically verify the convergence of the two-grid method (TGM) associated with the MGM proposed in the previous section.

4.1. Convergence of pre-smoother and post-smoother

Denote by $\mathbf{S}_l = \mathbf{D}_l^{-1}\mathbf{R}_l$ and $\tilde{\mathbf{S}}_l = \mathbf{P}_l^{\mathrm{T}}\tilde{\mathbf{D}}_l^{-1}\tilde{\mathbf{R}}_l\mathbf{P}_l$, the iteration matrix of (3.5) and the iteration matrix of (3.6) for $l \geq 3$, respectively.

Definition 1. A matrix $\mathbf{C} = [c_{ij}]_{i,j=1}^{m} \in \mathbb{R}^{m\times m}$ is called diagonally dominant (DD), if it satisfies

$$|c_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{m} |c_{ij}|, \qquad 1 \leq i \leq m.$$

If the m inequalities are all strict, then $\mathbf{C}$ is called strictly diagonally dominant (SDD).

Lemma 1. *For a DD Toeplitz matrix* $\mathbf{W} \in \mathbb{R}^{m\times m}$, $\mathbf{W}^{\mathrm{T}}$ *is also DD.*

Proof. Denote $\mathbf{W} = [w_{ij}]_{i,j=1}^{m}$. Since $\mathbf{W}$ is Toeplitz matrix, its entries can be written as

$$z_{i-j} = w_{ij}, \qquad i, j \in \{1, 2..., m\}.$$

For any $j_0 \in \{1, 2, ..., m\}$, take $i_0 = m + 1 - j_0$. Then, $w_{j_0,j_0} = z_0 = w_{i_0,i_0}$ and

$$w_{ij_0} = z_{i-j_0} = z_{i_0-(m+1-i)} = w_{i_0,m+1-i}, \quad 1 \le i \le m. \tag{4.1}$$

(4.1) implies that for any $j_0 \in \{1, 2, ..., m\}$, there exists $i_0 \in \{1, 2, ..., m\}$ such that

$$|w_{j_0 j_0}| - \sum_{\substack{i=1\\ i\neq j_0}}^{m} |w_{ij_0}| = |w_{i_0 i_0}| - \sum_{\substack{j=1\\ j\neq i_0}}^{m} |w_{i_0 j}|. \tag{4.2}$$

Since $\mathbf{W}$ is DD, (4.2) induces that for any $j \in \{1, 2, ..., m\}$

$$|w_{jj}| - \sum_{\substack{i=1\\ i\neq j}}^{m} |w_{ij}| \ge 0.$$

That's to say $\mathbf{W}^{\mathrm{T}}$ is DD. □

Theorem 2. *Assume that both* $\mathbf{G}_{\alpha,K}$ *and* $\mathbf{G}_{\beta,K}$ *are DD with positive diagonal entries (DDPDE) for any* $K > 0$. *Then,*

$$\|\mathbf{S}_l\|_\infty < 1, \qquad \|\tilde{\mathbf{S}}_l\|_\infty < 1, \qquad \forall l \ge 3.$$

Proof. Let $\mathbf{A}_{R,l}$ and $\mathbf{A}_{L,l}$ denote $\mathbf{A}_R$ in (3.1) and $\mathbf{A}_L$ in (3.2) with $M = M_l$, respectively. Then,

$$\mathbf{A}_{R,l} = \mathbf{I}_{M_l^2} + \eta_{x,l}\mathbf{B}_{R,x,l} + \eta_{y,l}\mathbf{B}_{R,y,l}, \qquad \mathbf{A}_{L,l} = \mathbf{I}_{\ddot{M}_l} + \eta_{x,l}\mathbf{B}_{L,x,l} + \eta_{y,l}\mathbf{B}_{L,y,l},$$

where $\ddot{M}_l = 3M_l^2 + 2M_l$, $\eta_{x,l} = \tau h_{x,l}^{-\alpha}$, $\eta_{y,l} = \tau h_{y,l}^{-\beta}$

$$\mathbf{B}_{R,x,l} = \mathbf{D}_{R,+,l}\left(\mathbf{I}_{M_l} \otimes \mathbf{G}_{\alpha,M_l}\right) + \mathbf{D}_{R,-,l}(\mathbf{I}_{M_l} \otimes \mathbf{G}_{\alpha,M_l}^{\mathrm{T}}),$$
$$\mathbf{B}_{R,y,l} = \mathbf{E}_{R,+,l}\left(\mathbf{G}_{\beta,M_l} \otimes \mathbf{I}_{M_l}\right) + \mathbf{E}_{R,-,l}(\mathbf{G}_{\beta,M_l}^{\mathrm{T}} \otimes \mathbf{I}_{M_l}),$$
$$\mathbf{B}_{L,x,l} = \mathbf{D}_{L,+,l}\hat{\mathbf{B}}_{\alpha,M_l} + \mathbf{D}_{L,-,l}\hat{\mathbf{B}}_{\alpha,M_l}^{\mathrm{T}}, \qquad \mathbf{B}_{L,y,l} = \mathbf{E}_{L,+,l}\check{\mathbf{B}}_{\beta,M_l} + \mathbf{E}_{L,-,l}\check{\mathbf{B}}_{\beta,M_l}^{\mathrm{T}},$$

$\mathbf{D}_{R,\pm,l}$, $\mathbf{E}_{R,\pm,l}$, $\mathbf{D}_{L,\pm,l}$ and $\mathbf{E}_{L,\pm,l}$ denote $\mathrm{diag}(d_\pm(\mathscr{P}_{R,x,M_l}, t_n))$, $\mathrm{diag}(e_\pm(\mathscr{P}_{R,x,M_l}, t_n))$, $\mathrm{diag}(d_\pm(\mathscr{P}_{L,x,M_l}, t_n))$ and $\mathrm{diag}(e_\pm(\mathscr{P}_{L,x,M_l}, t_n))$ for some n, respectively. Note that both $\mathbf{G}_{\alpha,K}$ and $\mathbf{G}_{\beta,K}$ are DDPDE and Toeplitz matrices for any $K > 0$. By Lemma 1, both $\mathbf{G}_{\alpha,K}^{\mathrm{T}}$ and $\mathbf{G}_{\beta,K}^{\mathrm{T}}$ are also DDPDE. Note also that $d_\pm \ge 0$, $e_\pm \ge 0$, $d_+ + d_- > 0$ and $e_+ + e_- > 0$. Thus, we conclude that both $\mathbf{A}_{R,l}$ and $\mathbf{A}_{L,l}$ are DDPDE. Recall that we use $\mathbf{A}_l$ to denote $\mathbf{A}_{R,l}$ and $\mathbf{A}_{L,l}$. That's to say, $\mathbf{A}_l$ is DDPDE. Since $\mathbf{D}_l$ is in addition a banded truncation of $\mathbf{A}_l$, $\mathbf{D}_l$ is also DDPDE, which guarantees the invertibility of $\mathbf{D}_l$.

Recall that we assume $\mathbf{A}_l$ is of size $K_l \times K_l$. Let $\mathbb{C}^{m\times n}$ denote the set of all $m \times n$ complex matrices. For a vector $\mathbf{x} = (x_1, x_2, ..., x_{K_l})^{\mathrm{T}} \in \mathbb{C}^{K_l\times 1}$ with $\|\mathbf{x}\|_\infty = 1$, let $\mathbf{y} = (y_1, y_2, ..., y_{K_l})^{\mathrm{T}} = \mathbf{S}_l\mathbf{x}$. Then, $\mathbf{R}_l\mathbf{x} = \mathbf{D}_l\mathbf{y}$. Let $|y_m| = \|\mathbf{y}\|_\infty$ for some $m \in \{1, 2, ..., K_l\}$. Recall that the banded matrix $\mathbf{D}_l = [d_{jk}^{(l)}]_{j,k=1}^{K_l}$ has a bandwidth ω and $\mathbf{A}_l = \mathbf{D}_l - \mathbf{R}_l$. On the other hand, let $\mathbf{B}_{x,l}$ denote $\mathbf{B}_{R,x,l}$ or $\mathbf{B}_{L,x,l}$ and let $\mathbf{B}_{y,l}$ denote $\mathbf{B}_{R,y,l}$ or $\mathbf{B}_{L,y,l}$. Then, $\mathbf{A}_l$ can be written as $\mathbf{A}_l = \mathbf{I}_{K_l} + \eta_{x,l}\mathbf{B}_{x,l} + \eta_{y,l}\mathbf{B}_{y,l}$. Denote $\mathbf{R}_l = [r_{jk}^{(l)}]_{j,k=1}^{K_l}$, $\mathbf{B}_{x,l} = [b_{jk}^{(x,l)}]_{j,k=1}^{K_l}$ and $\mathbf{B}_{y,l} = [b_{jk}^{(y,l)}]_{j,k=1}^{K_l}$. Then, $\mathbf{D}_l\mathbf{y} = \mathbf{R}_l\mathbf{x}$ implies

$$|y_m|\left(d_{mm}^{(l)} - \sum_{1\le k\le K_l, 0<|k-m|\le\omega} |d_{mk}^{(l)}|\right) \le \left|\sum_{k=1}^{K_l} d_{mk}^{(l)} y_k\right| = \left|\sum_{1\le k\le K_l, |k-m|>\omega} r_{mk}^{(l)} x_k\right| \le \sum_{1\le k\le K_l, |k-m|>\omega} |r_{mk}^{(l)}|. \tag{4.3}$$

Note that $r_{jk}^{(l)} = \eta_{x,l} b_{jk}^{(x,l)} + \eta_{y,l} b_{jk}^{(y,l)}$ for $|j - k| > \omega$ and

$$d_{jk}^{(l)} = \begin{cases} 1 + \eta_{x,l} b_{jj}^{(x,l)} + \eta_{y,l} b_{jj}^{(y,l)}, & j = k, \\ \eta_{x,l} b_{jk}^{(x,l)} + \eta_{y,l} b_{jk}^{(y,l)}, & 0 < |j - k| \le \omega, \\ 0, & |j - k| > \omega. \end{cases}$$

Hence, by (4.3) and the fact that $\mathbf{B}_{x,l}$ and $\mathbf{B}_{y,l}$ are both DDPDE,

$$\|\mathbf{S}_l\mathbf{x}\|_\infty = |y_m| \le \frac{\sum\limits_{1\le k\le K_l, |k-m|>\omega} \left|\eta_{x,l} b_{mk}^{(x,l)} + \eta_{y,l} b_{mk}^{(y,l)}\right|}{1+\eta_{x,l} b_{mm}^{(x,l)} + \eta_{y,l} b_{mm}^{(y,l)} - \sum\limits_{1\le k\le K_l, 0<|k-m|\le\omega} \left|\eta_{x,l} b_{mk}^{(x,l)} + \eta_{y,l} b_{mk}^{(y,l)}\right|}$$
$$\le \frac{\sum\limits_{1\le k\le K_l, |k-m|>\omega} \left|\eta_{x,l} b_{mk}^{(x,l)} + \eta_{y,l} b_{mk}^{(y,l)}\right|}{1+\sum\limits_{1\le k\le K_l, |k-m|>\omega} \left|\eta_{x,l} b_{mk}^{(x,l)} + \eta_{y,l} b_{mk}^{(y,l)}\right|} < 1. \tag{4.4}$$

Since $\mathbf{x}$ is an arbitrary vector on the unit sphere, (4.4) actually induces

$$\|\mathbf{S}_l\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|\mathbf{S}_l\mathbf{x}\|_\infty < 1, \qquad \forall l \ge 3.$$

Since $\tilde{\mathbf{A}} = \mathbf{P}_l\mathbf{A}_l\mathbf{P}_l^{\mathrm{T}}$ is the coefficient matrix under y-dominant ordering of grid points, $\tilde{\mathbf{A}}_l$ has similar structure to $\mathbf{A}_l$. Hence, it similarly holds that $\|\tilde{\mathbf{D}}_l^{-1}\tilde{\mathbf{R}}_l\|_\infty < 1$, which induces that

$$\|\tilde{\mathbf{S}}_l\|_\infty = \|\mathbf{P}_l^{\mathrm{T}}\tilde{\mathbf{D}}_l^{-1}\tilde{\mathbf{R}}_l\mathbf{P}_l\|_\infty \le \|\mathbf{P}_l^{\mathrm{T}}\|_\infty\|\tilde{\mathbf{D}}_l^{-1}\tilde{\mathbf{R}}_l\|_\infty\|\mathbf{P}_l\|_\infty = \|\tilde{\mathbf{D}}_l^{-1}\tilde{\mathbf{R}}_l\|_\infty < 1, \quad l \ge 3.$$

The proof is complete. □

Remark. We remark that $\mathbf{G}_{\gamma,K}$ arising from several discretization schemes satisfies the assumption in Theorem 2. For instance, the first-order discretization scheme proposed in [12] holds those assumptions for $\gamma \in (1,2)$. The second-order discretization scheme proposed in [20] holds those assumptions for $\gamma \in [\frac{\sqrt{17}-1}{2}, 2)$. Also, the discretization scheme proposed in [18] holds those assumptions for $\gamma \in [\gamma^*, 2)$ where $\gamma^* \approx 1.5546$ is a solution of the equation $3^{3-\gamma} - 4\cdot 2^{3-\gamma} + 6 = 0$. Note that convergence of $\mathscr{S}_l$ and $\tilde{\mathscr{S}_l}$ guarantee that high frequency error components [3] can be reduced in each pre- or post-smoothing iteration, which is a necessary property of smoothers to lead to the convergence of MGM; see for instance [3]. Besides, the assumption in Theorem 2 guarantees the invertibility of $\mathbf{D}_l$, under which the banded smoother works. Thus, in the following, we only focus on those discretization schemes which satisfy the assumption in Theorem 2.

4.2. Verification for convergence of TGM

Let $\nu = 1$. Then, the iteration matrix of TGM is given by [6]

$$\mathbf{T}_l = \tilde{\mathbf{S}}_l(\mathbf{I}_{K_l} - \mathbf{I}_{l-1}^{l}\mathbf{A}_{l-1}^{-1}\mathbf{I}_l^{l-1})\mathbf{S}_l, \qquad l \ge 3.$$

In the following discussion, we verify the convergence of TGM numerically. That is, we compute $\|\mathbf{T}_l\|_2$ with different l directly. Set $\omega = \tau = d_+ = e_- \equiv 1$ and $d_- = e_+ \equiv 5$. Let $h_{x,i} = h_{y,i} = 1/(M_i + 1)$ for $l \ge 2$. We also assume $\dot{b} = \mathring{b}$ and $\dot{d} = \mathring{d}$ (i.e., the rectangular domain).

For the choice of the real numbers $g_k^{(\gamma)}$ in (2.3) and (2.4), we refer to both the first-order shifted Grünwald formula proposed in [12] and the second-order discretization scheme proposed in [20]. Coefficients of the first-order shifted Grünwald formula in [12] are given by

$$g_k^{(\gamma)} = -w_k^{(\gamma)}, \quad k \ge 0, \tag{4.5}$$

where

$$w_0^{(\gamma)} = 1, \quad w_k^{(\gamma)} = \left(1 - \frac{\gamma+1}{k}\right) w_{k-1}^{(\gamma)}, \quad k \ge 1. \tag{4.6}$$

Coefficients of the second-order discretization scheme proposed in [20] are given by

$$g_0^{(\gamma)} = -\frac{\gamma}{2} w_0^{(\gamma)}, \quad g_k^{(\gamma)} = \frac{\gamma-2}{2} w_{k-1}^{(\gamma)} - \frac{\gamma}{2} w_k^{(\gamma)}, \quad k \ge 1, \tag{4.7}$$

where $w_k^{(\gamma)} (k \ge 0)$ are given by (4.6).

As mentioned above, the second-order discretization scheme (4.7) satisfies the assumption in Theorem 2 only when $\gamma \in [\frac{\sqrt{17}-1}{2}, 2)$. Thus, we compare (4.5) and (4.7) only for the case $\alpha, \beta \in [\frac{\sqrt{17}-1}{2}, 2)$. The corresponding computational results are listed in Table 1 for the first-order discretization scheme (4.5) and in Table 2 for the second-order discretization scheme (4.7).

Table 1
Values of $\|\mathbf{T}_l\|_2$ with different l and (α,β) when the first-order discretization scheme (4.5) is used.

M_l+1	(α,β) with $\alpha=\beta$			(α,β) with $\alpha\neq\beta$		
	(1.8, 1.8)	(1.9, 1.9)	(1.99, 1.99)	(1.6, 1.7)	(1.6, 1.8)	(1.6, 1.9)
2^3	0.35	0.39	0.44	0.33	0.36	0.39
2^4	0.49	0.57	0.68	0.42	0.46	0.52
2^5	0.59	0.69	0.83	0.52	0.57	0.64
2^6	0.67	0.77	0.92	0.62	0.69	0.77

Table 2
Values of $\|\mathbf{T}_l\|_2$ with different l and (α,β) when the second-order discretization scheme (4.7) is used.

M_l+1	(α,β) with $\alpha=\beta$			(α,β) with $\alpha\neq\beta$		
	(1.8, 1.8)	(1.9, 1.9)	(1.99, 1.99)	(1.6, 1.7)	(1.6, 1.8)	(1.6, 1.9)
2^3	0.29	0.36	0.44	0.28	0.31	0.36
2^4	0.37	0.49	0.67	0.41	0.47	0.53
2^5	0.44	0.59	0.82	0.56	0.65	0.73
2^6	0.51	0.67	0.91	0.76	0.90	0.99

When $\alpha=\beta$, $\|\mathbf{T}_l\|_2$ in Table 2 is smaller than that in Table 1. That means, the TGM associated with the scheme (4.7) has a better convergence than the TGM associated with the scheme (4.5) for the case $\alpha=\beta$. Tables 1–2 also show that when α is away from β, $\|\mathbf{T}_l\|_2$ in Table 1 increases much slower than $\|\mathbf{T}_l\|_2$ in Table 2 as l increases. Especially, when M_l is large, $\|\mathbf{T}_l\|_2$ in Table 1 is smaller than $\|\mathbf{T}_l\|_2$ in Table 2 for the case $\alpha\neq\beta$. That implies when α is away from β, the TGM associated with the scheme (4.5) is more robust than the TGM associated with the scheme (4.7) when the size of matrix changes.

5. Numerical results

In this section, we use three examples with rectangular domains, one example with L-shape domain and one example with U-shape domain to test the proposed MGM with the banded smoother (MGMBS). All numerical experiments are performed via MATLAB R2013a on a PC with the configuration: Intel(R) Core(TM) i5-4590 CPU 3.30 GHz and 8 GB RAM.

Define the relative error

$$\mathrm{E}_{N,M}=\frac{\|\mathbf{u}-\tilde{\mathbf{u}}\|_\infty}{\|\mathbf{u}\|_\infty},$$

where $\mathbf{u}$ and $\tilde{\mathbf{u}}$ denote the exact solution and the approximate solution deriving from some iterative solvers. Since the results of $\mathrm{E}_{N,M}$ of different solvers are always the same for the same problem, we won't list results of $\mathrm{E}_{N,M}$ for examples with rectangular domains in the following. But since it is unusual to utilize uniform-grid discretization to discretize the SFDE on non-rectangular domains, results of $\mathrm{E}_{N,M}$ for examples with non-rectangular domains will be listed to illustrate the applicability of the discretization. Note that there are N linear systems to be solved. Thus, we denote by "iter", the average of the N iteration numbers. Denote by CPU, the running time of some algorithms by unit second. For convenience, we use MGMBS(ω) to denote Algorithm 2 with the proposed banded smoother of a bandwidth ω. Moreover, we take $\mathbf{u}^n$ as an initial guess of $\mathbf{u}^{n+1}(0\le n\le N-1)$ for all experiments and all tested solvers in this section.

As the discussion above, we have different choices of discretization schemes for the approximation of the fractional derivatives and different choices of the bandwidth ω. As mentioned in the remark after Theorem 2, for the second-order discretization scheme (4.7), we focus MGMBS(ω) only on the case $\alpha,\beta\in[\frac{\sqrt{17}-1}{2},2)$. Hence, when $\min\{\alpha,\beta\}\in(1,\frac{\sqrt{17}-1}{2})$, we only use the first-order discretization scheme (4.5). In the following, we use Example 1 to determine the optimal value of ω and to examine which discretization scheme is more suitable for MGMBS(ω) when $\alpha,\beta\in[\frac{\sqrt{17}-1}{2},2)$.

Example 1. Consider the two-dimensional SFDE (1.1)–(1.3) with

$$\begin{aligned}
&u(x,y,t)=\exp(-t)x^2(2-x)^2y^2(2-y)^2, && d_+(x,y,t)=\exp(x)x^\alpha(1+y),\\
&d_-(x,y,t)=(4-x)(1+y), && e_+(x,y,t)=(1+y)y^\beta(1+x),\\
&e_-(x,y,t)=(2-y)(1+x), && \Omega=(0,2)\times(0,2),\qquad T=1.
\end{aligned}$$

We use MGMBS(ω) with both the first-order discretization scheme (4.5) and the second-order discretization scheme (4.7) to solve Example 1. The corresponding results are listed in Table 3 for the scheme (4.5) and Table 4 for the scheme (4.7).

Coinciding with Tables 1–2, Tables 3–4 show that (i) the performance of the proposed algorithm for the second-order discretization scheme (4.7) is better than that for the first-order discretization scheme (4.5) when $\alpha=\beta\in[\frac{\sqrt{17}-1}{2},2)$; (ii) the performance of the proposed algorithm for the first-order discretization scheme (4.5) is better than that for the

Table 3
Performance of MGMBS(ω) with different values of ω when $N = 2^4$, $\nu = 1$ and the first-order discretization scheme (4.5) is used.

(α, β)	$M+1$	MGMBS(1)		MGMBS(3)		MGMBS(4)		MGMBS(6)	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU
(1.1, 1.1)	2^7	11.0	2.03 s	11.0	2.09 s	11.0	2.14 s	11.0	2.19 s
	2^8	12.1	7.57 s	12.0	7.83 s	12.0	7.33 s	12.0	7.98 s
	2^9	14.0	57.48 s	14.0	58.53 s	14.0	58.20 s	14.0	60.68 s
(1.1, 1.5)	2^7	15.0	2.70 s	14.0	2.60 s	14.1	2.68 s	15.1	2.95 s
	2^8	17.0	10.47 s	16.1	9.37 s	16.0	10.10 s	17.0	10.96 s
	2^9	19.0	77.61 s	18.0	75.38 s	18.0	76.76 s	18.1	79.85 s
(1.6, 1.6)	2^7	9.0	1.65 s	13.0	2.41 s	13.0	2.46 s	14.0	2.73 s
	2^8	9.0	5.57 s	13.0	7.94 s	13.0	7.86 s	14.0	8.79 s
	2^9	9.1	37.98 s	13.0	53.97 s	14.0	58.67 s	15.0	64.89 s
(1.6, 1.9)	2^7	13.0	2.33 s	17.0	3.12 s	17.0	3.18 s	18.0	3.46 s
	2^8	14.0	8.33 s	18.0	10.76 s	19.0	10.96 s	19.0	12.41 s
	2^9	14.0	57.50 s	19.0	77.79 s	20.0	83.07 s	21.0	90.22 s
(1.9, 1.9)	2^7	15.0	2.70 s	20.0	3.66 s	21.0	3.92 s	22.0	4.20 s
	2^8	16.0	9.55 s	21.0	12.95 s	22.0	12.55 s	23.0	14.82 s
	2^9	16.0	65.43 s	23.0	94.15 s	24.0	98.38 s	25.0	106.15 s

Table 4
Performance of MGMBS(ω) with different values of ω when $N = 2^4$, $\nu = 1$ and the second-order discretization scheme (4.7) is used.

(α, β)	$M+1$	MGMBS(1)		MGMBS(3)		MGMBS(4)		MGMBS(6)	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU
(1.6, 1.6)	2^7	8.0	1.52 s	12.0	2.27 s	13.0	2.49 s	14.0	2.80 s
	2^8	8.0	4.86 s	13.0	7.79 s	14.0	8.39 s	15.0	10.00 s
	2^9	8.0	33.51 s	13.0	54.52 s	15.0	63.10 s	16.0	68.78 s
(1.6, 1.9)	2^7	12.0	2.19 s	21.0	3.84 s	22.0	4.07 s	23.0	4.34 s
	2^8	13.0	7.27 s	27.0	16.21 s	29.0	16.31 s	31.0	18.33 s
	2^9	13.0	52.95 s	31.0	124.28 s	35.0	142.28 s	39.0	163.69 s
(1.9, 1.9)	2^7	12.0	2.16 s	18.0	3.29 s	19.0	3.57 s	19.0	3.67 s
	2^8	12.0	7.12 s	19.0	10.85 s	20.0	11.60 s	21.0	13.24 s
	2^9	12.0	49.42 s	20.0	82.35 s	21.0	87.21 s	22.0	94.38 s

second-order discretization scheme (4.7) when α is away from β. Thus, in the following experiments, we focus MGMBS(ω) with the scheme (4.5) on the case $\alpha \neq \beta$ or $\min\{\alpha, \beta\} \in (1, \frac{\sqrt{17}-1}{2})$ and focus MGMBS(ω) with scheme (4.7) on the case $\alpha = \beta \in [\frac{\sqrt{17}-1}{2}, 2)$. Moreover, we see that the performance of MGMBS(1) is generally better than MGMBS(ω) with ω larger than 1. The reason may be that when $\omega = 1$, high frequency error components can be eliminated efficiently by smoothing procedure and low frequency error components can be removed via correction procedure. However, when ω is large, both low and high frequency error components are removed in the smoothing procedure, it may not be effective for the convergence of MGM iterations; see, for instance, [3]. Indeed, the computational cost for large ω is higher than that for $\omega = 1$. In the following experiments, we focus on the results for $\omega = 1$ only.

Denote by MGMJS, Algorithm 2 with Jacobi smoother. For the remaining numerical experiments, we compare MGMBS(1) with MGMJS and preconditioned GMRES methods to illustrate high efficiency of MGMBS(1). Note that when $\omega = 0$, both the pre-smoother (3.5) and the post-smoother (3.6) are exactly the Jacobi smoother. Hence, comparing MGMBS(1) and MGMJS also examines the importance of the positivity of the bandwidth. Moreover, we set $\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \le 10^{-7}$ as stopping criterion for GMRES in all experiments of this section, where $\mathbf{r}_k$ denotes the residual at kth GMRES iteration. The stopping criterion of MGMBS(1) is given by Algorithm 2. Also, it is well known that another way to generate the matrices on coarse grid, $\mathbf{A}_i$ and $\tilde{\mathbf{A}}_i$ is the Galerkin coarsening technique. i.e.,

$$\mathbf{A}_i = \mathbf{I}_{i+1}^{i}\mathbf{A}_{i+1}\mathbf{I}_i^{i+1}, \quad \tilde{\mathbf{A}}_i = \mathbf{I}_{i+1}^{i}\tilde{\mathbf{A}}_{i+1}\mathbf{I}_i^{i+1}, \quad 2 \le i \le l-1. \tag{5.1}$$

In order to further verify applicability of the proposed banded smoother, we also test MGMBS(1) with coarse grid matrices given by (5.1), which is denoted by GMGMBS(1). For the case of constant coefficients and in the rectangular domain, the finest grid matrix is exact block Toeplitz. Therefore, the coarser grid matrices will keep the block Toeplitz structure; see, for more details, [19]. Nevertheless, (5.1) will distort the block Toeplitz-like structure of the coarser grid matrices in the case of variable coefficients, which may lead to the expensive matrix-vector multiplication for the coarser grid matrices; see [14]. Thus, we only test GMGMBS(1) for the case of constant coefficient (see Example 2 below).

Table 5
Results of GMGMBS(1), MGMBS(1), MGMJS and BCCB preconditioner, when $N = 2^4$ and the scheme (4.5) is used.

(α, β)	$M+1$	GMGMBS(1)		MGMBS(1)		MGMJS		BCCB	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU
(1.1, 1.5)	2^7	6.3	0.91 s	12.0	1.26 s	33.6	5.16 s	14.4	1.92 s
	2^8	7.4	5.18 s	15.1	8.78 s	46.2	22.61 s	17.6	7.37 s
	2^9	9.3	34.79 s	18.2	51.52 s	62.4	219.66 s	20.9	67.21 s
(1.5, 1.5)	2^7	9.0	1.21 s	8.0	0.87 s	12.1	1.86 s	12.9	1.66 s
	2^8	8.0	5.48 s	8.0	4.84 s	12.1	6.84 s	14.0	5.92 s
	2^9	8.0	29.95 s	9.0	26.33 s	13.1	52.66 s	16.0	50.94 s
(1.6, 1.9)	2^7	15.0	1.95 s	15.0	1.56 s	39.1	6.54 s	16.9	2.02 s
	2^8	16.0	10.75 s	15.0	8.78 s	31.1	18.50 s	20.9	8.40 s
	2^9	15.0	54.16 s	15.0	42.72 s	36.1	130.45 s	25.0	82.56 s

Table 6
Results of GMGMBS(1), MGMBS(1), MGMJS and BCCB preconditioner, when $N = 2^4$ and the scheme (4.7) is used.

(α, β)	$M+1$	GMGMBS(1)		MGMBS(1)		MGMJS		BCCB	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU
(1.6, 1.6)	2^7	6.0	0.82 s	6.0	0.65 s	7.0	1.12 s	12.0	1.39 s
	2^8	6.0	4.28 s	6.0	3.72 s	7.0	3.29 s	14.0	5.87 s
	2^9	6.0	23.10 s	6.0	18.12 s	8.0	30.46 s	16.0	50.49 s
(1.75, 1.75)	2^7	8.0	1.08 s	8.0	0.87 s	9.0	1.41 s	12.1	1.43 s
	2^8	8.0	5.50 s	8.0	4.84 s	9.0	4.27 s	14.0	5.66 s
	2^9	8.0	29.95 s	8.0	23.56 s	9.0	34.10 s	17.0	54.38 s
(1.9, 1.9)	2^7	10.0	1.33 s	10.0	1.05 s	25.0	3.80 s	13.0	1.49 s
	2^8	10.0	6.83 s	10.0	5.96 s	18.0	8.48 s	15.0	5.86 s
	2^9	10.0	36.85 s	10.0	28.98 s	11.0	41.37 s	16.9	53.60 s
	2^{10}	10.0	168.07 s	10.0	127.18 s	11.0	211.03 s	20.5	253.65 s

Example 2. Consider the two-dimensional SFDE (1.1)–(1.3) with

$$u(x,y,t) = \exp(-t)x^2(2-x)^2y^2(2-y)^2,$$
$$d_+ = e_+ \equiv 1, \quad d_- = e_- \equiv 2, \quad \Omega = (0,2)\times(0,2), \quad T = 1.$$

In the case of constant coefficient, **A** is of block Toeplitz with Toeplitz block structure. It is well known that GMRES with Strang's block circulant with circulant block (BCCB) preconditioner [4] is an efficient solver for such linear systems. We solve Example 2 by GMRES with BCCB preconditioner and MGMJS, MGMBS(1), GMGMBS(1) with $\nu = 1$. The corresponding results are listed in Table 5 for the first-order discretization scheme (4.5) and in Table 6 for the second-order discretization scheme (4.7).

From Tables 5–6, we see that the performance of GMGMBS(1) is as almost the same as MGMBS(1), which implies that our proposed smoother works in the sense of both geometry and algebraic multigrid. Also, the iteration number and CPU cost of GMGMBS(1) and MGMBS(1) are in general less than both of MGMJS and GMRES with BCCB preconditioner, which means multigrid method with the proposed banded smoother is more efficient that MGMJS and BCCB for Example 2. Moreover, better performance of GMGMBS(1) and MGMBS(1) compared with MGMJS also suggests the importance of positivity of the bandwidth ω.

Example 3. Consider the two-dimensional SFDE (1.1)–(1.3) with

$$u(x,y,t) = \exp(-t)x^2(2-x)^2y^2(2-y)^2, \qquad d_+(x,y,t) = [1+\exp(-t)]\exp(x^2+y)x^\alpha,$$
$$d_-(x,y,t) = [1+\exp(-t)]\exp(2x-x^2+y)(2-x)^\alpha, \qquad e_+(x,y,t) = [1+\exp(-t)]\exp(y^2+x)y^\beta,$$
$$e_-(x,y,t) = [1+\exp(-t)]\exp(2y-y^2+x)(2-y)^\beta, \qquad \Omega = (0,2)\times(0,2), \quad T = 1.$$

Note that d_+, d_-, e_+ and e_- are no longer constants in Example 3. In order to apply the BCCB preconditioner, we take the averages of these coefficients on the grid points. Also, the row approximation preconditioner proposed in [15] is efficient for solving the SFDEs with non-constant coefficients. Take 5 interpolating points in each direction for the row approximation preconditioner and denote it by P(5). The results of BCCB, P(5), MGMBS(1) and MGMJS with $\nu = 1$, are listed in Table 7 for the first-order discretization scheme (4.5) and in Table 8 for the second-order discretization scheme (4.7).

From Tables 7–8, we see that both CPU cost and iteration number of MGMBS(1) are much less than those of other three solvers for Example 3, which means MGMBS(1) is the most efficient one among the four solvers. Moreover, better perfor-

Table 7
Results of MGMBS(1), MGMJS, P(5) and BCCB, when $N = 2^4$ and the scheme (4.5) is used.

(α, β)	$M+1$	MGMBS(1)		MGMJS		P(5)		BCCB	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU
(1.1, 1.5)	2^7	14.1	2.95 s	52.0	8.88 s	62.5	23.56 s	296.6	210.19 s
	2^8	17.0	11.34 s	72.0	37.15 s	89.8	130.26 s	824.3	2084.67 s
	2^9	20.0	89.64 s	96.0	391.12 s	134.7	1503.40 s	2134.6	59341.45 s
(1.5, 1.5)	2^7	8.0	1.87 s	16.0	2.78 s	36.8	13.32 s	142.5	53.45 s
	2^8	8.0	6.53 s	17.9	9.33 s	47.9	63.80 s	184.6	310.61 s
	2^9	9.0	46.92 s	18.0	73.74 s	62.9	562.59 s	233.1	5285.75 s
(1.6, 1.9)	2^7	12.1	2.59 s	57.0	9.56 s	44.8	18.01 s	91.2	23.65 s
	2^8	13.0	9.16 s	74.0	38.26 s	59.1	81.74 s	117.1	134.12 s
	2^9	13.0	62.92 s	91.0	372.73 s	79.8	772.04 s	151.7	2292.43 s

Table 8
Results of MGMBS(1), MGMJS, P(5) and BCCB, when $N = 2^4$ and the scheme (4.7) is used.

(α, β)	$M+1$	MGMBS(1)		MGMJS		P(5)		BCCB	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU
(1.6, 1.6)	2^7	8.0	1.88 s	14.0	2.53 s	34.8	12.75 s	140.8	51.70 s
	2^8	8.0	6.56 s	15.0	8.40 s	44.4	60.42 s	160.0	239.09 s
	2^9	8.0	43.35 s	16.0	68.26 s	57.2	511.85 s	179.8	3163.72 s
(1.75, 1.75)	2^7	7.0	1.71 s	17.0	3.02 s	33.1	12.15 s	77.6	18.52 s
	2^8	8.0	6.55 s	19.0	10.71 s	43.3	58.30 s	89.1	84.61 s
	2^9	8.0	43.40 s	19.0	97.99 s	54.5	472.28 s	105.4	1190.28 s
(1.9, 1.9)	2^7	10.0	2.23 s	24.0	4.19 s	35.3	12.91 s	55.7	10.28 s
	2^8	10.0	7.64 s	24.0	13.44 s	44.6	62.09 s	66.6	50.20 s
	2^9	11.0	54.99 s	24.0	97.99 s	57.1	506.85 s	81.4	730.81 s

mance of MGMBS(1) compared with MGMJS again demonstrates that positivity of the bandwidth ω is useful to accelerating the convergence of MGM and improving the efficiency.

Example 4. Consider two-dimensional SFDE with

$$u(x,y,t) = \exp(-t)x^2(1-x)^2(2-x)^2y^2(1-y)^2(2-y)^2, \quad d_+(x,y,t) = \exp\left(\sin^2(20y) + \alpha x\right),$$

$$d_-(x,y,t) = \exp\left(\sin^2(20y) + \alpha(2-x)\right), \quad e_+(x,y,t) = \exp\left(\sin^2(20x) + \beta y\right),$$

$$e_-(x,y,t) = \exp\left(\sin^2(20x) + \beta(2-y)\right), \quad \bar{\Omega} = ([0,2]\times[0,2]) \setminus ((1,2)\times(1,2)), \quad T = 1.$$

For Example 4, we extend the banded preconditioner proposed in [9] to solving the SFDE on L-shape domain. Let

$$\mathbf{A}^{(c)} = \mathbf{I}_{\ddot{M}} + \eta_x\left[\mathbf{D}_+\hat{\mathbf{B}}^{(c)}_{\alpha,M} + \mathbf{D}_-\left(\hat{\mathbf{B}}^{(c)}_{\alpha,M}\right)^{\mathrm{T}}\right] + \eta_y\left[\mathbf{E}_+\check{\mathbf{B}}^{(c)}_{\beta,M} + \mathbf{E}_-\left(\check{\mathbf{B}}^{(c)}_{\beta,M}\right)^{\mathrm{T}}\right],$$

$$\hat{\mathbf{B}}^{(c)}_{\alpha,M} = \begin{bmatrix} \mathbf{I}_M \otimes \mathbf{G}^{(c)}_{\alpha,\dot{M}} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{\bar{M}} \otimes \mathbf{G}^{(c)}_{\alpha,M} \end{bmatrix}, \qquad \check{\mathbf{B}}^{(c)}_{\beta,M} = \begin{bmatrix} \mathbf{G}^{(c,l,u)}_{\beta,\dot{M}} \otimes \mathbf{I}_{\dot{M}} & \mathbf{G}^{(c,r,u)}_{\beta,\dot{M}} \otimes \tilde{\mathbf{I}}_M \\ \mathbf{G}^{(c,l,d)}_{\beta,\dot{M}} \otimes \tilde{\mathbf{I}}^{\mathrm{T}}_M & \mathbf{G}^{(c,r,d)}_{\beta,\dot{M}} \otimes \mathbf{I}_M \end{bmatrix},$$

where $\mathbf{G}^{(c)}_{\alpha,\dot{M}}$ and $\mathbf{G}^{(c)}_{\alpha,M}$ are diagonal-compensated banded truncations of $\mathbf{G}_{\alpha,\dot{M}}$ and $\mathbf{G}_{\alpha,M}$, respectively [9], bandwidths of $\mathbf{G}^{(c)}_{\alpha,\dot{M}}$ and $\mathbf{G}^{(c,l,u)}_{\alpha,\dot{M}}$ are $2l$ and l, respectively, $\mathbf{G}^{(c,l,u)}_{\beta,\dot{M}} \in \mathbb{R}^{M\times M}$, $\mathbf{G}^{(c,r,u)}_{\beta,\dot{M}} \in \mathbb{R}^{M\times\bar{M}}$, $\mathbf{G}^{(c,l,d)}_{\beta,\dot{M}} \in \mathbb{R}^{\bar{M}\times M}$ and $\mathbf{G}^{(c,r,d)}_{\beta,\dot{M}} \in \mathbb{R}^{\bar{M}\times\bar{M}}$ denote the partitions of $\mathbf{G}^{(c)}_{\beta,\dot{M}}$ such that

$$\mathbf{G}^{(c)}_{\beta,\dot{M}} = \begin{bmatrix} \mathbf{G}^{(c,l,u)}_{\beta,\dot{M}} & \mathbf{G}^{(c,r,u)}_{\beta,\dot{M}} \\ \mathbf{G}^{(c,l,d)}_{\beta,\dot{M}} & \mathbf{G}^{(c,r,d)}_{\beta,\dot{M}} \end{bmatrix},$$

provided that $\mathbf{G}^{(c)}_{\beta,\dot{M}}$ is a diagonal-compensated banded truncation of $\mathbf{G}_{\beta,\dot{M}}$ [9] and bandwidth of $\mathbf{G}^{(c)}_{\beta,\dot{M}}$ is $2l$. Here, $l = \log_2(M+1)$. Then, we obtain a banded preconditioner $\mathbf{P}_b = \mathbf{L}_b\mathbf{U}_b$ such that $\mathbf{L}_b\mathbf{U}_b$ is the incomplete LU factorization with no fill-in (ILU(0)) of $\mathbf{A}^{(c)}$. We solve Example 4 by GMRES with the extended banded preconditioner $\mathbf{P}_b$, MGMBS(1) and

Table 9
Results of MGMBS(1), MGMJS and $\mathbf{P}_b$ when $N = 2^3$ and the scheme (4.5) is used.

(α, β)	$\dot{M}+1$	MGMBS(1)			MGMJS			$\mathbf{P}_b$		
		iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$
(1.1, 1.5)	2^7	12.0	2.34 s	2.18e−2	33.4	6.50 s	2.18e−2	46.1	5.03 s	2.18e−2
	2^8	14.1	7.77 s	1.08e−2	46.6	24.31 s	1.08e−2	91.1	47.96 s	1.08e−2
	2^9	17.1	49.79 s	5.11e−3	66.0	188.75 s	5.11e−3	181.1	1249.32 s	5.11e−3
(1.5, 1.5)	2^7	7.0	1.40 s	1.08e−2	12.0	2.71 s	1.08e−2	43.0	4.74 s	1.08e−2
	2^8	7.0	3.98 s	5.38e−3	10.0	5.76 s	5.38e−3	78.0	39.35 s	5.38e−3
	2^9	8.0	25.51 s	2.72e−3	10.0	30.37 s	2.72e−3	147.0	861.63 s	3.42e−4
(1.6, 1.9)	2^7	8.0	1.58 s	8.00e−3	48.0	9.36 s	8.00e−3	65.0	8.29 s	8.00e−3
	2^8	8.0	4.52 s	3.92e−3	–	–	–	123.1	78.98 s	3.92e−3
	2^9	8.0	25.50 s	1.95e−3	–	–	–	237.1	2061.80 s	1.95e−3

Table 10
Results of MGMBS(1), MGMJS and $\mathbf{P}_b$ when $N = 2^3$ and the scheme (4.7) is used.

(α, β)	$\dot{M}+1$	MGMBS(1)			MGMJS			$\mathbf{P}_b$		
		iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$
(1.6, 1.6)	2^7	5.0	1.02 s	2.09e−3	8.0	1.66 s	2.09e−3	41.0	4.30 s	2.09e−3
	2^8	5.0	2.92 s	6.64e−4	9.0	4.80 s	6.64e−4	77.0	36.69 s	6.64e−4
	2^9	5.0	16.91 s	3.42e−4	11.0	35.49 s	3.42e−4	147.0	854.85 s	3.42e−4
(1.75, 1.75)	2^7	6.0	1.23 s	1.84e−3	9.0	2.18 s	1.84e−3	52.0	6.15 s	1.84e−3
	2^8	6.0	3.46 s	5.48e−4	12.0	8.21 s	5.48e−4	98.0	56.89 s	5.48e−4
	2^9	6.0	19.99 s	2.43e−4	18.0	48.14 s	2.43e−4	189.9	1379.79 s	2.43e−4
(1.9, 1.9)	2^7	8.0	1.64 s	1.51e−3	17.0	3.09 s	1.51e−3	65.0	8.29 s	1.51e−3
	2^8	8.0	4.64 s	4.36e−4	18.0	9.18 s	4.36e−4	126.0	82.14 s	4.36e−4
	2^9	8.0	25.68 s	1.75e−4	44.0	127.70 s	1.75e−4	248.0	2244.46 s	1.75e−4
	2^{10}	9.0	157.96 s	1.18e−4	–	–	–	660.9	30207.37 s	1.18e−4

MGMJS with $\nu = 2$. The corresponding results are listed in Table 9 for the first-order discretization scheme (4.5) and in Table 10 for the second-order discretization scheme (4.7).

'–' denotes divergence of solver. Note that from Tables 9–10, $E_{N,M}$ of different solvers are always the same and small except for some cases of divergence, which suggests that the uniform-grid discretization of SFDE on L-shape domain, (2.9) is actually applicable. Also, iteration number and CPU cost of MGMBS(1) are much smaller than the other two solvers, which means MGMBS(1) is the most efficient one among the three solvers for solving the SFDE on L-shape domain. Moreover, we note that MGMJS diverges for the case of both $(\alpha, \beta) = (1.6, 1.9)$ and $(\alpha, \beta) = (1.9, 1.9)$ when M is large. That means the positivity of bandwidth ω can not only accelerate the convergence of MGM but also remedy the situation where MGMJS is not even applicable.

5.1. MGMBS for SFDE on U-shape domain

In this subsection, we extend the proposed MGM to solving SFDE on a U-shape domain which results from an SFDE on a rectangular domain. Consider the SFDE on rectangular domain with

$$u(x,y,t) = \begin{cases} \exp(-t)\left(\prod_{k=0}^{3}(x-k)^2\right)\left(\prod_{k=0}^{2}(y-k)^2\right), & (x,y,t) \in \bar{\Omega}_U \times [0,T], \\ 0, & (x,y,t) \in \bar{\Omega}_S \times [0,T], \end{cases} \tag{5.2}$$

$$\Omega = (0,3)\times(0,2), \quad T = 1, \quad d_+(x,y) = d_-(x,y) = \exp(\sin^2(20y) + \sin^2 x), \quad (x,y) \in \Omega,$$

$$e_+(x,y) = \exp(\sin^2(20x) + 2\sin^2 y), \quad e_-(x,y) = \exp(\sin^2(20x) + 3\sin^2(2y)), \quad (x,y) \in \Omega,$$

where $\bar{\Omega}_S = [1,2]\times[1,2]$, $\bar{\Omega}_U = \overline{\Omega \setminus \bar{\Omega}_S}$ is a U-shape domain. We assume that values of the solution $u(x,y,t)$ on the domain $(x,y,t) \in \bar{\Omega}_S \times [0,T]$ are already known while values of the solution $u(x,y,t)$ on the domain $(x,y,t) \in \Omega_U \times [0,T]$ are unknowns to be solved, where Ω_U denotes the interior of $\bar{\Omega}_U$. Then, the SFDE problem on the rectangular domain Ω is transformed into an SFDE on the U-shape domain $\bar{\Omega}_U$ such that

$$\frac{\partial u(x,y,t)}{\partial t} = d_+(x,y,t)\,{}_0D_x^{\alpha}u(x,y,t) + d_-(x,y,t)\,{}_xD_3^{\alpha}u(x,y,t) + e_+(x,y,t)\,{}_0D_y^{\beta}u(x,y,t) + e_-(x,y,t)\,{}_yD_{d(x)}^{\beta}u(x,y,t) + f(x,y,t), \qquad (x,y,t) \in \Omega_U \times (0,T], \tag{5.3}$$

$$u(x,y,t)=0, \qquad (x,y,t)\in\partial\Omega_U\times(0,T], \tag{5.4}$$

$$u(x,y,0)=x^2(x-1)^2(x-2)^2(x-3)^2y^2(1-y)^2(2-y)^2, \qquad (x,y)\in\bar{\Omega}_U, \tag{5.5}$$

where $\partial\Omega_U$ denotes the boundary of $\bar{\Omega}_U$, f is determined by (5.2),

$$d(x)=\begin{cases}2, & x\in(0,1)\cup(2,3),\\ 1, & x\in[1,2].\end{cases}$$

The rest of this subsection is devoted to (i) showing structure of the coefficient matrices resulting from uniform-grid discretization of the SFDE on U-shape domain, (5.3)–(5.5); (ii) demonstrating how to apply the proposed MGM to solving the corresponding linear systems.

Let N and M be positive integers. Denote by $\tau=T/N$ and $h=1/(M+1)$, the temporal-step size and the spatial-step size, respectively. Define the temporal girds, spatial grids in x-direction and spatial grids in y-direction by $\{t_n=n\tau|0\le n\le N\}$, $\{x_i=ih|0\le i\le \dot{M}_x+1\}$ and $\{y_j=jh|0\le j\le \dot{M}_y+1\}$, respectively, where $\dot{M}_x=3M+2$ and $\dot{M}_y=2M+1$. Then, the vectors consisting of spatial-grid points with x-dominant ordering and y-dominant ordering are respectively defined by

$$\mathscr{P}_{U,x,M}=\left(\mathscr{V}_{x,M,d},\mathscr{V}_{x,M,u}\right)^{\mathrm{T}}\in\mathbb{T}^{\ddot{M}\times 1} \quad \text{and} \quad \mathscr{P}_{U,y,M}=\left(\mathscr{V}_{y,M,l},\mathscr{V}_{y,M,m},\mathscr{V}_{y,M,r}\right)^{\mathrm{T}}\in\mathbb{T}^{\ddot{M}\times 1}, \tag{5.6}$$

where $\ddot{M}=M(\dot{M}_x+\dot{M}_y+1)$,

$$\mathscr{V}_{x,M,d}=\left(\{P_{i1}\}_{i=1}^{\dot{M}_x},\{P_{i2}\}_{i=1}^{\dot{M}_x},\ldots,\{P_{iM}\}_{i=1}^{\dot{M}_x}\right)\in\mathbb{T}^{1\times\dot{M}_xM},$$

$$\mathscr{V}_{x,M,u}=\left(\{\tilde{P}_{i,M+1}\}_{i=1}^{2M},\{\tilde{P}_{i,M+2}\}_{i=1}^{2M},\ldots,\{\tilde{P}_{i,\dot{M}_y}\}_{i=1}^{2M}\right)\in\mathbb{T}^{1\times(\dot{M}_y+1)M},$$

$$\mathscr{V}_{y,M,l}=\left(\{P_{1j}\}_{j=1}^{\dot{M}_y},\{P_{2j}\}_{j=1}^{\dot{M}_y},\ldots,\{P_{Mj}\}_{j=1}^{\dot{M}_y}\right)\in\mathbb{T}^{1\times\dot{M}_yM},$$

$$\mathscr{V}_{y,M,m}=\left(\{P_{M+1,j}\}_{j=1}^{M},\{P_{M+2,j}\}_{j=1}^{M},\ldots,\{P_{2M+2,j}\}_{j=1}^{M}\right)\in\mathbb{T}^{1\times(M+2)M},$$

$$\mathscr{V}_{y,M,r}=\left(\{P_{2M+3,j}\}_{j=1}^{\dot{M}_y},\{P_{2M+4,j}\}_{j=1}^{\dot{M}_y},\ldots,\{P_{\dot{M}_x,j}\}_{j=1}^{\dot{M}_y}\right)\in\mathbb{T}^{1\times\dot{M}_yM},$$

$$\tilde{P}_{ij}=\begin{cases}P_{ij}, & 1\le i\le M,\\ P_{i+M+2,j}, & M+1\le i\le 2M,\end{cases} \qquad M+1\le j\le\dot{M}_y,$$

P_{ij} denotes the point (x_i,y_j) for $0\le i\le\dot{M}_x+1$, $0\le j\le\dot{M}_y+1$, respectively.

By (2.3)–(2.4) and forward difference approximation of $\frac{\partial u}{\partial t}$, we obtain an implicit finite difference discretization of the SFDE on the U-shape domain $\bar{\Omega}_U$ as follows

$$\tau^{-1}(\mathbf{u}^{n+1}-\mathbf{u}^n)=-\left(h^{-\alpha}\mathbf{B}_x+h^{-\beta}\mathbf{B}_y\right)\mathbf{u}^{n+1}+\mathbf{f}^{n+1}, \qquad 0\le n\le N-1, \tag{5.7}$$

where $\mathbf{u}^n=u(\mathscr{P}_{U,x,M},t_n)$, $\mathbf{f}^n=f(\mathscr{P}_{U,x,M},t_n)$,

$$\mathbf{B}_x=\mathbf{D}_+\hat{\mathbf{B}}_{\mathrm{U},\alpha,M}+\mathbf{D}_-\hat{\mathbf{B}}_{\mathrm{U},\alpha,M}^{\mathrm{T}}, \qquad \mathbf{B}_y=\mathbf{E}_+\check{\mathbf{B}}_{\mathrm{U},\beta,M}+\mathbf{E}_-\check{\mathbf{B}}_{\mathrm{U},\beta,M}^{\mathrm{T}},$$

$$\mathbf{D}_\pm=\mathrm{diag}(d_\pm(\mathscr{P}_{U,x,M})), \qquad \mathbf{E}_\pm=\mathrm{diag}(e_\pm(\mathscr{P}_{U,x,M})),$$

$$\check{\mathbf{B}}_{\mathrm{U},\beta,M}=\begin{bmatrix}\mathbf{G}_{\beta,\dot{M}_y}^{(l,u)}\otimes\mathbf{I}_{\dot{M}_x} & \mathbf{G}_{\beta,\dot{M}_y}^{(r,u)}\otimes\hat{\mathbf{I}}_M\\ \mathbf{G}_{\beta,\dot{M}_y}^{(l,d)}\otimes\hat{\mathbf{I}}_M^{\mathrm{T}} & \mathbf{G}_{\beta,\dot{M}_y}^{(r,d)}\otimes\mathbf{I}_{2M}\end{bmatrix}, \qquad \hat{\mathbf{I}}_M^{\mathrm{T}}=\begin{bmatrix}\mathbf{I}_M & \mathbf{O}_{M\times\bar{M}_x} & \mathbf{O}_{M\times M}\\ \mathbf{O}_{M\times M} & \mathbf{O}_{M\times\bar{M}_x} & \mathbf{I}_M\end{bmatrix},$$

$$\hat{\mathbf{B}}_{\mathrm{U},\alpha,M}=\mathrm{blockdiag}(\mathbf{I}_M\otimes\mathbf{G}_{\alpha,\dot{M}_x},\ \mathbf{I}_{M+1}\otimes\mathbf{G}_{\alpha,\dot{M}_x}^{\sharp}),$$

provided that $\mathbf{G}_{\beta,\dot{M}_y}^{(l,u)}\in\mathbb{R}^{M\times M}$, $\mathbf{G}_{\beta,\dot{M}_y}^{(r,u)}\in\mathbb{R}^{M\times\bar{M}_y}$, $\mathbf{G}_{\beta,\dot{M}_y}^{(l,d)}\in\mathbb{R}^{\bar{M}_y\times M}$, $\mathbf{G}_{\beta,\dot{M}_y}^{(r,d)}\in\mathbb{R}^{\bar{M}_y\times\bar{M}_y}$ are partitions of $\mathbf{G}_{\beta,\dot{M}_y}$ such that

$$\mathbf{G}_{\beta,\dot{M}_y}=\begin{bmatrix}\mathbf{G}_{\beta,\dot{M}_y}^{(l,u)} & \mathbf{G}_{\beta,\dot{M}_y}^{(r,u)}\\ \mathbf{G}_{\beta,\dot{M}_y}^{(l,d)} & \mathbf{G}_{\beta,\dot{M}_y}^{(l,u)}\end{bmatrix},$$

$\mathbf{G}_{\alpha,\dot{M}_x}^{\sharp}\in\mathbb{R}^{\bar{M}_x\times\bar{M}_x}$ derives from deleting the median $\bar{M}_x$ columns and the median $\bar{M}_x$ rows of $\mathbf{G}_{\alpha,\dot{M}_x}$. Here, $\bar{M}_x=\dot{M}_x-2M$ and $\bar{M}_y=\dot{M}_y-M$.

Similar to the discussion in Section 3, the resulting task from (5.7) is to solve

$$\mathbf{A}\mathbf{u}=\mathbf{b}, \tag{5.8}$$

where $\mathbf{b} \in \mathbb{R}^{\ddot{M}\times 1}$ denotes some given right hand sides, $\mathbf{u}$ is the unknown vector to be solved,

$$\mathbf{A} = \mathbf{I}_{\ddot{M}} + \eta_x(\mathbf{D}_+\hat{\mathbf{B}}_{\mathrm{U},\alpha,M} + \mathbf{D}_-\hat{\mathbf{B}}^{\mathrm{T}}_{\mathrm{U},\alpha,M}) + \eta_y(\mathbf{E}_+\check{\mathbf{B}}_{\mathrm{U},\beta,M} + \mathbf{E}_-\check{\mathbf{B}}^{\mathrm{T}}_{\mathrm{U},\beta,M}),$$

with $\eta_x = \tau h^{-\alpha}$ and $\eta_y = \tau h^{-\beta}$. As we see, $\mathbf{A}$ in (5.8) is a block matrix with each block being BTL. Thus, matrix-vector multiplication of $\mathbf{A}$ in (5.8) requires only $\mathcal{O}(M^2 \log M)$ operations and only $\mathcal{O}(M^2)$ storage.

Now, we consider extending the proposed MGM to solving the linear system (5.8). We only need to redefine the restriction operator, the interpolation operator, the pre-smoothing iteration $\mathscr{S}_i$ and the post-smoothing iteration $\tilde{\mathscr{S}_i}$ in Algorithm 1. Let $M_i = 2^i - 1$ for $i \ge 2$ and $M = M_l$ for some $l > 2$. The corresponding spatial steps are given by $h_i = 1/(M_i + 1)$ for $2 \le i \le l$. Denote $\dot{M}_{x,i} = 3M_i + 2$, $\dot{M}_{y,i} = 2M_i + 1$ and $\ddot{M}_i = M_i(\dot{M}_{x,i} + \dot{M}_{y,i} + 1)$, for $i \ge 2$.

We firstly focus on construction of the pre-smoother $\mathscr{S}_i$ and the post-smoother $\tilde{\mathscr{S}_i}$. Denote $\mathbf{A}_i$, $\mathbf{A}$ in (5.8) with $M = M_i$. Split $\mathbf{A}_i$ as $\mathbf{A}_i = \mathbf{D}_i - \mathbf{R}_i$, where $\mathbf{D}_i$ is banded truncation of $\mathbf{A}_i$ with a bandwidth ω. Define the permutation matrices $\mathbf{P}_i$ such that

$$\mathscr{P}_{\mathrm{U},y,M_i} = \mathbf{P}_i \mathscr{P}_{\mathrm{U},x,M_i}, \qquad 3 \le i \le l.$$

Then, we obtain $\tilde{\mathbf{A}}_i = \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^{\mathrm{T}}$. Again, split $\tilde{\mathbf{A}}_i$ as $\tilde{\mathbf{A}}_i = \tilde{\mathbf{D}}_i - \tilde{\mathbf{R}}_i$, where $\tilde{\mathbf{D}}_i$ is banded truncation of $\tilde{\mathbf{A}}_i$ with bandwidth ω. Then, similar to the discussion in Section 3, for a linear system

$$\mathbf{A}_i\mathbf{x} = \mathbf{y}, \qquad 3 \le i \le l,$$

with a randomly given right hand side $\mathbf{y} \in \mathbb{R}^{\ddot{M}_i\times 1}$, we obtain two banded splitting iteration schemes as pre-smoother and post-smoother such that

$$\mathbf{x}^{k+1} = \mathscr{S}_i(\mathbf{x}^k, \mathbf{y}) := \mathbf{x}^k + \mathbf{D}_i^{-1}(\mathbf{y} - \mathbf{A}_i\mathbf{x}^k), \qquad 3 \le i \le l, \tag{5.9}$$

$$\mathbf{x}^{k+1} = \tilde{\mathscr{S}_i}(\mathbf{x}^k, \mathbf{y}) := \mathbf{x}^k + \mathbf{P}_i^{\mathrm{T}}\tilde{\mathbf{D}}_i^{-1}\mathbf{P}_i(\mathbf{y} - \mathbf{A}_i\mathbf{x}^k), \quad 3 \le i \le l. \tag{5.10}$$

Here, the construction of $\tilde{\mathbf{D}}_i$ requires only $\mathcal{O}(M_i^2)$ storage and only $\mathcal{O}(M_i^2)$ operations via the relationship $\tilde{\mathbf{A}}_i = \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^{\mathrm{T}}$.

Again, we still refer to piecewise linear restriction operator $\mathbf{I}_{i+1}^{i}$ and piecewise linear interpolation operator $\mathbf{I}_i^{i+1}$, which on U-shape domain are however defined by

$$\mathbf{I}_{i+1}^{i} = \begin{bmatrix} \mathbf{J}_{M_i} \otimes \hat{\mathbf{J}}_i & \\ & \check{\mathbf{J}}_i \otimes \bar{\mathbf{J}}_i \end{bmatrix}, \qquad \mathbf{I}_i^{i+1} = \begin{bmatrix} 4\mathbf{J}_{M_i}^{\mathrm{T}} \otimes \mathbf{J}_{\dot{M}_{x,i}}^{\mathrm{T}} & \mathbf{J}_i^{(r,u)} \\ & \mathbf{J}_i^{(r,d)} \end{bmatrix}, \quad 2 \le i \le l-1, \tag{5.11}$$

where $\mathbf{J}_{M_i}$ and $\mathbf{J}_{\dot{M}_{x,i}}$ are given by (3.8),

$$\tilde{\mathbf{J}}_i = \mathrm{blockdiag}(1, \mathbf{J}_{M_i}), \qquad \hat{\mathbf{J}}_i = \mathrm{blockdiag}(\mathbf{J}_{M_i}, \tilde{\mathbf{J}}_i, \tilde{\mathbf{J}}_i), \qquad \check{\mathbf{J}}_i = \mathrm{blockdiag}(\mathbf{J}_{M_i}, \mathbf{J}_{M_i})$$

$$\mathbf{J}_i^{(r,u)} = \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \bar{\mathbf{J}}_i^{\mathrm{T}} & \mathbf{O} \end{bmatrix}, \qquad \bar{\mathbf{J}}_i = \begin{bmatrix} \mathbf{J}_{M_i} & \tilde{\mathbf{O}}_i \\ \tilde{\mathbf{O}}_i & \mathbf{J}_{M_i} \end{bmatrix}, \qquad \mathbf{J}_i^{(r,d)} = \begin{bmatrix} 2 & \mathbf{O}_{1\times M_i} \\ \mathbf{e}_{M_{i+1},1} & 4\mathbf{J}_{M_i}^{\mathrm{T}} \end{bmatrix} \otimes \check{\mathbf{J}}_i^{\mathrm{T}},$$

where $\mathbf{e}_{M_{i+1},1}$ denotes the first column of $\mathbf{I}_{M_{i+1}}$, $\tilde{\mathbf{O}}_i = \mathbf{O}_{M_i\times(M_i+2)M_{i+1}}$, $\mathbf{O}$ denotes zero matrix with proper size.

Note that $\mathbf{A}_i$ is still a block matrix with each block being BTL for $2 \le i \le l$. Moreover, both the restriction and the interpolation operator in (5.11) are still sparse. Thus, similar to the discussion in Section 3, operations cost and storage requirement of Algorithm 1 for solving the SFDE on U-shape domain are still of $\mathcal{O}(M^2 \log M)$ and $\mathcal{O}(M^2)$, respectively. Besides, the convergence property of (5.9) and (5.10) can be similarly proved.

In the rest of this subsection, we firstly extend the banded preconditioner proposed in [9] to the linear system (5.8). And then, we solve the SFDE (5.3)–(5.5) by using GMRES with the extended banded preconditioner, MGMBS(1) and MGMJS and compare the results.

In order to construct the banded preconditioner, we solve the linear system under a permuted ordering such that

$$\tilde{\mathbf{A}}\tilde{\mathbf{u}} = \tilde{\mathbf{b}}, \tag{5.12}$$

where $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}_l$, $\tilde{\mathbf{u}} = \mathbf{P}_l\mathbf{u}$, $\tilde{\mathbf{b}} = \mathbf{P}_l\mathbf{b}$. It is easy to see that (5.12) is actually equivalent to (5.8).

For any $\mathbf{X} \in \mathbb{R}^{\dot{M}_x\times\dot{M}_x}$, $\mathbf{Y} \in \mathbb{R}^{\dot{M}_y\times\dot{M}_y}$, $\mathbf{Z} \in \mathbb{R}^{M\times M}$, define a mapping $\mathscr{F}_M$ such that

$$\mathscr{F}_M(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \mathbf{I}_{\ddot{M}} + \eta_x\left[\mathbf{D}_+\hat{\mathscr{B}}_M(\mathbf{X}) + \mathbf{D}_-\hat{\mathscr{B}}_M^{\mathrm{T}}(\mathbf{X})\right] + \eta_y\left[\mathbf{E}_+\check{\mathscr{B}}_M(\mathbf{Y}, \mathbf{Z}) + \mathbf{E}_-\check{\mathscr{B}}_M^{\mathrm{T}}(\mathbf{Y}, \mathbf{Z})\right],$$

where

$$\hat{\mathscr{B}}_M(\mathbf{X}) = \begin{bmatrix} \mathbf{X}^{(l,u)} \otimes \mathbf{I}_{\dot{M}_y} & \mathbf{X}^{(m,u)} \otimes \bar{\mathbf{I}}_M & \mathbf{X}^{(r,u)} \otimes \mathbf{I}_{\dot{M}_y} \\ \mathbf{X}^{(l,m)} \otimes \bar{\mathbf{I}}_M^{\mathrm{T}} & \mathbf{X}^{(m,m)} \otimes \mathbf{I}_M & \mathbf{X}^{(r,m)} \otimes \bar{\mathbf{I}}_M^{\mathrm{T}} \\ \mathbf{X}^{(l,d)} \otimes \mathbf{I}_{\dot{M}_y} & \mathbf{X}^{(m,d)} \otimes \bar{\mathbf{I}}_M & \mathbf{X}^{(r,d)} \otimes \mathbf{I}_{\dot{M}_y} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{X}^{(l,u)} & \mathbf{X}^{(m,u)} & \mathbf{X}^{(r,u)} \\ \mathbf{X}^{(l,m)} & \mathbf{X}^{(m,m)} & \mathbf{X}^{(r,m)} \\ \mathbf{X}^{(l,d)} & \mathbf{X}^{(m,d)} & \mathbf{X}^{(r,d)} \end{bmatrix},$$

$$\mathbf{X}^{(l,u)}, \mathbf{X}^{(r,u)}, \mathbf{X}^{(l,d)}, \mathbf{X}^{(r,d)} \in \mathbb{R}^{M\times M}, \qquad \mathbf{X}^{(m,u)}, \mathbf{X}^{(m,d)} \in \mathbb{R}^{M\times\bar{M}_x}, \qquad \mathbf{X}^{(l,m)}, \mathbf{X}^{(r,m)} \in \mathbb{R}^{\bar{M}_x\times M},$$

$$\mathbf{X}^{(m,m)} \in \mathbb{R}^{\bar{M}_x\times\bar{M}_x}, \qquad \check{\mathscr{B}}_M(\mathbf{Y}, \mathbf{Z}) = \mathrm{blockdiag}(\mathbf{I}_M \otimes \mathbf{Y}, \mathbf{I}_{\bar{M}_x} \otimes \mathbf{Z}, \mathbf{I}_M \otimes \mathbf{Y}).$$

Table 11
Results of MGMBS(1), MGMJS and $\mathbf{P}_b$ when $N = 1$ and the scheme (4.5) is used.

(α, β)	$\dot{M}_y + 1$	MGMBS(1)			MGMJS			$\mathbf{P}_b$		
		iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$
(1.1, 1.5)	2^7	22.0	0.80 s	3.09e−2	124.0	3.67 s	3.09e−2	43.0	0.82 s	3.09e−2
	2^8	32.0	3.17 s	1.40e−2	203.0	18.13 s	1.40e−2	69.0	6.36 s	1.40e−2
	2^9	42.0	23.53 s	1.52e−2	276.0	150.30 s	1.52e−2	109.0	118.46 s	1.52e−2
(1.5, 1.5)	2^7	11.0	0.36 s	1.67e−2	33.0	0.97 s	1.67e−2	63.0	1.31 s	1.67e−2
	2^8	12.0	1.25 s	6.37e−3	38.0	3.39 s	6.37e−3	110.0	12.87 s	6.37e−3
	2^9	13.0	8.08 s	9.22e−3	42.0	22.28 s	9.22e−3	194.0	341.24 s	9.22e−3
(1.6, 1.9)	2^7	10.0	0.31 s	3.91e−3	111.0	3.21 s	3.91e−3	75.0	1.73 s	3.91e−3
	2^8	11.0	1.17 s	5.03e−3	–	–	–	135.0	17.53 s	5.03e−3
	2^9	11.0	6.59 s	6.39e−3	–	–	–	244.0	498.76 s	6.39e−3

Table 12
Results of MGMBS(1), MGMJS and $\mathbf{P}_b$ when $N = 1$ and the scheme (4.7) is used.

(α, β)	$\dot{M}_y + 1$	MGMBS(1)			MGMJS			$\mathbf{P}_b$		
		iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$
(1.6, 1.6)	2^7	11.0	0.37 s	1.30e−2	29.0	0.85 s	1.30e−2	58.0	1.20 s	1.30e−2
	2^8	12.0	1.18 s	1.22e−2	34.0	2.91 s	1.22e−2	104.0	11.54 s	1.22e−2
	2^9	12.0	7.35 s	1.20e−2	615.0	339.47 s	1.20e−2	186.0	303.58 s	1.20e−2
	2^{10}	13.0	38.03 s	1.20e−2	–	–	–	*	*	*
(1.75, 1.75)	2^7	9.0	0.31 s	9.56e−3	32.0	0.98 s	9.56e−3	79.0	1.82 s	9.56e−3
	2^8	10.0	1.03 s	8.76e−3	163.0	14.97 s	8.76e−3	147.0	20.36 s	8.76e−3
	2^9	10.0	6.57 s	8.57e−3	–	–	–	276.0	659.40 s	8.57e−3
(1.9, 1.9)	2^7	9.0	0.28 s	7.41e−3	43.0	1.31 s	7.41e−3	105.0	2.89 s	7.41e−3
	2^8	9.0	0.94 s	6.59e−3	–	–	–	205.0	35.49 s	6.59e−3
	2^9	9.0	5.43 s	6.40e−3	–	–	–	430.0	883.24 s	6.40e−3

By straightforward calculation, it is easy to check that $\tilde{\mathbf{A}} = \mathscr{F}_M(\mathbf{G}_{\alpha,\dot{M}_x}, \mathbf{G}_{\beta,\dot{M}_y}, \mathbf{G}_{\beta,M})$.

Now, we consider constructing banded preconditioner for $\tilde{\mathbf{A}}$. Let $\mathbf{G}^{(c)}_{\beta,M}$, $\mathbf{G}^{(c)}_{\beta,\dot{M}_y}$ and $\mathbf{G}^{(c)}_{\alpha,\dot{M}_x}$ be diagonal-compensated banded truncations of $\mathbf{G}_{\beta,M}$, $\mathbf{G}_{\beta,\dot{M}_y}$ and $\mathbf{G}_{\alpha,\dot{M}_x}$, respectively [9]. Here, the bandwidths of the $\mathbf{G}^{(c)}_{\beta,M}$, $\mathbf{G}^{(c)}_{\beta,\dot{M}_y}$ and $\mathbf{G}^{(c)}_{\alpha,\dot{M}_x}$ are l, $2l$ and $3l$, respectively, with $l = \log_2(M+1)$. Then, we obtain a banded preconditioner $\mathbf{P}_b = \mathbf{L}_b\mathbf{U}_b$ such that $\mathbf{L}_b\mathbf{U}_b$ is the incomplete LU factorization with no fill-in (ILU(0)) of $\tilde{\mathbf{A}}^{(c)} = \mathscr{F}_M\left(\mathbf{G}^{(c)}_{\alpha,\dot{M}_x}, \mathbf{G}^{(c)}_{\beta,\dot{M}_y}, \mathbf{G}^{(c)}_{\beta,M}\right)$. The results of $\mathbf{P}_b$, MGMBS(1) and MGMJS with $\nu = 2$ for solving (5.3)–(5.5) are listed in Table 11 for the first-order discretization scheme (4.5) and in Table 12 for the second-order discretization scheme (4.7).

'−' and '∗' denote divergence and running out of memory, respectively. From Tables 11–12, we note that even for the smallest N (i.e., $N = 1$), $E_{N,M}$ are still small, which suggests that (5.7), the uniform-grid discretization of SFDE on U-shape domain is actually applicable. Clearly, both CPU cost and iteration number of MGMBS(1) are significantly smaller than the other two solvers, which means that MGMBS(1) is far more efficient than the other two solvers for solving the SFDE on U-shape domain. Note also that iteration number of MGMJS changes drastically and it even diverges in all cases but $(\alpha, \beta) = (1.1, 1.5)$, $(1.5, 1.5)$ of Tables 11–12 for large $\dot{M}_y$. This again implies the positivity of the bandwidth ω is useful to not only significantly improving efficiency of MGM but also reversing divergence of MGMJS.

Remark. Note that in Tables 3–12, iteration number of any solver mentioned above always has an evident tendency to increase as M increases when α is close to 1 and $|\alpha - \beta|$ is large; say, the case of $(\alpha, \beta) = (1.1, 1.5)$. Such a case may lead to the SFDE becoming an anisotropic problem, for which the multigrid method usually does not work well; see, for instance, [6]. Nevertheless, our proposed V-cycle MGM still has a better numerical performance compared with other solvers.

6. Concluding remarks

In this paper, we have proposed and studied a V-cycle MGM with the proposed banded smoother as a fast solver for the linear systems arising from uniform-grid discretization of two-dimensional time-dependent SFDEs on rectangular and non-rectangular domains. Complexity analysis shows that one iteration of MGMBS(ω) requires only $\mathcal{O}(M^2 \log M)$ operations and $\mathcal{O}(M^2)$ storage. Theoretically, we prove the convergence of the proposed banded smoother in the sense of infinity norm. Moreover, a number of numerical results in Section 5 show that the total operations cost and the total storage requirement of MGMBS(1) for recursively solving the N linear systems are of $\mathcal{O}(NM^2 \log M)$ and $\mathcal{O}(M^2)$, respectively when $|\alpha - \beta|$ is small. Moreover, via comparing MGMBS(1) with other solvers, it shows that MGMBS(1) is significantly more efficient than the other tested solvers. Also, with comparison between MGMBS(1) and MGMJS, it demonstrates that the positivity of ω is of great importance for MGM to remarkably improve its efficiency for solving SFDE problems and remedy divergence of MGMJS. We will consider a rigorous proof of convergence of MGMBS(ω), modifying the performance of our proposed method in the possibly anisotropic case (e.g., $(\alpha, \beta) = (1.1, 1.5)$) as our future research work.

Acknowledgements

The authors would like to thank the anonymous referees for their useful suggestions and comments that improved the presentation of this paper.

References

[1] O. Agrawal, Solution for a fractional diffusion-wave equation defined in a bounded domain, Nonlinear Dyn. 29 (2002) 145–155.
[2] J. Bouchaud, A. Georges, Anomalous diffusion in disordered media: statistical mechanisms, models and physical applications, Phys. Rep. 195 (1990) 127–293.
[3] W. Briggs, S. McCormick, A Multigrid Tutorial, SIAM, Philadelphia, 1987.
[4] R. Chan, X. Jin, A family of block preconditioners for block systems, SIAM J. Sci. Stat. Comput. 13 (1992) 1218–1235.
[5] R. Chan, M. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev. 38 (1996) 427–482.
[6] W. Hackbusch, Multi-Grid Methods and Applications, Springer Science and Business Media, 2013.
[7] Z. Hao, Z. Sun, W. Cao, A fourth-order approximation of fractional derivatives with its applications, J. Comput. Phys. 281 (2015) 787–805.
[8] R. Hilfer (Ed.), Applications of Fractional Calculus in Physics, World Scientific, Singapore, 2000.
[9] X. Jin, F. Lin, Z. Zhao, Preconditioned iterative methods for two-dimensional space-fractional diffusion equations, Commun. Comput. Phys. 18 (2015) 469–488.
[10] S. Lei, Y. Huang, Fast algorithms for high-order numerical methods for space-fractional diffusion equations, Int. J. Comput. Math. (2017), http://dx.doi.org/10.1080/00207160.2016.1149579, in press.
[11] Q. Liu, F. Liu, Y. Gu, P. Zhuang, J. Chen, I. Turner, A meshless method based on Point Interpolation Method (PIM) for the space fractional diffusion equation, Appl. Math. Comput. 256 (2015) 930–938.
[12] M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math. 56 (2006) 80–90.
[13] R. Metzler, J. Klafter, The random walk's guide to anomalous diffusion: a fractional dynamics approach, Phys. Rep. 339 (2000) 1–77.
[14] H. Pang, H. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. 231 (2012) 693–703.
[15] J. Pan, R. Ke, M. Ng, H. Sun, Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations, SIAM J. Sci. Comput. 36 (2014) A2698–A2719.
[16] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.
[17] T. Solomon, E. Weeks, H. Swinney, Observation of anomalous diffusion and Lëvy flights in a 2-dimensional rotating flow, Phys. Rev. 71 (1993) 3975–3979.
[18] E. Sousa, C. Li, A weighted finite difference method for the fractional diffusion equation based on the Riemann–Liouville derivative, Appl. Numer. Math. 90 (2015) 22–37.
[19] H. Sun, X. Jin, Q. Chang, Convergence of the multigrid method for ill-conditioned block Toeplitz systems, BIT 41 (2001) 179–190.
[20] W. Tian, H. Zhou, W. Deng, A class of second order difference approximations for solving space fractional diffusion equations, Math. Comput. 84 (2015) 1703–1727.
[21] U. Trottenberg, C. Oosterlee, A. Schülller, Multigrid, Academic Press, New York, 2001.