Journal of Computational Physics 323 (2016) 204–218

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A fast accurate approximation method with multigrid solver for two-dimensional fractional sub-diffusion equation [☆]

Xue-lei Lin [a], Xin Lu [a], Micheal 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 22 May 2016
Received in revised form 21 July 2016
Accepted 23 July 2016
Available online 27 July 2016

Keywords:
Block lower triangular Toeplitz matrix
Block ϵ-circulant approximation
Multigrid method
Fractional sub-diffusion equations

ABSTRACT

A fast accurate approximation method with multigrid solver is proposed to solve a two-dimensional fractional sub-diffusion equation. Using the finite difference discretization of fractional time derivative, a block lower triangular Toeplitz matrix is obtained where each main diagonal block contains a two-dimensional matrix for the Laplacian operator. Our idea is to make use of the block ϵ-circulant approximation via fast Fourier transforms, so that the resulting task is to solve a block diagonal system, where each diagonal block matrix is the sum of a complex scalar times the identity matrix and a Laplacian matrix. We show that the accuracy of the approximation scheme is of $O(\epsilon)$. Because of the special diagonal block structure, we employ the multigrid method to solve the resulting linear systems. The convergence of the multigrid method is studied. Numerical examples are presented to illustrate the accuracy of the proposed approximation scheme and the efficiency of the proposed solver.

1. Introduction

Consider an initial-boundary value problem of two-dimensional fractional sub-diffusion equation (FSDE) [5,24]:

$$ {}_{0}^{\mathcal{C}}\mathcal{D}_t^{\alpha} u = \nabla\cdot(p(x,y)\nabla u) + f(x,y,t),\ (x,y)\in\Omega,\ 0\le t\le T, \tag{1.1}$$

$$ u(x,y,t) = \phi(x,y,t),\ (x,y)\in\partial\Omega,\ 0\le t\le T, \tag{1.2}$$

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

where $\Omega=(x_L,x_R)\times(y_L,y_R)$ is a rectangular domain, $\nabla\cdot(p(x,y)\nabla u)$ is the elliptic operator, $p(x,y)$ is a smooth positive function such that $\forall(x,y)\in\Omega$, $p(x,y)\ge p_0>0$ with p_0 being a constant, $\partial\Omega$ is the boundary, $f(x,y,t)$ is the source term, ${}_{0}^{\mathcal{C}}\mathcal{D}_t^{\alpha} u$ is the Caputo's derivative of order α $(0<\alpha<1)$ with respect to t defined by

$$ {}_{0}^{\mathcal{C}}\mathcal{D}_t^{\alpha} u(x,y,t) = \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{\partial u(x,y,s)}{\partial s}(t-s)^{-\alpha}ds, \tag{1.4}$$

☆ This research was supported by research grants MYRG2016-00063-FST from University of Macau, 054/2015/A2 from FDCT of Macao, HKRGC GRF HKBU12301214 and HKBY FRG grant.

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

http://dx.doi.org/10.1016/j.jcp.2016.07.031
0021-9991/© 2016 Elsevier Inc. All rights reserved.

with $\Gamma(x)$ denoting the gamma function, $\phi(x, y, t)$ and $\psi(x, y)$ are the given Dirichlet boundary condition and initial condition, respectively.

The FSDE is a class of fractional differential equation, which has been widely and successfully used in modeling of description of fractional random walk, anomalous diffusive systems, unification of diffusion and wave propagation phenomenon; see [1,4,12,17–19]. Since analytical solutions to FSDEs are often unavailable, many numerical schemes are proposed for solving sub-diffusion problems (see [5–7,21,22,24,25]). The fractional differential operators are nonlocal, which leads to a character of history dependence and universal mutuality (see [24]). Hence, the computational cost is too expensive for solving the discrete problems obtained from the FSDE. This motivates us to develop fast algorithm for numerical schemes. More precisely, the associated coefficient matrix having $N \times N$ blocks is usually of block lower triangular Toeplitz (BLTT) structure with each block being of size $M^2 \times M^2$ when certain numerical scheme is applied on the FSDE (1.1)–(1.3). In general, direct solvers for the BLTT linear system are often time consuming. For example, block forward substitution method [8] as a direct solver requires at least $\mathcal{O}(N^2M^2 + NM^4)$ operations. When N or M is large, the number of operations for direct solvers becomes very large. Therefore, direct solvers may not be considered when the matrix size is large.

Block forward substitution method combined with some efficient iterative solvers for BLTT linear systems may reduce the complexity to $\mathcal{O}(N^2M^2)$ operations and only require $\mathcal{O}(NM^2)$ storage. Alternatively, Zhang and Sun in [24] proposed the alternating direction implicit schemes for solving high dimensional FSDE whose resulting BLTT linear system can be directly solved with $\mathcal{O}(N^2M^2)$ operations and $\mathcal{O}(NM^2)$ storage requirement. Nevertheless, their method is only available for $p(x, y)$ being a constant. Meanwhile, it has lower temporal accuracy compared with the non-ADI scheme. Even so, both of the above mentioned methods are still expensive, when N is large.

In order to reduce the computational cost, recently, Lu, Pang and Sun [15] proposed an approximate inversion method (AIM) for solving BLTT linear systems, which can be applied to the one dimensional FSDEs. More precisely, the corresponding BLTT matrix is firstly approximated by the block ϵ-circulant matrix [3,14], which can be block diagonalized via fast Fourier transform (FFT). Each block is a tri-diagonal matrix for the one dimensional case. Therefore, they can be inverted easily. The accuracy of the approximation scheme is shown to be of $O(\epsilon)$ under some sufficient conditions.

In this paper, we extend the AIM to solve the two-dimensional case. As in [15], the resulting discretized BLTT matrix is approximated by the block ϵ-circulant matrix via the FFT. Unlike that in [15], however, each block is no longer a tri-diagonal matrix since it is from the two dimensional problem. Indeed, to the two dimensional case, after block diagonalizing by the FFT, the resulting diagonal block matrix is the sum of a complex scalar times the identity matrix and a Laplacian matrix. Therefore, it would be extensive to invert it directly. To lower the computational workload, we propose to exploit the multigrid method (MGM) to solve the complex scalar shifted Laplacian linear systems, and establish the convergence of the corresponding multigrid solver. We also investigate the resulting BLTT matrix to satisfy the condition which can guarantee the accuracy of approximation to be of $O(\epsilon)$.

The proposed algorithm consists of two parts. The first part is for block diagonalization. The second part is for multigrid solvers. The computational cost is of $\mathcal{O}(M^2N\log N)$ operations and the storage cost is of $\mathcal{O}(NM^2)$ storage, respectively. Numerical examples are presented to illustrate the accuracy of the proposed approximation scheme and the efficiency of the proposed multigrid solver.

The rest of this paper is organized as follows. In Section 2, we propose the approximation method for solving (1.1)–(1.3) and study the accuracy of the method. In Section 3, we use the multigrid method for solving linear systems generated by the approximation method, and analyze the convergence of MGM. In Section 4, experimental results are presented to show the accuracy and efficiency of the proposed method. Finally, some concluding remarks are given in Section 5.

2. The approximation method

In this section, we propose the AIM for solving (1.1)–(1.3) and give a sufficient condition to guarantee a high accuracy of the AIM.

For a positive integer N, let $\tau = T/N$, $t_n = n\tau$ $(0 \le n \le N)$. Define the time-grid $\{t_n | 0 \le n \le N\}$ for discretization of $[0, T]$, $\{\mathbf{u}^n = u(\cdot, t_n) | 0 \le n \le N\}$. For a given function $w(t)$ defined on $t \in [0, T]$, define grid function $\{w^n = w(t_n) | 0 \le n \le N\}$. Without loss of generality, we assume the approximation to ${}_0^C\mathcal{D}_t^\alpha w|_{t=t_n}$ to be

$${}_0^C\mathcal{D}_t^\alpha w|_{t=t_n} \approx D_\tau^\alpha w^n := \sum_{i=1}^{n} g_{n-i}^{(\alpha)} w^i + g^{(n,\alpha)} w^0, \quad 1 \le n \le N, \tag{2.1}$$

where $g_i^{(\alpha)}$ $(i = 0, 1, \ldots, N)$, $g^{(i,\alpha)}$ $(i = 1, \ldots, N)$ are constants dependent on α, i, and N, which vary in different finite difference schemes. We denote by $\mathbf{B} \in \mathbb{R}^{M^2 \times M^2}$, the discretization of the elliptic operator $-\nabla\cdot(p(x, y)\nabla u)$. We assume that there are M^2 spatial unknowns to be determined. Applying (2.1) and $\mathbf{B}$ to the FSDE (1.1)–(1.3), we obtain a BLTT linear system as follows:

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{2.2}$$

where $\mathbf{u} = (\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^N)$ is the unknown to be solved, $\mathbf{b}$ is a given vector containing information about f, ϕ and ψ on grid points,

$$\mathbf{A} = \mathbf{T}_N \otimes \mathbf{I}_{M^2} + \mathbf{I}_N \otimes \left(g_0^{(\alpha)} \mathbf{I}_{M^2} + \mathbf{B} \right),$$

where $\mathbf{I}_k$ denotes the $k \times k$ identity matrix, $\mathbf{T}_N$ is a lower triangular Toeplitz with its first column being $\left(0, g_1^{(\alpha)}, g_2^{(\alpha)}, ..., g_{N-1}^{(\alpha)}\right)^{\mathrm{T}}$, "$\otimes$" denotes the Kronecker product.

Let ϵ be a small positive number. The BLTT matrix $\mathbf{A}$ can be approximated by [15]

$$\mathbf{A}_\epsilon = \mathbf{A} + \epsilon \tilde{\mathbf{T}}_N \otimes \mathbf{I}_{M^2}, \tag{2.3}$$

where $\tilde{\mathbf{T}}_N$ is an upper triangular Toeplitz matrix with its first row being $(0, g_{N-1}^{(\alpha)}, g_{N-2}^{(\alpha)}, ..., g_1^{(\alpha)})$. It is interesting to point out that $\mathbf{A}_\epsilon$ is called a *block ϵ-circulant matrix*.

Let $\mathbf{D}_\delta = \mathrm{diag}(1, \delta, ..., \delta^{N-1})$ with $\delta = \sqrt[N]{\epsilon}$; the $N \times N$ Fourier transformation matrix be given by

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

We note that the block ϵ-circulant matrix $\mathbf{A}_\epsilon$ can be block diagonalized by means of a combination of $\mathbf{F}_N$ and $\mathbf{D}_\delta$ as following [15]

$$\mathbf{A}_\epsilon = \left[(\mathbf{D}_\delta^{-1} \mathbf{F}_N^*) \otimes \mathbf{I}_{M^2} \right] \mathrm{diag}\left(\mathbf{\Lambda}_0, \mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_{N-1} \right) \left[(\mathbf{F}_N \mathbf{D}_\delta) \otimes \mathbf{I}_{M^2} \right], \tag{2.4}$$

where

$$\mathbf{\Lambda}_k = \mathbf{B} + \left(\sum_{j=0}^{N-1} \delta^j g_j^{(\alpha)} \omega^{kj} \right) \mathbf{I}_{M^2}, \quad k = 0, 1, \ldots, N-1. \tag{2.5}$$

From (2.3) and (2.4), we obtain

$$\mathbf{u}_\epsilon = \mathbf{A}_\epsilon^{-1} \mathbf{b} = \left[(\mathbf{D}_\delta^{-1} \mathbf{F}_N^*) \otimes \mathbf{I}_{M^2} \right] \mathrm{diag}\left(\mathbf{\Lambda}_0^{-1}, \mathbf{\Lambda}_1^{-1}, \ldots, \mathbf{\Lambda}_{N-1}^{-1} \right) \left[(\mathbf{F}_N \mathbf{D}_\delta) \otimes \mathbf{I}_{M^2} \right] \mathbf{b} \tag{2.6}$$

as an approximation to $\mathbf{u}$ by replacing $\mathbf{A}$ in (2.2) with $\mathbf{A}_\epsilon$, where the nonsingularity of $\mathbf{A}_\epsilon$ is guaranteed by Theorem 1.

In order to measure how well the approximation $\mathbf{u}_\epsilon$ is to $\mathbf{u}$, the exact solution of (2.2), we refer to theoretical results in [15] which are stated as following.

Theorem 1. *(See [15, Corollary 10]) Assume*

$$g_0^{(\alpha)} \geq \sum_{j=1}^{N-1} \left| g_j^{(\alpha)} \right|, \qquad \max_{1 \leq j \leq N-1} g_j^{(\alpha)} \leq 0, \tag{2.7}$$

$$\mathbf{B} \textit{ is a nonsingular } M\textit{-matrix}, \tag{2.8}$$

where definition of the terminology "nonsingular M-matrix" is given in [15]. Then,

i) *the block ϵ-circulant matrix $\mathbf{A}_\epsilon$ is invertible for $0 < \epsilon < 1$; and*
ii) *there exists a nonnegative constant c depending only on $\mathbf{A}$ such that*

$$\frac{\left\| \mathbf{A}_\epsilon^{-1} - \mathbf{A}^{-1} \right\|_\infty}{\left\| \mathbf{A}^{-1} \right\|_\infty} \leq [1 + c(1+\epsilon)]\epsilon = \mathcal{O}(\epsilon),$$

iii) *if $\mathbf{u}$ is a nonzero vector, then*

$$\frac{||\mathbf{u} - \mathbf{u}_\epsilon||_\infty}{||\mathbf{u}||_\infty} \leq \epsilon[1 + c(1+\epsilon)]\kappa(\mathbf{A}),$$

where $\kappa(\mathbf{A}) = ||\mathbf{A}^{-1}||_\infty ||\mathbf{A}||_\infty$ denotes the condition number of $\mathbf{A}$.

Remark. Theorem 1 shows that if the assumptions (2.7)–(2.8) hold and the condition number $\kappa(\mathbf{A})$ is not too big, $\mathbf{u}_\epsilon$ is a good approximation to $\mathbf{u}$ whenever ϵ is small enough. Indeed, the assumption (2.7) is valid for two most popular difference schemes. One is L_1 approximation (see [24]). The other is $L2\text{-}1_\sigma$ approximation (see [2]). Moreover, the assumption (2.8) is valid for a number of numerical methods; for examples, central difference discretizations on nonuniform grids, high order finite difference methods [9,16], finite element methods [11,20], and high order compact difference schemes [13, 23]. Furthermore, the assumptions (2.7) and (2.8) imply the invertibility of $\mathbf{A}$, hence the AIM for solving (1.1)–(1.3) makes sense. In the rest of this paper, we always assume that (2.7) and (2.8) hold. On the other hand, theoretically, the condition number $\kappa(\mathbf{A})$ in **iii)** of Theorem 1 may depend on the size of $\mathbf{A}$ and which discretization scheme is applied. However, numerical results in Section 4 show that $\mathbf{u}_\epsilon$ approximates $\mathbf{u}$ very well and the matrix size does not affect the accuracy of the approximation.

The proposed algorithm for (2.6) can be written into the following four steps:

$$\textit{Step1} \quad \textit{Compute } \tilde{\mathbf{b}} = \left[(\mathbf{F}_N\mathbf{D}_\delta)\otimes \mathbf{I}_{M^2}\right]\mathbf{b}, \tag{2.9}$$

$$\textit{Step2} \quad \textit{Compute } \mathbf{\Lambda}_k \textit{ for } k=0,1,\ldots,N-1 \textit{ by using } (2.5), \tag{2.10}$$

$$\textit{Step3} \quad \textit{Solve } \mathbf{\Lambda}_{k-1}\tilde{\mathbf{u}}_\epsilon^k = \tilde{\mathbf{b}}^k \textit{ for } \tilde{\mathbf{u}}_\epsilon^k,\ k=1,2,\ldots,N,\ \textit{with } \tilde{\mathbf{b}} = \left((\tilde{\mathbf{b}}^1)^{\mathrm{T}},(\tilde{\mathbf{b}}^2)^{\mathrm{T}},\ldots,(\tilde{\mathbf{b}}^N)^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{2.11}$$

$$\textit{Step4} \quad \textit{Compute } \mathbf{u}_\epsilon = \left[(\mathbf{D}_\delta^{-1}\mathbf{F}_N^*)\otimes \mathbf{I}_{M^2}\right]\tilde{\mathbf{u}}_\epsilon, \quad \textit{with } \tilde{\mathbf{u}}_\epsilon = \left((\tilde{\mathbf{u}}_\epsilon^1)^{\mathrm{T}},(\tilde{\mathbf{u}}_\epsilon^2)^{\mathrm{T}},\ldots,(\tilde{\mathbf{u}}_\epsilon^N)^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{2.12}$$

Rewrite $\mathbf{b}$ as $\mathbf{b} = \left((\mathbf{b}^1)^{\mathrm{T}},(\mathbf{b}^2)^{\mathrm{T}},\ldots,(\mathbf{b}^N)^{\mathrm{T}}\right)^{\mathrm{T}}$ with $\mathbf{b}^k \in \mathbb{R}^{M^2\times 1}$ $(1\le k\le N)$. It is easy to check that (2.9) and (2.12) are equivalent to

$$\tilde{\mathbf{b}}^{k+1} = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\delta^j\omega^{kj}\mathbf{b}^{j+1}, \quad k=0,1,\ldots,N-1, \tag{2.13}$$

$$\mathbf{u}_\epsilon^{k+1} = \frac{\mathbf{D}_\delta^{-1}}{\sqrt{N}}\sum_{j=0}^{N-1}\omega^{-kj}\tilde{\mathbf{u}}_\epsilon^{j+1}, \quad k=0,1,\ldots,N-1, \tag{2.14}$$

respectively, where $\mathbf{u}_\epsilon = \left((\mathbf{u}_\epsilon^1)^{\mathrm{T}},(\mathbf{u}_\epsilon^2)^{\mathrm{T}},\ldots,(\mathbf{u}_\epsilon^N)^{\mathrm{T}}\right)^{\mathrm{T}}$. From (2.13), (2.5) and (2.14), we see that (2.9), (2.10) and (2.12) can be implemented with $\mathcal{O}(NM^2\log N)$ operations and $\mathcal{O}(NM^2)$ storage requirement by using FFTs. Hence, (2.9)–(2.12) for solving the FSDE (1.1)–(1.3) requires $\mathcal{O}(NM^2\log N)+\mathscr{R}(M,N)$ operations and $\mathcal{O}(NM^2)+\mathscr{J}(M,N)$ storage, where $\mathscr{R}(M,N)$ and $\mathscr{J}(M,N)$ denote number of operations and storage required by (2.11), respectively. It is noticeable that

$$\bar{\mathbf{\Lambda}}_k = \mathbf{B} + \left(\sum_{j=0}^{N-1}\delta^j\omega^{-kj}g_j^{(\alpha)}\right)\mathbf{I}_{M^2} = \mathbf{B} + \left(\sum_{j=0}^{N-1}\delta^j\omega^{(N-k)j}g_j^{(\alpha)}\right)\mathbf{I}_{M^2} = \mathbf{\Lambda}_{N-k}, \quad 1\le k\le N-1,$$

$$\bar{\tilde{\mathbf{b}}}^{k+1} = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\delta^j\omega^{-kj}\mathbf{b}^{j+1} = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\delta^j\omega^{(N-k)j}\mathbf{b}^{j+1} = \tilde{\mathbf{b}}^{N-k+1}, \quad 1\le k\le N-1.$$

That means $\tilde{\mathbf{u}}_\epsilon^{k+1}$ is just complex conjugate of $\tilde{\mathbf{u}}_\epsilon^{N-k+1}$ respectively for $1\le k\le N-1$. Hence, we only need to solve first half of the linear systems in (2.11), which significantly reduces $\mathscr{R}(M,N)$ and $\mathscr{J}(M,N)$. In the next section, we propose the MGM to solve (2.11). And it is shown in Section 3 that both $\mathscr{R}(M,N)$ and $\mathscr{J}(M,N)$ are of $\mathcal{O}(NM^2)$ by the MGM. Thus, we in advance conclude that the AIM with the MGM (AIMGM) requires only $\mathcal{O}(NM^2\log N)$ operations and $\mathcal{O}(NM^2)$ storage for solving (1.1)–(1.3).

3. The multigrid solver

In this section, we study the MGM to solve a sequence of linear systems

$$\mathbf{\Lambda}_{k-1}\tilde{\mathbf{u}}_\epsilon^k = \tilde{\mathbf{b}}^k, \quad k=0,1,\ldots,\lfloor N/2\rfloor+1,$$

where $\lfloor\cdot\rfloor$ denotes the floor function. For simplicity, we neglect the subscripts and rewrite these linear systems as follows:

$$\mathbf{\Lambda}\mathbf{x} = \mathbf{y}, \tag{3.1}$$

where $\mathbf{y}$ denotes $\tilde{\mathbf{b}}^k$ for some k. According to (2.5), $\mathbf{\Lambda}$ can be written as

$$\mathbf{\Lambda} = \mathbf{B} + \gamma\mathbf{I}_{M^2}, \tag{3.2}$$

where γ denotes $\sum_{j=0}^{N-1}\delta^j g_j^{(\alpha)}\omega^{kj}$ for some k. Since $\mathbf{\Lambda}$ is of size $M^2\times M^2$, $\mathbf{\Lambda}$ can be a huge matrix if M is large. Moreover, $\mathbf{\Lambda}$ has complex diagonal entries. We may consider to solve iterative solvers by solving the normalized equations of (3.1) or transforming (3.1) into a larger real linear system. The disadvantage is that the condition number of the corresponding linear system may become larger which leads a slower convergence rate.

In this paper, we study the structure of $\mathbf{\Lambda}$ and note that $\mathbf{\Lambda}$ is a shifted discrete elliptic operator with complex shift γ. Meanwhile,

$$\Re(\gamma) = \frac{1}{2}(\gamma + \bar{\gamma}) = \sum_{j=0}^{N-1} \delta^j \cos(\frac{2\pi kj}{N}) g_j^{(\alpha)} \geq g_0^{(\alpha)} - \sum_{j=1}^{N-1} \left| g_j^{(\alpha)} \right| \geq 0, \quad 0 \leq k \leq N-1, \tag{3.3}$$

where $\Re(\cdot)$ denotes real part of a complex number. That means condition number of $\mathbf{\Lambda}$ does not exceed that of $\mathbf{B}$. We make use of multigrid approach to solve the linear system in (3.1) directly.

We denote by $\mathbf{I}_h^H$, $\mathbf{I}_H^h$ and $\mathscr{S}$, restriction operator, interpolation operator and smoother. Define a sequence of spatial grids-size such that $M_2 < M_3 < \cdots < M_l = M$, where $M_i = 2^i - 1$ and M_2 denotes that spatial grids-size at coarsest level. Also, we denote by $\mathbf{B}^{(i)}$, $\mathbf{B}$ defined on the $M_i \times M_i$ spatial grids. Let $\mathbf{\Lambda}^{(i)} = \mathbf{B}^{(i)} + \gamma \mathbf{I}_{M_i^2}$. Then, one iteration of MGM is given by

Algorithm 1 Single step of MGM(ν, q).

```
function $\mathbf{u}^h$ =MGM$(i, \mathbf{u}_0, \mathbf{f}^h, q)$
if $i == 2$ then
    $\mathbf{u}^h = (\mathbf{\Lambda}^{(i)})^{-1}\mathbf{f}^h$;
    return $\mathbf{u}^h$;
else
    iterate $\mathbf{u}^h = \mathscr{S}(\mathbf{\Lambda}^{(i)}, \mathbf{f}^h, \mathbf{u}^h)$ $\nu$ times with initial guess $\mathbf{u}_0$; %presmoothing
    $\mathbf{e} = \mathrm{MGM}(i-1, \mathbf{0}, \mathbf{I}_h^H(\mathbf{f}^h - \mathbf{\Lambda}^{(i)}\mathbf{u}^h), q)$; % $\mathbf{0}$ denotes zero initial guess
    if $i > 3$ then
        for $j = 2 : q$
            $\mathbf{e} = \mathrm{MGM}(i-1, \mathbf{e}, \mathbf{I}_h^H(\mathbf{f}^h - \mathbf{\Lambda}^{(i)}\mathbf{u}^h), q)$;
        end
    end if
    $\mathbf{u}^h = \mathbf{u}^h + \mathbf{I}_H^h\mathbf{e}$;%correction
    iterate $\mathbf{u}^h = \mathscr{S}(\mathbf{\Lambda}^{(i)}, \mathbf{f}^h, \mathbf{u}^h)$ $\nu$ times; %postsmoothing
    return $\mathbf{u}^h$;
end if
end
```

We furthermore define MGM(ν, q) iteration for solving the system (3.1) as follows:

Algorithm 2 MGM(ν, q) iteration.

```
Set : $\mathbf{u}^{(0)} = \mathbf{0}$; $\mathbf{r}^{(0)} = \mathbf{y}$;
do
     $\mathbf{u}^{(k)} = \mathrm{MGM}(l, \mathbf{u}^{(k-1)}, \mathbf{y}, q)$;
     $\mathbf{r}^{(k)} = \mathbf{y} - \mathbf{\Lambda} u^{(k)}$;        %compute current residual
until  $\frac{||\mathbf{r}^{(k)}||_\infty}{||\mathbf{r}^{(0)}||_\infty} < 10^{-8}$        %stopping criterion
```

Let us estimate the complexity of Algorithm 1 with different q and ν. We denote by $\mathcal{J}(M)$ and $\mathcal{R}(M)$, the amount of memory and number of operations required in Algorithm 1 respectively. Let J_i denotes the amount of memory required by Algorithm 1 at level i $(2 \leq i \leq l)$. Then, by Algorithm 1, we have

$$J_i \leq c_1 M_i^2, \quad i \geq 3, \quad J_2 = c_2, \tag{3.4}$$

where c_1 is a positive constant independent of l, i, ν, and q, c_2 is the amount of memory required by Algorithm 1 at the coarsest level. Moreover,

$$\inf_{i \geq 2} \frac{M_{i+1}^2}{M_i^2} \geq \inf_{i \geq 2} \frac{(2^{i+1}-2)^2}{(2^i - 1)^2} = 4. \tag{3.5}$$

Hence, we by (3.4) and (3.5) have

$$\begin{aligned} \mathcal{J}(M) &= \sum_{i=2}^{l} J_i \leq c_2 + c_1 \sum_{i=3}^{l} M_i^2 \\ &\leq c_2 + c_1 M_l^2 \sum_{i=3}^{l} \left(\frac{1}{4}\right)^{l-i} = c_2 + \frac{4(1-4^{2-l})c_1 M_l^2}{3} \leq c_2 + \frac{4c_1 M_l^2}{3} = \mathcal{O}(M^2). \end{aligned} \tag{3.6}$$

(3.6) shows that Algorithm 1 requires amount of memory proportional to number of unknowns in (3.1) for any positive ν and q.

We denote by $R_i(\nu, q)$ $(i \geq 2)$, the number of operations required by Algorithm 1 with $M = M_i$. Note also that ν times of smoothing iteration at level i $(i \geq 3)$ requires at most $c_3 \nu M_i^2$ operations with c_3 being independent of i, l, ν and q. By Algorithm 1, it holds following recursive inequalities,

$$R_2(\nu,q)=c_4,\quad R_3(\nu,q)\le c_3\nu N_3+R_2(\nu,q),\quad R_i(\nu,q)\le c_3\nu M_i^2+qR_{i-1}(\nu,q),\ i\ge 4,\tag{3.7}$$

where c_4 is the number of operations required at the coarsest level in Algorithm 1. By (3.5) and (3.7),

$$R_3(\nu,q)\le \frac{c_3\nu M_l^2}{4^{l-3}}+c_4,\qquad R_i(\nu,q)\le \frac{c_3\nu M_l^2}{4^{l-i}}+qR_{i-1}(\nu,q),\ 4\le i\le l.\tag{3.8}$$

Then, it follows trivially from applying induction to (3.8) that

$$R_l(\nu,q)\le q^{l-3}c_4+c_3\nu M_l^2\sum_{i=0}^{l-3}\left(\frac{q}{4}\right)^i\le\begin{cases}\frac{4c_3\nu M_l^2}{4-q}+q^{l-3}c_4, & 1\le q\le 3,\\ c_3\nu(l-2)M_l^2+q^{l-3}c_4, & q=4,\\ \frac{c_3\nu 4^{3-l}q^{l-2}M_l^2}{q-4}+q^{l-3}c_4, & q\ge 5,\end{cases}\qquad \text{for } l\ge 3.\tag{3.9}$$

With (3.9), we finally have following theorem.

Theorem 2. *Take $1\le q\le 3$. Then,*

$$\mathcal{R}(M)\le c_5\nu M^2,$$

where $c_5=4c_3+c_4$ is a constant.

Proof. It follows from (3.9) that

$$\mathcal{R}(M)\le\left(\frac{4c_3\nu M_l^2}{4-q}+q^{l-3}c_4\right)=\left(\frac{4c_3\nu}{4-q}+\frac{q^lc_4}{M_l^2q^3}\right)M_l^2\le\left[\frac{4c_3\nu}{4-q}+\frac{3^lc_4}{(2^l-1)^2q^3}\right]M_l^2\le c_5\nu M^2,$$

which completes the proof. □

In the next subsection, we analyze the convergence of MGM for this special complex scalar shifted Laplacian linear system. In particular, we consider and study central difference approximation to the elliptic operator.

3.1. The convergence analysis of two grid method

In this subsection, we first analyze the convergence of two grid method (TGM) [10].

We assume $p(x,y)\equiv\eta$ for some positive constant η and define

$$\mathbf{I}_h^H=\frac{1}{16}\mathbf{J}\otimes\mathbf{J},\quad \mathbf{I}_H^h=4(\mathbf{I}_h^H)^{\mathrm{T}},\quad \mathbf{J}=\begin{bmatrix}1&2&1&&&&&\\&&1&2&1&&&\\&&&&\ddots&&&\\&&&&&1&2&1\end{bmatrix},\tag{3.10}$$

$$\mathbf{B}=\text{blocktridiag}[\mathbf{H},\mathbf{G},\mathbf{H}],\ \mathbf{H}=-\eta h^{-2}\mathbf{I}_M,\ \mathbf{G}=\mu\eta h^{-2}\text{tridiag}[-1,2,-1]+2\eta h^{-2}\mathbf{I}_M,\tag{3.11}$$
$$h=(y_R-y_L)/(M+1),\ \mu=(y_R-y_L)^2/(x_R-x_L)^2.$$

Denote $\tilde{\mathbf{G}}=\mathbf{G}+\gamma\mathbf{I}_M$. Since $\mathbf{B}$ is defined, $\mathbf{\Lambda}$ is defined. Furthermore, we define $\mathscr{S}$ for (3.1) as the weighted block Jacobi (WBJ) smoother:

$$\mathbf{x}^{k+1}=\mathscr{S}(\mathbf{\Lambda},\mathbf{y},\mathbf{x}^k):=\mathbf{S}\mathbf{x}^k+\frac{1}{2}\left(\mathbf{I}_M\otimes\tilde{\mathbf{G}}^{-1}\right)\mathbf{y},\tag{3.12}$$

where $\mathbf{S}=\mathbf{I}_{M^2}-\frac{1}{2}\left(\mathbf{I}_M\otimes\tilde{\mathbf{G}}^{-1}\right)\mathbf{\Lambda}$ denotes the iteration matrix of $\mathscr{S}$, $\mathbf{x}^k$ is an initial guess. Similar to (3.12), we can define WBJ smoother for linear system with coefficient matrix $\mathbf{\Lambda}^{(i)}$ and arbitrarily given right hand sides. Therefore, the iteration matrix of TGM is given by [10]

$$\mathbf{T}(\nu):=\mathbf{S}^\nu\mathbf{K}_H^h\mathbf{S}^\nu,$$

where $\mathbf{K}_H^h$ is the correction operator defined by

$$\mathbf{K}_H^h=\mathbf{I}-\mathbf{I}_H^h\mathbf{\Lambda}_H^{-1}\mathbf{I}_h^H\mathbf{\Lambda},$$

with $\mathbf{\Lambda}_H$ denoting $\mathbf{\Lambda}$ on coarse grid and $\mathbf{I}$ denoting the identity. Hereinafter, we make notations:

$$\mathbb{K}=\{1,2,...,M\},\quad \mathbb{K}^r=\{1,2,...,\bar{M}-1\},\ \bar{M}=(M+1)/2,\quad \mathbb{K}^b=\mathbb{K}^r\cup\{\bar{M}\},$$
$$\theta_i=\frac{\pi i}{M+1},\qquad i'=M+1-i,\quad j'=M+1-j,\quad \forall i,j\in\mathbb{K}.$$

Define two dimensional sine transform as

$$\mathbf{Q}=\frac{2}{M+1}\left[\sin(i\theta_j)\right]_{i,j=1}^{M}\otimes\left[\sin(i\theta_j)\right]_{i,j=1}^{M}.$$

For a complex vector

$$\Xi=(\xi_{11},...,\xi_{M1},\xi_{12},...,\xi_{M2},...,\xi_{1M},...,\xi_{MM})^{\mathrm{T}}\in\mathbb{C}^{M^2\times 1},\tag{3.13}$$

where $\mathbb{C}^{m\times n}$ is the set of all $m\times n$ complex matrices, define the permutation matrix $\mathbf{P}$ such that

$$\mathbf{P}\Xi:=(\hat{\Xi}_{11},...,\hat{\Xi}_{\bar{M}1},\hat{\Xi}_{12},...,\hat{\Xi}_{\bar{M}2},......,\hat{\Xi}_{1\bar{M}},...,\hat{\Xi}_{\bar{M}\bar{M}})^{\mathrm{T}}\in\mathbb{C}^{M^2\times 1},$$

with

$$\hat{\Xi}_{ij}=(\xi_{ij},\xi_{ij'},\xi_{i'j},\xi_{i'j'}),\ (i,j)\in\mathbb{K}^r\times\mathbb{K}^r,\qquad \hat{\Xi}_{\bar{M}j}=(\xi_{\bar{M}j},\xi_{\bar{M}j'}),\ j\in\mathbb{K}^r$$
$$\hat{\Xi}_{i\bar{M}}=(\xi_{i\bar{M}},\xi_{i'\bar{M}}),\ i\in\mathbb{K}^r,\qquad \hat{\Xi}_{\bar{M}\bar{M}}=\xi_{\bar{M}\bar{M}}.$$

Lemma 3. $\mathbf{S}=\tilde{\mathbf{Q}}^{\mathrm{T}}\hat{\mathbf{S}}\tilde{\mathbf{Q}}$*, where* $\tilde{\mathbf{Q}}=\mathbf{PQ}$ *and*

$$\hat{\mathbf{S}}=\operatorname{diag}\left(\hat{S}_{11},...,\hat{S}_{\bar{M}1},\hat{S}_{12},...,\hat{S}_{\bar{M}2},......,\hat{S}_{1\bar{M}},...,\hat{S}_{\bar{M}\bar{M}}\right),$$

with

$$\hat{S}_{ij}=\operatorname{diag}(r_{ij},r_{ij'},r_{i'j},r_{i'j'}),\ (i,j)\in\mathbb{K}^r\times\mathbb{K}^r,\qquad \hat{S}_{\bar{M}j}=\operatorname{diag}(r_{\bar{M}j},r_{\bar{M}j'}),\ j\in\mathbb{K}^r,$$
$$\hat{S}_{i\bar{M}}=\operatorname{diag}(r_{i\bar{M}},r_{i'\bar{M}}),\ i\in\mathbb{K}^r,\qquad S_{\bar{M}\bar{M}}=r_{\bar{M}\bar{M}},$$
$$r_{ij}=\frac{1}{2}+\frac{\cos\theta_j}{4\mu\sin^2(\frac{\theta_i}{2})+\gamma^*+2},\qquad (i,j)\in\mathbb{K}\times\mathbb{K},\qquad \gamma^*=\eta^{-1}h^2\gamma.$$

Proof. $\forall\mathbf{e}\in\mathbb{C}^{M^2\times 1}$, we denote $\tilde{\mathbf{e}}=\mathbf{Se}$. Rewrite $\mathbf{e}$ and $\tilde{\mathbf{e}}$ in the form of

$$\mathbf{e}=[(\mathbf{e}_1)^{\mathrm{T}},(\mathbf{e}_2)^{\mathrm{T}},...,(\mathbf{e}_M)^{\mathrm{T}}]^{\mathrm{T}},\quad \mathbf{e}_k=(e_{1k},e_{2k},...,e_{Mk})^{\mathrm{T}},\quad k\in\mathbb{K},$$
$$\tilde{\mathbf{e}}=[(\tilde{\mathbf{e}}_1)^{\mathrm{T}},(\tilde{\mathbf{e}}_2)^{\mathrm{T}},...,(\tilde{\mathbf{e}}_M)^{\mathrm{T}}]^{\mathrm{T}},\quad \tilde{\mathbf{e}}_k=(\tilde{e}_{1k},\tilde{e}_{2k},...,\tilde{e}_{Mk})^{\mathrm{T}},\quad k\in\mathbb{K}.$$

By (3.12), it holds that

$$\tilde{\mathbf{G}}\tilde{\mathbf{e}}_l=\begin{cases}0.5\tilde{\mathbf{G}}\mathbf{e}_1+0.5\eta h^{-2}\mathbf{e}_2,\quad l=1,\\ 0.5\tilde{\mathbf{G}}\mathbf{e}_l+0.5\eta h^{-2}(\mathbf{e}_{l-1}+\mathbf{e}_{l+1}),\quad 1<l<M,\\ 0.5\tilde{\mathbf{G}}\mathbf{e}_M+0.5\eta h^{-2}\mathbf{e}_{M-1},\quad l=M.\end{cases}\tag{3.14}$$

Let $\Xi=\frac{2}{M+1}\mathbf{Q}^{\mathrm{T}}\mathbf{e}$ be of the form in (3.13). By orthogonality of $\mathbf{Q}$, $\mathbf{e}=\frac{M+1}{2}\mathbf{Q}\Xi$. We furthermore have

$$e_{kl}=\sum_{(i,j)\in\mathbb{K}\times\mathbb{K}}\xi_{ij}\sin(k\theta_i)\sin(l\theta_j),\quad (k,l)\in\mathbb{K}\times\mathbb{K}.\tag{3.15}$$

Expand $\tilde{\mathbf{e}}$ in the similar form

$$\tilde{e}_{kl}=\sum_{(i,j)\in\mathbb{K}\times\mathbb{K}}\tilde{\xi}_{ij}\sin(k\theta_i)\sin(l\theta_j),\quad (k,l)\in\mathbb{K}\times\mathbb{K}.\tag{3.16}$$

By (3.14), (3.15) and (3.16),

$$\sum_{(i,j)\in\mathbb{K}\times\mathbb{K}}\delta_i\tilde{\xi}_{ij}\sin(k\theta_i)\sin(l\theta_j)=\sum_{(i,j)\in\mathbb{K}\times\mathbb{K}}\eta_{ij}\xi_{ij}\sin(k\theta_i)\sin(l\theta_j),\quad k,l\in\mathbb{K},$$

where

$$\delta_i=2\eta h^{-2}+4\mu\eta h^{-2}\sin^2(\theta_i/2)+\gamma,\quad \eta_{ij}=0.5\delta_i+\eta h^{-2}\cos\theta_j,\quad i,j\in\mathbb{K}.$$

Hence,

$$\tilde{\xi}_{ij} = \delta_i^{-1}\eta_{ij}\xi_{ij} = r_{ij}\xi_{ij}, \quad i, j \in \mathbb{K}. \tag{3.17}$$

Combining (3.17) and (3.16), we obtain

$$\mathbf{Se} = \tilde{\mathbf{e}} = \frac{M+1}{2}\mathbf{QP}^{\mathrm{T}}\hat{\mathbf{S}}\mathbf{P}\,\Xi = \frac{M+1}{2}\mathbf{QP}^{\mathrm{T}}\hat{\mathbf{S}}\mathbf{P}\left(\frac{2}{M+1}\mathbf{Q}^{\mathrm{T}}\mathbf{e}\right) = \tilde{\mathbf{Q}}^{\mathrm{T}}\hat{\mathbf{S}}\tilde{\mathbf{Q}}\mathbf{e}.$$

Taking **e** over all columns of identity matrix, the proof is complete. □

Lemma 4. *(See [10, Theorem 8.1.4]) Let $\tilde{\mathbf{Q}}$ be given by Lemma 3. Then, the correction operator $\mathbf{K}_H^h$ satisfies that*

$$\mathbf{K}_H^h = \tilde{\mathbf{Q}}^{\mathrm{T}}\hat{\mathbf{K}}_H^h\tilde{\mathbf{Q}} := \tilde{\mathbf{Q}}^{\mathrm{T}}\mathrm{diag}\left(K_{11}, ..., K_{\bar{M}1}, K_{12}, ..., K_{\bar{M}2}, ..., K_{1\bar{M}}, ..., K_{\bar{M}\bar{M}}\right)\tilde{\mathbf{Q}},$$

where

$$K_{ij} = I_4 - \tilde{d}_{ij}^{-1}\hat{I}_H^h(i,j)\hat{I}_h^H(i,j)\Lambda_h^{(ij)}, \quad K_{\bar{M}j} = K_{i\bar{M}} = \mathrm{diag}(1,1),\ i,j \in \mathbb{K}^r, \quad K_{\bar{M}\bar{M}} = 1,$$

with

$$\Lambda_h^{(ij)} = \mathrm{diag}(d_{ij}, d_{ij'}, d_{i'j}, d_{i'j'}), \quad (i,j) \in \mathbb{K}^r \times \mathbb{K}^r,$$
$$d_{ij} = 4(a\alpha_i + b\alpha_j) + \gamma,\ (i,j) \in (\mathbb{K} \setminus \{\bar{M}\}) \times (\mathbb{K} \setminus \{\bar{M}\}),$$
$$\tilde{d}_{ij} = 4(a\alpha_i\beta_i + b\alpha_j\beta_j) + \gamma, \quad (i,j) \in \mathbb{K}^r \times \mathbb{K}^r,$$
$$\hat{I}_h^H(i,j) = \frac{1}{2}[\beta_i\beta_j, -\beta_i\alpha_j, -\alpha_i\beta_j, \alpha_i\alpha_j], \quad \hat{I}_H^h(i,j) = 4\left(\hat{I}_h^H(i,j)\right)^{\mathrm{T}}, \quad i,j \in \mathbb{K}^r,$$
$$\alpha_i = \sin^2(\theta_i/2), \quad \beta_i = \cos^2(\theta_i/2),\ i \in \mathbb{K}, \quad b = \eta h^{-2},\ a = \mu b.$$

Lemma 3 and Lemma 4 show that **S** and $\mathbf{K}_H^h$ can be simultaneously block diagonalized by $\tilde{\mathbf{Q}}$. Furthermore, we have following theorem.

Theorem 5.

$$||\mathbf{T}(\nu)||_2 \le \frac{C_0}{\nu - 2}, \quad \forall \nu \ge \nu_0, \tag{3.18}$$

where

$$\nu_0 = \mu + \frac{1}{2\mu} + \frac{3}{2}, \quad C_0 = \max\left\{4a_1\left(\frac{\mu+1}{2\mu+1}\right)^{a_1+2}, a_2\left(\frac{\mu+2}{2\mu+2}\right)^{2a_2+4}, 2(\mu^{-1}+1)\right\}$$
$$a_1 = \ln^{-1}\left[(2\mu+1)/(\mu+1)\right], \quad a_2 = \ln^{-1}\left[(2\mu+2)/(\mu+2)\right]^2.$$

Proof. In this proof, we always assume $\nu \ge \nu_0$. By Lemma 3 and Lemma 4,

$$\mathbf{T}(\nu) = \tilde{\mathbf{Q}}^{\mathrm{T}}\hat{\mathbf{S}}^{\nu}\hat{\mathbf{K}}_H^h\hat{\mathbf{S}}^{\nu}\tilde{\mathbf{Q}}.$$

Since $\tilde{\mathbf{Q}}$ is also an unitary matrix, we by unitary invariance of 2-norm have

$$||\mathbf{T}(\nu)||_2 = ||\hat{\mathbf{S}}^{\nu}\hat{\mathbf{K}}_H^h\hat{\mathbf{S}}^{\nu}||_2 = \max_{i,j\in\mathbb{K}^b}||T_{ij}||_2, \qquad T_{ij} = \hat{S}_{ij}^{\nu}K_{ij}\hat{S}_{ij}^{\nu}, \quad i,j \in \mathbb{K}^b. \tag{3.19}$$

Hence, we only need to prove $||T_{ij}||_2 \le C_0(\nu-2)^{-1}$, $\forall i,j \in \mathbb{K}^b$. By straightforward calculation,

$$T_{\bar{M}\bar{M}} = r_{\bar{M}\bar{M}}^{2\nu},\ T_{\bar{M}j} = \mathrm{diag}(r_{\bar{M}j}^{2\nu}, r_{\bar{M}j'}^{2\nu}),\ T_{i\bar{M}} = \mathrm{diag}(r_{i\bar{M}}^{2\nu}, r_{i'\bar{M}}^{2\nu}), \quad i,j \in \mathbb{K}^r.$$

Hence,

$$\max\left\{\max_{i\in\mathbb{K}^b}||T_{i\bar{M}}||_2, \max_{j\in\mathbb{K}^r}||T_{\bar{M}j}||_2\right\} \le \left(\frac{\mu+2}{2\mu+2}\right)^{2\nu} \le \frac{C_0}{\nu-2}. \tag{3.20}$$

Now, we consider estimation of $||T_{ij}||_2$ $(i,j \in \mathbb{K}^r)$. Let $||\cdot||_F$ denote Frobenius norm. Then, it suffices to show $||T_{ij}||_F \le C_0(\nu-2)^{-1}$ $(i,j \in \mathbb{K}^r)$, since $||T_{ij}||_2 \le ||T_{ij}||_F$. We also denote by $\tilde{T}_{xy}$, T_{ij} with $x = \alpha_i$, $y = \alpha_j$ when $i,j \in \mathbb{K}^r$, i.e. $\tilde{T}_{\alpha_i\alpha_j} = T_{ij}$ for any $i,j \in \mathbb{K}^r$. Again, by straightforward calculation,

$$\tilde{T}_{xy} = \begin{bmatrix} E_0(x,y) & E_1(x,y) & E_2(x,y) & E_3(x,y) \\ E_1(x,\tilde{y}) & E_0(x,\tilde{y}) & E_3(x,\tilde{y}) & E_2(x,\tilde{y}) \\ E_2(\tilde{x},y) & E_3(\tilde{x},y) & E_0(\tilde{x},y) & E_1(\tilde{x},y) \\ E_3(\tilde{x},\tilde{y}) & E_2(\tilde{x},\tilde{y}) & E_1(\tilde{x},\tilde{y}) & E_0(\tilde{x},\tilde{y}) \end{bmatrix}$$

where

$$E_0(x,y) = r^{2\nu}(x,y)\left[1-\tilde{x}^2\tilde{y}^2 g(x,y)\right],\ E_1(x,y) = \left[r(x,y)r(x,\tilde{y})\right]^{\nu}\tilde{x}^2 y\tilde{y}g(x,\tilde{y}),$$
$$E_2(x,y) = \left[r(x,y)r(\tilde{x},y)\right]^{\nu}x\tilde{x}\tilde{y}^2 g(\tilde{x},y),\ E_3(x,y) = [r(x,y)r(\tilde{x},\tilde{y})]^{\nu}x\tilde{x}y\tilde{y}g(\tilde{x},\tilde{y}),$$
$$r(x,y) = \frac{2\mu x + 2^{-1}\gamma^* + 2\tilde{y}}{4\mu x + \gamma^* + 2},\quad g(x,y) = \frac{4(ax+by)+\gamma}{4\left(ax\tilde{x}+by\tilde{y}\right)+\gamma},\quad \tilde{x} = 1-x, \tilde{y} = 1-y.$$

Note that

$$\max_{i,j\in\mathbb{K}^r}||T_{ij}||_F \le \sup_{x,y\in(0,1)}||\tilde{T}_{x,y}||_F \le 2\sqrt{\sum_{i=0}^{3}\sup_{x,y\in(0,1)}|E_i(x,y)|^2}.$$

Hence, it suffices to show

$$\sup_{x,y\in(0,1)}|E_i(x,y)| \le \frac{C_0}{4(\nu-2)},\quad \forall i\in\{0,1,2,3\}.$$

We recall that both γ and γ^* have nonnegative real parts, which will be frequently used in this proof.

$$\begin{aligned}\sup_{x,y\in(0,1)}|E_0(x,y)| &= \sup_{x,y\in(0,1)}|r(x,y)|^{2\nu}\left|\frac{4ax\tilde{x}(1-\tilde{x}\tilde{y}^2)+4by\tilde{y}(1-\tilde{x}^2\tilde{y})+\gamma(1-\tilde{x}^2\tilde{y}^2)}{4(ax\tilde{x}+by\tilde{y})+\gamma}\right| \\ &\le \sup_{x,y\in(0,1)}(1-\tilde{x}^2\tilde{y}^2)|r(x,y)|^{2\nu} \\ &\le \sup_{x\in(0,1)}\left\{\sup_{y\in[0.5,1)}|r(x,y)|^{2\nu},\ \sup_{y\in(0,0.5]}|r(x,y)|^{2\nu}\left[1-\tilde{x}^2\tilde{y}^2\right]\right\} \\ &\le \max\left\{\frac{1}{2^{2\nu}},\ \sup_{x\in(0,1)}\sup_{y\in(0,0.5]}|r(x,y)|^{2\nu}\left[1-\tilde{x}^2\tilde{y}^2\right]\right\}.\end{aligned} \tag{3.21}$$

Since $1-\tilde{x}^2\tilde{y}^2 = (1-\tilde{x}\tilde{y})(1+\tilde{x}\tilde{y}) \le 2(x+y)$,

$$\begin{aligned}\sup_{x\in(0,1)}\sup_{y\in(0,0.5]}|r(x,y)|^{2\nu}\left[1-\tilde{x}^2\tilde{y}^2\right] &\le 2\sup_{x\in(0,1)}\sup_{y\in(0,0.5]}|r(x,y)|^{2\nu}(x+y) \\ &\le 2\left[\sup_{x\in(0,1)}\left(\frac{\mu x+1}{2\mu x+1}\right)^{2\nu}x + \sup_{y\in(0,0.5]}(1-y)^{2\nu}y\right].\end{aligned} \tag{3.22}$$

By solving $\left[(1-y)^{2\nu}y\right]' = 0$, it is easy to find that

$$\max_{y\in[0,0.5]}(1-y)^{2\nu}y = \left(\frac{2\nu}{2\nu+1}\right)^{2\nu}\left(\frac{1}{2\nu+1}\right) \le \frac{1}{2\nu+1}. \tag{3.23}$$

Let $f(x) = x(\mu x+1)^{2\nu}(2\mu x+1)^{-2\nu}$. Then

$$f'(x) = (2\mu x+1)^{-2\nu-1}(\mu x+1)^{2\nu-1}[2\mu^2x^2+\mu(3-2\nu)x+1],\quad x\in[0,1].$$

Let $g(x) = 2\mu^2x^2+\mu(3-2\nu)x+1$. Then, it is easy to check that $g(x)$ has a unique zero point $x_* = (4\mu)^{-1}[2\nu-3-\sqrt{4\nu^2-12\nu+1}]$ over the domain $(0,1]$. And it holds that

$$g(x)\ge 0,\ \forall x\in(0,x_*),\qquad g(x)\le 0,\ \forall x\in[x_*,1].$$

In addition, $f(x)\le x, \forall x\in[0,1]$. Therefore,

$$\max_{x\in(0,1]}f(x) = f(x_*) \le x_*. \tag{3.24}$$

Note also that

$$\left[4[(2\nu-3)^2-8]^{-0.5}+[(2\nu-3)^2-8]^{0.5}\right]^2=16[(2\nu-3)^2-8]^{-1}+(2\nu-3)^2\geq(2\nu-3)^2$$

$$\Longleftrightarrow$$

$$4[4\nu^2-12\nu+1]^{-0.5}\geq 2\nu-3-\sqrt{4\nu^2-12\nu+1}.$$

Hence,

$$x_*\leq\mu^{-1}[4\nu^2-12\nu+1]^{-0.5}\leq\mu^{-1}[4\nu^2-12\nu+1-4\nu+15]^{-0.5}=(2\mu)^{-1}(\nu-2)^{-1}. \tag{3.25}$$

By (3.24) and (3.25),

$$\max_{x\in(0,1)} f(x)\leq(2\mu)^{-1}(\nu-2)^{-1}. \tag{3.26}$$

By (3.21), (3.22), (3.23) and (3.26),

$$\sup_{x,y\in(0,1)}|E_0(x,y)|\leq\max\left\{\frac{1}{2^{2\nu}},\frac{2^{-1}(\mu^{-1}+1)}{\nu-2}\right\}\leq\frac{C_0}{4(\nu-2)}. \tag{3.27}$$

$$\begin{aligned}\sup_{x,y\in(0,1)}|E_1(x,y)|&=\sup_{x,y\in(0,1)}|r(x,y)r(x,\tilde{y})|^{\nu}\tilde{x}\tilde{y}\left|\frac{4(ax\tilde{x}y+b\tilde{x}y\tilde{y}\beta_i)+\tilde{x}y\gamma}{4(a\tilde{x}+by\tilde{y})+\gamma}\right|\\&\leq\sup_{x,y\in(0,1)}|r(x,y)r(x,\tilde{y})|^{\nu}\leq\max_{x\in(0,1)}\left\{\max_{y\in(0,0.5]}|r(x,1-y)|^{\nu},\max_{y\in[0.5,1)}|r(x,y)|^{\nu}\right\}\\&\leq\frac{1}{2^{\nu}}\leq\left(\frac{\mu+1}{2\mu+1}\right)^{\nu}\leq\frac{C_0}{4(\nu-2)}.\end{aligned} \tag{3.28}$$

Similarly to (3.28), one can prove that

$$\sup_{x,y\in(0,1)}|E_3(x,y)|\leq C_0 4^{-1}(\nu-2)^{-1}. \tag{3.29}$$

$$\begin{aligned}\sup_{x,y\in(0,1)}|E_2(x,y)|&=\sup_{x,y\in(0,1)}|r(x,y)r(\tilde{x},y)|^{\nu}\tilde{x}\tilde{y}\left|\frac{4(ax\tilde{x}\tilde{y}+bxy\tilde{y})+x\tilde{y}\gamma}{4(a\tilde{x}+by\tilde{y})+\gamma}\right|\\&\leq\sup_{x,y\in(0,1)}|r(x,y)|^{\nu}|r(\tilde{x},y)|^{\nu}\leq\sup_{x\in(0,1)}\left\{\sup_{y\in(0,0.5]},\sup_{y\in[0.5,1)}\right\}\left\{|r(x,y)r(\tilde{x},y)|^{\nu}\right\}\\&\leq\max\left\{\sup_{x\in(0,1)}|r(x,0)r(\tilde{x},0)|^{\nu},\frac{1}{2^{2\nu}}\right\}\\&\leq\max\left\{\sup_{x\in(0,1)}\left[\frac{(\mu x+1)(\mu\tilde{x}+1)}{(2\mu x+1)(2\mu\tilde{x}+1)}\right]^{\nu},\frac{1}{2^{2\nu}}\right\}.\end{aligned} \tag{3.30}$$

Let $\Phi(x)=\left[(\mu x+1)(\mu\tilde{x}+1)\right]^{\nu}\left[(2\mu x+1)(2\mu\tilde{x}+1)\right]^{-\nu}$. Since $\Phi(x)=\Phi(1-x)$, $\forall x\in[0,1]$. Hence, $\max\limits_{x\in[0,1]}\Phi(x)=\max\limits_{x\in[0,0.5]}\Phi(x)$. Since

$$\max_{x\in[0,0.5]}\Phi'(x)=\max_{x\in[0,0.5]}\frac{\mu^2(2\mu+3)(2x-1)}{[4\mu^2x(1-x)+2\mu+1]^2}\leq 0.$$

Therefore,

$$\max_{x\in[0,1]}\Phi(x)=\Phi(0)=\left(\frac{\mu+1}{2\mu+1}\right)^{\nu}. \tag{3.31}$$

By (3.30) and (3.31),

$$\sup_{x,y\in(0,1)}|E_2(x,y)|\leq\frac{C_0}{4(\nu-2)}, \tag{3.32}$$

which completes the proof. □

3.2. The convergence analysis of MGM

In this subsection, we show the convergence of MGM(ν,q). Denote by $\mathbf{M}(\nu,q)$, the iteration matrix of MGM(ν,q). We denote by $\mathbf{M}_i(\nu,q)$, $\mathbf{T}_i(\nu)$, $\mathbf{S}_i$, $\mathbf{M}(\nu,q)$, $\mathbf{T}(\nu)$ and $\mathbf{S}$ with $M=M_i$ respectively. Also, we denote by $\mathbf{I}_{i-1}^{i}$, $\mathbf{I}_{i}^{i-1}$, the interpolation operator and restriction operator between level $i-1$ and i. Then, $\mathbf{M}_i(\nu,q)$ with positive integer $i\geq 3$ can be written as [10]

$$\mathbf{M}_i(\nu,q)=\mathbf{T}_i(\nu)+\mathbf{S}_i^{\nu}\mathbf{I}_{i-1}^{i}\mathbf{M}_{i-1}^{q}(\nu,q)\mathbf{W}_i^{i-1}\mathbf{S}_i^{\nu},\quad i\geq 4, \tag{3.33}$$

where $\mathbf{W}_i^{i-1}=\left(\mathbf{\Lambda}^{(i-1)}\right)^{-1}\mathbf{I}_i^{i-1}\mathbf{\Lambda}^{(i)}$, $\mathbf{M}_3(\nu,q)=\mathbf{T}_3(\nu)$. Denote by $\tilde{\mathbf{Q}}_h$, $\tilde{\mathbf{Q}}_H$, $\tilde{\mathbf{Q}}$ on fine grid and coarse grid, respectively. Neglect superscript and subscript of $\mathbf{W}_i^{i-1}$, and rewrite it as $\mathbf{W}_h^H$.

Lemma 6. *(See [10, Theorem 8.1.2-3])*

$$\mathbf{I}_H^h=\tilde{\mathbf{Q}}_h^{\mathrm{T}}\hat{P}\tilde{\mathbf{Q}}_H:=\tilde{\mathbf{Q}}_h^{\mathrm{T}}\mathrm{diag}(\hat{p}_{11},\ldots,\hat{p}_{\bar{M}1},\hat{p}_{12},\ldots,\hat{p}_{\bar{M}2},\ldots\ldots,\hat{p}_{1\bar{M}},\ldots,\hat{p}_{\bar{M}\bar{M}})\tilde{\mathbf{Q}}_H,$$
$$\mathbf{W}_h^H=\tilde{\mathbf{Q}}_H^{\mathrm{T}}\hat{\mathbf{W}}_h^H\tilde{\mathbf{Q}}_h:=\tilde{\mathbf{Q}}_H^{\mathrm{T}}\mathrm{diag}(\hat{V}_{11},\ldots,\hat{V}_{\bar{M}1},\hat{V}_{12},\ldots,\hat{V}_{\bar{M}2},\ldots\ldots,\hat{V}_{1\bar{M}},\ldots,\hat{V}_{\bar{M}\bar{M}})\tilde{\mathbf{Q}}_h,$$

with

$$\hat{p}_{ij}=\begin{cases}\hat{I}_H^h(i,j), & i,j\in\mathbb{K}^r,\\ [0,0]^{\mathrm{T}}, & i=\bar{M},\ j\in\mathbb{K}^r,\\ [0,0]^{\mathrm{T}}, & i\in\mathbb{K}^r,\ j=\bar{M},\\ 0, & i=\bar{M},\ j=\bar{M},\end{cases},\qquad \hat{V}_{ij}=\begin{cases}\tilde{d}_{ij}^{-1}\hat{I}_h^H(i,j)\Lambda_h^{(ij)}, & i,j\in\mathbb{K}^r,\\ [0,0], & i=\bar{M},\ j\in\mathbb{K}^r,\\ [0,0], & i\in\mathbb{K}^r,\ j=\bar{M},\\ 0, & i=\bar{M},\ j=\bar{M},\end{cases}$$

where $\hat{I}_H^h(i,j)$, $\hat{I}_h^H(i,j)$, $\tilde{d}_{ij}$, $\Lambda_h^{(ij)}$ are all given by Lemma 4.

Denote by $\mathbf{S}_h$, $\mathbf{S}$ on fine grid. By Lemma 3 and Lemma 6, we immediately obtain

$$\begin{aligned}&||\mathbf{S}_h^{\nu}\mathbf{I}_H^h||_2\leq||\mathbf{S}_h^{\nu}||_2||\mathbf{I}_H^h||_2\leq||\mathbf{I}_H^h||_2=\max_{i,j\in\mathbb{K}^b}||\hat{p}_{ij}||_2=\max_{i,j\in\mathbb{K}^r}||\hat{I}_H^h(i,j)||_2=\\&\max_{i,j\in\mathbb{K}^r}\sqrt{4(\alpha_i^2+\beta_i^2)(\alpha_j^2+\beta_j^2)}=\max_{i,j\in\mathbb{K}^r}\sqrt{4^{-1}[3+\cos(2\theta_i)][3+\cos(2\theta_j)]}\leq 2.\end{aligned} \tag{3.34}$$

By Lemma 3, proof of Theorem 5 and Lemma 6,

$$\begin{aligned}&||\mathbf{W}_h^H\mathbf{S}_h^{\nu}||_2=\max_{i,j\in\mathbb{K}^b}||\hat{V}_{ij}\hat{S}_{ij}^{\nu}||_2\leq\max_{i,j\in\mathbb{K}^b}||\hat{V}_{ij}||_2\leq\max_{i,j\in\mathbb{K}^r}||\hat{V}_{ij}||_2\leq\\&\sup_{x,y\in(0,1)}\sqrt{4^{-1}[|\tilde{x}\tilde{y}g(x,y)|^2+|\tilde{x}yg(x,\tilde{y})|^2+|x\tilde{y}g(\tilde{x},y)|^2+|xyg(\tilde{x},\tilde{y})|^2]}\leq 1,\end{aligned} \tag{3.35}$$

where x, y, $\tilde{x}$, $\tilde{y}$, $g(x,y)$ are notations used in the proof of Theorem 5. Actually, Theorem 5, (3.33), (3.34) and (3.35) provide following recursive inequalities

$$||\mathbf{M}_i(\nu,q)||_2\leq\begin{cases}\frac{C_0}{2-\nu}+2||\mathbf{M}_{i-1}(\nu,q)||_2^q, & \forall i\geq 4,\\ \frac{C_0}{2-\nu}, & i=3,\end{cases}\qquad\forall\nu\geq\nu_0, \tag{3.36}$$

where both C_0 and ν_0 are given by Theorem 5. With (3.36), we have following theorem.

Theorem 7. *(See [10, Lemma 7.1.6]) Let $q\geq 2$. Take*

$$\nu^*=\max\left\{\frac{C_0q(2q)^{(q-1)^{-1}}}{q-1}+2,\nu_0\right\},\quad C^*=\frac{C_0q}{q-1}.$$

Then,

$$\sup_{i\geq 3}||\mathbf{M}_i(\nu,q)||_2\leq\frac{C^*}{\nu-2},\quad\forall\nu\geq\nu^*,$$

where ν_0 and C_0 are given by Theorem 5.

By Theorem 7, we see that the convergence rate of MGM(ν,q) $(q\geq 2)$ can be controlled by ν. Moreover, MGM(ν,q) $(q\geq 2)$ has a small convergence rate independent of both matrix size and γ whenever ν is large enough. This implies that Algorithm 2 converges within $\mathcal{O}(1)$ iterations. Hence, it follows from Theorem 2 that Algorithm 2 requires $\mathcal{O}(M^2)$ operations for $q\in\{2,3\}$ and properly large ν. Finally, since there are $\lfloor N/2\rfloor+1$ linear systems in (2.11) needing to be solved, we conclude that (2.11) requires $\mathcal{O}(NM^2)$ storage and $\mathcal{O}(NM^2)$ operations by using Algorithm 2.

4. Numerical results

In this section, we use two examples to test AIMGM. All numerical experiments are performed via Console Application Visual C++ in Visual Studio 2012 on a PC with the configuration: Intel(R) Core(TM) i5-4590 CPU 3.30 GHz and 8 GB RAM.

For the choice of approximation of Caputo's derivative ${}_0^C\mathcal{D}_t^\alpha$ in (2.1), we refer to L_1 formula in [24] such that

$$g_k^{(\alpha)}=\begin{cases}[\tau^\alpha\Gamma(2-\alpha)]^{-1}, & k=0,\\ [(k+1)^{1-\alpha}-2k^{1-\alpha}+(k-1)^{1-\alpha}][\tau^\alpha\Gamma(2-\alpha)]^{-1}, & 1\le k\le N-1,\end{cases} \tag{4.1}$$

$$g^{(k,\alpha)}=[(k-1)^{1-\alpha}-k^{1-\alpha}][\tau^\alpha\Gamma(2-\alpha)]^{-1},\quad 1\le k\le N. \tag{4.2}$$

Define spatial grids $x_i=x_L+ih_1$ $(0\le i\le M+1)$, $y_j=y_L+jh_2$ $(0\le j\le M+1)$, $h_1=(x_R-x_L)/(M+1)$, $h_2=(y_R-y_L)/(M+1)$. For any function $v(x,y)$ defined on Ω, denote by v_{ij}, $v(x_i,y_j)$ for $i,j\in\{0,1,...,M+1\}$. Then, the central difference approximation to $-\nabla\cdot(p(x,y)\nabla v)$ is defined as follows

$$\begin{aligned}&-\nabla\cdot(p(x,y)\nabla v)|_{(x_i,y_j)}\approx\\ &\frac{p_{i+\frac12,j}(v_{ij}-v_{i+1,j})+p_{i-\frac12,j}(v_{i,j}-v_{i-1,j})}{h_1^2}+\frac{p_{i,j+\frac12}(v_{ij}-v_{i,j+1})+p_{i,j-\frac12}(v_{i,j}-v_{i,j-1})}{h_2^2}\end{aligned} \tag{4.3}$$

where

$$p_{i+s_1,j+s_2}=p(x_L+(i+s_1)*h_1,y_L+(j+s_2)*h_2),\ 1\le i,j\le M,\quad s_1,s_2\in\{-1/2,\ 0,\ 1/2\}.$$

It is easy to check that (4.2), (4.1) and (4.3) satisfy assumption (2.7) and (2.8). Hence, we use (4.2), (4.1) and (4.3) in all experiments in this section. Moreover, we choose zebra-line Gauss–Seidel (ZLGS) smoother [10] in numerical experiments of this section since it has better numerical performance, although we prove convergence of MGM with WBJ smoother in last section. In details, $\mathbf{\Lambda}$ and $\mathbf{y}$ in (3.1) resulting from (4.3) have following form

$$\mathbf{\Lambda}=\begin{bmatrix}\mathbf{G}_1 & \mathbf{H}_1 & & & \\ \mathbf{H}_1 & \mathbf{G}_2 & \mathbf{H}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{H}_{M-2} & \mathbf{G}_{M-1} & \mathbf{H}_{M-1}\\ & & & \mathbf{H}_{M-1} & \mathbf{G}_M\end{bmatrix},\ \mathbf{y}=\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_M\end{bmatrix},\ \mathbf{G}_i,\mathbf{H}_i\in\mathbb{C}^{M\times M},\ \mathbf{y}_i\in\mathbb{C}^{M\times 1},$$

where $\mathbf{G}_i$'s are all nonsingular tridiagonal blocks. Let $\mathbf{x}^{\text{old}}$ be an initial guess of $\mathbf{x}$ such that $\mathbf{x}^{\text{old}}=\left[(\mathbf{x}_1^{\text{old}})^{\mathrm{T}},(\mathbf{x}_2^{\text{old}})^{\mathrm{T}},\ldots,(\mathbf{x}_M^{\text{old}})^{\mathrm{T}}\right]^{\mathrm{T}}$ with $\mathbf{x}_i^{\text{old}}\in\mathbb{C}^{M\times 1}$. Then, one iteration of ZLGS relaxation is defined as

$$\begin{cases}\mathbf{x}_i^{\text{new}}=\mathbf{G}_i^{-1}\left(\mathbf{y}_i-\mathbf{H}_{i-1}\mathbf{x}_{i-1}^{\text{old}}-\mathbf{H}_i\mathbf{x}_{i+1}^{\text{old}}\right), & i \text{ even},\\ \mathbf{x}_i^{\text{new}}=\mathbf{G}_i^{-1}\left(\mathbf{y}_i-\mathbf{H}_{i-1}\mathbf{x}_{i-1}^{\text{new}}-\mathbf{H}_i\mathbf{x}_{i+1}^{\text{new}}\right), & i \text{ odd}.\end{cases}$$

Moreover, we still use $\mathbf{I}_h^H$ and $\mathbf{I}_H^h$ defined by (3.10) in this section.

According to the suggestion in [15], we set $\epsilon=0.5\times10^{-8}$ to test the approximation method. Although Theorem 7 implies linear convergence only under the condition of large ν and $q\ge 2$, MGM(1, 1) shows a linear convergence in numerical results. Hence, we choose $\nu=q=1$ in this section for minimizing number of operations each MGM iteration required.

Define the relative error

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

where we denote by $\mathbf{u}$ and $\tilde{\mathbf{u}}$ denote exact solution, approximate solution derived from some numerical method to the FSDE (1.1)–(1.3) on the gird $\Omega_h\times\{t_n|1\le n\le N\}$ respectively. Note that AIMGM only requires to solve first $\lfloor N/2\rfloor+1$ linear systems in (2.11) by using Algorithm 2. Hence, we denote by "iter", the average iteration number for Algorithm 2 solving the $\lfloor N/2\rfloor+1$ linear systems. We also denote by CPU, the running time of some algorithm by unit second.

Example 1. ([24]) In this example, we consider the problem (1.1)–(1.3) with

$$u(x,y,t)=\sin(x)\sin(y)t^2,\quad f(x,y,t)=\sin(x)\sin(y)\left[\frac{2t^{2-\alpha}}{\Gamma(3-\alpha)}+2t^2\right],\quad p(x,y)\equiv 1,$$

on the domain $\Omega\times[0,0.5]$, where $\Omega=(0,\pi)\times(0,\pi)$. We solve the problem in Example 1 with AIMGM and BD-ADI scheme [24] respectively. And the results are listed in Table 1 and Table 2.

Table 1
CPU time and temporal order of AIMGM and BD-ADI scheme when $M + 1 = 512$.

α	N	AIMGM			BD-ADI	
		iter	CPU	$E_{N,M}$	CPU	$E_{N,M}$
1/6	8	7.0	1.263 s	1.1331e−3	0.14 s	4.0907e−2
	16	7.0	2.262 s	3.4622e−4	0.265 s	1.9345e−2
	32	7.0	4.461 s	1.0557e−4	0.639 s	8.8761e−3
	64	7.0	8.658 s	3.2656e−5	1.84 s	4.0137e−3
1/2	32	8.0	4.976 s	1.1615e−3	0.624 s	6.5003e−4
	64	8.0	9.765 s	4.1761e−4	1.809 s	2.2950e−4
	128	8.0	19.578 s	1.5016e−4	14.57 s	7.9873e−5
	256	8.0	39.686 s	5.4395e−5	76.248 s	2.7126e−5
0.99	100	8.9	16.854 s	5.7608e−3	6.938 s	5.9007e−3
	200	8.9	34.379 s	2.8715e−3	42.692 s	2.9395e−3
	400	8.9	68.53 s	1.4056e−3	220.614 s	1.4623e−3
	800	8.7	134.316 s	7.0614e−4	921.213 s	7.2712e−4

Table 2
CPU time and error of AIMGM and BD-ADI scheme when $N = 2^{13}$, $\alpha = 0.01$.

$M + 1$	AIMGM			BD-ADI	
	iter	CPU	$E_{N,M}$	CPU	$E_{N,M}$
4	1.0	0.015 s	3.4545e−2	0.534 s	3.4474e−2
8	5.0	0.171 s	8.5479e−3	2.685 s	8.4773e−3
16	6.0	0.826 s	2.1321e−3	28.796 s	2.0590e−3
32	7.0	3.946 s	5.3271e−4	250.910 s	4.5946e−4
64	7.0	16.309 s	1.3245e−4	1539.707 s	5.9896e−5
128	7.0	66.87 s	3.3016e−5	8394.270 s	3.9975e−5

Table 1 and Table 2 show that CPU cost of BD-ADI increases much faster than that of AIMGM when N and M increase, which well reflects that AIMGM and BD-ADI require $\mathcal{O}(M^2 N \log N)$, $\mathcal{O}(M^2 N^2)$ operations, respectively. Especially, in Table 2, AIMGM runs even 124 times faster than BD-ADI when $M = 127$. Moreover, the average iteration number of MGM changes slightly in both Table 1 and Table 2, which implies a good convergence result of MGM. In addition, from Table 1 and Table 2, we note that error reduction rate of AIMGM is higher than that of BD-ADI scheme in Table 1 and accuracy of the two schemes are comparable with each other in Table 2, which matches convergence analysis in [24] well. Finally, it is noticeable that when $\alpha = 1/6$, AIMGM is much more accurate than BD-ADI in Table 1 although it runs a little slower than BD-ADI at that case.

Example 2. In this example, we consider the problem (1.1)–(1.3) with

$$f(x, y, t) = \frac{\Gamma(4) x y t^{3-\alpha}}{\Gamma(4-\alpha)} - \exp(xy + x^2 + y^2) t^3,$$

$$p(x, y) = \exp(xy),\ u(x, y, t) = xyt^3,$$

on the domain $\Omega \times [0, 0.5]$, where $\Omega = (-1, 1) \times (-1, 1)$. As mentioned in Section 1, the block forward substitution method may work with some iterative method. Here, we combine the block forward substitution method with MGM, which is called BFSMGM. In details, the algorithm of BFSMGM for solving the linear system (2.2) is given by

Algorithm 3 BFSMGM.

```
solve $\left(\mathbf{B} + g_0^{(\alpha)} \mathbf{I}_{M^2}\right) \mathbf{u}^1 = \mathbf{b}^1$ with Algorithm 2;
for $k = 2 : N$
    $\mathbf{f} = \mathbf{b}^k$;
    for $i = 1 : k-1$
        $\mathbf{f} = \mathbf{f} - g_{k-i}^{(\alpha)} \mathbf{u}^i$;
    end
    solve $\left(\mathbf{B} + g_0^{(\alpha)} \mathbf{I}_{M^2}\right) \mathbf{u}^k = \mathbf{f}$ with Algorithm 2;
end
```

Since BD-ADI scheme is only available in the case that $p(x, y)$ is constant, we solve the problem in Example 2 with only AIMGM and BFSMGM respectively. Note that there are N linear systems need to be solved by Algorithm 2 in Algorithm 3. Thus, we denote 'iter' as the average iteration number of BFSMGM. The results are listed in Table 3, Table 4.

Table 3
CPU time and error of AIMGM and BFSMGM when $M+1=512$.

α	N	AIMGM			BFSMGM		
		iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$
1/6	32	7.0	4.727 s	2.6110e−5	7.0	4.851 s	2.6110e−5
	64	7.0	9.344 s	7.8700e−6	7.0	9.952 s	7.8784e−6
	128	7.0	18.798 s	2.3455e−6	7.0	20.592 s	2.3381e−6
	256	7.0	38.110 s	6.7767e−7	7.0	44.694 s	6.7846e−7
	512	7.0	76.191 s	1.8332e−7	7.0	102.349 s	1.8663e−7
1/2	50	7.0	7.316 s	1.5970e−4	7.0	7.706 s	1.5970e−4
	100	7.0	14.405 s	5.7553e−5	7.0	15.646 s	5.7548e−5
	200	7.0	29.491 s	2.0605e−5	7.0	33.617 s	2.0605e−5
	400	7.0	59.099 s	7.3412e−6	7.0	75.686 s	7.3414e−6
	800	7.0	120.358 s	2.6017e−6	7.0	185.494 s	2.6034e−6
0.99	50	7.0	7.285 s	3.2581e−3	7.0	7.722 s	3.2582e−3
	100	7.0	14.430 s	1.6240e−3	7.0	15.990 s	1.6241e−3
	200	7.0	29.564 s	8.0791e−4	7.0	33.994 s	8.0827e−4
	400	7.0	59.049 s	4.0154e−4	7.0	75.666 s	4.0256e−4
	800	7.0	121.269 s	1.9948e−4	7.0	184.845 s	2.0169e−4

Table 4
CPU time and error of AIMGM and BFSMGM when $N=15000, \alpha=0.01$.

$M+1$	AIMGM			BFSMGM		
	iter	CPU	$E_{N,M}$	iter	CPU	$E_{N,M}$
4	1.0	0.031 s	5.7306e−4	1.0	0.920 s	5.7306e−4
8	5.0	0.312 s	1.1432e−4	5.0	3.478 s	1.1432e−4
16	6.0	1.560 s	2.3803e−5	6.0	21.918 s	2.3803e−5
32	6.0	6.661 s	5.3799e−6	6.0	76.448 s	5.3797e−6
64	7.0	30.992 s	1.2683e−6	7.0	328.002 s	1.2683e−6
128	7.0	127.977 s	3.0824e−7	7.0	1370.954 s	3.0795e−7

Table 3 and Table 4 well reflect that AIMGM and BFSMGM require $\mathcal{O}(M^2 N \log N)$, $\mathcal{O}(M^2 N^2)$ operations, respectively. Especially in Table 4, AIMGM runs much faster than BFSMGM. Iteration numbers of MGM in both Table 3 and Table 4 varying slightly also imply linear convergence of MGM. Accuracy of AIMGM and BFSMGM are comparable with each other in both Table 3 and Table 4.

5. Concluding remarks

In this paper, we have proposed and studied AIMGM consisting of the approximation method (2.9)–(2.12) and multigrid method as a fast solver for solving the FSDE (1.1)–(1.3). Theoretically, sufficient conditions (2.7) and (2.8) are exploited to guarantee high accuracy of the approximation method. The convergence of MGM in the case of constant coefficients is shown for this special complex shifted Laplacian linear system. Both complexity analysis and numerical results show that AIMGM requires $\mathcal{O}(M^2 N \log N)$ operations which is remarkably cheaper than that of both BD-ADI scheme and BFSMGM as shown in numerical experiments in Section 4. In addition, complexity analysis in Section 3 shows that AIMGM requires $\mathcal{O}(NM^2)$ storage, which implies $\mathcal{O}(NM^2)$ memory requirement of AIMGM. It is interesting to consider AIMGM for higher dimensional problem of fractional sub-diffusion equations for future research work.

References

[1] O. Agrawal, Solution for a fractional diffusion–wave equation defined in a bounded domain, Nonlinear Dyn. 29 (2002) 145–155.
[2] A. Alikhanov, A new difference scheme for the time fractional diffusion equation, J. Comput. Phys. 280 (2015) 424–438.
[3] D. Bini, Parallel solution of certain Toeplitz linear systems, SIAM J. Comput. 13 (1984) 268–276.
[4] J. Bouchaud, A. Georges, Anomalous diffusion in disordered media: statistical mechanisms, models and physical applications, Phys. Rep. 195 (1990) 127–293.
[5] H. Brunner, L. Ling, M. Yamamoto, Numerical simulations of 2D fractional subdiffusion problems, J. Comput. Phys. 229 (2010) 6613–6622.
[6] J. Chen, F. Liu, Q. Liu, X. Chen, V. Anh, I. Turner, Numerical simulation for the three-dimension fractional sub-diffusion equation, Appl. Math. Model. 38 (2014) 3695–3705.
[7] S. Chen, F. Liu, X. Jiang, I. Turner, K. Burrage, Fast finite difference approximation for identifying parameters in a two-dimensional space-fractional nonlocal model with variable diffusivity coefficients, SIAM J. Numer. Anal. 54 (2016) 606–624.
[8] G. Golub, C. Loan, Matrix Computations, 3rd edition, Johns Hopkins University Press, Baltimore, 1996.
[9] M. Gupta, R. Manohar, J. Stephenson, High order difference schemes for two-dimensional elliptic equations, Numer. Methods Partial Differ. Equ. 1 (1985) 71–80.
[10] W. Hackbusch, Multigrid Methods and Applications, Springer Science and Business Media, 2013.

[11] X. He, T. Lin, Y. Lin, Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions, Int. J. Numer. Anal. Model. 8 (2011) 284–301.
[12] R. Hilfer (Ed.), Applications of Fractional Calculus in Physics, World Scientific, Singapore, 2000.
[13] C. Ji, Z. Sun, The high-order compact numerical algorithms for the two-dimensional fractional sub-diffusion equation, Appl. Math. Comput. 269 (2015) 775–791.
[14] F. Lin, W. Ching, M. Ng, Fast inversion of triangular Toeplitz matrices, Theor. Comput. Sci. 315 (2004) 511–523.
[15] X. Lu, H. Pang, H. Sun, Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations, Numer. Linear Algebra Appl. 22 (2015) 866–882.
[16] R. Lynch, J. Rice, High accuracy finite difference approximation to solutions of elliptic partial differential equations, Proc. Natl. Acad. Sci. 75 (1978) 2541–2544.
[17] R. Metzler, J. Klafter, The random walk's guide to anomalous diffusion: a fractional dynamics approach, Phys. Rep. 339 (2000) 1–77.
[18] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.
[19] T. Solomon, E. Weeks, H. Swinney, Observation of anomalous diffusion and Lévy flights in a 2-dimensional rotating flow, Phys. Rev. 71 (1993) 3975–3979.
[20] R. Stenberg, On some techniques for approximating boundary conditions in the finite element method, J. Comput. Appl. Math. 63 (1995) 139–148.
[21] F. Zeng, C. Li, F. Liu, I. Turner, Numerical algorithms for time-fractional subdiffusion equation with second-order accuracy, SIAM J. Sci. Comput. 37 (2015) A55–A78.
[22] F. Zeng, C. Li, F. Liu, I. Turner, The use of finite difference/element approximations for solving the time-fractional subdiffusion equation, SIAM J. Sci. Comput. 35 (2013) 2976–3000.
[23] J. Zhao, T. Zhang, R. Corless, Convergence of the compact finite difference method for second-order elliptic equations, Appl. Math. Comput. 182 (2006) 1454–1469.
[24] Y. Zhang, Z. Sun, Alternating direction implicit schemes for the two-dimensional fractional sub-diffusion equations, J. Comput. Phys. 230 (2011) 8713–8728.
[25] M. Zheng, F. Liu, V. Anh, I. Turner, A high-order spectral method for the multi-term time-fractional diffusion equations, Appl. Math. Model. 40 (2016) 4970–4985.