ARTICLE IN PRESS

Journal of Computational Physics xxx (2011) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

Multigrid method for fractional diffusion equations ☆

Hong-Kui Pang [a], Hai-Wei Sun [b,*]

[a] School of Mathematical Sciences, Xuzhou Normal University, Xuzhou, 221116 Jiangsu, China
[b] Department of Mathematics, University of Macau, Macao, China

ARTICLE INFO

Article history:
Received 13 May 2011
Received in revised form 24 September 2011
Accepted 6 October 2011
Available online xxxx

Keywords:
Fractional diffusion equation
Toeplitz matrices
Fast Fourier transform
Multigrid method
Damped-Jacobi method

ABSTRACT

The fractional diffusion equation is discretized by the implicit finite difference scheme with the shifted Grünwald formula. The scheme is unconditionally stable and the coefficient matrix possesses the Toeplitz-like structure. A multigrid method is proposed to solve the resulting system. Meanwhile, the fast Toeplitz matrix–vector multiplication is utilized to lower the computational cost with only $\mathcal{O}(N\log N)$ complexity, where N is the number of the grid points. Numerical experiments are given to demonstrate the efficiency of the method.

1. Introduction

Consider an initial-boundary value problem of a fractional diffusion equation (FDE):

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

where $1<\alpha<2$, $f(x,t)$ is the source term, and $d_\pm(x,t)\geqslant 0$. Here the left-sided and the right-sided fractional derivatives $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ and $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$ are defined in the Grünwald–Letnikov form [23]:

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

where $\lfloor x\rfloor$ denotes the floor of x, $g_0^{(\alpha)}=1$, and

☆ The research was supported by the research grant MYRG140(Y1-L2)-FST11-SHW from University of Macau.
* Corresponding author.
E-mail addresses: panghongkui@163.com (H.-K. Pang), HSun@umac.mo (H.-W. Sun).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.10.005

$$g_k^{(\alpha)} = \frac{(-1)^k}{k!}\alpha(\alpha-1)\cdots(\alpha-k+1), \quad k = 1, 2, \ldots. \tag{2}$$

The FDEs emerge from numerous topics such as modeling chaotic dynamics of classical conservative systems [38], groundwater contaminant transport [2,3], turbulent flow [6,27], and applications in biology [18], finance [24], image processing [1], and physics [28]. Unlike the second-order diffusion equation, there are very few cases of FDEs in which the closed-form analytical solutions are available. Therefore, in the last decade, numerical solutions for the FDEs have been developed intensively [4,10–12,15–17,19–22,29,30,33]. Nevertheless, two conspicuous issues for the FDEs, compared to the second-order diffusion equations, should be addressed:

- Fractional differential operators are nonlocal, hence the stability becomes very sensitive on the corresponding numerical approximations. In [20,21], it was shown that a naive discretization of the fractional diffusion, even though implicit, leads to unconditionally unstable.
- Numerical methods for FDEs tend to generate full coefficient matrices, which require computational cost of $\mathcal{O}(N^3)$ and storage of $\mathcal{O}(N^2)$, where N is the number of the grid points. It is quite different from the second-order diffusion equations which usually yield sparse coefficient matrices with $\mathcal{O}(N)$ nonzero entries and can be solved very efficiently by fast iterative methods with $\mathcal{O}(N)$ complexity.

In 2004, Meerschaet and Tadjeran [20] proposed a shifted Grünwald discretization to approximate the FDE with a left-sided fractional derivative. Later, they extended the same idea to the FDE with two-sided fractional derivatives [21]. Their methods were proved to be unconditionally stable. Nevertheless, their methods, as well as other finite difference or finite element methods [10–12], still yield full coefficient matrices, and then suffer from $\mathcal{O}(N^2)$ storage and $\mathcal{O}(N^3)$ complexity.

This bottleneck was not broken until recently, Wang et al. [35] discovered that the coefficient matrix by the Meerschaet–Tadjeran's method [21] possesses a Toeplitz-like structure. More precisely, such a full matrix can be written as the sum of diagonal-multiply-Toeplitz matrices. Thus the storage requirement is significantly reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. It is well known that Toeplitz matrices possess great structures and properties, and their matrix–vector multiplications can be computed by the fast Fourier transform (FFT) with $\mathcal{O}(N\log N)$ complexity [8,9]. With this advantage, Wang et al. used a banded matrix with banded-width of $\mathcal{O}(\log N)$ to approximate the full coefficient matrix, and then proposed a fast finite different scheme for solving (1). Their proposed scheme, which is actually developed from the Meerschaet–Tadjeran's method, is of only $\mathcal{O}(N\log^2 N)$ complexity at each time step to solve (1). Compared to other numerical methods with $\mathcal{O}(N^3)$ complexity, the computational cost by their scheme is much lowered. However, from the theoretical point of view, the unconditional stability of their scheme has not been proved. As mentioned in [20,21], the stability of numerical methods for the FDE is very sensitive and should be carefully studied. Therefore, it motivates us to develop fast algorithms for the discretized system by the Meerschaet–Tadjeran's finite difference method whose unconditional stability is guaranteed. Most recently, Wang and Wang, two authors of [35], proposed the conjugate gradient normal residual (CGNR) method [36] to solve the discretized system of (1) by the Meerschaet–Tadjeran's method. Their numerical results show that the CGNR method gives a fast convergence provided that the diffusion coefficients $d_{\pm}(x,t)$ are very small (in that case the coefficient matrix is well-conditioned). Nevertheless, if the diffusion coefficients are not too small, the resulting system will become ill-conditioned and hence the CGNR method converges very slow; see, for instance, Section 4.

The multigrid method (MGM) is usually exploited for solving ill-conditioned systems [37,25]. Moreover, the MGM has also been successfully employed to solve ill-conditioned Toeplitz systems [7,13,14,31,32]. In [7], a V-cycle MGM was proposed to solve a class of ill-conditioned Toeplitz systems. The smoothing operator of the MGM was chosen as the damped-Jacobi method which can retain the computational cost with $\mathcal{O}(N\log N)$ complexity for the Toeplitz system. The coarse grid operator was constructed by the Galerkin algorithm in algebraic means. Such a construction of the coarse grid operator can be efficiently implemented for the Toeplitz system. Nevertheless, we note that the coefficient matrix of (1) may not be exactly Toeplitz as it contains terms of diagonal-multiply-Toeplitz matrices. Therefore, the algebraic way to generate the coarse grid operator may distort its Toeplitz-like structure and hence lead to an expensive iteration.

In this paper, we follow the idea in [7] to develop an MGM for solving the discretized system of the FDE (1) by the Meerschaet–Tadjeran's method. Our smoothing operator, as in [7], is the damped-Jacobi which also preserves $\mathcal{O}(N\log N)$ complexity for the Toeplitz-like system. Unlike [7], however, our coarse grid operator is constructed by the discretization of (1) with the coarse grid size [37]. In other words, the coarse grid operator is defined in geometric means and then keeps the Toeplitz-like structure. Therefore, our MGM can be carried out to solve the discretized system with $\mathcal{O}(N\log N)$ complexity in total at each time step. Numerical results will illustrate the efficiency and robustness of our method.

The paper is organized as follows. In Section 2, we discretize (1) by the Meerschaet–Tadjeran's finite difference method and introduce some properties of the Toeplitz matrix. In Section 3 we propose a V-cycle MGM for the discretized system of (1). In Section 4 we report the numerical results. At last we give the concluding remarks in Section 5.

2. Finite difference method and Toeplitz-like structure

Let N and M be positive integers, and $\Delta x = (x_R - x_L)/N$ and $\Delta t = T/M$ be the sizes of spatial grid and time step, respectively. We define a spatial and temporal partition $x_n = x_L + n\Delta x$ for $n = 0, 1, \ldots, N$ and $t_m = m\Delta t$ for $m = 0, 1, \ldots, M$. Let

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

$u_n^{(m)} = u(x_n, t_m), d_{\pm,n}^{(m)} = d_\pm(x_n, t_m)$, and $f_n^{(m)} = f(x_n, t_m)$. In [20,21], Meerschaert and Tadjeran proposed the shifted Grünwald approximations,

$$\frac{\partial^\alpha u(x_n,t_m)}{\partial_+ x^\alpha} = \frac{1}{\Delta x^\alpha}\sum_{k=0}^{n+1} g_k^{(\alpha)} u_{n-k+1}^{(m)} + \mathcal{O}(\Delta x),$$
$$\frac{\partial^\alpha u(x_n,t_m)}{\partial_- x^\alpha} = \frac{1}{\Delta x^\alpha}\sum_{k=0}^{N-n+1} g_k^{(\alpha)} u_{n+k-1}^{(m)} + \mathcal{O}(\Delta x),$$

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

$$\frac{u_n^{(m+1)} - u_n^{(m)}}{\Delta t} = \frac{d_{+,n}^{(m+1)}}{\Delta x^\alpha}\sum_{k=0}^{n+1} g_k^{(\alpha)} u_{n-k+1}^{(m+1)} + \frac{d_{-,n}^{(m+1)}}{\Delta x^\alpha}\sum_{k=0}^{N-n+1} g_k^{(\alpha)} u_{n+k-1}^{(m+1)} + f_n^{(m+1)} \tag{3}$$

is unconditionally stable. We note that $g_k^{(\alpha)}$ satisfy the following proposition.

Proposition 1 (See [20,21,35]). *Let $1 < \alpha < 2$ and $g_k^{(\alpha)}$ be defined in (2). Then we have*

$$\begin{cases} g_0^{(\alpha)} = 1, \quad g_1^{(\alpha)} = -\alpha < 0, \quad g_2^{(\alpha)} > g_3^{(\alpha)} > \cdots > 0, \\ \sum_{k=0}^{\infty} g_k^{(\alpha)} = 0, \quad \sum_{k=0}^{n} g_k^{(\alpha)} < 0, \quad \text{for } n \geqslant 1. \end{cases} \tag{4}$$

Let $u^{(m)} = \left(u_1^{(m)}, u_2^{(m)}, \ldots, u_{N-1}^{(m)}\right)^{\mathsf{T}}, f^{(m)} = \left(f_1^{(m)}, f_2^{(m)}, \ldots, f_{N-1}^{(m)}\right)^{\mathsf{T}}$, and I be the identity matrix with an appropriate size. Then the numerical scheme (3) can be written in the following matrix form

$$\left(I + A_{\Delta x}^{(m+1)}\right) u^{(m+1)} = u^{(m)} + \Delta t f^{(m+1)} \tag{5}$$

with

$$A_{\Delta x}^{(m+1)} = D_{+,\Delta x}^{(m+1)} G_\alpha + D_{-,\Delta x}^{(m+1)} G_\alpha^{\mathsf{T}}, \tag{6}$$

where $D_{\pm,\Delta x}^{(m+1)} = -(\Delta t/\Delta x^\alpha)\cdot \operatorname{diag}\left(d_{\pm,1}^{(m+1)}, \ldots, d_{\pm,N-1}^{(m+1)}\right)$ and

$$G_\alpha = \begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 \\ \vdots & g_2^{(\alpha)} & g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{N-2}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_{N-1}^{(\alpha)} & g_{N-2}^{(\alpha)} & \cdots & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}. \tag{7}$$

We remark that the matrix $I + A_{\Delta x}^{(m+1)}$ is strongly diagonally dominant and nonsingular; see [35].

In the following, we will study how to calculate the action of $A_{\Delta x}^{(m+1)} v$ for some vector v with only $\mathcal{O}(N \log N)$ operations. We note that G_α in (7) is Toeplitz and independent of m. An n-by-n matrix B_n is said to be *Toeplitz* if

$$B_n = \begin{bmatrix} b_0 & b_{-1} & \cdots & b_{2-n} & b_{1-n} \\ b_1 & b_0 & b_{-1} & \cdots & b_{2-n} \\ \vdots & b_1 & b_0 & \ddots & \vdots \\ b_{n-2} & \cdots & \ddots & \ddots & b_{-1} \\ b_{n-1} & b_{n-2} & \cdots & b_1 & b_0 \end{bmatrix}.$$

In order to carry out a fast Toeplitz matrix–vector multiplication, we introduce the circulant matrix. As a special case of Toeplitz matrix, an $n \times n$ matrix is called *circulant* if it is defined with the following form:

$$C_n = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \cdots & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}.$$

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

Moreover, a circulant matrix can be diagonalized by the Fourier matrix F_n; i.e.,

$$C_n = F_n^* \Lambda_n F_n, \tag{8}$$

where the entries of F_n are given by

$$[F_n]_{j,k} = \frac{1}{\sqrt{n}} \exp\left(\frac{2\pi \mathbf{i} jk}{n}\right), \quad 0 \leqslant j,\ k \leqslant n-1$$

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

From (8), we can determine Λ_n in $\mathcal{O}(n \log n)$ operations by taking the FFT of the first column of C_n; see [9]. Suppose v is a given vector. The multiplication $C_n v$ is then computed by a couple of FFTs with $\mathcal{O}(n \log n)$ complexity provided that Λ_n is already obtained.

For the Toeplitz matrix–vector product $B_n v$, we first embed B_n into a $2n \times 2n$ circulant matrix; i.e.,

$$\begin{bmatrix} B_n & \times \\ \times & B_n \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} B_n v \\ \dagger \end{bmatrix}.$$

Now that we are back to the circulant case, the multiplication is carried out as discussed before. The computational cost of $B_n v$ remains as $\mathcal{O}(n \log n)$.

We recall that in (6), $A_{\Delta x}^{(m+1)} = D_{+,\Delta x}^{(m+1)} G_\alpha + D_{-,\Delta x}^{(m+1)} G_\alpha^{\mathsf{T}}$, where G_α is an $(N-1)$-by-$(N-1)$ Toeplitz matrix. Therefore, $A_{\Delta x}^{(m+1)}$ contains two Toeplitz matrices G_α and G_α^{T}, hence the action of $A_{\Delta x}^{(m+1)} v$ generally needs two Toeplitz matrix–vector multiplications. In the following, we present a fast way to calculate the product of $A_{\Delta x}^{(m+1)} v$ which only requires about one Toeplitz matrix–vector multiplication provided that v is a real vector.

Let

$$J = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

be the $(N-1)$-by-$(N-1)$ anti-identity matrix. We note that

$$J^2 = I \quad \text{and} \quad J G_\alpha J = G_\alpha^{\mathsf{T}}.$$

Let $\hat{v} = v + \mathbf{i} J v$. Then $G_\alpha \hat{v}$ is calculated by only one Toeplitz matrix–vector multiplication. On the other hand, we rewrite $G_\alpha \hat{v}$ into two parts:

$$G_\alpha \hat{v} = G_\alpha v + \mathbf{i} G_\alpha J v = G_\alpha v + \mathbf{i} J G_\alpha^{\mathsf{T}} v \equiv v_1 + \mathbf{i} v_2.$$

We then carry out

$$A_{\Delta x}^{(m+1)} v = D_{+,\Delta x}^{(m+1)} \cdot v_1 + D_{-,\Delta x}^{(m+1)} \cdot J v_2 \tag{9}$$

with only two diagonal matrix–vector multiplications whose computational cost can be neglected when compared to the Toeplitz matrix–vector multiplication. Hence the computational cost for calculating the product of $A_{\Delta x}^{m+1} v$ is roughly equal to one Toeplitz matrix–vector multiplication with $\mathcal{O}(N \log N)$ complexity.

3. Multigrid method

3.1. V-cycle multigrid algorithm

For a general linear system

$$A_h u_h = f_h, \tag{10}$$

the V-cycle multigrid can be described as the algorithm below [26], where we have to provide the smoothing operator (or smoother) `smooth` with ν_1 and ν_2 being the numbers of pre- and post-smoothing steps respectively, the prolongation operator I_H^h, the restriction operator I_h^H, the coarse grid operators A_H, and the coarsest grid size h_0.

Algorithm 1: u_h = V-cycle (A_h, u_0, f_h)

1. Pre-smooth: $u_h := \texttt{smooth}^{\nu_1}(A_h, u_0, f_h)$
2. Get residual: $r_h = f_h - A_h u_h$
3. Coarsen: $r_H = I_h^H r_h$
4. If $(H == h_0)$

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

(*Algorithm 1 continued*)

Algorithm 1: $u_h = V$-cycle (A_h, u_0, f_h)

5. Solve: $A_H \xi_H = r_H$
6. Else
7. Recursion: $\xi_H = \text{V-cycle}(A_H, 0, r_H)$
8. Endif
9. Correct: $u_h := u_h + I_H^h \xi_H$
10. Post-smooth: $u_h := \texttt{smooth}^{\nu_2}(A_h, u_h, f_h)$
11. Return u_h

For one-dimensional problems, the classical restriction operator I_h^H is defined as [37]

$$I_h^H = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & \\ & & 1 & 2 & 1 & & \\ & & & & \ddots & & \\ & & & & & 1 & 2 & 1 \end{bmatrix} \tag{11}$$

and the prolongation operator $I_H^h = 2\left(I_h^H\right)^{\top}$. In general, the smoother `smooth` in Algorithm 1 can be written as

$$\texttt{smooth}(A_h, u_0, f_h) = S_h u_0 + (I - S_h) A_h^{-1} f_h, \tag{12}$$

where S_h is the iterative matrix of the smoothing operator. Typical smoothing operators are the damped-Jacobi, Gauss–Seidel, and Richardson iterations; see [5,26,37]. Usually, the Gauss–Seidel is chosen as the smoother for sparse linear systems. Nevertheless, it is too expensive to carry out the Gauss–Seidel for a dense matrix. Alternatively, the damped-Jacobi as the smoother is the best choice for dense Toeplitz matrices; see [7,31,32].

The damped-Jacobi iterative matrix is defined as follows,

$$S_{h,\omega} = I - \omega \cdot \text{diag}\,(A_h)^{-1} \cdot A_h, \tag{13}$$

where ω is the damped parameter. The smoother `smooth` in (12) therefore becomes

$$\texttt{smooth}_{\,\omega}(A_h, u_0, f_h) = S_{h,\omega} u_0 + \omega \cdot \text{diag}\,(A_h)^{-1} \cdot f_h. \tag{14}$$

From the computational point of view, the complexity to carry out one iteration of the damped-Jacobi is roughly the same as that of one matrix–vector multiplication; see (13) and (14).

In this paper, we will employ the V-cycle MGM Algorithm 1 with the damped-Jacobi as smoother to solve the system (5). We note that at each time step, the finest grid operator, the unknown vector, and the right-hand side of (10) become

$$A_h = I + A_{\Delta x}^{(m+1)}, \quad u_h = u^{(m+1)}, \quad \text{and} \quad f_h = u^{(m)} + \Delta t f^{(m+1)}. \tag{15}$$

3.2. Convergence analysis

We now study the convergence of the MGM based on the frame of [7,31]. Firstly, we assume that both $d_{\pm}(x,t) = d > 0$ are constants. We remark that this assumption is only for convenience of our primary investigation for the convergence analysis.

Under the above assumption, A_h in (15) becomes a real symmetric Toeplitz matrix with the following form,

$$A_h = I + A_{\Delta x}^{(m+1)} = I - \frac{d\Delta t}{\Delta x^{\alpha}} \begin{bmatrix} 2g_1^{(\alpha)} & g_0^{(\alpha)} + g_2^{(\alpha)} & g_3^{(\alpha)} & \cdots & g_{N-2}^{(\alpha)} & g_{N-1}^{(\alpha)} \\ g_0^{(\alpha)} + g_2^{(\alpha)} & 2g_1^{(\alpha)} & g_0^{(\alpha)} + g_2^{(\alpha)} & \cdots & \cdots & g_{N-2}^{(\alpha)} \\ g_3^{(\alpha)} & g_0^{(\alpha)} + g_2^{(\alpha)} & 2g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & g_3^{(\alpha)} \\ g_{N-2}^{(\alpha)} & \ddots & \ddots & \ddots & 2g_1^{(\alpha)} & g_0^{(\alpha)} + g_2^{(\alpha)} \\ g_{N-1}^{(\alpha)} & g_{N-2}^{(\alpha)} & \cdots & g_3^{(\alpha)} & g_0^{(\alpha)} + g_2^{(\alpha)} & 2g_1^{(\alpha)} \end{bmatrix} \equiv [a_{|j-k|}]_{(N-1)\times(N-1)}, \tag{16}$$

where

$$a_0 = 1 - 2\gamma g_1^{(\alpha)} = 1 + 2\gamma\alpha, \quad a_1 = -\gamma\left(g_0^{(\alpha)} + g_2^{(\alpha)}\right) = -\gamma[1 + \alpha(\alpha - 1)/2], \tag{17}$$

and $a_k = -\gamma g_{k+1}^{(\alpha)}$ for $k = 2, \ldots, N-2$, with $\gamma = d\Delta t/\Delta x^{\alpha}$.

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

We note that the matrix A_h is strongly diagonally dominant; see [35]. Moreover, the main diagonal entry $a_0 = 1 + 2\gamma\alpha$ is positive. By the following theorem, we conclude that the matrix A_h is symmetric positive definite.

Theorem 1 (See [34, Corollary 1.22, p.23]). *If a matrix is real symmetric strictly diagonally dominant or irreducibly diagonally dominant with positive diagonal entries, then it is positive definite.*

Since A_h is symmetric positive definite, we then define the following inner products which are useful in the convergence analysis of MGMs; see [25, pp. 77–78]:

$$\langle u, v\rangle_0 = \langle \mathrm{diag}(A_h)u, v\rangle, \quad \langle u, v\rangle_1 = \langle A_h u, v\rangle, \quad \langle u, v\rangle_2 = \langle \mathrm{diag}(A_h)^{-1}A_h u, A_h v\rangle. \tag{18}$$

Here $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. Their respective norms are denoted by $\|\cdot\|_j$, $j = 0,1,2$. As in [7,31], the coarser grid operator A_H is defined by the Galerkin algorithm; i.e., $A_H = I_h^H A_h I_H^h$. Furthermore, in Algorithm 1, if the coarsest grid size $H = h_0 = 2h$, i.e., A_H is inverted directly, the algorithm is called the two-grid method (TGM). The TGM is rarely used in practice since the coarse grid operator may still be too large to be solved exactly. However, it is useful from a theoretical point of view as the first step to study the MGM convergence usually begins from the TGM; see [25,26,37]. In the following, we study the convergence of the TGM for (16).

As the smoothing operator is chosen to be the damped-Jacobi, the following theorem shows that $\|S_{h,\omega}\|_1 \leqslant 1$ if ω is properly chosen.

Theorem 2 (See [25, p. 84]). *For a symmetric positive definite matrix A_h, suppose that the damped parameter ω in (13) is properly chosen to fulfill*

$$1/\omega \geqslant \rho(\mathrm{diag}(A_h)^{-1}A_h), \tag{19}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix. Then $S_{h,\omega}$ in (13) satisfies

$$\|S_{h,\omega}e_h\|_1^2 \leqslant \|e_h\|_1^2 - \omega\|e_h\|_2^2, \quad \forall e_h \in \mathbb{R}^{N-1}. \tag{20}$$

The inequality (20) is called the *smoothing condition*. We see that $\|S_{h,\omega}\|_1 \leqslant 1$ if ω satisfies (19). Note that $A_h = I + A_{\Delta x}^{(m+1)}$ is symmetric positive definite and diagonally dominant. Thus we have [26, p. 119]

$$\rho(\mathrm{diag}(A_h)^{-1}[A_h - \mathrm{diag}(A_h)]) \leqslant 1,$$

hence

$$\rho(\mathrm{diag}(A_h)^{-1}A_h) \leqslant 2.$$

Therefore, it suggests that

$$\omega = 1/2, \tag{21}$$

satisfies (19) in Theorem 2.

For the TGM, the correction operator is given by

$$T_{\mathrm{TGM}} = I - I_H^h(A_H)^{-1}I_h^H A_h,$$

hence the convergence factor of the TGM is $\|(S_{h,\omega})^{\nu_2}T_{\mathrm{TGM}}(S_{h,\omega})^{\nu_1}\|_1$; see [25, p. 89]. For simplicity, we only consider the case of $\nu_1 = 0$ and $\nu_2 = 1$. The other cases can be established similarly as we have $\|S_{h,\omega}\|_1 \leqslant 1$. Thus the convergence factor of the TGM is given by $\|S_{h,\omega}\cdot T_{\mathrm{TGM}}\|_1$. The following theorem gives a general estimate on this quantity.

Theorem 3 ([25, p. 89]). *Let A_h be symmetric positive definite and $\omega > 0$ be chosen such that $S_{h,\omega}$ satisfies the smoothing condition (20); i.e.,*

$$\|S_{h,\omega}e_h\|_1^2 \leqslant \|e_h\|_1^2 - \omega\|e_h\|_2^2, \quad \forall e_h \in \mathbb{R}^{N-1}.$$

Suppose that I_H^h has full rank and that there exists a scalar $\beta > 0$ such that

$$\min_{e_H \in \mathbb{R}^{N/2-1}} \|e_h - I_H^h e_H\|_0^2 \leqslant \beta\|e_h\|_1^2, \quad \forall e_h \in \mathbb{R}^{N-1}. \tag{22}$$

Then $\beta \geqslant \omega$ and the convergence factor of the TGM satisfies

$$\|S_{h,\omega}\cdot T_{\mathrm{TGM}}\|_1 \leqslant \sqrt{1 - \omega/\beta}.$$

We note that the damped-Jacobi is chosen as smoother with $\omega = 1/2$ as in (21). Thus, the smoothing condition (20) holds for A_h in (16). Therefore we only have to find a suitable β to establish (22) to get the TGM convergence result; i.e., the convergence factor of the TGM is independent of N.

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

Let L_{N-1} = tridiag$(-1,2,-1)$ be the $(N-1)\times(N-1)$ one-dimensional discrete Laplacian matrix. Then L_{N-1} is also a symmetric positive definite Toeplitz matrix. In the following lemma, $A_{\Delta x}^{(m+1)}$ in (16) is split into two symmetric positive definite matrices, which will be employed to establish (22) later.

Lemma 1. *Let $A_{\Delta x}^{(m+1)}$ be in (16) and $A_{rest} = A_{\Delta x}^{(m+1)} + a_1 L_{N-1}$ with a_1 in (17). Then A_{rest} is symmetric positive definite.*

Proof. Since both $A_{\Delta x}^{(m+1)}$ and L_{N-1} are symmetric Toeplitz, $A_{\rm rest}$ is also symmetric Toeplitz. We denote

$$A_{\rm rest} = A_{\Delta x}^{(m+1)} + a_1 L_{N-1} \equiv [\tilde{a}_{j-k}]_{(N-1)\times(N-1)}.$$

Then, by (17) and (4), we have

$$\tilde{a}_0 = a_0 - 1 + 2a_1 = \gamma\left[-2g_1^{(\alpha)} - 2\left(g_0^{(\alpha)} + g_2^{(\alpha)}\right)\right] > 0, \quad \tilde{a}_1 = \tilde{a}_{-1} = 0 \tag{23}$$

and

$$\tilde{a}_j = \tilde{a}_{-j} = -\gamma g_{j+1}^{(\alpha)} < 0, \quad \text{for} \quad j = 2,\ldots,N-2.$$

Furthermore, by (23) and Proposition 1, for the kth row of $A_{\rm rest}$ ($k = 1,\ldots,N-1$), we have

$$\begin{aligned}\tilde{a}_0 - \sum_{j=k-1\, j\neq 0}^{k-N+1} |\tilde{a}_j| &= \gamma\left[-2g_1^{(\alpha)} - 2\left(g_0^{(\alpha)} + g_2^{(\alpha)}\right)\right] - \gamma\left(\sum_{j=3}^{k} g_j^{(\alpha)} + \sum_{j=3}^{N-k} g_j^{(\alpha)}\right)\\ &\geqslant \gamma\left[-2g_1^{(\alpha)} - 2\left(g_0^{(\alpha)} + g_2^{(\alpha)}\right)\right] - \gamma\left(\sum_{j=3}^{N-1} g_j^{(\alpha)} + \sum_{j=3}^{N-1} g_j^{(\alpha)}\right) = -2\gamma\left(\sum_{j=0}^{N-1} g_j^{(\alpha)}\right) > 0.\end{aligned}$$

Therefore, $A_{\rm rest}$ is strictly diagonally dominant. Recall that $\tilde{a}_0$ in (23), the diagonal entry of $A_{\rm rest}$, is positive. By Theorem 1, we conclude that $A_{\rm rest}$ is symmetric positive definite. □

From Lemma 1, it is easy to obtain

$$\langle e_h, A_h e_h\rangle = \langle e_h, (I - a_1 L_{N-1} + A_{\rm rest}) e_h\rangle \geqslant \langle e_h, (I - a_1 L_{N-1}) e_h\rangle, \quad \forall e_h \in \mathbb{R}^{N-1}. \tag{24}$$

This inequality will be exploited to prove the TGM convergence in the theorem below.

Theorem 4. *Suppose that A_h is as in (16) and $\omega = 1/2$ in (13) is chosen such that $S_{h,\omega}$ satisfies the smoothing condition (20). Then the convergence factor of the TGM satisfies*

$$\|S_{h,\omega}\cdot T_{\rm TGM}\|_1 \leqslant \sqrt{3}/2.$$

Proof. For any $e_h = (e_1, e_2, \ldots, e_{N-1})^{\rm T} \in \mathbb{R}^{N-1}$, we define

$$e_H = (\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_{N/2-1})^{\rm T} \in \mathbb{R}^{N/2-1},$$

where

$$\tilde{e}_j = e_{2j}, \quad 1 \leqslant j \leqslant N/2 - 1.$$

For ease of indexing, we set $e_0 = e_N = 0$. Note that I_H^h is defined in (11), the norm $\|\cdot\|_0$ in (18), and diag $(A_h) = a_0 I$, we have

$$\|e_h - I_H^h e_H\|_0^2 = a_0 \sum_{j=0}^{N/2-1}\left(e_{2j+1} - \frac{1}{2}e_{2j} - \frac{1}{2}e_{2j+2}\right)^2.$$

Thus (22) is proved if we can bound the right hand side above by $\beta\langle e_h, A_h e_h\rangle$ for some β independent of e_h. To do so, we observe that for the right hand side above, we have

$$\begin{aligned}a_0 \sum_{j=0}^{N/2-1}\left(e_{2j+1} - \frac{1}{2}e_{2j} - \frac{1}{2}e_{2j+2}\right)^2 &= a_0 \sum_{j=0}^{N/2-1}\left(e_{2j+1}^2 + \frac{1}{4}e_{2j}^2 + \frac{1}{4}e_{2j+2}^2 - e_{2j}e_{2j+1} - e_{2j+1}e_{2j+2} + \frac{1}{2}e_{2j}e_{2j+2}\right)\\ &\leqslant a_0 \sum_{j=0}^{N/2-1}\left(e_{2j+1}^2 + \frac{1}{4}e_{2j}^2 + \frac{1}{4}e_{2j+2}^2 - e_{2j}e_{2j+1} - e_{2j+1}e_{2j+2} + \frac{1}{4}e_{2j}^2 + \frac{1}{4}e_{2j+2}^2\right)\\ &\leqslant a_0 \sum_{j=0}^{N/2-1}\left(e_{2j+1}^2 + \frac{1}{2}e_{2j}^2 + \frac{1}{2}e_{2j+2}^2 - e_{2j}e_{2j+1} - e_{2j+1}e_{2j+2}\right) = a_0 \sum_{j=1}^{N-1}\left(e_j^2 - e_j e_{j+1}\right).\end{aligned}$$

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

From the above inequality, we have

$$\sum_{j=1}^{N-1} e_j^2 \geqslant \sum_{j=1}^{N-1} e_j e_{j+1}. \tag{25}$$

On the other hand, by Lemma 1 and inequalities (24) and (25), we have

$$\begin{aligned}\|e_h\|_1^2 = \langle e_h, A_h e_h\rangle &\geqslant \langle e_h, (I - a_1 L_{N-1}) e_h\rangle = \sum_{j=1}^{N-1}\left[(1-2a_1)e_j^2 + 2a_1 e_j e_{j+1}\right] = \sum_{j=1}^{N-1}\left[\left(\frac{1}{2}-2a_1\right)e_j^2 + \frac{1}{2}e_j^2 + 2a_1 e_j e_{j+1}\right]\\ &\geqslant \left(\frac{1}{2}-2a_1\right)\sum_{j=1}^{N-1}\left(e_j^2 - e_j e_{j+1}\right).\end{aligned}$$

Hence (22) holds with

$$\beta = \frac{2a_0}{1-4a_1} > 0.$$

According to Theorem 3, with $\omega = 1/2$, and a_0, a_1 in (17), we have

$$\begin{aligned}\|S_{h,\omega}\cdot T_{\mathrm{TGM}}\|_1 &\leqslant \sqrt{1-\frac{\omega}{\beta}} = \sqrt{1-\frac{1-4a_1}{4a_0}} = \sqrt{1-\frac{1+4\gamma+2\gamma\alpha(\alpha-1)}{4(1+2\gamma\alpha)}} = \sqrt{1-\frac{1+2\gamma\alpha+2\gamma[1+(\alpha-1)^2]}{4(1+2\gamma\alpha)}} \leqslant \frac{\sqrt{3}}{2}\\ &< 1. \qquad \square\end{aligned}$$

Theorem 4 shows that the convergence of the TGM is independent of N in the finest grid level. In other words, the TGM converges linearly.

3.3. Implementation and computational cost

As we have mentioned above, the assumption of $d_{\pm}(x,t)$ being constants is just for convenience of our studying the TGM convergence. In general, the fractional diffusion coefficients $d_{\pm}(x,t)$ are not constants. Hence $A_{\Delta x}^{(m+1)}$ in (5) is no longer a symmetric positive definite Toeplitz matrix. Instead, $A_{\Delta x}^{(m+1)}$ is Toeplitz-like and is of the form of a diagonal-multiply-Toeplitz matrix. If the coarse grid operators A_H are constructed by the Galerkin algorithm as in [7], their Toeplitz-like structure may be distorted. As a result, the fast Toeplitz matrix–vector multiplication cannot be utilized to carry out the damped-Jacobi iteration on coarse grid levels. This shortcoming will lead an expensive iteration. In order to implement the V-cycle MGM Algorithm 1 efficiently, we define A_H by a geometry way; i.e., A_H are constructed by the coefficient matrices in (5) with coarse grid sizes. Therefore, all A_H still retain the Toeplitz-like structure and hence we can implement the damped-Jacobi on coarse grid levels using the fast Toeplitz matrix–vector multiplication (9). We remark that this treatment is reasonable as it has been widely used in the literature where it is called the geometric MGM; see, [5,26,37].

In the following, we provide all inputs to carry out the V-cycle MGM Algorithm 1 to solve (5) at each time step:

- The finest grid operator $A_h = I + A_{\Delta x}^{(m+1)}$ with the grid size $h = \Delta x$. The coarse grid operators $A_H = I + A_{2^k\Delta x}^{(m+1)}$ with the grid sizes $H = 2^k\Delta x$, for $1 \leqslant k \leqslant \log_2 N - 1$. The coarsest grid size is $h_0 = (x_R - x_L)/2$.
- The right hand side $f_h = u^{(m)} + \Delta t\, f^{(m+1)}$ and the initial guess $u_0 = u^{(m)}$.
- The smoothing operator is the damped-Jacobi (14), where one pre-smoother with $\omega_{\mathrm{pre}} = 1$ and one post-smoother with $\omega_{\mathrm{post}} = 1/2$ are carried out at each grid level; see, for instance, [7].
- The prolongation operator I_H^h and the restriction operator I_h^H are defined in (11).

From Algorithm 1 and the damped-Jacobi iteration method (14), we note that the main cost on each grid level depends on the matrix–vector multiplication of $A_{2^k\Delta x}^{(m+1)} v$ for some vector v for $k = 0, 1, \ldots, \log_2 N - 1$. Recall that all $A_{2^k\Delta x}^{(m+1)}$ possess the Toeplitz-like structure and hence the multiplication $A_{2^k\Delta x}^{(m+1)} v$ can be calculated by (9) with $\mathcal{O}(N/2^k \log(N/2^k))$ complexity. Therefore, the total computational cost per V-cycle MGM iteration for solving (5) is roughly

$$\left(1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^{\log_2 N - 1}}\right)\cdot \mathcal{O}(N\log N) = \mathcal{O}(N\log N).$$

Thus, if the V-cycle MGM converges linearly, the computational cost for solving (9) at each time step is of $\mathcal{O}(N\log N)$ in total. We remark that the MGM convergence has not been proved in this paper. However, in the following section, our numerical experiments show that the V-cycle MGM is robust and its convergence factor is independent of N.

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

4. Numerical results

In this section, we employ the V-cycle MGM listed as Algorithm 1 in Section 3 to solve the linear system (5). The stopping criterion is

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

where $r^{(k)}$ is the residual vector after k iterations. In all tables, "N" denotes the number of spatial grid points, "M" denotes the number of time steps, "Error" denotes the error between the true solution and the approximation under the infinity norm at the last time step, "CPU(s)" denotes the total CPU time in seconds for solving the whole discretized system, and "Iter" denotes the average number of iterations required to solve (5) at each time step. As comparisons, we also carry out the Gaussian elimination (GE) method and the CGNR method [36], with the same stopping criterion of the MGM, to solve the corresponding system. All numerical experiments are run in MATLAB 7.11 (R2010b) on a PC with the configuration: Intel (R) Xeon (R) CPU W3520 2.67 GHz and 6 GB RAM.

Example 1 (*See* [35]). Consider an FDE (1) whose data are given as follows: $\alpha = 1.8$, $x_L = 0$, $x_R = 2$, $T = 1$, $d_+(x,t) = \Gamma(1.2)x^{1.8}$, $d_-(x,t) = \Gamma(1.2)(2-x)^{1.8}$, and

$$f(x,t) = -32e^{-t}\left(x^2 + \frac{1}{8}(2-x)^2(8+x^2) - \frac{5}{2}[x^3 + (2-x)^3] + \frac{25}{22}[x^4 + (2-x)^4]\right).$$

The initial condition is chosen as $u(x,0) = 4x^2(2-x)^2$. Then the true solution to the corresponding FDE is given by $u(x,t) = 4e^{-t}x^2(2-x)^2$; see [21,35].

We implement the GE method, the CGNR method, and the V-cycle MGM to solve the discretized system of this equation, respectively. Table 1 shows that the GE method is the most time consuming method as it requires $\mathcal{O}(N^3)$ complexity at each time step. By the Toeplitz-like structure of the coefficient matrix, both of the CGNR method and the MGM can be carried out with $\mathcal{O}(N \log N)$ operations for one iteration. Nevertheless, the average number of iterations by the CGNR method is much larger than that by the V-cycle MGM. In addition, the average number of iterations of the MGM are independent of the spatial grid number N, while the CGNR method needs more iterations as N increases.

Example 2. We modify an example in [35] that the diffusion coefficients are multiplied by a parameter. Consider an anomalous diffusive process of a Gaussian pulse. The corresponding FDE is homogeneous with $\alpha = 1.5$ and the diffusion coefficients

$$d_+(x,t) = \delta(1 + x^2 + t^2), \quad d_-(x,t) = \delta[1 + (2-x)^2 + t^2],$$

where δ is a parameter (the case as $\delta = 0.001$ is the example in [35]). Other data are provided as follows, $x_L = 0$, $x_R = 2$, and $T = 1$. The initial condition is given by

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

with the mean $x_c = 1.2$ and the standard deviation $\sigma = 0.08$. Since the closed-form analytical solution is not available for this problem, analogous to [35], we use the solution generated with the finest spatial grid size and the smallest time step size as the reference solution.

The aim of this example is to demonstrate how the condition number affects the numerical performances of the iterative methods. Therefore, only the CGNR method and the V-cycle MGM are carried out and their numerical results are presented in Table 2. We remark that the GE method is more expensive than both of the CGNR method and the MGM for this example. Therefore, it is not necessary to show its results in Table 2.

As in Table 1, we show the average number of iterations and the CPU time in Table 2. We also provide the condition numbers ("Cond") of the resulting coefficient matrices at the first time step with different spatial grid sizes and values of δ. We remark that with the same value of δ and the spatial grid size, the condition numbers of the coefficient matrices in other time steps are similar to that at the first time step.

Table 1
Comparisons for solving Example 1 by the GE method, the CGNR method, and the V-cycle MGM method, respectively.

N	M	Error	GE	CGNR		MGM	
			CPU(s)	Iter	CPU(s)	Iter	CPU(s)
2^6	2^5	1.7434e−2	0.06	77.2	0.28	7.0	0.20
2^7	2^6	8.3522e−3	0.46	213.8	1.63	7.0	0.49
2^8	2^7	4.0836e−3	5.45	613.8	12.72	7.0	1.24
2^9	2^8	2.0185e−3	120.95	1768.8	99.86	7.0	3.16
2^{10}	2^9	1.0034e−3	2671.86	5076.0	888.61	6.0	7.27

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

Table 2
Numerical performances by the CGNR method and the *V*-cycle MGM for solving Example 2, respectively.

δ	N	M	Error	CGNR		MGM		Cond.
				Iter	CPU(s)	Iter	CPU(s)	
0.001	2^6	2^5	9.2822e−3	4.1	0.02	4.0	0.12	1.09
	2^7	2^6	5.3180e−3	4.1	0.04	4.0	0.30	1.13
	2^8	2^7	2.6723e−3	4.1	0.10	4.0	0.75	1.18
	2^9	2^8	1.1862e−3	4.0	0.26	3.6	1.70	1.26
	2^{10}	2^9	4.0250e−4	4.1	0.85	3.2	4.08	1.38
	2^{11}	2^{10}		4.1	3.05	3.0	11.22	1.54
0.01	2^6	2^5	9.5954e−3	11.4	0.04	6.0	0.18	1.86
	2^7	2^6	5.4136e−3	12.7	0.10	6.0	0.43	2.27
	2^8	2^7	2.7261e−3	14.5	0.32	6.0	1.11	2.84
	2^9	2^8	1.2149e−3	16.6	0.98	5.1	2.44	3.64
	2^{10}	2^9	4.1294e−4	20.0	3.65	5.0	6.27	4.78
	2^{11}	2^{10}		24.7	16.66	4.0	14.76	6.37
0.1	2^6	2^5	8.5691e−4	59.3	0.20	6.4	0.19	9.42
	2^7	2^6	3.9352e−4	88.8	0.70	6.2	0.46	13.55
	2^8	2^7	1.7621e−4	121.8	2.49	5.4	1.00	19.30
	2^9	2^8	7.2951e−5	165.7	9.42	5.2	2.49	27.37
	2^{10}	2^9	2.3618e−5	222.9	39.13	5.1	6.43	38.71
	2^{11}	2^{10}		297.5	196.05	5.1	18.55	54.70
1	2^6	2^5	7.5423e−5	82.4	0.27	8.2	0.24	71.40
	2^7	2^6	3.1841e−5	216.2	1.66	8.0	0.59	115.24
	2^8	2^7	1.3793e−5	510.8	10.35	7.3	1.36	175.46
	2^9	2^8	5.6769e−6	1059.4	60.46	7.1	3.35	258.39
	2^{10}	2^9	1.8502e−6	2088.0	368.86	6.9	8.69	373.59
	2^{11}	2^{10}		3325.4	2186.03	6.3	22.93	534.78

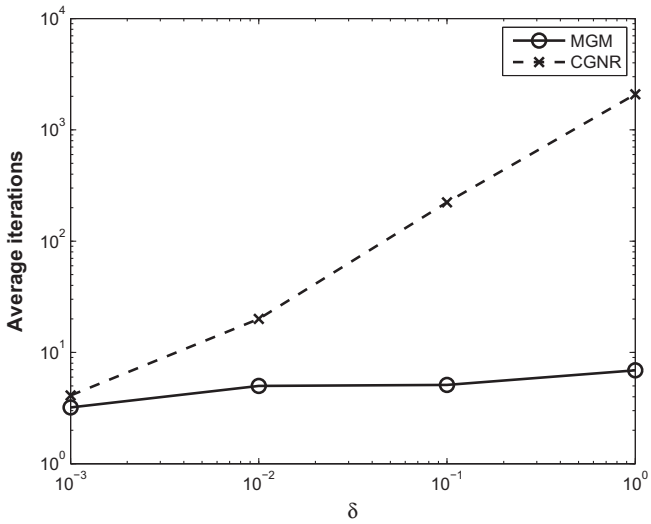

Fig. 1. Average number of iterations versus values of δ by the MGM and the CGNR method with $N = 2^{10}$ and $M = 2^9$ for solving Example 2.

From Table 2, we see that the condition numbers of the coefficient matrices become larger as δ increases. In other words, the resulting systems gradually become ill-conditioned. Moreover, the average numbers of iterations by our MGM are independent of N for all δ. In contrast, the CGNR method needs more iterations when N increases even for a small δ (for $\delta \geqslant 0.01$). We also plot Fig. 1 to show the average number of iterations by the MGM and the CGNR method versus the values of δ with $N = 2^{10}$ and $M = 2^9$. We see from Fig. 1 that the average number of iterations of the CGNR method increases much faster than that of the MGM.

5. Concluding remarks

In this paper, we have employed the *V*-cycle MGM to solve the discretized system (5) of the FDE (1) by the Meerschaet–Tadjeran's finite difference method. In our MGM, the damped-Jacobi iterative method is chosen as the smoothing operator

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005

and the coarse grid operators are constructed by the geometry means to retain the Toeplitz-like structure. The computational cost has been reduced to $\mathcal{O}(N \log N)$ complexity by utilizing the fast Toeplitz matrix–vector multiplication. Two examples have been given to illustrate the efficiency and robustness of the MGM. The TGM convergence has been theoretically studied. The MGM convergence, however, has not been considered, despite the fact that the numerical experiments have shown its linear convergence. In our future work, we will apply the MGM to solve multi-dimensional FDEs.

References

[1] J. Bai, X. Feng, Fractional-order anisotropic diffusion for image denoising, IEEE Trans. Image Proc. 16 (2007) 2492–2502.
[2] D. Benson, S.W. Wheatcraft, M.M. Meerschaert, Application of a fractional advection–dispersion equation, Water Resour. Res. 36 (2000) 1403–1413.
[3] D. Benson, S.W. Wheatcraft, M.M. Meerschaert, The fractional-order governing equation of Lévy motion, Water Resour. Res. 36 (2000) 1413–1423.
[4] B. Beumer, M. Kovács, M.M. Meerschaert, Numerical solutions for fractional reaction–diffusion equations, Comput. Math. Appl. 55 (2008) 2212–2226.
[5] W. Briggs, A Multigrid Tutorial, SIAM, Philadelphia, PA, 1987.
[6] B.A. Carreras, V.E. Lynch, G.M. Zaslavsky, Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence model, Phys. Plasma 8 (2001) 5096–5103.
[7] R. Chan, Q. Chang, H. Sun, Multigrid method for ill-conditioned symmetric Toeplitz systems, SIAM J. Sci. Comput. 19 (1998) 516–529.
[8] R. Chan, X. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, 2007.
[9] R. Chan, M. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev. 38 (1996) 427–482.
[10] M. Cui, Compact finite difference method for the fractional diffusion equation, J. Comput. Phys. 228 (2009) 7792–7804.
[11] W. Deng, Finite element method for the space and time fractional Fokker–Planck equation, SIAM J. Numer. Anal. 47 (2008) 204–226.
[12] V.J. Ervin, N. Heuer, J.P. Roop, Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation, SIAM J. Numer. Anal. 45 (2007) 572–591.
[13] G. Fiorentino, S. Serra, Multigrid methods for Toeplitz matrices, Calcolo 28 (1991) 283–305.
[14] T. Huckle, J. Staudacher, Multigrid precodnitioning and Toeplitz matrices, ETNA 13 (2002) 81–105.
[15] T.A.M. Langlands, B.I. Henry, The accuracy and stability of an implicit solution method for the fractional diffusion equation, J. Comput. Phys. 205 (2005) 719–736.
[16] Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, J. Comput. Phys. 225 (2007) 1533–1552.
[17] F. Liu, V. Anh, I. Turner, Numerical solution of the space fractional Fokker–Planck equation, J. Comput. Appl. Math. 166 (2004) 209–219.
[18] R.L. Magin, Fractional Calculus in Bioengineering, Begell House Publishers, 2006.
[19] M.M. Meerschaert, H.P. Scheffler, C. Tadjeran, Finite difference methods for two-dimensional fractional dispersion equation, J. Comput. Phys. 211 (2006) 249–261.
[20] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for fractional advection–dispersion flow equations, J. Comput. Appl. Math. 172 (2004) 65–77.
[21] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math. 56 (2006) 80–90.
[22] D.A. Murio, Implicit finite difference approximation for time fractional diffusion equations, Comput. Math. Appl. 56 (2008) 1138–1145.
[23] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.
[24] M. Raberto, E. Scalas, F. Mainardi, Waiting-times and returns in high-frequency financial data: an empirical study, Phys. A 314 (2002) 749–755.
[25] J. Ruge, K. Stuben, Algebraic Multigrid, in: S. McCormick (Ed.), Multigrid Methods, vol. 4, SIAM, Philadephia, 1987.
[26] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.
[27] M.F. Shlesinger, B.J. West, J. Klafter, Lévy dynamics of enhanced diffusion: application to turbulence, Phys. Rev. Lett. 58 (1987) 1100–1103.
[28] I.M. Sokolov, J. Klafter, A. Blumen, Fractional kinetics, Phys. Today Nov. (2002) 48–54.
[29] E. Sousa, Finite difference approximates for a fractional advection diffusion problem, J. Comput. Phys. 228 (2009) 4038–4054.
[30] L. Su, W. Wang, Z. Yang, Finite difference approximations for the fractional advection-diffusion equation, Phys. Lett. A 373 (2009) 4405–4408.
[31] H. Sun, R. Chan, Q. Chang, A note on the convergence of the two-grid method for Toeplitz systems, Comput. Math. Appl. 34 (1997) 11–18.
[32] H. Sun, X. Jin, Q. Chang, Convergence of the multigrid method for ill-conditioned block Toeplitz systems, BIT 41 (2001) 179–190.
[33] C. Tadjeran, M.M. Meerschaert, H.P. Scheffler, A second-order accurate numerical approximation for the fractional diffusion equation, J. Comput. Phys. 213 (2006) 205–213.
[34] R.S. Varga, Matrix Iterative Analysis, second ed., Springer-Verlag, Berlin, Heideberg, 2000.
[35] H. Wang, K. Wang, T. Sircar, A direct $O(N\log^2 N)$ finite difference method for fractional diffusion equations, J. Comput. Phys. 229 (2010) 8095–8104.
[36] K. Wang, H. Wang, A fast characteristic finite difference method for fractional advection-diffusion equations, Adv. Water Resour. 34 (2011) 810–816.
[37] P. Wesseling, An Introduction to Multigrid Methods, John Wiley & Sons Ltd., Chichester, 1992.
[38] G.M. Zaslavsky, D. Stevens, H. Weitzner, Self-similar transport in incomplete chaos, Phys. Rev. E 48 (1993) 1683–1694.

Please cite this article in press as: H.-K. Pang, H.-W. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. (2011), doi:10.1016/j.jcp.2011.10.005