Journal of Computational Physics 229 (2010) 8095–8104

Contents lists available at ScienceDirect

Journal of Computational Physics

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

# A direct $O(N\log^2 N)$ finite difference method for fractional diffusion equations

Hong Wang [b,a], Kaixin Wang [b,*], Treena Sircar [a]

[a] Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA
[b] School of Mathematics, Shandong University, Jinan, Shandong 250100, China

ARTICLE INFO

*Article history:*
Received 22 February 2010
Received in revised form 25 June 2010
Accepted 9 July 2010
Available online 21 July 2010

*Keywords:*
Anomalous diffusion
Circulant and Toeplitz matrices
Fast finite difference methods
Fast Fourier transform
Fractional diffusion equations

ABSTRACT

Fractional diffusion equations model phenomena exhibiting anomalous diffusion that can not be modeled accurately by the second-order diffusion equations. Because of the non-local property of fractional differential operators, the numerical methods have full coefficient matrices which require storage of $O(N^2)$ and computational cost of $O(N^3)$ where $N$ is the number of grid points.

In this paper we develop a fast finite difference method for fractional diffusion equations, which only requires storage of $O(N)$ and computational cost of $O(N\log^2 N)$ while retaining the same accuracy and approximation property as the regular finite difference method. Numerical experiments are presented to show the utility of the method.



## 1. Introduction

Fractional diffusion equations model phenomena exhibiting anomalous diffusion that cannot be modeled accurately by the second-order diffusion equations. For instance, in contaminant transport in groundwater flow the solutes moving through aquifers do not generally follow a Fickian, second-order partial differential equation because of large deviations from the stochastic process of Brownian motion. Instead, a governing equation with a fractional-order anomalous diffusion provides a more adequate and accurate description of the movement of the solutes [3].

The history of fractional calculus dates back to late 17th century [27], but it was not until late 20th century when fractional differential equations began to find wide applications. Since then, fractional diffusion equations have been used in modeling turbulent flow [5,32], chaotic dynamics of classical conservative systems [38], groundwater contaminant transport [2,3], and applications in biology [20], physics [33], chemistry [14], and even finance [29,30] (see [13,24,25,28] for additional information). While analytical methods, such as the Fourier transform method, the Laplace transform methods, and the Mellin transform method, have been developed to seek closed-form analytical solutions for fractional partial differential equations [28], there are very few cases in which the closed-form analytical solutions are available, as in the context of integer-order partial differential equations. Therefore, numerical means have to be used in general.

In the last decade or so, extensive research has been carried out on the development of numerical methods for fractional partial differential equations, including finite difference methods [1,7,15,17,21–23,26,34–36,39], finite element methods [9–11,19], and spectral methods [16,18]. Compared to the second-order diffusion equations, the fractional diffusion equations have the following salient features: (i) Fractional differential operators are nonlocal and so raise subtle stability issues on the corresponding numerical approximations; (ii) Numerical methods for fractional diffusion equations tend to yield full

* Corresponding author.
*E-mail addresses:* kx.wang@mail.sdu.edu.cn, kxwang@yahoo.cn (K. Wang).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.07.011

coefficient matrices, which have computational cost of $O(N^3)$ and storage of $O(N^2)$ with $N$ being the number of unknowns. This is in contrast to numerical methods for second-order diffusion equations which usually generate banded coefficient matrices of $O(N)$ nonzero entries and can be solved by fast solution methods such as multigrid methods, domain decomposition methods, and wavelet methods.

In [22,23] it was shown that a naive discretization of the anomalous diffusion, even though implicit, leads to unstable discretizations. A shifted Grünwald discretization was proposed to approximate the fractional diffusion equation and was proved to be stable and convergent. Numerical experiments showed that these methods generate satisfactory numerical results. Alternatively, several explicit finite difference schemes, such as the fractional central and Lax-Wendroff schemes, were proposed and analyzed in [34]. In [18] a finite difference approximation was used in the temporal discretization while a Legendre spectral approximation was adopted in spatial discretization in the numerical solution of time-fractional diffusion equations. Finite element methods for fractional diffusion equations were proposed and analyzed in [9–11]. These methods still generate full coefficient matrices and so require storage of $O(N^2)$ and computational cost of $O(N^3)$.

The goal of this paper is to develop a fast finite difference method for space-fractional diffusion equations, which has a significantly reduced storage requirement of $O(N)$ and computational cost $O(N\log^2 N)$ while retaining the same accuracy as the existing numerical methods. The rest of the paper is organized as follows. In Section 2 we present the fractional diffusion equation and its regular finite difference approximation, and go over their properties. In Section 3 we study how to reduce the computational cost in the inversion of the coefficient matrix from $O(N^3)$ to $O(N\log^2 N)$ in the new finite difference method. In Section 4 we analyze how to reduce the computational cost of the right-hand side from $O(N^2)$ to $O(N\log N)$ and to reduce the storage requirement from $O(N^2)$ to $O(N)$, respectively. In Section 5 we carry out numerical experiments to compare the performance of the newly developed method with the regular finite difference method.

## 2. Fractional diffusion equations and its finite difference approximation

We consider the following initial-boundary value problem of an anomalous diffusion equation of order $1 < \alpha < 2$ [6,22,23]

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

Note that the case $1 < \alpha < 2$ is useful in applications [2]. It is also the physically meaningful case, as explained in [31]. Here the left-sided (+) and the right-sided (−) fractional derivatives $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ and $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$ can be defined in the (computationally feasible) Grünwald–Letnikov form [28]

$$\begin{aligned}
\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} &= \lim_{h\to 0^+}\frac{1}{h^\alpha}\sum_{k=0}^{\lfloor (x-x_L)/h\rfloor} g_k^{(\alpha)} u(x-kh,t),\\
\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} &= \lim_{h\to 0^+}\frac{1}{h^\alpha}\sum_{k=0}^{\lfloor (x_R-x)/h\rfloor} g_k^{(\alpha)} u(x+kh,t),
\end{aligned} \tag{2}$$

where $\lfloor x\rfloor$ represents the floor of $x$ and $g_k^{(\alpha)} = (-1)^k\binom{\alpha}{k}$ with $\binom{\alpha}{k}$ being the fractional binomial coefficients. It is clear that the coefficients $g_k^{(\alpha)}$ can be evaluated recursively

$$g_0^{(\alpha)} = 1, \quad g_k^{(\alpha)} = \left(1 - \frac{\alpha+1}{k}\right) g_{k-1}^{(\alpha)} \quad \text{for } k \geqslant 1. \tag{3}$$

Moreover, the coefficients $g_k^{(\alpha)}$ satisfy the following properties [22,23,28]

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

In this paper we focus on the development of a fast numerical method for problem (1). For the existence and uniqueness of the weak solution to fractional partial differential equations, we refer to [16]. Let $N$ and $M$ be positive integers and $h = (x_R - x_L)/N$ and $\Delta t = T/M$ be the sizes of spatial grid and time step, respectively. We define a spatial and temporal partition $x_i = x_L + i\,h$ for $i = 0,1,\ldots,N$ and $t^m = m\Delta t$ for $m = 0,1,\ldots,M$. Let $u_i^m = u(x_i,t^m)$, $d_{+,i}^m = d_+(x_i,t^m)$, $d_{-,i}^m = d_-(x_i,t^m)$, and $f_i^m = f(x_i,t^m)$. While the first-order time derivative in (1) can be discretized by a standard first-order time difference quotient, the discretization of the fractional spatial derivative requires careful investigation. Meerschaert and Tadjeran [22,23] showed that a fully implicit finite difference scheme with a direct truncation of the series in (2) turns out to be unstable! Actually even the case $\alpha = 2$ is unstable if the one-sided difference is used. Instead, they proposed to use the following shifted Grünwald approximations

$$\begin{aligned}\frac{\partial^\alpha u(x_i,t^m)}{\partial_+ x^\alpha} &= \frac{1}{h^\alpha}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^m + O(h),\\ \frac{\partial^\alpha u(x_i,t^m)}{\partial_- x^\alpha} &= \frac{1}{h^\alpha}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^m + O(h)\end{aligned} \tag{5}$$

and proved that the corresponding implicit finite difference scheme

$$\frac{u_i^{m+1}-u_i^m}{\Delta t} - \frac{d_{+,i}^{m+1}}{h^\alpha}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^{m+1} - \frac{d_{-,i}^{m+1}}{h^\alpha}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^{m+1} = f_i^{m+1} \tag{6}$$

is unconditionally stable and convergent. Numerical experiments show that this scheme generates very satisfactory numerical approximations.

Let $u^m = [u_1^m, u_2^m, \ldots, u_{N-1}^m]^T$, $f^m = [f_1^m, f_2^m, \ldots, f_{N-1}^m]^T$, $A^m = \left[a_{ij}^m\right]_{i,j=1}^{N-1}$, and $I$ be the identity matrix of order $N-1$. Then the numerical scheme (6) can be expressed in the following matrix form

$$(I + A^{m+1})u^{m+1} = u^m + \Delta t f^{m+1}. \tag{7}$$

Here the entries of matrix $A^{m+1}$ are given by

$$a_{ij}^{m+1} = \begin{cases} -\left(r_{+,i}^{m+1} + r_{-,i}^{m+1}\right)g_1^{(\alpha)}, & j = i,\\ -\left(r_{+,i}^{m+1} g_2^{(\alpha)} + r_{-,i}^{m+1} g_0^{(\alpha)}\right), & j = i-1,\\ -\left(r_{+,i}^{m+1} g_0^{(\alpha)} + r_{-,i}^{m+1} g_2^{(\alpha)}\right), & j = i+1,\\ -r_{+,i}^{m+1} g_{i-j+1}^{(\alpha)}, & j < i-1,\\ -r_{-,i}^{m+1} g_{j-i+1}^{(\alpha)}, & j > i+1, \end{cases} \tag{8}$$

where

$$r_{+,i}^{m+1} = d_{+,i}^{m+1}\Delta t/h^\alpha, \quad r_{-,i}^{m+1} = d_{-,i}^{m+1}\Delta t/h^\alpha. \tag{9}$$

It is clear that $a_{i,j}^{m+1} \leqslant 0$ for all $i \neq j$. We further conclude from (4) and (8) that

$$\begin{aligned} a_{i,i}^{m+1} - \sum_{j=1, j\neq i}^{N-1} |a_{ij}^{m+1}| &= -\left(r_{+,i}^{m+1} + r_{-,i}^{m+1}\right)g_1^{(\alpha)} - r_{i,+}^{m+1}\sum_{k=0,k\neq 1}^{i} g_k^{(\alpha)} - r_{-,i}^{m+1}\sum_{k=0,k\neq 1}^{N-i} g_k^{(\alpha)}\\ &\geqslant -\left(r_{+,i}^{m+1} + r_{-,i}^{m+1}\right)g_1^{(\alpha)} - \left(r_{i,+}^{m+1} + r_{-,i}^{m+1}\right)\sum_{k=0,k\neq 1}^{\infty} g_k^{(\alpha)}\\ &= -\left(r_{+,i}^{m+1} + r_{-,i}^{m+1}\right)g_1^{(\alpha)} + \left(r_{i,+}^{m+1} + r_{-,i}^{m+1}\right)g_1^{(\alpha)} = 0. \end{aligned} \tag{10}$$

Thus, the coefficient matrix $I + A^{m+1}$ is a nonsingular, strictly diagonally dominant $M$-matrix, and so the scheme is monotone [37].

## 3. A finite difference method with banded coefficient matrices

Like many other numerical methods for fractional partial differential equations, the finite difference method (7) has a full coefficient matrix. Consequently, it requires an $O(N^3)$ operations to solve the system (7) and an $O(N^2)$ storage per time step. Furthermore, the well known iterative methods developed for sparse systems arising from the discretization of integer-order partial differential equations, such as multigrid method, domain decomposition method, and wavelet method, fail to significantly reduce the computational cost since each iteration requires $O(N^2)$ computations.

As the first step, we want to develop a fast finite difference method that has significantly reduced computational cost compared to the regular finite difference method (7) while retaining the same accuracy. More specifically, we impose the following desired properties:

- The method should have a banded coefficient matrix $I + A_k^{m+1}$ instead of the full matrix $I + A^{m+1}$ in the scheme (7).
- $A_k^{m+1}$ can accurately approximate the full matrix $A^{m+1}$.
- $I + A_k^{m+1}$ can be inverted with a significantly reduced computational cost than $I + A^{m+1}$.
- Inversion of the coefficient matrix $I + A_k^{m+1}$ has approximately the same computational cost as the evaluation of the right side of the method.

To explore the feasibility of approximating the full matrix $A^{m+1}$ by a banded matrix, we need to investigate the decay property of the entries of $A^{m+1}$ as the column indices moves away from the diagonal. We note from (4) that $g_k^{(\alpha)}$ decreases monotonically to zero as $k$ tends to infinity. To find out the rate of the decay, we rewrite $g_k^{(\alpha)}$ as

$$g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k} = \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}, \tag{11}$$

where $\Gamma(\cdot)$ is the Gamma function. We utilize the asymptotic expansion for the Gamma function

$$\Gamma(x) = \sqrt{2\pi} x^{x-\frac{1}{2}} e^{-x} \left(1 + O\left(\frac{1}{x}\right)\right), \quad x \to +\infty \tag{12}$$

to conclude that

$$g_k^{(\alpha)} = \frac{1}{\Gamma(-\alpha) k^{\alpha+1}} \left(1 + O\left(\frac{1}{k}\right)\right). \tag{13}$$

In other words, $g_k^{(\alpha)}$ decreases to zero asymptotically at a rate of $\alpha + 1$. We combine (8) and (13) to conclude that the remainder sum of the coefficients $a_{ij}^{m+1}$ (up to the factors $r_{-,i}^{m+1}$ and $r_{+,i}^{m+1}$) outside of the $2k+1$ diagonals decay at a rate of $\alpha$

$$\sum_{j \geqslant i+k} g_{j-i+1}^{(\alpha)} + \sum_{j \leqslant i-k} g_{i-j+1}^{(\alpha)} = O(k^{-\alpha}). \tag{14}$$

Recall from (4) that $\sum_{j=0}^{\infty} \left| g_j^{(\alpha)} \right| = 2\alpha$. Based on these discussions, we split the full matrix $A^{m+1}$ in (7) as follows

$$A^{m+1} = A_k^{m+1} + A_0^{m+1}, \tag{15}$$

where $A_k^{m+1}$ contains the $2k+1$ diagonals of $A^{m+1}$ and zero entries elsewhere, and $A_0^{m+1} = A^{m+1} - A_k^{m+1}$ contains the remaining nonzero entries of $A^{m+1}$. We see from (14) that

$$\frac{\left\| A_0^{m+1} \right\|_\infty}{\left\| A^{m+1} \right\|_\infty} = O(k^{-\alpha}) \to 0 \quad \text{as } k \to \infty, \tag{16}$$

where $\|A^{m+1}\|_\infty = \max_{1 \leqslant i \leqslant N-1} \sum_{j=1}^{N-1} \left| a_{ij}^{m+1} \right|$ is the $\infty$-norm of a matrix. Namely,

$$\frac{\left\| A^{m+1} - A_k^{m+1} \right\|_\infty}{\left\| A^{m+1} \right\|_\infty} = O(k^{-\alpha}) \to 0 \quad \text{as } k \to \infty. \tag{17}$$

On the other hand, as $k$ increases the computational cost of inverting $I + A_k^{m+1}$ also increases. In fact, a direct solution method requires $O(k^2 N)$ operations. In addition, we will show in the next subsection that the computational cost in evaluating the right-hand side of the fast finite difference method is $O(N \log N)$. Based on these observations, we choose the bandwidth $k = \log N$ in $A_k^{m+1}$. This guarantees that the computational cost of inverting $I + A_k^{m+1}$ is $O(N \log^2 N)$, which is close to the computational cost of evaluating the right-hand side. Meanwhile, the approximation property (17) is still valid and now takes the form

$$\frac{\left\| A^{m+1} - A_k^{m+1} \right\|_\infty}{\left\| A^{m+1} \right\|_\infty} = O(\log^{-\alpha} N) \to 0 \quad \text{as } N \to \infty. \tag{18}$$

Namely, as the number of unknowns $N$ increases, the relative weight of the banded matrix $A_k^{m+1}$ over the full matrix $A^{m+1}$ increases too, even if the bandwidth $k$ increases only as $\log N$ in contrast to the linear increase of the width of the full matrix $A^{m+1}$.

In short, the banded matrix $A_k^{m+1}$ possesses the desired properties we discussed at the beginning of this subsection. We hence split the scheme (7) as follows

$$\left(I + A_k^{m+1}\right) u^{m+1} = u^m - A_0^{m+1} u^{m+1} + \Delta t f^{m+1}. \tag{19}$$

The remaining issue in the development of the fast methods is how to approximate the $u^{m+1}$ in the second term on the right-hand side of the scheme. A straightforward approximation of the$u^{m+1}$ by $u^m$ would deteriorate the accuracy of the scheme.

To enhance the accuracy of the approximation, we evaluate the $u^{m+1}$ on the right-hand side of (19) by approximating the unknown increment $u^{m+1} - u^m$ by the known increment $u^m - u^{m-1}$ at the previous time step

$$u^{m+1} = u^m + (u^{m+1} - u^m) \approx u^m + (u^m - u^{m-1}) = 2u^m - u^{m-1}, \tag{20}$$

which yields the following approximation $\hat{u}^{m+1}$ of $u^{m+1}$

$$\hat{u}^{m+1} = \begin{cases} 2u^m - u^{m-1}, & \text{if } m \geqslant 1, \\ u^0, & \text{if } m = 0. \end{cases} \tag{21}$$

We substitute the $\hat{u}^{m+1}$ for $u^{m+1}$ in the second term on the right-hand side of the scheme (19) to obtain the following finite difference scheme

$$\begin{aligned}&\left(I+A_k^{m+1}\right)u^{m+1}=\left(I-2A_0^{m+1}\right)u^m+A_0^{m+1}u^{m-1}+\Delta t f^{m+1},\quad m\geqslant 1,\\&\left(I+A_k^1\right)u^1=\left(I-A_0^1\right)u^0+\Delta t f^1.\end{aligned}\tag{22}$$

## 4. A fast $O(N\log^2 N)$ finite difference method with an $O(N)$ storage

In the finite difference method (22), the computational cost of inverting the coefficient matrix has been reduced to $O(N\log^2 N)$. However, the computational cost of the right-hand side and the storage requirement are both $O(N^2)$. The goal of this section is to reduce the storage requirement to $O(N)$ and the computational cost of right-hand side to $O(N\log N)$.

We conclude from (8) that the coefficient matrix $A^{m+1}$ can be decomposed as follows

$$A^{m+1}=-\mathrm{diag}(r_+^{m+1})A_L-\ \mathrm{diag}(r_-^{m+1})A_R.\tag{23}$$

Here $\mathrm{diag}(r_+^{m+1})$ and $\mathrm{diag}(r_-^{m+1})$ are diagonal matrices of order $N-1$ with their $i$th entries $r_{+,i}^{m+1}$ and $r_{-,i}^{m+1}$ being defined in (9) for $i=1,2,\ldots,N-1$. The matrices $A_L$ and $A_R$ are matrices of order $N-1$ and are defined by

$$A_L=\begin{bmatrix}g_1^{(\alpha)}&g_0^{(\alpha)}&0&\ldots&0&0\\g_2^{(\alpha)}&g_1^{(\alpha)}&g_0^{(\alpha)}&\ddots&\ddots&0\\\vdots&g_2^{(\alpha)}&g_1^{(\alpha)}&\ddots&\ddots&\vdots\\\vdots&\ddots&\ddots&\ddots&\ddots&0\\g_{N-2}^{(\alpha)}&\ddots&\ddots&\ddots&g_1^{(\alpha)}&g_0^{(\alpha)}\\g_{N-1}^{(\alpha)}&g_{N-2}^{(\alpha)}&\cdots&\cdots&g_2^{(\alpha)}&g_1^{(\alpha)}\end{bmatrix},$$

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

Thus, we need only to store $r_+^{m+1}=\left[r_{+,1}^{m+1},r_{+,2}^{m+1},\ldots,r_{+,N-1}^{m+1}\right]^T$, $r_-^{m+1}=\left[r_{-,1}^{m+1},r_{-,2}^{m+1},\ldots,r_{-,N-1}^{m+1}\right]^T$, and $g^{(\alpha)}=\left[g_0^{(\alpha)},g_1^{(\alpha)},\ldots,g_{N-1}^{(\alpha)}\right]^T$ which have $3N-2$ parameters, instead of the full matrix $A^{m+1}$ which has $(N-1)^2$ parameters. In particular, the vector $g^{(\alpha)}$ is independent of space or time and can be stored a priori for a given $\alpha$ and $N$.

We next study how to reduce the computational cost of the right-hand side of scheme (22), which involves the multiplication of $A^{m+1}$ (or a portion of it) with an arbitrary vector, from $O(N^2)$ to $O(N\log N)$. We conclude from (23) that this in turn requires the multiplication of $A_L$ and $A_R$ with an arbitrary vector in $O(N\log N)$ operations, since $\mathrm{diag}(r_+^{m+1})$ and $\mathrm{diag}(r_-^{m+1})$ are diagonal matrices.

### 4.1. Toeplitz and circulant matrices

We note that both matrices $A_L$ and $A_R$ are Toeplitz matrices. A Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant. In general, an $n\times n$ Toeplitz matrix $T_n$ is completely determined by a sequence of $2n-1$ numbers $\{t_i\}_{i=1-n}^{n-1}$ such that the $(i,j)$-entry of the matrix $T_n(i,j)=t_{j-i}$ for $i,j=1,\ldots,n$, i.e.,

$$T_n=\begin{bmatrix}t_0&t_1&t_2&\ldots&t_{n-2}&t_{n-1}\\t_{-1}&t_0&t_1&\ldots&t_{n-3}&t_{n-2}\\t_{-2}&t_{-1}&t_0&\ddots&\ddots&t_{n-3}\\\vdots&\vdots&\ddots&\ddots&\ddots&\vdots\\t_{2-n}&t_{3-n}&\ddots&\ddots&t_0&t_1\\t_{1-n}&t_{2-n}&t_{3-n}&\ldots&t_{-1}&t_0\end{bmatrix}.$$

To describe the fast algorithm, we also need circulant matrices. A circulant matrix is a matrix in which each row vector is rotated one element to the right relative to the preceding row vector. In general, an $n \times n$ circulant matrix $C_n$ is completely determined by a sequence of $n$ numbers $\{c_i\}_{i=0}^{n-1}$ such that the $(i,j)$-entry of the matrix $C_n(i,j) = c_{(j-i) \bmod n}$ for $i,j = 1,\ldots,n$.

$$C_n = \begin{bmatrix} c_0 & c_1 & c_2 & \ldots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ldots & c_{n-3} & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \ddots & \ddots & c_{n-3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & c_3 & \ddots & \ddots & c_0 & c_1 \\ c_1 & c_2 & c_3 & \ldots & c_{n-1} & c_0 \end{bmatrix}.$$

It is known that a circulant matrix $C_n$ can be decomposed as follows [8,12]

$$C_n = F_n^{-1} \mathrm{diag}(F_n c) F_n, \tag{24}$$

where $c = [c_0, c_{n-1}, c_{n-2}, \cdots, c_2, c_1]^T$ is the first column vector of $C_n$ and $F_n$ is the $n \times n$ discrete Fourier transform matrix in which the $(j,l)$-entry $F_n(j,l)$ of the matrix $F_n$ is given by

$$F_n(j,l) = \frac{1}{\sqrt{n}} \exp\left(-\frac{2\pi i j l}{n}\right) \quad 0 \leqslant j,\ l \leqslant n-1, \tag{25}$$

where $\mathrm{i} = \sqrt{-1}$.

It is clear that a circulant matrix $C_n$ is a Toeplitz matrix with $c_{-l} = c_{n-l}$ for $l = 1,\ldots,n-1$ but the converse is not true. Nevertheless, an $n \times n$ Toeplitz matrix $T_n$ can be embedded into a $2n \times 2n$ circulant matrix $C_{2n}$ as follows [4,12]

$$C_{2n} = \begin{bmatrix} T_n & B_n \\ B_n & T_n \end{bmatrix}, \quad B_n = \begin{bmatrix} 0 & t_{1-n} & \ldots & t_{-2} & t_{-1} \\ t_{n-1} & 0 & t_{1-n} & \ldots & t_{-2} \\ \vdots & t_{n-1} & 0 & \ddots & \vdots \\ t_2 & \vdots & \ddots & \ddots & t_{1-n} \\ t_1 & t_2 & \ldots & t_{n-1} & 0 \end{bmatrix}. \tag{26}$$

### 4.2. A fast $O(N \log N)$ algorithm for the evaluation of $A^{m+1}u$

To efficiently evaluate the right-hand side of the scheme (22) in $O(N \log N)$ operations, we use the following $O(N \log N)$ algorithm to evaluate the matrix–vector multiplication of $A^{m+1}u$ (or $A_0^{m+1}u$), based on the decomposition (23) of the matrix $A^{m+1}$, the diagonalization (24) of a circulant matrix and the embedding (26):

1. Introduce two $(2N-2) \times (2N-2)$ matrices and one $2N-2$ vector

$$C_{2N-2,L} = \begin{bmatrix} A_L & B_L \\ B_L & A_L \end{bmatrix}, \quad C_{2N-2,R} = \begin{bmatrix} A_R & B_R \\ B_R & A_R \end{bmatrix}, \quad u_{2N-2} = \begin{bmatrix} u \\ 0 \end{bmatrix}. \tag{27}$$

Here $B_L$ and $B_R$ are defined as in (26) with $n = N-1$ and $T_n$ being replaced by $A_L$ and $A_R$ respectively. It is clear that

$$C_{2N-2,L} u_{2N-2} = \begin{bmatrix} A_L u \\ B_L u \end{bmatrix}, \quad C_{2N-2,R} u_{2N-2} = \begin{bmatrix} A_R u \\ B_R u \end{bmatrix}. \tag{28}$$

Thus, the matrix–vector products $A_L u$ and $A_R u$ can be obtained as the first half of the matrix–vector products $C_{2N-2,L} u_{2N-2}$ and $C_{2N-2,R} u_{2N-2}$, respectively.

2. Evaluate the matrix–vector products $w_{2N-2} = F_{2N-2} u_{2N-2}$ in $O(N \log N)$ operations. In fact, $F_{2N-2} u_{2N-2}$ is the discrete Fourier transform of $u_{2N-2}$, which can be achieved in $O((2N)\log(2N)) = O(N \log N)$ operations via the fast Fourier transform (FFT).
3. Similarly evaluate $v_{2N-2,L} = F_{2N-2} c_{2N-2,L}$ and $v_{2N-2,R} = F_{2N-2} c_{2N-2,R}$ in $O(N \log N)$ operations, where $c_{2N-2,L}$ and $c_{2N-2,R}$ are the first column vectors of $C_{2N-2,L}$ and $C_{2N-2,R}$, respectively.
4. Evaluate the Hadamard products $z_{2N-2,L} = w_{2N-2} \cdot v_{2N-2,L} = [w_1 v_{1,L}, \ldots, w_{2N-2} v_{2N-2,L}]^T$ and $z_{2N-2,R} = w_{2N-2} \cdot v_{2N-2,R} = [w_1 v_{1,R}, \ldots, w_{2N-2} v_{2N-2,R}]^T$ in $O(N)$ operations.
5. Evaluate $y_{2N-2,L} = F_{2N-2}^{-1} z_{2N-2,L}$ and $y_{2N-2,R} = F_{2N-2}^{-1} z_{2N-2,R}$ in $O(N \log N)$ operations via inverse FFT. Combining (27) and (28) yields that

$$y_{2N-2,L} = \begin{bmatrix} y_L \\ y'_L \end{bmatrix} = C_{2N-2,L} u_{2N-2} = \begin{bmatrix} A_L u \\ B_L u \end{bmatrix},$$
$$y_{2N-2,R} = \begin{bmatrix} y_R \\ y'_R \end{bmatrix} = C_{2N-2,R} u_{2N-2} = \begin{bmatrix} A_R u \\ B_R u \end{bmatrix}. \tag{29}$$

6. Evaluate the Hadamard products $u_L = r_+^{m+1} \cdot y_L$ and $u_R = r_-^{m+1} \cdot y_R$ in $O(N)$ operations. Use (23) to evaluate $A^{m+1}u = -u_L - u_R$ in $O(N)$ operations.

## 5. Numerical experiments

In this section we carry out numerical experiments to study the performance of the fast finite difference method developed in this paper and to compare its performance with the finite difference methods with full coefficient matrices which were developed in [22,23] and were shown to generate very satisfactory results.

### 5.1. Simulation of an example with a known analytical solution

We consider the fractional diffusion Eq. (1) with an anomalous diffusion of order $\alpha = 1.8$ and the left-sided and right-sided diffusion coefficients

$$d_+(x,t) = \Gamma(1.2)x^{1.8}, \quad d_-(x,t) = \Gamma(1.2)(2-x)^{1.8}. \tag{30}$$

The spatial domain is $[x_L, x_R] = [0,2]$, the time interval is $[0,T] = [0,1]$. The source term and the initial condition are given by

$$f(x,t) = -32e^{-t}\left[x^2 + (2-x)^2 + 0.125x^2(2-x)^2 - 2.5(x^3 + (2-x)^3) + \frac{25}{22}(x^4 + (2-x)^4)\right], \tag{31}$$
$$u_0(x) = 4x^2(2-x)^2.$$

The true solution to the corresponding fractional diffusion Eq. (1) is given by [23]

$$u(x,t) = 4e^{-t}x^2(2-x)^2. \tag{32}$$

In the numerical experiments, we solve the problem by both the fast finite difference method and the regular finite difference method (6) and denote their solutions by $u_{FFD}^m$ and $u_{FD}^m$, respectively. Let $u^m$ be the numerical solution $u_{FFD}^m$ or $u_{FD}^m$ at time step $t^m$ and $u(x,t^m)$ be the true solution to problem (1). In Table 1 we present the errors $\|u_{FFD}^M - u(\cdot,t^M)\|_{L^\infty}$ and $\|u_{FD}^M - u(\cdot,t^M)\|_{L^\infty}$ for different spatial mesh sizes and time steps. These results are very encouraging and show that fast finite difference method developed in this paper generates numerical solutions with same accuracy as the regular finite difference method, despite the fact that the former has significantly reduced the storage and computational cost of the latter from $O(N^2)$ and $O(N^3)$ to $O(N)$ and $O(N\log^2 N)$, respectively. In the current context, with the fast finite difference method we need only to invert a banded coefficient matrix with the bandwidth $21 = 2k+1$ where $k = \log N$ instead of the full matrix of 1024 in the regular finite difference method. We also present a representative plot with $N = 256$ and $M = 128$ in Fig. 1, which shows that the fast finite difference solution and the regular finite difference solution both sit on the curve of the true solution without noticeable artifacts.

### 5.2. Simulation of the fundamental solution of the fractional diffusion equation

In this example run we use the fast finite difference method and the regular finite difference method (6) to solve for the fundamental solution of the homogeneous fractional diffusion Eq. (1) which is subject to the initial condition of a Dirac delta function $\delta(x)$ located at $x = 0$. The fundamental solution $u(x,t)$ is expressed via the inverse Fourier transform as follows [3]

**Table 1**
Comparison of the fast finite difference (*FFD*) method with the regular finite difference (*FD*) method in the simulation of the fractional diffusion problem with a known analytical solution.

| | $N$ | $M$ | Error | CPU (s) |
|---|---|---|---|---|
| *FD* | $2^6$ | $2^5$ | $1.74342 \times 10^{-2}$ | $1.09 \times 10$ |
| | $2^7$ | $2^6$ | $8.35225 \times 10^{-3}$ | $5.83 \times 10$ |
| | $2^8$ | $2^7$ | $4.08363 \times 10^{-3}$ | $4.24 \times 10^2$ |
| | $2^9$ | $2^8$ | $2.01853 \times 10^{-3}$ | $3.28 \times 10^3$ |
| | $2^{10}$ | $2^9$ | $1.00343 \times 10^{-3}$ | $2.61 \times 10^4$ |
| *FFD* | $2^6$ | $2^5$ | $1.55820 \times 10^{-2}$ | 1.84 |
| | $2^7$ | $2^6$ | $7.02644 \times 10^{-3}$ | 7.74 |
| | $2^8$ | $2^7$ | $3.18051 \times 10^{-3}$ | $3.37 \times 10$ |
| | $2^9$ | $2^8$ | $1.41064 \times 10^{-3}$ | $1.50 \times 10^2$ |
| | $2^{10}$ | $2^9$ | $5.95001 \times 10^{-4}$ | $6.53 \times 10^2$ |

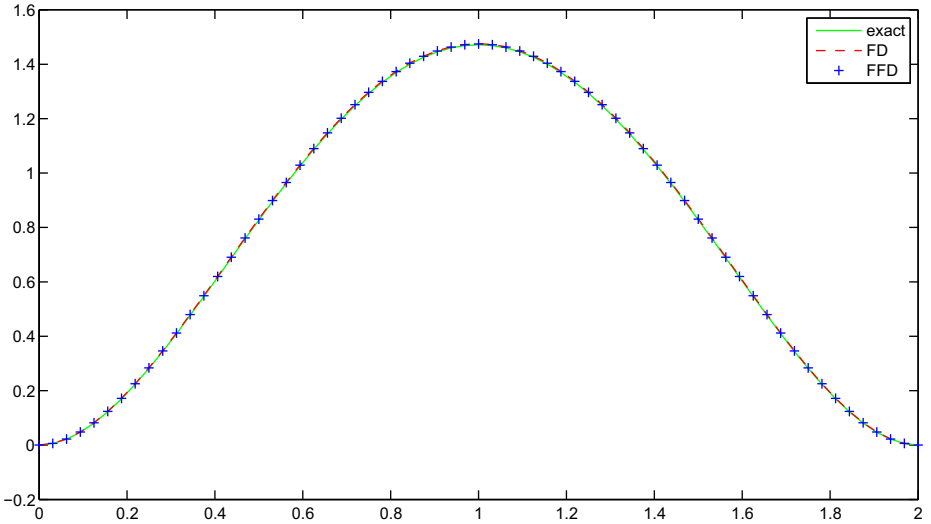


**Fig. 1.** The true solution $u$ (marked by "—"), the fast finite difference solution $u_{FFD}$ (marked by "+"), and the finite difference solution $u_{FD}$ (marked by "– –") in Section 5.1 at time $T = 1$ with $h = \frac{1}{128}$ and $\Delta t = \frac{1}{128}$.

$$u(x,t) = \frac{1}{\pi}\int_0^\infty e^{-2D\left|\cos\left(\frac{\pi\alpha}{2}\right)\right|t\xi^\alpha}\cos(\xi x)d\xi.$$

In the numerical experiment, the data are chosen as follows: $\alpha = 1.8$, $(x_L, x_R) = (-1, 1)$, $(0, T) = (0, 1)$, $d_+(x,t) = d_-(x,t) = D = 0.005$, and $f(x,t) = 0$. The initial condition for both numerical methods is chosen to be a discrete delta function, which is chosen to be $1/h$ at $x = 0$ and 0 at all other nodes.

We present the errors $\left\|u^M_{FFD} - u(\cdot, t^M)\right\|_{L^2}$ and $\left\|u^M_{FD} - u(\cdot, t^M)\right\|_{L^2}$ for different spatial mesh sizes and time steps in Table 2. These results again show that fast finite difference method developed in this paper generates numerical solutions with same accuracy as the regular finite difference method, even though the former has significantly reduced storage and computational cost from the latter. Finally, we present a representative plot with $N = 256$ and $M = 128$ in Fig. 2, which again shows that the fast finite difference method and the regular finite difference method generate solutions with the same accuracy without noticeable artifacts.

### 5.3. Simulation of an anomalously diffused Gaussian pulse

In this example run we simulate an anomalous diffusive process of a Gaussian pulse subject to the homogeneous fractional diffusion Eq. (1) with the initial condition given by

$$u_0(x) = \exp\left(-\frac{(x - x_c)^2}{2\sigma^2}\right).$$

In the numerical experiments, the data is chosen as follows: the order $\alpha = 1.5$, $(x_L, x_R) = (0, 2)$, $(0, T) = (0, 1)$, the mean $x_c = 1.2$ and the standard deviation $\sigma = 0.08$. The diffusion coefficients are chosen to depend on $x$ and $t$: $d_+(x,t) = 0.001(1 + x^2 + t^2)$ and $d_-(x,t) = 0.001(1 + (2 - x)^2 + t^2)$.

**Table 2**
Comparison of the fast finite difference (*FFD*) method with the regular finite difference (*FD*) method in the simulation of the fundamental solution of the fractional diffusion equation.

| | $N$ | $M$ | Error | CPU (s) |
|---|---|---|---|---|
| *FD* | $2^6$ | $2^5$ | $7.14116 \times 10^{-2}$ | 8.75 |
| | $2^7$ | $2^6$ | $2.13480 \times 10^{-2}$ | $5.65 \times 10$ |
| | $2^8$ | $2^7$ | $7.46298 \times 10^{-3}$ | $4.34 \times 10^2$ |
| | $2^9$ | $2^8$ | $2.95111 \times 10^{-3}$ | $3.20 \times 10^3$ |
| | $2^{10}$ | $2^9$ | $1.69132 \times 10^{-3}$ | $2.53 \times 10^4$ |
| *FFD* | $2^6$ | $2^5$ | $7.14034 \times 10^{-2}$ | 1.87 |
| | $2^7$ | $2^6$ | $2.13571 \times 10^{-2}$ | 7.69 |
| | $2^8$ | $2^7$ | $7.47406 \times 10^{-3}$ | $3.50 \times 10$ |
| | $2^9$ | $2^8$ | $2.95296 \times 10^{-3}$ | $1.45 \times 10^2$ |
| | $2^{10}$ | $2^9$ | $1.69132 \times 10^{-3}$ | $6.21 \times 10^2$ |

Since the closed-form analytical solution is not available for this problem, we use the regular finite difference method with small time step $\Delta t = \frac{1}{80}$ and fine space grid $h = \frac{1}{1024}$ to generate a fine-scale numerical solution as a reference solution. We then compare the fast finite difference solution and the regular finite difference solution on coarse spatial meshes and time steps with the reference solution to measure the errors as in Sections 5.1 and 5.2 and present the results in Table 4. We have the same observations as we did in Sections 5.1 and 5.2.

In summary, the numerical experiments in this section show significant reduction of computational time, which coincides with the theoretical analysis. For example, with 1024 computational nodes, the new scheme developed in this paper has about 40 times of CPU reduction than the standard scheme. This is in addition to the significant reduction in the storage. In short, these results indeed show the utility of the method. Finally, even though we did not present a theoretical proof of the stability of the proposed numerical scheme, the numerical results presented in Table 3 indicate that the new scheme has the same stability constraint as the standard finite difference scheme which was proven to be unconditionally stable [22].

**Table 3**
Comparison of the fast finite difference (*FFD*) method with the regular finite difference (*FD*) method with $N = 2^8$.

| | $M = 2^3$ | $M = 2^4$ | $M = 2^5$ |
|---|---|---|---|
| *FD* | $2.12997 \times 10^{-1}$ | $9.98252 \times 10^{-2}$ | $4.63279 \times 10^{-2}$ |
| *FFD* | $2.16415 \times 10^{-1}$ | $1.00613 \times 10^{-1}$ | $4.65139 \times 10^{-2}$ |

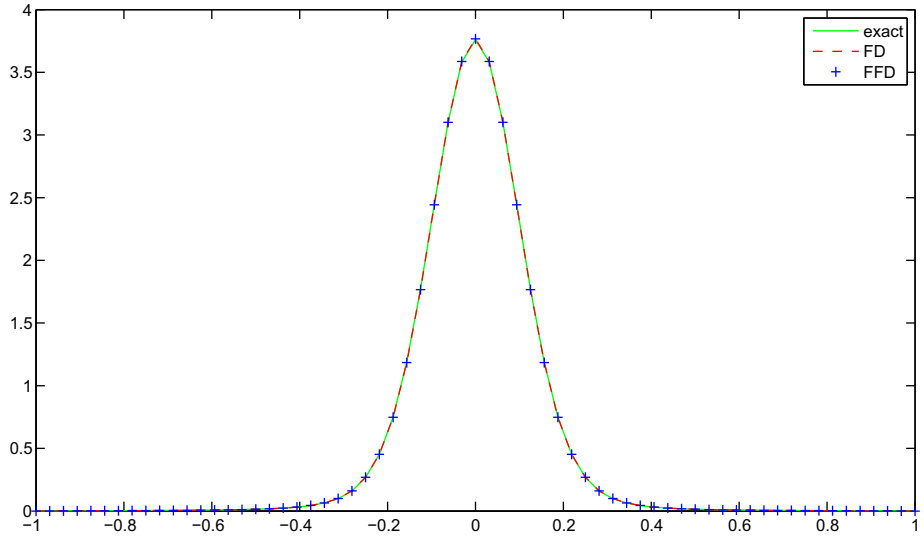


**Fig. 2.** The true solution $u$ (marked by "—"), the fast finite difference solution $u_{FFD}$ (marked by "+"), and the finite difference solution $u_{FD}$ (marked by "– –") in Section 5.2 at time $T = 1$ with $h = \frac{1}{128}$ and $\Delta t = \frac{1}{128}$.

**Table 4**
Comparison of the fast finite difference (*FFD*) method with the regular finite difference (*FD*) method in the simulation of an anomalously diffused Gaussian pulse.

| | $N$ | $M$ | Error | CPU (s) |
|---|---|---|---|---|
| *FD* | $2^6$ | $2^5$ | $9.16435 \times 10^{-3}$ | 9.16 |
| | $2^7$ | $2^6$ | $5.20005 \times 10^{-3}$ | $5.19 \times 10$ |
| | $2^8$ | $2^7$ | $2.55436 \times 10^{-3}$ | $3.73 \times 10^2$ |
| | $2^9$ | $2^8$ | $1.07155 \times 10^{-3}$ | $2.88 \times 10^3$ |
| | $2^{10}$ | $2^9$ | $1.69132 \times 10^{-3}$ | $2.53 \times 10^4$ |
| *FFD* | $2^6$ | $2^5$ | $9.16451 \times 10^{-3}$ | 1.51 |
| | $2^7$ | $2^6$ | $5.20020 \times 10^{-3}$ | 6.74 |
| | $2^8$ | $2^7$ | $2.55434 \times 10^{-3}$ | $3.02 \times 10$ |
| | $2^9$ | $2^8$ | $1.07144 \times 10^{-3}$ | $1.32 \times 10^2$ |
| | $2^{10}$ | $2^9$ | $2.88486 \times 10^{-4}$ | $5.78 \times 10^2$ |

## Acknowledgments

This work was supported in part by the National Science Foundation under Grant No. EAR-0934747. The authors would like to express their sincere thanks to the referees for their very helpful comments and suggestions, which greatly improved the quality of this paper.

## References

[1] B. Beumer, M. Kovács, M.M. Meerschaert, Numerical solutions for fractional reaction–diffusion equations, Comput. Math. Appl. 55 (2008) 2212–2226.
[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] A. Böttcher, B. Silbermann, Introduction to Large Truncated Toeplitz Matrices, Springer, New York, 1999.
[5] B.A. Carreras, V.E. Lynch, G.M. Zaslavsky, Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence models, Phys. Plasma 8 (2001) 5096–5103.
[6] A. Chaves, Fractional diffusion equation to describe Lévy flights, Phys. Lett. A 239 (1998) 13–16.
[7] M. Cui, Compact finite difference method for the fractional diffusion equation, J. Comput. Phys. 228 (2009) 7792–7804.
[8] P.J. Davis, Circulant Matrices, Wiley-Intersciences, New York, 1979.
[9] W. Deng, Finite element method for the space and time fractional Fokker–Planck equation, SIAM J. Numer. Anal. 47 (2008) 204–226.
[10] 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.
[11] V.J. Ervin, J.P. Roop, Variational formulation for the stationary fractional advection dispersion equation, Numer. Methods Part. Different. Equat. 22 (2005) 558–576.
[12] R.M. Gray, Toeplitz and circulant matrices: a review, Found. Trends Commun. Inform. Theory 2 (3) (2006) 155–239.
[13] R. Hilfer, Applications of Fractional Calculus in Physics, Word Scientific, Singapore, 2000.
[14] J.W. Kirchner, X. Feng, C. Neal, Fractal stream chemistry and its implications for contaminant transport in catchments, Nature 403 (2000) 524–526.
[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] X. Li, C. Xu, The existence and uniqueness of the week solution of the space-time fractional diffusion equation and a spectral method approximation, Commun. Comput. Phys. 8 (2010) 1016–1051.
[17] R. Lin, F. Liu, V. Anh, I. Turner, Stability and convergence of a new explicit finite-difference approximation for the variable-order nonlinear fractional diffusion equation, Appl. Math. Comput. 212 (2009) 435–445.
[18] Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, J. Comput. Phys. 225 (2007) 1533–1552.
[19] F. Liu, V. Anh, I. Turner, Numerical solution of the space fractional Fokker–Planck equation, J. Comput. Appl. Math. 166 (2004) 209–219.
[20] R.L. Magin, Fractional Calculus in Bioengineering, Begell House Publishers, 2006.
[21] M.M. Meerschaert, H.P. Scheffler, C. Tadjeran, Finite difference methods for two-dimensional fractional dispersion equation, J. Comput. Phys. 211 (2006) 249–261.
[22] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for fractional advection–dispersion flow equations, J. Comput. Appl. Math. 172 (2004) 65–77.
[23] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math. 56 (2006) 80–90.
[24] R. Metler, J. Klafter, The random walk's guide to anomalous diffusion: a fractional dynamics approach, Phys. Rep. 339 (2000) 1–77.
[25] R. Metler, J. Klafter, The restaurant at the end of random walk: recent developments in the description of anomalous transport by fractional dynamics, J. Phys. A 37 (2004) R161–R208.
[26] D.A. Murio, Implicit finite difference approximation for time fractional diffusion equations, Comput. Math. Appl. 56 (2008) 1138–1145.
[27] K.B. Oldham, J. Spanier, The Fractional Calculus, Academic Press, New York, 1974.
[28] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.
[29] M. Raberto, E. Scalas, F. Mainardi, Waiting-times and returns in high-frequency financial data: an empirical study, Physica 314 (2002) 749–755.
[30] L. Sabatelli, S. Keating, J. Dudley, P. Richmond, Waiting time distributions in financial markets, Eur. Phys. J. B 27 (2002) 273–275.
[31] R. Schumer, D.A. Benson, M.M. Meerschaert, S.W. Wheatcraft, Eulerian derivation of the fractional advection–dispersion equation, J. Contam. Hydrol. 38 (2001) 69–88.
[32] M.F. Shlesinger, B.J. West, J. Klafter, Lévy dynamics of enhanced diffusion: application to turbulence, Phys. Rev. Lett. 58 (1987) 1100–1103.
[33] I.M. Sokolov, J. Klafter, A. Blumen, Fractional kinetics, Phys. Today Nov. (2002) 28–53.
[34] E. Sousa, Finite difference approximates for a fractional advection diffusion problem, J. Comput. Phys. 228 (2009) 4038–4054.
[35] L. Su, W. Wang, Z. Yang, Finite difference approximations for the fractional advection–diffusion equation, Phys. Lett. A 373 (2009) 4405–4408.
[36] 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.
[37] R.S. Varga, Matrix Iterative Analysis, second ed., Springer-Verlag, Berlin, Heideberg, 2000.
[38] G.M. Zaslavsky, D. Stevens, H. Weitzner, Self-similar transport in incomplete chaos, Phys. Rev. E 48 (1993) 1683–1694.
[39] Y. Zhang, A finite difference method for fractional partial differential equation, Appl. Math. Comput. 215 (2009) 524–529.