INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 9, Number 3, Pages 658–666

A FAST SECOND-ORDER FINITE DIFFERENCE METHOD FOR SPACE-FRACTIONAL DIFFUSION EQUATIONS

TREENA S. BASU AND HONG WANG

Abstract. Fractional diffusion equations provide an adequate and accurate description of transport processes that exhibit anomalous diffusion that cannot be modeled accurately by classical second-order diffusion equations. However, numerical discretizations of fractional diffusion equations yield full coefficient matrices, which require a computational operation of $O(N^3)$ per time step and a memory of $O(N^2)$ for a problem of size N. In this paper we develop a fast second-order finite difference method for space-fractional diffusion equations, which only requires memory of $O(N)$ and computational work of $O(N \log^2 N)$. Numerical experiments show the utility of the method.

Key Words. circulant and Toeplitz matrix, fast direct solver, fast finite difference methods, fractional diffusion equations

1. Introduction

Fractional diffusion equations model phenomena exhibiting anomalous diffusion that cannot be modeled accurately by classical 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 [4].

Compared to the classical second-order diffusion equations, the fractional diffusion equations have salient features which introduce new difficulties. From a computational point of view, fractional differential operators are nonlocal and so raise subtle stability issues on the corresponding numerical approximations. Numerical methods for space-fractional diffusion equations yield full coefficient matrices, which require a computational operation of $O(N^3)$ per time step and a memory of $O(N^2)$ for a problem of size N. 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 $O(N)$ (or $O(N \log N)$) operations per time step with $O(N)$ memory requirement.

Meerschaert and Tadjeran [7, 8] showed that a direct truncation of the Grünwald-Letnikov form of fractional derivative, even though discretized implicitly in time, leads to unstable discretizations. They proposed a shifted Grünwald discretization to approximate the fractional diffusion equation and proved the unconditional stability and convergence of the corresponding finite difference scheme. Numerical experiments showed that these methods generate satisfactory numerical results.

Received by the editors February 27, 2011 and, in revised form, August 1, 2011.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.

However, the shifted Grünwald discretization is only first-order accurate in space. Tadjeran et al [11] developed a Crank-Nicolson scheme which is second-order accurate in time. They recovered second-order spatial accuracy by a Richardson extrapolation. However, these methods still generate full coefficient matrices and so require storage of $O(N^2)$ and computational work of $O(N^3)$ per time step.

In this paper we develop a fast second-order finite difference method for two-sided space-fractional diffusion equations. The method has a significantly reduced memory requirement of $O(N)$ and computational work of $O(N \log^2 N)$ per time step. The method is an extension of the fast solution method developed in [13] and can also viewed as an extension of the superfast method [1, 2, 3], which was a direct solution method of $O(N \log^2 N)$ operations for a symmetric positive-definite Toeplitz system. The rest of the paper is organized as follows. In Section 2 we present the fractional diffusion equation and its Crank-Nicolson finite difference approximation. In Section 3 we develop the fast second-order finite difference method. In Section 4 we carry out numerical experiments to compare the performance of the fast finite difference method with the Crank-Nicolson finite difference method developed and analyzed in [11].

2. Fractional diffusion equations and its finite difference approximation

We consider the following initial-boundary value problem of a two-sided space-fractional diffusion equation with an anomalous diffusion of order $1 < \alpha < 2$

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

The left-sided $(+)$ and the right-sided $(-)$ fractional derivatives $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ and $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$ of equation (1) are defined in the Grünwald-Letnikov form

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

where $\lfloor x \rfloor$ represents the floor of x and the Grünwald weights $g_k^{(\alpha)}$ are defined as $g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k}$ where $\binom{\alpha}{k}$ represents fractional binomial coefficients. We note that the Grünwald weights $g_k^{(\alpha)}$ have the recursive relation

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

Moreover, for $1 < \alpha < 2$ the coefficients $g_k^{(\alpha)}$ satisfy the following properties:

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

We now focus on the development of a fast numerical method for problem (1). 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. The spatial and temporal partitions are defined as $x_i = x_L + ih$ 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 $f_i^m = f(x_i, t^m)$. We discretize the first-order time derivative in (1) by a standard first-order time difference quotient, but the for the discretization of the fractional spatial derivative we use the shifted Grünwald approximations

$$\begin{aligned} \frac{\partial^\alpha u(x_i, t^m)}{\partial_+ x^\alpha} &= \frac{1}{h^\alpha} \sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^m + a_1 h + O(h^2), \\ \frac{\partial^\alpha u(x_i, t^m)}{\partial_- x^\alpha} &= \frac{1}{h^\alpha} \sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^m + b_1 h + O(h^2) \end{aligned} \tag{5}$$

where a_1 and b_1 do not depend on the grid size h. The Crank-Nicolson finite difference scheme is formulated as follows

$$\begin{aligned} &\frac{u_i^{m+1} - u_i^m}{\Delta t} - \frac{1}{2}\Big(\frac{d_{+,i}^{m+1}}{h^\alpha} \sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^{m+1} - \frac{d_{+,i}^m}{h^\alpha} \sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^m\Big) \\ &\qquad -\frac{1}{2}\Big(\frac{d_{-,i}^{m+1}}{h^\alpha} \sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^{m+1} - \frac{d_{-,i}^m}{h^\alpha} \sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^m\Big) \\ &= \frac{1}{2}\big(f_i^{m+1} + f_i^m\big), \end{aligned} \tag{6}$$

which was proved to be unconditionally stable and convergent [11]. Numerical experiments show that this scheme generates very satisfactory numerical approximations. Let $u^m = \big[u_1^m, u_2^m, \ldots, u_{N-1}^m\big]^T$, $f^m = \big[f_1^m, f_2^m, \ldots, f_{N-1}^m\big]^T$, $A^m = \big[a_{i,j}^m\big]_{i,j=1}^{N-1}$, and I be the identity matrix of order $N-1$. Then the numerical scheme (6) can be expressed in the following matrix form

$$\left(I + \frac{\Delta t}{2h^\alpha} A^{m+1}\right) u^{m+1} = \left(I - \frac{\Delta t}{2h^\alpha} A^m\right) u^m + \frac{\Delta t}{2}(f^m + f^{m+1}). \tag{7}$$

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

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

It is clear that $a_{i,j}^{m+1} \le 0$ for all $i \ne j$ and that the coefficient matrix $I + (\Delta t/(2h^\alpha))A^{m+1}$ is a nonsingular, strictly diagonally dominant M-matrix. (8) implies that the Crank-Nicolson scheme has a full coefficient matrix, which has a memory requirement of $O(N^2)$ and and computational work of $O(N^3)$ per time step.

To develop a fast solution method, we carefully explore the structure of the coefficient matrices. We conclude from (8) that the stiffness matrices A^{m+1} and

A^m can be decomposed as follows

$$
\begin{aligned}
A^{m+1} &= -\mathrm{diag}\left(d_+^{m+1}\right) A_L - \mathrm{diag}\left(d_-^{m+1}\right) A_R, \\
A^m &= -\mathrm{diag}\left(d_+^{m}\right) A_L - \mathrm{diag}\left(d_-^{m}\right) A_R.
\end{aligned} \tag{9}
$$

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

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

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

Instead of storing two full matrices A^{m+1} and A^m which have $2(N-1)^2$ parameters we need only store the $5N-4$ parameters, $d_+^{m+1} = \left[d_{+,1}^{m+1}, d_{+,2}^{m+1}, \ldots, d_{+,N-1}^{m+1}\right]^T$, $d_-^{m+1} = \left[d_{-,1}^{m+1}, d_{-,2}^{m+1}, \ldots, d_{-,N-1}^{m+1}\right]^T$, $d_+^{m} = \left[d_{+,1}^{m}, d_{+,2}^{m}, \ldots, d_{+,N-1}^{m}\right]^T$, $d_-^{m} = \left[d_{-,1}^{m}, d_{-,2}^{m}, \ldots, d_{-,N-1}^{m}\right]^T$, and $\mathrm{g}^{(\alpha)} = \left[g_0^{(\alpha)}, g_1^{(\alpha)}, \ldots, g_{N-1}^{(\alpha)}\right]^T$. In particular, the fractional binomial coefficient vector $\mathrm{g}^{(\alpha)}$ depends only on the size of the spatial partition and the order of the anomalous diffusion but is independent of time or space. So it can be preprocessed and stored in advance.

3. A fast second-order finite difference method

In this section we extend the idea of our previous work [13] to develop a fast second-order finite difference method for problem (1). The development essentially consists of three steps: (i) We apply an operator-splitting technique to split the stiffness matrix A^{m+1} as the sum of a banded matrix A_k^{m+1} and a remaining matrix $A_O^{m+1} = A^{m+1} - A_k^{m+1}$. (ii) We then move A_O^{m+1} to the right-hand side of the numerical scheme and approximate the unknown solution by an extrapolation in time to retain a second-order accuracy. (iii) We carefully explore the structure of the coefficient matrices on the right-hand side of the numerical scheme to develop a fast algorithm to evaluate the right-hand side.

Let A_k^{m+1} contains the $2k+1$ diagonals of A^{m+1} and zero entries elsewhere, and $A_O^{m+1} = A^{m+1} - A_k^{m+1}$ contains the remaining nonzero entries of A^{m+1}. Then we split the stiffness matrix A^{m+1} as

$$A^{m+1} = A_k^{m+1} + A_O^{m+1}. \tag{10}$$

It was shown in [13] that if the bandwidth $k = \log N$, then

$$\frac{\left\|A^{m+1} - A_k^{m+1}\right\|_\infty}{\left\|A^{m+1}\right\|_\infty} = O(\log^{-\alpha} N) \to 0 \qquad \text{as } N \to \infty. \tag{11}$$

In other words, 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}. Moreover, such choice of the bandwidth k guarantees the computational work of inverting $I + (\Delta t/(2h^\alpha))A_k^{m+1}$ is $O(N \log^2 N)$.

We hence split the scheme (7) as follows

$$\begin{aligned}\left(I + \frac{\Delta t}{2h^\alpha} A_k^{m+1}\right) u^{m+1} &= \left(I - \frac{\Delta t}{2h^\alpha} A^m\right) u^m - \frac{\Delta t}{2h^\alpha} A_O^{m+1} u^{m+1} \\ &\quad + \frac{\Delta t}{2}\left(f^m + f^{m+1}\right).\end{aligned} \tag{12}$$

The issue that remains is how to approximate the u^{m+1} in the second term on the right-hand side of the scheme (12). To enhance the accuracy of the approximation, we evaluate the u^{m+1} on the right-hand side of (12) by a quadratic extrapolation in time which yields the following approximation $\hat{u}^{m+1}$ of u^{m+1}

$$\hat{u}^{m+1} \approx 3u^m - 3u^{m-1} + u^{m-2}, \quad m \geq 2 \tag{13}$$

Now it remains to approximate the solution u at the first and second time step i.e., u^1 and u^2. We use the fast implicit Euler method in [13] to compute u^1 and u^2. Even though these approximations have only second-order local truncation error, the numerical analysis theory of time-dependent problems tells us that this treatment still retains the same second-order global truncation error since these approximations are used only for the first two steps. Thus after substituting $\hat{u}^{m+1}$ of (13) for u^{m+1} in the second term on the right-hand side of the scheme (12) and evaluating u^1 and u^2 via the fast implicit Euler method, we obtain the following fast second-order finite difference scheme (F2FD)

$$\begin{aligned}\left(I + \frac{\Delta t}{2h^\alpha} A_k^{m+1}\right) u^{m+1} &= \left(I - \frac{\Delta t}{2h^\alpha} A^m - 3\frac{\Delta t}{2h^\alpha} A_O^{m+1}\right) u^m \\ &\quad + 3\frac{\Delta t}{2h^\alpha} A_O^{m+1} u^{m-1} - \frac{\Delta t}{2h^\alpha} A_O^{m+1} u^{m-2} \\ &\quad + \frac{\Delta t}{2}\left(f^m + f^{m+1}\right), \quad m \geq 2, \\ \left(I + \frac{\Delta t}{h^\alpha} A_k^2\right) u^2 &= \left(I - 2\frac{\Delta t}{h^\alpha} A_O^2\right) u^1 + \frac{\Delta t}{h^\alpha} A_O^2 u^0 + \Delta t f^2 \\ \left(I + \frac{\Delta t}{h^\alpha} A_k^1\right) u^1 &= \left(I - \frac{\Delta t}{h^\alpha} A_O^1\right) u^0 + \Delta t f^1.\end{aligned} \tag{14}$$

Finally, we turn to issue (iii), i.e., the fast evaluation of the right-hand side of the scheme. We recall from the matrix decomposition (9) that a fast evaluation of the right-hand side of the scheme (14) boils down to the fast evaluation of the matrix-vector multiplication of $A_L u$ and $A_R u$. Note that both A_L and A_R are

Toeplitz matrices, which can be embedded into $(2N-2)$-by-$(2N-2)$ circulant matrix $C_{2N-2,L}$ and $C_{2N-2,R}$, respectively. A Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant, while a circulant matrix is a matrix in which each row vector is rotated one element to the right relative to the preceding row vecor. 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.,

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

while 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 $i,j = 1,\ldots,n$.

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

It is known that a circulant matrix C_{2N-2} can be decomposed as follows [5, 6]

$$C_{2N-2} = F_{2N-2}^{-1}\, \mathrm{diag}\,(F_{\mathrm{2N-2}}c)\ \ F_{\mathrm{2N-2}} \tag{15}$$

where c is the first column vector of C_{2N-2} and F_{2N-2} is the $(2N-2)$-by-$(2N-2)$ discrete Fourier transform matrix. For any given $N-1$ dimensional vector u, $u_{2N-2} = \left[u, 0\right]^T$. (15) shows that $C_{2N-2}u_{2N-2}$ can be computed efficiently via the fast Fourier transform in $O((2N)\log(2N)) = O(N \log N)$ operations. Thus, the matrix-vector products $A_L u$ and $A_R u$ can be computed in $O(N \log N)$ operations and so are $A^{m+1}u$ and $A^m u$.

4. Numerical experiment

In this section we carry out numerical experiments to investigate the performance of the fast second-order finite difference method. We consider the fractional diffusion equation (1) with an anomalous diffusion of order $\alpha = 1.8$ and the left-sided and right-sided diffusion coefficients

$$d_+(x,t) = 1.32\Gamma(1.2)x^{1.8}, \quad d_-(x,t) = 1.32\Gamma(1.2)(2-x)^{1.8}. \tag{16}$$

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

$$
\begin{aligned}
f(x,t) &= -16e^{1-t}\Big[x^2(1-x)^2 + 2.64(x^2 + (1-x)^2) \\
&\qquad -13.2\big(x^3 + (1-x)^3\big) + 12\big(x^4 + (1-x)^4\big)\Big], \\
u_0(x) &= 16ex^2(1-x)^2.
\end{aligned} \tag{17}
$$

The true solution to the fractional diffusion equation (1) is given by [8]

$$
u(x,t) = 16e^{1-t}x^2(1-x)^2. \tag{18}
$$

Tadjeran et al. proved that Crank-Nicolson scheme (7) is unconditionally stable and is convergent with the accuracy of $O((\Delta t)^2 + h)$ [11]. They further used the Richardson extrapolation to recover the second-order spatial convergence, which involves finding the numerical solution u_h on a coarse grid h and then finding the numerical solution $u_{h/2}$ on a fine grid $h/2$ and then computing the extrapolated solution on the coarse spatial gird h by $u_h = 2u_{h/2} - u_h$. It was shown that the extrapolated solution has a second-order accuracy in space and time $O((\Delta t)^2 + h)^2)$.

$N = M$	$\lVert u_{CN}^M - u^M \rVert_{L^\infty}$	$\lVert u_{CN}^M - u^M \rVert_{L^2}$	CPU(seconds)
2^6	7.98656×10^{-3}	6.80845×10^{-3}	1.95312×10^{1}
2^7	3.79054×10^{-3}	3.23428×10^{-3}	1.50718×10^{2}
2^8	1.84519×10^{-3}	1.57456×10^{-3}	1.23123×10^{3}
2^9	9.10152×10^{-4}	7.76630×10^{-4}	9.86920×10^{3}
2^{10}	4.51975×10^{-4}	3.85650×10^{-4}	7.65370×10^{4}
	c=0.58, r = 1.03	c = 0.49, r = 1.03	
	$\lVert u_{F2FD}^M - u^M \rVert_{L^\infty}$	$\lVert u_{F2FD}^M - u^M \rVert_{L^2}$	
2^6	8.00233×10^{-3}	6.82157×10^{-3}	5.04687
2^7	3.79848×10^{-3}	3.24100×10^{-3}	2.28593×10^{1}
2^8	1.84957×10^{-3}	1.57831×10^{-3}	1.02171×10^{2}
2^9	9.13258×10^{-4}	7.79294×10^{-4}	4.13281×10^{2}
2^{10}	4.53797×10^{-4}	3.87218×10^{-4}	1.83257×10^{3}
	c=0.58, r = 1.03	c = 0.49, r = 1.03	

TABLE 1. Comparison of the fast second order finite difference (F2FD) method with the Crank-Nicolson method (CN) with Gaussian elimination

In the numerical experiment, we solve the problem by the Crank-Nicolson method (6) (CN), the fast second-order finite difference method (F2FD), the Crank-Nicolson method with extrapolation (CNE), and the fast second-order finite difference method with extrapolation (F2FDE) and denote their respective solutions by u_{CN}^m, u_{CNE}^m, u_{F2FD}^m, and u_{F2FDE}^m. Let u_h^m be the numerical solution u_{CN}^m, u_{CNE}^m, u_{F2FD}^m, or u_{F2FDE}^m at time step t^m and $u^m = u(x, t^m)$ be the true solution to problem (1). In Table 1 we choose $h = \Delta t$ and present the errors $\lVert u_{CN}^M - u^M \rVert_{L^\infty}$, $\lVert u_{F2FD}^M - u^M \rVert_{L^\infty}$, and $\lVert u_{F2FD}^M - u^M \rVert_{L^2}$, $\lVert u_{CN}^M - u^M \rVert_{L^2}$ for different mesh sizes. We then use a linear regression to fit the convergence rate r and the associated constant c in the L^2 and L^∞ norm

$$
\lVert u_h^M - u^M \rVert_{L^p} \le ch^r, \qquad p = 2, \infty. \tag{19}
$$

$N = M$	$\|u^M_{CNE} - u^M\|_{L^\infty}$	$\|u^M_{CNE} - u^M\|_{L^2}$	CPU(seconds)
2^6	4.827901×10^{-4}	3.46162×10^{-4}	1.57656×10^{2}
2^7	1.21258×10^{-4}	8.68653×10^{-5}	1.37660×10^{3}
2^8	3.03853×10^{-5}	2.17572×10^{-5}	9.91226×10^{3}
2^9	7.60523×10^{-6}	5.44443×10^{-6}	7.992601×10^{4}
	c=1.94, r = 1.99	c= 1.40, r = 1.99	
	$\|u^M_{F2FDE} - u^M\|_{L^\infty}$	$\|u^M_{F2FDE} - u^M\|_{L^2}$	
2^6	4.82081×10^{-4}	3.45777×10^{-4}	2.58125×10^{1}
2^7	1.20027×10^{-4}	8.60525×10^{-5}	1.14453×10^{2}
2^8	2.79488×10^{-5}	2.01151×10^{-5}	4.69015×10^{2}
2^9	6.88048×10^{-6}	4.95583×10^{-6}	2.47515×10^{3}
	c=2.44, r = 2.04	c =1.73, r = 2.04	

TABLE 2. Comparison of the fast second order finite difference with extrapolation (F2FDE) with the Crank-Nicolson method with extrapolation (CNE)

In Table 1 we also present the corresponding CPU times of the Crank-Nicolson method (6) (CN) and the fast second-order finite difference method (F2FD) from $N = M = 64$ to $N = M = 1024$. Each time we reduce the size of the spatial meshes by half, the total number of unknowns per time step is doubled. Consequently, the required memory increases 4 times and computational cost increases 8 times. If the time step size is reduced by half too, then the overall consumed CPU time for solving the finite difference method is expected to increase $2 \times 2^3 = 16$ times as predicted by the leading order behavior. We can in fact observe that the CPU time increases around 10 times or so each time we reduce the spatial mesh and time step size by half. The CPU time increases not exactly 16 times due to the overhead effect from the computations of other terms. In contrast, the computational work of the fast second-order finite difference method is $O(N \log^2 N)$. Each time we refine the size of the spatial meshes by half the required memory increases twice and when reducing the time step by half,the CPU time increases by 4 times. Our numerical experiments seem to coincide with this analysis.

Finally, in Table 2 we present the numerical results of the Crank-Nicolson method with Richardson extrapolation (CNE) side by side with fast second-order finite difference method with Richardson extrapolation (F2FDE), which indeed shows a second-order convergence rate in space and time.

In summary, the numerical experiments in this section show significant reduction of computational time, which coincides with the 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 1 indicate that the new scheme has the same stability constraint as the standard finite difference scheme which was proven to be unconditionally stable [7].

Acknowledgments

This work was supported in part by the National Science Foundation under Grant No. EAR-0934747.

References

1. G. Ammar and P. Gader, A variant of the Gohberg-Semencul formula involving circulant matrices, *SIAM Journal on Matrix Analysis and Applications*, 12 (1991), 534–540
2. G.S. Ammar and W.B. Gragg, The generalized Schur algorithm for the superfast solution of Toeplitz systems, Rational Approximation and its Applications in Mathematics and Physics, Lecture Notes in Mathematics, 1237 (1987), 315–330
3. G.S. Ammar and W.B. Gragg, Superfast solution of real positive definite Toeplitz systems, *SIAM Journal on Matrix Anal. Appl.*, 9 (1988), 61–76
4. D. Benson, S.W. Wheatcraft, and M.M. Meerschaert, The fractional-order governing equation of Lévy motion, *Water Resour. Res.* 36 (2000), 1413-1423.
5. P.J. Davis, *Circulant Matrices*, Wiley-Intersciences, New York, 1979.
6. R.M. Gray, *Toeplitz and Circulant Matrices: A Review*, Foundations and Trends in Communications and Information Theory, Vol 2, Issue 3, pp 155-239, 2006.
7. M.M. Meerschaert and C. Tadjeran, Finite difference approximations for fractional advection-dispersion flow equations, *J. Comput. Appl. Math.*, 172 (2004) 65-77.
8. M.M. Meerschaert and C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, *Appl. Numer. Math.*, 56 (2006) 80-90.
9. K.B. Oldham and J. Spanier, *The Fractional Calculus*, Academic Press, New York, 1974.
10. I. Podlubny, *Fractional Differential Equations*, Academic Press, New York, 1999.
11. C. Tadjeran, M.M. Meerschaert, and H.P. Scheffler, A second-order accurate numerical approximation for the fractional diffusion equation, *J. Comput. Phys.*, 213 (2006) 205-213.
12. R.S. Varga, *Matrix iterative Analysis*, Second Edition, Springer-Verlag, Berlin Heideberg, 2000.
13. H. Wang, K. Wang, and T. Sircar, A direct $O(Nlog^2N)$ finite difference method for fractional diffusion equations, *Journal of Computational Physics*, 229 (2010), 8095-8104

Department of Mathematics, University of South Carolina, Columbia South Carolina 29208, USA

E-mail: `sircar@mailbox.sc.edu` `hwang@math.sc.edu`