

A NEW HIGH-ORDER METHOD FOR THE TIME-FRACTIONAL DIFFUSION EQUATION WITH A SOURCE

HE YANG

ABSTRACT. In this paper, we propose a new high-order finite difference method to solve the time-fractional diffusion equation with a source. We first construct a finite difference approximation of the Caputo fractional derivative of order α ($0 < \alpha < 1$), and show that the convergence rate of our approximation is $(4 - \alpha)$. We then investigate the properties of the fractional differentiation matrix for our new approximations, and introduce an implicit finite difference method which employs such approximations for the time discretization of the fractional diffusion equation, coupled with a Fourier-type expansion in space. By taking advantage of the special structure of our fractional differentiation matrix, each of the linear systems resulted from our new high-order approximations for each mode of time-fractional diffusion equation can be solved in order $O(N^2)$. Numerical experiments about the performance of our method in evaluating fractional derivatives, and solving fractional ordinary differential equations and time-fractional diffusion equation are also presented, to demonstrate the efficiency of our method.

1. INTRODUCTION

Fractional partial differential equations (FPDEs) have been introduced extensively to replace the classical partial differential equations model in many applications. Roughly speaking, there are two types of fractional differential equations, namely, space-fractional differential equations and time-fractional differential equations. Some examples that belong to the former type include the random walk with Markovian waiting time and power law step length density, which leads to non-locality in space dimension. Instead, a continuous random walk with non-Markovian processes gives a time-fractional differential equation, which indicates a long memory property of the solution.

Along with the development of the mathematical modeling using FPDEs, there has been increasing need to design numerical methods to solve these equations. Many numerical methods, including finite difference methods [20, 15], finite volume methods [12, 14], discontinuous Galerkin methods [4, 22] and spectral methods [13, 23], have been proposed for the discretization of spatial fractional derivatives. As

2010 *Mathematics Subject Classification.* 34A08, 65L12, 97N50.

Key words and phrases. Fractional differential equations, Caputo derivative, time-fractional diffusion equation, high-order method.

Submitted Sep. 5, 2019. Revised Sep. 29, 2019.

for the numerical methods of time-fractional diffusion equation, the so-called $L1$ method [16] has been widely used for the discretization of time-fractional derivative in the equation due to its simplicity. However, the $L1$ method only has convergence order of $(2 - \alpha)$ when used to discretize fractional derivative of order α ($0 < \alpha < 1$). For α being close to 1, $L1$ method has only around first order accuracy. In order to obtain higher accuracy in time, the fractional linear multistep methods [24, 25] have been proposed for the time discretization to solve the time-fractional diffusion equation. In [25], the second order convergence rate in time have been shown. Another innovative method was constructed in [2] to extend the classical $L1$ method to $L2 - 1_\sigma$ method in time, which has convergence order of $(3 - \alpha)$. By smartly choosing the value of a parameter $\sigma \in (0, 1)$, and approximating the equation at time $t_{j+\sigma}$, the author of [2] obtained a second order in time method.

In this paper, we consider the same time-fractional diffusion equation with a source as in [16, 24, 25, 2]. The FPDE is simply defined by replacing the classical time derivative in the diffusion equation with fractional derivative of order α ($0 < \alpha < 1$). That is,

$${}_0D_t^\alpha u = u_{xx} + f(x, t), \quad x \in [0, 1], \quad t \in [0, T], \quad (1)$$

where ${}_0D_t^\alpha$ is the fractional derivative in the Caputo's sense. There has been numerous studies to investigate the time-fractional diffusion equation [1, 6, 7, 8, 9, 10, 11, 17, 18, 19]. Our focus is to develop a higher order discretization method for Caputo fractional derivative, which can be applied to solve (1) without increasing too much computational cost. Our proposed method can be regarded as an extension of $L1$ method and $L2 - 1_\sigma$ method. The way to discretize the Caputo fractional derivative of u is to use cubic interpolation which requires the value of u at four different discrete time. Because of this, we no longer have lower-triangular fractional differentiation matrix as in [16] and [2]. However, we can show that if we apply spatial Fourier transform to (1) and then use our proposed fractional differentiation matrix, we end up with linear systems which can be easily transferred to lower-Hessenberg linear system. Each of such linear system can be solved in $O(N^2)$ flops.

The remaining of the paper is as follows: in Section 2, we introduce the basic properties of fractional calculus and the classical $L1$ method, describe our proposed numerical methods for Caputo derivative, the properties of fractional differentiation matrix, and how to apply our high order discretization method to solve fractional differential equations as well as time-fractional diffusion equation. Some theoretical results including the convergence order and the special properties about the differentiation matrix are also proved. In Section 3, numerical experiments are presented to verify some of our theoretical findings in Section 2, to demonstrate the performance of our scheme.

2. A HIGH ORDER DISCRETIZATION OF FRACTIONAL DERIVATIVE

2.1. Background. Historically, there are many different ways to define fractional derivative. The most popular definitions among those are probably Riemann-Liouville and Caputo fractional derivatives. The starting point of these two concepts is the so-called Riemann-Liouville fractional integral operator. For $n - 1 <$

$\alpha < n$ (n is a positive integer) and any locally integrable function f , the Riemann-Liouville integral of f is defined to be

$$J^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-s)^{\alpha-1} f(s) ds. \quad (2)$$

Note that in the definition (2), the lower limit of the integral is chosen to be 0 for the convenience of the presentation, and it can be any real number in general. It is easy to see that if $\alpha = n$, the right hand side of the equation above is just equal to the n -fold integral of function f . Therefore, the definition given by (2) is a generalization of the integer cases. The Riemann-Liouville and Caputo fractional derivative of order α are simply as $D^n J^{n-\alpha}$ and $J^{n-\alpha} D^n$, respectively. Here, D^n is the standard derivative of integer order n .

In particular, when $\alpha \in (0, 1)$, the Caputo fractional derivative of f is defined as follows

$${}^C D_x^\alpha f(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-\eta)^{-\alpha} f'(\eta) d\eta. \quad (3)$$

There are advantages of using either of these two fractional derivatives. But in our paper, we choose Caputo fractional derivative because we can use the same initial condition as the standard diffusion equation. Alternatively, if we choose Riemann-Liouville fractional derivative in time, we have to use different initial condition involving fractional derivatives, which is unnatural for many real applications. For the rest of the paper, we only consider the Caputo fractional derivative, and thus we ignore the notation C in the definition (3).

One of the simplest methods to discretize ${}_0 D_t^\alpha f$ ($0 < \alpha < 1$) is $L1$ method [16]. Let $t_s = s\tau$ ($s = 0, 1, \dots$) be the time step with constant step size τ . The $L1$ method is given by approximating f' with finite difference within each subinterval $[t_s, t_{s+1}]$. That is,

$$\begin{aligned} {}_0 D_{t_{n+1}}^\alpha f &= \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^n \int_{t_i}^{t_{i+1}} (t_{n+1}-\eta)^{-\alpha} f'(\eta) d\eta \\ &\approx \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^n \frac{f(t_{i+1}) - f(t_i)}{\tau} \int_{t_i}^{t_{i+1}} (t_{n+1}-s)^{-\alpha} ds \\ &= \sum_{i=0}^n b_i (f(t_{i+1}) - f(t_i)), \end{aligned} \quad (4)$$

where $b_i = \tau^{-\alpha} [(n+1-i)^{1-\alpha} - (n-i)^{1-\alpha}] / \Gamma(2-\alpha)$. It is easy to see that the fractional differentiation matrix corresponding to such approximation is lower-triangular, if we ignore the first column which contains the coefficients from initial data of f . Therefore, it is convenient to apply $L1$ method to solve FPDEs. However, the error of the approximation above is only $O(\tau^{2-\alpha})$, which approaches $O(\tau)$ as $\alpha \rightarrow 1^-$. The well-known Grünwald-Letnikov (GL) approximation of ${}_0 D_{t_{n+1}}^\alpha f$ is given by

$${}_0 D_{t_{n+1}}^\alpha f \approx \frac{1}{h^\alpha} \left(f(t_{n+1}) - \sum_{i=1}^{n+1} C_i^\alpha f(t_{n+1-i}) \right), \quad (5)$$

where $C_i^\alpha = (-1)^{i-1} \frac{\Gamma(\alpha+1)}{\Gamma(i+1)\Gamma(\alpha-i+1)}$. It is proved in [21] that the GL approximation leads to an error of $O(\tau)$. For the rest of Section 2, we will introduce our higher order method based on $L1$ and $L2 - 1_\sigma$ methods.

2.2. Third Order Approximation for Caputo Fractional Derivatives. We now define our discretization of ${}_0D_T^\alpha u$, for $0 < \alpha < 1$, $T > 0$ and sufficiently smooth function $u = u(t)$. We first partition $[0, T]$ into equidistant grid: $0 = t_0 < t_1 < \dots < t_N = T$, with $t_s = s\tau$ for $s = 0, 1, \dots, N$ and $\tau = T/N$. We then approximate ${}_0D_T^\alpha u$ by rewriting it as

$$\begin{aligned} {}_0D_T^\alpha u &= \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t_1} (t_N - \eta)^{-\alpha} u'(\eta) d\eta \\ &+ \frac{1}{\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} u'(\eta) d\eta \\ &+ \frac{1}{\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-\alpha} u'(\eta) d\eta. \end{aligned} \quad (6)$$

For $s \geq 2$, we use $\Pi_{3,s}u$ to denote the cubic interpolation at grid points t_{s-2} , t_{s-1} , t_s and t_{s+1} . We define our discretization as

$$\begin{aligned} \delta_N^\alpha u &= \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t_1} (t_N - \eta)^{-\alpha} (\Pi_{3,2}u(\eta))' d\eta \\ &+ \frac{1}{\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} (\Pi_{3,s}u(\eta))' d\eta \\ &+ \frac{1}{\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-\alpha} (\Pi_{3,N-1}u(\eta))' d\eta. \end{aligned} \quad (7)$$

It is easy to show that the interpolation polynomial $\Pi_{3,s}u$ is given as

$$\begin{aligned} \Pi_{3,s}u(t) &= -u(t_{s-2}) \frac{(t-t_{s-1})(t-t_s)(t-t_{s+1})}{6\tau^3} + u(t_{s-1}) \frac{(t-t_{s-2})(t-t_s)(t-t_{s+1})}{2\tau^3} \\ &- u(t_s) \frac{(t-t_{s-2})(t-t_{s-1})(t-t_{s+1})}{2\tau^3} + u(t_{s+1}) \frac{(t-t_{s-2})(t-t_{s-1})(t-t_s)}{6\tau^3}. \end{aligned}$$

Therefore, we have

$$\begin{aligned} (\Pi_{3,s}u(t))' &= -u(t_{s-2}) \frac{3t^2 - 6t_s t + 3t_s^2 - \tau^2}{6\tau^3} \\ &+ u(t_{s-1}) \frac{3t^2 - 2(3t_s - \tau)t + 3t_s^2 - 2t_s\tau - 2\tau^2}{2\tau^3} \\ &- u(t_s) \frac{3t^2 - 2(3t_s - 2\tau)t + 3t_s^2 - 4t_s\tau - \tau^2}{2\tau^3} \\ &+ u(t_{s+1}) \frac{3t^2 - 2(3t_s - 3\tau)t + 3t_s^2 - 6t_s\tau + 2\tau^2}{6\tau^3}, \end{aligned}$$

which is a quadratic polynomial. We can further collect all of the coefficients of the resulting polynomial, and obtain $(\Pi_{3,s}u(t))' = \rho_{2,s}t^2 + \rho_{1,s}t + \rho_{0,s}$, where

$$\rho_{2,s} = \frac{1}{2\tau^3}(-u_{s-2} + 3u_{s-1} - 3u_s + u_{s+1}), \quad (8)$$

$$\rho_{1,s} = -\frac{t_s}{\tau^3}(-u_{s-2} + 3u_{s-1} - 3u_s + u_{s+1}) + \frac{1}{\tau^2}(u_{s-1} - 2u_s + u_{s+1}), \quad (9)$$

and

$$\begin{aligned}\rho_{0,s} &= \frac{t_s^2}{2\tau^3}(-u_{s-2} + 3u_{s-1} - 3u_s + u_{s+1}) - \frac{t_s}{\tau^2}(u_{s-1} - 2u_s + u_{s+1}) \\ &+ \frac{1}{6\tau}(u_{s-2} - 6u_{s-1} + 3u_s + 2u_{s+1}).\end{aligned}\quad (10)$$

Note that each coefficient of $(\Pi_{3,s}u(t))'$ in (8)-(10) is a linear combinations of finite difference approximations. To be more specific, $\frac{1}{2\tau^3}(-u_{s-2} + 3u_{s-1} - 3u_s + u_{s+1})$ is a first order approximation of $u'''(t_s)$, $\frac{1}{\tau^2}(u_{s-1} - 2u_s + u_{s+1})$ is a second order approximation of $u''(t_s)$ and $\frac{1}{6\tau}(u_{s-2} - 6u_{s-1} + 3u_s + 2u_{s+1})$ is a third order approximation of $u'(t_s)$.

Now we can calculate $\int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} (\Pi_{3,s}u(\eta))' d\eta$ using (8)-(10) and the following equations:

$$\begin{aligned}\int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} d\eta &= \frac{\tau^{1-\alpha}}{1-\alpha} ((N-s+1)^{1-\alpha} - (N-s)^{1-\alpha}), \\ \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} \eta d\eta &= -\frac{\tau^{2-\alpha}}{2-\alpha} ((N-s+1)^{2-\alpha} - (N-s)^{2-\alpha}) \\ &+ t_N \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} d\eta, \\ \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} \eta^2 d\eta &= \frac{\tau^{3-\alpha}}{3-\alpha} ((N-s+1)^{3-\alpha} - (N-s)^{3-\alpha}) \\ &- 2t_N \frac{\tau^{2-\alpha}}{2-\alpha} ((N-s+1)^{2-\alpha} - (N-s)^{2-\alpha}) \\ &+ t_N^2 \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} d\eta.\end{aligned}\quad (11)$$

Let $a_s := \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} d\eta$, $b_s := \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} \eta d\eta$ and $c_s := \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} \eta^2 d\eta$. Then, for $2 \leq s \leq N-1$, we have

$$\int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} u'(\eta) d\eta \approx \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} (\Pi_{3,s}u(\eta))' d\eta = \sum_{i=0}^3 d_s^{(i+1)} u_{s-2+i},\quad (12)$$

where

$$\begin{aligned}d_s^{(1)} &= -\frac{c_s}{2\tau^3} + \frac{t_s}{\tau^3} b_s + \left(-\frac{t_s^2}{2\tau^3} + \frac{1}{6\tau}\right) a_s, \\ d_s^{(2)} &= \frac{3c_s}{2\tau^3} + \left(-\frac{3t_s}{\tau^3} + \frac{1}{\tau^2}\right) b_s + \left(\frac{3t_s^2}{2\tau^3} - \frac{t_s}{\tau^2} - \frac{1}{\tau}\right) a_s, \\ d_s^{(3)} &= -\frac{3c_s}{2\tau^3} + \left(\frac{3t_s}{\tau^3} - \frac{2}{\tau^2}\right) b_s + \left(-\frac{3t_s^2}{2\tau^3} + \frac{2t_s}{\tau^2} + \frac{1}{2\tau}\right) a_s, \\ d_s^{(4)} &= \frac{c_s}{2\tau^3} + \left(-\frac{t_s}{\tau^3} + \frac{1}{\tau^2}\right) b_s + \left(\frac{t_s^2}{2\tau^3} - \frac{t_s}{\tau^2} + \frac{1}{3\tau}\right) a_s.\end{aligned}$$

Note that when $s = 1$ or N , we need to use a different interpolation polynomial to make sure that the points used are within the domain (see equation (7)). One can

derive that for general $s = 1, 2, \dots, N$,

$$\begin{aligned} d_s^{(1)} &= -\frac{c_s}{2\tau^3} + \frac{t_\star}{\tau^3}b_s + \left(-\frac{t_\star^2}{2\tau^3} + \frac{1}{6\tau}\right)a_s, \\ d_s^{(2)} &= \frac{3c_s}{2\tau^3} + \left(-\frac{3t_\star}{\tau^3} + \frac{1}{\tau^2}\right)b_s + \left(\frac{3t_\star^2}{2\tau^3} - \frac{t_\star}{\tau^2} - \frac{1}{\tau}\right)a_s, \\ d_s^{(3)} &= -\frac{3c_s}{2\tau^3} + \left(\frac{3t_\star}{\tau^3} - \frac{2}{\tau^2}\right)b_s + \left(-\frac{3t_\star^2}{2\tau^3} + \frac{2t_\star}{\tau^2} + \frac{1}{2\tau}\right)a_s, \\ d_s^{(4)} &= \frac{c_s}{2\tau^3} + \left(-\frac{t_\star}{\tau^3} + \frac{1}{\tau^2}\right)b_s + \left(\frac{t_\star^2}{2\tau^3} - \frac{t_\star}{\tau^2} + \frac{1}{3\tau}\right)a_s, \end{aligned}$$

where $\star = s$ if $2 \leq s \leq N-1$; $\star = 2$ if $s = 1$; and $\star = N-1$ if $s = N$. According to (7) and the equations above, we have the discrete operator defined as

$$\begin{aligned} \delta_N^\alpha u &= \left(\sum_{i=0}^3 d_1^{(i+1)} u_i + \sum_{s=2}^{N-1} \sum_{i=0}^3 d_s^{(i+1)} u_{s-2+i} + \sum_{i=0}^3 d_N^{(i+1)} u_{N-3+i} \right) / \Gamma(1-\alpha) \\ &= \frac{\sum_{s=0}^N e_s u_s}{\Gamma(1-\alpha)}, \end{aligned} \quad (13)$$

where

$$e_s = \begin{cases} \sum_{i=0}^s d_{s+2-i}^{(i+1)} + d_1^{(s)}, & \text{for } 0 \leq s \leq 3 \\ \sum_{i=0}^3 d_{s+2-i}^{(i+1)}, & \text{for } 4 \leq s \leq N-4 \\ \sum_{i=0}^{N-s} d_{s-1+i}^{(4-i)} + d_N^{(s+4-N)}, & \text{for } N-3 \leq s \leq N, \end{cases} \quad (14)$$

which concludes the formulation of our discrete Caputo operator $\delta_N^\alpha u$. Next, we show the convergence order of our finite difference approximation.

Theorem 2.1. For $\alpha \in (0, 1)$ and $u(t) \in C^4[0, T]$, there is

$$|{}_0D_T^\alpha u - \delta_N^\alpha u| = O(\tau^{4-\alpha}). \quad (15)$$

Proof. From equation (7), we have

$$\begin{aligned} &{}_0D_T^\alpha u - \delta_N^\alpha u \\ &= \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t_1} (t_N - \eta)^{-\alpha} (u - \Pi_{3,2}u(\eta))' d\eta \\ &+ \frac{1}{\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} (u - \Pi_{3,s}u(\eta))' d\eta \\ &+ \frac{1}{\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-\alpha} (u - \Pi_{3,N-1}u(\eta))' d\eta := \Lambda_1 + \Lambda_2 + \Lambda_3. \end{aligned} \quad (16)$$

We first estimate Λ_2 term. Recall that

$$u(\eta) - \Pi_{3,s}u(\eta) = \frac{u''''(\xi_s)}{24} (\eta - t_{s-2})(\eta - t_{s-1})(\eta - t_s)(\eta - t_{s+1}), \quad (17)$$

where $\eta \in [t_{s-2}, t_{s+1}]$, $\xi_s \in (t_{s-2}, t_{s+1})$, for $2 \leq s \leq N-1$. Therefore,

$$\begin{aligned}
|\Lambda_2| &= \left| \frac{1}{\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-\alpha} (u(\eta) - \Pi_{3,s}u(\eta))' d\eta \right| \\
&= \left| -\alpha \frac{1}{\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-1-\alpha} (u(\eta) - \Pi_{3,s}u(\eta)) d\eta \right| \\
&= \left| -\frac{\alpha}{24} \frac{1}{\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-1-\alpha} u''''(\xi_s) (\eta - t_{s-2})(\eta - t_{s-1}) \right. \\
&\quad \left. \times (\eta - t_s)(\eta - t_{s+1}) d\eta \right| \\
&\leq \frac{\alpha \max |u''''|}{24\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-1-\alpha} (\eta - t_{s-2})(\eta - t_{s-1})(t_s - \eta) \\
&\quad \times (t_{s+1} - \eta) d\eta \\
&\leq \frac{\alpha \tau^4 \max |u''''|}{6\Gamma(1-\alpha)} \sum_{s=2}^{N-1} \int_{t_{s-1}}^{t_s} (t_N - \eta)^{-1-\alpha} d\eta \\
&= \frac{\alpha \tau^4 \max |u''''|}{6\Gamma(1-\alpha)} \int_{t_1}^{t_{N-1}} (t_N - \eta)^{-1-\alpha} d\eta \\
&= \frac{\tau^4 \max |u''''|}{6\Gamma(1-\alpha)} \left(\frac{1}{\tau^\alpha} - \frac{1}{(N-1)\alpha\tau^\alpha} \right) < \frac{\tau^{4-\alpha} \max |u''''|}{6\Gamma(1-\alpha)}. \tag{18}
\end{aligned}$$

Here we have used integration by parts and the fact that $(u(\eta) - \Pi_{3,s}u(\eta))$ vanishes at t_{s-1} and t_s in the second equality above. Since $u(t) \in C^4[0, T]$, $\max |u''''|$ is bounded for $t \in [0, T]$, which leads to the conclusion that $\Lambda_2 = O(\tau^{4-\alpha})$. Since the integrand in term Λ_1 has no singularity in $[t_0, t_1]$, we can follow the previous procedures to show that $\Lambda_1 = O(\tau^{4-\alpha})$. As for the term Λ_3 , there is singularity for the integrand at $\eta = t_N$. Since for $\eta \in [t_{N-1}, t_N]$, there exists $\xi \in (t_{N-3}, t_N)$, such that $(t_N - \eta)^{-\alpha} (u(\eta) - \Pi_{3,N-1}u(\eta)) = -\frac{u''''(\xi)}{24} (\eta - t_{N-3})(\eta - t_{N-2})(\eta - t_{N-1})(t_N - \eta)^{1-\alpha}$, $(t_N - \eta)^{-\alpha} (u(\eta) - \Pi_{3,N-1}u(\eta)) = 0$ at $\eta = t_N$. Thus, we have

$$\begin{aligned}
|\Lambda_3| &= \left| \frac{1}{\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-\alpha} (u(\eta) - \Pi_{3,N-1}u(\eta))' d\eta \right| \\
&= \left| -\alpha \frac{1}{\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-1-\alpha} (u(\eta) - \Pi_{3,N-1}u(\eta)) d\eta \right| \\
&= \left| \frac{\alpha}{24} \frac{1}{\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-\alpha} u''''(\xi_s) (\eta - t_{N-3})(\eta - t_{N-2})(\eta - t_{N-1}) d\eta \right| \\
&\leq \left| \frac{\alpha \tau^3 \max |u''''|}{4\Gamma(1-\alpha)} \int_{t_{N-1}}^{t_N} (t_N - \eta)^{-\alpha} d\eta \right| = \left| \frac{\alpha \tau^3 \max |u''''|}{4\Gamma(2-\alpha)} \tau^{1-\alpha} \right| = O(\tau^{4-\alpha}). \tag{19}
\end{aligned}$$

Therefore, we have shown that each of Λ_i ($i = 1, 2, 3$) is of order $O(\tau^{4-\alpha})$, which implies ${}_0D_T^\alpha u - \delta_N^\alpha u = \Lambda_1 + \Lambda_2 + \Lambda_3 = O(\tau^{4-\alpha})$. \square

From the theorem above, we know that our proposed approximation is of order $O(\tau^{4-\alpha})$ for $\alpha \in (0, 1)$. As $\alpha \rightarrow 1^-$, the convergence order approaches 3.

2.3. Properties of Fractional Differentiation Matrix. From the previous section, we have shown that our approximation lead to high order accuracy when used as a forward operator. But if we would like to apply such finite difference approximation to the numerical solution of fractional differential equations, we have to construct the fractional differential matrix. Suppose we are solving a fractional differential equation involving ${}_0D_T^\alpha u$, where u is a sufficiently smooth function in t and $\alpha \in (0, 1)$. We first define $0 = t_0 < t_1 < \dots < t_N = T$ where $t_i = i\tau$ with $\tau = T/N$. Then for $1 \leq i \leq N$, the i^{th} row of the fractional differentiation matrix A represents the coefficients of u_0, u_1, \dots, u_N in our discretization of ${}_0D_{t_i}^\alpha u$, and A is a $N \times (N + 1)$ matrix. For $2 \leq j \leq i \leq N - 1$, let $d_{i,j}^{(1)}, d_{i,j}^{(2)}, d_{i,j}^{(3)}$ and $d_{i,j}^{(4)}$ be the coefficient of u_{j-2}, u_{j-1}, u_j and u_{j+1} in $\int_{t_{j-1}}^{t_j} (t_i - \eta)^{-\alpha} (\Pi_{3,j}u(\eta))' d\eta$, respectively. That is, the first subindex i in $d_{i,j}$ determines the term $(t_i - \eta)^{-\alpha}$ in the integrand, and the second index j indicates the bound of the integral to be $[t_{j-1}, t_j]$. Recall our definition of discretization of Caputo derivative in the previous section, we use $\Pi_{3,j}u(\eta)$, the cubic interpolation of u at t_{j-2}, t_{j-1}, t_j and t_{j+1} to approximate u within the interval $[t_{j-1}, t_j]$, when $2 \leq j \leq N - 1$. Therefore, one can show the following equalities about $d_{i,j}^{(k)}$ for $k = 1, 2, 3, 4$ and $2 \leq j \leq i \leq N - 1$:

$$\begin{aligned}
d_{i,j}^{(1)} &= -\frac{1}{6\tau^3} \int_{t_{j-1}}^{t_j} (t_i - \eta)^{-\alpha} [(\eta - t_j)(\eta - t_{j+1}) + (\eta - t_{j-1})(\eta - t_{j+1}) \\
&\quad + (\eta - t_{j-1})(\eta - t_j)], \\
d_{i,j}^{(2)} &= \frac{1}{2\tau^3} \int_{t_{j-1}}^{t_j} (t_i - \eta)^{-\alpha} [(\eta - t_j)(\eta - t_{j+1}) + (\eta - t_{j-2})(\eta - t_{j+1}) \\
&\quad + (\eta - t_{j-2})(\eta - t_j)], \\
d_{i,j}^{(3)} &= -\frac{1}{2\tau^3} \int_{t_{j-1}}^{t_j} (t_i - \eta)^{-\alpha} [(\eta - t_{j-1})(\eta - t_{j+1}) + (\eta - t_{j-2})(\eta - t_{j+1}) \\
&\quad + (\eta - t_{j-1})(\eta - t_{j-2})], \\
d_{i,j}^{(4)} &= \frac{1}{6\tau^3} \int_{t_{j-1}}^{t_j} (t_i - \eta)^{-\alpha} [(\eta - t_{j-1})(\eta - t_j) + (\eta - t_{j-2})(\eta - t_{j-1}) \\
&\quad + (\eta - t_{j-2})(\eta - t_j)].
\end{aligned} \tag{20}$$

In order to prove some main properties of the fractional differential matrix A , we need the following lemma about $d_{i,j}^{(k)}$.

Lemma 2.2. For $k = 1, 2, 3, 4$ and $2 \leq j \leq i \leq N - 2$, there is $d_{i+1,j+1}^{(k)} = d_{i,j}^{(k)}$.

Proof. This lemma can be proved by applying change of variable: $\eta' = \eta + \tau$ in the formulation of $d_{i,j}^{(k)}$ in (20). \square

Now we discuss the properties of the N -by- $(N + 1)$ fractional differentiation matrix A . For the convenience of the presentation, we use $A(i, j)$ to denote the (i, j) entry of matrix A , and represent the submatrix of A from i_1^{th} to i_2^{th} row and j_1^{th} to j_2^{th} column by $A(i_1 : i_2, j_1 : j_2)$. Theorem 2.3 gives an important result about the special structure of a $(N - 3)$ -by- $(N - 3)$ submatrix of A .

Theorem 2.3. $A(3:N-1, 5:N+1)$ is a lower-triangular Toeplitz matrix.

Proof. Let $B = A(3:N-1, 5:N+1)$, then $B(i, j) = A(i + 2, j + 4)$. To show B is a lower-triangular matrix, we only need to prove $A(i + 2, j + 4) = B(i, j) = 0$ when $1 \leq i < j \leq N - 3$. We first take $j = i + 1 \leq N - 3$, then $B(i, j) = A(i + 2, i + 5)$ is the coefficient of u_{i+4} in the approximation of ${}_0D_{t_{i+2}}^\alpha u$. But by our algorithm in Section 2.2, we only need a linear combination of u_0, u_1, \dots, u_{i+3} to approximate ${}_0D_{t_{i+2}}^\alpha u$. Therefore, $A(i + 2, j + 4) = 0$ for $j = i + 1, i + 2, \dots, N - 3$, and B is a lower-triangular matrix.

Next, we prove that B is also a Toeplitz matrix. We first consider the diagonal elements of B . By definition, $B(i, i) = A(i + 2, i + 4) = d_{i+2, i+2}^{(4)}/\Gamma(1 - \alpha)$ for $1 \leq i \leq N - 3$. From Lemma 2.2, $d_{i+2, i+2}^{(4)}/\Gamma(1 - \alpha) = d_{i+1, i+1}^{(k)}/\Gamma(1 - \alpha) = A(i + 1, i + 3) = B(i - 1, i - 1)$. Thus, the diagonal elements of B are constant. We then consider the sub-diagonal elements of B . Since

$$\begin{aligned} B(i + 1, i) &= A(i + 3, i + 4) = (d_{i+3, i+3}^{(3)} + d_{i+3, i+2}^{(4)})/\Gamma(1 - \alpha) \\ &= (d_{i+4, i+4}^{(3)} + d_{i+4, i+3}^{(4)})/\Gamma(1 - \alpha) \\ &= A(i + 4, i + 5) = B(i + 2, i + 1), \end{aligned}$$

for $i = 1, 2, \dots, N - 5$, the sub-diagonal elements are also constant. Similarly, we can show the results for other descending off-diagonal elements in the same manner. \square

With Theorem 2.3, we can save the computational cost for the construction of fractional differentiation matrix. That is, we only need to compute the last row of this submatrix to obtain all of its the entries, and the total number of entries that require our computation reduces from $O(N^2)$ to $O(N)$. Another important property of the fractional differentiation matrix A is that the entries with largest absolute values are mostly concentrated near the diagonal, and we can gives bounds for most of the lower-triangular elements of A .

Theorem 2.4. *Given $\alpha \in (0, 1)$ and $\tau > 0$, the following estimates are satisfied:*

$$\begin{aligned} |A(s, 1)| &\leq \frac{C_1\tau^{-\alpha}}{\Gamma(2 - \alpha)}[s^{1-\alpha} - (s - 2)^{1-\alpha}], \quad \text{for } s \geq 2 \\ |A(s, 2)| &\leq \frac{C_2\tau^{-\alpha}}{\Gamma(2 - \alpha)}[s^{1-\alpha} - (s - 2)^{1-\alpha}] \\ &\quad + \frac{C_3\tau^{-\alpha}}{\Gamma(2 - \alpha)}[(s - 2)^{1-\alpha} - (s - 3)^{1-\alpha}], \quad \text{for } s \geq 3 \\ |A(s, 3)| &\leq \frac{C_4\tau^{-\alpha}}{\Gamma(2 - \alpha)}[s^{1-\alpha} - (s - 2)^{1-\alpha}] + \frac{C_5\tau^{-\alpha}}{\Gamma(2 - \alpha)}[(s - 2)^{1-\alpha} - (s - 3)^{1-\alpha}] \\ &\quad + \frac{C_6\tau^{-\alpha}}{\Gamma(2 - \alpha)}[(s - 3)^{1-\alpha} - (s - 4)^{1-\alpha}], \quad \text{for } s \geq 4, \end{aligned} \tag{21}$$

where $C_i > 0$, $i = 1, 2, \dots, 6$ are constants independent of s , α and τ . Moreover, for general integer $4 \leq j \leq N - 3$ and $s \geq j + 1$, we have

$$\begin{aligned}
|A(s, j)| &\leq \frac{C_7 \tau^{-\alpha}}{\Gamma(2-\alpha)} [(s+3-j)^{1-\alpha} - (s+2-j)^{1-\alpha}] \\
&+ \frac{C_8 \tau^{-\alpha}}{\Gamma(2-\alpha)} [(s+2-j)^{1-\alpha} - (s+1-j)^{1-\alpha}] \\
&+ \frac{C_9 \tau^{-\alpha}}{\Gamma(2-\alpha)} [(s+1-j)^{1-\alpha} - (s-j)^{1-\alpha}] \\
&+ \frac{C_{10} \tau^{-\alpha}}{\Gamma(2-\alpha)} [(s-j)^{1-\alpha} - (s-j-1)^{1-\alpha}], \tag{22}
\end{aligned}$$

where C_7, C_8, C_9 and C_{10} are positive constants which only depend on j , and independent of s , α and τ .

Proof. We only prove inequality (22), and (21) can be shown in a similar manner. Recall that when $4 \leq j \leq N - 3$ and $s \geq j$, $A(s, j)$ is the coefficient of u_{j-1} in the approximation of ${}_0D_{t_s}^\alpha u$, and it is easy to show that $A(s, j) = (d_{s,j-2}^{(4)} + d_{s,j-1}^{(3)} + d_{s,j}^{(2)} + d_{s,j+1}^{(1)}) / \Gamma(1-\alpha)$, where $d_{i,j}^{(k)}$ for $k = 1, 2, 3, 4$ are defined in (20). From the last equation of (20), we have

$$\begin{aligned}
d_{s,j-2}^{(4)} &= \frac{1}{6\tau^3} \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} [3\eta^2 - 6(t_{j-2} - \tau)\eta + (3t_{j-2}^2 - 6t_{j-2}\tau + 2\tau^2)] d\eta \\
&= \frac{1}{2\tau^3} \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} \eta^2 d\eta + \frac{3-j}{\tau^2} \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} \eta d\eta \\
&+ \frac{3(j-2)^2 - 6(j-2) + 2}{6\tau} \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} d\eta \\
&\leq \left[\frac{(j-2)^2}{2\tau} + \frac{(3-j)(j-3)}{\tau} + \frac{3(j-2)^2 - 6(j-2) + 2}{6\tau} \right] \\
&\times \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} d\eta \\
&= \frac{1}{\tau} \left(j - \frac{8}{3} \right) \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} d\eta \\
&= \left(j - \frac{8}{3} \right) \frac{\tau^{-\alpha}}{1-\alpha} [(s+3-j)^{1-\alpha} - (s+2-j)^{1-\alpha}]. \tag{23}
\end{aligned}$$

Note that we have used the fact that $\frac{3-j}{\tau^2} < 0$ when $j \geq 4$ in the first inequality above. Also, from the second equality in (23), we have

$$\begin{aligned}
d_{s,j-2}^{(4)} &\geq \left[\frac{(j-3)^2}{2\tau} + \frac{(3-j)(j-2)}{\tau} + \frac{3(j-2)^2 - 6(j-2) + 2}{6\tau} \right] \\
&\times \int_{t_{j-3}}^{t_{j-2}} (t_s - \eta)^{-\alpha} d\eta \\
&= \left(\frac{17}{6} - j \right) \frac{\tau^{-\alpha}}{1-\alpha} [(s+3-j)^{1-\alpha} - (s+2-j)^{1-\alpha}]. \tag{24}
\end{aligned}$$

Combining (23) and (24), we have

$$|d_{s,j-2}^{(4)}| \leq \left(j - \frac{8}{3}\right) \frac{\tau^{-\alpha}}{1-\alpha} [(s+3-j)^{1-\alpha} - (s+2-j)^{1-\alpha}], \quad (25)$$

for $j \geq 4$. Following the same procedures, we can obtain similar inequalities about $d_{s,j-1}^{(3)}$, $d_{s,j}^{(2)}$ and $d_{s,j+1}^{(1)}$:

$$\begin{aligned} |d_{s,j-1}^{(3)}| &\leq \frac{(3j-4)\tau^{-\alpha}}{1-\alpha} [(s+2-j)^{1-\alpha} - (s+1-j)^{1-\alpha}], \\ |d_{s,j}^{(2)}| &\leq \frac{(3j-\frac{1}{2})\tau^{-\alpha}}{1-\alpha} [(s+1-j)^{1-\alpha} - (s-j)^{1-\alpha}], \\ |d_{s,j+1}^{(1)}| &\leq \frac{(j+\frac{2}{3})\tau^{-\alpha}}{1-\alpha} [(s-j)^{1-\alpha} - (s-j-1)^{1-\alpha}]. \end{aligned} \quad (26)$$

We can conclude the proof by summing up 25 and the inequalities in 26, and let $C_7 = j - \frac{8}{3}$, $C_8 = 3j - 4$, $C_9 = 3j - \frac{1}{2}$ and $C_{10} = j + \frac{2}{3}$. \square

Theorem 2.4 gives sharp estimates about lower-triangular part of matrix A . Specifically, all the entries below the $(1, 1)$ element in the first column are less than $\frac{C_1\tau^{-\alpha}}{\Gamma(2-\alpha)}[s^{1-\alpha} - (s-2)^{1-\alpha}]$. For fixed $\tau > 0$ and $\alpha \in (0, 1)$, that upper bound is monotonically decreasing as $s \geq 2$ increases. Since $[s^{1-\alpha} - (s-2)^{1-\alpha}]$ approaches zero as s goes to infinity, we can see that $A(s, 1)$ is a very small number for large enough s . This observation indicates that even though the time-fractional derivative has a long memory property, the early behavior of the function becomes less important to the fractional derivative at a large later time. Similar conclusion can be drawn from the other estimates in the theorem above.

2.4. A $(4-\alpha)$ -Order in Time Scheme for Time-Fractional Diffusion Equation. In this section, we present our numerical method for time-fractional diffusion equation (1), with initial condition $u(x, 0) = g(x)$ for $0 \leq x \leq 1$, and homogeneous boundary condition, i.e. $u(0, t) = u(1, t) = 0$ for $0 \leq t \leq T$. Due to the homogeneous boundary condition, we can write our exact solution using Fourier sine series:

$$u(x, t) = \sum_{k=1}^{\infty} \hat{u}_k(t) \sin(k\pi x). \quad (27)$$

It is important to mention that we can use Fourier series with basis $e^{j2\pi xk}$ if periodic boundary conditions are given, and Fourier cosine series with basis $\cos(k\pi x)$ given homogeneous Neumann boundary conditions. We plug the equation into (1) to obtain

$${}_0D_t^\alpha \hat{u}_k(t) = -(k\pi)^2 \hat{u}_k(t) + \hat{f}_k(t), \quad (28)$$

for $k = 1, 2, \dots$. So for each mode \hat{u}_k , we solve the fractional ordinary differential equation (28) using our fractional differentiation matrix constructed in Section 2.3. That is, let $t_j = j\tau$ for $j = 0, 1, \dots, N$ with $\tau = T/N$, and use \hat{U}_{0k} , \hat{U}_k and \hat{F}_k to denote the column vectors $[\hat{u}_k(t_0), \hat{u}_k(t_1), \dots, \hat{u}_k(t_N)]^T$, $[\hat{u}_k(t_1), \hat{u}_k(t_2), \dots, \hat{u}_k(t_N)]^T$ and $[\hat{f}_k(t_1), \hat{f}_k(t_2), \dots, \hat{f}_k(t_N)]^T$, respectively. Then we have $A\hat{U}_{0k} = -(k\pi)^2\hat{U}_k + \hat{F}_k$ for any positive integer k . Note that $\hat{u}_k(t_0)$ can be obtained from the sine transform of the initial condition, so we only need to solve \hat{U}_k . Let $A = [l|R]$, where l is the first column of the matrix A and R is an N -by- N matrix starting from

the second column of A . Therefore, for any mode k , we solve the following linear system

$$(R + (k\pi)^2 I)\hat{U}_k = \hat{F}_k - \hat{u}_k(t_0)l. \quad (29)$$

Here I represents an N -by- N identity matrix. Since we need to solve (29) for many Fourier modes, it is necessary to find a fast solver for each linear system. Although the matrix $(R + (k\pi)^2 I)$ is neither a lower-triangular nor a lower-Hessenberg matrix, it can be transformed to lower-Hessenberg matrix using simple row operations. Recall when we construct A , each row of A represents the coefficients that come from cubic interpolation. It is easy to show that $A(1, j) = A(2, j) = 0$ for $j \geq 5$, and $A(i, j) = 0$ when $3 \leq i \leq N - 2$ and $j \geq i + 3$. Therefore, we know $R(i, j) = 0$ when $j > i + 1$ and $i \geq 2$, and $R(1, j) = 0$ when $j \geq 3$. So we can use the $(2, 3)$ entry of matrix $(R + (k\pi)^2 I)$ to eliminate the $(1, 3)$ entry to obtain a lower-Hessenberg matrix. We then solve the linear system (29) using Hessenberg LU which requires only $O(N^2)$ flops [5]. In practice, we choose a finite number of modes (denoted by M) in the Fourier sine series. We will specify the choice of M for each numerical example about the time-fractional diffusion equation.

3. NUMERICAL EXPERIMENTS

In this section, we provide several numerical examples to demonstrate the performance of our proposed method. We implement all the simulations using MATLAB R2019a.

3.1. Accuracy Test of the Discrete Operator δ_N^α . Let $u(t) = t^{4+\alpha}$, $\alpha \in (0, 1)$. The exact Caputo fractional derivative of order α is

$${}_0D_t^\alpha t^{4+\alpha}|_{t=1} = \frac{\Gamma(5 + \alpha)}{24}. \quad (30)$$

We compute the absolute error between the exact value and numerical approximation using equation (13). Our results are presented in Table 1, from which we can observe the 3.5th and 3.1th convergence orders when $\alpha = 0.5$ and 0.9, respectively. Such observation is consistent with our theoretical results in Theorem 2.1: $(4 - \alpha)^{th}$ order of convergence. For $\alpha = 0.1$, our results indicate the convergence order of around 4, which is a little better than the theoretical convergence order $4 - \alpha$. The computational time is around 0.02 seconds for all the tests in Table 1. We further compare the performance of our proposed method with that of the classical $L1$ and Grünwald-Letnikov (GL) method in Table 2. We observe that the convergence order of the $L1$ method is about 1.5 and the GL method is of first order convergence. In addition, our method leads to error with much smaller magnitude. As far as the computational time is concerned, the $L1$ method takes less than 0.01 seconds, followed by the GL method which takes about 0.01 seconds. Our method takes slightly longer time, i.e., 0.02 seconds.

3.2. Fractional Ordinary Differential Equation. We now use our discrete operator δ_n^α to solve the following fractional ordinary differential equation:

$${}_0D_t^\alpha u = \frac{\Gamma(5 + \alpha)}{24} t^4, \quad u(0) = 0. \quad (31)$$

The exact solution of the initial value problem above is $t^{4+\alpha}$. We compute the numerical solutions up to $T = 1$ and estimate the convergence order (see Table 3). Among the results with three choices of α , numerical solutions with $\alpha = 0.1$ have

N	$\alpha = 0.1$		$\alpha = 0.5$		$\alpha = 0.9$	
	Error	Order	Error	Order	Error	Order
10	2.9048e-5		1.3085e-3		1.5340e-2	
20	1.7380e-6	4.0629	1.1807e-4	3.4702	1.8979e-3	3.0148
40	1.0407e-7	4.0618	1.0503e-5	3.4908	2.2763e-4	3.0596
80	6.2136e-9	4.0660	9.2894e-7	3.4991	2.6912e-5	3.0804
160	3.9328e-10	3.9818	8.2182e-8	3.4987	3.1585e-6	3.0909

TABLE 1. Error and convergence order of ${}_0D_t^\alpha t^{4+\alpha}$ at $t = 1$

N	Our method		$L1$ method		GL method	
	Error	Order	Error	Order	Error	Order
10	1.3085e-3		9.1701e-2		2.0487e-1	
20	1.1807e-4	3.4702	3.5244e-2	1.3796	1.0569e-1	0.9549
40	1.0503e-5	3.4908	1.3131e-2	1.4244	5.3678e-2	0.9774
80	9.2894e-7	3.4991	4.8029e-3	1.4510	2.7050e-2	0.9887
160	8.2182e-8	3.4987	1.7368e-3	1.4675	1.3578e-2	0.9944

TABLE 2. Error and convergence order of ${}_0D_t^\alpha t^{4+\alpha}$ at $t = 1$ using different methods. ($\alpha = 0.5$)

the smallest errors compared with other two cases, and numerical solutions with $\alpha = 0.9$ have the largest errors for any fixed N . For $\alpha = 0.5$ or $\alpha = 0.9$, the convergence order approaches $(4 - \alpha)$ as we refine the mesh. When $\alpha = 0.1$, the convergence order is between 3.7 and 3.8, which is a little less than $(4 - \alpha)$, which might be caused by the round-off errors. The overall convergence order of our proposed methods for solving fractional ordinary differential equation is consistent with our theoretical results. The computational time for all the tests in Table 3 is less than 0.1 seconds. In Table 4, we list the numerical results when we solve (31) using our proposed method, the $L1$ and the GL method. We can see the first order convergence for the GL method, the $(2 - \alpha)^{th}$ convergence order for the $L1$ method, and the $(4 - \alpha)^{th}$ order convergence for our method. Since we need to solve a linear system for all of the three methods, it takes longer computational time compared to the numerical tests in the previous section. In particular, the computational time for the $L1$ method and the GL method is about 0.01 seconds, and the computational time for our proposed method is around 0.08 seconds.

N	$\alpha = 0.1$		$\alpha = 0.5$		$\alpha = 0.9$	
	Error	Order	Error	Order	Error	Order
10	2.6259e-5		5.1768e-4		2.6380e-3	
20	2.0672e-6	3.6676	5.3197e-5	3.2826	3.7113e-4	2.8294
40	1.5415e-7	3.7453	5.0116e-6	3.4080	4.6212e-5	3.0056
80	1.1211e-8	3.7813	4.5688e-7	3.4554	5.5352e-6	3.0616
160	8.4387e-10	3.7318	4.1042e-8	3.4766	6.5320e-7	3.0830

TABLE 3. l_∞ error and convergence order of numerical solution to the initial value problem (31) for $t \in [0, 1]$

N	Our method		$L1$ method		GL method	
	Error	Order	Error	Order	Error	Order
10	5.1768e-4		5.0941e-2		1.1404e-1	
20	5.3197e-5	3.2826	1.9410e-2	1.3920	5.6647e-2	1.0095
40	5.0116e-6	3.4080	7.2249e-3	1.4258	2.8226e-2	1.0050
80	4.5688e-7	3.4554	2.6456e-3	1.4494	1.4088e-2	1.0026
160	4.1042e-8	3.4766	9.5804e-4	1.4654	7.0376e-3	1.0013

TABLE 4. l_∞ error and convergence order to the initial value problem (31) for $t \in [0, 1]$ using different methods. ($\alpha = 0.5$)

3.3. Time-Fractional Diffusion Equation with a Source. In this section, we test the performance of our scheme by solving the time-fractional diffusion equation with a source.

Example 1. We first consider the fractional diffusion equation whose exact solution is

$$u(x, t) = t^2 \sin(2\pi x). \quad (32)$$

In this case, the force term is given by

$$f(x, t) = \frac{2}{\Gamma(3 - \alpha)} t^{2-\alpha} \sin(2\pi x) + 4\pi^2 t^2 \sin(2\pi x). \quad (33)$$

We compute the errors at $T = 1$ using time step size $dt = 0.1$ and $M = 63$ modes, and observe from the top row of Figure 1 that the l_∞ errors are both equal to 5.5511×10^{-16} when $\alpha = 0.1$ or 0.5 , and the error is 2.2204×10^{-15} when $\alpha = 0.9$. Such observation implies that our numerical solution is accurate to within machine epsilon, and that is because we are using $(4 - \alpha)^{th}$ order method and the truncation error corresponding to this exact solution is equal to zero. Even with only $M = 7$ modes, our scheme still can lead to numerical solutions with error being within machine precision. In [16], the authors considered the same numerical example and applied their $(2 - \alpha)^{th}$ order time discretization method ($L1$ method) and spectral Galerkin/collocation method in space. Our high-order accurate numerical solutions outperform their results for this example. The error of the $L1$ and the GL method can be seen from the mid and bottom row of Figure 1. For both cases, our proposed method leads to more accurate results. The computational time of our method, $L1$ and GL method are 0.09, 0.03 and 0.02 seconds, respectively. The time history of the numerical solutions using our method is given in Figure 2. We can see that the behavior of the numerical solutions is consistent with that of the exact solutions. The computational time of our method at $t = 1, 2$ and 3 are about 0.09, 0.16 and 0.22 seconds, respectively.

Example 2. We then consider the example with exact solution being

$$u(x, t) = t^{4+\alpha} \sin(3\pi x). \quad (34)$$

In this case, the force term is defined as $\frac{\Gamma(5+\alpha)}{24} t^4 \sin(3\pi x) + (3\pi)^2 t^{4+\alpha} \sin(3\pi x)$. For all of the simulations results in Table 5, we fix the number of modes to be $M = 63$. We can see the convergence order to be around 4 when $\alpha = 0.1$ or 0.9 , which is even better than our theoretical prediction $(4 - \alpha)$. When $\alpha = 0.5$, the the order approaches 3.6. The errors of three choices of α are plotted in Figure 3. We can see that as we increase the value of α , the magnitude of the error increases. The

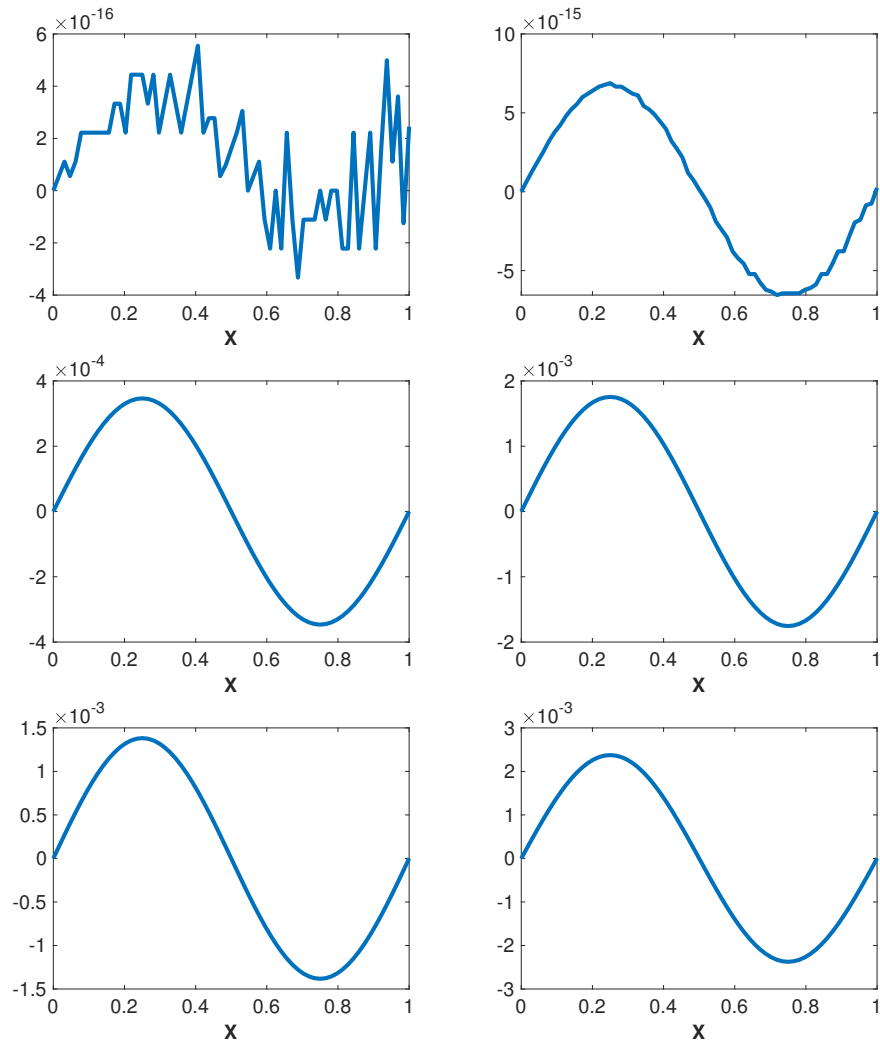


FIGURE 1. The error of our proposed method, $L1$ method and the GL method to the problem in example 1 at $T = 1$. The exact solution is $t^2 \sin(2\pi x)$. Here $dt = 0.1$ and $M = 63$ are used. Top row: the error of our proposed method. Mid row: the error of the $L1$ method. Bottom row: the error of the GL method. Left column: $\alpha = 0.5$. Right column: $\alpha = 0.9$.

computational time for each of the three cases, i.e., $\alpha = 0.1, 0.5$ and 0.9 , is about 0.08 seconds.

As a comparison, we also use the $L1$ method, which has $(2-\alpha)$ convergence order, for the time discretization (see equation (4)). We use spatial Fourier transform in both cases for fair comparison. We compute the l_∞ and l_2 errors when $\alpha = 0.5$, and present the results in Table 6. When we use our proposed method and choose a relatively coarse time step $dt = 0.1$, the l_2 error is 7.7591×10^{-5} , which is still more accurate than the $(2-\alpha)^{th}$ order method with 16 times more refined mesh size,

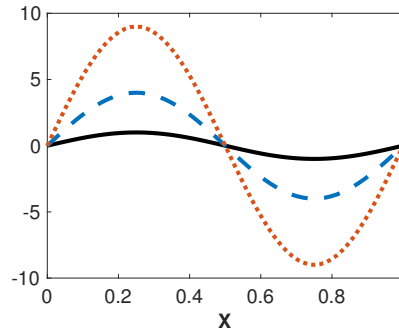


FIGURE 2. The numerical solution of the problem in example 1 using our proposed method at $T = 1, 2$ and 3 . The exact solution is $t^2 \sin(2\pi x)$. Here $dt = 0.1$ and $M = 63$ are used.

N	$\alpha = 0.1$		$\alpha = 0.5$		$\alpha = 0.9$	
	Error	Order	Error	Order	Error	Order
10	3.2084e-07		1.3716e-05		1.3980e-04	
20	1.9127e-08	4.0692	1.1982e-06	3.5169	1.4455e-05	3.2737
40	1.1402e-09	4.0683	1.0181e-07	3.5569	1.2450e-06	3.5374
80	6.7697e-11	4.0741	8.4417e-09	3.5922	7.7913e-08	3.9981
160	4.2604e-12	3.9900	6.8220e-10	3.6293	1.4017e-09	5.7966

TABLE 5. l_∞ error and convergence order of numerical solution to time-fractional diffusion equation with exact solution given by (34) at $t = 1$.

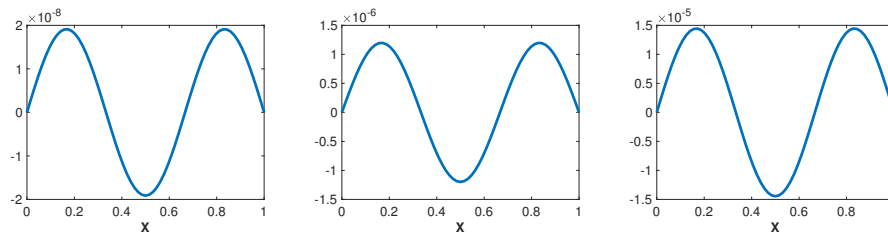


FIGURE 3. The error of our proposed method to time-fractional diffusion equation at $T = 1$. Exact solution: $t^{4+\alpha} \sin(3\pi x)$. Here $N = 20$ and $M = 63$ are used. From left to right: $\alpha = 0.1, 0.5$ and 0.9 .

i.e. the l_2 error of $(2 - \alpha)^{th}$ order method is 1.0856×10^{-4} when $dt = 6.25 \times 10^{-3}$. This illustrates that it is more efficient to use high-order method for this case. The computational time of our method when $N = 10, 20$ and 40 is about 0.1 seconds. When $N = 80$ or 160 , the computational time is about 0.2 seconds.

Example 3. For two examples above, we always assume that the exact solution has sine function, one of the basis element in the Fourier sine series. Now let's consider the case when the exact solution does not has sine function. That is, we

N	Our method				L1 method			
	l_∞ error	Order	l_2 error	Order	l_∞ error	Order	l_2 error	Order
10	1.3716e-05		7.7591e-05		1.0132e-03		5.7312e-03	
20	1.1982e-06	3.5169	6.7779e-06	3.5170	3.8939e-04	1.3796	2.2027e-03	1.3796
40	1.0181e-07	3.5569	5.7595e-07	3.5568	1.4508e-04	1.4244	8.2072e-04	1.4243
80	8.4417e-09	3.5922	4.7753e-08	3.5923	5.3070e-05	1.4509	3.0021e-04	1.4509
160	6.8220e-10	3.6293	3.8591e-09	3.6293	1.9192e-05	1.4674	1.0856e-04	1.4675

TABLE 6. l_∞ errors and convergence order of numerical solution to time-fractional diffusion equation with exact solution given by (34) at $t = 1$. Our method vs the L1 method ($(2 - \alpha)^{th}$ order in time). ($\alpha = 0.5$)

assume the exact solution is given by

$$u(x, t) = t^{4+\alpha}x(1-x). \quad (35)$$

The corresponding force term is given by $f(x, t) = \frac{\Gamma(5+\alpha)}{24}t^4x(1-x) + 2t^{4+\alpha}$. When we run the simulations for $\alpha = 0.5$, we can observe that the number of modes is very crucial for this example. For fixed $N = 10$, when we use $M = 63$, the l_∞ error is equal to 4.3047×10^{-5} and the l_2 error is 2.6864×10^{-4} . When $M = 127$ is used, the l_∞ error decreases to 1.0817×10^{-05} and the l_2 error goes down to 8.8610×10^{-5} . If we further double the number of modes to $M = 255$, both errors are 2.7050×10^{-6} and 2.2830×10^{-5} for l_∞ and l_2 errors, respectively. When we take $M = 1023$, these two errors further decrease to 8.2001×10^{-7} and 1.2188×10^{-5} . However, if we double the number of modes again, both errors increase due to round-off error. Therefore, in our simulations, we choose the mode large enough such that the error is minimized for each N . As a result, we take $M = 2^{10} - 1$ for $N = 20$; $M = 2^{11} - 1$ for $N = 40$; $M = 2^{13} - 1$ for $N = 80$ and $M = 2^{16} - 1$ for $N = 160$. For different values of α , the errors are plotted in Figure 4. The computational time for the examples in this figure is about 0.1 seconds. The distribution of the errors is symmetric around the midpoint of the spatial domain, and the largest error is at $x = 0.5$ for all of the three cases. Similar behavior that the magnitude of the largest error increases as α gets larger can be also observed. We then compute the l_2 and l_∞ errors and convergence orders in Table 7, and in this case, we obtain convergence order greater than $(4 - \alpha)$. Note that in this example, we increase the number of modes as we choose larger N . The computational time is about 0.1 seconds when $N = 20$. It increases to 0.2 seconds when $N = 40$, and it becomes 1.4 seconds when $N = 80$. Finally, when we take $N = 160$ and $M = 2^{16} - 1$, the computational time is about 36.7 seconds. In practice, we do not have to choose the number of modes as large as in this example, in order to get very accurate results.

4. CONCLUSION

In this paper, a high-order finite difference method is proposed to solve the time-fractional diffusion equation with a source. Both theoretical analysis and numerical results show that the convergence order of our finite difference method is $(4 - \alpha)$ when it is used to approximate the Caputo fractional derivative of order α with $\alpha \in (0, 1)$. When we solve the time-fractional diffusion equation with a source using Fourier-type expansion in space and various method (including the L1 and

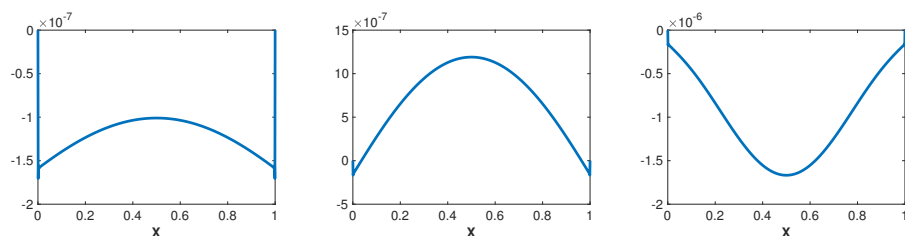


FIGURE 4. The error of our proposed method to time-fractional diffusion equation at $T = 1$. Exact solution: $t^{4+\alpha}x(1-x)$. From left to right: $\alpha = 0.1, 0.5$ and 0.9 .

N	l_∞ error	Order	l_2 error	Order
10	1.8860e-05		4.2857e-04	
20	1.1902e-06	3.9861	2.6295e-05	4.0267
40	4.7762e-08	4.6392	1.4321e-06	4.1986
80	2.6552e-09	4.1690	1.1323e-07	3.6608
160	4.1669e-11	5.9937	3.8377e-09	4.8829

TABLE 7. l_∞ error and convergence order of numerical solution to time-fractional diffusion equation with exact solution given by (35) at $t = 1$. ($\alpha = 0.5$)

Grünwald-Letnikov method) for the time discretization, we observe that our method leads to more accurate numerical solutions.

REFERENCES

- [1] E.A. Abdel-Rehim, A.M.A. El-Sayed and A.S. Hashem, Simulation of the approximate solutions of the time-fractional multi-term wave equations, *Computers & Mathematics with Applications*, vol. 73, no. 6, pp. 1134-1154, 2017.
- [2] A. A. Alikhanov, A new difference scheme for the time fractional diffusion equation, *Journal of Computational Physics*, vol. 280, pp. 424-438, 2015
- [3] A. Bueno-Orovio, D. Kay and K. Burrage, Fourier spectral methods for fractional-in-space reaction-diffusion equations, *BIT Numerical Mathematics*, vol. 54, no. 4, pp. 937-954, 2014.
- [4] W.H. Deng and J.S. Hesthaven, Local discontinuous Galerkin methods for fractional diffusion equations, *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 47, no. 6, pp. 1845-1864, 2013.
- [5] G. Golub and C. Van Loan, *Matrix Computations (3rd Ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [6] R. Gorenflo and E.A. Abdel-Rehim, Discrete models of time-fractional diffusion in a potential well, *Fractional Calculus and Applied Analysis*, vol. 8, no. 2, pp. 173-200, 2005.
- [7] R. Gorenflo and E.A. Abdel-Rehim, Convergence of the Grünwald-Letnikov scheme for time-fractional diffusion, *Journal of Computational and Applied Mathematics*, vol. 205, no. 2, pp. 871-881, 2007.
- [8] R. Gorenflo and E.A. Abdel-Rehim, From power laws to fractional diffusion: the direct way, arXiv preprint, arXiv:0801.0142, 2007.
- [9] R. Gorenflo, Y. Luchko and M. Yamamoto, Time-fractional diffusion equation in the fractional Sobolev spaces, *Fractional Calculus and Applied Analysis*, vol. 18, no. 3, pp. 799-820, 2015.
- [10] R. Gorenflo, F. Mainardi, D. Moretti, G. Pagnini and P. Paradisi, Discrete random walk models for space-time fractional diffusion, *Chemical physics*, vol. 284, no. 1-2, pp. 521-541, 2002.

- [11] R. Gorenflo, F. Mainardi, D. Moretti and P. Paradisi, Time fractional diffusion: a discrete random walk approach, *Nonlinear Dynamics*, vol. 29, no. 1-4, pp. 129-143, 2002.
- [12] H. Hejazi, T. Moroney and F. Liu, Stability and convergence of a finite volume method for the space fractional advection-dispersion equation, *Journal of Computational and Applied Mathematics*, vol. 255, no. 1, pp. 684-697, 2014.
- [13] J. Huang, N. Nie and Y. Tang, A second order finite difference-spectral method for space fractional diffusion equations, *Science China Mathematics*, vol. 57, no. 6, pp. 1303-1317, 2014.
- [14] J. Jia and H. Wang, A fast finite volume method for conservative space-fractional diffusion equations in convex domains, *Journal of Computational Physics*, vol. 310, pp. 63-84, 2016.
- [15] J. Jia and H. Wang, Fast finite difference methods for space-fractional diffusion equations with fractional derivative boundary conditions, *Journal of Computational Physics*, vol. 293, pp. 359-369, 2015.
- [16] Y. Lin and C. Xu, Finite Difference/spectral approximations for the time-fractional diffusion equation, *Journal of Computational Physics*, vol. 225, pp. 1533-1552, 2007.
- [17] F. Mainardi, Fractional relaxation-oscillation and fractional diffusion-wave phenomena, *Chaos, Solitons and Fractals*, vol. 7, no. 9, pp. 1461-1477, 1996.
- [18] F. Mainardi, The fundamental solutions for the fractional diffusion-wave equation, *Applied Mathematics Letters*, vol. 9, no. 6, pp. 23-28, 1996.
- [19] F. Mainardi, Y. Luchko and G. Pagnini, The fundamental solution of the space-time fractional diffusion equation, *Fractional Calculus and Applied Analysis*, vol. 4, no. 2, pp. 153-192, 2001.
- [20] M. Meerschaert and Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, *Applied Numerical Mathematics*, vol. 56, no. 1, pp. 80-90, 2006.
- [21] I. Podlubny, *Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solutions and Some of their Applications*, Academic Press, San Diego, CA, 1999.
- [22] Q. Xu and J.S. Hesthaven, Discontinuous Galerkin method for fractional convection-diffusion equations, *SIAM Journal on Numerical Analysis*, vol. 52, no. 1, pp. 405-423, 2014.
- [23] H. Yang, J. Guo and J.-H. Jung, Schwartz duality of the Dirac delta function for the Chebyshev collocation approximation to the fractional advection equation, *Applied Mathematics Letters*, vol. 64, pp. 205-212, 2017.
- [24] F. Zeng, C. Li, F. Liu and I. Turner, The use of finite difference/element approaches for solving the time-fractional subdiffusion equation, *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A2976-A3000, 2013.
- [25] F. Zeng, C. Li, F. Liu and I. Turner, *SIAM Journal on Scientific Computing*, vol. 37, no. 1, pp. A55-A78, 2015.

HE YANG

DEPARTMENT OF MATHEMATICS, AUGUSTA UNIVERSITY, AUGUSTA, GA 30912, USA

E-mail address: hyang1@augusta.edu