INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 17, Number 2, Pages 195–211

A FINITE DIFFERENCE SCHEME FOR CAPUTO-FABRIZIO FRACTIONAL DIFFERENTIAL EQUATIONS

XU GUO, YUTIAN LI, AND TIEYONG ZENG

Abstract. In this work, we consider a new fractional derivative with nonsingular kernel introduced by Caputo–Fabrizio (CF) and propose a finite difference method for computing the CF fractional derivatives. Based on an iterative technique, we can reduce the computational complexity from $O(J^2N)$ to $O(JN)$, and the corresponding storage will be cut down from $O(JN)$ to $O(N)$, which makes the computation much more efficient. Besides, by adopting piece-wise Lagrange polynomials of degrees 1, 2, and 3, we derive the second, third, and fourth order discretization formulas respectively. The error analysis and numerical experiments are carefully provided for the validation of the accuracy and efficiency of the presented method.

Key words. Caputo–Fabrizio derivative, fractional differential equations, higher order scheme.

1. Introduction

As an intensively developing area of the calculus during the past decades, fractional calculus has received much attention from both physicists and mathematicians, because it can describe the memory and hereditary properties of various materials and can faithfully capture the dynamics of physical processes in many research fields, including physics, engineering, chemistry, biology, and economics.

In the literature, there are two mostly used definitions for fractional differentiation, namely, the Riemann–Liouville (RL) and Caputo fractional operators. The RL definition plays an important role in the theory of fractional calculus and has many applications in pure mathematics, such as the definitions of new functions, see [32]. Besides, it has been found that the RL derivative is useful to characterize anomalous diffusion, Lévy flights and traps, and so forth [24]. On the other hand, practical considerations require proper definitions of fractional derivatives, which provide initial conditions with clear physical interpretation for the differential equations of fractional order. Therefore, even though the definition of Caputo derivative is more restrictive than the RL, it seems to be more welcome as well as crucial in practical applications.

Caputo's fractional derivative also has numerous applications in different areas of science [7, 9, 12, 21]. Let us start with the definition of the traditional Caputo fractional derivative [23, 32]. Given $b > 0$, $u \in H^1(0, b)$, and $0 < \alpha < 1$ with α being the fractional order, then the well-known Caputo fractional derivative of order α is defined by

$$(1) \qquad {}^{C}D^{\alpha}u(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} u'(s)\mathrm{d}s, \qquad t > 0.$$

Despite of the great success in applications, the singularity of the Caputo derivative in its kernel brings both theoretical and numerical difficulties. There are many

Received by the editors April 30, 2019.
2000 *Mathematics Subject Classification.* 26A33, 35R11, 65N06.

investigations have been done for numerically solving differential equations with the Caputo derivative, see [3, 14, 29, 30]. A recent progress is made by J. Zhang and his collaborators on the fast algorithms for the Caputo derivatives [22, 33], where the sum-of-exponential approximation is used to approximate the smooth kernel and then an iterative scheme is introduced. For numerical strategies to overcome the initial singularity, one can refer to the recent work [26] and for stability analysis, see the work [27, 28].

To have a smooth kernel and at the same time to keep the nonlocal property, Caputo and Fabrizio proposed a new kind fractional derivative in [10]. Indeed, replacing the kernel $(t-s)^{-\alpha}$ by the function $e^{-\alpha(t-s)/(1-\alpha)}$, and replacing $\Gamma(1-\alpha)$ by $(1-\alpha)$, we obtain the new Caputo–Fabrizio (CF) fractional derivative of order $0<\alpha<1$.

Definition 1 (Caputo–Fabrizio fractional derivative)**.** *Let $0<\alpha<1$, the Caputo–Fabrizio fractional derivative of order α of a function u is defined by*

$$ {}^{\mathrm{CF}}_{0}D^{\alpha}_{t}u(t) := \frac{1}{1-\alpha}\int_0^t e^{-\frac{\alpha}{1-\alpha}(t-s)}u'(s)\mathrm{d}s, \qquad t\geq 0. \tag{2}$$

Notice that in the original definition [10], there is a normalization factor $M(\alpha)$ in the CF derivative, which satisfies $M(0)=M(1)=1$. In a later paper [11], this factor $M(\alpha)$ is chosen to be the identity.

According to the new definition, it is clear that if u is a constant function, then ${}^{\mathrm{CF}}_{0}D^{\alpha}_{t}u=0$ as in the usual Caputo derivative. The main difference between the old and the new definitions is that, contrary to the old definition, the new kernel has no singularity at $t=s$. This suggests that the CF fractional model can describe the behavior of classical viscoelastic materials, thermal media, electromagnetic systems, etc. In fact, the original Caputo definition of fractional derivative appears to be particularly convenient for those mechanical phenomena, related with plasticity, fatigue, damage and with electromagnetic hysteresis. When these effects are not present, it seems more appropriate to use the new fractional derivative [10]. The CF derivative brings more and more attention in physics and engineering science, see [2, 4, 5, 6, 11, 17]. It is worthy to mention that there are some other kinds of nonlocal operators have been developed and used in variant time-nonlocal evolution models for describing anomalous diffusive dynamics; see for example [1, 13, 16, 35].

The objective of the present work is to develop a finite difference algorithm for the equations involving the CF fractional derivatives, which is crucial for many important applications. For example, consider the following fractional diffusion equation

$$ {}^{\mathrm{CF}}_{0}D^{\alpha}_{t}u = u_{xx}+f. \tag{3}$$

A popular method for solving such an equation is to use the piecewise linear interpolation of $u(x,t)$ on each time interval, and the order of accuracy of the method is 2. A similar method for the equation with traditional Caputo derivative has an accuracy of order $2-\alpha$, and the loss of order is due to the singularity appearing in the Caputo derivative. Moreover, the existing schemes for solving Eq. (3) require the storage of the solution at all previous time steps, so the computational complexity of these schemes is $O(J^2N)$, and the storage is $O(JN)$ on average, with J being the total number of time steps and N the number of grid points in space.

This leads to a serious bottleneck for long time simulations, especially when solving the time fractional partial differential equations. While for the usual diffusion equations, one only needs to store the solution at a fixed number of time steps and the computational complexity is linear with respect to J. Thus it is necessary to develop an efficient algorithm for solving the equations with the CF fractional derivatives.

The transitional Caputo derivative can be approximated by finite difference formulas of types $L1$, $L2$ or $L2$-1_σ, see [3, 25, 29, 33], see also [34] for a most recent progress in this direction. An important observation is that the kernel in the CF derivative is exponential, hence an iterative technique, which is first proposed in [22] for Caputo derivative, can be applied to the CF definition. Using this idea, we can reduce the computational complexity from $O(J^2N)$ to $O(JN)$, and the corresponding storage will be cut down from $O(JN)$ to $O(N)$, which improve the computation efficiency significantly.

Another aim of the present paper is to develop a higher order finite difference scheme for solving the CF fractional differential equations. To this end, we use high order interpolating polynomials for u, instead of the original piece-wise linear interpolation. In this work, based on a piece-wise Lagrange polynomial of degree 1, 2, and 3, we propose some second, third, and fourth order difference formulas for the CF derivative respectively. It is worthy to mention that Chen, Shi and Deng proposed a Lubich's type difference formulas for the Fokker–Planck equation with CF derivatives recently [15].

The rest part of the paper is organised as follows. In Section 2, we propose a new kind of finite difference algorithms for the CF fractional derivatives. Using the Lagrange polynomials, we derive a class of second, third, and fourth order discretization formulas. The error analysis is further provided in Section 3. In order to verify the accuracy and efficiency of the present method, we also carefully construct several numerical experiments in Section 4. Some concluding remarks are made in the last section.

2. Iterative algorithms for Caputo–Fabrizio derivatives

In this section, we shall develop a new kind of finite difference algorithms based on the iterative technique and with up to fourth order accuracy for the CF fractional derivatives, and prove that they are efficient in solving the time fractional differential equations under the CF definition.

2.1. An iterative algorithm with second order accuracy. We now introduce the iterative algorithm of the CF fractional derivative with second order accuracy.

For a positive integer J, let the temporal step size $\tau = T/J$, and denote $t^j = j\tau$, $\{u^j\}_{j=0}^J$ as the mesh function approximating $u(t)$ at time $\{t^j\}_{j=0}^J$. If we use the linear interpolation for $u(t)$, then the first order time derivative can be approximated with a second order difference scheme as

$$u'(s) \approx \frac{u^j - u^{j-1}}{\tau}$$

on the interval (t^{j-1}, t^j). Hence we can rewrite the CF definition as

$$ {}_0^{\mathrm{CF}}D_t^\alpha u(t^j) \approx \frac{1}{1-\alpha} \sum_{k=1}^{j} \left(\frac{u^k - u^{k-1}}{\tau} \int_{t^{k-1}}^{t^k} e^{-\frac{\alpha}{1-\alpha}(t^j - s)} \mathrm{d}s \right), \quad t^j \geq 0, $$

at some specific time point t^j, $j = 0, 1, 2, \cdots, J$. As one may notice, in order to compute the above derivative, all the history values of u, i.e. u^k for $k = 0, 1, \cdots, j-1$ are needed, which requires a great deal of memory and storage, especially when solving a PDE problem.

Since the kernel of the CF derivative is an exponential function, we can decompose the derivative into two parts

$$ {}_0^{\mathrm{CF}}D_t^\alpha u(t^j) = \frac{1}{1-\alpha} \left(\int_0^{t^{j-1}} e^{-\frac{\alpha}{1-\alpha}(t^j - s)} u'(s) \mathrm{d}s + \int_{t^{j-1}}^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^j - s)} u'(s) \mathrm{d}s \right). \tag{4} $$

The first part of the CF fractional derivative has the form

$$ \begin{aligned} \frac{1}{1-\alpha} \int_0^{t^{j-1}} e^{-\frac{\alpha}{1-\alpha}(t^j - s)} u'(s) \mathrm{d}s &= \frac{1}{1-\alpha} \int_0^{t^{j-1}} e^{-\frac{\alpha}{1-\alpha}(t^{j-1} + \tau - s)} u'(s) \mathrm{d}s \\ &= e^{-\frac{\alpha\tau}{1-\alpha}} \frac{1}{1-\alpha} \int_0^{t^{j-1}} e^{-\frac{\alpha}{1-\alpha}(t^{j-1} - s)} u'(s) \mathrm{d}s \\ &= e^{-\frac{\alpha\tau}{1-\alpha}} \, {}_0^{\mathrm{CF}}D_t^\alpha u(t^{j-1}). \end{aligned} \tag{5} $$

If we further denote

$$ C_1 := e^{-\frac{\alpha\tau}{1-\alpha}}, \qquad u_{hist} := {}_0^{\mathrm{CF}}D_t^\alpha u(t^{j-1}), $$

then the first part is equal to $C_1 u_{hist}$.

Motivated by this, we can perform the computation iteratively, and in each step, the history part is already computed and requires only the previous value u^{j-1}, which saves the computation resources significantly.

For the other term, we just need to evaluate the integral on (t^{j-1}, t^j). One possible approach is to use the Lagrangian polynomial [33]. We denote the linear interpolation over the time interval (t^{j-1}, t^j) with $1 \leq j \leq J$ by

$$ \Pi_{1,j} u(t) = u^{j-1} \frac{t^j - t}{\tau} + u^j \frac{t - t^{j-1}}{\tau}. $$

A direct calculation gives

$$ \left(\Pi_{1,j} u(t)\right)' = \frac{u^j - u^{j-1}}{\tau}, $$

and hence the second part of the CF derivative can be approximated as

$$ \begin{aligned} \frac{1}{1-\alpha} \int_{t^{j-1}}^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^j - s)} u'(s) \mathrm{d}s &\approx \frac{1}{1-\alpha} \left(\frac{u^j - u^{j-1}}{\tau} \right) \int_{t^{j-1}}^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^j - s)} \mathrm{d}s \\ &= \frac{1}{\alpha\tau} (1 - C_1)(u^j - u^{j-1}) \\ &:= C_2 (u^j - u^{j-1}), \end{aligned} \tag{6} $$

where C_2 is given by

$$ C_2 := \frac{1}{\alpha\tau}(1 - C_1) = \frac{1}{\alpha\tau}(1 - e^{-\frac{\alpha\tau}{1-\alpha}}), $$

Therefore, the CF derivative at time point t^j can be discretized as

$$ {}^{\mathrm{CF}}_{0}D_t^{\alpha} u(t^j) \approx C_1 u_{hist} + C_2\big(u^j - u^{j-1}\big) := {}^{\mathrm{CF}}\Delta_2^{\alpha} u(t^j), \tag{7} $$

which has second order accuracy.

2.2. Higher order scheme. The method mentioned in the last subsection greatly saves the computational cost, and improves the efficiency, whereas it is only second order accurate in time, which is the same as the direct method. In the following part, we shall consider to improve the accuracy to a higher order. It is evident from Eq. (4) that the CF derivative can be split into two parts: the history part and an integral, and the accuracy only depends on the integral part. We attempt to improve the computation accuracy of the integral part in what follows.

Denote the quadratic interpolation over the interval (t^{j-1}, t^j) as

$$ \Pi_{2,j} u(t) = u^{j-2}\frac{(t-t^{j-1})(t-t^j)}{2(\tau)^2} - u^{j-1}\frac{(t-t^{j-2})(t-t^j)}{2(\tau)^2} + u^j\frac{(t-t^{j-2})(t-t^{j-1})}{2(\tau)^2}, $$

then the first order derivative is calculated as

$$ \big(\Pi_{2,j} u(t)\big)' = \frac{u^{j-1}-u^{j-2}}{\tau}\cdot\frac{t^{j-1/2}-t}{\tau} + \frac{u^j-u^{j-1}}{\tau}\cdot\frac{t-t^{j-3/2}}{\tau}. $$

Now set

$$ C_3 := \frac{1}{\alpha\tau}\big(1-(1-\alpha)C_2\big). $$

By substituting the above expression of $\big(\Pi_{2,j} u(t)\big)'$ into the formula (4), the second part of the CF fractional derivative can be rewritten as

$$ \begin{aligned} &\frac{1}{1-\alpha}\int_{t^{j-1}}^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^j-s)} u'(s)\mathrm{d}s \\ &\approx \left(C_3 + \frac{\tau^2}{2}C_2\right)u^j - 2C_3 u^{j-1} + \left(C_3 - \frac{\tau^2}{2}C_2\right)u^{j-2}. \end{aligned} \tag{8} $$

Hence the CF derivative has the form

$$ \begin{aligned} {}^{\mathrm{CF}}_{0}D_t^{\alpha} u(t^j) \approx & C_1 u_{hist} + \left(C_3 + \frac{\tau^2}{2}C_2\right)u^j - 2C_3 u^{j-1} + \left(C_3 - \frac{\tau^2}{2}C_2\right)u^{j-2} \\ := & {}^{\mathrm{CF}}\Delta_3^{\alpha} u(t^j). \end{aligned} \tag{9} $$

Here we use the values of three points u^j, u^{j-1}, and u^{j-2} to evaluate the CF derivative and the local truncation error is $O(\tau^3)$.

It is worthy to mention that, since the kernel is smooth, we can apply an integration by parts to the integral term of the CF derivative. This is an alternative method to compute the CF derivative. The integral in the second part of the derivative can be decomposed as

$$ \begin{aligned} &\int_{t^{j-1}}^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^j-s)} u'(s)\mathrm{d}s \\ =& e^{-\frac{\alpha}{1-\alpha}(t^j-s)} u(s)\Big|_{t^{j-1}}^{t^j} - \int_{t^{j-1}}^{t^j} \frac{\alpha}{1-\alpha} e^{-\frac{\alpha}{1-\alpha}(t^j-s)} u(s)\mathrm{d}s, \end{aligned} \tag{10} $$

and we can approximate $u(t)$ by an interpolating polynomial and to obtain a quadrature rule for this integral. It can be shown after a careful calculation that these two approaches for evaluating the CF derivative result in exactly the same formula.

For instance, if we take the cubic interpolation over the time interval (t^{j-1}, t^j) with $1 \le j \le J$ as

$$
\begin{aligned}
\Pi_{3,j}u(t) \quad = \quad & u^{j-3}\frac{(t-t^{j-2})(t-t^{j-1})(t-t^j)}{-6\tau^3} + u^{j-2}\frac{(t-t^{j-3})(t-t^{j-1})(t-t^j)}{2\tau^3} \\
& -u^{j-1}\frac{(t-t^{j-3})(t-t^{j-2})(t-t^j)}{2\tau^3} + u^j\frac{(t-t^{j-3})(t-t^{j-2})(t-t^{j-1})}{6\tau^3}.
\end{aligned}
\tag{11}
$$

Then by using the method of integration by parts,

$$
\begin{aligned}
{}_0^{\mathrm{CF}}D_t^{\alpha}u(t^j) \quad \approx \quad & C_1 u_{hist} + u^j\Big[\frac{1}{\alpha\tau}\big(-C_4+\frac{3}{2}\big) + C_2\big(C_4^2 - C_4 + \frac{1}{3}\big)\Big] \\
& + u^{j-1}\Big[\frac{1}{\alpha\tau}\big(3C_4 - \frac{7}{2}\big) + C_2\big(-3C_4^2 + 2C_4 + \frac{1}{2}\big)\Big] \\
& + u^{j-2}\Big[\frac{1}{\alpha\tau}\big(-3C_4 + \frac{5}{2}\big) + C_2\big(3C_4^2 - C_4 - 1\big)\Big] \\
& + u^{j-3}\Big[\frac{1}{\alpha\tau}\big(C_4 - \frac{1}{2}\big) + C_2\big(-C_4^2 + \frac{1}{6}\big)\Big] \\
:= \quad & {}^{\mathrm{CF}}\Delta_4^{\alpha}u(t^j),
\end{aligned}
\tag{12}
$$

with C_4 defined as

$$C_4 := \frac{1-\alpha}{\alpha\tau}.$$

Since we adopt four points in formula (12), the truncation error hereby should be $O(\tau^4)$, and the error analysis will be provided in Section 3.

Note that there will be a significant cancellation error in evaluating the coefficients of u^j, u^{j-1}, u^{j-2}, and u^{j-3} when τ is small [22]. In this case, the coefficients can be computed by taking a few terms of a Taylor expansion for the exponential terms. For example, the coefficient of u^j can be expressed as

$$
\begin{aligned}
& \frac{1}{\alpha\tau}\left(-C_4+\frac{3}{2}\right) + C_2\left(C_4^2 - C_4 + \frac{1}{3}\right) \\
\approx & \frac{1}{1-\alpha}\left[1 - \frac{3}{8}\frac{1}{C_4} + \frac{19}{180}\frac{1}{C_4^2} - \frac{17}{720}\frac{1}{C_4^3} + \frac{22}{5040}\frac{1}{C_4^4} - \frac{83}{120960}\frac{1}{C_4^5} + \cdots\right].
\end{aligned}
\tag{13}
$$

Furthermore, in the fourth order method, we need to use the values of u at three previous time levels, i.e. u^{j-1}, u^{j-2}, and u^{j-3}, in the computation of u^j. To start the iteration, we have to compute the values u^1 and u^2, and the initial value u^0 is given as the initial condition. To this end, we shall first use a second order method to obtain the values of u^1 and u^2. Besides, in order to keep the overall computation accuracy and efficiency and to prevent the pollution on the later computation results, it is necessary to improve the accuracy of u^1 and u^2. Therefore, we perform the Richardson extrapolation on the second order method twice to get three values, with an error $O\left(\tau^4\right)$. Now let u_1^1, u_2^1 and u_4^1 denote the approximate values of u^1 by the second order method with steps being τ, $\tau/2$, and $\tau/4$ respectively, then

$$u^1 = \frac{32u_4^1 - 12u_2^1 + u_1^1}{21} + O\left(\tau^4\right).$$

Since the extrapolation is only used for two time steps, the stability of the extrapolation will not be an issue.

2.3. An alternative approach based on the integral equation. One reviewer pointed to us that the differential equation with CF derivative can be reformulated to an equivalent integral equations. For example, if we consider a fractional diffusion equation

$$ {}_0^{\mathrm{CF}}D_t^{\alpha} u(t) - \Delta u(t) = f(t), \qquad \text{with} \quad u(0) = u_0. \tag{14} $$

If one sets $v(t) = e^{\frac{\alpha}{1-\alpha}t}u(t)$ and $F(t) = (1-\alpha)e^{\frac{\alpha}{1-\alpha}t}f(t) + u_0$, after an integration by parts, the original problem is equivalent to

$$ v(t) - \frac{\alpha}{1-\alpha}\int_0^t v(s)ds - (1-\alpha)\Delta v(t) = F(t), \qquad \text{with} \quad v(0) = u_0. \tag{15} $$

The last equation is an integral equation with constant kernel and easy to solve by an efficient algorithm. Indeed, by approximating $v(t)$ using piece-wise Lagrange polynomials of degree 1, 2 or 3, one can get the corresponding Newton–Cotes formula (trapezoidal, Simpson's and Simpson's 3/8 rules, respectively) for the integral term, and the Newton–Cotes formula can be performed iteratively for efficiently solving the integral equation. For instance, if Simpson's rule is used, we then have a scheme with third order accuracy, for which the error analysis is very similar to that in Section 3 and hence omitted here. In the numerical simulations of the fractional diffusion equation problem considered in Section 4.2, we compared the two methods, IMFD and IMI, that is the iterative method for fractional derivative proposed in Section 2.2 and the iterative method for the integral equation form discussed above. The numerical results suggest that these two methods are numerically equivalent in terms of convergence rate and computational cost. We should also mention that there are other methods, like collocation methods, for solving the Volterra type integral equation (15); see [8].

3. Error analysis

3.1. Truncation error. The finite difference formula proposed in (12) is consistent with fourth order accuracy in time and the error analysis is given in the following theorem.

Theorem 2. *For any $\alpha \in (0,1)$, and $u(t) \in \mathcal{C}^4[0,T]$,*

$$ \left| {}_0^{\mathrm{CF}}D_t^{\alpha} u(t) - {}^{\mathrm{CF}}\Delta_4^{\alpha} u(t) \right| = |R(t)| \le C\tau^4, \quad \text{for each } t^{j-1} < t < t^j, \tag{16} $$

where the constant $C = 27M/(8(1-\alpha))$ with

$$ M = \max_{0\le t\le T} \left| u^{(4)}(t) \right|. $$

Proof. For $t^{j-1} < t < t^j$, the value $u(t)$ is approximated by $\Pi_{3,j}u(t)$ based on u^j, u^{j-1}, u^{j-2} and u^{j-3}, that is

$$ u(t) \approx \Pi_{3,j}u(t) = \sum_{k=0}^{3} u^{j-k}L_k(t), \qquad 0 \le t \le t^3, $$

where $L_k(t)$ is the Lagrange coefficient polynomials of degree three. Recall that

$$ |u(t) - \Pi_{3,j}u(t)| \le \frac{(t^j - t^{j-3})^4}{4!} \left| \max_{t^{j-3}\le t\le t^j} u^{(4)}(t) \right| \le \frac{(3\tau)^4}{4!}M. $$

Using a integration by part, we have

$$
\begin{aligned}
\left|R(t^3)\right| =& \left|\frac{1}{1-\alpha}\int_0^{t^3} e^{-\frac{\alpha}{1-\alpha}(t^3-s)}u'(s)\mathrm{d}s - \frac{1}{1-\alpha}\int_0^{t^3} e^{-\frac{\alpha}{1-\alpha}(t^3-s)}\big(\Pi_{3,3}u(s)\big)'\mathrm{d}s\right| \\
=& \frac{\alpha}{(1-\alpha)^2}\left|\int_0^{t^3} e^{-\frac{\alpha}{1-\alpha}(t^3-s)}\big[u(s)-\Pi_{3,3}u(s)\big]\mathrm{d}s\right|,
\end{aligned}
$$

which leads to

$$
\begin{aligned}
\left|R(t^j)\right| \le& \frac{\alpha}{(1-\alpha)^2}\int_0^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^3-s)}\left|u(s)-\Pi_{3,3}u(s)\right|\mathrm{d}s \\
\le& \frac{27M}{8(1-\alpha)}\tau^4.
\end{aligned}
$$

Here we have made use of the fact

$$
\int_0^{t^j} e^{-\frac{\alpha}{1-\alpha}(t^j-s)}\mathrm{d}s = \frac{1-\alpha}{\alpha}\left(1-e^{-\frac{\alpha}{1-\alpha}t^j}\right) \le \frac{1-\alpha}{\alpha}.
$$

□

The proofs of the second order and third order truncation errors are similar to that of the fourth order case.

3.2. Stability and convergence for the full concretization of a fractional diffusion equation. Consider a fractional diffusion problem

$$
\begin{cases}
{}^{\mathrm{CF}}_{\ 0}D_t^{\alpha}u = u_{xx}+f, & 0<x<1, \quad t>0, \\
u(0,t)=u(1,t)=0, & t\ge 0, \\
u(x,t)=u_0(x), & 0\le x\le 1.
\end{cases} \tag{17}
$$

Let h and τ denote the step sizes for space and time directions respectively. In the rectangle $\{(x,t)\,;\,0\le x\le 1, 0\le t\le T\}$ we introduce the mesh $\Omega_{h,\tau}=\Omega_h\times\Omega_\tau$ with $\Omega_h=\{x_n=nh\}_{n=1}^{N}$ and $\Omega_\tau=\{t^j=j\tau\}_{j=1}^{J}$. The general difference scheme for the problem (17) is given by

$$
\begin{cases}
{}^{\mathrm{CF}}\Delta^{\alpha}u_n^{j+1} = \delta^2 u_n^{j+1}+f_n^{j+1}, \quad 1\le n\le N-1, \quad j=0,\cdots,J-1, \\
u_0^j=0, \quad u_N^j=0, \qquad j=1,2,\cdots,J, \\
u_n^0=u_0(x_n), \qquad 0\le n\le N,
\end{cases} \tag{18}
$$

where the ${}^{\mathrm{CF}}\Delta^{\alpha}u_n^{j+1}$ is a difference operator approximating the CF derivative ${}^{\mathrm{CF}}_{\ 0}D_t^{\alpha}u(x_n,t^{j+1})$ and it takes the form

$$
{}^{\mathrm{CF}}\Delta^{\alpha}u_n^{j+1} = \sum_{k=0}^{j} g_k^{j+1}(u_n^{k+1}-u_n^k)
$$

with the coefficients g_k^{j+1}.

We shall recall the following two results in [3]. Although The original results in [3] is for the Caputo derivative, we can easily verify that the same arguments will lead to the same results for the CF derivative.

Theorem 3 ([3])**.** *If*

$$g_j^{j+1} > g_{j-1}^{j+1} > \cdots > g_0^{j+1} \geq C > 0, \tag{19}$$

$$\frac{g_j^{j+1}}{2g_j^{j+1} - g_{j-1}^{j+1}} \leq \sigma \leq 1, \tag{20}$$

where $j = 0, 1, \ldots, J-1$, $g_{-1}^1 = 0$, and C is a constant. Then the difference scheme (18) is unconditionally stable and its solution satisfies the following a priori estimate

$$\|u^{j+1}\|^2 \leq \|u^0\|^2 + \frac{1}{2\mu C} \max_{0 \leq j \leq J} \|f^j\|^2, \tag{21}$$

where μ is a constant such that $(-\delta^2 u, u) \geq \mu \|u\|^2$. (Recall that the central difference for second order derivative is negatively definite.)

Theorem 4 ([3])**.** *If the conditions of Theorem 3 are satisfied and difference scheme (18) has the approximation order $O(h^{r_1} + \tau^{r_2})$, where r_1 and r_2 are some known positive numbers, then the solution of the difference scheme (18) converges to the solution of the problem (17) with the order equal to the order of the approximation error $O(h^{r_1} + \tau^{r_2})$.*

Therefore, to show the stability and convergence of the difference scheme (18), we only need to prove that the coefficients g_k^j in the second, third and fourth difference formulas of the CF derivative satisfy conditions (19)-(20) with the choice $\sigma = 1$.

Lemma 5. *For the second order difference formula of the CF derivative introduced in (7)*

$${}^{\mathrm{CF}}\Delta_2^\alpha u_n^{j+1} = \sum_{k=0}^{j} g_k^{j+1} (u_n^{k+1} - u_n^k),$$

the coefficients $\{g_k^{j+1}\}$ satisfy the conditions (19)-(20) with $\sigma = 1$.

Proof. A straightforward calculation shows that

$$g_j^{j+1} = C_2 = \frac{1}{\alpha\tau}(1 - e^{-\frac{\alpha\tau}{1-\alpha}}),$$

and

$$g_k^{j+1} = C_1^{j-k} C_2 = e^{-(j-k)\frac{\alpha\tau}{1-\alpha}} \frac{1}{\alpha\tau}(1 - e^{-\frac{\alpha\tau}{1-\alpha}}),$$

where C_1 and C_2 are given in Section 2.1. It is readily seen that g_k^{j+1} is decreasing as k increases. Moreover,

$$g_0^{j+1} = e^{-\frac{\alpha j\tau}{1-\alpha}} \frac{1}{\alpha\tau}(1 - e^{-\frac{\alpha\tau}{1-\alpha}}) \geq e^{-\frac{\alpha T}{1-\alpha}} \frac{1}{\alpha\tau} \frac{\alpha\tau}{2(1-\alpha)} \geq e^{-\frac{\alpha T}{1-\alpha}} \frac{1}{2(1-\alpha)} := C,$$

where we have made use of the fact that $j\tau \leq J\tau = T$ and the inequality $1 - e^{-x} \geq x/2$ for $0 \leq x \leq 1$. Hence $\{g_k^j\}$ satisfy condition (19). Now

$$\frac{g_j^{j+1}}{2g_j^{j+1} - g_{j-1}^{j+1}} = \frac{C_2}{2C_2 - C_1 C_2} = \frac{1}{2 - C_1} = \frac{1}{2 - e^{-\frac{\alpha\tau}{1-\alpha}}} \leq 1,$$

which verifies the condition (20). This completes the proof of this lemma. $\square$

By a similar calculation, we can easily show that the third order and fourth order schemes ${}^{\mathrm{CF}}\Delta_3^{\alpha}$ and ${}^{\mathrm{CF}}\Delta_4^{\alpha}$ proposed in Section 2.2 are also satisfy the conditions (19)-(20), hence the stability and convergence of the corresponding difference scheme are proved by Theorem 4.

4. Numerical simulations

In this section, we present three numerical experiments to investigate and illustrate the accuracy and efficiency of the schemes. The L_∞ norm is used to measure the numerical errors, and all the simulations in this paper are carried out in the Matlab platform.

4.1. Numerical results for an ODE. Consider the following ordinary differential equation (ODE)

$$ {}^{\mathrm{CF}}_{0}D_t^{\alpha} u(t) = \lambda u(t) + f(t) \tag{22}$$

with the given exact solution $u(t) = \cos t$. We then calculate the corresponding initial value and source term given by

$$u_0(t) = 1, \qquad f(t) = -\frac{(1-\alpha)(C_1 - \cos t) + \alpha \sin t}{(1-\alpha)^2 + \alpha^2} - \lambda \cos t.$$

The ODE is solved at time $T = 1$ with time step size $\tau = T/J$, the parameter $\lambda = 1$, and $\alpha = 0.5, 0.8$, respectively. The numbers of time steps are taken as $J = 2^n$ with $n = 6, 7, 8, 9$.

We take the second order and fourth order formulas (7), (12) for the discretization of the given ODE. The maximum computational errors and convergence orders are listed in Table 1 and 2, which confirms the desired convergence rates perfectly.

TABLE 1. The second order discretization for Eq. (22).

$J = 2^n$	$\alpha = 0.5$		$\alpha = 0.8$	
n	error	rate	error	rate
6	1.14e-4		2.84e-4	
7	2.84e-5	2.00	7.11e-5	2.00
8	7.11e-6	2.00	1.78e-5	2.00
9	1.78e-6	2.00	4.44e-6	2.00

TABLE 2. The fourth order discretization for Eq. (22).

$J = 2^n$	$\alpha = 0.5$		$\alpha = 0.8$	
n	error	rate	error	rate
6	8.35e-9		2.09e-8	
7	5.36e-10	3.96	1.34e-9	3.96
8	3.39e-11	3.98	8.47e-11	3.98
9	2.27e-12	3.90	5.31e-12	4.00

4.2. Numerical results for a linear PDE. Next we solve a fractional diffusion problem

$$ {}_{0}^{\mathrm{CF}}D_t^{\alpha} u(x,t) = u_{xx}(x,t) + f(x,t) \tag{23} $$

in the domain $0 < t \leq 1$, with zero boundary conditions, and the initial condition is set to be

$$u(x,0) = u_0(x) = x^4(1-x)^4.$$

For a well-posed problem, we impose a generalized Dirichlet boundary condition for x in the complementary set of the solution domain, *i.e.*, $x \in \mathbb{R} \setminus (0,1)$. The exact solution of the fractional partial differential equation is given by

$$u(x,t) = x^4(1-x)^4 e^{-t},$$

and source term f can be solved as

$$f = -\frac{1}{2\alpha - 1}\left(e^{-t} - e^{-\frac{\alpha}{1-\alpha}t}\right) x^4(1-x)^4 - 4e^{-t}x^2(1-x)^2\left(3 - 14x + 14x^2\right).$$

Choose the number of time steps as $J = 2^n$ with $n = 5, 6, 7, 8$, and set the fractional order $\alpha = 0.3, 0.7$. We now use the formulas (7) and (12) for the discretization of the CF fractional derivative, and adopt second and fourth order central difference on the diffusion term respectively. For the comparison purpose, the iterative methods based on the integral equation form mentioned in Section 2.3 are also examined. The maximum errors and the convergence rates are listed in Tables 3 and 4, where IMFD stands for the iterative methods for fractional derivative (7) and (12), and IMI stands for the iterative methods for reduced integral form in Section 2.3. The results in Tables 3 and 4 show that these schemes have the optimal convergence in the order of $O(\tau^2 + h^2)$ and $O(\tau^4 + h^4)$, respectively.

Additionally, we also compare the computational time between the direct method and the iterative algorithms in this example. Let the fractional order $\alpha = 0.7$, and $J = 2^n$ with $n = 6, 7, 8, 9$. The results in Table 5 show that, although the direct method and the iterative methods have the same convergence, the iterative algorithm saves the CPU time significantly, especially when a large number of iterations are taken. As for the two iterative methods IMFD and IMI, the CPU times are about the same scale. This suggests that the two iterative methods have the same convergence rate and the same complexity, they are equivalent in principle.

TABLE 3. The second order discretization for Eq. (23).

$J = 2^n$	$\alpha = 0.3$				$\alpha = 0.7$			
	IMFD		IMI		IMFD		IMI	
n	error	rate	error	rate	error	rate	error	rate
5	2.06e-6		2.06e-6		2.12e-6		2.12e-6	
6	5.13e-7	2.01	5.13e-7	2.01	5.28e-7	2.00	5.30e-7	2.00
7	1.28e-7	2.00	1.28e-7	2.00	1.32e-7	2.00	1.32e-7	2.00
8	3.21e-8	2.00	3.20e-8	2.00	3.30e-8	2.00	3.31e-8	2.00

TABLE 4. The fourth order discretization for Eq. (23).

$J = 2^n$	$\alpha = 0.3$				$\alpha = 0.7$			
	IMFD		IMI		IMFD		IMI	
n	error	rate	error	rate	error	rate	error	rate
5	6.38e-8		6.38e-8		6.91e-8		6.90e-8	
6	4.19e-9	3.93	4.19e-9	3.93	4.54e-9	3.93	4.53e-9	3.93
7	2.68e-10	3.96	2.68e-10	3.96	2.91e-10	3.96	2.89e-10	3.97
8	1.71e-11	3.97	1.70e-11	3.98	1.84e-11	3.98	1.82e-11	3.99

TABLE 5. Comparison between the direct method and iterative algorithms.

$J = 2^n$	direct method		IMFD		IMI	
n	time	rate	time	rate	time	rate
6	1.09e-2		8.95e-3		8.85e-3	
7	2.52e-2	2.00	1.07e-2	2.00	1.01e-2	2.00
8	7.93e-2	2.00	3.23e-2	2.00	1.17e-2	2.00
9	3.17e-1	2.00	8.79e-2	2.00	3.73e-2	2.00

4.3. Numerical results for a fractional diffusion equation with a nonlinear source term. We then manage to solve a fractional diffusion equation with a nonlinear source term.

Consider the following problem

$$\begin{cases} {}_0^{\mathrm{CF}}D_t^{\alpha} u(x,t) = u_{xx} + |u|^p u + f(x,t), \\ u(x,0) = u_0(x,t), \end{cases} \tag{24}$$

where p, μ are constant numbers, and p is positive. This equation is very similar to a fractional Klein–Gordon equation, only with the value range of α changed from $(1,2)$ to $(0,1)$.

Now we take $0 < t \leq 1$, with vanishing boundary conditions, and the exact solution is set to be

$$u(x,t) = x(1-x)\sin t.$$

The corresponding initial value and source term f can be solved as

$$u(x,0) = 0,$$

and

$$\begin{aligned} f &= \frac{x(1-x)}{1-2\alpha+2\alpha^2}\left(-\alpha e^{-\frac{\alpha}{1-\alpha}t} + \alpha\cos t + (1-\alpha)\sin t\right) \\ &\quad +2\sin t + \mu x^2(1-x)^2|\sin t|\sin t. \end{aligned}$$

The proper Dirichlet boundary condition can be obtained in a similar way as in Example 2.

The numbers of time steps are taken as $J = 2^n$ with $n = 4,5,6,7$, and the fractional order α is set to be 0.8. Let $p = 1$, $\mu = 1$. We now use the second and fourth order central difference on the diffusion term respectively, and the CF fractional derivative is discretized by formulas (7) and (12). The nonlinear term in Eq. (24) is evaluated through a Newton's iteration method. The maximum errors and the convergence rates are listed in Table 6, which confirms the desired convergence rates perfectly for a nonlinear diffusion problem.

TABLE 6. The second and fourth order discretizations for Eq. (24).

$J = 2^n$	second order		fourth order	
n	error	rate	error	rate
4	9.69e-5		2.81e-8	
5	2.36e-5	2.04	1.80e-9	3.96
6	5.80e-6	2.02	1.14e-10	3.98
7	1.44e-6	2.01	7.27e-12	3.97

For the nonlinear problem, the proposed methods have a good performance for all time t if the initial condition is zero, but for non-zero initial condition, the methods produce a relative large error for the first few time steps. This phenomenon is shown in Figure 1, where the errors are plotted for all (x, t). In this simulation, we choose the second order scheme and the non-zero initial value example has the exact solution $u(x, t) = x(1 - x)\cos t$.

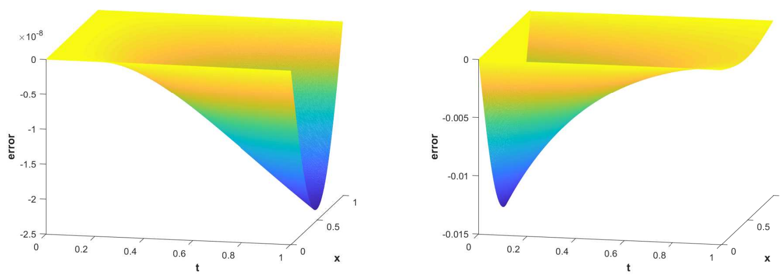

FIGURE 1. Errors of the second-order scheme for problem (24). Left: zero initial value; Right: non-zero initial value.

4.4. A time-space fractional diffusion equation. Finally we consider to solve a time-space fractional equation. Assume that the target problem is

$$\begin{cases} {}^{\mathrm{CF}}_{0}D^{\alpha}_{t}u(x,t) = {}^{RL}_{0}D^{\beta}_{x}u(x,t) + {}^{RL}_{x}D^{\beta}_{1}u(x,t) + f(x,t), \\ u(x,0) = u_0(x,t), \end{cases} \tag{25}$$

where the space fractional derivative in (25) is under the Riemann–Liouville (RL) definition [32],

$$\begin{aligned} {}^{RL}_{0}D^{\beta}_{x}u &:= \frac{1}{\Gamma(n-\beta)}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_0^x \frac{u(y)}{(x-y)^{\beta+1-n}}\mathrm{d}y, \\ {}^{RL}_{x}D^{\beta}_{1}u &:= \frac{(-1)^n}{\Gamma(n-\beta)}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_x^1 \frac{u(y)}{(y-x)^{\beta+1-n}}\mathrm{d}y, \end{aligned}$$

with $n = \lfloor\beta\rfloor + 1$ being the smallest integer exceeding β. In this example, we take $\beta \in (1, 2)$, hence $n = 2$. We remark that the RL derivative might be replaced by the tempered fractional derivatives, which has many applications in physics and other sciences. The fast algorithm of the tempered fractional derivatives developed in [19, 20] can be used with the method presented here to achieve an efficient difference method of order $O(h^4 + \tau^4)$.

Take the exact solution of the above problem to be

$$u(x,t) = x^5(1-x)^5 e^{-t},$$

and the generalized boundary condition can be fixed accordingly for this showing example.

Then in the domain $0 < x < 1$, $0 < t \le 1$, the initial condition can be calculated as

$$u(x,0) = x^5(1-x)^5,$$

and the source term has the form

$$\begin{aligned} f(x,t) \;\; = \;\; & -\frac{1}{2\alpha-1}\left(e^{-t} - e^{-\frac{\alpha}{1-\alpha}t}\right) x^5(1-x)^5 \\ & -e^{-t}\Big[\frac{\Gamma(6)}{\Gamma(6-\beta)}\big(x^{5-\beta} + (1-x)^{5-\beta}\big) - 5\frac{\Gamma(7)}{\Gamma(7-\beta)}\big(x^{6-\beta} + (1-x)^{6-\beta}\big) \\ & +10\frac{\Gamma(8)}{\Gamma(8-\beta)}\big(x^{7-\beta} + (1-x)^{7-\beta}\big) - 10\frac{\Gamma(9)}{\Gamma(9-\beta)}\big(x^{8-\beta} + (1-x)^{8-\beta}\big) \\ & +5\frac{\Gamma(10)}{\Gamma(10-\beta)}\big(x^{9-\beta} + (1-x)^{9-\beta}\big) - \frac{\Gamma(11)}{\Gamma(11-\beta)}\big(x^{10-\beta} + (1-x)^{10-\beta}\big)\Big]. \end{aligned}$$

We are going to use a higher order difference formula for the space fractional derivative terms of Eq. (25). In [18] a fourth order difference scheme was derived based on the weighted and shifted Lubich's formula, and it is shown that when the shifted parameters are chosen as $(1,-1,2,-2)$, all the eigenvalues of the difference matrix have negative real parts, which grantees the numerical stability. For the time fractional term in (25), we use the fourth order difference formula introduced in Section 2.

The numbers of time steps are taken as $J = 2^n$ with $n = 6,7,8,9$, and the fractional orders (α,β) are set to be $(0.3, 1.5)$ and $(0.7, 1.8)$ respectively. Table 7 shows the maximum errors at time $t = 1$ with $\tau = h$; the numerical results confirm the convergence with the global truncation error $O\left(h^4 + \tau^4\right)$.

TABLE 7. The fourth order discretization for Eq. (25).

$J = 2^n$	$\alpha = 0.3,\ \beta = 1.5$		$\alpha = 0.7,\ \beta = 1.8$	
n	error	rate	error	rate
6	1.45e-8		2.02e-8	
7	1.00e-9	3.86	1.27e-9	3.99
8	6.54e-11	3.93	8.17e-11	3.96
9	4.17e-12	3.97	5.14e-12	3.99

5. Conclusion

The Caputo definition of fractional derivative is commonly used as time fractional derivative, but the singularity in its kernel brings both theoretical and numerical difficulties. Caputo–Fabrizio (CF) defined a new fractional derivative with smooth kernel, which is drawing more and more attention in applications and analysis. In the present work, we provide an iterative algorithm for computing the CF fractional derivatives. Noting the kernel is an exponential function, a time marching scheme can be evaluated in an iterative manner, which reduces the storage and computation complexity significantly. To have high-order convergence rate, we also consider the use of Lagrange interpolating polynomials of degree one, two and three, which leads to finite difference formulas for the CF derivatives with accuracy of order $O\left(\tau^2\right)$, $O\left(\tau^3\right)$ and $O\left(\tau^4\right)$ respectively. Several numerical experiments are designed to examine the convergence rate and the computational complexity of the proposed finite difference formulas. These examples include ordinary differential equations, linear and nonlinear diffusion equations, and time-space fractional diffusion equations with the CF time derivatives. All the numerical simulations confirm the convergence rates of the higher-order schemes. An equivalent formulation of the equation involving CF derivative is a Volterra type integral equation, to which the iterative method can be applied in a similar way, and the two iterative methods are numerically equivalent in terms of convergence rate and computational cost. The numerical analysis of the fully discrete schemes for the nonlinear fractional diffusion equations and for the time-space fractional diffusion equations are not included in the current paper and we shall address this issue in a future work.

Acknowledgments

All authors would like to thank the two reviewers for their thoughtful comments and efforts towards improving our manuscript. This work was supported in part by the President's Fund from the Chinese University of Hong Kong, Shenzhen and by the National Natural Science Foundation of China [grant number 11801480, 11901354].

References

[1] M. S. Abdo and S. K. Panchal, Fractional integro-differential equations involving ψ-Hilfer fractional derivative, Adv. Appl. Math. Mech., 11 (2019), pp. 338-359.

[2] F. Ali, M. Saqib, I. Khan, and N. A. Sheikh, Application of Caputo-Fabrizio derivatives to MHD free convection flow of generalized Walters'-B fluid model, The European Physical Journal Plus, 131 (2016), 377.

[3] A. A. Alikhanov, A new difference scheme for the time fractional diffusion equation, J. Comput. Phys., 280 (2015), pp. 424-438.

[4] T. M. Atanacković, S. Pilipović, and D. Zorica. Properties of the Caputo-Fabrizio fractional derivative and its distributional settings, Fractional Calculus and Applied Analysis, 21 (2018), pp. 29-44.

[5] A. Atangana On the new fractional derivative and application to nonlinear Fishers reaction-diffusion equation, Appl. Math. Comput., 273 (2016), pp. 948-956.

[6] A. Atangana and D. Baleanu, Caputo-Fabrizio derivative applied to groundwater flow within confined aquifer, Journal of Engineering Mechanics, 143 (2017), D4016005.

[7] R. L. Bagley, and P. J. Torvik, On the fractional calculus model of viscoelastic behavior, Journal of Rheology, 30 (1986), pp. 133-155.

[8] H. Brunner, Collocation Methods for Volterra Integral and Related Functional Differential Equations, Cambridge University Press, 2004

[9] M. Caputo, Linear models of dissipation whose Q is almost frequency independent: II, Geophysical Journal of the Royal Astronomical Society, 13 (1967), pp. 529-539.

[10] M. Caputo and M. Fabrizio, A new definition of fractional derivative without singular kernel, Progress in Fractional Differentiation and Applications, 1 (2015), pp. 1-13.

[11] M. Caputo and M. Fabrizio, Applications of new time and spatial fractional derivatives with exponential kernels, Progress in Fractional Differentiation and Applications, 2 (2016), pp. 1-11.

[12] M. Caputo and F. Mainardi, A new dissipation model based on memory mechanism, Pure and Applied Geophysics, 91 (1971), pp. 134-147.

[13] A. V. Chechkin, R. Gorenflo, and I. M. Sokolov, Retarding subdiffusion and accelerating superdiffusion governed by distributed-order fractional diffusion equations. Phys. Rev. E, 66 (2002), 046129.

[14] F. Chen, Q. Xu, and J. S. Hesthaven, A multi-domain spectral method for time-fractional differential equations J. Comput. Phys., 293 (2015), pp. 157-172.

[15] M. Chen, J. Shi, and W. Deng, High order algorithms for Fokker-Planck equation with Caputo-Fabrizio fractional derivative, prepint (2018), arXiv preprint arXiv:1809.03263.

[16] Q. Du, J. Yang, and Z. Zhou, Analysis of a nonlocal-in-time parabolic equation, Discrete Contin. Dyn. Syst. Ser. B, 22 (2017), pp. 339-368.

[17] J. F. Gómez-Aguilar, M. G. López-López, V. M. Alvarado-Martínez J. Reyes-Reyes, and M. Adam-Medina, Modeling diffusive transport with a fractional derivative without singular kernel, Phys. A, 447 (2016), pp. 467-481.

[18] X. Guo, Y. Li, and H. Wang, A fourth-order scheme for space fractional diffusion equations, J. Comput. Phys., 737 (2018), pp. 410-424.

[19] X. Guo, Y. Li, and H. Wang, A fast finite difference method for tempered fractional diffusion equations, Commun. Comput. Phys., 24 (2018), pp. 531-556.

[20] X. Guo, Y. Li, and H. Wang, A high order finite difference method for tempered fractional diffusion equations with applications to the CGMY model, SIAM J. Sci. Comput., 40 (2018), pp. A3322-A3343.

[21] R. Hilfer, Applications of Fractional Calculus in Physics, World Scientific, Singapore, 2000.

[22] S. Jiang, J. Zhang, Q. Zhang and Z. Zhang, Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations, Commun. Comput. Phys., 21 (2017), pp. 650-678.

[23] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, Theory and Applications of Fractional Differential Equations, Elsevier, Amsterdam, 2006.

[24] C. Li, D. Qian, and Y. Chen, On Riemann–Liouville and Caputo derivatives, Discrete Dynamics in Nature and Society, 2011 (2011).

[25] D. Li and J. Zhang, Efficient implementation to numerically solve the nonlinear time fractional parabolic problems on unbounded spatial domain, J. Comput. Phys., 322 (2016), pp. 415-428.

[26] H. Liao, D. Li, J. Zhang, Sharp error estimate of the nonuniform L1 formula for reaction-subdiffusion equations, SIAM J. Numer. Anal., 56 (2018), pp. 1112-1133.

[27] H. Liao, W. McLean, and J. Zhang, A discrete Grönwall inequality with applications to numerical schemes for subdiffusion problems, SIAM J. Numer. Anal., to appear. https://epubs.siam.org/doi/abs/10.1137/16M1175742

[28] S.M. Lin, M. Azaiez and C.J. Xu, A fractional Stokes equation and its spectral approximations, Inter. J. Numer. Analysis and Modelling., 15 (2018), pp. 170-192.

[29] Y. Lin and C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, J. Comput. Phys., 225 (2007), pp. 1533-1552.

[30] Ch. Lubich, Discretized fractional calculus. SIAM J. Math. Anal. 17, (1986), 704C719.

[31] J. Losada, and J. J. Nieto, Properties of a new fractional derivative without singular kernel, Progress in Fractional Differentiation and Applications, 1 (2015), pp. 87-92.

[32] I. Podlubny, Fractional Differential Equations, Academic Press, San Diego, 1999.

[33] Y. Yan, Z. Sun, and J. Zhang, Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations: a second-order scheme, Communications in Computational Physics, 22 (2017), pp. 1028-1048.

[34] X. Yang, Y. Yang, Y. Chen and J. Liu, Jacobi spectral collocation method sased on Lagrange interpolation polynomials for solving nonlinear fractional integro-differential equations, Adv. Appl. Math. Mech., 10 (2018), pp. 1440-1458.

[35] H. Wang, and X. Zheng, Wellposedness and regularity of the variable-order time-fractional diffusion equations. J. Math. Anal. Appl., 475 (2019), pp. 1778-1802.

Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong; and Geotechnical and Structural Engineering Research Center, Shandong University, Jinan, China
E-mail: `xguo@math.cuhk.edu.hk`

School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, China
E-mail: `liyutian@cuhk.edu.cn`

Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
E-mail: `zeng@math.cuhk.edu.hk`