

Fast Evaluation of the Caputo Fractional Derivative and its Applications to Fractional Diffusion Equations

Shidong Jiang¹, Jiwei Zhang^{2,*}, Qian Zhang² and Zhimin Zhang^{2,3}

¹ Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA.

² Beijing Computational Science Research Center, Beijing 100093, China.

³ Department of Mathematics, Wayne State University, Detroit, MI 48202, USA.

Communicated by Tao Zhou

Received 19 August 2016; Accepted (in revised version) 23 November 2016

Abstract. The computational work and storage of numerically solving the time fractional PDEs are generally huge for the traditional direct methods since they require total $\mathcal{O}(N_S N_T)$ memory and $\mathcal{O}(N_S N_T^2)$ work, where N_T and N_S represent the total number of time steps and grid points in space, respectively. To overcome this difficulty, we present an efficient algorithm for the evaluation of the Caputo fractional derivative ${}^C_0 D_t^\alpha f(t)$ of order $\alpha \in (0, 1)$. The algorithm is based on an efficient sum-of-exponentials (SOE) approximation for the kernel $t^{-1-\alpha}$ on the interval $[\Delta t, T]$ with a uniform absolute error ε . We give the theoretical analysis to show that the number of exponentials N_{exp} needed is of order $\mathcal{O}(\log N_T)$ for $T \gg 1$ or $\mathcal{O}(\log^2 N_T)$ for $T \approx 1$ for fixed accuracy ε . The resulting algorithm requires only $\mathcal{O}(N_S N_{\text{exp}})$ storage and $\mathcal{O}(N_S N_T N_{\text{exp}})$ work when numerically solving the time fractional PDEs. Furthermore, we also give the stability and error analysis of the new scheme, and present several numerical examples to demonstrate the performance of our scheme.

AMS subject classifications: 33C10, 33F05, 35Q40, 35Q55, 34A08, 35R11, 26A33

Key words: Fractional derivative, Caputo derivative, sum-of-exponentials approximation, fractional diffusion equation, fast convolution algorithm, stability analysis.

1 Introduction

Over the last few decades the fractional calculus has received much attention of both physical scientists and mathematicians since they can faithfully capture the dynamics of

*Corresponding author. *Email addresses:* shidong.jiang@njit.edu (S. Jiang), jwzhang@csrc.ac.cn (J. Zhang), qianzhang@csrc.ac.cn (Q. Zhang), zmzhang@csrc.ac.cn (Z. Zhang)

physical process in many applied sciences including biology, ecology, and control system. The anomalous diffusion, also referred to as the non-Gaussian process, has been observed and validated in many phenomena with accurate physical measurement [19,28,29,46,48,51]. The mathematical and numerical analysis of the fractional calculus became a subject of intensive investigations.

In this paper, we consider a fast evaluation of the following fractional partial differential equation:

$${}_0^C D_t^\alpha u(x,t) = \Delta u(x,t) + F(u,x,t), \quad 0 < \alpha < 1, \quad (1.1)$$

where the Caputo fractional derivative ${}_0^C D_t^\alpha u(x,t)$ is defined by the formula

$${}_0^C D_t^\alpha u(x,t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{u^{(m)}(x,\tau)}{(t-\tau)^{\alpha+1-m}} d\tau, \quad m-1 < \alpha < m, \quad m \in \mathbb{Z}. \quad (1.2)$$

The existing schemes for solving (1.1) require the storage of the solution at all previous time steps and the computational complexity of these schemes is $\mathcal{O}(N_T^2 N_S)$ with N_T the total number of time steps and N_S the number of grid points in space. This is in dark contrast with the usual diffusion equations where one only needs to store the solution at a fixed number of time steps and the computational complexity is linear with respect to N_T .

It is easy to see that the difficulty is caused by the Caputo fractional derivative appeared in (1.1). Indeed, one of the popular schemes of discretizing the Caputo fractional derivative is the so-called $L1$ approximation [15,16,23,30,32,36,39–41,52], which is simply based on the piecewise linear interpolation of u on each subinterval. For $0 < \alpha < 1$, the order of accuracy of the $L1$ approximation is $2 - \alpha$. There are also high-order discretization schemes by using piecewise high-order polynomial interpolation of u [10,17,33,47] and [9,34,37]. For each spatial point x , these methods require the storage of all previous function values $u(0), u(t_1), \dots, u(t_n)$ and $\mathcal{O}(n)$ flops at the n th step. Thus it requires on average $\mathcal{O}(N_T)$ storage and the total computational cost is $\mathcal{O}(N_T^2)$, which forms a bottleneck for long time simulations, especially when one tries to solve the time fractional partial differential equations (PDEs).

Here we present an efficient scheme for solving the fractional PDEs (1.1). Our key observation is that the Caputo derivative can be evaluated almost as efficient as the usually derivatives (besides some logarithmic factors). We first split the convolution integral in (1.2) into two parts - a local part containing the integral from $t - \Delta t$ to t , and a history part containing the integral from 0 to $t - \Delta t$. The local part is approximated using the standard $L1$ approximation. For the history part, integration by parts leads to a convolution integral of u with the kernel $t^{-1-\alpha}$. We show that $t^{-1-\alpha}$ ($0 < \alpha < 1$) admits an efficient sum-of-exponentials (SOE) approximation on the interval $[\delta, T]$ with $\delta = \Delta t$, a uniform absolute error ε and the number of exponentials needed is of the order

$$N_{\text{exp}} = \mathcal{O} \left(\log \frac{1}{\varepsilon} \left(\log \log \frac{1}{\varepsilon} + \log \frac{T}{\delta} \right) + \log \frac{1}{\delta} \left(\log \log \frac{1}{\varepsilon} + \log \frac{1}{\delta} \right) \right). \quad (1.3)$$

That is, for fixed precision ε , we have $N_{\text{exp}} = \mathcal{O}(\log N_T)$ for $T \gg 1$ or $N_{\text{exp}} = \mathcal{O}(\log^2 N_T)$ for $T \approx 1$, assuming that $N_T = \frac{T}{\Delta t}$. The approximation can be used to accelerate the evaluation of the convolution via the standard recurrence relation. The resulting algorithm has nearly optimal complexity – $\mathcal{O}(N_T N_{\text{exp}})$ work and $\mathcal{O}(N_{\text{exp}})$ storage. We would like to remark here that the SOE approximations have been applied to speed up the evaluation of the convolution integrals in many applications (see, for example, [1, 6, 20, 21, 24, 26, 31, 38, 45, 50, 53, 54, 58, 59] and references therein).

When we incorporate the fast evaluation scheme of the Caputo fractional derivative to solve the fractional diffusion equations (both linear and nonlinear), the resulting algorithm for solving the fractional PDEs is both efficient and stable. The computational cost of the new algorithm is $\mathcal{O}(N_S N_T N_{\text{exp}})$ as compared with $\mathcal{O}(N_S N_T^2)$ for direct methods and the storage requirement is only $\mathcal{O}(N_S N_{\text{exp}})$ as compared with $\mathcal{O}(N_S N_T)$ for direct methods, since one needs to store the solution in the whole computational spatial domain at all times. Furthermore, we have carried out a rigorous and detailed analysis to prove that our scheme is unconditionally stable with respect to arbitrary step sizes. With these two properties, our scheme provides an efficient and reliable tool for long time large scale simulation of fractional PDEs. In literatures, there are many other efforts to speed up the evaluation of weakly singular kernel. Lubich and Schädle [44] present a fast convolution for non-reflecting boundary conditions. Baffet and Hesthaven [3] compressed the kernel in the Laplace domain and obtained a sum-of-poles approximation for the Laplace transform of the kernel. And other fast approximation to time-fractional derivative is based on the block triangular Toeplitz matrix or block triangular Toeplitz-like matrix, see [27, 43].

The paper is organized as follows. In Section 2, we describe the fast algorithm for the evaluation of the Caputo fractional derivative and provide rigorous error analysis of our discretization scheme. In Section 3, we apply our fast algorithm to solve the linear fractional diffusion PDEs and present the stability and error analysis for the overall scheme. In Section 4, we study the nonlinear fractional diffusion PDEs and demonstrate that our fast algorithm has the same order of convergence as the direct method in this case. Finally, we conclude our paper with a brief discussion on the extension and generalization of our scheme.

2 Fast evaluation of the Caputo fractional derivative

In this section, we consider the fast evaluation of the Caputo fractional derivative for $0 < \alpha < 1$. Suppose that we would like to evaluate the Caputo fractional derivative on the interval $[0, T]$ over a set of grid points $\Omega_t := \{t_n, n = 0, 1, \dots, N_T\}$, with $t_0 = 0$, $t_{N_T} = T$, and $\Delta t_n = t_n - t_{n-1}$.

We first split the convolution integral in (1.2) into a sum of local part and history part, that is,

$${}_0^C D_t^\alpha u(t)|_{t=t_n} = \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} \frac{u'(s) ds}{(t_n - s)^\alpha}$$

$$\begin{aligned}
&= \frac{1}{\Gamma(1-\alpha)} \int_{t_{n-1}}^{t_n} \frac{u'(s)ds}{(t_n-s)^\alpha} + \frac{1}{\Gamma(1-\alpha)} \int_0^{t_{n-1}} \frac{u'(s)ds}{(t_n-s)^\alpha} \\
&:= C_l(t_n) + C_h(t_n),
\end{aligned} \tag{2.1}$$

where the last equality defines the local part and the history part, respectively. For the local part, we apply the standard $L1$ approximation, which approximates $u(s)$ on $[t_{n-1}, t_n]$ by a linear polynomial (with $u(t_{n-1})$ and $u(t_n)$ as the interpolation nodes) or $u'(s)$ by a constant $\frac{u(t_n)-u(t_{n-1})}{\Delta t_n}$. We have

$$C_l(t_n) \approx \frac{u(t_n)-u(t_{n-1})}{\Delta t_n \Gamma(1-\alpha)} \int_{t_{n-1}}^{t_n} \frac{1}{(t_n-s)^\alpha} ds = \frac{u(t_n)-u(t_{n-1})}{\Delta t_n^\alpha \Gamma(2-\alpha)}. \tag{2.2}$$

For the history part, we apply the integration by parts to eliminate $u'(s)$ and have

$$\begin{aligned}
C_h(t_n) &= \frac{1}{\Gamma(1-\alpha)} \int_0^{t_{n-1}} \frac{u'(s)ds}{(t_n-s)^\alpha} \\
&= \frac{1}{\Gamma(1-\alpha)} \left[\frac{u(t_{n-1})}{\Delta t_n^\alpha} - \frac{u(t_0)}{t_n^\alpha} - \alpha \int_0^{t_{n-1}} \frac{u(s)ds}{(t_n-s)^{1+\alpha}} \right].
\end{aligned} \tag{2.3}$$

2.1 Efficient sum-of-exponentials approximation for the power function

We now show that the kernel $t^{-\beta}$ ($0 < \beta < 2$) can be approximated via a sum-of-exponentials approximation efficiently on the interval $[\delta, T]$ with $\delta = \min_{1 \leq n \leq N} \Delta t_n$ and the absolute error ε . That is, there exist positive real numbers s_i and w_i ($i = 1, \dots, N_{\text{exp}}$) such that for $0 < \beta < 2$,

$$\left| \frac{1}{t^\beta} - \sum_{i=1}^{N_{\text{exp}}} \omega_i e^{-s_i t} \right| \leq \varepsilon, \quad t \in [\delta, T], \tag{2.4}$$

where N_{exp} is given by (1.3). The evaluation of the history part via (2.3) only needs the case $1 < \beta < 2$. However, one may also evaluate the history part directly without integration by parts and the case $0 < \beta < 1$ is then needed. Meanwhile, one can obtain the code for s_i and ω_i in (2.4) from the homepage <https://web.njit.edu/~jiang> or <http://www.csrc.ac.cn/~jwzhang/>.

We start our proof from the following integral representation of the power function.

Lemma 2.1. For any $\beta > 0$,

$$\frac{1}{t^\beta} = \frac{1}{\Gamma(\beta)} \int_0^\infty e^{-t \cdot s} s^{\beta-1} ds. \tag{2.5}$$

Proof. This follows from a change of variable $x = t \cdot s$ and the integral definition of the Γ function [49]. \square

Eq. (2.5) can be viewed as a representation of $t^{-\beta}$ using an infinitely many (continuous) exponentials. In order to obtain an efficient SOE approximation, we first truncate the

integral to a finite interval, then subdivide the finite interval into a set of dyadic intervals and discretize the integral on each dyadic interval with proper quadratures.

We now assume $0 < \beta < 2$.

Lemma 2.2. For $t \geq \delta > 0$,

$$\left| \frac{1}{\Gamma(\beta)} \int_p^\infty e^{-ts} s^{\beta-1} ds \right| \leq \begin{cases} e^{-\delta p} \frac{p^{\beta-1}}{\Gamma(\beta)} \frac{1}{\delta}, & 0 < \beta \leq 1, \\ e^{-\delta p} 2^{\beta-1} \left(\frac{p^\beta}{\Gamma(\beta)} + \frac{1}{\delta^\beta} \right), & 1 < \beta < 2. \end{cases} \quad (2.6)$$

Proof. For the case of $0 < \beta \leq 1$,

$$\begin{aligned} \left| \frac{1}{\Gamma(\beta)} \int_p^\infty e^{-ts} s^{\beta-1} ds \right| &= \left| \frac{1}{\Gamma(\beta)} e^{-tp} \int_0^\infty e^{-tx} (x+p)^{\beta-1} dx \right| \\ &\leq e^{-\delta p} \frac{p^{\beta-1}}{\Gamma(\beta)} \int_0^\infty e^{-\delta x} dx \\ &\leq e^{-\delta p} \frac{p^{\beta-1}}{\Gamma(\beta)} \frac{1}{\delta}. \end{aligned} \quad (2.7)$$

For the case of $1 < \beta < 2$,

$$\begin{aligned} \left| \frac{1}{\Gamma(\beta)} \int_p^\infty e^{-ts} s^{\beta-1} ds \right| &= \left| \frac{1}{\Gamma(\beta)} e^{-tp} \int_0^\infty e^{-tx} (x+p)^{\beta-1} dx \right| \\ &\leq e^{-\delta p} \frac{1}{\Gamma(\beta)} \left(\int_0^p (2p)^{\beta-1} dx + \int_p^\infty e^{-\delta x} (2x)^{\beta-1} dx \right) \\ &\leq e^{-\delta p} \frac{2^{\beta-1}}{\Gamma(\beta)} \left(p^\beta + \int_0^\infty e^{-\delta x} x^{\beta-1} dx \right) \\ &\leq e^{-\delta p} 2^{\beta-1} \left(\frac{p^\beta}{\Gamma(\beta)} + \frac{1}{\delta^\beta} \right). \end{aligned} \quad (2.8)$$

This completes the proof. \square

Remark 2.1. The truncation error can be made arbitrarily small for fixed δ by choosing sufficiently large p . Usually we have $\delta < 1$ and if one would like to bound the truncation error by $\varepsilon < 1/e$, then $\delta p > 1$ or $p > 1/\delta$. When $0 < \beta \leq 1$,

$$e^{-\delta p} \frac{p^{\beta-1}}{\Gamma(\beta)} \frac{1}{\delta} < e^{-\delta p} \frac{p^\beta}{\Gamma(\beta) p \delta} < e^{-\delta p} p, \quad (2.9)$$

when $1 < \beta < 2$,

$$e^{-\delta p} 2^{\beta-1} \left(\frac{p^\beta}{\Gamma(\beta)} + \frac{1}{\delta^\beta} \right) < e^{-\delta p} 2^{\beta-1} \left(\frac{p^\beta}{\Gamma(\beta)} + p^\beta \right) < 5e^{-\delta p} p^2. \quad (2.10)$$

Thus, $p = \mathcal{O}(\log(\frac{1}{\varepsilon\delta})/\delta)$ is sufficient to bound the truncation error by ε .

We now proceed to discuss the discretization error for the integral on the interval $[0, p]$. On each dyadic interval, we will discretize the integral via standard Gauss-Legendre quadrature and analyze the discretization error. The error can be controlled more or less uniformly since each dyadic interval is separated from the singular point, i.e., the origin by its own length.

Lemma 2.3. Consider a dyadic interval $[a, b] = [2^j, 2^{j+1}]$ and let s_1, \dots, s_n and w_1, \dots, w_n be the nodes and weights for n -point Gauss-Legendre quadrature on the interval. Then for $\beta \in (0, 2)$ and $n > 1$,

$$\left| \int_a^b e^{-ts} s^{\beta-1} ds - \sum_{k=1}^n w_k s_k^{\beta-1} e^{-s_k t} \right| < 2^{\frac{5}{2}} \pi a^\beta \left(\frac{e^{1/e}}{4} \right)^{2n}. \tag{2.11}$$

Proof. For any interval $[a, b]$, the standard estimate for n -point Gauss-Legendre quadrature [49] yields

$$\left| \int_a^b e^{-ts} s^{\beta-1} ds - \sum_{k=1}^n w_k s_k^{\beta-1} e^{-s_k t} \right| \leq \frac{(b-a)^{2n+1}}{2n+1} \frac{(n!)^4}{[(2n)!]^3} \max_{s \in (a,b)} \left| D_s^{2n}(e^{-st} s^{\beta-1}) \right|, \tag{2.12}$$

where D_s denotes the derivative with respect to s . Applying Stirling's approximation [49]

$$\sqrt{2\pi} n^{n+1/2} e^{-n} < n! < 2\sqrt{\pi} n^{n+1/2} e^{-n}, \tag{2.13}$$

we obtain

$$\frac{(n!)^4}{[(2n)!]^3} < 2\sqrt{\pi} \left(\frac{e}{8} \right)^{2n} \frac{\sqrt{n}}{n^{2n}}. \tag{2.14}$$

Observe now $|(\beta-1) \dots (\beta-k)| \leq k!$ for $k > 1$, and thus

$$\begin{aligned} \left| D_s^k s^{\beta-1} \right| &= s^{\beta-1} \leq 2\sqrt{\pi} \sqrt{2n} (2n)^k e^{-k} s^{\beta-k-1}, \quad \text{for } k=0, \\ \left| D_s^k s^{\beta-1} \right| &= |(\beta-1) s^{\beta-k-1}| \leq 2\sqrt{\pi} \sqrt{2n} (2n)^k e^{-k} s^{\beta-k-1}, \quad \text{for } k=1, \\ \left| D_s^k s^{\beta-1} \right| &= |(\beta-1) \dots (\beta-k) s^{\beta-k-1}| \leq k! s^{\beta-k-1} \\ &\leq 2\sqrt{\pi} k^{k+1/2} e^{-k} s^{\beta-k-1} \leq 2\sqrt{\pi} \sqrt{2n} (2n)^k e^{-k} s^{\beta-k-1}, \quad \text{for } k > 1. \end{aligned} \tag{2.15}$$

We also have

$$D_s^{2n-k} e^{-st} = (-t)^{2n-k} e^{-st}. \tag{2.16}$$

Combining (2.15), (2.16) with the Leibniz rule, we obtain

$$\begin{aligned} \left| D_s^{2n}(e^{-st} s^{\beta-1}) \right| &= \left| \sum_{k=0}^{2n} \binom{2n}{k} (D_s^{2n-k} e^{-st}) (D_s^k s^{\beta-1}) \right| \\ &\leq 2\sqrt{\pi} \sqrt{2n} s^{\beta-1} e^{-st} \sum_{k=0}^{2n} \binom{2n}{k} t^{2n-k} (2n)^k e^{-k} s^{-k} \\ &= 2\sqrt{\pi} \sqrt{2n} s^{\beta-1} e^{-st} \left(t + \frac{2n}{es} \right)^{2n}. \end{aligned} \tag{2.17}$$

Combining (2.12), (2.14), (2.17) and $b - a = a$, we have

$$\begin{aligned} \left| \int_a^b e^{-ts} s^{\beta-1} ds - \sum_{k=1}^n w_k s_k^{\beta-1} e^{-s_k t} \right| &\leq \max_{a \leq s \leq b} 2\sqrt{2}\pi(b-a)s^{\beta-1} e^{-st} \left(\frac{e(b-a)t}{8n} + \frac{b-a}{4s} \right)^{2n} \\ &\leq 2^{\frac{5}{2}} \pi a^\beta e^{-at} \left(\frac{eat}{8n} + \frac{1}{4} \right)^{2n}. \end{aligned} \tag{2.18}$$

And (2.11) follows from the fact

$$\max_{x>0} e^{-x} \left(\frac{ex}{8n} + \frac{1}{4} \right)^{2n} = \left(\frac{e^{1/e}}{4} \right)^{2n}, \quad n \geq 2. \tag{2.19}$$

This completes the proof. □

We now consider the end interval $[0, a]$. We will use Gauss-Jacobi quadrature with the weight function $s^{\beta-1}$ to discretize the integral. Under this weight, the function to be integrated is a simple exponential function. Thus the discretization error can be controlled to be arbitrarily small with properly chosen parameters.

Lemma 2.4. *Let s_1, \dots, s_n and w_1, \dots, w_n ($n \geq 2$) be the nodes and weights for n -point Gauss-Jacobi quadrature with the weight function $s^{\beta-1}$ on the interval. Then for $0 < t < T$, $\beta \in (0, 2)$ and $n > 1$,*

$$\left| \int_0^a e^{-ts} s^{\beta-1} ds - \sum_{k=1}^n w_k e^{-s_k t} \right| < 4\sqrt{\pi} a^\beta n^{3/2} \left(\frac{e}{8} \right)^{2n} \left(\frac{aT}{n-1} \right)^{2n}. \tag{2.20}$$

Proof. The standard estimate for n -point Gauss-Jacobi quadrature [49] yields

$$\left| \int_0^a e^{-ts} s^{\beta-1} ds - \sum_{k=1}^n w_k e^{-s_k t} \right| \leq \frac{a^{2n+\beta}}{2n+\beta} \frac{(n!)^2 [\Gamma(n+\beta)]^2}{(2n)! [\Gamma(2n+\beta)]^2} \max_{s \in (0,a)} |D_s^{2n} e^{-st}|. \tag{2.21}$$

For $n > 1$, we have $\Gamma(n+\beta) < \Gamma(n + \lceil \beta \rceil) = (n + \lceil \beta \rceil - 1)!$, $\Gamma(2n+\beta) > \Gamma(2n + \lfloor \beta \rfloor) = (2n + \lfloor \beta \rfloor - 1)!$, $2n + \beta > 2n + \lfloor \beta \rfloor$. Thus,

$$\begin{aligned} \left| \int_0^a e^{-ts} s^{\beta-1} ds - \sum_{k=1}^n w_k e^{-s_k t} \right| &\leq \max_{0 \leq s \leq a} \frac{a^{2n+\beta}}{2n + \lfloor \beta \rfloor} \frac{(n!)^2 [(n + \lceil \beta \rceil - 1)!]^2}{(2n)! [(2n + \lfloor \beta \rfloor - 1)!]^2} t^{2n} e^{-st} \\ &\leq \frac{a^\beta}{2n} \frac{(n!)^2 [(n + \lceil \beta \rceil - 1)!]^2}{(2n)! [(2n + \lfloor \beta \rfloor - 1)!]^2} (aT)^{2n} \\ &\leq 4\sqrt{\pi} a^\beta n^{3/2} \left(\frac{e}{8} \right)^{2n} \left(\frac{aT}{n-1} \right)^{2n}, \end{aligned} \tag{2.22}$$

where the last inequality follows from Stirling's approximation (2.13) and the fact that $\lceil \beta \rceil - \lfloor \beta \rfloor \leq 1$. □

We are now in a position to combine the last three lemmas to give an efficient SOE approximation for $t^{-\beta}$ on $[\delta, T]$ for $\beta \in (0, 2)$ as follows.

Theorem 2.1. Let $0 < \delta \leq t \leq T$ ($\delta \leq 1$ and $T \geq 1$), let $\varepsilon > 0$ be the desired precision, let $n_o = \mathcal{O}(\log \frac{1}{\varepsilon})$, let $M = \mathcal{O}(\log T)$, and let $N = \mathcal{O}(\log \log \frac{1}{\varepsilon} + \log \frac{1}{\delta})$. Furthermore, let $s_{o,1}, \dots, s_{o,n_o}$ and $w_{o,1}, \dots, w_{o,n_o}$ be the nodes and weights for the n_o -point Gauss-Jacobi quadrature on the interval $[0, 2^{-M}]$, let $s_{j,1}, \dots, s_{j,n_s}$ and $w_{j,1}, \dots, w_{j,n_s}$ be the nodes and weights for n_s -point Gauss-Legendre quadrature on small intervals $[2^j, 2^{j+1}]$, $j = -M, \dots, -1$, where $n_s = \mathcal{O}(\log \frac{1}{\varepsilon})$, and let $s_{j,1}, \dots, s_{j,n_l}$ and $w_{j,1}, \dots, w_{j,n_l}$ be the nodes and weights for n_l -point Gauss-Legendre quadrature on large intervals $[2^j, 2^{j+1}]$, $j = 0, \dots, N$, where $n_l = \mathcal{O}(\log \frac{1}{\varepsilon} + \log \frac{1}{\delta})$. Then for $t \in [\delta, T]$ and $\beta \in (0, 2)$,

$$\left| \frac{1}{t^\beta} - \left(\sum_{k=1}^{n_o} e^{-s_{o,k}t} w_{o,k} + \sum_{j=-M}^{-1} \sum_{k=1}^{n_s} e^{-s_{j,k}t} s_{j,k}^{\beta-1} w_{j,k} + \sum_{j=0}^N \sum_{k=1}^{n_l} e^{-s_{j,k}t} s_{j,k}^{\beta-1} w_{j,k} \right) \right| \leq \varepsilon. \quad (2.23)$$

Remark 2.2. The important fact which emerges from this theorem is that the total number of exponentials needed to approximate $t^{-\beta}$ for $0 < \delta \leq t \leq T$ with an absolute error ε is given by the formula (1.3), which gives the right complexity estimate for N_{exp} . However, even though the proof of Theorem 2.1 is explicitly constructive and one may even obtain the concrete number of exponentials needed by more detailed calculation, the resulting number of exponentials following the construction in Theorem 2.1 is *unnecessarily* large. This is because the construction is based on *locally* optimal quadrature on each dyadic interval, leading to a large prefactor in the \mathcal{O} complexity.

As a global after-processing optimization, one may apply modified Prony's method in [5] to reduce the number of exponentials for nodes on the interval $(0, 1)$, while standard model reduction method in [55] can be applied to reduce the number of exponentials for nodes on the interval $[1, p]$. Our numerical experiments indicate that there is about a factor of ten in the reduction of the number of exponentials after these further optimization steps. Thus, we state Theorem 2.1 using the \mathcal{O} notation.

Table 1 and Table 2 list the actual number of exponentials needed to approximate $t^{-1-\alpha}$ with various ε and $N_T = T/\Delta t$ after applying the reduction algorithms in Remark 2.2. We observe that the number of exponentials needed is very modest even for high accuracy approximations. Indeed, one needs less than 90 terms in order to march one million steps with 9-digit or 8-digit accuracy.

2.2 Fast evaluation of the history part

We replace the convolution kernel $t^{-1-\alpha}$ by its SOE approximation in (2.4) to approximate the history part defined in (2.3) as follows:

Table 1: Number of exponentials needed to approximate $t^{-1-\alpha}$ with fixed $\Delta t=10^{-3}$.

$\Delta t=10^{-3}$		$\alpha=0.2$				$\alpha=0.5$			
N_{exp}	$\frac{T}{\Delta t}$	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
ϵ									
	10^{-3}	25	29	33	37	30	35	40	44
	10^{-6}	34	40	45	51	39	46	52	59
	10^{-9}	43	51	58	66	49	57	65	74

Table 2: Number of exponentials needed to approximate $t^{-1-\alpha}$ with fixed $T=1$.

$T=1$		$\alpha=0.2$				$\alpha=0.5$			
N_{exp}	Δt	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
ϵ									
	10^{-3}	25	34	43	55	30	41	52	66
	10^{-6}	34	45	58	69	39	51	65	81
	10^{-8}	40	51	64	79	43	57	72	84

$$\begin{aligned}
 C_h(t_n) &\approx \frac{1}{\Gamma(1-\alpha)} \left[\frac{u(t_{n-1})}{\Delta t_n^\alpha} - \frac{u(t_0)}{t_n^\alpha} - \alpha \sum_{i=1}^{N_{\text{exp}}} \omega_i \int_0^{t_{n-1}} e^{-(t_n-\tau)s_i} u(\tau) d\tau \right] \\
 &= \frac{1}{\Gamma(1-\alpha)} \left[\frac{u(t_{n-1})}{\Delta t_n^\alpha} - \frac{u(t_0)}{t_n^\alpha} - \alpha \sum_{i=1}^{N_{\text{exp}}} \omega_i U_{\text{hist},i}(t_n) \right]. \tag{2.24}
 \end{aligned}$$

Define

$$U_{\text{hist},i}(t_n) = \int_0^{t_{n-1}} e^{-(t_n-\tau)s_i} u(\tau) d\tau.$$

To evaluate $U_{\text{hist},i}(t_n)$ for $n=1,2,\dots,N_T$, we observe the following simple recurrence relation:

$$U_{\text{hist},i}(t_n) = e^{-s_i \Delta t_n} U_{\text{hist},i}(t_{n-1}) + \int_{t_{n-2}}^{t_{n-1}} e^{-s_i(t_n-\tau)} u(\tau) d\tau. \tag{2.25}$$

At each time step, we only need $\mathcal{O}(1)$ work to compute $U_{\text{hist},i}(t_n)$ since $U_{\text{hist},i}(t_{n-1})$ is known at that point. Thus, the total work is reduced from $\mathcal{O}(N_T^2)$ to $\mathcal{O}(N_T N_{\text{exp}})$, and the total memory requirement is reduced from $\mathcal{O}(N_T)$ to $\mathcal{O}(N_{\text{exp}})$.

One may compute the integral on the right hand side of (2.25) by interpolating u via a linear function and then evaluating the resulting approximation analytically. We have

$$\begin{aligned}
 \int_{t_{n-2}}^{t_{n-1}} e^{-s_i(t_n-\tau)} u(\tau) d\tau &\approx \frac{e^{-s_i \Delta t_n}}{s_i^2 \Delta t_{n-1}} \left[(e^{-s_i \Delta t_{n-1}} - 1 + s_i \Delta t_{n-1}) u(t_{n-1}) \right. \\
 &\quad \left. + (1 - e^{-s_i \Delta t_{n-1}} - e^{-s_i \Delta t_{n-1}} s_i \Delta t_{n-1}) u(t_{n-2}) \right]. \tag{2.26}
 \end{aligned}$$

We note that the weights in front of $u(t_{n-1})$ and $u(t_{n-2})$ in (2.26) are subject to significant cancellation error when $s_i \Delta t_{n-1}$ is small. In that case, we can compute the weights by a Taylor expansion of exponentials with a small number of terms.

Remark 2.3. From above analysis, the fast evaluation of the Caputo fractional derivative is valid for non-uniform mesh. For simplicity, we will use uniform mesh in time with $\Delta t_n = \Delta t$ in the remainder of the section.

2.3 Error analysis

Define mesh functions

$$u^n = u(t_n), \quad 1 \leq n \leq N_T.$$

It is straightforward to verify that our scheme of evaluating the Caputo fractional derivative is equivalent to the following formula

$$\begin{aligned} {}_0^C D_t^\alpha u(t)|_{t=t_n} &\approx \frac{u^n - u^{n-1}}{\Delta t^\alpha \Gamma(2-\alpha)} + \frac{1}{\Gamma(1-\alpha)} \left[\frac{u^{n-1}}{\Delta t^\alpha} - \frac{u^0}{t_n^\alpha} - \alpha \sum_{i=1}^{N_{\text{exp}}} \omega_i U_{\text{hist},i}(t_n) \right] \\ &\triangleq {}_0^{FC} \mathbb{D}_t^\alpha u^n, \quad \text{for } n > 2. \end{aligned} \tag{2.27}$$

Noting that $U_{\text{hist},i}(t_n) = 0$ when $n = 1$, we have

$${}_0^{FC} \mathbb{D}_t^\alpha u^1 = \frac{\Delta t^{-\alpha}}{\Gamma(2-\alpha)} (u^1 - u^0). \tag{2.28}$$

Recall that the L1-approximation (based on the linear interpolation of the function u) of the Caputo derivative ${}_0^C D_t^\alpha u$ (see, for example, [42, 48]) is defined by the formula

$${}_0^C \mathbb{D}_t^\alpha u^n = \frac{\Delta t^{-\alpha}}{\Gamma(2-\alpha)} \left[a_0^{(\alpha)} u^n - \sum_{k=1}^{n-1} (a_{n-k-1}^{(\alpha)} - a_{n-k}^{(\alpha)}) u^k - a_{n-1}^{(\alpha)} u^0 \right], \tag{2.29}$$

where $a_k^{(\alpha)} := (k+1)^{1-\alpha} - k^{1-\alpha}$. The following lemma, which can be found in [52], establishes an error bound for the L1-approximation (2.29).

Lemma 2.5 (see [40, 52]). *Suppose that $f(t) \in C^2[0, t_n]$ and*

$$R^n u := {}_0^C D_t^\alpha u(t)|_{t=t_n} - {}_0^C \mathbb{D}_t^\alpha u^n, \tag{2.30}$$

where $0 < \alpha < 1$. Then

$$|R^n u| \leq \frac{\Delta t^{2-\alpha}}{\Gamma(2-\alpha)} \left(\frac{1-\alpha}{12} + \frac{2^{2-\alpha}}{2-\alpha} - (1+2^{-\alpha}) \right) \max_{0 \leq t \leq t_n} |u''(t)|. \tag{2.31}$$

The following lemma provides an error bound for our fast evaluation scheme, denoted by ${}_0^{FC} \mathbb{D}_t^\alpha u^n$ in (2.27) and (2.28).

Lemma 2.6. *Suppose that $u(t) \in C^2[0, t_n]$ and let*

$${}^F R^n u := {}^C D_t^\alpha u(t) \Big|_{t=t_n} - {}^F C D_t^\alpha u^n, \tag{2.32}$$

where $0 < \alpha < 1$. Then

$$|{}^F R^n u| \leq \frac{\Delta t^{2-\alpha}}{\Gamma(2-\alpha)} \left(\frac{1-\alpha}{12} + \frac{2^{2-\alpha}}{2-\alpha} - (1+2^{-\alpha}) \right) \max_{0 \leq t \leq t_n} |u''(t)| + \frac{\alpha \varepsilon t_{n-1}}{\Gamma(1-\alpha)} \max_{0 \leq t \leq t_{n-1}} |u(t)|. \tag{2.33}$$

Proof. Obviously the only difference between our approximation ${}^F C D_t^\alpha u^n$ and the L1-approximation ${}^C D_t^\alpha u^n$ is that the convolution kernel admits an absolute error bounded by ε in its sum-of-exponentials approximation (2.4), namely,

$$\left| {}^F C D_t^\alpha u^n - {}^C D_t^\alpha u^n \right| \leq \frac{\alpha \varepsilon}{\Gamma(1-\alpha)} \sum_{l=1}^{n-1} \int_{t_{l-1}}^{t_l} |\Pi_{1,l} u(s)| ds, \tag{2.34}$$

where $\Pi_{1,l} u(t) = u(t_{l-1}) \frac{t_l-t}{\Delta t} + u(t_l) \frac{t-t_{l-1}}{\Delta t}$. And the triangle inequality leads to

$$|{}^F R^n u| \leq |{}^C R^n u| + \frac{\alpha \varepsilon}{\Gamma(1-\alpha)} \sum_{l=1}^{n-1} \int_{t_{l-1}}^{t_l} |\Pi_{1,l} u(s)| ds, \tag{2.35}$$

where

$$\sum_{l=1}^{n-1} \int_{t_{l-1}}^{t_l} |\Pi_{1,l} u(s)| ds \leq \max_{0 \leq t \leq t_{n-1}} |u(t)| t_{n-1}. \tag{2.36}$$

Combining Lemma 2.5 and (2.36), we obtain Lemma 2.6. □

We also have the following useful inequality. The proof is given in Appendix A.

Lemma 2.7. *For any mesh functions $g = \{g^k | 0 \leq k \leq N\}$ defined on Ω_t , the following inequality holds:*

$$\Delta t \sum_{k=1}^n ({}^F C D_t^\alpha g^k) g^k \geq \frac{t_n^{-\alpha} - 2\alpha \varepsilon t_{n-1}}{2\Gamma(1-\alpha)} \Delta t \sum_{k=1}^n (g^k)^2 - \frac{t_n^{1-\alpha} + \alpha(1-\alpha)\varepsilon t_{n-1} \Delta t}{2\Gamma(2-\alpha)} (g^0)^2. \tag{2.37}$$

The results in Lemmas 2.6 and 2.7 are useful in proving the stability and convergence properties of the overall numerical scheme for solving time-fractional PDEs when our fast evaluation scheme is applied to compute the Caputo fractional derivative.

3 Application I: Linear fractional diffusion equation

We now consider the computation of the initial value (IV) problem of the linear fractional diffusion equation, given by

$${}^C_0D_t^\alpha u(x,t) = u_{xx}(x,t) + f(x,t), \quad x \in \mathbb{R}, \quad t > 0, \tag{3.1}$$

$$u(x,0) = u_0(x), \quad x \in \mathbb{R}, \tag{3.2}$$

$$u(x,t) \rightarrow 0, \quad \text{when } |x| \rightarrow \infty, \tag{3.3}$$

where the initial data u_0 and the source term $f(x,t)$ are assumed to be compactly supported in the interval $\Omega_i := \{x|x_l < x < x_r\}$. To solve this problem using a finite difference scheme, one needs to truncate the computational domain to a finite interval and impose some boundary conditions at the end points, see [2,4,7,14–16,18,22]. The exact nonreflecting boundary conditions for the above problem have been derived in [15] via standard Laplace transform method and it is shown in [15] that the above problem is equivalent to the following initial-boundary value problem (IBVP)

$${}^C_0D_t^\alpha u(x,t) = u_{xx}(x,t) + f(x,t), \quad x \in \Omega_i, \quad t > 0, \tag{3.4}$$

$$u(x,0) = u_0(x), \quad x \in \Omega_i, \tag{3.5}$$

$$\frac{\partial u(x,t)}{\partial x} = \frac{1}{\Gamma(1-\frac{\alpha}{2})} \int_0^t \frac{u_s(x,s)}{(t-s)^{\frac{\alpha}{2}}} ds := {}^C_0D_t^{\frac{\alpha}{2}} u(x,t), \quad x = x_l, \tag{3.6}$$

$$\frac{\partial u(x,t)}{\partial x} = -\frac{1}{\Gamma(1-\frac{\alpha}{2})} \int_0^t \frac{u_s(x,s)}{(t-s)^{\frac{\alpha}{2}}} ds := -{}^C_0D_t^{\frac{\alpha}{2}} u(x,t), \quad x = x_r. \tag{3.7}$$

We now incorporate the fast evaluation of the Caputo fractional derivative into the existing finite difference (FD) scheme introduced in [15] to numerically solve the IBVP (3.4)-(3.7), establish the corresponding stability and convergence properties, and present an example to demonstrate the performance of our scheme by comparing it with the direct scheme given in [15].

3.1 Construction of the finite difference scheme with fast evaluation

We first introduce some standard notations. For two given positive integers N_T and N_S , let $\{t_n\}_{n=0}^{N_T}$ be an equidistant partition of $[0, T]$ with $t_n = n\Delta t$ and $\Delta t = T/N_T$, and let $\{x_j\}_{j=0}^{N_S}$ be a partition of (x_l, x_r) with $x_i = x_l + ih$ and $h = (x_r - x_l)/N_S$. Denote

$$\begin{aligned} \delta_t u_i^n &= \frac{u_i^n - u_i^{n-1}}{\Delta t}, & \delta_x u_{i+\frac{1}{2}}^n &= \frac{u_{i+1}^n - u_i^n}{h}, \\ \delta_x^2 u_i^n &= \frac{\delta_x u_{i+\frac{1}{2}}^n - \delta_x u_{i-\frac{1}{2}}^n}{h}, & \tilde{\delta}_x u_i^n &= \frac{u_{i+1}^n - u_{i-1}^n}{2h}. \end{aligned}$$

Lemma 3.1 (see [52]). *Suppose that $u \in C^3[x_l, x_r]$. Then*

$$u_{xx}(x_0) - \frac{2}{h} \left[\delta_x u_{\frac{1}{2}} - u_x(x_0) \right] = -\frac{h}{3} u_{xxx}(x_0 + \theta_1 h), \quad \theta_1 \in (0, 1), \tag{3.8}$$

$$u_{xx}(x_{N_S}) - \frac{2}{h} \left[u_x(x_{N_S}) - \delta_x u_{N_S - \frac{1}{2}} \right] = \frac{h}{3} u_{xxx}(x_{N_S} - \theta_2 h), \quad \theta_2 \in (0, 1). \tag{3.9}$$

The FD scheme in [15] for the problem (3.4)-(3.7) can be written by

$${}^C_0\mathbb{D}_t^\alpha u_i^n = \delta_x^2 u_i^n + f_i^n, \quad 1 \leq i \leq N_S - 1, \quad 1 \leq n \leq N_T, \tag{3.10}$$

$${}^C_0\mathbb{D}_t^\alpha u_0^n = \frac{2}{h} \left[\delta_x u_{\frac{1}{2}}^n - {}^C_0\mathbb{D}_t^{\alpha/2} u_0^n \right] + f_0^n, \quad 1 \leq n \leq N_T, \tag{3.11}$$

$${}^C_0\mathbb{D}_t^\alpha u_{N_S}^n = \frac{2}{h} \left[-\delta_x u_{N_S - \frac{1}{2}}^n - {}^C_0\mathbb{D}_t^{\alpha/2} u_{N_S}^n \right] + f_{N_S}^n, \quad 1 \leq n \leq N_T, \tag{3.12}$$

$$u_i^0 = u_0(x_i), \quad 0 \leq i \leq N_S. \tag{3.13}$$

Replacing the standard L1-approximation ${}^C_0\mathbb{D}$ for the Caputo derivative by our fast evaluation scheme ${}^{FC}_0\mathbb{D}$, we obtain a fast FD scheme of the following form

$${}^{FC}_0\mathbb{D}_t^\alpha u_i^n = \delta_x^2 u_i^n + f_i^n, \quad 1 \leq i \leq N_S - 1, \quad 1 \leq n \leq N_T, \tag{3.14}$$

$${}^{FC}_0\mathbb{D}_t^\alpha u_0^n = \frac{2}{h} \left[\delta_x u_{\frac{1}{2}}^n - {}^{FC}_0\mathbb{D}_t^{\alpha/2} u_0^n \right] + f_0^n, \quad 1 \leq n \leq N_T, \tag{3.15}$$

$${}^{FC}_0\mathbb{D}_t^\alpha u_{N_S}^n = \frac{2}{h} \left[-\delta_x u_{N_S - \frac{1}{2}}^n - {}^{FC}_0\mathbb{D}_t^{\alpha/2} u_{N_S}^n \right] + f_{N_S}^n, \quad 1 \leq n \leq N_T, \tag{3.16}$$

$$u_i^0 = u_0(x_i), \quad 0 \leq i \leq N_S. \tag{3.17}$$

3.2 Stability and error analysis of the new scheme

The stability and error analysis of the scheme (3.10)-(3.13) have been given in [15]. In this section, we study the corresponding theoretical analysis for our new scheme (3.14)-(3.17). We first denote $\mathcal{S}_h = \{u | u = (u_0, u_1, \dots, u_{N_S})\}$.

Lemma 3.2 ([15]). *For any mesh function $u \in \mathcal{S}_h$, the following inequality holds*

$$\|u\|_\infty^2 \leq \theta \|\delta_x u\|^2 + \left(\frac{1}{\theta} + \frac{1}{L}\right) \|u\|^2, \quad \forall \theta > 0,$$

where L is the length of the computational domain and here $L = x_r - x_l$.

We now show the prior estimate for the solution of the scheme (3.14)-(3.17).

Theorem 3.1 (Prior Estimate). *Suppose $\{u_i^k | 0 \leq i \leq N_S, 0 \leq k \leq N_T\}$ is the solution of the finite difference scheme (3.14)-(3.17). Then for any $1 \leq n \leq N_T$,*

$$\begin{aligned} \Delta t \sum_{k=1}^n \|u^k\|_\infty^2 &\leq \frac{2(1 + \sqrt{1 + L^2 \mu})}{L\mu} \left(\rho \|u^0\|^2 + \varrho [(u_0^0)^2 + (u_{N_S}^0)^2] \right. \\ &\quad \left. + \frac{\Delta t}{8\nu} \sum_{k=1}^n [(hf_0^k)^2 + (hf_{N_S}^k)^2] + \frac{\Delta t}{\mu} \sum_{k=1}^n h \sum_{i=1}^{N_S-1} (f_i^k)^2 \right), \end{aligned} \tag{3.18}$$

where

$$\begin{aligned} \rho &= \frac{T^{1-\alpha}}{2\Gamma(2-\alpha)}, & \mu &= \frac{T^{-\alpha} - 2\alpha\epsilon T}{\Gamma(1-\alpha)}, \\ \varrho &= \frac{T^{1-\frac{\alpha}{2}}}{2\Gamma(2-\frac{\alpha}{2})}, & \nu &= \frac{T^{-\frac{\alpha}{2}} - \alpha\epsilon T}{\Gamma(1-\frac{\alpha}{2})}. \end{aligned} \tag{3.19}$$

Proof. Multiplying hu_i^k on both sides of (3.14), and summing up for i from 1 to $N_S - 1$, we have

$$h \sum_{i=1}^{N_S-1} ({}^{\text{FC}}\mathbb{D}_t^\alpha u_i^k) u_i^k - h \sum_{i=1}^{N_S-1} (\delta_x^2 u_i^k) u_i^k = h \sum_{i=1}^{N_S-1} f_i^k u_i^k.$$

Multiplying $\frac{h}{2}u_0^k$ and $\frac{h}{2}u_{N_S}^k$ on both sides of (3.15) and (3.16), respectively, then adding the results with the above identity, we obtain

$$\begin{aligned} &({}^{\text{FC}}\mathbb{D}_t^\alpha u^k, u^k) + \left[-(\delta_x v_{\frac{1}{2}}^k) u_0^k - h \sum_{i=1}^{N_S-1} (\delta_x^2 u_i^k) u_i^k + (\delta_x u_{N_S-\frac{1}{2}}^k) u_{N_S}^k \right] \\ &+ ({}^{\text{FC}}\mathbb{D}_t^{\frac{\alpha}{2}} u_0^k) u_0^k + ({}^{\text{FC}}\mathbb{D}_t^{\frac{\alpha}{2}} u_{N_S}^k) u_{N_S}^k = \frac{1}{2}(hf_0^k) u_0^k + h \sum_{i=1}^{N_S-1} f_i^k u_i^k + \frac{1}{2}(hf_{N_S}^k) u_{N_S}^k. \end{aligned} \tag{3.20}$$

Applying the summation by parts, we have

$$-(\delta_x v_{\frac{1}{2}}^k) u_0^k - h \sum_{i=1}^{N_S-1} (\delta_x^2 u_i^k) u_i^k + (\delta_x u_{N_S-\frac{1}{2}}^k) u_{N_S}^k = \|\delta_x u^k\|^2. \tag{3.21}$$

Substituting (3.21) into (3.20), and multiplying Δt on both sides of the resulting identity, and summing up for k from 1 to n , it follows from Lemma 2.7 that

$$\begin{aligned} &\Delta t \frac{t_n^{-\alpha} - 2\alpha\epsilon t_{n-1}}{2\Gamma(1-\alpha)} \sum_{k=1}^n \|u^k\|^2 + \Delta t \frac{t_n^{-\frac{\alpha}{2}} - \alpha\epsilon t_{n-1}}{2\Gamma(1-\frac{\alpha}{2})} \sum_{k=1}^n [(u_0^k)^2 + (u_{N_S}^k)^2] + \Delta t \sum_{k=1}^n \|\delta_x u^k\|^2 \\ &\leq \frac{t_n^{1-\alpha} + \alpha(1-\alpha)\epsilon t_{n-1}\Delta t}{2\Gamma(2-\alpha)} \|u^0\|^2 + \frac{t_n^{1-\frac{\alpha}{2}} + \frac{\alpha}{2}(1-\frac{\alpha}{2})\epsilon t_{n-1}\Delta t}{2\Gamma(2-\frac{\alpha}{2})} [(u_0^0)^2 + (u_{N_S}^0)^2] \\ &+ \Delta t \sum_{k=1}^n \left[\frac{1}{2}(hf_0^k) u_0^k + h \sum_{i=1}^{N_S-1} f_i^k u_i^k + \frac{1}{2}(hf_{N_S}^k) u_{N_S}^k \right]. \end{aligned} \tag{3.22}$$

Applying the Cauchy-Schwarz inequality, we obtain

$$\frac{1}{2}(hf_0^k) u_0^k + h \sum_{i=1}^{N_S-1} f_i^k u_i^k + \frac{1}{2}(hf_{N_S}^k) u_{N_S}^k$$

$$\begin{aligned}
 &\leq \frac{t_n^{-\frac{\alpha}{2}} - \alpha \varepsilon t_{n-1}}{2\Gamma(1-\frac{\alpha}{2})} \left[(u_0^k)^2 + (u_{N_s}^k)^2 \right] + \frac{\Gamma(1-\frac{\alpha}{2})}{8(t_n^{-\frac{\alpha}{2}} - \alpha \varepsilon t_{n-1})} \left[(hf_0^k)^2 + (hf_{N_s}^k)^2 \right] \\
 &\quad + h \sum_{i=1}^{N_s-1} \left[\frac{t_n^{-\alpha} - 2\alpha \varepsilon t_{n-1}}{4\Gamma(1-\alpha)} (u_i^k)^2 + \frac{\Gamma(1-\alpha)}{t_n^{-\alpha} - 2\alpha \varepsilon t_{n-1}} (f_i^k)^2 \right] \\
 &\leq \frac{t_n^{-\frac{\alpha}{2}} - \alpha \varepsilon t_{n-1}}{2\Gamma(1-\frac{\alpha}{2})} \left[(u_0^k)^2 + (u_{N_s}^k)^2 \right] + \frac{\Gamma(1-\frac{\alpha}{2})}{8(t_n^{-\frac{\alpha}{2}} - \alpha \varepsilon t_{n-1})} \left[(hf_0^k)^2 + (hf_{N_s}^k)^2 \right] \\
 &\quad + \frac{t_n^{-\alpha} - 2\alpha \varepsilon t_{n-1}}{4\Gamma(1-\alpha)} \|u^k\|^2 + h \sum_{i=1}^{N_s-1} \frac{\Gamma(1-\alpha)}{t_n^{-\alpha} - 2\alpha \varepsilon t_{n-1}} (f_i^k)^2. \tag{3.23}
 \end{aligned}$$

The substitution of (3.23) into (3.22) produces

$$\begin{aligned}
 \frac{\mu^n}{4} \Delta t \sum_{k=1}^n \|u^k\|^2 + \Delta t \sum_{k=1}^n \|\delta_x u^k\|^2 &\leq \rho^n \|u^0\|^2 + \varrho^n [(u_0^0)^2 + (u_{N_s}^0)^2] \\
 &\quad + \frac{\Delta t}{8\nu^n} \sum_{k=1}^n [(hf_0^k)^2 + (hf_{N_s}^k)^2] + \frac{\Delta t}{\mu^n} \sum_{k=1}^n h \sum_{i=1}^{N_s-1} (f_i^k)^2, \tag{3.24}
 \end{aligned}$$

where

$$\begin{aligned}
 \rho^n &= \frac{t_n^{1-\alpha} + \alpha(1-\alpha)\varepsilon t_{n-1}\Delta t}{2\Gamma(2-\alpha)}, & \mu^n &= \frac{t_n^{-\alpha} - 2\alpha \varepsilon t_{n-1}}{\Gamma(1-\alpha)}, \\
 \varrho^n &= \frac{t_n^{1-\frac{\alpha}{2}} + \frac{\alpha}{2}(1-\frac{\alpha}{2})\varepsilon t_{n-1}\Delta t}{2\Gamma(2-\frac{\alpha}{2})}, & \nu^n &= \frac{t_n^{-\frac{\alpha}{2}} - \alpha \varepsilon t_{n-1}}{\Gamma(1-\frac{\alpha}{2})}. \tag{3.25}
 \end{aligned}$$

Taking $\theta^n > 0$ such that $\frac{1/\theta^n + 1/L}{\theta^n} = \frac{\mu^n}{4}$ (i.e., $\theta^n = 2(1 + \sqrt{1 + L^2\mu^n}) / (L\mu^n)$), and following from Lemma 3.2, we have

$$\Delta t \sum_{k=1}^n \|u^k\|_\infty^2 \leq \frac{2(1 + \sqrt{1 + L^2\mu^n})}{L\mu^n} \left(\frac{\mu^n}{4} \Delta t \sum_{k=1}^n \|u^k\|^2 + \Delta t \sum_{k=1}^n \|\delta_x u^k\|^2 \right). \tag{3.26}$$

Combining (3.24) with (3.26), and notice

$$\rho^n \leq \rho, \quad \varrho^n \leq \varrho, \quad \mu^n \geq \mu, \quad \nu^n \geq \nu,$$

we obtain the inequality (3.18). □

The priori estimate leads to the stability of the scheme (3.14)-(3.17).

Theorem 3.2 (Stability). *The scheme (3.14)-(3.17) is unconditionally stable for any given compactly supported initial data and source term.*

We now present the error analysis of the scheme (3.14)-(3.17) whose proof can be found in Appendix B.

Theorem 3.3 (Error Analysis). *Suppose $u(x,t) \in C_{x,t}^{4,2}([x_l, x_r] \times [0, T])$ and $\{u_i^k | 0 \leq i \leq N_S, 0 \leq k \leq N_T\}$ are solutions of the problem (3.4)-(3.7) and the difference scheme (3.14)-(3.17), respectively. Let $e_i^k = u_i^k - u(x_i, t_k)$. Then there exists a positive constant c_2 such that*

$$\sqrt{\Delta t \sum_{k=1}^n \|e^k\|_\infty^2} \leq c_2(h^2 + \Delta t^{2-\alpha} + \varepsilon), \quad 1 \leq n \leq N_T, \tag{3.27}$$

where $c_2^2 = \frac{4c_1^2 \Gamma(1 + \sqrt{1 + L^2 \mu})}{L\mu} (\frac{1}{v} + \frac{L}{\mu})$ with c_1 is a positive constant (see (B.5)-(B.7)), and μ, v are defined in (3.19).

Remark 3.1. The convergence rate established in [15] for the scheme (3.10)-(3.13) is $C(h^2 + \Delta t^{2-\alpha})$, which differs from the error estimate in (3.27) by a tolerance error ε , see Fig. 1 for example. This implies that the fast scheme (3.14)-(3.17) has almost the same accuracy and convergence rate as the direct scheme (3.10)-(3.13) when ε is chosen to be smaller than $h^2 + \Delta t^{2-\alpha}$, but is much faster ($\mathcal{O}(N_S N_T N_{\text{exp}})$ vs $\mathcal{O}(N_S N_T^2)$ work) and needs less storage ($\mathcal{O}(N_S N_{\text{exp}})$ vs $\mathcal{O}(N_S N_T)$ memory) when the total time step number N_T is moderate and large.

3.3 Numerical results

In this section, we illustrate the performance of our scheme (3.14)-(3.17) by comparing it with the scheme (3.10)-(3.13) from various aspects – convergence rate and computational cost. We observe that the scheme (3.14)-(3.17) has the same convergence rate $\mathcal{O}(h^2 + \Delta t^{2-\alpha})$, but is much faster than the scheme (3.10)-(3.13) even when N_T is of moderate size.

Example 3.1. Now we use the example in [15] with the exact solution of the problem (3.1)-(3.3)

$$u(x,t) = x^4(\pi - x)^4 [\exp(-x)t^{3+\alpha} + 1], \quad (x,t) \in \Omega_i \times [0, T], \tag{3.28}$$

and the source term $f(x,t)$ in the form of

$$f(x,t) = \begin{cases} \Gamma(4+\alpha)x^4(\pi-x)^4 \exp(-x)t^3/6 - x^2(\pi-x)^2\{t^{3+\alpha} \exp(-x) \\ [x^2(56-16x+x^2) - 2\pi x(28-12x+x^2) + \pi^2(12-8x+x^2)] \\ + 4(3\pi^2 - 14\pi x + 14x^2)\}, & (x,t) \in \Omega_i \times [0, T], \\ 0, & (x,t) \notin \Omega_i \times [0, T]. \end{cases}$$

We define the maximum norm of the error and the convergence rates with respect to temporal and spatial sizes, respectively by the formulas

$$E(h, \Delta t) = \sqrt{\Delta t \sum_{k=1}^{N_T} \|e^k\|_\infty^2}, \quad r_t = \log_2 \frac{E(h, \Delta t)}{E(h, \Delta t/2)}, \quad r_s = \log_2 \frac{E(h, \Delta t)}{E(h/2, \Delta t)},$$

Table 3: Errors and convergence orders in time with fixed spatial size $h = \pi/20000$ for the fast scheme (3.14)-(3.17) and the direct scheme (3.10)-(3.13).

Δt	$\alpha = 0.2$				$\alpha = 0.5$			
	Fast scheme		Direct scheme		Fast scheme		Direct scheme	
	$E(h, \tau)$	r_t	$E(h, \tau)$	r_t	$E(h, \tau)$	r_t	$E(h, \tau)$	r_t
1/10	1.570e-02	1.70	1.570e-02	1.70	8.151e-02	1.48	8.151e-02	1.48
1/20	4.846e-03	1.70	4.844e-03	1.70	2.922e-02	1.47	2.923e-02	1.47
1/40	1.489e-03	1.72	1.488e-03	1.71	1.052e-02	1.48	1.052e-02	1.48
1/80	4.524e-04	1.70	4.541e-04	1.72	3.781e-03	1.48	3.784e-03	1.48
1/160	1.395e-04		1.375e-04		1.357e-03		1.357e-03	

Table 4: The errors, convergence orders in space, and CPU time in seconds with fixed temporal step size $\Delta t = 1/30000$ for the fast scheme (3.14)-(3.17) and the direct scheme (3.10)-(3.13).

h	$\alpha = 0.2$				$\alpha = 0.5$			
	Fast scheme		Direct scheme		Fast scheme		Direct scheme	
	$E(h, \tau)$	r_s	$E(h, \tau)$	r_s	$E(h, \tau)$	r_s	$E(h, \tau)$	r_s
$\pi/10$	8.862e-01	2.06	8.632e-01	2.03	8.652e-01	2.05	8.035e-01	1.99
$\pi/20$	2.122e-01	2.01	2.107e-01	2.01	2.087e-01	2.01	2.016e-01	1.99
$\pi/40$	5.258e-02	2.00	5.244e-02	2.00	5.177e-02	2.00	5.064e-02	1.99
$\pi/80$	1.312e-02	2.00	1.311e-02	2.00	1.292e-02	2.00	1.272e-02	1.99
$\pi/160$	3.280e-03		3.277e-03		3.229e-03		3.195e-03	
CPU(s)	43.37		3304.66		43.65		2226.06	

where the error e^k is measured against the exact solution (3.28).

In our calculation, we set the computational domain $\Omega_i = [0, \pi]$, $T = 1$ and the tolerance precision $\varepsilon = 10^{-7}$ for the SOE approximation. Table 3 shows the errors in maximum norm and temporal convergence rate $\mathcal{O}(\Delta t^{2-\alpha})$ for $\alpha = 0.2, 0.5$ computed by the schemes (3.14)-(3.17) and (3.10)-(3.13), respectively, where the spatial mesh size is fixed at $h = \frac{\pi}{20000}$. And Table 4 shows the relative L_∞ errors, spatial convergence rate $\mathcal{O}(h^2)$, and CPU time for $\alpha = 0.2$ and $\alpha = 0.5$, where the temporal mesh size is fixed at $\Delta t = \frac{1}{30000}$.

From Tables 3 and 4, it is not surprising to see that both schemes have the same errors and convergence rate $\mathcal{O}(h^2 + \Delta t^{2-\alpha})$. The reason is that, as shown in Theorem 3.3, the influence of the SOE approximation error ε is negligible when it is chosen to be very small. To further investigate the conclusion of Theorem 3.3, we plot in Fig. 1 the evolution of the difference of the solutions to the schemes (3.14)-(3.17) and (3.10)-(3.13) using the same mesh sizes. One can see that the difference of the solutions is smaller than the SOE approximation precision.

In Fig. 2, we plot out the CPU time (in seconds) of the two schemes by fixing all other parameters and only varying the total number of time steps N_T . We observe that while the direct scheme scales like $\mathcal{O}(N_T^2)$, the CPU time increases almost linearly with the total number of time steps N_T for the fast scheme.

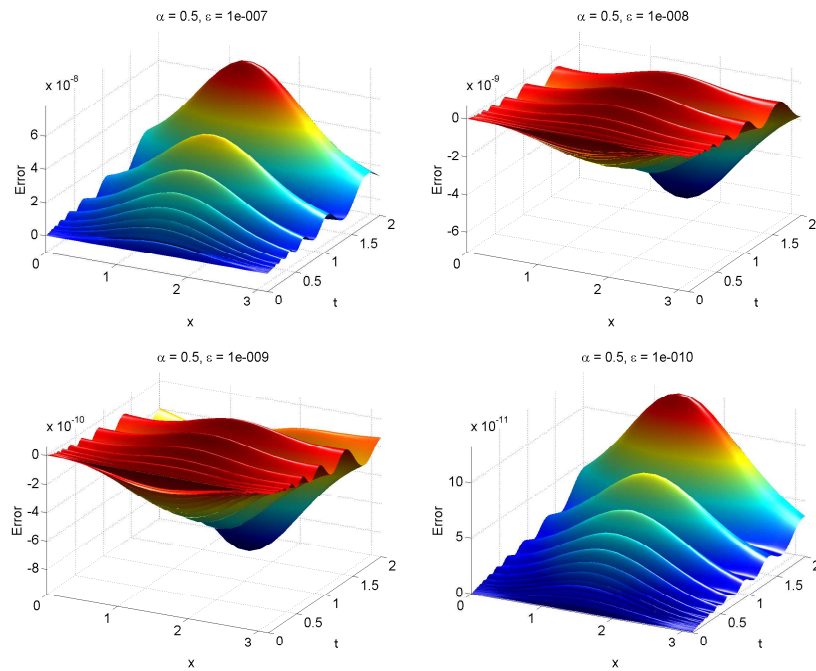


Figure 1: Example 3.1: The evolution of the difference (i.e. $Error = {}^F U_h^n - {}^D U_h^n$) of the solutions to the schemes (3.14)-(3.17) (denote by ${}^D U_h^n$) and (3.10)-(3.13) (denote by ${}^F U_h^n$) for $\epsilon = 1e-7, 1e-8, 1e-9, 1e-10$, respectively. Here we set $T = 2$, $\Omega_i = [0, \pi]$, $M = 80$, $N = 2000$, and $\alpha = 0.5$.

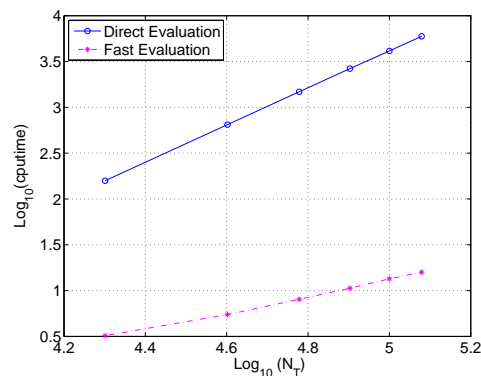


Figure 2: Example 3.1: The log-log (in base 10) plot of the CPU time (in seconds) versus the total number of time steps N_T for Application I. Here $N_S = 30$ and $\alpha = 0.5$.

4 Application II: Nonlinear fractional diffusion equation

In this section, we apply the fast evaluation of the Caputo derivative to solve the following nonlinear fractional diffusion equation

$$\begin{aligned} {}_0^C D_t^\alpha u(x, t) &= u_{xx} + f(u), \\ u(x, 0) &= u_0(x). \end{aligned} \tag{4.1}$$

If $f(u) = -u(1-u)$ or $f(u) = -0.1u(1-u)(u-0.001)$ (4.1) is called the time fractional Fisher equation or Huxley equation, respectively.

Suppose that the initial value $u_0(x)$ is compactly supported on $\Omega_i = [x_l, x_r]$. Efficient nonlinear absorbing boundary conditions (ABCs) have been derived in [8,35,56,57]. With these ABCs, the IVP (4.1) on unbounded domain can be reformulated into an IBVP on a bounded domain, given by

$${}_0^C D_t^\alpha u(x,t) = u_{xx} + f(u), \quad 0 < t \leq T, \quad x \in \Omega_i, \quad (4.2)$$

$$u(x,0) = u_0(x), \quad x \in \Omega_i, \quad (4.3)$$

$$(\partial_x + 3s_0^{\frac{\alpha}{2}}) {}_0^C D_t^\alpha u + (3s_0^\alpha \partial_x + s_0^{\frac{3\alpha}{2}}) u = (\partial_x + 3s_0^{\frac{\alpha}{2}}) f(u), \quad x = x_r, \quad (4.4)$$

$$(\partial_x - 3s_0^{\frac{\alpha}{2}}) {}_0^C D_t^\alpha u + (3s_0^\alpha \partial_x - s_0^{\frac{3\alpha}{2}}) u = (\partial_x - 3s_0^{\frac{\alpha}{2}}) f(u), \quad x = x_l, \quad (4.5)$$

where s_0 is the Padé expansion point that is chosen in such a way that it is close to the frequency of the wave touching on the artificial boundaries.

The discretization of the problem (4.2)-(4.5) is given as

$${}_0^C \mathbb{D}_t^\alpha U_i^n = \delta_x^2 U_i^n + f(U_i^{n-1}), \quad 1 \leq i \leq M-1, \quad 1 \leq n \leq N_T, \quad (4.6)$$

$$(\tilde{\delta}_x + 3s_0^{\frac{\alpha}{2}}) {}_0^C \mathbb{D}_t^\alpha U_{M-1}^n + (3s_0^\alpha \tilde{\delta}_x + s_0^{\frac{3\alpha}{2}}) U_{M-1}^n = (\tilde{\delta}_x + 3s_0^{\frac{\alpha}{2}}) f(U_{M-1}^{n-1}), \quad 1 \leq n \leq N_T, \quad (4.7)$$

$$(\tilde{\delta}_x - 3s_0^{\frac{\alpha}{2}}) {}_0^C \mathbb{D}_t^\alpha U_1^{n-1} + (3s_0^\alpha \tilde{\delta}_x - s_0^{\frac{3\alpha}{2}}) U_1^{n-1} = (\tilde{\delta}_x - 3s_0^{\frac{\alpha}{2}}) f(U_1^{n-1}), \quad 1 \leq n \leq N_T. \quad (4.8)$$

Under the assumption that $f(u) \in C^2([0,T])$, it has been shown in [35] that the finite different scheme (4.6)-(4.8) has the convergence rate of $\mathcal{O}(h^2 + \Delta t)$ in L_∞ norm, where the L_∞ norm is defined by $\|e\|_\infty = \max_{0 \leq i \leq M} |e_i|$. Here we focus on the speed-up of the evaluation of the time fractional PDEs. Thus, we replace the approximation ${}_0^C \mathbb{D}_t^\alpha$ by our fast evaluation scheme ${}_0^{FC} \mathbb{D}_t^\alpha$, and have a new discretized scheme given by

$${}_0^{FC} \mathbb{D}_t^\alpha U_i^n = \delta_x^2 U_i^n + f(U_i^{n-1}), \quad 1 \leq i \leq M-1, \quad 1 \leq n \leq N_T, \quad (4.9)$$

$$(\tilde{\delta}_x + 3s_0^{\frac{\alpha}{2}}) {}_0^{FC} \mathbb{D}_t^\alpha U_{M-1}^n + (3s_0^\alpha \tilde{\delta}_x + s_0^{\frac{3\alpha}{2}}) U_{M-1}^n = (\tilde{\delta}_x + 3s_0^{\frac{\alpha}{2}}) f(U_{M-1}^{n-1}), \quad 1 \leq n \leq N_T, \quad (4.10)$$

$$(\tilde{\delta}_x - 3s_0^{\frac{\alpha}{2}}) {}_0^{FC} \mathbb{D}_t^\alpha U_1^{n-1} + (3s_0^\alpha \tilde{\delta}_x - s_0^{\frac{3\alpha}{2}}) U_1^{n-1} = (\tilde{\delta}_x - 3s_0^{\frac{\alpha}{2}}) f(U_1^{n-1}), \quad 1 \leq n \leq N_T. \quad (4.11)$$

Using the similar process of numerical analysis as those in [35], we can obtain the convergence rate of $\mathcal{O}(h^2 + \Delta t + \varepsilon)$ in L_∞ norm for the fast scheme (4.9)-(4.11). In fact, one only needs to replace the error estimate of ${}_0^C \mathbb{D}_t^\alpha$ in Lemma 2.5 by the one in Lemma 2.6.

4.1 Numerical examples

We will give two examples – the Fisher equation and the Huxley equation to illustrate the performance of our scheme. For both examples, in order to investigate the convergence orders of our scheme, the reference solution is computed over a large interval $\Omega_r = [-12, 12]$ with very small mesh sizes $h = 2^{-10}$, and $\Delta t = 2^{-14}$. We then set $\Omega_i = [-6, 6]$

and the precision for the sum-of-exponentials approximation of the convolution kernel to $\varepsilon=10^{-9}$ and $T=1$. The temporal step size is fixed at $\Delta t=2^{-14}$ when testing the order of convergence in space; and the spatial step size is fixed at $h=2^{-10}$ when testing the order of convergence in time.

Example 4.1. We consider the time fractional Fisher equation (i.e. $f(u) = -u(1-u)$ in (4.1)) with the double Gaussian initial value

$$u(x,0) = \exp(-10(x-0.5)^2) + \exp(-10(x+0.5)^2). \tag{4.12}$$

Tables 5 and 6 show again the same accuracy and convergence order $\mathcal{O}(h^2 + \Delta t)$ in L_∞ norm for both schemes (4.6)-(4.8) and (4.9)-(4.11) with $\alpha=0.2, 0.5$. We plot out in Fig. 4 the evolution of the difference of two solutions by taking $\varepsilon= 1e-7, 1e-8, 1e-9, 1e-10$, respectively. We observe that the difference decreases as ε decreases. Fig. 3 shows the CPU time in seconds for both schemes versus N_T . Again, one can see that the fast scheme has almost linear complexity in N_T and is much faster than the direct scheme.

Table 5: Errors and convergence orders in time for the fractional Fisher equation by fixing $h=2^{-10}$.

Δt	$\alpha=0.2$				$\alpha=0.5$			
	Fast scheme		Direct scheme		Fast scheme		Direct scheme	
	$\ e^n\ _\infty$	r_t	$\ e^n\ _\infty$	r_t	$\ e^n\ _\infty$	r_t	$\ e^n\ _\infty$	r_t
1/10	8.601e-04	1.02	8.601e-04	1.02	2.194e-03	1.09	2.194e-03	1.09
1/20	4.243e-04	1.01	4.243e-04	1.01	1.030e-03	1.06	1.030e-03	1.06
1/40	2.104e-04	1.01	2.104e-04	1.01	4.934e-04	1.04	4.934e-04	1.04
1/80	1.046e-04		1.046e-04		2.392e-04		2.392e-04	

Table 6: Errors, convergence orders in space, and CPU time for fractional Fisher equation by fixing $\Delta t=2^{-14}$.

h	$\alpha=0.2$				$\alpha=0.5$			
	Fast scheme		Direct scheme		Fast scheme		Direct scheme	
	$\ e^n\ _\infty$	r_s	$\ e^n\ _\infty$	r_s	$\ e^n\ _\infty$	r_s	$\ e^n\ _\infty$	r_s
1/80	8.196e-04	1.96	8.196e-04	1.96	6.045e-04	1.97	6.045e-04	1.97
1/160	2.112e-04	2.01	2.112e-04	2.01	1.547e-04	2.01	1.547e-04	2.01
1/320	5.246e-05	2.00	5.246e-05	2.00	3.842e-05	2.00	3.842e-05	2.00
1/640	1.316e-05		1.316e-05		9.649e-06		9.649e-06	
CPU(s)	38.80		1071.91		40.22		850.58	

Example 4.2. We consider the fractional Huxley equation (i.e. $f(u) = -0.1u(1-u)(u - 0.001)$ in (4.1)) with the double Gaussian initial value, see Eq. (4.12). Again, Tables 7 and 8 present the numerical results for $\alpha=0.2, 0.5$, which show that our fast scheme (4.9)-(4.11) has the same convergence order $\mathcal{O}(h^2 + \Delta t)$ in L_∞ norm as the direct scheme (4.6)-(4.8),

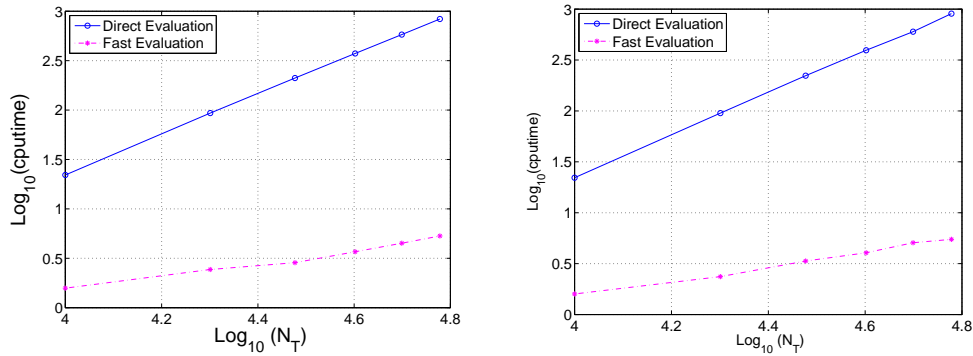


Figure 3: The log-log (in base 10) plot of the CPU time (in seconds) versus the total number of time steps N_T for two schemes. Here $N_S=80$ and $\alpha=0.5$. The left panel shows the results for the Fisher equation, and the right panel shows the results for the Huxley equation.

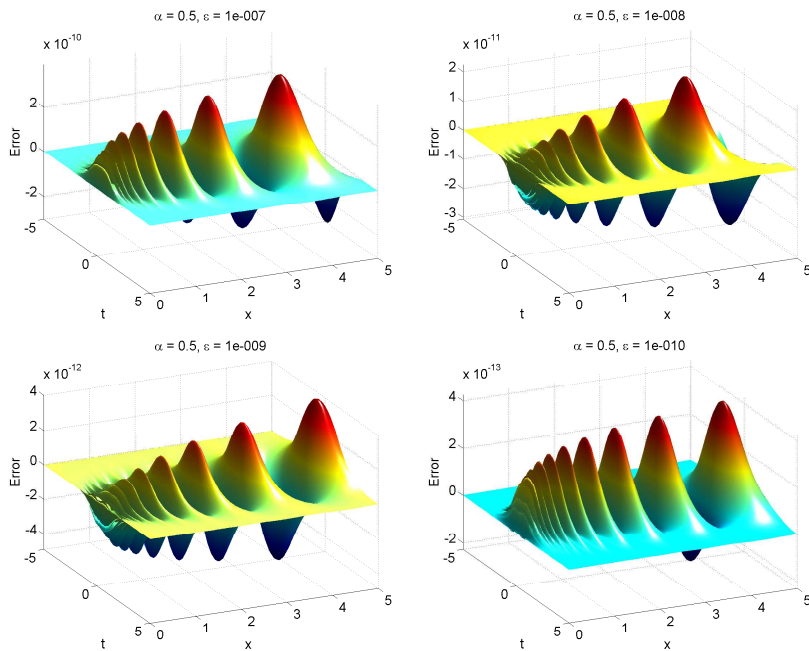


Figure 4: Example 4.1 (Fisher equation): The evolution of the difference (i.e. $Error = F U_h^n - D U_h^n$) between the solution to the scheme (4.9)-(4.11) (denote by $D U_h^n$) and the solution to the scheme (4.6)-(4.8) (denote by $F U_h^n$) for $\epsilon = 1e-7, 1e-8, 1e-9, 1e-10$, respectively. In the calculation, we take $T=5, [x_l, x_r] = [-5, 5], M=160, N=2000$ and $\alpha=0.5$.

but takes much less computational time in Fig. 3. The dependence of ϵ is plotted in Fig. 5 to show the difference of the numerical solutions to both schemes. Again, the above numerical tests demonstrate our theoretical analysis.

Table 7: Errors and convergence orders in time for fractional Huxley equation with fixed spatial step size $h=2^{-10}$.

Δt	$\alpha=0.2$				$\alpha=0.5$			
	Fast scheme		Direct scheme		Fast scheme		Direct scheme	
	$\ e^n\ _\infty$	r_t	$\ e^n\ _\infty$	r_t	$\ e^n\ _\infty$	r_t	$\ e^n\ _\infty$	r_t
1/10	1.089e-03	1.04	1.089e-03	1.04	2.961e-03	1.06	2.961e-03	1.06
1/20	5.304e-04	1.02	5.304e-04	1.02	1.418e-03	1.04	1.418e-03	1.04
1/40	2.614e-04	1.01	2.614e-04	1.01	6.896e-04	1.03	6.896e-04	1.03
1/80	1.294e-04		1.294e-04		3.379e-04		3.379e-04	

Table 8: Errors, convergence orders in space, and CPU time for fractional Huxley equation by fixing $\Delta t=2^{-14}$.

h	$\alpha=0.2$				$\alpha=0.5$			
	Fast scheme		Direct scheme		Fast scheme		Direct scheme	
	$\ e^n\ _\infty$	r_s	$\ e^n\ _\infty$	r_s	$\ e^n\ _\infty$	r_s	$\ e^n\ _\infty$	r_s
1/80	9.051e-04	1.96	9.051e-04	1.96	6.882e-04	1.98	6.882e-04	1.98
1/160	2.319e-04	2.01	2.319e-04	2.01	1.746e-04	2.01	1.746e-04	2.01
1/320	5.762e-05	2.00	5.762e-05	2.00	4.344e-05	2.00	4.344e-05	2.00
1/640	1.447e-05		1.447e-05		1.091e-05		1.091e-05	
CPU(s)	38.69		1106.58		40.14		855.87	

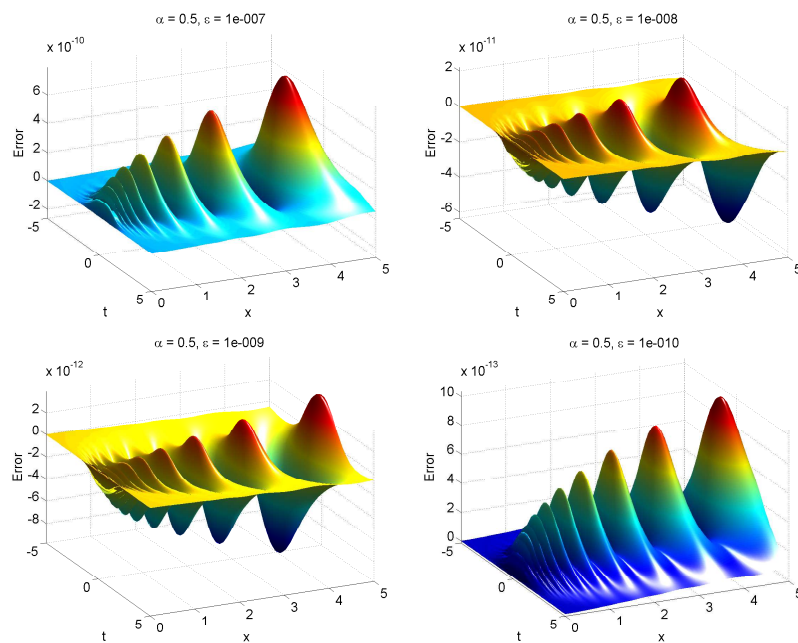


Figure 5: Example 4.2 (Huxley equation): The evolution of the difference (i.e. $Error = F U_h^n - D U_h^n$) between the solution to the scheme (4.9)-(4.11) (denote by $D U_h^n$) and the solution to the scheme (4.6)-(4.8) (denote by $F U_h^n$) for $\epsilon = 1e-7, 1e-8, 1e-9, 1e-10$, respectively. In the calculation, we take $T=5, [x_l, x_r] = [-5, 5], M=160, N=2000$ and $\alpha=0.5$.

5 Conclusions

We have presented an efficient algorithm for solving fractional partial differential equations. The algorithm combines the fast evaluation scheme for Caputo fractional derivative ${}_0^C D_t^\alpha f(t)$ and standard finite difference scheme. We have proved a prior estimate and error bound about the solution of our new FD scheme which leads to the stability of the new scheme. The numerical results on linear and nonlinear fraction PDEs show that our new scheme has the same order of convergence as the existing standard FD schemes, but with nearly optimal complexity in CPU time and storage.

Our work can be extended along several directions. First, it is straightforward to design high order schemes for the evaluation of fractional derivatives. Second, one may develop fast high-order algorithms for solving fractional PDEs which contains fractional derivatives in both time and space when the current scheme is combined with other existing schemes [11–13, 25]. Third, the scheme can be easily incorporated with other discretization schemes for the spatial variable such as finite element methods. Fourth, efficient and stable artificial boundary conditions can be designed using similar techniques in [24] for solving fractional PDEs in high dimensions. These issues are currently under investigation and the results will be reported on a later date.

Acknowledgments

Shidong Jiang is supported by NSF under grant DMS-1418918. Jiwei Zhang is partially supported by NSFC under grants 91430216 and U1530401. Zhimin Zhang is partially supported by NSFC grants 11471031, 91430216, U1530401, and by NSF under grant DMS-1419040. The authors would like to thank the referees for helpful comments.

A The proof of Lemma 2.7

Proof. We rewrite the definition of (2.27) by

$$\begin{aligned} {}_0^{FC} \mathbb{D}_t^\alpha u^n &= \frac{u(t_n) - u(t_{n-1})}{(1-\alpha)\Delta t_n^\alpha \Gamma(1-\alpha)} + \frac{1}{\Gamma(1-\alpha)} \left[\frac{u(t_{n-1})}{\Delta t_n^\alpha} - \frac{u(t_0)}{t_n^\alpha} - \alpha \sum_{i=1}^{N_{\text{exp}}} \omega_i U_{\text{hist},i}(t_n) \right] \\ &= \frac{\Delta t_n^{-\alpha}}{\Gamma(1-\alpha)} \left(\frac{u^n}{1-\alpha} - \left(\frac{\alpha}{1-\alpha} + a_0 \right) u^{n-1} - \sum_{l=1}^{n-2} (a_{n-l-1} + b_{n-l-2}) u^l - \left(b_{n-2} + \frac{1}{n^\alpha} \right) u^0 \right), \end{aligned} \quad (\text{A.1})$$

where

$$a_n = \alpha \Delta t^\alpha \sum_{j=1}^{N_{\text{exp}}} \omega_j e^{-ns_j \Delta t} \lambda_j^1, \quad b_n = \alpha \Delta t^\alpha \sum_{j=1}^{N_{\text{exp}}} \omega_j e^{-ns_j \Delta t} \lambda_j^2,$$

$$\lambda_j^1 = \frac{e^{-s_j \Delta t}}{s_j^2 \Delta t} (e^{-s_j \Delta t} - 1 + s_j \Delta t), \quad \lambda_j^2 = \frac{e^{-s_j \Delta t}}{s_j^2 \Delta t} (1 - e^{-s_j \Delta t} - e^{-s_j \Delta t} s_j \Delta t).$$

Applying the definition (A.1) and the Cauchy-Schwarz inequality, we have

$$\begin{aligned} ({}^{\text{FC}}\mathbb{D}_t^\alpha g^k)g^k &= \frac{1}{\Delta t^\alpha \Gamma(1-\alpha)} \left(\frac{1}{1-\alpha} (g^k)^2 - \left(\frac{\alpha}{1-\alpha} + a_0 \right) g^{k-1} g^k \right. \\ &\quad \left. - \sum_{l=1}^{k-2} (a_{k-l-1} + b_{k-l-2}) g^l g^k - \left(b_{k-2} + \frac{1}{k^\alpha} \right) g^0 g^k \right) \\ &\geq \frac{1}{\Delta t^\alpha \Gamma(1-\alpha)} \left[\left(\frac{2-\alpha}{2(1-\alpha)} - \frac{1}{2} \sum_{l=0}^{k-2} (a_l + b_l) - \frac{1}{2k^\alpha} \right) (g^k)^2 \right. \\ &\quad \left. - \frac{1}{2} \left(\frac{\alpha}{1-\alpha} + a_0 \right) (g^{k-1})^2 - \frac{1}{2} \sum_{l=1}^{k-2} (a_{k-l-1} + b_{k-l-2}) (g^l)^2 \right. \\ &\quad \left. - \frac{1}{2} \left(b_{k-2} + \frac{1}{k^\alpha} \right) (g^0)^2 \right]. \end{aligned} \tag{A.2}$$

Summing the above inequality from $k=1$ to n , we obtain

$$\begin{aligned} \Delta t \sum_{k=1}^n ({}^{\text{FC}}\mathbb{D}_t^\alpha g^k)g^k &\geq \frac{\Delta t^{1-\alpha}}{\Gamma(1-\alpha)} \left[\sum_{k=2}^n \left(\frac{1}{1-\alpha} - \frac{\alpha}{2(1-\alpha)} - \frac{1}{2} \sum_{l=0}^{k-2} (a_l + b_l) - \frac{1}{2k^\alpha} \right) (g^k)^2 \right. \\ &\quad \left. - \sum_{k=2}^n \left(\frac{\alpha}{2(1-\alpha)} + \frac{a_0}{2} \right) (g^{k-1})^2 - \frac{1}{2} \sum_{k=2}^n \sum_{l=1}^{k-2} (a_{k-l-1} + b_{k-l-2}) (g^l)^2 \right. \\ &\quad \left. + \frac{1}{2(1-\alpha)} (g^1)^2 - \frac{1}{2(1-\alpha)} (g^0)^2 - \frac{1}{2} \sum_{k=2}^n \left(b_{k-2} + \frac{1}{k^\alpha} \right) (g^0)^2 \right] \\ &= \frac{\Delta t^{1-\alpha}}{\Gamma(1-\alpha)} \sum_{k=1}^n \left(C_k (g^k)^2 - C_0 (g^0)^2 \right), \end{aligned} \tag{A.3}$$

where the coefficients C_k ($k=0, 1, \dots, n$) are given by the formula

$$C_k^n = \begin{cases} \frac{1}{2(1-\alpha)} + \frac{1}{2} \sum_{k=2}^n \left(b_{k-2} + \frac{1}{k^\alpha} \right), & k=0, \\ \frac{1}{2} - \frac{1}{2} \sum_{l=0}^{n-2} (a_l + b_l) + \frac{1}{2} b_{n-2}, & k=1, \\ 1 - \frac{1}{2} \sum_{l=0}^{k-2} (a_l + b_l) - \frac{1}{2k^\alpha} - \frac{1}{2} \sum_{l=0}^{n-k-1} (a_l + b_l) + \frac{1}{2} b_{n-k-1}, & 2 \leq k < n, \\ \frac{2-\alpha}{2(1-\alpha)} - \frac{1}{2} \sum_{l=0}^{k-2} (a_l + b_l) - \frac{1}{2k^\alpha}, & k=n. \end{cases} \tag{A.4}$$

From (2.4), we have the estimate

$$\frac{1}{t^{1+\alpha}} - \varepsilon \leq \sum_{j=1}^{N_{\text{exp}}} \omega_j e^{-s_j t} \leq \frac{1}{t^{1+\alpha}} + \varepsilon. \tag{A.5}$$

It is also straightforward to verify that

$$\sum_{l=0}^{k-2} (a_l + b_l) = \alpha \Delta t^\alpha \int_{\Delta t}^{k\Delta t} \sum_{j=1}^{N_{\text{exp}}} \omega_j e^{-s_j t} dt. \tag{A.6}$$

Combining (A.5) and (A.6), we obtain

$$\left(1 - \frac{1}{k^\alpha}\right) - \alpha \Delta t^\alpha t_{k-1} \varepsilon \leq \sum_{l=0}^{k-2} (a_l + b_l) \leq \left(1 - \frac{1}{k^\alpha}\right) + \alpha \Delta t^\alpha t_{k-1} \varepsilon. \tag{A.7}$$

Substituting (A.7) into (A.4) yields the following estimates

$$\left\{ \begin{array}{l} C_0^n \leq \frac{n^{1-\alpha}}{2(1-\alpha)} + \frac{\alpha \Delta t^\alpha t_{n-1} \varepsilon}{2}, \\ C_1^n \geq \frac{1}{2} - \frac{1}{2} \sum_{l=0}^{n-2} (a_l + b_l) \geq \frac{1}{2n^\alpha} - \alpha \Delta t^\alpha t_{n-1} \varepsilon, \\ C_k^n = 1 - \frac{1}{2} \sum_{l=0}^{k-2} (a_l + b_l) - \frac{1}{2k^\alpha} - \frac{1}{2} \sum_{l=0}^{n-k-1} (a_l + b_l) + \frac{1}{2} b_{n-k-1} \\ \quad \geq \frac{1}{2n^\alpha} - \alpha \Delta t^\alpha t_{n-1} \varepsilon, \quad 2 \leq k \leq n-1, \\ C_n^n \geq \frac{2-\alpha}{2(1-\alpha)} - \sum_{l=0}^{n-2} (a_l + b_l) - \frac{1}{2n^\alpha} \geq \frac{1}{2n^\alpha} - \alpha \Delta t^\alpha t_{n-1} \varepsilon. \end{array} \right. \tag{A.8}$$

Combining (A.3) and (A.8), we obtain Lemma 2.7. □

B The proof of Theorem 3.3

Proof. We observe that the error e_i^k satisfies the following FD scheme:

$${}_0^{\text{FC}}\mathcal{D}_t^\alpha e_i^k = \delta_x^2 e_i^k + T_i^k, \quad 1 \leq i \leq N_S - 1, \quad 1 \leq k \leq N_T, \tag{B.1}$$

$${}_0^{\text{FC}}\mathcal{D}_t^\alpha e_0^k = \frac{2}{h} \left[\delta_x e_{\frac{1}{2}}^k - {}_0^{\text{FC}}\mathcal{D}_t^{\alpha/2} e_0^k \right] + T_0^k, \tag{B.2}$$

$${}_0^{\text{FC}}\mathcal{D}_t^\alpha e_{N_S}^k = \frac{2}{h} \left[-\delta_x e_{N_S-\frac{1}{2}}^k - {}_0^{\text{FC}}\mathcal{D}_t^{\alpha/2} e_{N_S}^k \right] + T_{N_S}^k, \tag{B.3}$$

$$e_i^0 = 0, \quad 0 \leq i \leq N_S. \tag{B.4}$$

where the truncation terms T^k at the interior and boundary points are given by the formulas

$$\begin{aligned}
 T_i^k &= - \left[{}_0^C D_t^\alpha u(x_i, t_k) - {}_0^{FC} \mathbb{D}_t^\alpha U_i^k \right] + \left[u_{xx}(x_i, t_k) - \delta_x^2 U_i^k \right], \quad 1 \leq i \leq N_S - 1, \quad 1 \leq k \leq N_T, \\
 T_0^k &= \left\{ u_{xx}(x_0, t_k) - \frac{2}{h} \left[\delta_x U_{\frac{1}{2}}^k - u_x(x_0, t_k) \right] - \frac{2}{h} \left[{}_0^C D_t^{\frac{\alpha}{2}} u(x_0, t_k) - {}_0^{FC} \mathbb{D}_t^{\frac{\alpha}{2}} U_0^k \right] \right\} \\
 &\quad - \left[{}_0^C D_t^\alpha u(x_0, t_k) - {}_0^{FC} \mathbb{D}_t^\alpha U_0^k \right], \quad 1 \leq k \leq N_T, \\
 T_{N_S}^k &= \left\{ u_{xx}(x_{N_S}, t_k) - \frac{2}{h} \left[u_x(x_{N_S}, t_k) - \delta_x U_{N_S - \frac{1}{2}}^k \right] - \frac{2}{h} \left[{}_0^C D_t^{\frac{\alpha}{2}} u(x_{N_S}, t_k) - {}_0^{FC} \mathbb{D}_t^{\frac{\alpha}{2}} U_{N_S}^k \right] \right\} \\
 &\quad - \left[{}_0^C D_t^\alpha u(x_{N_S}, t_k) - {}_0^{FC} \mathbb{D}_t^\alpha U_{N_S}^k \right], \quad 1 \leq k \leq N_T.
 \end{aligned}$$

Using Lemma 3.1 and Taylor expansion, it is easy to show that the truncation terms T^k satisfy the following error bounds

$$\left| T_i^k \right| \leq c_1 (\Delta t^{2-\alpha} + h^2 + \varepsilon), \quad 1 \leq i \leq N_S - 1, \quad 1 \leq k \leq N_T, \tag{B.5}$$

$$\left| T_0^k \right| \leq c_1 (\Delta t^{2-\alpha} + h + \frac{\Delta t^{2-\alpha/2}}{h} + \frac{\varepsilon}{h}), \quad 1 \leq k \leq N_T, \tag{B.6}$$

$$\left| T_{N_S}^k \right| \leq c_1 (\Delta t^{2-\alpha} + h + \frac{\Delta t^{2-\alpha/2}}{h} + \frac{\varepsilon}{h}), \quad 1 \leq k \leq N_T \tag{B.7}$$

with c_1 some positive constant. Thus, for $h \leq 1$ and $\Delta t \leq 1$, we have

$$\begin{aligned}
 &\frac{1}{4\nu} \left[(hT_0^k)^2 + (hT_{N_S}^k)^2 \right] + \frac{2}{\mu} h \sum_{i=1}^{N_S-1} (T_i^k)^2 \\
 &\leq \frac{c_1^2}{2\nu} \left(h\Delta t^{2-\alpha} + \Delta t^{2-\frac{\alpha}{2}} + h^2 + \varepsilon \right)^2 + \frac{2c_1^2 L}{\mu} (\Delta t^{2-\alpha} + h^2 + \varepsilon)^2 \\
 &\leq \frac{2c_1^2}{\nu} (\Delta t^{2-\alpha} + h^2 + \varepsilon)^2 + \frac{2c_1^2 L}{\mu} (\Delta t^{2-\alpha} + h^2 + \varepsilon)^2 \\
 &\leq 4c_1^2 \left(\frac{1}{\nu} + \frac{L}{\mu} \right) (\Delta t^{2-\alpha} + h^2) + 4c_1^2 \left(\frac{1}{\nu} + \frac{L}{\mu} \right) \varepsilon^2.
 \end{aligned} \tag{B.8}$$

A direct application of Theorem 3.1 to the system (B.1)-(B.4) produces

$$\Delta t \sum_{k=1}^n \|e^k\|_\infty^2 \leq \frac{\Delta t (1 + \sqrt{1 + L^2 \mu})}{L \mu} \sum_{k=1}^n \left(\frac{1}{4\nu} \left[(hT_0^k)^2 + (hT_{N_S}^k)^2 \right] + \frac{2}{\mu} h \sum_{i=1}^{N_S-1} (T_i^k)^2 \right). \tag{B.9}$$

Substituting (B.8) into (B.9), simplifying the resulting expressions, and taking the square root for both sides, we obtain (3.27). \square

References

- [1] B. Alpert, L. Greengard, and T. Hagstrom, Rapid evaluation of nonreflecting boundary kernels for time-domain wave propagation, *SIAM J. Numer. Anal.*, 37 (2000), 1138–1164.
- [2] A. A. Awotunde, R. A. Ghanam, and N. Tatar, Artificial boundary condition for a modified fractional diffusion problem, *Bound. Value Probl.*, 1 (2015), 1–17.
- [3] D. Baffet and Jan S. Hesthaven, A Laplace transform based kernel reduction scheme for fractional differential equations, <https://infoscience.epfl.ch/record/212955>, 2016.
- [4] H. Brunner, H. Han, and D. Yin, Artificial boundary conditions and finite difference approximations for a time-fractional diffusion-wave equation on a two-dimensional unbounded spatial domain, *J. Comput. Phys.*, 276 (2014), 541–562.
- [5] G. Beylkin and L. Monzón, On approximation of functions by exponential sums, *Appl. Comput. Harmon. Anal.*, 19 (2005), 17–48.
- [6] G. Beylkin and L. Monzón, Approximation by exponential sums revisited, *Appl. Comput. Harmon. Anal.*, 28 (2010), 131–149.
- [7] H. Brunner, H. Han, and D. Yin, The maximum principle for time-fractional diffusion equations and its application, *Numer. Funct. Anal. Optim.*, 36 (2015), 1307–1321.
- [8] H. Brunner, X. Wu and J. Zhang, Computational Solution of Blow-Up Problems for Semilinear Parabolic PDEs on Unbounded Domains. *SIAM J. Sci. Comput.*, 31(2010), 4478–4496.
- [9] J.X. Cao, C.P. Li and Y.Q. Chen, High-order approximation to Caputo derivatives and Caputo-type advection-diffusion equations (II), *Frac. Calc. and Appl. Anal.* 18.3 (2015), 735–761.
- [10] J. Cao and C. Xu, A high order scheme for the numerical solution of the fractional ordinary differential equations, *J. Comput. Phys.*, 238 (2013), 154–168.
- [11] C. Chen, F. Liu, V. Anh, and I. Turner, Numerical methods for solving a two-dimensional variable-order anomalous subdiffusion equation, *Math. Comp.*, 81 (2012), 345–366.
- [12] M. Cui, Compact finite difference method for the fractional diffusion equation, *J. Comput. Phys.*, 228 (2009), 7792–7804.
- [13] M. Cui, Compact alternating direction implicit method for two-dimensional time fractional diffusion equation, *J. Comput. Phys.*, 231 (2012), 2621–2633.
- [14] J. R. Dea, Absorbing boundary conditions for the fractional wave equation, *Appl. Math. Comput.*, 219 (2013), 9810–9820.
- [15] G. H. Gao, Z. Z. Sun, and Y. N. Zhang, A finite difference scheme for fractional sub-diffusion equations on an unbounded domain using artificial boundary conditions, *J. Comput. Phys.*, 231 (2012), 2865–2879.
- [16] G. H. Gao and Z. Z. Sun, The finite difference approximation for a class of fractional sub-diffusion equations on a space unbounded domain, *J. Comput. Phys.*, 236 (2013), 443–460.
- [17] G. H. Gao and Z. Z. Sun, A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications, *J. Comput. Phys.*, 259 (2014), 33–50.
- [18] R. Ghaffari and S. M. Hosseini, Obtaining artificial boundary conditions for fractional sub-diffusion equation on space two-dimensional unbounded domains, *Comput. Math. Appl.*, 68 (2014), 13–26.
- [19] B. L. Guo, Q. Xu and A. L. Zhu A second-order finite difference method for two-dimensional fractional percolation equations, *Commun. Comput. Phys.*, 19 (2016), 733-757.
- [20] L. Greengard and P. Lin, Spectral approximation of the free-space heat kernel, *Appl. Comput. Harmon. Anal.*, 9 (2000), 83–97.
- [21] L. Greengard and J. Strain, A fast algorithm for the evaluation of heat potentials, *Comm.*

- Pure Appl. Math., 43 (1990), 949–963.
- [22] H. Han and X. Wu, *Artificial Boundary Method*, Tsinghua Univ. Press, 2013.
 - [23] X. Huang and D. Yin, Artificial boundary conditions and finite difference approximations for a time-fractional diffusion-wave equation on high dimensional unbounded spatial domain, *J. Comput. Phys.*, to appear.
 - [24] S. Jiang and L. Greengard, Efficient representation of nonreflecting boundary conditions for the time-dependent Schrödinger equation in two dimensions, *Comm. Pure Appl. Math.*, 61 (2008), 261–288.
 - [25] S. Jiang, L. Greengard, and W. Bao, Fast and accurate evaluation of nonlocal Coulomb and dipole-dipole interactions via the nonuniform FFT, *SIAM J. Sci. Comput.*, 36.5 (2014), B777–B794.
 - [26] S. Jiang, L. Greengard, and S. Wang, Efficient sum-of-exponentials approximations for the heat kernel and their applications, *Adv. Comput. Math.*, 41.3 (2015), 529–551.
 - [27] R. Ke, M. Ng, and H. Sun, A fast direct method for block triangular Toeplitz-like with tridiagonal block systems from time-fractional partial differential equations, *J. Comput. Phys.*, 303 (2015), 203–211.
 - [28] A. Kilbas, H. Srivastava, and J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier Science and Technology, Boston, 2006.
 - [29] T. Langlands and B. Henry, Fractional chemotaxis diffusion equations, *Phys. Rev. E*, 81 (2010), 051102.
 - [30] T. Langlands and B. Henry, The accuracy and stability of an implicit solution method for the fractional diffusion equation, *J. Comput. Phys.*, 205 (2005), 719–736.
 - [31] S. Lei and H. Sun, A circulant preconditioner for fractional diffusion equations, *J. Comput. Phys.*, 242 (2013), 715–725.
 - [32] C. Li, W. Deng, and Y. Wu, Numerical analysis and physical simulations for the time fractional radial diffusion equation, *Comput. Math. Appl.*, 62 (2011), 1024–1037.
 - [33] C. Li, A. Chen, and J. Ye, Numerical approaches to fractional calculus and fractional ordinary differential equation, *J. Comput. Phys.*, 230 (2011), 3352–3368.
 - [34] C. Li, R. Wu and H. Ding, High-order approximation to Caputo derivatives and Caputo-type advection-diffusion equations, *Commun. Appl. Indu. Math.*, 6.2, e-536 (2014), 1-32.
 - [35] 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), 415–428.
 - [36] D. Li, C. Zhang, M. Ran, A linear finite difference scheme for generalized time fractional Burgers equation, *Appl. Math. Model.*, 40 (2016), 6069–6081.
 - [37] H. Li, J. Cao, C. Li, High-order approximation to Caputo derivatives and Caputo-type advection-diffusion equations (III), *J. Comput. Appl. Math.*, 299 (2016), 159–175.
 - [38] J. Li, A fast time stepping method for evaluating fractional integrals, *SIAM J. Sci. Comput.*, 31 (2010), 4696–4714.
 - [39] L.M. Li, D. Xu, Alternating direction implicit Galerkin finite element method for the two-dimensional time fractional evolution equation, *Numer. Math. Theor. Meth. Appl.*, 7 (2014), 41–57.
 - [40] Y. Lin and C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, *J. Comput. Phys.*, 225 (2007), 1533–1552.
 - [41] Y. Lin, X. Li, and C. Xu, Finite difference/spectral approximations for the fractional cable equation, *Math. Comp.*, 80 (2011), 1369–1396.
 - [42] Y. Lin and C. Xu, Finite difference/spectral approximations for time-fractional diffusion

- equation, *J. Comput. Phys.*, 225 (2007), 1533–1552.
- [43] X. Lu, H. Sun, and H. Pang, Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional subdiffusion equations, *J. Numer. Lin. Alg. Appl.*, 22 (2015), 866–882.
 - [44] C. Lubich and A. Schädle, Fast convolution for nonreflecting boundary conditions, *SIAM J. Sci. Comput.* 24 (2002) 161–182.
 - [45] W. McLean, Fast summation by interval clustering for an evolution equation with memory, *SIAM J. Numer. Anal.*, 34.6 (2012), 3039–3056.
 - [46] R. Metzler and J. Klafter, The random walks guide to anomalous diffusion: a fractional dynamics approach, *Phys. Rep.*, 339 (2000).
 - [47] Z. Odibat, Approximations of fractional integrals and Caputo fractional derivatives, *Appl. Math. Comput.*, 178 (2006), 527–533.
 - [48] K. B. Oldham and J. Spanier, *The Fractional Calculus*, Academic Press, New York, 1974.
 - [49] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST Handbook of Mathematical Functions*, Cambridge University Press, New York, NY, 2010.
 - [50] H. Pang and H. Sun, Multigrid method for fractional diffusion equations, *J. Comput. Phys.*, 231 (2012), 693–703.
 - [51] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, 1999.
 - [52] Z. Sun and X. Wu, A fully discrete difference scheme for a diffusion-wave system, *Appl. Numer. Math.*, 56 (2006), 193–209.
 - [53] H. Wang, K. Wang, and T. Sircar, A direct $\mathcal{O}(N \log^2 N)$ finite difference method for fractional diffusion equations, *J. Comput. Phys.*, 229 (2010), 8095–8104.
 - [54] H. Wang and S. B. Treena, A fast finite difference method for two-dimensional space-fractional diffusion equations, *SIAM J. Sci. Comput.*, 34 (2012), 2444–2458.
 - [55] K. Xu and S. Jiang, A bootstrap method for sum-of-poles approximations, *J. Sci. Comput.*, 55 (2013), 16–39.
 - [56] J. Zhang, Z. Xu, X. Wu, Unified approach to split absorbing boundary conditions for nonlinear Schrödinger equations, *Phys. Rev. E*, 78 (2008), 026709.
 - [57] J. Zhang, Z. Xu, X. Wu, Unified approach to split absorbing boundary conditions for nonlinear Schrödinger equations: Two dimensional case, *Phys. Rev. E*, 79 (2009), 046711.
 - [58] Q. Zhang, J. Zhang, S. Jiang, Z. Zhang, Numerical solution to a linearized time fractional KdV equation on unbounded domains, accepted by *Math. Comput.*, 2016.
 - [59] C. Zheng, Approximation, stability and fast evaluation of exact artificial boundary condition for one-dimensional heat equation, *J. Comput. Math.*, 25 (2007), 730–745.