Computers and Mathematics with Applications 108 (2022) 24–32

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

www.elsevier.com/locate/camwa

A fast numerical scheme for a variably distributed-order time-fractional diffusion equation and its analysis

Jinhong Jia [a], Hong Wang [b], Xiangcheng Zheng [c,*]

[a] School of Mathematics and Statistics, Shandong Normal University, Jinan, Shandong 250358, China
[b] Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA
[c] School of Mathematical Sciences, Peking University, Beijing 100871, China

ARTICLE INFO

Keywords:
Variably distributed-order time fractional diffusion equation
Finite element method
Error estimate
Divide and conquer

ABSTRACT

We develop a fast numerical method for a variably distributed-order time-fractional diffusion equation modeling the anomalous diffusion with uncertainties in inhomogeneous medium. Different from the analysis techniques of the commonly-used schemes for time-fractional problems like L1 methods, error estimates are proved based on a novel discretization coefficient splitting. The proposed method has the same accuracy as traditional schemes, while only $O(N^{3/2}\log N)$ computations and $O(N\log N)$ storage are required for generating and storing temporal discretization coefficients, where N refers to the number of time steps. We further design a corresponding fast divide and conquer algorithm to reduce the computational complexity of solving the linear system from $O(MN^2)$ of the time-stepping method to $O(MN\log^3 N)$ with M being the number of spatial nodes. Numerical experiments are presented to substantiate the theoretical results.

1. Introduction

Field tests showed that distributed-order fractional partial differential equations could adequately describe the complex processes with memory or hereditary effects, e.g., the anomalous diffusive transport in heterogenous porous media, and thus attracted extensive research interests [3,6,7,10,12,16–19,22–25,42–44,47,49]. In this paper we consider the following mobile-immobile variably distributed-order time-fractional diffusion equation proposed in [34,40,46]

$$\begin{aligned} &\partial_t u + k(t)\mathbb{D}_t^{\rho} u - K\Delta u = f(\boldsymbol{x},t), \quad (\boldsymbol{x},t)\in\Omega\times(0,T];\\ &u(\boldsymbol{x},0)=u_0(\boldsymbol{x}), \quad \boldsymbol{x}\in\Omega; \quad u(\boldsymbol{x},t)=0, \quad (\boldsymbol{x},t)\in\partial\Omega\times[0,T]. \end{aligned} \tag{1}$$

Here $\Omega\subset\mathbb{R}^d$ $(1\le d\le 3)$ is a simply connected bounded domain with the piecewise smooth boundary $\partial\Omega$ and convex corners, $\boldsymbol{x}\in\Omega$, $K>0$ is the diffusivity coefficient, $k(t)\ge 0$ is the variable mass partition coefficient between the mobile phase and the immobile phase, f is the source or sink term and u_0 is the prescribed initial data.

Model (1) is derived from the mobile-immobile time-fractional diffusion equation, in which the second left-hand side term of (1) is replaced by the standard (variable-order) Caputo fractional derivative, that has been investigated in several works [30,34,39,45,48]. Recently the commonly-used distributed-order fractional derivative $\mathcal{D}_t^{\rho} g(t) := \int_0^1 \rho(\alpha)\partial_t^{1-\alpha} g(t)d\alpha$ is invoked in the mobile-immobile time-fractional diffusion equation to obtain (1) with $\mathcal{D}_t^{\rho} u$ as the second left-hand side term [32,36]. The current problem (1) is proposed in [40] and generalizes the existing models by using the following variably distributed-order fractional derivative operator $\mathbb{D}_t^{\rho}$ defined by [2,27,40]

$$\mathbb{D}_t^{\rho} g(t) := \int_0^1 \rho(\alpha,t)\partial_t^{1-\alpha} g(t)d\alpha, \tag{2}$$

where $\partial_t^{1-\alpha}$ represents the Caputo fractional derivative operator defined via the convolution $\partial_t^{1-\alpha} g := \left(t^{\alpha-1}/\Gamma(\alpha)\right) * (\partial_t g)$ [4,21,29,31] and the probability density function $\rho(\alpha,t)$ satisfies $\int_0^1 \rho(\alpha,t)d\alpha = 1$ for $t\in[0,T]$. Concerning the standard distributed-order fractional derivative, (2) serves as its generalization by accommodating the time dependence of the probability density function to describe more complex dynamics [5,13,15,37].

A fully-discrete finite element method for model (1) was developed and analyzed in [40], in which the composite middle rectangular formula with mesh size $O(N^{-1/2})$ (N refers to the number of time steps) was used to discretize the distributed-order integral without loss of the first-order temporal accuracy and the resulting single-order fractional derivatives were approximated by the $L1$ scheme [8,9,14,26,41]. Such

* Corresponding author.
E-mail address: zhengxch@math.pku.edu.cn (X. Zheng).

https://doi.org/10.1016/j.camwa.2021.12.016
Received 27 July 2021; Received in revised form 4 October 2021; Accepted 27 December 2021
Available online 13 January 2022

discretization requires $O(N^{5/2})$ computational cost. Furthermore, the computational cost of solving the linear system of the time-stepping finite element scheme in [40] is $O(MN^2)$ with M being the number of degree of freedom of the finite element space, which is computationally expensive.

Concerning the aforementioned issues, we intend to improve the computational efficiency in two aspects. Different from the L1 methods, we first develop and analyze a novel fast approximated scheme for the variably distributed-order fractional derivative (2), which reduces the computational cost of generating the temporal discretization coefficients to $O(N^2\log N)$. Based on this discretization, we develop different numerical analysis techniques from [40] to carry out error estimates of the corresponding finite element method. We then propose a fast divide and conquer (DAC) method [20,28] of the finite element scheme, which further reduces the computational cost and the storage of the coefficients to $O(N^{3/2}\log N)$ and $O(N\log N)$, respectively, and the computational cost of solving the linear system to $O(MN\log^3 N)$ by the all-at-once manner.

The rest of this paper is organized as follows. In Section 2 we go over the existing theoretical and numerical results for model (1). In Section 3 we develop and analyze a fast approximated scheme of the variably distributed-order fractional derivative, based on which we prove the error estimates for the corresponding finite element method in Section 4. In Section 5 we develop a fast DAC algorithm to solve the linear system of the finite element scheme. Several numerical experiments are performed to substantiate the theoretical findings in the last section.

2. Preliminaries

We introduce notations and norms, and refer the well-posedness and regularity of model (1) as well as its numerical methods from the literature.

2.1. Well-posedness and regularity

For a given interval $\mathcal{I}$, let $C^m(\mathcal{I})$ with $m\in\mathbb{N}$ be the space of continuously differentiable functions of order m on $\mathcal{I}$. For $r\geq 0$, let $H^r(\Omega)$ be the fractional Sobolev space of order r on Ω, $H_0^r(\Omega)$ be the subspace of $H^r(\Omega)$ enforced with proper zero boundary conditions, and $C^m(\mathcal{I};\mathcal{X})$, with $\mathcal{X}$ being a Banach space, be the space of continuously differentiable functions of order m on $\mathcal{I}$ with respect to the norm $\|\cdot\|_{\mathcal{X}}$. All these spaces are equipped with standard norms [1].

Let $\{\lambda_i,\varphi_i\}_{i=1}^\infty$ be the eigenpairs of $-K\Delta$, and $\check{H}^s(\Omega) := \big\{v\in L^2(\Omega) : |v|^2_{\check{H}^s(\Omega)} := \sum_{i=1}^\infty \lambda_i^s(v,\varphi_i)^2 < \infty\big\}$ be the subspace of $H^s(\Omega)$ with $\|v\|_{\check{H}^s(\Omega)} := (\|v\|^2_{L^2(\Omega)} + |v|^2_{\check{H}^s(\Omega)})^{1/2}$ such that $\check{H}^0(\Omega)=L^2(\Omega)$ and $\check{H}^2(\Omega)=H^2(\Omega)\cap H_0^1(\Omega)$ [33,38].

We then make the following assumptions on $\rho(\alpha,t)$:

Assumption A. supp $\rho\subset[\underline{\alpha},\overline{\alpha}]$ for constants $0<\underline{\alpha}<\overline{\alpha}\leq 1$ and $t\in[0,T]$, and $\rho,\rho_t\in C([0,1]\times[0,T])$.

In the rest of the paper we use Q to denote a generic positive constant that may assume different values at different occurrences. We may drop the subscript L^2 in $(\cdot,\cdot)_{L^2}$ and $\|\cdot\|_{L^2}$, and the space-time domains in the Sobolev spaces and norms, e.g., abbreviate $H^1(\mathcal{I};H^2(\Omega))$ as $H^1(H^2)$, when no confusion occurs.

We cite the well-posedness and regularity of (1) from [40] for future use.

Lemma 2.1. *Suppose $k\in C[0,T]$, $u_0\in\check{H}^{\gamma+2}$ and $f\in H^1(\check{H}^\gamma)$ for $\gamma>d/2$, and the Assumption A holds. Then problem (1) has a unique solution $u\in C^1([0,T];\check{H}^\gamma)$ with the following stability estimate*

$$\|u\|_{C^1([0,T];\check{H}^s)} \leq Q(\|u_0\|_{\check{H}^{2+s}} + \|f\|_{H^1(\check{H}^s)}),\quad 0\leq s\leq\gamma.$$

where $Q>0$ is a constant depending on $\underline{\alpha}$, $\|k\|_{C([0,T])}$ and T.

Lemma 2.2. *Suppose $k\in C^1[0,T]$, $u_0\in\check{H}^{4+s}$, $f\in H^1(\check{H}^{2+s})\bigcap H^2(\check{H}^s)$ for $s\geq 0$, and the Assumption A holds. Then $u\in C^2((0,T];\check{H}^s)$ and the following estimate holds for $0<t\ll 1$,*

$$\|u\|_{C^2([t,T];\check{H}^s)} \leq Qt^{\underline{\alpha}-1}(\|u_0\|_{\check{H}^{4+s}} + \|f\|_{H^1(\check{H}^{2+s})} + \|f\|_{H^2(\check{H}^s)}),$$

where $Q>0$ is a constant depending on $\underline{\alpha}$, $\|k\|_{C^1[0,T]}$ and T.

2.2. L1 finite element method: a brief review

Let $0<N,L\in\mathbb{N}$ and we define uniform partitions on $[0,T]$ and $[\underline{\alpha},\overline{\alpha}]$ by $t_n := n\tau$ for $n=0,1,\cdots,N$ with $\tau := T/N$, and $\alpha_l := l\sigma$ for $0\leq l\leq L$ with $\sigma := (\overline{\alpha}-\underline{\alpha})/L$. We also define a quasi-uniform partition on Ω with the mesh diameter h, and let S_h be the space of continuous and piecewise linear functions on Ω with respect to the partition. The Ritz projection operator $\Pi_h : H_0^1(\Omega)\to S_h$ is given by

$$\big(K\nabla\Pi_h g,\nabla\chi_h\big) = \big(K\nabla g,\nabla\chi_h\big),\quad \forall\chi_h\in S_h,$$

which has the following approximation property [1,11]

$$\|g-\Pi_h g\| \leq Qh^2\|g\|_{H^2(\Omega)},\quad \forall g\in H^2(\Omega)\cap H_0^1(\Omega). \tag{3}$$

The discrete Laplace operator $\mathcal{L}_h : S_h\to S_h$ is defined for any $\zeta\in S_h$ by

$$(\mathcal{L}_h\zeta,\chi_h) = (K\nabla\zeta,\nabla\chi_h),\quad \forall\chi_h\in S_h.$$

Denote $u_n := u(\boldsymbol{x},t_n)$, $k_n := k(t_n)$ and $f_n := f(\boldsymbol{x},t_n)$. We use the backward Euler scheme and the $L1$ scheme to discrete $\partial_t u$ and $\partial_t^{1-\alpha}u$, respectively, at $t=t_n$ with $\alpha\in[\underline{\alpha},\overline{\alpha}]$ by

$$\begin{aligned}
\partial_t u_n &= \frac{u_n-u_{n-1}}{\tau} + \frac{1}{\tau}\int_{t_{n-1}}^{t_n}\partial_{tt}u(t-t_{n-1})dt := \delta_\tau u_n + E_n,\\
\partial_{t_n}^{1-\alpha}u_n &= \sum_{k=1}^n b_{n,k}(u_k-u_{k-1}) + \frac{1}{\Gamma(\alpha)}\sum_{k=1}^n\int_{t_{k-1}}^{t_k}\frac{\partial_s u-\delta_\tau u_k}{(t_n-s)^{1-\alpha}}ds,
\end{aligned} \tag{4}$$

where the coefficients $b_{n,k}$ for $1\leq k\leq n$ are given by

$$b_{n,k} := \frac{(t_n-t_{k-1})^\alpha-(t_n-t_k)^\alpha}{\tau\Gamma(1+\alpha)} = \frac{(n-k+1)^\alpha-(n-k)^\alpha}{\tau^{1-\alpha}\Gamma(1+\alpha)}. \tag{5}$$

We use the composite middle rectangular formula to approximate the integral with respect to α in $\mathbb{D}_{t_n}^\rho u$ as

$$\mathbb{D}_{t_n}^\rho u = \sigma\sum_{l=1}^L \rho(\alpha_{l-1/2},t_n)\partial_t^{1-\alpha_{l-1/2}}u + \hat{F}_n, \tag{6}$$

where the corresponding truncation error could be expressed as

$$\hat{F}_n : = \sum_{l=1}^L\int_{\alpha_{l-1}}^{\alpha_l}\int_{\alpha_{l-1/2}}^{\alpha}\int_{\alpha_{l-1/2}}^{\xi}\partial_{\eta\eta}\big(\partial_{t_n}^{1-\eta}u\big)d\eta d\xi d\alpha.$$

We apply the first right-hand side term of the second equation of (4) to approximate $\partial_{t_n}^{1-\alpha}u_n$ in (6) to obtain its discretization at $t=t_n$ as

$$\begin{aligned}
\mathbb{D}_{t_n}^\rho u_n &\approx \sigma\sum_{l=1}^L\rho(\alpha_{l-1/2},t_n)\sum_{k=1}^n b_{n,k}^l(u_k-u_{k-1}) + \hat{F}_n\\
&:= \sigma\sum_{k=1}^n \hat{a}_{n,k}(u_k-u_{k-1}) := \hat{\delta}_{t_n}^\rho u_n + \hat{F}_n,
\end{aligned} \tag{7}$$

where $b_{n,k}^l$ is given in (5) with $\alpha=\alpha_{l-1/2}$ and

$$\hat{a}_{n,k} = \sigma\sum_{l=1}^L\rho(\alpha_{l-1/2},t_n)b_{n,k}^l,\quad 1\leq k\leq n.$$

We apply the preceding discretizations to obtain an L1 finite element scheme of (1): find $\hat{U}_n\in S_h$ for $n=1,2,\cdots,N$ such that $\forall\chi_h\in S_h$

$$(\delta_\tau \hat{U}_n + k(t_n)\hat{\delta}^{\rho}_{t_n}\hat{U}_k, \chi_h) + (K\nabla \hat{U}_n, \nabla \chi_h) = (f, \chi_h). \tag{8}$$

Error estimates of this scheme are given in the following theorem.

Theorem 2.1. *Suppose $u_0 \in \check{H}^4$, $f \in H^1(\check{H}^2)\bigcap H^2(L^2)$, the Assumption A holds, and $\max_{t\in[0,T]}\|\rho(\cdot,t)\|_{W^{2,1}(\underline{\alpha},\overline{\alpha})} < \infty$. Then the following error estimate holds*

$$\|u-\hat{U}\|_{\hat{L}^\infty(L^2)} := \max_{1\le n\le N}\|u_n - \hat{U}_n\|_{L^2}$$
$$\le Q(\|u_0\|_{\check{H}^4} + \|f\|_{H^1(\check{H}^2)} + \|f\|_{H^2(L^2)})(\sigma^2 + \tau + h^2).$$

Proof. We could follow [40] to prove this theorem. The only difference lies in the estimate of $\hat{F}_n$. By assumptions on ρ, $\hat{F}_n$ is bounded by

$$\begin{aligned}
\|\hat{F}_n\| &\le \sum_{l=1}^{L}\int_{\alpha_{l-1}}^{\alpha_l}\left\|\int_{\alpha_{l-1/2}}^{\alpha}\int_{\alpha_{l-1/2}}^{\xi}\partial_{\eta\eta}(\partial_t^{1-\eta}u)d\eta d\xi\right\|d\alpha\\
&\le Q\sigma^2\sum_{l=1}^{L}\int_{\alpha_{l-1}}^{\alpha_l}\left[\left|\left(\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\right)''\int_0^{t_n}\frac{\partial_s u ds}{(t_n-s)^{1-\alpha}}ds\right|\right.\\
&\quad +2\left|\left(\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\right)'\int_0^{t_n}\frac{\partial_s u\ln(t_n-s)ds}{(t_n-s)^{1-\alpha}}\right|\\
&\quad \left.+\left|\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\int_0^{t_n}\frac{\partial_s u\ln^2(t_n-s)ds}{(t_n-s)^{1-\alpha}}\right|\right]d\alpha\\
&\le Q\sigma^2\|u\|_{C^1([0,1];L^2)}\sum_{l=1}^{L}\int_{\alpha_{l-1}}^{\alpha_l}\left[\left|\left(\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\right)''\int_0^{t_n}\frac{ds}{(t_n-s)^{1-\alpha}}ds\right|\right.\\
&\quad \left.+2\left|\left(\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\right)'\int_0^{t_n}\frac{\ln(t_n-s)ds}{(t_n-s)^{1-\alpha}}\right|+\left|\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\int_0^{t_n}\frac{\ln^2(t_n-s)ds}{(t_n-s)^{1-\alpha}}\right|\right]d\alpha.
\end{aligned}$$

For each t_n, we apply $(t_n-s)^\varepsilon|\ln(t_n-s)| \le Q$ for $0<\varepsilon\ll 1$ to get

$$\int_0^{t_n}\frac{\ln(t_n-s)ds}{(t_n-s)^{1-\alpha}} = \int_0^{t_n}\frac{(t_n-s)^\varepsilon\ln(t_n-s)}{(t_n-s)^{1-\alpha+\varepsilon}} \le Q$$

and a similar estimate with $\ln(t_n-s)$ in this equation replaced by $\ln^2(t_n-s)$, based on which we bound $\hat{F}_n$ by $\|\hat{F}_n\|_{\hat{L}^\infty(L^2)} \le Q\|u\|_{C^1([0,T];L^2)}\sigma^2$. For the remaining part of the proof we refer [40] for the details. □

3. An efficient discretization for variably distributed-order fractional derivative

When implementing the scheme (8) in the previous section, we have to compute the coefficients $\{\hat{a}_{n,k}\}$ for $1\le k\le n\le N$, totally $O(LN^2)$ operations. To maintain the first-order accuracy in time, we should set $L=O(N^{1/2})$ (such that $\sigma^2 = \big(O(N^{-1/2})\big)^2 = O(N^{-1})$), which leads to $O(N^{5/2})$ computations and $O(N^2)$ storage for computing and storing these coefficients. To improve the efficiency without loss of accuracy, we develop a novel scheme by discretizing the variably distributed-order fractional derivative in a different approach.

3.1. Formulation of fast scheme

Given a function $z=z(t)$ on $t\in[0,T]$ with $z_n := z(t_n)$, we first discretize $\mathbb{D}^{\rho}_{t_n}z_n$ by replacing $\partial^{1-\alpha}_{t_n}z_n$ by (4)

$$\mathbb{D}^{\rho}_{t_n}z_n = \sum_{k=1}^{n}c_{n,k}(z_k - z_{k-1}) + r_n(z), \tag{9}$$

where the coefficients $c_{n,k}$ for $1\le k\le n$ and local truncation error r_n are given by

$$\begin{aligned}
c_{n,k} &:= \int_{\underline{\alpha}}^{\overline{\alpha}}\rho(\alpha,t_n)b_{n,k}d\alpha = \int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)}{\tau^{1-\alpha}\Gamma(1+\alpha)}\Big((n-k+1)^\alpha-(n-k)^\alpha\Big)d\alpha,\\
r_n(z) &:= \sum_{k=1}^{n}\int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\int_{t_{k-1}}^{t_k}\frac{z'(s)-\delta_\tau z_k}{(t_n-s)^{1-\alpha}}dsd\alpha.
\end{aligned} \tag{10}$$

Then we consider the approximation of $c_{n,k}$ for $1\le k\le n\le N$ by defining an auxiliary function

$$g_a^m(x) := \frac{(a+1)^x\ln^m(a+1) - a^x\ln^m a}{m!}, \quad x\in[0,1], \tag{11}$$

where $a, m\in\mathbb{N}$ are two nonnegative parameters. For a fixed positive integer S, we apply the Taylor expansion of the exponential function to expand $(n-k+1)^\alpha-(n-k)^\alpha$ for $n-k\ge 1$ at $\alpha=\underline{\alpha}$ as

$$\begin{aligned}
&(n-k+1)^\alpha-(n-k)^\alpha\\
&=\sum_{v=0}^{S}\frac{(\alpha-\underline{\alpha})^v}{v!}\left[(n-k+1)^{\underline{\alpha}}\ln^v(n-k+1)-(n-k)^{\underline{\alpha}}\ln^v(n-k)\right]\\
&\quad+\frac{(\alpha-\underline{\alpha})^{S+1}}{(S+1)!}\left[(n-k+1)^{\theta_{n,k}}\ln^{S+1}(n-k+1)-(n-k)^{\theta_{n,k}}\ln^{S+1}(n-k)\right]\\
&=\sum_{v=0}^{S}(\alpha-\underline{\alpha})^v g^v_{n-k}(\underline{\alpha}) + (\alpha-\underline{\alpha})^{S+1}g^{S+1}_{n-k}(\theta_{n,k}),
\end{aligned}$$

where $\theta_{n,k}\in[\underline{\alpha},\alpha]$ is a constant depending on $n-k$ and S. By dropping the local truncation error, we obtain an approximation of $c_{n,k}$ as

$$\tilde{c}_{n,k} = \begin{cases} c_{n,k}, & n-k\le S,\\ \displaystyle\sum_{v=0}^{S}\Big(\int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)(\alpha-\underline{\alpha})^v}{\tau^{1-\alpha}\Gamma(1+\alpha)}d\alpha\Big)g^v_{n-k}(\underline{\alpha}), & n-k\ge S+1,\end{cases} \tag{12}$$

with errors $H_{n,k} := c_{n,k}-\tilde{c}_{n,k}$ given by

$$H_{n,k} = \begin{cases} 0, & n-k\le S\\ \displaystyle\int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)(\alpha-\underline{\alpha})^{S+1}g^{S+1}_{n-k}(\theta_{n,k})}{\tau^{1-\alpha}\Gamma(1+\alpha)}d\alpha, & n-k\ge S+1.\end{cases} \tag{13}$$

By replacing $c_{n,k}$ with $\tilde{c}_{n,k}$ in (9), we obtain an approximation of $\mathbb{D}^{\rho}_t z$ at $t=t_n$ as

$$\begin{aligned}
\mathbb{D}^{\rho}_{t_n}z_n &= \sum_{k=1}^{n}\tilde{c}_{n,k}(z_k-z_{k-1}) + q_n(z) + r_n(z)\\
&:= \tilde{\delta}^{\rho}_{t_n}z_n + q_n(z) + r_n(z),
\end{aligned} \tag{14}$$

where $r_n(z)$ is given in (10) and $q_n(z)$ is defined by

$$q_n(z) = \sum_{k=1}^{n}H_{n,k}(z_k-z_{k-1}).$$

3.2. Estimates of local truncation errors

Lemma 3.1. *The coefficients $\{c_{n,k}\}_{k=1}^n$ in (10) satisfy*

$$\begin{cases}\displaystyle\int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)d\alpha}{\tau^{1-\alpha}\Gamma(1+\alpha)} = c_{n,n} > c_{n,n-1} > \cdots > c_{n,k} > \cdots > c_{n,1} > 0,\\ \displaystyle\int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)d\alpha}{\tau^{1-\alpha}(n-k+1)^{1-\alpha}\Gamma(\alpha)} \le c_{n,k} \le \int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)d\alpha}{\tau^{1-\alpha}(n-k)^{1-\alpha}\Gamma(\alpha)}, \ k<n.\end{cases}$$

Proof. These results follow from the properties of $b_{n,k}$ [35]

$$\begin{cases}\displaystyle\frac{1}{\tau^{1-\alpha}\Gamma(1+\alpha)} = b_{n,n} > b_{n,n-1} > \cdots > b_{n,k} > \cdots > b_{n,1} > 0,\\ \displaystyle\frac{1}{\tau^{1-\alpha}(n-k+1)^{1-\alpha}\Gamma(\alpha)} \le b_{n,k} \le \frac{1}{\tau^{1-\alpha}(n-k)^{1-\alpha}\Gamma(\alpha)}, \ k<n.\end{cases} \quad \square$$

Lemma 3.2. *Suppose that S in (12) satisfies*

Condition S : $S \geq e^{\mu}\ln N - 1$ for $\mu \geq 1$ satisfying $e^{\mu}\left(\mu - 1 - \ln(\overline{\alpha} - \underline{\alpha})\right) \geq 3 - \underline{\alpha}$.

Then there exists a constant $\hat{Q}$ such that

$$\sum_{k=1}^{n} |c_{n,k} - \tilde{c}_{n,k}| = \sum_{k=1}^{n} |H_{n,k}| \leq \hat{Q}N^{-1}, \quad 1 \leq n \leq N.$$

Remark 3.1. To roughly show the value of μ, let $\mu \geq 1$ satisfy a more constrained inequality than that in Condition S, that is, $e^{\mu}(\mu - 1) \geq 3$. Then we approximately solve this inequality to find that $\mu \geq 1.61$. Therefore it suffices to set $S = \lfloor e^{1.61}\ln N - 1\rfloor = O(\log N)$ to meet the Condition S. We will see in Theorem 4.2 that the Condition S is sufficient to ensure the first-order accuracy in time of the errors of the fast numerical scheme (18) using the discretization $\tilde{\delta}^{\rho}_{t_n}$ of the variably distributed-order derivative operator $\mathbb{D}^{\rho}_{t_n}$.

Proof. We estimate $H_{n,k}$ by analyzing the auxiliary function $g^m_a(x)$ defined in (11). It is clear that $g^{S+1}_{n-k}(\theta) > 0$ and

$$\frac{d}{d\theta}\left(g^{S+1}_{n-k}(\theta)\right) = \frac{(n-k+1)^{\theta}\ln^{S+2}(n-k+1) - (n-k)^{\theta}\ln^{S+1}(n-k)}{(S+1)!} > 0,$$

which means $g^{S+1}_{n-k}(\theta)$ is monotonically increasing for $\theta \in [0,1]$ and thus

$$g^{S+1}_{n-k}(\theta_{n,k}) \leq g^{S+1}_{n-k}(1) = \frac{(n-k+1)\ln^{S+1}(n-k+1) - (n-k)\ln^{S+1}(n-k)}{(S+1)!}.$$

Therefore we bound $H_{n,k}$ in (13) by

$$|H_{n,k}| \quad \leq \Big(\int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\rho(\alpha,t_n)(\alpha - \underline{\alpha})^{S+1}}{\tau^{1-\alpha}\Gamma(1+\alpha)}d\alpha\Big) g^{S+1}_{n-k}(1),$$

which leads to

$$\sum_{k=1}^{n}|H_{n,k}| \leq \int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\rho(\alpha,t_n)(\alpha-\underline{\alpha})^{S+1}}{\tau^{1-\alpha}\Gamma(1+\alpha)}d\alpha \sum_{k=1}^{n} g^{S+1}_{n-k}(1) \leq \frac{N^{2-\underline{\alpha}}(\overline{\alpha}-\underline{\alpha})^{S+1}\ln^{S+1}N}{(S+1)!}.$$

We use the Stirling's formula $(S+1)! \geq (S+1)^{S+3/2}e^{-(S+1)}$ to obtain

$$\frac{(\overline{\alpha}-\underline{\alpha})^{S+1}\ln^{S+1}N}{(S+1)!} \leq \frac{1}{\sqrt{S+1}}\left(\frac{e(\overline{\alpha}-\underline{\alpha})\ln N}{S+1}\right)^{S+1} \leq \left(\frac{e(\overline{\alpha}-\underline{\alpha})\ln N}{S+1}\right)^{S+1}.$$

In order to maintain the first-order accuracy in time, we require S to satisfy

$$\left(\frac{e(\overline{\alpha}-\underline{\alpha})\ln N}{S+1}\right)^{S+1} \leq N^{\underline{\alpha}-3} \Rightarrow (S+1)\ln\left(\frac{e(\overline{\alpha}-\underline{\alpha})\ln N}{S+1}\right) \leq -(3-\underline{\alpha})\ln N.$$

By assuming $S+1 = e^{\mu}\ln N$ for some $\mu \geq 1$, we have

$$e^{\mu}\ln\left(e^{1-\mu}(\overline{\alpha}-\underline{\alpha})\right) \leq -(3-\underline{\alpha}) \Rightarrow e^{\mu}\left((\mu-1)-\ln(\overline{\alpha}-\underline{\alpha})\right) \geq 3-\underline{\alpha}.$$

Thus we finish the proof of the lemma. □

Theorem 3.1. *Suppose that $z \in C^1[0,T]\bigcap C^2(0,T]$ with*

$$\|z\|_{C^1[0,1]} \leq Q_0, \quad \|z\|_{C^2[t,T]} \leq Q_0 t^{\underline{\alpha}-1}, \quad 0 < t \ll 1.$$

If S satisfies the Condition S, we have the following estimates

$$|r_n(z)| \leq QQ_0 n^{\underline{\alpha}-1}N^{-\underline{\alpha}}, \quad |q_n(z)| \leq QQ_0\hat{Q}N^{-1},$$

where the constant $\hat{Q}$ is given in Lemma 3.2.

Proof. The r_n can be bounded similar to Theorem 4.1 in [40], and for q_n we use the estimate of $H_{n,k}$ in Lemma 3.2 to get

$$|q_n(z)| \leq \sum_{k=1}^{n}|H_{n,k}||z_k - z_{k-1}| \leq 2\|z\|_{C[0,T]}\sum_{k=1}^{n}|H_{n,k}| \leq QQ_0\hat{Q}N^{-1}.$$

Thus we finish the proof. □

3.3. Fast coefficient generation and efficient storage

For implementation, the coefficients $\tilde{c}_{n,k}$ can be generated and stored by a fast algorithm. Denote

$$C^{\nu}_n := \int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\rho(\alpha,t_n)(\alpha-\underline{\alpha})^{\nu}}{\tau^{1-\alpha}\Gamma(1+\alpha)}d\alpha, \quad 1 \leq n \leq N, \quad 0 \leq \nu \leq S,$$

then $\tilde{c}_{n,k}$ in (12) for $n-k \geq S+1$ can be expressed as

$$\tilde{c}_{n,k} = \sum_{\nu=0}^{S} C^{\nu}_n g^{\nu}_{n-k}(\underline{\alpha}). \tag{15}$$

Thus we require to compute $c_{n,k}$ for $n-k \leq S$, C^{ν}_n for $S+1 \leq n \leq N, 0 \leq \nu \leq S$ and $g^{\nu}_{n-k}(\underline{\alpha})$ for $S+1 \leq n-k \leq N-1$.

As discussed in Remark 3.1, we set $S = O(\log N)$ to ensure the $O(\tau)$ temporal accuracy. We use the composite middle rectangle formula to compute $c_{n,k}$ for $n-k \leq S$ with the $O(\tau)$ accuracy by setting the mesh size $\sigma = O(\sqrt{\tau})$, which requires $O(N^{3/2}\log N)$ computations and $O(N\log N)$ storage, respectively.

Similarly, $O(N^{3/2}\log N)$ computations and $O(N\log N)$ memories are required in evaluating and storing C^{ν}_n for $S+1 \leq n \leq N$ and $0 \leq \nu \leq S$, respectively. Thus computing $\tilde{c}_{n,k}$ for $n-k \geq S+1$ totally requires $O(N^2\log N)$ operations by (15).

We summarize the above discussions in the following theorem.

Theorem 3.2. *Under the Condition S, the coefficients $\tilde{c}_{n,k}$ for $1 \leq k \leq n \leq N$ can be efficiently computed and stored in $O(N^2\log N)$ operations.*

4. Error estimate for fast finite element scheme

In Section 3 we developed an efficient discretization scheme $\tilde{\delta}^{1-\rho(\alpha,t_n)}_{\tau}$ for the variably distributed-order fractional derivative operator $\mathbb{D}^{\rho(\cdot,t)}_t$. In this section, we apply this to construct a fast finite element scheme for model (1) and prove its error estimates.

We apply (14) to construct a different discretization of $\mathbb{D}^{\rho}_t u$ from the $L1$ scheme as

$$\mathbb{D}^{\rho}_{t_n}u_n = \tilde{\delta}^{\rho}_{t_n}u_n + R_n + F_n, \tag{16}$$

where

$$\tilde{\delta}^{\rho}_{t_n}u_n = \sum_{k=1}^{n}\tilde{c}_{n,k}(u_k - u_{k-1}), \quad F_n := \sum_{k=1}^{n} H_{n,k}(u_k - u_{k-1}),$$

$$R_n := \sum_{k=1}^{n}\int_{\underline{\alpha}}^{\overline{\alpha}}\frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\int_{t_{k-1}}^{t_k}\frac{\partial_s u(\cdot,s) - \delta_\tau u_k}{(t_n - s)^{1-\alpha}}ds\,d\alpha,$$

and $\tilde{c}_{n,k}$ and $H_{n,k}$ are given in (12) and (13), respectively.

We plug the discretization of $\partial_t u$ in (4) and $\mathbb{D}^{\rho}_{t_n}u_n$ in (16) into (1), integrate the resulting equation multiplied by $\chi \in H^1_0(\Omega)$ on Ω to get a weak formulation of model (1): for any $\chi \in H^1_0(\Omega)$ and $1 \leq n \leq N$,

$$(\delta_\tau u_n + k_n\tilde{\delta}^{\rho}_{t_n}u_n, \chi) + (K\nabla u_n, \nabla\chi) = (f_n, \chi) - (k_n(R_n + F_n) + E_n, \chi). \tag{17}$$

We drop the local truncation errors on the right-hand side of (17) to obtain a fast finite element scheme of (1): find $\tilde{U}_n \in S_h$ with $\tilde{U}_0 = \Pi_h u_0$ for $1 \leq n \leq N$ such that for any $\chi_h \in S_h$ and $1 \leq n \leq N$,

$$(\delta_\tau\tilde{U}_n + k_n\tilde{\delta}^{\rho}_{t_n}\tilde{U}_n, \chi_h) + (K\nabla\tilde{U}_n, \nabla\chi_h) = (f_n, \chi_h). \tag{18}$$

Let $\eta = u - \Pi_h u$ and $u_n - U_n = u_n - \Pi_h u_n + \Pi_h u_n - U_n =: \eta_n + \xi_n$, we bound $\eta_n = (u - \Pi_h u)(\boldsymbol{x}, t_n)$ in the following theorem.

Theorem 4.1. *Suppose that the assumptions of Lemma 2.2 hold, then η can be estimated as follows*

$$\|\delta_\tau\eta\|_{\hat{L}^{\infty}(L^2)} + \|\tilde{\delta}^{\rho}_{t_n}\eta\|_{\hat{L}^{\infty}(L^2)} \leq Q(\|u_0\|_{\check{H}^4} + \|f\|_{H^1(\check{H}^2)})h^2,$$

where Q is a constant independent of τ, h and u.

Proof. We use Lemma 2.1 and (3) to get

$$\|\delta_\tau \eta\|_{\hat{L}^\infty(L^2)} = \max_{1\le n\le N} \frac{1}{\tau}\left\|(I-\Pi_h)\int_{t_{n-1}}^{t_n} \partial_t u dt\right\| \le Qh^2\|u\|_{C^1([0,T];H^2)}.$$

From (14), we express $\tilde{\delta}^\rho_{t_n}\eta_n$ by

$$\tilde{\delta}^\rho_{t_n}\eta_n = \sum_{k=1}^n \tilde{c}_{n,k}(\eta_k-\eta_{k-1}) = \sum_{k=1}^n c_{n,k}(\eta_k-\eta_{k-1}) + \sum_{k=1}^n H_{n,k}(\eta_k-\eta_{k-1}). \tag{19}$$

The first term on the right-hand side of (19) can be bounded by

$$\begin{aligned}\left\|\sum_{k=1}^n c_{n,k}(\eta_k-\eta_{k-1})\right\| &= \left\|\sum_{k=1}^n c_{n,k}(I-\Pi_h)\int_{t_{k-1}}^{t_k}\partial_s u(\cdot,s)ds\right\| \\ &\le Q\|u\|_{C^1([0,T];H^2)}\tau h^2\sum_{k=1}^n c_{n,k} \le Q\|u\|_{C^1([0,T];\check{H}^2)}h^2,\end{aligned}$$

where we use $\mathbb{D}^\rho_{t_n} t = \int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\rho(\alpha,t_n)t_n^\alpha}{\Gamma(\alpha+1)}d\alpha \le Q$ to obtain

$$\begin{aligned}\mathbb{D}^\rho_{t_n} t &= \int_{\underline{\alpha}}^{\overline{\alpha}} \rho(\alpha,t)\partial_t^{1-\alpha} t d\alpha\Big|_{t=t_n} = \int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\int_0^{t_n}\frac{ds}{(t_n-s)^{1-\alpha}}d\alpha \\ &= \sum_{k=1}^n\int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\rho(\alpha,t_n)}{\Gamma(\alpha)}\int_{t_{k-1}}^{t_k}\frac{ds}{(t_n-s)^{1-\alpha}}d\alpha \\ &= \sum_{k=1}^n\int_{\underline{\alpha}}^{\overline{\alpha}} \frac{\tau^\alpha\rho(\alpha,t_n)}{\Gamma(\alpha+1)}\Big[(n-k+1)^\alpha-(n-k)^\alpha\Big]d\alpha = \tau\sum_{k=1}^n c_{n,k}\le Q.\end{aligned}$$

Similarly, we use Lemma 3.2 to estimate the second term on the right-hand side of (19) by

$$\begin{aligned}&\left\|\sum_{k=1}^n H_{n,k}(\eta_k-\eta_{k-1})\right\| \le \left\|\sum_{k=1}^n H_{n,k}(I-\Pi_h)\int_{t_{k-1}}^{t_k}\partial_s u(\cdot,s)ds\right\| \\ &\le Q\|u\|_{C^1([0,T];\check{H}^2)}\tau h^2\sum_{k=1}^n|H_{n,k}| \le Q\hat{Q}\|u\|_{C^1([0,T];\check{H}^2)}\tau^2h^2.\end{aligned}$$

We insert the previous estimates into (19) to finish the proof. □

Before the error estimates of the fast scheme (18), it is worth mentioning that the coefficients $\{\tilde{c}_{n,k}\}$ in $\tilde{\delta}^\rho_{t_n}u_n$ in (14) do not enjoy the monotonicity like $\{b_{n,k}\}$ and $\{c_{n,k}\}$ (cf. Lemma 3.1), which is a key ingredient in error estimates of the L1 schemes of (variably) distributed-order tFDEs. Consequently, the techniques to prove Theorem 2.1 in [40] do not apply. To resolve this issue, we design the following decomposition for entries in $\{\tilde{c}_{n,k}\}$

$$\begin{aligned}|\tilde{c}_{n,k+1}-\tilde{c}_{n,k}| &= |c_{n,k+1}-c_{n,k}+\left(\tilde{c}_{n,k+1}-c_{n,k+1}\right)+\left(c_{n,k}-\tilde{c}_{n,k}\right)| \\ &\le c_{n,k+1}-c_{n,k}+|\tilde{c}_{n,k+1}-c_{n,k+1}|+|c_{n,k}-\tilde{c}_{n,k}|.\end{aligned} \tag{20}$$

By employing this technique we could circumvent the loss the monotonicity of the coefficients and prove error estimates of the fast scheme (18) in the following theorem.

Theorem 4.2. *Under the assumptions of Lemma 2.2 and the Condition S, the following estimate holds for (18)*

$$\|u-\tilde{U}\|_{\hat{L}^\infty(L^2)} \le Q(\|u_0\|_{\check{H}^4}+\|f\|_{H^1(\check{H}^2)}+\|f\|_{H^2(L^2)})(\tau+h^2).$$

Proof. We subtract (18) from (17) and set $\chi_h=\chi=\xi_n$ to get

$$(\delta_\tau\xi_n + k_n\tilde{\delta}^\rho_{t_n}\xi_n,\xi_n) + (K\nabla\xi_n,\nabla\xi_n) = -(G_n,\xi_n), \tag{21}$$

where $G_n := \delta_\tau\eta_n + k_n\tilde{\delta}^\rho_{t_n}\eta_n + k_n(R_n+F_n)+E_n$. We rewrite $\tilde{\delta}^\rho_{t_n}\xi_n$ as

$$\tilde{\delta}^\rho_{t_n}\xi_n = \tilde{c}_{n,n}\xi_n - \sum_{k=1}^{n-1}(\tilde{c}_{n,k+1}-\tilde{c}_{n,k})\xi_k - \tilde{c}_{n,1}\xi_0.$$

Correspondingly, (21) can be reformulated as

$$\begin{aligned}&\left(1+\tau k(t_n)\tilde{c}_{n,n}\right)\|\xi_n\|^2 + \tau(K\nabla\xi_n,\nabla\xi_n) \\ &\quad\le (\xi_{n-1},\xi_n) + \tau k_n\Big(\sum_{k=1}^{n-1}(\tilde{c}_{n,k+1}-\tilde{c}_{n,k})\xi_k,\xi_n\Big) + \tau(G_n,\xi_n).\end{aligned} \tag{22}$$

Apply (20) and the Cauchy inequality to cancel $\|\xi_n\|$ on both sides to obtain

$$\begin{aligned}\left(1+\tau k(t_n)\tilde{c}_{n,n}\right)\|\xi_n\| \le \|\xi_{n-1}\| + \tau k_n\sum_{k=1}^{n-1}\big(&c_{n,k+1}-c_{n,k}+|c_{n,k}-\tilde{c}_{n,k}| \\ &+|c_{n,k+1}-\tilde{c}_{n,k+1}|\big)\|\xi_k\| + \tau\|G_n\|.\end{aligned} \tag{23}$$

By (23), for $n=1$, we have

$$\|\xi_1\| \le \tau\|G_1\| \le \tau(1+2\hat{Q}\|k\|_{C[0,T]}\tau)\|G_1\|,$$

where $\hat{Q}$ is given in Lemma 3.2. We assume

$$\|\xi_m\| \le P_m\sum_{j=1}^m\|G_j\|,\quad P_m := \tau(1+2\hat{Q}\|k\|_{C[0,T]}\tau)^m,\quad 1\le m\le n-1, \tag{24}$$

plug (24) with $2\le m\le n-1$ into (23), and use the fact that $P_N > P_{N-1} > \cdots > P_1 > \tau$ and

$$\begin{aligned}&\sum_{k=1}^{n-1}(c_{n,k+1}-c_{n,k}+|c_{n,k+1}-\tilde{c}_{n,k+1}|+|\tilde{c}_{n,k}-c_{n,k}|) \\ &\quad\le \sum_{k=1}^{n-1}(c_{n,k+1}-c_{n,k}) + 2\sum_{k=1}^{n-2}|c_{n,k}-\tilde{c}_{n,k}| \le c_{n,n}+2\hat{Q} \text{ (using Lemma 3.2)}\end{aligned}$$

to arrive

$$\begin{aligned}(1+\tau k_nc_{n,n})\|\xi_n\| &\le P_{n-1}\sum_{j=1}^{n-1}\|G_j\| + \tau k_n\Big(P_{n-1}\sum_{j=1}^{n-1}\|G_j\|\Big) \\ &\quad\times\sum_{k=1}^{n-1}\left(c_{n,k+1}-c_{n,k}+|\tilde{c}_{n,k+1}-c_{n,k+1}|+|\tilde{c}_{n,k}-c_{n,k}|\right) + \tau\|G_n\| \\ &\le \Big(P_{n-1}\sum_{j=1}^n\|G_j\|\Big)(1+\tau k_nc_{n,n}+2\tau k_n\hat{Q}).\end{aligned}$$

Consequently we obtain

$$\begin{aligned}\|\xi_n\| &\le \Big(P_{n-1}\sum_{j=1}^n\|G_j\|\Big)\Big(1+\frac{2\tau k_n\hat{Q}}{1+\tau k_nc_{n,n}}\Big) \\ &\le \Big(P_{n-1}\sum_{j=1}^n\|G_j\|\Big)(1+2\hat{Q}\|k\|_{C[0,T]}\tau) = P_n\sum_{j=1}^n\|G_j\|,\end{aligned}$$

that is, (24) holds for $m=n$ and so for any $m\ge 2$ by mathematical induction. By using the fact that $(1+2\tau\|k\|_{C[0,T]}\hat{Q})^N\le Q$, we remain to bound $\tau\sum_{n=1}^N\|G_n\|$.

E_n could be bounded by

$$\begin{aligned}\|E_n\| &\le \frac{1}{\tau}\int_{t_{n-1}}^{t_n}\|\partial_{tt}u(\cdot,t_n)\||t-t_n|dt \\ &\le Q(\|u_0\|_{\check{H}^4}+\|f\|_{H^1(\check{H}^2)}+\|f\|_{H^2(L^2)})n^{\underline{\alpha}-1}N^{-\underline{\alpha}}.\end{aligned}$$

Similar to the proof of Theorem 3.1, we bound R_n by applying the regularities of $u(\boldsymbol{x},t)$ provided in Lemmas 2.1–2.2

$$\|R_n\| \le Q(\|u_0\|_{\check{H}^4}+\|f\|_{H^1(\check{H}^2)}+\|f\|_{H^2(L^2)})n^{\underline{\alpha}-1}N^{-\underline{\alpha}}.$$

We use Lemma 3.2 to estimate F_n by

$$\|F_n\| \le Q\tau\|u\|_{C^1([0,T];L^2)}\sum_{k=1}^n|H_{n,k}| \le Q\hat{Q}\|u\|_{C^1([0,T];L^2)}\tau^2.$$

Combining these estimates with Theorem 4.1, G_n with $1 \le n \le N$ can be bounded by

$$\|G_n\| \le Q(\|u_0\|_{\check{H}^4} + \|f\|_{H^1(\check{H}^2)} + \|f\|_{H^2(L^2)})(h^2 + n^{\alpha-1}N^{-\underline{\alpha}}).$$

Applying the fact that $\sum_{n=1}^{N} n^{\underline{\alpha}-1} \le QN^{\underline{\alpha}}$, we get

$$\tau \sum_{n=1}^{N} \|G_n\| \quad \le Q(\|u_0\|_{\check{H}^4} + \|f\|_{H^1(\check{H}^2)} + \|f\|_{H^2(L^2)})(\tau + h^2),$$

which completes the proof. $\square$

Remark 4.1. In the proof of Theorem 4.2, we use the exact form of C_n^v without discretization, and thus the result is independent from the mesh size σ of the quadrature. In the numerical simulation, we compute C_n^v via the composite middle rectangular formula with $O(\sigma^2)$ accuracy, and we set $\sigma = O(\tau^{1/2})$ in order to maintain the first-order accuracy in time.

5. A fast all-at-once DAC solver

Let $\{\psi_i(\boldsymbol{x})\}_{i=1}^M$, where M refers to the number of degree of freedom of the finite element space, be the basic functions of S_h satisfying $\psi_i(\boldsymbol{x}_i) = 1$ and $\psi_i(\boldsymbol{x}_j) = 0$ for $j \ne i$. The $M \times M$ matrices $\mathbf{B}_M$ and $\mathbf{B}_S$ denote the mass and stiffness matrices where

$$(\mathbf{B}_M)_{i,j} = \int_\Omega \psi_i(\boldsymbol{x}) \cdot \psi_j(\boldsymbol{x}) d\boldsymbol{x}, \qquad (\mathbf{B}_S)_{i,j} = K \int_\Omega \nabla\psi_i(\boldsymbol{x}) \cdot \nabla\psi_j(\boldsymbol{x}) d\boldsymbol{x}.$$

Let U_i^n $(1 \le n \le N,\ 1 \le i \le M)$ be the finite element solution of (18) at $t = t_n$ and $\boldsymbol{x} = \boldsymbol{x}_i$. We define a lower triangular matrix $\mathbf{A} = (a_{n,k})_{n,k=1}^N$ to store the discretization coefficients of δ_τ and $\tilde{\delta}_{t_n}^\rho$ by

$$a_{n,k} = \begin{cases} \dfrac{1}{\tau} + \kappa_n \tilde{c}_{n,n}, & k = n, \\ -\dfrac{1}{\tau} + \kappa_n(\tilde{c}_{n,n-1} - \tilde{c}_{n,n}), & k = n-1, \\ \kappa_n(\tilde{c}_{n,k} - \tilde{c}_{n,k+1}), & 1 \le k \le n-2, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{U}^{(n)} = [U_1^n, U_2^n, \cdots, U_M^n]^\top$ and $\boldsymbol{F}^{(n)} = [F_1^n, F_2^n, \cdots, F_M^n]^\top$ with $F_i^n = \int_\Omega f(\boldsymbol{x}, t_n)\psi_i(\boldsymbol{x}) d\boldsymbol{x}$ for $1 \le n \le N$. Then the matrix form of (18) is

$$\left(a_{n,n}\mathbf{B}_M + \mathbf{B}_S\right)\boldsymbol{U}^{(n)} = \boldsymbol{F}^{(n)} - \sum_{k=0}^{n-1} a_{n,k}\mathbf{B}_M \boldsymbol{U}^{(k)}, \quad 1 \le n \le N. \tag{25}$$

For each time step, $O(M)$ operations are required to compute $\mathbf{B}_M \boldsymbol{U}^{(k)}$ for $0 \le k \le n-1$, and hence $O(nM)$ operations are needed for evaluating $\sum_{k=0}^{n-1} a_{n,k}\mathbf{B}_M \boldsymbol{U}^{(k)}$. As the computational complexity for solving the linear system (25) is $O(M)$, totally $O(MN + N^2)$ storage and $O(MN^2)$ computations are needed for solving (25) by the time-stepping method, which is expensive and motivates the development of the fast solver.

5.1. Fast solver and its analysis

We solve the finite element system (18) by rewriting it as the following all-at-once system

$$\left(\mathbf{A} \otimes \mathbf{B}_M + \mathbf{I}_N \otimes \mathbf{B}_S\right)\boldsymbol{U} = \boldsymbol{F}, \tag{26}$$

where $\mathbf{I}_N$ is a $N \times N$ identity matrix, $\boldsymbol{U} = [(\boldsymbol{U}^{(1)})^\top, (\boldsymbol{U}^{(2)})^\top, \cdots, (\boldsymbol{U}^{(N)})^\top]^\top$ and $\boldsymbol{F} = [\boldsymbol{F}_1^{(1)\top}, \boldsymbol{F}^{(2)\top}, \cdots, \boldsymbol{F}^{(N)\top}]^\top$. From (12) and (15), we decompose the matrix $\mathbf{A}$ into the sum of a band matrix $\mathbf{A}_b$ and a sum of diagonal matrices multiplied by Toeplitz matrices $\mathbf{A}_t$,

$$\mathbf{A} = \mathbf{A}_b + \mathbf{A}_t := \mathbf{A}_b + \text{diag}(\boldsymbol{K}) \sum_{v=0}^{S} \text{diag}(\boldsymbol{C}_N^v)\mathbf{T}^{(v)}, \tag{27}$$

where $(\mathbf{A}_b)_{i,j} = a_{i,j}$ for $|i-j| \le S$ and zeros elsewhere, $\boldsymbol{K} = [\kappa_1, \kappa_2, \cdots, \kappa_N]^\top$, $\boldsymbol{C}_N^v = [C_1^v, C_2^v, \cdots, C_N^v]^\top$ and the Toeplitz matrix $\mathbf{T}^{(v)} = toep(\boldsymbol{t}_c^{(v)}, \mathbf{0})$ with $\boldsymbol{t}_c^{(v)} = (t_{c,i}^v)_{i=1}^N$ and $\mathbf{0}$ being its first column and first row, respectively. Here $t_{c,i}^{(v)}$ is given by $t_{c,i}^{(v)} = g_{i-1}^{(v)}(\underline{\alpha}) - g_{i-2}^{(v)}(\underline{\alpha})$ for $S+1 \le i \le N$ and $t_{c,i}^{(v)} = 0$ otherwise.

We divide $\mathbf{A}$ into four $N/2 \times N/2$ matrices as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_3 & \mathbf{A}_2 \end{bmatrix},$$

and accordingly divide $\boldsymbol{U} = [\boldsymbol{U}_1^\top, \boldsymbol{U}_2^\top]^\top$ and $\boldsymbol{F} = [\boldsymbol{F}_1^\top, \boldsymbol{F}_2^\top]^\top$. Then we could solve (26) by equivalently solving the following two linear subsystems

$$\begin{cases} \left(\mathbf{A}_1 \otimes \mathbf{B}_M + \mathbf{I}_{N/2} \otimes \mathbf{B}_S\right)\boldsymbol{U}_1 = \boldsymbol{F}_1, \\ \left(\mathbf{A}_2 \otimes \mathbf{B}_M + \mathbf{I}_{N/2} \otimes \mathbf{B}_S\right)\boldsymbol{U}_2 = \boldsymbol{F}_2 - \left(\mathbf{A}_3 \otimes \mathbf{B}_M\right)\boldsymbol{U}_1. \end{cases} \tag{28}$$

As each subsystem in (28) has a similar structure as (26), we could repeat the previous steps to obtain a fast DAC (fDAC) algorithm presented in Algorithm 1, and we remain to estimate its computational cost in the following.

Algorithm 1 The fDAC algorithm for (26).

function $\boldsymbol{U} = \text{fDAC}(\mathbf{A}, \mathbf{B}_M, \mathbf{B}_S, \boldsymbol{F})$
  $P = \text{length}(\mathbf{A}(:,1))$
  if $P \le S$
    solve
    $$(a_{p,p}\mathbf{B}_M + \mathbf{B}_S)\boldsymbol{U}^{(p)} = \boldsymbol{F}^{(p)} - \sum_{k=0}^{p-1} a_{p,k}\mathbf{B}_M \boldsymbol{U}^{(k)},$$
    for $1 \le p \le P$
  else
    $\boldsymbol{U}_1 = \text{fDAC}(\mathbf{A}_1, \mathbf{B}_M, \mathbf{B}_S, \boldsymbol{F}_1)$
    $\boldsymbol{F}_2 = \boldsymbol{F}_2 - (\mathbf{A}_3 \otimes \mathbf{B}_M)\boldsymbol{U}_1$
    $\boldsymbol{U}_2 = \text{fDAC}(\mathbf{A}_2, \mathbf{B}_M, \mathbf{B}_S, \boldsymbol{F}_2)$
  end if
end function

Lemma 5.1. *For any $\boldsymbol{v} \in \mathbb{R}^{N/2}$, the matrix-vector multiplication $\mathbf{A}_3\boldsymbol{v}$ can be carried out in $O(N\log^2 N)$ operations, and hence $\left(\mathbf{A}_3 \otimes \mathbf{B}_M\right)\boldsymbol{U}_1$ in (28) can be computed in $O(MN\log^2 N)$ operations.*

Proof. By (27), we have

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_3 & \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_b^1 & \mathbf{0} \\ \mathbf{A}_b^3 & \mathbf{A}_b^2 \end{bmatrix} + \begin{bmatrix} \mathbf{A}_t^1 & \mathbf{0} \\ \mathbf{A}_t^3 & \mathbf{A}_t^2 \end{bmatrix},$$

which yields $\mathbf{A}_3^{(1)}\boldsymbol{v} = \mathbf{A}_b^3\boldsymbol{v} + \mathbf{A}_t^3\boldsymbol{v}$. As $\mathbf{A}_b$ is a band matrix with bandwidth S, there are only $(S+1)(S+2)/2$ nonzero entries on the top right corner of $\mathbf{A}_b^3$, and thus the computational complexity of $\mathbf{A}_b^3\boldsymbol{v}$ is $O(S^2) = O(\log^2 N)$. By (15), the $N/2 \times N/2$ matrix $\mathbf{A}_t^3$ can be decomposed by a sum with $S+1$ summands with each summand being a diagonal matrix multiplied by a Toeplitz matrix. By applying the fast Fourier transform [20], the matrix-vector multiplication $\mathbf{A}_t^3\boldsymbol{v}$ can be carried out in $O(SN/2\log N/2) = O(N\log^2 N)$ operations. We thus proved the first statement of the lemma.

We use the property of Kronecker product to get

$$(\mathbf{A}_t^3 \otimes \mathbf{B}_M)\boldsymbol{U}_1 = vec\left(\mathbf{B}_M\mathbf{X}(\mathbf{A}_t^3)^\top\right) = vec\left(\mathbf{B}_M(\mathbf{A}_t^3\mathbf{X}^\top)\right),$$

where $\mathbf{X}$ is the matrix form of $\boldsymbol{U}_1$ given by reshaping $\boldsymbol{U}_1$ into a $M \times N/2$ matrix with its i-th row representing the unknowns at $\boldsymbol{x}_i$, and $vec(\cdot)$ reshapes a $M \times N/2$ matrix into a vector by a inverse manner. Therefore, the multiplication $\mathbf{Z} = \mathbf{A}_t^3\mathbf{X}^\top$ requires $O(MN\log^2 N)$ operations. We combine the fact that the computational complexity of $\mathbf{B}_M\mathbf{Z}$ is $O(MN)$ to prove the second statement of this lemma. $\square$

In the following theorems, we show the efficiency of the fDAC algorithm. Due to the dividing operations in the DAC method, we assume that $S = N/2^J$ for some $0 < J \in \mathbb{N}$ such that S satisfies the Condition S.

Theorem 5.1. *$O(N^{3/2}\log N)$ computations and $O(N\log N)$ storage are needed to compute and store the components of $\mathbf{A}$, respectively.*

Proof. We only need to compute and store the components $\mathbf{A}_b, \boldsymbol{K}, \boldsymbol{C}_N^{\nu}$, $\boldsymbol{t}_c^{(\nu)}$ of $\mathbf{A}$ in (27) for Algorithm 1 with totally $O(N\log N)$ memory requirement, rather than directly generating $\{\tilde{c}_{n,k}\}$. The computational cost of $\mathbf{A}_b$ and $\boldsymbol{C}_N^{(\nu)}$ is $O(N^{3/2}\log N)$ by using the composite middle rectangular formula, while $O(N)$ and $O(N\log N)$ operations are required for $\boldsymbol{K}$ and $t_c^{(\nu)}$, respectively. Thus $O(N^{3/2}\log N)$ computations are needed for generating components of $\mathbf{A}$. □

Theorem 5.2. *The fDAC algorithm requires $O(MN\log^3 N)$ computations for solving the linear system. Compared with the $O(MN^2)$ computational cost of the time-stepping method mentioned before Section 5.1, the fDAC algorithm significantly improves the efficiency.*

Proof. By assumptions on S, it holds $J = O(\log N)$. By Lemma 5.1, we require $O\big(MN/2\log^2 N/2\big)$ operations to compute $\big(\mathbf{A}_3 \otimes \mathbf{B}_M\big)\boldsymbol{U}_1$ in (28). As the fast DAC algorithm repeats the dividing operations like (28), the total computational complexity of matrix-vector multiplications in the dividing procedure can be computed by

$$M \times O\Big(\frac{N}{2}\log^2\frac{N}{2} + 2\cdot\frac{N}{2^2}\log^2\frac{N}{2^2} + \cdots + 2^{J-1}\cdot\frac{N}{2^J}\log^2\frac{N}{2^J}\Big)$$
$$= O\Big(\frac{MN}{2}\Big(\log^2 N\Big(1+\frac{1}{2}+\cdots+2^{J-1}\cdot\frac{1}{2^{J-1}}\Big)\Big)\Big) = O(MN\log^3 N).$$

After the dividing procedure, we need to solve 2^J subsystems, each of which requires $O(MS+S^2)$ computations. Thus the total computational complexity is

$$2^J \times O(MS+S^2) = O(MN\log N + N\log^2 N).$$

Adding the above two results yields the total computational cost of the fDAC algorithm. □

6. Numerical experiments

We carry out numerical experiments to investigate the performance of the time-stepping finite element method (FEM) for solving (8) and the fast finite element method (fFEM) for solving (26) by the fDAC algorithm. All numerical experiments are implemented on Matlab R2016b on a computer with Intel(R) Core(TM) i7-9700 and Ram 16 GB.

We choose $T=1, \Omega=(0,1)$ or $(0,1)^2$, $K=0.01$ and use a uniform rectangular partition on Ω with the mesh size h. The probability density function $\rho(\alpha,t)$ is set in the form of Gaussian hump

$$\rho(\alpha,t) = \frac{1}{\sqrt{2.5\pi\times 10^{-3}(t+1)}}\exp\Big(-\frac{(\alpha-0.5)^2}{2.5\times 10^{-3}(t+1)}\Big),$$

and $\underline{\alpha}=0.3$ and $\overline{\alpha}=0.7$. We give the smooth or singular solutions in the following four cases:

(i) $u(x,t)=t^2\sin(\pi x)$, (ii) $u(x,t)=t^{1+\underline{\alpha}}\sin(\pi x)$,
(iii) $u(x,y,t)=t^2\sin(\pi x)\sin(\pi y)$, (iv) $u(x,y,t)=t^{1+\underline{\alpha}}\sin(\pi x)\sin(\pi y)$.

The right-hand side terms f are evaluated correspondingly. For all applications of the composite middle rectangle formula, we set $L=\sigma^{-1}=\lfloor\sqrt{N}\rfloor+1$ to ensure the $O(\tau)$ accuracy. We set $S=2^{\lfloor e^{1.04}\log N/\log 2\rfloor+1}$ to satisfy the Condition S.

Table 1
Errors and convergence rates for (i)-(iv).

	τ	$\|u-\hat{U}\|_{\hat{L}^\infty(L^2)}$	ι	$\|u-\tilde{U}\|_{\hat{L}^\infty(L^2)}$	$\tilde{\iota}$
(i)	2^{-5}	1.4217E-02	-	1.4217E-02	-
	2^{-6}	7.0730E-03	1.01	7.0730E-03	1.01
	2^{-7}	3.4950E-03	1.02	3.4950E-03	1.02
	2^{-8}	1.7418E-03	1.00	1.7418E-03	1.00
(ii)	2^{-5}	6.2315E-03	-	6.2315E-03	-
	2^{-6}	3.2608E-03	0.93	3.2608E-03	0.93
	2^{-7}	1.6606E-03	0.97	1.6606E-03	0.97
	2^{-8}	8.5363E-04	0.96	8.5363E-04	0.96
(iii)	2^{-5}	9.7524E-03	-	9.7524E-03	-
	2^{-6}	4.8879E-03	1.00	4.8879E-03	1.00
	2^{-7}	2.4503E-03	1.00	2.4503E-03	1.00
	2^{-8}	1.2559E-03	0.96	1.2559E-03	0.96
(iv)	2^{-5}	4.2474E-03	-	4.2474E-03	-
	2^{-6}	2.2505E-03	0.92	2.2505E-03	0.92
	2^{-7}	1.1768E-03	0.94	1.1768E-03	0.94
	2^{-8}	6.3820E-04	0.88	6.3820E-04	0.88

Table 2
CPU times of computing $\{c_{n,k}\}$ and components of $\tilde{c}_{n,k}$.

N	CPU_M	$CPU_{\tilde{M}}$
2^9	16.3 s	4.8 s
2^{10}	1 min 33 s	14 s
2^{11}	8 min 26 s	38 s
2^{12}	45 min 52 s	1 min 47 s
2^{13}	4 h 30 min	5 min
2^{14}	/	14 min 10 s
2^{15}	–	39 min 46 s
2^{16}	–	1 h 52 min

6.1. Comparison of convergence rates

As the spatial discretization is standard, we measure the temporal convergence rates of the FEM and fFEM with $h=1/200$ for (i)–(ii) and $h=1/100$ for (iii)–(iv) by

$$\|u-\hat{U}\|_{\hat{L}^\infty(L^2)} \le QN^{-\iota}, \quad \|u-\tilde{U}\|_{\hat{L}^\infty(L^2)} \le QN^{-\tilde{\iota}},$$

and present the numerical results in Table 1. We observe that the fFEM has almost the same accuracy as the FEM. Moreover, both methods have first order temporal convergence rates, which is consistent with the theoretical analysis.

6.2. CPU times of generating coefficients

We record the CPU times CPU_M for computing $\{c_{n,k}\}$ and $CPU_{\tilde{M}}$ for computing the components of $\{\tilde{c}_{n,k}\}$, and present the numerical results in Table 2 and the left plot of Fig. 1. The symbol '/' implies that the running time of generating $\{c_{n,k}\}$ is more than 20 hours and we stop it, and the symbol '–' implies that the storage is out of memory. From the numerical results we observe that the fast approximation (16) is much more efficient than (7) in generating the coefficients. For instance, when $N=2^{14}$, more than 20 hours are needed to compute $\{c_{n,k}\}$, while only 7 minutes are required for computing the components of $\{\tilde{c}_{n,k}\}$ as proved in Theorem 5.1. For large N like 2^{15}, the memory of the computer is not enough to store $\{c_{n,k}\}$, while the fast approximation still works.

6.3. CPU times of solving the linear systems

We record the CPU times CPU_C of solving the time-stepping FEM and $CPU_{\tilde{C}}$ of solving (26) by fDAC algorithm by setting $h=2^{-4}$ for (i)–(ii) and $h=2^{-3}$ for (iii)–(iv). The numerical results are presented in Tables 3-4 and the right plot of Fig. 1 for case (i), which demonstrates that the fFEM is much more efficient in solving the linear systems than the FEM.

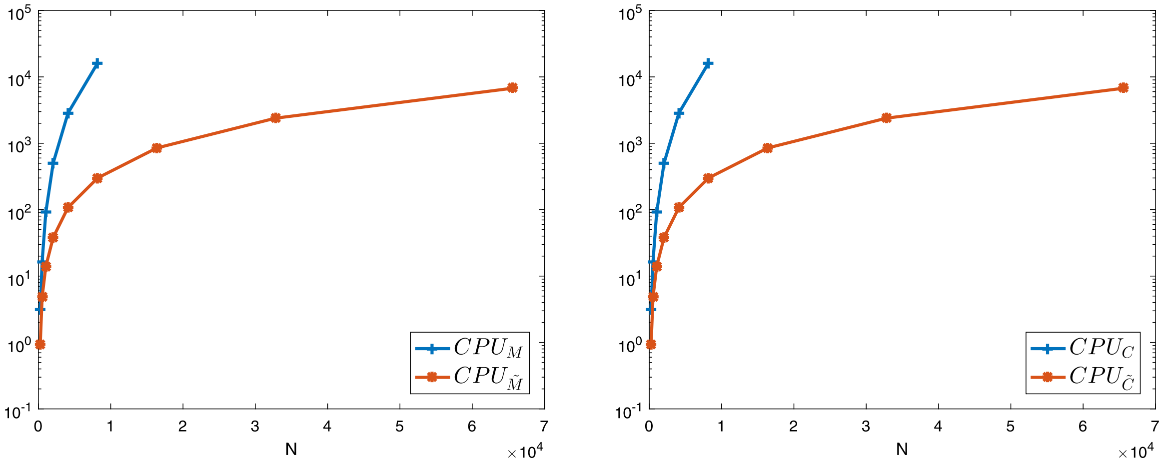

Fig. 1. CPU times of computing the coefficients (left) and solving the linear systems (right).

Table 3
CPU times of solving FEM and fFEM for (i)–(ii).

N	(i)		(ii)	
	CPU_C	$CPU_{\tilde{C}}$	CPU_C	$CPU_{\tilde{C}}$
2^9	0.096 s	0.21 s	0.098 s	0.14 s
2^{10}	0.38 s	0.40 s	0.38 s	0.30 s
2^{11}	1.53 s	0.71 s	1.54 s	0.66 s
2^{12}	6.31 s	1.32 s	6.28 s	1.39 s
2^{13}	26.1 s	2.69 s	25.5 s	2.78 s
2^{14}	/	5.65 s	/	6.01 s
2^{15}	–	12.1 s	–	12.0 s
2^{16}	–	25.2 s	–	25.2 s

Table 4
CPU times of solving FEM and fFEM for (iii)–(iv).

N	(iii)		(iv)	
	CPU_C	$CPU_{\tilde{C}}$	CPU_C	$CPU_{\tilde{C}}$
2^9	0.17 s	0.40 s	0.17 s	0.42 s
2^{10}	0.65 s	0.92 s	0.65 s	0.86 s
2^{11}	2.58 s	2.07 s	2.56 s	1.81 s
2^{12}	10.8 s	3.85 s	10.4 s	3.93 s
2^{13}	41.8 s	8.10 s	41.9 s	8.07 s
2^{14}	/	18.3 s	/	17.1 s
2^{15}	–	35.6 s	–	35.5 s
2^{16}	–	1 min 14 s	–	1 min 16 s

Acknowledgements

The authors would like to express their most sincere thanks to the referees for their very helpful comments and suggestions, which greatly improved the quality of this paper. This work was partially funded by the National Natural Science Foundation of China under Grants 11971272 and 12001337, by the Natural Science Foundation of Shandong Province under Grant ZR2019BA026, by the ARO MURI Grant W911NF-15-1-0562, by the National Science Foundation under Grant DMS-2012291, by the China Postdoctoral Science Foundation 2021TQ0017 and 2021M700244, by the International Postdoctoral Exchange Fellowship Program (Talent Introduction Program) YJ20210019. All data generated or analyzed during this study are included in this article.

References

[1] R.A. Adams, J.J.F. Fournier, Sobolev Spaces, Elsevier, San Diego, 2003.
[2] A. Alikhanov, Numerical methods of solutions of boundary value problems for the multi-term variable-distributed order diffusion equation, Appl. Math. Comput. 268 (2015) 12–22.
[3] D. Benson, R. Schumer, M.M. Meerschaert, S.W. Wheatcraft, Fractional Dispersion, Lévy Motions, and the MADE Tracer Tests, Springer, Netherlands, 2001.
[4] M. Caputo, M. Fabrizio, The kernel of the distributed order fractional derivatives with an application to complex materials, Fractal Fract. 1 (2017) 13.
[5] A.V. Chechkin, R. Gorenflo, I.M. Sokolov, Retarding subdiffusion and accelerating superdiffusion governed by distributed-order fractional diffusion equations, Phys. Rev. E 66 (2002) 046129.
[6] M. Chen, W. Deng, High order algorithms for the fractional substantial diffusion equation with truncated Lévy flights, SIAM J. Sci. Comput. 37 (2015) A890–A917.
[7] W. Deng, J. Hesthaven, Local discontinuous Galerkin methods for fractional diffusion equations, ESAIM: M2AN 47 (2013) 1845–1864.
[8] K. Diethelm, N.J. Ford, Numerical solution methods for distributed order differential equations, Fract. Calc. Appl. Anal. 4 (2001) 531–542.
[9] R. Du, A. Alikhanov, Z. Sun, Temporal second order difference schemes for the multi-dimensional variable-order time fractional sub-diffusion equations, Comput. Math. Appl. 79 (2020) 2952–2972.
[10] V.J. Ervin, Regularity of the solution to fractional diffusion, advection, reaction equations in weighted Sobolev spaces, J. Differ. Equ. 278 (2021) 294–325.
[11] Y. Efendiev, T. Hou, Multiscale Finite Element Methods: Theory and Applications, Surveys and Tutorials in the Applied Mathematical Sciences, vol. 4, Springer-Verlag, New York, 2009.
[12] Z. Fang, H. Sun, H. Wang, A fast method for variable-order Caputo fractional derivative with applications to time-fractional diffusion equations, Comput. Math. Appl. 80 (2020) 1443–1458.
[13] R. Garrappa, A. Giusti, F. Mainardi, Variable-order fractional calculus: a change of perspective, Commun. Nonlinear Sci. Numer. Simul. 102 (2021) 105904.
[14] J. Jia, H. Wang, A fast finite difference method for distributed-order space-fractional partial differential equations on convex domains, Comput. Math. Appl. 73 (2018) 2031–2043.
[15] J. Jia, H. Wang, Analysis of a hidden memory variably distributed-order space-fractional diffusion equation, Appl. Math. Lett. 124 (2022) 107617.
[16] J. Jia, H. Wang, X. Zheng, A preconditioned fast finite element approximation to variable-order time-fractional diffusion equations in multiple space dimensions, Appl. Numer. Math. 163 (2021) 15–29.
[17] S. Jiang, J. Zhang, Q. Zhang, Z. Zhang, Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations, Commun. Comput. Phys. 21 (2017) 650–678.
[18] B. Jin, B. Li, Z. Zhou, Subdiffusion with a time-dependent coefficient: analysis and numerical solution, Math. Comput. 88 (2019) 2157–2186.
[19] X. Jin, F. Lin, Z. Zhao, Preconditioned iterative methods for two-dimensional space-fractional diffusion equations, Commun. Comput. Phys. 18 (2015) 469–488.
[20] R. Ke, M. Ng, H. Sun, A fast direct method for block triangular Toeplitz-like with tri-diagonal block systems from time-fractional partial differential equations, J. Comput. Phys. 303 (2015) 203–211.
[21] E. Kharazmi, M. Zayernouri, G. Karniadakis, Petrov–Galerkin and spectral collocation methods for distributed order differential equations, SIAM J. Sci. Comput. 39 (2017) A1003–A1037.
[22] K. Le, W. McLean, M. Stynes, Existence, uniqueness and regularity of the solution of the time-fractional Fokker–Planck equation with general forcing, Commun. Pure Appl. Anal. 18 (2019) 2765–2787.
[23] B. Li, H. Luo, X. Xie, Analysis of a time-stepping scheme for time fractional diffusion problems with nonsmooth data, SIAM J. Numer. Anal. 57 (2019) 779–798.
[24] D. Li, H. Liao, W. Sun, J. Wang, J. Zhang, Analysis of L1-Galerkin FEMs for time-fractional nonlinear parabolic problems, Commun. Comput. Phys. 24 (2018) 86–103.
[25] H. Liao, T. Tang, T. Zhou, A second-order and nonuniform time-stepping maximum-principle preserving scheme for time-fractional Allen-Cahn equations, J. Comput. Phys. 414 (2020) 109473.

[26] Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation 225 (2) (2007) 1533–1552.
[27] C. Lorenzo, T. Hartley, Variable order and distributed order fractional operators, Nonlinear Dyn. 29 (2002) 57–98.
[28] X. Lu, H. Pang, H. Sun, Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations, Numer. Linear Algebra Appl. 22 (2015) 866–882.
[29] Y. Luchko, Boundary value problems for the generalized time-fractional diffusion equation of distributed order, Fract. Calc. Appl. Anal. 12 (2009) 409–422.
[30] O. Nikan, J. Machado, A. Golbabai, T. Nikazad, Numerical approach for modeling fractal mobile/immobile transport model in porous and fractured media, Int. Commun. Heat Mass Transf. 111 (2020) 104443.
[31] I. Podlubny, Fractional Differential Equations, Academic Press, 1999.
[32] W. Qiu, D. Xu, H. Chen, J. Guo, An alternating direction implicit Galerkin finite element method for the distributed-order time-fractional mobile-immobile equation in two dimensions, Comput. Math. Appl. 80 (2020) 3156–3172.
[33] K. Sakamoto, M. Yamamoto, Initial value/boundary value problems for fractional diffusion-wave equations and applications to some inverse problems, J. Math. Anal. Appl. 382 (2011) 426–447.
[34] R. Schumer, D.A. Benson, M.M. Meerschaert, B. Baeumer, Fractal mobile/immobile solute transport, Water Resour. Res. 39 (2003) 1–12.
[35] M. Stynes, E. O'Riordan, J.L. Gracia, Error analysis of a finite difference method on graded mesh for a time-fractional diffusion equation, SIAM J. Numer. Anal. 55 (2017) 1057–1079.
[36] N. Su, P. Nelson, S. Connor, The distributed-order fractional diffusion-wave equation of groundwater flow: theory and application to pumping and slug tests, J. Hydrol. 529 (2015) 1262–1273.
[37] H. Sun, A. Chang, Y. Zhang, W. Chen, A review on variable-order fractional differential equations: mathematical foundations, physical models, numerical methods and applications, Fract. Calc. Appl. Anal. 22 (2019) 27–59.
[38] V. Thomée, Galerkin Finite Element Methods for Parabolic Problems, Lecture Notes in Mathematics, vol. 1054, Springer-Verlag, New York, 1984.
[39] H. Wang, X. Zheng, Wellposedness and regularity of the variable-order time-fractional diffusion equations, J. Math. Anal. Appl. 475 (2019) 1778–1802.
[40] Z. Yang, X. Zheng, H. Wang, A variably distributed-order time-fractional diffusion equation: analysis and approximation, Comput. Methods Appl. Mech. Eng. 367 (2020) 113118.
[41] H. Ye, F. Liu, V. Anh, Compact difference scheme for distributed-order time-fractional diffusion-wave equation on bounded domains, J. Comput. Phys. 298 (2015) 652–660.
[42] M. Zayernouri, G.E. Karniadakis, Fractional Sturm-Liouville eigen-problems: theory and numerical approximations, J. Comput. Phys. 47 (2013) 2108–2131.
[43] F. Zeng, Z. Zhang, G. Karniadakis, A generalized spectral collocation method with tunable accuracy for variable-order fractional differential equations, SIAM J. Sci. Comput. 37 (2015) A2710–A2732.
[44] J. Zhang, Z. Fang, H. Sun, Exponential-sum-approximation technique for variable-order time-fractional diffusion equations, J. Appl. Math. Comput. (2021), https://doi.org/10.1007/s12190-021-01528-7.
[45] Y. Wang, A high-order compact finite difference method and its extrapolation for fractional mobile/immobile convection-diffusion equations, Calcolo 54 (2017) 733–768.
[46] Y. Zhang, C. Green, B. Baeumer, Linking aquifer spatial properties and non-Fickian transport in mobile-immobile like alluvial settings, J. Hydrol. 512 (2014) 315–331.
[47] X. Zheng, H. Wang, A hidden-memory variable-order fractional optimal control model: analysis and approximation, SIAM J. Control Optim. 59 (2021) 1851–1880.
[48] X. Zheng, H. Wang, Optimal-order error estimates of finite element approximations to variable-order time-fractional diffusion equations without regularity assumptions of the true solutions, IMA J. Numer. Anal. 41 (2021) 1522–1545.
[49] X. Zheng, H. Wang, An error estimate of a numerical approximation to a hidden-memory variable-order space-time fractional diffusion equation, SIAM J. Numer. Anal. 58 (2020) 2492–2514.

Preflight Results

Document Information

Title: A fast numerical scheme for a variably distributed-order time-fractional diffusion equation and its analysis
Author: Jinhong Jia
Creator: Elsevier
Producer: Acrobat Distiller 9.5.5 (Windows)

Preflight Information

Profile: Convert to PDF/A-1b
Version: Qoppa jPDFPreflight v2022R1.00
Date: Sep 5, 2022 6:27:21 PM

Legend: (X) - Can NOT be fixed by PDF/A-1b conversion.
(!X) - Could be fixed by PDF/A-1b conversion. User chose to be warned in PDF/A settings.

Page 6 Results

(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.
(X) Font widths must be the same in both the font dictionary and the embedded font file.