



A fast algorithm for time-fractional diffusion equation with space-time-dependent variable order

Jinhong Jia¹ · Hong Wang² · Xiangcheng Zheng³

Received: 4 October 2022 / Accepted: 31 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

We investigate a fast algorithm for the time-fractional diffusion equation with a space-time-dependent variable order, which models, e.g., the subdiffusion with varying memory effects. In addition to the traditional L1 discretization of the time-fractional derivative, we perform a further approximation for the L1 coefficients, analyze the structures of the resulting all-at-once system, and apply the divide and conquer method to obtain a fast numerical algorithm. Due to the spatial dependence of the variable order and the further approximation to the L1 coefficients, the temporal discretization coefficients are coupled with the inner product of the finite element method and lack the monotonicity, which are rarely encountered in previous works and thus motivate novel analysis methods and computational techniques. Compared with the standard time-stepping methods with L1 discretization, the proposed algorithm reduces the complexity of solving the all-at-once system from $O(MN^2)$ to $O(MN \ln^3 N)$, where M stands for the spatial degree of freedom and N refers to the number of time steps. Numerical experiments are provided to substantiate the theoretical findings.

Keywords Space-time-dependent variable order · Time-fractional diffusion equation · Finite element method · Error estimate · Divide and conquer

Mathematics Subject Classification (2010) 35R11 · 65M15 · 65M60

✉ Xiangcheng Zheng
xzheng@sdu.edu.cn

Jinhong Jia
jhjia@sdu.edu.cn

Hong Wang
hwang@math.sc.edu

¹ School of Mathematics and Statistics, Shandong Normal University, Jinan 250358, China

² Department of Mathematics, University of South Carolina, Columbia 29208, SC, USA

³ School of Mathematics, Shandong University, Jinan 250100, China

1 Introduction

Time-fractional diffusion equations have been widely applied in various fields and have attracted extensive mathematical and numerical investigations [1–11] due to their model capability for challenging phenomena with memory effects. As the order of the time-fractional diffusion equation relates to the fractal dimension of the porous media, the variable-order fractional operators are introduced to accommodate the structure changes of the surroundings in, e.g., bioclogging and hydrofracturing in gas and oil recovery [7, 8, 12, 13], which leads to the following time-fractional diffusion equation with a space-time-dependent variable order proposed in, e.g., [14–17]

$$\begin{aligned} \partial_t u + \kappa(\mathbf{x}, t) {}^C D_t^{\gamma(\mathbf{x}, t)} u - \nabla \cdot (\mathbf{K}(\mathbf{x}) \nabla u) &= f, \quad (\mathbf{x}, t) \in \Omega \times (0, T]; \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega; \quad u(\mathbf{x}, t) = 0, \quad (\mathbf{x}, t) \in \partial\Omega \times [0, T]. \end{aligned} \quad (1.1)$$

Here $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$) is a simply connected bounded domain with smooth boundary $\partial\Omega$, $\mathbf{x} = (x_1, \dots, x_d)$, $\mathbf{K}(\mathbf{x}) := (k_{i,j}(\mathbf{x}))_{i,j=1}^d$ is a symmetric and coercive diffusion tensor, $\kappa(\mathbf{x}, t) > 0$ is the ratio of the normal to anomalous diffusion particles, $f(\mathbf{x}, t)$ is the source or sink term, and the fractional derivative with space-time-dependent variable order $0 \leq \gamma(\mathbf{x}, t) < 1$ is given by [18–20]

$${}^C D_t^{\gamma(\mathbf{x}, t)} g(\mathbf{x}, t) := \frac{1}{\Gamma(1 - \gamma(\mathbf{x}, t))} \int_0^t \frac{\partial_s g(\mathbf{x}, s)}{(t - s)^{\gamma(\mathbf{x}, t)}} ds. \quad (1.2)$$

There exist extensive investigations for the fast algorithms for constant-order fractional problems [11, 21–24], while the corresponding studies for variable-order fractional models are far from well developed. For space-fractional problems, Pang and Sun proposed a fast algorithm for the variable-order space-fractional advection-diffusion equations with nonlinear source terms in their pioneering work [50]. By the properties of the elements of coefficient matrices, the off-diagonal blocks were approximated by low-rank matrices and a fast algorithm based on the polynomial interpolation was developed to approximate the coefficient matrices. The approximation could be constructed in $O(kN)$ operations and requires $O(kN)$ storage with N and k being the number of unknowns and the approximants, respectively, and the matrix-vector multiplication could be implemented in $O(kN \log N)$ complexity, which leads to a fast iterative solver for the resulting linear systems. A different approach was proposed in [26] to develop a fast divide and conquer indirect collocation method for the variable-order space-fractional diffusion equations, where the stiffness matrix was approximated by the finite sum of Toeplitz matrices multiplied by diagonal matrices. The approximation was shown to be asymptotically consistent with the original problem, which requires $O(N \log^2 N)$ memory and $O(N \log^3 N)$ computational complexity with N being the numbers of unknowns.

There are also some recent progresses for the fast algorithms of time-fractional problems with time-dependent variable order $\gamma = \gamma(t)$. In [27] a fast algorithm was developed for the subdiffusion equation with time-dependent variable order based on a shifted binary block partition and uniform polynomial approximations. In [25] a

fast all-at-once method was proposed for the variable-order time-fractional diffusion equation by approximating the off-diagonal blocks by low-rank matrices based on the polynomial interpolation. In [28, 29] the exponential-sum-approximation technique was developed to approximate the variable-order fractional derivative and corresponding time-fractional problems. This method was then combined with the L_2 - 1_σ formula to construct a fast second-order approximation to the Caputo variable-order fractional derivative and the corresponding subdiffusion problem [30].

Despite the growing studies for the fast algorithms for time-fractional problems with time-dependent variable order $\gamma = \gamma(t)$, there are rare studies for those with space-time-dependent variable order $\gamma = \gamma(x, t)$, and it is unclear whether the existing methods in the literature could be extended for such more complicated problems. Due to the space dependence of the variable order, the temporal discretization coefficients of the variable-order fractional derivative could not be separated from the inner product of the finite element method, which makes the numerical analysis and fast implementation intricate. Furthermore, in order to develop the fast algorithm, we perform a further approximation for the coefficients of the L_1 method for the variable-order time-fractional derivative, which results in the loss of monotonicity of the coefficients that is critical in error estimates. In this work we overcome these difficulties to develop and analyze a fast algorithm for model (1.1), which reduces the complexity of solving the all-at-once system from $O(MN^2)$ to $O(MN \ln^2 N)$, where M stands for the spatial degree of freedom and N refers to the number of time steps.

The rest of this paper is organized as follows. In Section 2 we develop a fast approximation for the space-time-dependent variable-order fractional derivative and then model (1.1). In Section 3 we prove error estimate of the proposed algorithm. In Section 4 we implement the fast approximate scheme by the divide and conquer algorithm and carry out corresponding analysis. Various numerical experiments are performed in Section 5 and we address concluding remarks in the last section.

2 Fast approximation scheme

Let $H^m(\Omega) := W^{m,2}(\Omega)$ be the Sobolev space of functions with weak derivatives up to order m in $L^2(\Omega)$, and $H_0^m(\Omega)$ be its subspace with zero boundary condition up to order $m - 1$. For a Banach space $\mathcal{B}$, define $W^{m,p}(0, T; \mathcal{B})$ with respect to the norm $\|\cdot\|_{\mathcal{B}}$ for $1 \leq p < \infty$ by [31]

$$W^{m,p}(0, T; \mathcal{B}) := \left\{ v : [0, T] \rightarrow \mathcal{B} : \|\partial_t^k v(\cdot, t)\|_{\mathcal{B}} \in L^p(0, T), 0 \leq k \leq m \right\}$$

equipped with the norm

$$\|v\|_{W^{m,p}(0,T;\mathcal{B})} := \left(\sum_{k=0}^m \int_0^T \|\partial_t^k v(\cdot, t)\|_{\mathcal{B}}^p dt \right)^{1/p}.$$

In particular, we have $L^p(0, T; \mathcal{B}) := W^{0,p}(0, T; \mathcal{B})$. For simplicity, we may drop the subscript L^2 and the domain Ω in the Sobolev spaces and norms and write $W^{m,p}(\mathcal{B})$ for $W^{m,p}(0, T; \mathcal{B})$ when no confusion occurs.

For a positive integer N , we define a uniform temporal partition on $[0, T]$ by $t_n = n\tau$ ($0 \leq n \leq N$) with $\tau = T/N$ and a quasi-uniform partition of Ω with the maximal mesh diameter h , and let S_h be the space of continuous and piecewise linear functions on Ω with respect to the spatial partition. The Ritz projection $\mathcal{P}_h : H_0^1 \rightarrow S_h$ is defined by

$$(\mathbf{K}\nabla(g - \mathcal{P}_h g), \nabla\chi) = 0, \quad \forall \chi \in S_h,$$

which satisfies the following approximation property [32]

$$\|g - \mathcal{P}_h g\| \leq Qh^2 \|g\|_{H^2}, \quad g \in H^2 \cap H_0^1. \quad (2.1)$$

We further introduce the discrete-in-time space $\hat{L}^\infty(L^2)$ equipped with the norm for $v = \{v_j\}_{j=1}^N$ with $v_j \in L^2(0, l)$

$$\|v\|_{\hat{L}^\infty(L^2)} := \max_{1 \leq j \leq N} \|v_j\|.$$

To discretize model (1.1), we denote $u_n := u(\mathbf{x}, t_n)$, $\kappa_n := \kappa(\mathbf{x}, t_n)$, $\gamma_n := \gamma(\mathbf{x}, t_n)$ and $f_n := f(\mathbf{x}, t_n)$ for brevity. We approximate the derivatives u_t and ${}_0^C D_t^{\gamma(x,t)} u$ at $t = t_n$ by the backward Euler and $L1$ schemes

$$\begin{aligned} u_t(\mathbf{x}, t)|_{t=t_n} &= \frac{u_n - u_{n-1}}{\tau} + \frac{1}{\tau} \int_{t_{n-1}}^{t_n} \partial_{tt} u(\mathbf{x}, t)(t - t_{n-1}) dt \\ &:= \delta_\tau u_n(\mathbf{x}) + E_n, \end{aligned} \quad (2.2)$$

$$\begin{aligned} {}_0^C D_t^{\gamma(x,t)} u(\mathbf{x}, t)|_{t=t_n} &= \sum_{k=1}^n \int_{t_{k-1}}^{t_k} \frac{\delta_\tau u_k + (\partial_s u(\mathbf{x}, s) - \delta_\tau u_k) ds}{\Gamma(1 - \gamma_n)(t_n - s)^{\gamma_n}} \\ &:= \sum_{k=1}^n \hat{b}_{n,k}(u_k - u_{k-1}) + R_n, \end{aligned} \quad (2.3)$$

where

$$\begin{aligned} \hat{b}_{n,k} &:= \frac{1}{\Gamma(2 - \gamma_n)\tau^{\gamma_n}} ((n - k + 1)^{1-\gamma_n} - (n - k)^{1-\gamma_n}), \\ R_n &:= \sum_{k=1}^n R_{n,k} = \frac{1}{\Gamma(1 - \gamma_n)\tau} \sum_{k=1}^n \int_{t_{k-1}}^{t_k} \frac{\int_{t_{k-1}}^{t_k} \int_z^s \partial_{\theta\theta} u(\mathbf{x}, \theta) d\theta dz}{(t_n - s)^{\gamma_n}} ds. \end{aligned} \quad (2.4)$$

To derive a fast algorithm, we further approximate the coefficients $\{\hat{b}_{n,k}\}$ for $n-k \geq 1$ via the Taylor expansion at the power $\bar{\gamma}$

$$\begin{aligned} & (n-k+1)^{1-\gamma_n} - (n-k)^{1-\gamma_n} \\ &= \sum_{s=0}^S \frac{(\bar{\gamma} - \gamma_n)^s}{s!} \left((n-k+1)^{1-\bar{\gamma}} \ln^s(n-k+1) \right. \\ & \quad \left. - (n-k)^{1-\bar{\gamma}} \ln^s(n-k) \right) + \tilde{P}_{n,k}, \end{aligned} \quad (2.5)$$

where S is the number of the expansion terms and $\bar{\gamma} := \frac{1}{2}(\gamma_* + \gamma^*)$ where γ_* and γ^* refer to the lower and upper limits of γ over $\Omega \times [0, T]$. $\tilde{P}_{n,k}$ represents the local truncation error given by

$$\begin{aligned} \tilde{P}_{n,k} &= \frac{(\bar{\gamma} - \gamma_n)^{S+1}}{(S+1)!} \left((n-k+1)^{1-\epsilon_{n,k}} \ln^{S+1}(n-k+1) \right. \\ & \quad \left. - (n-k)^{1-\epsilon_{n,k}} \ln^{S+1}(n-k) \right), \end{aligned}$$

where $\epsilon_{n,k}$ lies in between γ_n and $\bar{\gamma}$ depending on n, k and $\mathbf{x}$. Substitute (2.5) into $\hat{b}_{n,k}$ in (2.4) to get

$$\begin{aligned} \hat{b}_{n,k} &= \sum_{s=0}^S \frac{(\bar{\gamma} - \gamma_n)^s}{s! \Gamma(2 - \gamma_n) \tau^{\gamma_n}} \left((n-k+1)^{1-\bar{\gamma}} \ln^s(n-k+1) \right. \\ & \quad \left. - (n-k)^{1-\bar{\gamma}} \ln^s(n-k) \right) + P_{n,k} =: b_{n,k} + P_{n,k} \end{aligned} \quad (2.6)$$

for $n-k \geq 1$ where $P_{n,k} = \tilde{P}_{n,k} / (\Gamma(2 - \gamma_n) \tau^{\gamma_n})$. For the case $n = k$, we set $b_{n,n} = \hat{b}_{n,n}$ and $P_{n,k} = 0$ for completeness. Consequently, we replace $\hat{b}_{n,k}$ by $b_{n,k}$ in (2.3) to obtain a further approximation of ${}^C D_t^{\gamma(x,t)} u$ at $t = t_n$

$$\begin{aligned} {}^C D_t^{\gamma(x,t)} u|_{t=t_n} &= \sum_{k=1}^n b_{n,k} (u_k - u_{k-1}) + \sum_{k=1}^n P_{n,k} (u_k - u_{k-1}) + R_n \\ &:= \delta_\tau^{\gamma_n} u_n + F_n + R_n, \end{aligned} \quad (2.7)$$

where

$$F_n = \sum_{k=1}^n P_{n,k} (u_k - u_{k-1}). \quad (2.8)$$

We invoke (2.2) and (2.7) in (1.1) and integrate the resulting equation multiplied by $\chi \in H_0^1(\Omega)$ to get the weak formulation

$$\begin{aligned} & (\delta_\tau u_n, \chi) + (\kappa_n \delta_\tau^{\gamma_n} u_n, \chi) + (\mathbf{K} \nabla u_n, \nabla \chi) \\ &= (f_n, \chi) - (E_n + \kappa_n (R_n + F_n), \chi), \quad \forall \chi \in H_0^1(\Omega). \end{aligned} \quad (2.9)$$

We drop the local truncation errors to obtain a fast finite element scheme: find $U_n \in S_h$ for $n = 1, 2, \dots, N$ with $U_0 = \mathcal{P}_h u_0$ such that

$$(\delta_\tau U_n, \chi_h) + (\kappa_n \delta_\tau^{\gamma_n} U_n, \chi_h) + (\mathbf{K} \nabla U_n, \chi_h) = (f_n, \chi_h), \quad \forall \chi_h \in S_h. \quad (2.10)$$

3 Error estimate

We prove error estimate of the fast scheme (2.10). The main difficulties we overcome lie in the loss of monotonicity of $\{b_{n,k}\}$ due to the Taylor expansion (2.5) and the coupling of the variable order and the inner product of the finite element method, which are rarely encountered in the literature and novel analysis techniques are developed to account for these issues, cf. the proof of Theorem 3.1.

3.1 Auxiliary estimates

We prove several auxiliary estimates based on the regularity of the solutions derived in [16], which shows that under suitable smoothness assumptions on the data, the solution u is bounded as follows

$$\|u\|_{W^{2,p}(L^2)} + \|u\|_{W^{1,p}(H^2)} \leq Q, \quad (3.1)$$

where $1 \leq p \leq 2$ and $p < 1/\gamma_0$ with

$$\gamma_0 := \|\gamma(\cdot, 0)\|_{L^\infty(\Omega)}. \quad (3.2)$$

Without loss of generality, we assume there exists a $0 < \gamma^* < 1$ such that

$$0 \leq \gamma(\mathbf{x}, t) \leq \gamma^*, \quad \forall (\mathbf{x}, t) \in \Omega \times [0, T]. \quad (3.3)$$

Lemma 3.1 *If $S \geq \lfloor 3 \ln N \rfloor$, $P_{n,k}$ can be bounded by*

$$\sum_{k=1}^n |P_{n,k}| < 1, \quad \sum_{n=k}^N |P_{n,k}| < 1. \quad (3.4)$$

Proof To bound $P_{n,k}$, we define the monotone decreasing function $y(\alpha) = (n - k + 1)^{1-\alpha} \ln^{S+1}(n - k + 1) - (n - k)^{1-\alpha} \ln^{S+1}(n - k)$ for $n - k \geq 1$ on $\alpha \in [0, 1]$. Consequently, we have for $\epsilon_{n,k} \in [0, 1]$

$$\begin{aligned} y(\epsilon_{n,k}) &= (n - k + 1)^{1-\epsilon_{n,k}} \ln^{S+1}(n - k + 1) - (n - k + 1)^{1-\epsilon_{n,k}} \ln^{S+1}(n - k) \\ &< (n - k + 1) \ln^{S+1}(n - k + 1) - (n - k) \ln^{S+1}(n - k) = y(0). \end{aligned}$$

As $|\bar{\gamma} - \gamma_n| < 1/2$ and $1/\Gamma(2 - \gamma_n) < 1.2$ for $\gamma_n < 1$, we bound $P_{n,k}$ by

$$|P_{n,k}| \leq \frac{1.2N^{\gamma^*}}{2^{S+1}(S+1)!} \left((n-k+1) \ln^{S+1}(n-k+1) - (n-k) \ln^{S+1}(n-k) \right).$$

Therefore, we have

$$\begin{aligned} \sum_{k=1}^n |P_{n,k}| &\leq 1.2N^{\gamma^*} \sum_{k=1}^{n-1} \frac{(n-k+1) \ln^{S+1}(n-k+1) - (n-k) \ln^{S+1}(n-k)}{2^{S+1}(S+1)!} \\ &\leq \frac{1.2N^{\gamma^*}}{2^{S+1}(S+1)!} \sum_{k=1}^{n-1} \left((n-k+1) \ln^{S+1}(n-k+1) - (n-k) \ln^{S+1}(n-k) \right) \\ &\leq \frac{1.2N^{1+\gamma^*} \ln^{S+1} N}{2^{S+1}(S+1)!}. \end{aligned}$$

By setting $S+1 \geq 3 \ln N$ and applying the Stirling's formula

$$(S+1)! \geq \frac{(S+1)^{S+3/2}}{e^{S+1}} \quad (3.5)$$

we get

$$\begin{aligned} \frac{\ln^{S+1} N}{2^{S+1}(S+1)!} &\leq \frac{(e \ln N/2)^{S+1}}{(S+1)^{S+3/2}} \\ &\leq \frac{1}{\sqrt{S+1}} \left(\frac{e \ln N}{2(S+1)} \right)^{S+1} \leq \frac{1}{\sqrt{S+1}} \left(\frac{e}{6} \right)^{3 \ln N}. \end{aligned} \quad (3.6)$$

We invoke this and $2 + 3 \ln \frac{e}{6} < 0$ in the estimate of $P_{n,k}$ to get

$$\sum_{k=1}^n |P_{n,k}| \leq \frac{1.2N^{1+\gamma^*} \ln^{S+1} N}{2^{S+1}(S+1)!} \leq \frac{1.2N^2}{\sqrt{S+1}} \left(\frac{e}{6} \right)^{3 \ln N} \leq N^{2+3 \ln \frac{e}{6}} < 1.$$

Similarly, we estimate the second statement of (3.4) by

$$\begin{aligned} \sum_{n=k}^N |P_{n,k}| &\leq \frac{1.2N}{2^{S+1}(S+1)!} \\ &\quad \times \sum_{n=k+1}^N \left((n-k+1) \ln^{S+1}(n-k+1) - (n-k) \ln^{S+1}(n-k) \right) \\ &\leq \frac{1.2N^{1+\gamma^*} \ln^{S+1} N}{2^{S+1}(S+1)!} < 1 \end{aligned}$$

to complete the proof. $\square$

Lemma 3.2 *Under the conditions (3.1)–(3.3) and $S \geq \lfloor 3 \ln N \rfloor$, E_n , F_n and R_n could be estimated by*

$$\sum_{n=1}^N \|E_n\| + \sum_{n=1}^N \|R_n\| \leq Q \|u\|_{W^{2,1}(L^2)}, \quad (3.7)$$

$$\sum_{n=1}^N \|F_n\| \leq \|u\|_{W^{1,1}(L^2)}. \quad (3.8)$$

for some constant Q independent from u and τ .

Proof We estimate E_n and R_n in (2.2) and (2.4) by

$$\begin{aligned} \sum_{n=1}^N \|E_n\| &\leq \frac{1}{\tau} \sum_{n=1}^N \int_{t_{n-1}}^{t_n} \|\partial_{tt} u\| (t - t_{n-1}) dt \\ &\leq \int_0^T \|\partial_{tt} u\| dt \leq \|u\|_{W^{2,1}(L^2)}, \\ \sum_{n=1}^N \|R_n\| &\leq Q \sum_{n=1}^N \sum_{k=1}^n \int_{t_{k-1}}^{t_k} \frac{\int_{t_{k-1}}^{t_k} \|\partial_{zz} u(\mathbf{x}, z)\| dz}{(t_n - s)^{\gamma^*}} ds \\ &\leq Q \sum_{k=1}^N \int_{t_{k-1}}^{t_k} \|\partial_{zz} u\| dz \sum_{n=k}^N \int_{t_{k-1}}^{t_k} \frac{ds}{(t_n - s)^{\gamma^*}} \\ &\leq Q \|u\|_{W^{2,1}(L^2)}. \end{aligned}$$

We invoke (3.4) in (2.8) to bound F_n by

$$\begin{aligned} \sum_{n=1}^N \|F_n\| &\leq \sum_{n=1}^N \sum_{k=1}^n |P_{n,k}| \int_{t_{k-1}}^{t_k} \|\partial_t u\| dt \\ &\leq \sum_{k=1}^N \int_{t_{k-1}}^{t_k} \|\partial_t u\| dt \sum_{n=k}^N |P_{n,k}| \\ &\leq \sum_{k=1}^N \int_{t_{k-1}}^{t_k} \|\partial_t u\| dt = \|u\|_{W^{1,1}(L^2)}, \end{aligned} \quad (3.9)$$

which completes the proof. $\square$

3.2 Error estimate

Let $u_n - U_n = u_n - \mathcal{P}_h u_n + \mathcal{P}_h u_n - U_n := \eta_n + \xi_n$, and we bound η_n in the following lemma.

Lemma 3.3 Under the conditions (3.1)–(3.3), η_n could be estimated by

$$\sum_{n=1}^N \left(\|\delta_\tau \eta_n\| + \|\kappa_n \delta_\tau^{\gamma_n} \eta_n\| \right) \leq Q \|u\|_{W^{1,1}(H^2)} h^2 \tau^{-1}. \quad (3.10)$$

Proof By (2.1), we have

$$\sum_{n=1}^N \|\delta_\tau \eta_n\| \leq \frac{1}{\tau} \sum_{n=1}^N \int_{t_{n-1}}^{t_n} \|(\mathcal{I} - \mathcal{P}_h) \partial_t u\| dt \leq Q \|u\|_{W^{1,1}(H^2)} h^2 \tau^{-1}.$$

We estimate $\delta_\tau^{\gamma_n} u$ by

$$\begin{aligned} \sum_{n=1}^N \|\delta_\tau^{\gamma_n} \eta_n\| &= \frac{1}{\tau} \sum_{n=1}^N \left\| \sum_{k=1}^n b_{n,k} (\mathcal{I} - \mathcal{P}_h) \int_{t_{k-1}}^{t_k} \partial_t u dt \right\| \\ &\leq \frac{1}{\tau} \sum_{n=1}^N \left\| \sum_{k=1}^n (b_{n,k} - \hat{b}_{n,k} + \hat{b}_{n,k}) (\mathcal{I} - \mathcal{P}_h) \int_{t_{k-1}}^{t_k} \partial_t u dt \right\| \\ &\leq \frac{1}{\tau} \sum_{n=1}^N \left\| \sum_{k=1}^n \hat{b}_{n,k} (\mathcal{I} - \mathcal{P}_h) \int_{t_{k-1}}^{t_k} \partial_t u dt \right\| \\ &\quad + \frac{1}{\tau} \sum_{n=1}^N \left\| \sum_{k=1}^n P_{n,k} (\mathcal{I} - \mathcal{P}_h) \int_{t_{k-1}}^{t_k} \partial_t u dt \right\|. \end{aligned} \quad (3.11)$$

The first term on the right-hand side of (3.11) can be bounded by

$$\begin{aligned} &\frac{1}{\tau} \sum_{n=1}^N \left\| \sum_{k=1}^n \hat{b}_{n,k} (\mathcal{I} - \mathcal{P}_h) \int_{t_{k-1}}^{t_k} \partial_t u dt \right\| \\ &\leq Q h^2 \tau^{-1} \sum_{n=1}^N \sum_{k=1}^n \int_{t_{k-1}}^{t_k} \|\partial_t u\|_{H^2} dt \int_{t_{k-1}}^{t_k} \frac{ds}{(t_n - s)^{\gamma^*}} \\ &\leq Q h^2 \tau^{-1} \sum_{k=1}^N \int_{t_{k-1}}^{t_k} \|\partial_t u\|_{H^2} dt \sum_{n=k}^N \int_{t_{k-1}}^{t_k} \frac{ds}{(t_n - s)^{\gamma^*}} \\ &\leq Q \|u\|_{W^{1,1}(H^2)} h^2 \tau^{-1}. \end{aligned}$$

We Lemma 3.4 to bound the second term on the right-hand side of (3.11) by

$$\begin{aligned}
 & \frac{1}{\tau} \sum_{n=1}^N \left\| \sum_{k=1}^n P_{n,k} (\mathcal{I} - \mathcal{P}_h) \int_{t_{k-1}}^{t_k} \partial_t u dt \right\| \\
 & \leq Q h^2 \tau^{-1} \sum_{n=1}^N \sum_{k=1}^n |P_{n,k}| \int_{t_{k-1}}^{t_k} \|\partial_t u\|_{H^2} dt \\
 & = Q h^2 \tau^{-1} \sum_{k=1}^N \int_{t_{k-1}}^{t_k} \|\partial_t u\|_{H^2} dt \sum_{n=k}^N |P_{n,k}| \\
 & \leq Q \|u\|_{W^{1,1}(H^2)} h^2 \tau^{-1}.
 \end{aligned}$$

Therefore, we finish the proof. $\square$

Theorem 3.1 *The following error estimate holds for the fast finite element scheme (2.10) for τ small enough*

$$\|u - U\|_{\hat{L}^\infty(L^2)} \leq Q(\tau + h^2),$$

where Q is a constant independent from τ and h .

Proof We subtract (2.10) from (2.9) to get the error equation

$$\begin{aligned}
 & (\delta_\tau(u_n - U_n), \chi_h) + (\kappa_n \delta_\tau^{\gamma_n}(u_n - U_n), \chi_h) \\
 & + (\mathbf{K} \nabla(u_n - U_n), \nabla \chi_h) = -(E_n + \kappa_n(R_n + F_n), \chi_h).
 \end{aligned}$$

We invoke $u_n - U_n = \eta_n + \xi_n$ and set $\chi_h = \xi_n$ to get

$$(\delta_\tau \xi_n, \xi_n) + (\kappa_n \delta_\tau^{\gamma_n} \xi_n, \xi_n) + (\mathbf{K} \nabla \xi_n, \nabla \xi_n) = -(G_n, \xi_n). \quad (3.12)$$

where $G_n := E_n + \kappa_n(R_n + F_n) + \delta_\tau \eta_n + \kappa_n \delta_\tau^{\gamma_n} \eta_n$. We rewrite $\delta_\tau^{\gamma_n} \xi_n$ in the form

$$\delta_\tau^{\gamma_n} \xi_n = b_{n,n} \xi_n + \sum_{k=1}^{n-1} (b_{n,k} - b_{n,k+1}) \xi_k,$$

and use the Cauchy inequality to (3.12) multiplied by 2τ to obtain

$$\begin{aligned}
 & \|\xi_n\|^2 + 2\tau(\kappa_n b_{n,n} \xi_n, \xi_n) + 2\tau(\mathbf{K} \nabla \xi_n, \nabla \xi_n) \\
 & \leq \|\xi_{n-1}\|^2 + 2\tau \sum_{k=1}^{n-1} (\kappa_n (b_{n,k+1} - b_{n,k}) \xi_k, \xi_n) + 2\tau \|G_n\| \|\xi_n\|.
 \end{aligned} \quad (3.13)$$

We split $b_{n,k+1} - b_{n,k}$ by inserting the coefficients $\hat{b}_{n,k+1}$ and $\hat{b}_{n,k}$ given in (2.4)

$$\begin{aligned} |b_{n,k+1} - b_{n,k}| &= |(\hat{b}_{n,k+1} - \hat{b}_{n,k}) + (b_{n,k+1} - \hat{b}_{n,k+1}) - (b_{n,k} - \hat{b}_{n,k})| \\ &\leq (\hat{b}_{n,k+1} - \hat{b}_{n,k}) + |\hat{b}_{n,k+1} - b_{n,k+1}| + |\hat{b}_{n,k} - b_{n,k}| \\ &= (\hat{b}_{n,k+1} - \hat{b}_{n,k}) + |P_{n,k+1}| + |P_{n,k}|. \end{aligned}$$

We apply the monotonicity property of $\{\hat{b}_{n,k}\}_{k=1}^n$ (i.e., $\hat{b}_{n,k+1} > \hat{b}_{n,k}$) and $2|\xi_k \xi_n| \leq \xi_k^2 + \xi_n^2$ to bound the second term on the right-hand side of (3.13) by

$$\begin{aligned} &2\tau \sum_{k=1}^{n-1} (\kappa_n (b_{n,k+1} - b_{n,k}) \xi_k, \xi_n) \\ &\leq \tau \sum_{k=1}^{n-1} (\kappa_n |b_{n,k+1} - b_{n,k}| \xi_k, \xi_k) + \tau \sum_{k=1}^{n-1} (\kappa_n |b_{n,k+1} - b_{n,k}| \xi_n, \xi_n) \\ &\leq \tau \sum_{k=1}^{n-1} (\kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k} + |P_{n,k+1}| + |P_{n,k}|) \xi_k, \xi_k) \\ &\quad + \tau \sum_{k=1}^{n-1} (\kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k} + |P_{n,k+1}| + |P_{n,k}|) \xi_n, \xi_n). \end{aligned} \tag{3.14}$$

By Lemma 3.1, the second sum on the right-hand side of (3.14) could be bounded by

$$\begin{aligned} &\tau \sum_{k=1}^{n-1} (\kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k} + |P_{n,k+1}| + |P_{n,k}|) \xi_n, \xi_n) \\ &\leq \tau \kappa_n \hat{b}_{n,n} \|\xi_n\|^2 + 2\tau \|\xi_n\|^2. \end{aligned}$$

We invoke these estimates in (3.13) and add the resulting equation from $n = 1$ to $n^*(n^* \leq N)$ and cancel the like terms to get

$$\begin{aligned} &\|\xi_{n^*}\|^2 + \tau \sum_{n=1}^{n^*} (\mathbf{K} \nabla \xi_n, \nabla \xi_n) \\ &\leq \tau \sum_{n=1}^{n^*} \sum_{k=1}^{n-1} (\kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k} + |P_{n,k+1}| + |P_{n,k}|) \xi_k, \xi_k) + \tau \sum_{n=1}^{n^*} \|G_n\| \|\xi_n\| \\ &= \tau \sum_{k=1}^{n^*-1} \left(\sum_{n=k+1}^{n^*} \kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k}) \xi_k, \xi_k \right) + \tau \sum_{n=1}^{n^*} \|G_n\| \|\xi_n\| \\ &\quad + \tau \sum_{k=1}^{n^*-1} \left(\sum_{n=k+1}^{n^*} \kappa_n (|P_{n,k}| + |P_{n,k+1}|) \xi_k, \xi_k \right), \end{aligned} \tag{3.15}$$

By

$$\begin{aligned}
 \sum_{n=k+1}^{n^*} \kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k}) &\leq \|\kappa\|_{L^\infty(L^\infty)} \sum_{n=k+1}^{n^*} (\hat{b}_{n,k+1} - \hat{b}_{n,k}) \\
 &= \|\kappa\|_{L^\infty(L^\infty)} \sum_{n=k+1}^{n^*} \frac{1}{\Gamma(1-\gamma(\mathbf{x}, t_n))} \\
 &\quad \times \left(\int_{t_k}^{t_{k+1}} \frac{1}{(t_n - s)^{\gamma(\mathbf{x}, t_n)}} ds - \int_{t_{k-1}}^{t_k} \frac{1}{(t_n - s)^{\gamma(\mathbf{x}, t_n)}} ds \right) \\
 &= \|\kappa\|_{L^\infty(L^\infty)} \sum_{n=k+1}^{n^*} \frac{1}{\Gamma(1-\gamma(\mathbf{x}, t_n))} \int_{t_{k-1}}^{t_k} \int_{t_{n-1}}^{t_n} \partial_z (- (z - s)^{-\gamma(\mathbf{x}, t_n)}) dz ds \\
 &\leq Q \|\kappa\|_{L^\infty(L^\infty)} \int_{t_{k-1}}^{t_k} \sum_{n=k+1}^{n^*} \int_{t_{n-1}}^{t_n} \frac{dz}{(z - s)^{1+\gamma^*}} ds \\
 &= Q \|\kappa\|_{L^\infty(L^\infty)} \int_{t_{k-1}}^{t_k} \int_{t_k}^{t_{n^*}} \frac{dz}{(z - s)^{1+\gamma^*}} ds \\
 &\leq Q \|\kappa\|_{L^\infty(L^\infty)} \int_{t_{k-1}}^{t_k} \frac{ds}{(t_k - s)^{\gamma^*}} = Q \|\kappa\|_{L^\infty(L^\infty)} \tau^{1-\gamma^*} \leq Q,
 \end{aligned}$$

the first right-hand side term of (3.15) could be bounded by

$$\tau \sum_{k=1}^{n^*-1} \left(\sum_{n=k+1}^{n^*} \kappa_n (\hat{b}_{n,k+1} - \hat{b}_{n,k}) \xi_k, \xi_k \right) \leq Q \tau \sum_{k=1}^{n^*-1} (\xi_k, \xi_k).$$

By Lemma 3.4, the third term on the right-hand side of (3.15) can be bounded by

$$\tau \sum_{k=1}^{n^*-1} \left(\sum_{n=k+1}^{n^*} \kappa_n (|P_{n,k}| + |P_{n,k+1}|) \xi_k, \xi_k \right) \leq 2 \|\kappa\|_{L^\infty(L^\infty)} \tau \sum_{k=1}^{n^*-1} (\xi_k, \xi_k).$$

Therefore, (3.15) can be bounded by

$$\|\xi_{n^*}\|^2 + 2\tau \sum_{n=1}^{n^*} (\mathbf{K} \nabla \xi_n, \nabla \xi_n) \leq Q \tau \sum_{k=1}^{n^*-1} \|\xi_k\|^2 + \tau \sum_{n=1}^{n^*} \|G_n\| \|\xi_n\|.$$

We apply the discrete Gronwall inequality to obtain

$$\|\xi_{n^*}\|^2 \leq Q \tau \sum_{n=1}^{n^*} \|G_n\| \|\xi_n\|.$$

Let $\|\xi_{n^*}\| := \max_{1 \leq n \leq N} \|\xi_n\|$ such that $\|\xi_{n^*}\| \leq Q\tau \sum_{n=1}^{n^*} \|G_n\|$, and we use (3.7), (3.8) and (3.10) to estimate G_n as

$$\|\xi_{n^*}\| \leq Q\tau \sum_{n=1}^{n^*} \|G_n\| \leq Q(\tau + h^2),$$

which completes the proof. $\square$

4 Efficient implementation

We intend to develop a fast implementation of the scheme (2.10) by analyzing the structures of the coefficient matrices and then accordingly applying the divide and conquer algorithm for computation. Due to the spatial dependence of variable order, it becomes much more challenging to separate the all-at-once coefficient matrix into the Kronecker product of temporal and spatial parts, and then to approximate the temporal part by a sum of Toeplitz matrices multiplied by diagonal matrices. Novel analysis and matrix splittings are applied to resolve these issues to develop a fast divide and conquer (FDAC) algorithm for the scheme (2.10).

4.1 Structure of coefficient matrices

We first study the properties of the coefficients $\{b_{n,k}\}$. Denote $c_n^s(\mathbf{x}) := \frac{(\bar{\gamma} - \gamma_n)^s}{s! \Gamma(2 - \gamma_n) \tau^{\gamma_n}}$ and $d_k^s := (k+1)^{1-\bar{\gamma}} \ln^s(k+1) - k^{1-\bar{\gamma}} \ln^s k$ such that $b_{n,k}$ for $n-k \geq 1$ can be decomposed by

$$b_{n,k} = \sum_{s=0}^S c_n^s(\mathbf{x}) d_{n-k}^s.$$

Thus $\delta_\tau^{\gamma_n} u_n$ in (2.6) could be expressed as

$$\begin{aligned} \delta_\tau^{\gamma_n} u_n &= b_{n,n} u_n + (b_{n,n-1} - b_{n,n}) u_{n-1} - b_{n,1} u_0 \\ &\quad + \sum_{k=1}^{n-2} \sum_{s=0}^S c_n^s(\mathbf{x}) (d_{n-k}^s - d_{n-k-1}^s) u_k. \end{aligned}$$

Let $\{\phi_i(\mathbf{x})\}_{j=1}^M$ be the basis of S_h satisfying $\phi_i(\mathbf{x}_i) = 1$ and $\phi_i(\mathbf{x}_j) = 0$ for $j \neq i$ where M is the degree of freedom of S_h , $U_n = \sum_{j=1}^M U_j^n \phi_j(\mathbf{x})$, $\mathbf{U} = [(\mathbf{U}^1)^\top, (\mathbf{U}^2)^\top, \dots, (\mathbf{U}^N)^\top]^\top$ with $\mathbf{U}^n = [U_1^n, U_2^n, \dots, U_M^n]^\top$ for $1 \leq n \leq N$, and $\mathbf{F} = [(\mathbf{F}^1)^\top, (\mathbf{F}^2)^\top, \dots, (\mathbf{F}^N)^\top]^\top$ with $\mathbf{F}^n = [F_1^n, F_2^n, \dots, F_M^n]^\top$ and $F_i^n =$

$(f(\mathbf{x}, t_n) + b_{n,1}u_0, \phi_i)$ for $1 \leq i \leq M$ and $1 \leq n \leq N$. Then the all-at-once linear system of (2.10) can be expressed as

$$(\mathbf{A} + \mathbf{I}_N \otimes \mathbf{B})\mathbf{U} = \mathbf{F}. \quad (4.1)$$

Here $\mathbf{B} = (B_{i,j})_{i,j=1}^M$ is the stiffness matrix defined by $B_{i,j} = (\mathbf{K} \nabla \phi_j(\mathbf{x}), \nabla \phi_i(\mathbf{x}))$ and the matrix $\mathbf{A} = (\mathbf{A}_{n,k})_{n,k=1}^N \in \mathbb{R}^{(NM) \times (NM)}$ is a sub-triangular block matrix with zero blocks $\mathbf{A}_{n,k} = \mathbf{0}$ for $k \geq n + 1$ and nonzero blocks $\mathbf{A}_{n,k}$ for $k \leq n$ defined by

$$(\mathbf{A}_{n,k})_{i,j} = \begin{cases} ((\tau^{-1} + b_{n,n})\phi_j, \phi_i), & k = n, \\ ((-\tau^{-1} + (b_{n,n-1} - b_{n,n})\phi_j, \phi_i), & k = n - 1, \\ \sum_{s=0}^S (d_{n-k}^s - d_{n-k-1}^s)(c_n^s(\mathbf{x})\phi_j, \phi_i), & 1 \leq k \leq n - 2. \end{cases}$$

As $c_n^s(\mathbf{x})$ is independent from k and d_{n-k}^s is independent from $\mathbf{x}$, $d_{n-k}^s - d_{n-k+1}^s$ could be separated from the inner product. To derive a fast matrix-vector multiplication algorithm, we define a tri-diagonal matrix $\mathbf{M}_n^s \in \mathbb{R}^{M \times M}$ by $(\mathbf{M}_n^s)_{i,j} = (c_n^s(\mathbf{x})\phi_j, \phi_i)$ such that $\mathbf{A}_{n,k}$ for $n - k \geq 2$ could be expressed by

$$\mathbf{A}_{n,k} = \sum_{s=0}^S (d_{n-k}^s - d_{n-k-1}^s) \mathbf{M}_n^s. \quad (4.2)$$

Then we decompose $\mathbf{A}$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{3,2} & \mathbf{A}_{3,3} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \mathbf{A}_{N-1,N-2} & \mathbf{A}_{N-1,N-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{A}_{N,N-1} & \mathbf{A}_{N,N} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{A}_{N-1,1} & \mathbf{A}_{N-1,2} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{N,1} & \mathbf{A}_{N,2} & \cdots & \mathbf{A}_{N,N-2} & \mathbf{0} & \mathbf{0} \end{bmatrix} := \mathbf{A}^c + \mathbf{A}^L. \quad (4.3)$$

By (4.2), $\mathbf{A}^L$ could be decomposed as

$$\mathbf{A}^L = \sum_{s=0}^S \mathbf{D}^s (\mathbf{T}^s \otimes \mathbf{I}_M), \quad (4.4)$$

where $\mathbf{D}^s$ for $0 \leq s \leq S$ are quasi-diagonal matrices with $\{\mathbf{M}_n^s\}_{n=1}^N$ being their diagonal blocks

$$\mathbf{D}^s = \begin{bmatrix} \mathbf{M}_1^s & & & & \\ & \mathbf{M}_2^s & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \mathbf{M}_N^s \end{bmatrix},$$

and $\mathbf{T}^s$ for $0 \leq s \leq S$ are toeplitz matrices given by

$$\mathbf{T}^s = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & 0 \\ d_2^s - d_1^s & 0 & 0 & \cdots & \cdots & 0 \\ d_3^s - d_2^s & d_2^s - d_1^s & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ d_{N-2}^s - d_{N-3}^s & \ddots & \cdots & \cdots & 0 & 0 \\ d_{N-1}^s - d_{N-2}^s & d_{N-2}^s - d_{N-3}^s & \cdots & d_2^s - d_1^s & 0 & 0 \end{bmatrix}.$$

4.2 FDAC algorithm

Based on the above analysis on the structures of the matrices, we intend to apply the divide and conquer algorithm. Let $N = 2^L$ such that $L = O(\ln N)$, $\mathbf{U} = [\bar{\mathbf{U}}^\top, \overline{\overline{\mathbf{U}}}^\top]^\top$, $\mathbf{F} = [\bar{\mathbf{F}}^\top, \overline{\overline{\mathbf{F}}}^\top]^\top$ and then we divide $\mathbf{A}^c$, $\mathbf{D}^s$ and $\mathbf{T}^s$ into 2×2 blocks

$$\mathbf{A}^c = \begin{bmatrix} \mathbf{A}_{11}^c & \mathbf{0} \\ \mathbf{A}_{21}^c & \mathbf{A}_{22}^c \end{bmatrix}, \mathbf{D}^s = \begin{bmatrix} \bar{\mathbf{D}}^s & \mathbf{0} \\ \mathbf{0} & \overline{\overline{\mathbf{D}}}^s \end{bmatrix}, \mathbf{T}^s = \begin{bmatrix} \mathbf{T}_1^s & \mathbf{0} \\ \mathbf{T}_2^s & \mathbf{T}_1^s \end{bmatrix}.$$

Consequently, (4.1) could be divided into two sub-linear systems as

$$\begin{aligned} & [\mathbf{A}_{11}^c + \sum_{s=0}^S \bar{\mathbf{D}}^s (\mathbf{T}_1^s \otimes \mathbf{I}_M) + \tau \mathbf{I}_{N/2} \otimes \mathbf{S}] \bar{\mathbf{U}} = \bar{\mathbf{F}}, \\ & [\mathbf{A}_{22}^c + \sum_{s=0}^S \overline{\overline{\mathbf{D}}}^s (\mathbf{T}_1^s \otimes \mathbf{I}_M) + \tau \mathbf{I}_{N/2} \otimes \mathbf{S}] \overline{\overline{\mathbf{U}}} \\ & \quad = \overline{\overline{\mathbf{F}}} - \sum_{s=0}^S \overline{\overline{\mathbf{D}}}^s (\mathbf{T}_2^s \otimes \mathbf{I}_M) \bar{\mathbf{U}} - \mathbf{A}_{21}^c \bar{\mathbf{U}}. \end{aligned} \quad (4.5)$$

As each subsystem in (4.5) has a similar structure as (4.1), we could repeat the dividing procedure to obtain the FDAC algorithm presented in Algorithm 1.

We remain to estimate the computational cost of the FDAC algorithm.

Algorithm 1: FDAC algorithm for (4.5).

Data: $\mathbf{D}^s, \mathbf{T}^s, \mathbf{F}$
Result: $\mathbf{U} = \text{FDAC}(\mathbf{D}^s, \mathbf{T}^s, \mathbf{F})$
 $\dim = \text{length}(\mathbf{F})/M$;
if $\dim \leq 2$ **then**
 solve $(\mathbf{A} + \mathbf{I}_N \otimes \mathbf{S})\mathbf{U} = \mathbf{F}$;
else
 $\bar{\mathbf{U}} = \text{FDAC}(\bar{\mathbf{D}}^s, \mathbf{T}_1^s, \bar{\mathbf{F}})$;
 $\bar{\bar{\mathbf{F}}} = \bar{\mathbf{F}} - \sum_{s=0}^S \bar{\mathbf{D}}^s (\mathbf{T}_2^s \otimes \mathbf{I}_M) \bar{\mathbf{U}} - \mathbf{A}_{21}^c \bar{\mathbf{U}}$;
 $\bar{\bar{\mathbf{U}}} = \text{FDAC}(\bar{\bar{\mathbf{D}}}^s, \mathbf{T}_1^s, \bar{\bar{\mathbf{F}}})$;
 $\mathbf{U} = [\bar{\mathbf{U}}^\top, \bar{\bar{\mathbf{U}}}^\top]^\top$.
end if

Lemma 4.1 For any $\mathbf{v} \in \mathbb{R}^{MN/2}$, the matrix-vector multiplication $\bar{\bar{\mathbf{D}}}^s (\mathbf{T}_2^s \otimes \mathbf{I}_M) \mathbf{v}$ could be carried out in $O(MN \ln N)$ operations.

Proof Divide $\mathbf{v}$ into $N/2$ parts $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{N/2}$ with each part having M entries and let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{N/2}] \in \mathbb{R}^{M \times (N/2)}$. Then we have

$$(\mathbf{T}_2^s \otimes \mathbf{I}_M) \mathbf{v} = \text{vec}(\mathbf{V} \mathbf{T}_2^{s\top}) = \text{vec}((\mathbf{T}_2^s \mathbf{V}^\top)^\top). \quad (4.6)$$

Here $\mathbf{I}_M \in \mathbb{R}^{M \times M}$ represents the identity matrix and $\text{vec}(\mathbf{X})$ represents the vectorization of matrix $\mathbf{X}$ which is the column vector formed by arranging all the columns of $\mathbf{X}$. Let $\mathbf{V}^\top = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \dots, \boldsymbol{\omega}_M]$ such that $\mathbf{T}_2^s \mathbf{V}^\top = [\mathbf{T}_2^s \boldsymbol{\omega}_1, \mathbf{T}_2^s \boldsymbol{\omega}_2, \dots, \mathbf{T}_2^s \boldsymbol{\omega}_M]$. By the fast Fourier transform, the multiplication of the Toeplitz matrix $\mathbf{T}_2^s$ and the vector $\boldsymbol{\omega}_i$ could be computed in $O(N \ln N)$ operations, and consequently $\mathbf{Y} = \mathbf{T}_2^s \mathbf{V}^\top$ requires $O(MN \ln N)$ operations. As $\bar{\mathbf{D}}^s$ is a quasi-diagonal matrix with each diagonal block being a tri-diagonal matrix, evaluating $\bar{\bar{\mathbf{D}}}^s \text{vec}(\mathbf{Y})$ requires $O(MN)$ operations, which completes the proof. $\square$

Theorem 4.1 The all-at-once linear system (4.1) could be implemented in $O(MN \ln^3 N)$ operations by the FDAC Algorithm 1.

Proof As $\mathbf{A}_{21}^c$ has only one nonzero block of order M on its up and right position, the third term on the right-hand side of the second sub-linear system in (4.5) could be computed in $O(M)$ operations. The second right-hand side term could be computed in $O(MN \ln^2 N)$ operations by Lemma 4.1. Consequently, computing the right-hand side terms of the second sub-linear system of (4.5) requires $O(MN \ln^2 N)$ operations, and thus the matrix-vector multiplications in repeating the dividing procedure as (4.5) could be computed by

$$\begin{aligned}
 &O(MN \ln^2 N) + 2 \times O\left(\frac{MN}{2} \ln^2 \frac{N}{2}\right) + \dots + 2^L \times O\left(\frac{MN}{2^L} \ln^2 \frac{N}{2^L}\right) \\
 &= O(MN \ln^3 N),
 \end{aligned}$$

which completes the proof. $\square$

5 Numerical experiments

We test the effectiveness and efficiency of the fast finite element method (2.10) by comparing it with the time-stepping solver (TSS) for the standard finite element method: find $\hat{U}_n \in S_h$ with $U_0 = \mathcal{P}_h u_0$ such that for $n = 1, 2, \dots, N$

$$(\delta_\tau \hat{U}_n, \chi_h) + (\kappa_n \hat{\delta}_\tau^{\gamma(x, t_n)} \hat{U}_n, \chi_h) + (\mathbf{K} \nabla \hat{U}_n, \nabla \chi_h) = (f_n, \chi_h), \quad \forall \chi_h \in S_h \quad (5.1)$$

where $\hat{\delta}_\tau^{\gamma_n} u_n := \sum_{k=1}^n \hat{b}_{n,k} (u_k - u_{k-1})$ denotes the standard $L1$ discretization of ${}_0^C D_t^{\gamma(x, t)} u$ at $t = t_n$. Due to the space dependence of the variable order, the coefficients $\{\hat{b}_{n,k}\}$ in (2.4) could not be split from the inner product and $O(MN^2)$ operations are required to solve (5.1) by TSS, which is computationally expensive for large N .

We measure the errors and convergence rates of the scheme (2.10) as well as the CPU times (in seconds) of the TSS for (5.1) and the FDAC algorithm for (2.10). As the spatial discretization is standard, we only measure the temporal convergence rates of the two methods in the sense

$$\|u - \hat{U}\|_{\hat{L}^\infty(L^2)} \leq Q\tau^{\hat{\nu}}, \quad \|u - U\|_{\hat{L}^\infty(L^2)} \leq Q\tau^{\nu}.$$

All experiments are performed on Matlab 2016b on a computer with Intel(R) Core(TM) i5-6500U, CPU 3.2GHz and 8.00GB RAM. The symbol “\” indicates that the code has already run for more than 3 h and is terminated.

5.1 One-dimensional problems

We perform numerical experiments for model (1.1) in one space dimension involving both singular and nonsingular solutions. We set $\Omega \times (0, T] = [0, 1] \times (0, 1]$ with singular solutions near $t = 0$, $\kappa(x, t) = 10 + xt$, $K = 0.01$, $f(x, t) = 1$.

5.1.1 Comparison of model (1.1) with different kinds of variable orders

We present solutions of model (1.1) with constant fractional order $\gamma = \gamma_0$, time-dependent variable order $\gamma = \gamma_0 \cos(\pi t/2)$ and space-time-dependent variable order $\gamma = \gamma_0 \sin(\pi x/2) \cos(\pi t/2)$ in Fig. 1, from which we observe that the solutions of model (1.1) with constant fractional order and time-dependent variable order are similar, while the solution of model (1.1) with space-time-dependent variable order has salient differences in comparison with them and thus have potential to model more complex phenomena. This indicates that the space-time-dependent variable order could improve the modeling capability of time-fractional problems.

5.1.2 Model (1.1) with singular solutions

In this experiment we test the performance of FDAC for model (1.1) by setting the variable-order $\gamma(x, t) = \gamma_0(x + 1)/2 \cos(\pi t/2)$ where the parameter γ_0 coincides with the definition (3.2) for $\gamma(x, t)$. We choose $\gamma_0 = 0.1, 0.5$ and 0.9 respectively,

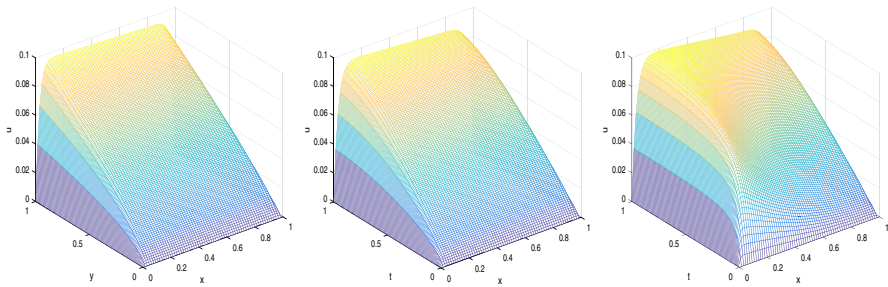


Fig. 1 Solutions of (1.1) with constant fractional order $\gamma = \gamma_0$ (left), time-dependent variable order $\gamma = \gamma_0 \cos(\pi t/2)$ (middle) and space-time-dependent variable order $\gamma = \gamma_0 \sin(\pi x/2) \cos(\pi t/2)$ (right)

which corresponds to different singularities of the solutions at $t = 0$ characterized by the index p in (3.1). As the exact solutions are not available, numerical solutions computed by the TSS under $\tau = 2^{-12}$ and $h = 2^{-8}$ serve as the reference solutions. We compare the accuracy of the FDAC algorithm for (2.10) and the TSS for (5.1) and present the numerical results in Table 1 and Fig. 2, from which we observe that both methods generate the same solution curves with the same accuracy under different singularities of the solutions, as well as the same first-order temporal convergence rates as proved in Theorem 3.1.

Table 1 Errors and convergence rates with different γ_0 for Example 5.1.2

γ_0	N	$\ u - \hat{U}\ _{\hat{L}^\infty(0,T;L^2)}$	$\hat{\nu}$	$\ u - U\ _{\hat{L}^\infty(0,T;L^2)}$	ν
0.1	2^4	7.9540E-3	—	7.9540E-3	—
	2^5	4.5895E-3	0.79	4.5895E-3	0.79
	2^6	2.4406E-3	0.91	2.4406E-3	0.91
	2^7	1.2462E-3	0.97	1.2462E-3	0.97
	2^8	6.1467E-4	1.02	6.1467E-4	1.02
0.5	2^4	5.2271E-3	—	5.2271E-3	—
	2^5	3.0232E-3	0.79	3.0232E-3	0.79
	2^6	1.7142E-3	0.82	1.7142E-3	0.82
	2^7	9.2288E-4	0.89	9.2288E-4	0.89
	2^8	4.7149E-4	0.97	4.7149E-4	0.97
0.9	2^4	2.5275E-3	—	2.5275E-3	—
	2^5	1.5021E-3	0.75	1.5021E-3	0.75
	2^6	8.8278E-4	0.77	8.8278E-4	0.77
	2^7	4.9364E-4	0.84	4.9364E-4	0.84
	2^8	2.6501E-4	0.90	2.6501E-4	0.90

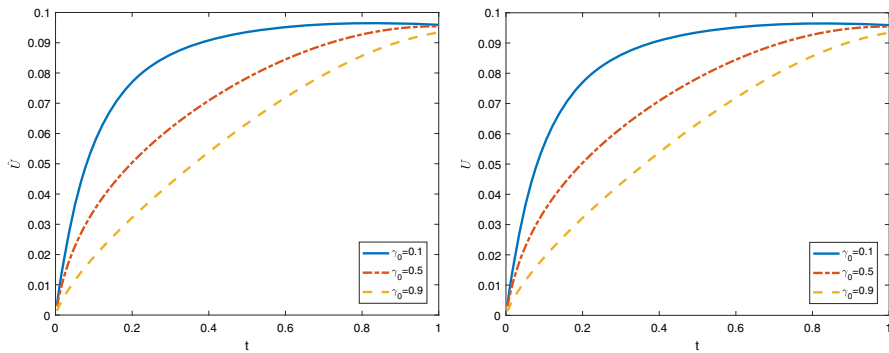


Fig. 2 Numerical solutions of Example 5.1.2 at $x = 0.5$ with different γ_0 solved by TSS (left) and FDAC (right)

5.1.3 Model (1.1) with smooth solutions

Let $\Omega \times (0, T] = [0, 1] \times (0, 1]$, the smooth solution $u = t^2 \sin(\pi x)$, $\kappa(x, t) = (1 + xt/2)$, $\gamma(x, t) = 0.5 + 0.25t \sin(2\pi x)$ and the diffusion coefficient $K = 0.01$. The right-hand side term $f(x, t)$ is evaluated accordingly. We compare the accuracy of the FDAC algorithm for (2.10) and the TSS of (5.1) by setting $h = 2^{-6}$ and present the numerical results in Table 2 and Fig. 3, from which we obtain the same conclusions as Example 5.1.2. We then compare the CPU times CPU_U of solving (2.10) by FDAC and $CPU_{\hat{U}}$ of solving (5.1) by TSS under $h = 2^{-4}$ in Table 3, which indicates that the FDAC algorithm is much more efficient than the TSS.

5.2 Two-dimensional problems

We perform numerical experiments for model (1.1) in two space dimensions involving both the piecewise constant variable order and a more generalized space-time variable order.

5.2.1 Model (1.1) with a piecewise constant variable order

Let $\Omega \times (0, T] = (0, 1)^2 \times (0, 1]$, $u(x, y, t) = (t^2 + 1)x \sin(\pi x)y \sin(\pi y)$, $\kappa(x, y, t) = 1 + 0.5xyt$ and $\mathbf{K} = \text{diag}(0.01, 0.01)$. The variable order $\gamma = \gamma(x, y)$

Table 2 Errors and convergence rates for Example 5.1.3

N	$\ u - \hat{U}\ _{\hat{L}^\infty(L^2)}$	$\hat{v}$	$\ u - U\ _{\hat{L}^\infty(L^2)}$	v
2^4	1.0287E-2	—	1.0287E-2	—
2^5	4.4424E-3	1.21	4.4424E-3	1.21
2^6	1.9469E-3	1.19	1.9469E-3	1.19
2^7	8.7313E-4	1.16	8.7313E-4	1.16
2^8	4.1425E-4	1.08	4.1425E-4	1.08

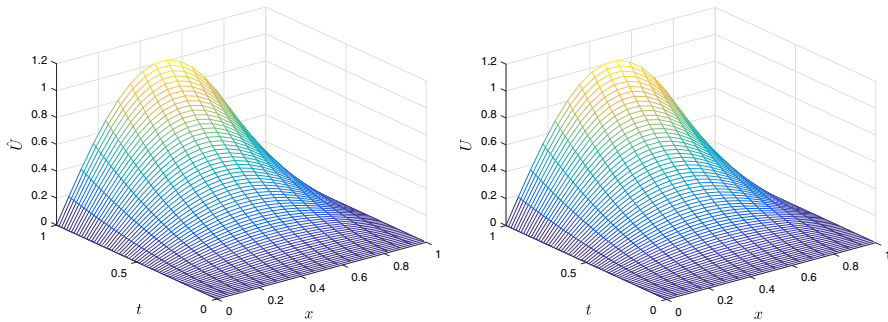


Fig. 3 Numerical solutions of Example 5.1.3 solved by TSS (left) and FDAC (right)

takes the form of a piecewise constant function such that $\gamma(x, y) = 0.55$ on $(x, y) \in (0, 0.5] \times (0, 0.5]$, $\gamma(x, y) = 0.45$ on $(x, y) \in (0.5, 1) \times (0, 0.5] \cup (0, 0.5] \times (0.5, 1)$ and $\gamma(x, y) = 0.75$ on $(x, y) \in (0.5, 1) \times (0.5, 1)$. The right-hand side term $f(x, y, t)$ is computed correspondingly.

We present errors and convergence rates under both FDAC and TSS in Table 4, as well as their CPU times in Table 5, from which we observe that the fast method FDAC has the same accuracy as TSS but is much more efficient than the TSS, which indicates that the proposed fast method is applicable for model (1.1) with piecewise constant variable order, which, in comparison with the generalized case in Section 5.2.2, is more tractable over the domain of interest for a particular application field.

5.2.2 Model (1.1) with a generalized space-time variable order

In this subsection we set $\gamma(x, y, t) = 0.5 + 0.25t \sin(2\pi x) \sin(2\pi y)$ and the other data are the same as those in Section 5.2.1. Similar to previous examples, both methods have the same accuracy and convergence rates as shown in Table 6, and generate the same solution patterns as shown in Fig. 4. Furthermore, we plot the CPU times of TSS and FDAC in Fig. 5, from which we observe that the CPU time of TSS grows quadratically with respect to N , while the CPU time of FDAC algorithm grows linearly as predicted in Theorem 4.1.

6 Concluding remarks

In this work we developed a fast finite element method for the time-fractional diffusion equation with a space-time-dependent variable order. We approximated the L1

Table 3 CPU times (in seconds) under TSS and FDAC for Example 5.1.3

N	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}
$CPU_{\hat{U}}$	2.74	10.8	42.7	168	674	2692	\	\	\	\	\
CPU_U	0.12	0.19	0.40	0.87	1.87	3.86	7.91	20	58	296	1654

Table 4 Errors and convergence rates for Example 5.2.1 with $h = 2^{-6}$

N	$\ u - \hat{U}\ _{\hat{L}^\infty(0,T;L^2)}$	$\hat{\nu}$	$\ u - U\ _{\hat{L}^\infty(0,T;L^2)}$	ν
2^4	1.5131E-2	—	1.5131E-2	—
2^5	7.3551E-3	1.04	7.3551E-3	1.04
2^6	3.5693E-3	1.04	3.5693E-3	1.04
2^7	1.7240E-3	1.05	1.7240E-3	1.05
2^8	8.2262E-4	1.07	8.2262E-4	1.07

Table 5 CPU times (in seconds) for Example 5.2.1 with $h = 2^{-4}$

N	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
$CPU_{\hat{U}}$	40.5	164	616	2459	9904	\	\	\
CPU_U	1.51	2.31	5.01	10.7	23.1	66.4	105	252

Table 6 Errors and convergence rates for Example 5.2.2 with $h = 2^{-6}$

N	$\ u - \hat{U}\ _{\hat{L}^\infty(0,T;L^2)}$	$\hat{\nu}$	$\ u - U\ _{\hat{L}^\infty(0,T;L^2)}$	ν
2^4	4.8342E-3	—	4.8342E-3	—
2^5	2.3533E-3	1.04	2.3533E-3	1.04
2^6	1.1476E-3	1.04	1.1476E-3	1.04
2^7	5.6092E-4	1.03	5.6092E-4	1.03
2^8	2.7660E-4	1.02	2.7660E-4	1.02

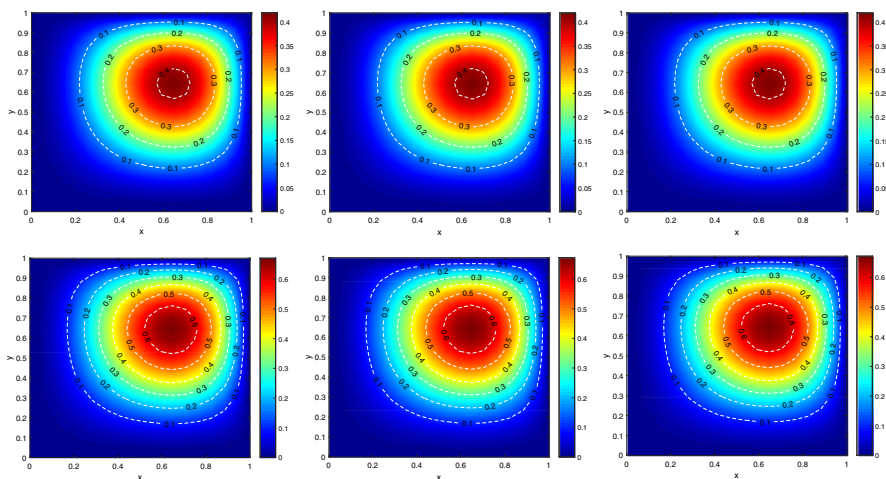


Fig. 4 Numerical solutions for Example 5.2.2 at $t = T/2$ (the first row) and $t = T$ (the second row). First column: the exact solutions; Second column: numerical solutions of TSS; Third column: numerical solutions of FDAC

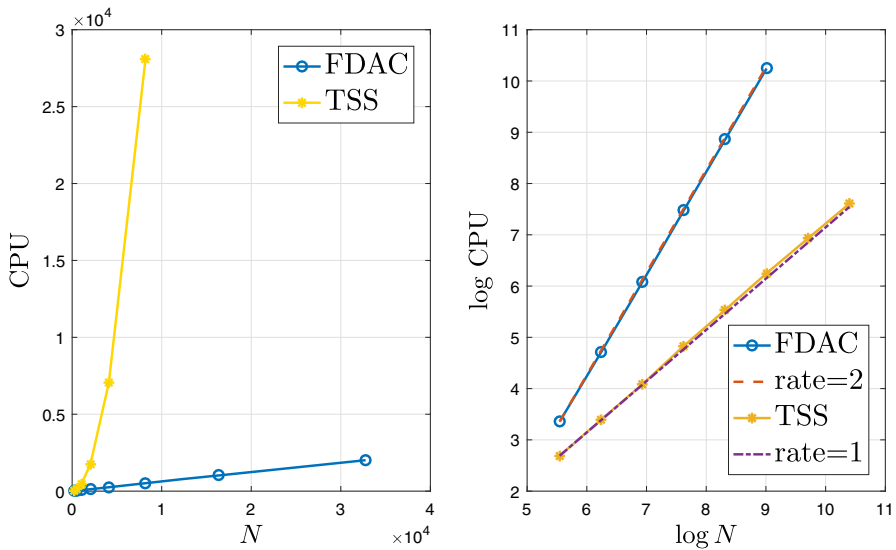


Fig. 5 CPU times of TSS and FDAC algorithms with $h = 2^{-4}$ for Example 5.2.2 (left) and those under the log-log coordinate system (right)

coefficients of the fractional derivative by Taylor expansion and proved the optimal-order error estimate of the approximated scheme. A fast DAC algorithm was accordingly derived, which reduced the computational cost from $O(MN^2)$ to $O(MN \ln^3 N)$. Various numerical experiments were performed to test the proposed methods.

Below we use the enhanced oil recovery [33–37] as an example application to outline the motivation of our study in this paper. To improve the efficiency of secondary recovery, in which water is injected into oil reservoir to displace the resident oil out of the reservoir, surfactant or other chemicals may be mixed with water as the injecting fluid to “wash” the oil and displace the resident oil out. To model the displacement process, let $c(\mathbf{x}, t)$ be the concentration of an invading fluid and let $p(\mathbf{x}, t)$ and $\mathbf{u}(\mathbf{x}, t)$ be the pressure and Darcy velocity of the total fluid mixture. Under the assumption that the injected solvent fluid and the resident oil fluid is fully miscible and incompressible, the governing system of PDEs for modeling the miscible displacement of one incompressible fluid by another in a two-dimensional porous medium reservoir Ω with a nonuniform local elevation over a time period of $[0, T]$ can be formulated as follows [34–37]

$$\begin{aligned} \phi(\partial_t c + \kappa(\mathbf{x}, t)_0^C D_t^{\gamma(\mathbf{x}, t)} c) + \nabla \cdot (\mathbf{u}c - \mathbf{D}(\mathbf{x}, \mathbf{u})\nabla c) &= \bar{c}q, \\ \nabla \cdot \mathbf{u} &= q, \quad \mathbf{u} = -\frac{\mathbf{K}}{\mu(c)}(\nabla p - \rho g \nabla d), \quad (\mathbf{x}, t) \in \Omega \times (0, T]. \end{aligned} \quad (6.1)$$

The last two equations are derived by combining the mass conservation for the fluid mixture with the incompressibility condition and Darcy’s law. Here $\mathbf{K}$ is the intrinsic permeability tensor of the medium, μ is the concentration-dependent viscosity of the

fluid mixture, ρ is the density of the fluid mixture, g is the magnitude of gravitational acceleration, $d(\mathbf{x})$ is the reservoir depth, $q(\mathbf{x}, t)$ is the external source and sink term that accounts for the effect of injection and production wells.

The first PDE in (6.1) governs the transport of the invading fluid, $\phi(\mathbf{x})$ is the porosity of the medium, $\mathbf{D}(\mathbf{x}, \mathbf{u})$ is the diffusion-dispersion tensor that accounts for the effect of the molecular diffusion and the velocity-dependent mechanical dispersion, $\bar{c}(\mathbf{x}, t)$ is either the specified concentration of the injected fluid at injection wells or $\bar{c}(\mathbf{x}, t) = c(\mathbf{x}, t)$ is the resident concentration at production wells. The time-fractional derivative in the PDE accounts for the movement of the portion of the particles of the injected fluid that get absorbed to the micropores in the geological formation and is experiencing subdiffusive advective transport [38, 39]. Thus, the travel time of the adsorbed injected solvent particles may deviate from that of the particles in the bulk fluid phase [40], leading to an anomalous subdiffusive transport process that is characterized by a sub-linear growth of the particle's mean square displacement with respect to the time t [41]. This motivates the use of a time-fractional PDE to model subdiffusive advective transport PDE [42, 43] as an alternative to the conventional integer-order diffusive advective transport PDE [35, 37]. However, the conventional time-fractional diffusion PDE admits solution with initial singularity [9, 44] that is not physically relevant because it does not capture the Fickian diffusive behavior near the initial time while it can successfully capture the subdiffusive behavior for sufficiently large time. Instead, the two time-scale time-fractional diffusion PDE in (6.1) provides a physically relevant model of subdiffusive transport process [14, 45, 46]. Here $\kappa(\mathbf{x}, t)$ is the partition coefficient, namely $\kappa/(1 + \kappa)$ portion of the total invading fluid is absorbed to the porous medium and the rest stays in the bulk fluid mixture.

In oil recovery oil reservoirs often have insufficient permeability due to the existence of micropores, resulting in a large amount of adsorbed oil mass and significantly decreased flow rate to the wellbore and so significantly reduced recovery efficiency. Hydraulic fracturing is often used to increase the pore sizes and so the permeability of the porous media [47, 48]. The change of the structure of the porous media results in the change of the fractal dimension of the media [7] via the Hurst index, which in turn leads to the change of the order γ of the variable-order time-fractional diffusive advective transport PDE in (6.1) [18, 19, 45]. Furthermore, since the hydrofracturing is typically heterogeneous in space, the variable fractional order depends on both time and space, i.e., $\gamma = \gamma(\mathbf{x}, t)$ and so the form of the transport PDE in (6.1). A sequentially decoupled time-stepping discretization was developed and analyzed for the integer-order analogue of problem (6.1) [35, 37, 49]. This allows us to focus on the fast solution of the time-fractional subdiffusive advective transport PDE in (6.1). Since the advection term in the equation is local, we drop the term in the equation and focus on the variable-order time-fractional diffusion PDE (1.1) in our study in this paper.

Author contribution Xiangcheng Zheng and Jinhong Jia: methodology, writing—original draft. Hong Wang: conceptualization, review and editing.

Funding This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 11971272 and 12001337, by the Natural Science Foundation of Shandong Province under Grant ZR2019BA026, and by the National Science Foundation under Grant No. DMS-2012291.

Availability of supporting data All data generated or analyzed during this study are included in this article.

Declarations

Ethical approval Not applicable

Competing interests The authors declare no competing interests.

References

- Deng, W.: Finite element method for the space and time fractional Fokker-Planck equation. *SIAM J. Numer. Anal.* **47**, 204–226 (2009)
- K. Diethelm, The analysis of fractional differential equations. An application-oriented exposition using differential operators of Caputo type. *Lecture Notes in Mathematics*, 2004. Springer-Verlag, Berlin, 2010
- Gu, X., Sun, H., Zhao, Y., Zheng, X.: An implicit difference scheme for time-fractional diffusion equations with a time-invariant type variable order. *Appl. Math. Lett.* **120**, 107270 (2021)
- Jin, B., Li, B., Zhou, Z.: Subdiffusion with a time-dependent coefficient: Analysis and numerical solution. *Math. Comp.* **88**, 2157–2186 (2019)
- Li, D., Liao, H., Sun, W., Wang, J., Zhang, J.: Analysis of L1-Galerkin FEMs for time-fractional nonlinear parabolic problems. *Commun. Comput. Phys.* **24**, 86–103 (2018)
- Li, Y., Wang, H., Zheng, X.: A viscoelastic Timoshenko beam model: regularity and numerical approximation. *J. Sci. Comput.* to appear
- Meerschaert, M.M., Sikorskii, A.: *Stochastic Models for Fractional Calculus*. De Gruyter Studies in Mathematics (2011)
- Obembe, A., Hossain, M., Abu-Khamsin, S.: Variable-order derivative time fractional diffusion model for heterogeneous porous media. *J. Petrol. Sci. Eng.* **152**, 391–405 (2017)
- Stynes, M., O’Riordan, E., Gracia, J.L.: Error analysis of a finite difference method on graded mesh for a time-fractional diffusion equation. *SIAM J. Numer. Anal.* **55**, 1057–1079 (2017)
- Zayernouri, M., Karniadakis, G.: Discontinuous Spectral Element Methods for Time- and Space-Fractional Advection Equations. *SIAM J. Sci. Comput.* **36**, B684–B707 (2014)
- Zhao, X., Hu, X., Cai, W., Karniadakis, G.E.: Adaptive finite element method for fractional differential equations using hierarchical matrices. *Comput. Meth. Appl. Mech. Engrg.* **325**, 56–76 (2017)
- Garrappa, R., Giusti, A., Mainardi, F.: Variable-order fractional calculus: A change of perspective. *Commun. Nonlinear Sci. Numer. Simul.* **102**, 105904 (2021)
- Feng, L., Turner, I., Perré, P., Burrage, K.: An investigation of nonlinear time-fractional anomalous diffusion models for simulating transport processes in heterogeneous binary media. *Commun. Nonlinear Sci. Numer. Simul.* **92**, 105454 (2021)
- Schumer, R., Benson, D.A., Meerschaert, M.M., Baeumer, B.: Fractal mobile/immobile solute transport. *Water Resources Res.* **39**, 1–12 (2003)
- Zaky, M., Bockstal, K., Taha, T., Suragan, D., Hendy, A.: An L1 type difference/Galerkin spectral scheme for variable-order time-fractional nonlinear diffusion-reaction equations with fixed delay. *J. Comput. Appl. Math.* **420**, 114832 (2023)
- Zheng, X., Wang, H.: A time-stepping finite element method for a time-fractional partial differential equation of hidden-memory space-time variable order. *Elect. Trans. Numer. Anal.* **55**, 652–670 (2022)
- Zheng, X., Wang, H.: A time-fractional partial differential equation with a space-time dependent hidden-memory variable order: analysis and approximation. *BIT Numer. Math.* **61**, 1453–1481 (2021)
- Lorenzo, C., Hartley, T.: Variable order and distributed order fractional operators. *Nonlinear Dyn.* **29**, 57–98 (2002)
- Sun, H., Chang, A., Zhang, Y., Chen, W.: A review on variable-order fractional differential equations: Mathematical foundations, physical models, numerical methods and applications. *Fract. Calc. Appl. Anal.* **22**, 27–59 (2019)
- Zeng, F., Zhang, Z., Karniadakis, G.: A generalized spectral collocation method with tunable accuracy for variable-order fractional differential equations. *SIAM J. Sci. Comput.* **37**, A2710–A2732 (2015)

21. Bai, Z., Lu, K., Pan, J.: Diagonal and Toeplitz splitting iteration methods for diagonal-plus-Toeplitz linear systems from spatial fractional diffusion equations. *Numer. Linear Algebra Appl.* **24**, e2093 (2017)
22. Diethelm, K.: An efficient parallel algorithm for the numerical solution of fractional differential equations. *Fract. Calc. Appl. Anal.* **14**, 475–490 (2011)
23. Fu, H., Ng, M., Wang, H.: A divide-and-conquer fast finite difference method for space-time fractional partial differential equation. *Comput. Math. Appl.* **73**, 1233–1242 (2017)
24. Jiang, S., Zhang, J., Zhang, Q., Zhang, Z.: Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations. *Commun. Comput. Phys.* **21**, 650–678 (2017)
25. Pang, H., Qin, H., Sun, H.: All-at-once method for variable-order time fractional diffusion equations. *Numer. Algor.* **90**, 31–57 (2021)
26. Jia, J., Zheng, X., Fu, H., Dai, P., Wang, H.: A fast method for variable-order space-fractional diffusion equations. *Numer. Algor.* **85**, 1519–1540 (2020)
27. Fang, Z., Sun, H., Wang, H.: A fast method for variable-order Caputo fractional derivative with applications to time-fractional diffusion equations. *Comput. Math. Appl.* **80**, 1443–1458 (2020)
28. Zhang, J., Fang, Z., Sun, H.: Exponential-sum-approximation technique for variable-order time-fractional diffusion equations. *J. Appl. Math. Comput.* **68**, 323–347 (2022)
29. Zhang, J., Fang, Z., Sun, H.: Robust fast method for variable-order time-fractional diffusion equations without regularity assumptions of the true solutions. *Appl. Math. Comput.* **430**, 127273 (2022)
30. Zhang, J., Fang, Z., Sun, H.: Fast second-order evaluation for variable-order Caputo fractional derivative with applications to fractional sub-diffusion equations. *Numer. Math. Theory Methods Appl.* **15**, 200–226 (2022)
31. Adams, R.A., Fournier, J.J.: *Sobolev Spaces*. Elsevier, San Diego (2003)
32. Thomée, V.: *Galerkin Finite Element Methods for Parabolic Problems*. Lecture Notes in Mathematics, vol. 1054. Springer-Verlag, New York (1984)
33. Balhoff, M.: *An Introduction to Multiphase, Multicomponent Reservoir Simulation*. Developments in Petroleum Science, vol. 75. Elsevier (2022)
34. Bear, J.: *Dynamics of Fluids in Porous Media*. Elsevier, New York (1972)
35. Douglas, J., Ewing, R., Wheeler, M.: A time-discretization procedure for a mixed finite element approximation of miscible displacement in porous media. *RAIRO Anal. Numer.* **17**, 249–265 (1983)
36. R.E. Ewing (ed.), *The Mathematics of Reservoir Simulation*, in *Research Frontiers in Applied Mathematics*, 1, SIAM Philadelphia, 1984
37. Wang, H., Liang, D., Ewing, R., Lyons, S., Gin, G.: An approximation to miscible fluid flows in porous media with point sources and sinks by a Eulerian-Lagrangian localized adjoint method and mixed finite element methods. *SIAM J. Sci. Comput.* **22**, 561–581 (2000)
38. Sharma, A., Namsani, S., Singh, J.: Molecular simulation of shale gas adsorption and diffusion in inorganic nanopores. *Mol. Simul.* **41**, 414–422 (2015)
39. Ungerer, P., Collett, J., Yiannourakou, M.: Molecular modeling of the volumetric and thermodynamic properties of Kerogen: influence of organic type and maturity. *Energy & Fuels* **29**, 91–105 (2015)
40. Zhokh, A., Strizhak, P.: Non-Fickian diffusion of methanol in mesoporous media: geometrical restrictions or adsorption-induced? *J. Chem. Phys.* **146**, 124704 (2017)
41. Chepizhko, O., Peruani, F.: Diffusion, subdiffusion, and trapping of active particles in heterogeneous media. *Phys. Rev. Lett.* **111**, 160604 (2013)
42. Dentz, M., Cortis, A., Scher, H., Berkowitz, B.: Time behavior of solute transport in heterogeneous media: transition from anomalous to normal transport. *Adv. Water Resour.* **27**, 155–173 (2004)
43. Metzler, R., Klafter, J.: The random walk's guide to anomalous diffusion: A fractional dynamics approach. *Phys. Rep.* **339**, 1–77 (2000)
44. Sakamoto, K., Yamamoto, M.: Initial value/boundary value problems for fractional diffusion-wave equations and applications to some inverse problems. *J. Math. Anal. Appl.* **382**, 426–447 (2011)
45. Zheng, X., Wang, H.: 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**, 1522–1545 (2021)
46. Zhang, Y., Green, C., Baeumer, B.: Linking aquifer spatial properties and non-Fickian transport in mobile-immobile like alluvial settings. *J. Hydrol.* **512**, 315–331 (2014)
47. Gandossi, L., Von Estorff, U.: An overview of hydraulic fracturing and other formation stimulation technologies for shale gas production. Scientific and Technical Research Reports, Joint Research Centre

- of the European Commission; Publications Office of the European Union (2015). <https://doi.org/10.2790/379646>
48. King, G.: Hydraulic fracturing 101: What every representative, environmentalist, regulator, reporter, investor, university researcher, neighbor and engineer should know about estimating frac risk and improving frac performance in unconventional gas and oil wells. SPE Hydraulic Fracturing Technology Conference, SPE 152596, Woodlands, Texas, February 6-8 (2012)
 49. Wang, H.: An optimal-order error estimate for a family of ELLAM-MFEM approximations to porous medium flow. *SIAM J. Numer. Anal.* **46**, 2133–2152 (2008)
 50. Pang, H., Sun, H.: A fast algorithm for the variable-order spatial fractional advection-diffusion equation. *J. Sci. Comput.* **87**, 15 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.