Journal of Scientific Computing (2022) 91:54
https://doi.org/10.1007/s10915-022-01820-z

Numerical Analysis of a Fast Finite Element Method for a Hidden-Memory Variable-Order Time-Fractional Diffusion Equation

Jinhong Jia[1] · Hong Wang[2] · Xiangcheng Zheng[3]

Received: 19 November 2021 / Revised: 23 February 2022 / Accepted: 4 March 2022

Abstract
We investigate a fast finite element scheme to a hidden-memory variable-order time-fractional diffusion equation. Different from the traditional L1 methods, a fast approximation to the hidden-memory variable-order fractional derivative is derived to reduce the computational cost of generating coefficients from $O(N^2)$ to $O(N \log N)$, where N refers to the number of time steps. We then develop different techniques from the analysis of L1 methods to prove error estimates for the corresponding fast fully-discrete finite element scheme. Furthermore, a fast divide and conquer algorithm is proposed to reduce the complexity of solving the linear systems from $O(MN^2)$ to $O(MN \log^2 N)$ where M stands for the spatial degree of freedom. Numerical experiments are presented to substantiate the theoretical results.

Keywords Variable-order time-fractional diffusion equation · Hidden-memory · Finite element method · Optimal-order error estimate · Divide and conquer

Mathematics Subject Classification 35R11 · 65M15 · 65M60

1 Introduction

Time-fractional diffusion equations with constant fractional order have been widely applied in various fields and have attracted extensive mathematical and numerical investigations [2–5,

✉ Xiangcheng Zheng
zhengxch@math.pku.edu.cn

Jinhong Jia
jhjia@sdnu.edu.cn

Hong Wang
hwang@math.sc.edu

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

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

3 School of Mathematical Sciences, Peking University, Beijing 100871, China

Published online: 04 April 2022

10, 12, 14–17, 28]. However, the corresponding studies for their variable-order counterparts are far from well developed due to the significant differences caused by the variable fractional order. For instance, in numerical computations of time-fractional diffusion equations of constant fractional order, the translation invariance property of the convolution in fractional operators admits developing fast solution techniques, e.g., the Toeplitz-matrix-based fast preconditioned methods [13, 21], the divide and conquer (DAC) method [7, 13], and the exponential sum approximation methods [11, 32]. However, there are very few results for fast evaluations of variable-order time-fractional problems.

In some very recent works, fast algorithms for time-fractional diffusion problems involving the following variable-order fractional derivative [19, 29]

$$\bar{\partial}_t^{\alpha(t)} g(t) := \frac{1}{\Gamma(1-\alpha(t))} \int_0^t \frac{\partial_s g(s)}{(t-s)^{\alpha(t)}} ds,$$

where $\Gamma(\cdot)$ represents the Euler Gamma function, have been developed. A fast finite difference method on a shifted binary block partition [6, 8] and an all-at-once method by low-rank matrix approximation and DAC method [20] for a variable-order subdiffusion equation were also developed. A preconditioned fast DAC (fDAC) method was proposed in [9] for a two-term variable-order time-fractional diffusion equation. In [32] an exponential-sum-approximation, which approximates the weak-singular kernel directly by exponential functions without using its Laplace transform, has been proposed and analyzed, which extends the sum-of-exponential method in, e.g. [11], to treat the variable-order fractional derivative. However, the corresponding investigations for the time-fractional problems consisting the hidden-memory variable-order fractional operators [19, 24]

$$ {}_0I_t^{1-\alpha(t)} g(t) := \int_0^t \frac{g(s)ds}{\Gamma(1-\alpha(s))(t-s)^{\alpha(s)}}, \quad \partial_t^{\alpha(t)} g(t) := {}_0I_t^{1-\alpha(t)} g'(t) \tag{1.1}$$

are rarely found in the literature.

Compared with $\bar{\partial}_t^{\alpha(t)}$, the hidden-memory fractional derivative operator $\partial_t^{\alpha(t)}$ has salient features. For any fixed $t \in [0, T]$, the convolution kernel in $\bar{\partial}_t^{\alpha(t)}$ could be integrated in a closed form as in the constant-order case, and the coefficients of its discretization schemes such as the L1 method [18, 25] retains monotonicity that is critical in error estimates. However, $\partial_t^{\alpha(t)}$ does not enjoy these benefits due to the impact of the hidden memory, which makes the mathematical and numerical analysis intricate. In [34] the well-posedness and regularity of a hidden-memory variable-order time-fractional diffusion equation were proved, based on which a fully discrete numerical scheme incorporated with the L1 discretization in time was proposed and analyzed. However, the corresponding fast algorithms remain untreated in the literature to our best knowledge.

Motivated by the above discussions, we investigate fast numerical methods for the following hidden-memory variable-order time-fractional diffusion equation with $0 \le \alpha(t) \le \alpha^* < 1$ [27, 35, 36]

$$\begin{aligned} &\partial_t u(\boldsymbol{x}, t) + \kappa(t)\partial_t^{\alpha(t)} u(\boldsymbol{x}, t) + \mathcal{L}u(\boldsymbol{x}, t) = f(\boldsymbol{x}, t), \quad (\boldsymbol{x}, t) \in \Omega \times (0, T]; \\ &u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega; \quad u(\boldsymbol{x}, t) = 0, \quad (\boldsymbol{x}, t) \in \partial\Omega \times [0, T]. \end{aligned} \tag{1.2}$$

Here $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$) is a simply connected bounded domain with smooth boundary $\partial\Omega$, $\boldsymbol{x} = (x_1, \cdots, x_d)$, $\mathcal{L} := -\nabla \cdot (\mathbf{K}(\boldsymbol{x})\nabla)$, $\mathbf{K}(\boldsymbol{x}) := \left(k_{i,j}(\boldsymbol{x})\right)_{i,j=1}^d$ is a symmetric and coercive diffusion tensor, $\kappa(t)$ is the ratio of sub-diffusion particles versus normal diffusion and $f(\boldsymbol{x}, t)$ is source or sink term. Model (1.2) is the variable-order analogue of the mobile-immobile time-fractional diffusion equations proposed in, e.g., [22, 31], in which both the integer-order and fractional order time derivatives are involved in order to describe the dynamic mass exchange between mobile and immobile phases. Compared with the subdiffusion equations, which only have fractional time derivatives in the models, the time drift term $\partial_t u$ describes the motion time and thus helps to distinguish the mobile and immobile status [30].

In this work we develop a fast approximation of the hidden-memory variable-order fractional derivative to reduce the computational cost of generating coefficients from $O(N^2)$ to $O(N \log N)$, and then apply this to construct a fully-discrete finite element scheme. The key ingredient of this fast approximation lies in expanding the temporal discretization coefficients in terms of the variable order via the Taylor series, which differs from the methods like the exponential-sum-approximation and the hierarchical matrix method (cf. [33]) that rely on the exponential or Taylor expansions for the power function kernels. This is also the critical step to treat the variable fractional order such that the scheme could be efficiently implemented as in the constant fractional order case without loss of accuracy. By different techniques from the analysis of L1 methods for constant-order and variable-order time-fractional problems, we analyze this fast approximation method and prove error estimates for the corresponding fast fully-discrete finite element scheme to model (1.2). We further develop an fDAC algorithm to reduce the complexity of solving the linear systems from $O(MN^2)$ to $O(MN \log^2 N)$ based on the Toeplitz-like matrices generated from the fast approximation scheme via the fast Fourier transform (FFT).

The rest of the paper is organized as follows. In Sect. 2 we present some preliminaries and review the standard finite element method for the proposed model. In Sect. 3 we develop an efficient discretization for the hidden-memory variable-order fractional derivative and a corresponding fully-discrete scheme for the model, as well as deriving a fDAC algorithm for the resulting linear system. In Sect. 4 we prove optimal-order error estimates of the efficient finite element scheme. We perform numerical experiments to test the performance of the fast method in the last section.

2 Preliminaries

Let $C^m[0,T]$ with $m \in \mathbb{N}$ be the space of continuous function with continuous derivatives up to order m, equipped with standard norms [1]. For a Banach space $\mathcal{X}$, let $C^m([0,T];\mathcal{X})$ be the space of continuously differentiable functions of order m on $[0,T]$ with respect to the norm $\|\cdot\|_{\mathcal{X}}$. Let $\{\lambda_i,\phi_i\}_{i=1}^{\infty}$ be the eigen-pairs of $\mathcal{L}$ and $\check{H}^s(\Omega) = \{g \in L^2(\Omega) : \|g\|^2_{\check{H}^s(\Omega)} := \sum_{i=1}^{\infty} \lambda_i^s (g,\phi_i)^2 < \infty\}$ be the subspace of $H^s(\Omega)$ [1]. Let Q_m be fixed positive constants and Q be a generic positive constant with different values at different occurrences. In particular, Q is always independent from the mesh parameters of numerical schemes. For convenience, we may drop the subscript L^2 in $(\cdot,\cdot)_{L^2}$ and $\|\cdot\|_{L^2}$ and the notation Ω is the Sobolev space norms.

We refer the well-posedness and regularity of (1.2) proved by [34, 35].

Theorem 2.1 *If $u_0 \in \check{H}^{2+\gamma}$, $f \in H^1(0,T;\check{H}^{\gamma})$ with $\gamma > d/2$ and $\lim_{t\to 0^+}(\alpha(t)-\alpha(0))\ln t$ exists, then (1.2) has a unique solution $u \in C^1([0,T];\check{H}^{\gamma})$ with*

$$\|u\|_{C^1([0,T];\check{H}^{\gamma})} \le Q\big(\|u_0\|_{\check{H}^{2+\gamma}} + \|f\|_{H^1(0,T;\check{H}^{\gamma})}\big). \tag{2.1}$$

Here $Q = Q(\alpha^*, \|\kappa\|_{C[0,T]}, T)$.

Theorem 2.2 *If $\alpha, \kappa \in C^1[0,T]$, $f \in H^1(0,T;\check{H}^{2+s}) \bigcap H^2(0,T;\check{H}^s)$ and $u_0 \in \check{H}^{4+s}$ for some $s \geq 0$, then $u \in C^2((0,T];\check{H}^s)$ with the following estimate holds for $0 < \epsilon \ll T$*

$$\|u\|_{C^2([\epsilon,T];\check{H}^s)} \leq Q\epsilon^{-\alpha(0)}\big(\|u_0\|_{\check{H}^{4+\gamma}} + \|f\|_{H^1(0,T;\check{H}^{2+\gamma})} + \|f\|_{H^2(0,T;\check{H}^s)}\big).$$

Furthermore, if $\alpha(0) = 0$, then the solution $u \in C^2([0,T];\check{H}^s)$ has the global estimate

$$\|u\|_{C^2([0,T];\check{H}^s)} \leq Q\big(\|u_0\|_{\check{H}^{4+s}} + \|f\|_{H^1(0,T;\check{H}^{2+s})} + \|f\|_{H^2(0,T;\check{H}^s)}\big).$$

Here $Q = Q(\|\alpha\|_{C^1([0,T])}, \|\kappa\|_{C^1[0,T]}, T)$.

Based on these theoretical results, we refer the numerical discretization of model (1.2) from [34]. Define a partition on $[0,T]$ by $t_n := n\tau$ for $0 \leq n \leq N$ with $\tau = T/N$ and a quasi-uniform partition on Ω with mesh diameter h. Let S_h be the space of continuous and piecewise linear functions on Ω with respect to this partition. Denote $u_n := u(\boldsymbol{x}, t_n)$, $\kappa_n := \kappa(t_n)$, $\alpha_n = \alpha(t_n)$ and $f_n := f(\boldsymbol{x}, t_n)$. We follow [34] to discretize $\partial_t u$ and $\partial_t^{\alpha(t)} u$ at $t = t_n$ for $1 \leq n \leq N$ by

$$\begin{aligned}
\partial_t u(\boldsymbol{x}, t_n) &= \frac{u_n - u_{n-1}}{\tau} + \frac{1}{\tau}\int_{t_{n-1}}^{t_n} \partial_{tt} u(\boldsymbol{x}, s)(s - t_{n-1})ds \\
&:= \delta_\tau u_n + E_n, \\
\partial_t^{\alpha_n} u(\boldsymbol{x}, t_n) &:= \sum_{k=1}^{n} b_{n,k}(u_k - u_{k-1}) + \hat{R}_n + R_n \\
&:= \hat{\delta}_\tau^{\alpha_n} u_n + \hat{R}_n + R_n,
\end{aligned} \tag{2.2}$$

where

$$b_{n,k} := \frac{(t_n - t_{k-1})^{1-\alpha_k} - (t_n - t_k)^{1-\alpha_k}}{\Gamma(2-\alpha_k)\tau}, \tag{2.3}$$

$$\hat{R}_n = \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \Big[\frac{\partial_s u(\boldsymbol{x}, s)}{\Gamma(1-\alpha(s))(t_n - s)^{\alpha(s)}} - \frac{\partial_s u(\boldsymbol{x}, s)}{\Gamma(1-\alpha_k)(t_n - s)^{\alpha_k}}\Big]ds, \tag{2.4}$$

$$R_n = \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \frac{1}{\tau\Gamma(1-\alpha_k)(t_n - s)^{\alpha_k}}\Big[\int_{t_{k-1}}^{t_k}\int_z^s \partial_{\theta\theta} u(\boldsymbol{x}, \theta)d\theta dz\Big]ds. \tag{2.5}$$

Then the fully discrete finite element method for model (1.2) reads: find $\hat{U}_n \in S_h$ for $n = 1, 2, \cdots, N$ with $\hat{U}_0(\boldsymbol{x}) := \Pi_h u_0(\boldsymbol{x})$, such that

$$(\delta_\tau \hat{U}_n, \chi_h) + \kappa_n(\delta_\tau^{\alpha_n} \hat{U}_n, \chi_h) + (\mathbf{K}\nabla \hat{U}_n, \nabla\chi) = (f_n, \chi_h), \ \forall \chi_h \in S_h. \tag{2.6}$$

We could directly follow [34] to show that, under the regularity of the solutions in Theorems 2.1–2.2, the scheme (2.6) generates a numerical solution of $O(\tau + h^2)$ accuracy. For the implementation, due to the impact of the hidden-memory variable fractional order, the translation invariance property of the coefficients $\{b_{n,k}\}$ is lost and in the time-stepping procedure of the scheme (2.6), we have to compute the coefficients $\{b_{n,k}\}$ by $O(N^2)$ operations and solve the linear systems in $O(N^2 M)$ operations (M refers to the number of spatial nodes), which is computationally expensive.

3 Efficient Discretization and Fast Implementation

In this section, we propose an efficient discretization for model (1.2) and study its fast implementation.

3.1 An Efficient Finite Element Scheme

We use the $(S+1)$-term Taylor expansion of the exponential function $(n-k+1)^{1-\alpha_k} - (n-k)^{1-\alpha_k}$ for $n-k \geq 2$ at $1-\overline{\alpha}$ with $\overline{\alpha} = (\alpha_* + \alpha^*)/2$ in $b_{n,k}$ to get

$$
\begin{aligned}
&(n-k+1)^{1-\alpha_k} - (n-k)^{1-\alpha_k} \\
&\quad = \sum_{s=0}^{S} \frac{(\overline{\alpha}-\alpha_k)^s}{s!}\Big[(n-k+1)^{1-\overline{\alpha}} \ln^s(n-k+1) \\
&\quad -(n-k)^{1-\overline{\alpha}} \ln^s(n-k)\Big] \\
&\quad +\frac{(\overline{\alpha}-\alpha_k)^{S+1}}{(S+1)!}\Big[(n-k+1)^{1-\xi_{n,k}} \ln^{S+1}(n-k+1) \\
&\quad -(n-k)^{1-\xi_{n,k}} \ln^{S+1}(n-k)\Big],
\end{aligned}
\tag{3.1}
$$

where $\xi_{n,k}$ is a constant between α_k and $\overline{\alpha}$, which depends on n and k. Insert this expansion into $b_{n,k}$ and discard the local truncation errors to obtain an approximation of $b_{n,k}$ for $n-k \geq 2$

$$
\begin{aligned}
c_{n,k} = \sum_{s=0}^{S} &\frac{(\overline{\alpha}-\alpha_k)^s}{s!\Gamma(2-\alpha_k)\tau^{\alpha_k}}\Big[(n-k+1)^{1-\overline{\alpha}} \ln^s(n-k+1) \\
&-(n-k)^{1-\overline{\alpha}} \ln^s(n-k)\Big],
\end{aligned}
\tag{3.2}
$$

and the local truncation errors $r_{n,k} := b_{n,k} - c_{n,k}$ $(n-k \geq 2)$ are given by

$$
\begin{aligned}
r_{n,k} = &\frac{(\overline{\alpha}-\alpha_k)^{S+1}}{(S+1)!\Gamma(2-\alpha_k)\tau^{\alpha_k}}\Big[(n-k+1)^{1-\xi_{n,k}} \ln^{S+1}(n-k+1) \\
&-(n-k)^{1-\xi_{n,k}} \ln^{S+1}(n-k)\Big].
\end{aligned}
\tag{3.3}
$$

For the case $n-k = 0, 1$, we set $c_{n,k} = b_{n,k}$. We replace $b_{n,k}$ by $c_{n,k}$ in (2.2) to get a different discretization of the time-fractional derivative $\partial_t^{\alpha(t)} u$ at $t = t_n$

$$
\begin{aligned}
\partial_{t_n}^{\alpha_n} u_n :&= \tilde{\delta}_\tau^{\alpha_n} u_n + \tilde{R}_n + \hat{R}_n + R_n \\
&= \sum_{k=1}^{n} c_{n,k}(u_k - u_{k-1}) + \tilde{R}_n + \hat{R}_n + R_n,
\end{aligned}
\tag{3.4}
$$

where $\hat{R}_n$ and R_n are given in (2.4) and (2.5), respectively, and $\tilde{R}_n$ is defined by

$$
\tilde{R}_n := \sum_{k=1}^{n-1} r_{n,k}(u_k - u_{k-1}). \tag{3.5}
$$

We substitute (3.4) and the first equation in (2.2) into (1.2) and integrate the resulting equation multiplied by $\chi \in H_0^1(\Omega)$ on Ω to get

$$
\begin{aligned}
(\delta_\tau u_n, \chi) + \kappa_n(\tilde{\delta}_\tau^{\alpha_n} u_n, \chi) + (\mathbf{K}\nabla u_n, \nabla\chi) \\
= (f_n, \chi) - (\kappa_n(\tilde{R}_n + \hat{R}_n + R_n) + E_n, \chi).
\end{aligned}
\tag{3.6}
$$

Ignoring the local truncation errors we get the finite element scheme: find $U_n \in S_h$ for $1 \leq n \leq N$ with $U_0 = \Pi_h u_0$ such that for any $\chi_h \in S_h$

$$
(\delta_\tau U_n, \chi_h) + \kappa_n(\tilde{\delta}_\tau^{\alpha_n} U_n, \chi_h) + (\mathbf{K}\nabla U_n, \nabla\chi_h) = (f_n, \chi_h). \tag{3.7}
$$

Throughout the paper, we set the parameter S as

$$S = \lceil 3 \ln N \rceil \tag{3.8}$$

and we will show in subsequent sections that the accuracy of the numerical scheme (3.7) is not affected under this choice but the computational costs and storage are significantly reduced. To be specific, we will find from the derivation of (4.6) that (3.8) is almost the optimal choice to ensure that the summation in (4.6) is less than 1, which in turn leads to the first-order accuracy in (4.7) and finally in the numerical scheme.

3.2 Fast Coefficient Generation

Let $\{\psi_j(\boldsymbol{x})\}_{j=1}^M$ be the basic functions of S_h satisfying $\psi_j(\boldsymbol{x}_j) = 1$ and $\psi_j(\boldsymbol{x}_i) = 0$ for $i \neq j$, where M is the number of degree of freedom of the finite element space. Let $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{M\times M}$ be the corresponding mass and stiffness matrices in (3.7), respectively, and $\boldsymbol{F}^{(n)} = [F_1^n, F_2^n, \cdots, F_M^n]^\top$ with $F_j^n = (f(\cdot, t_n), \psi_j(\cdot))$ for $1 \leq n \leq N$. We define a lower triangular matrix $\mathbf{A} = (a_{n,k})_{n,k=1}^N$ to store the discretization coefficients of operators δ_τ and $\tilde{\delta}_\tau^\rho$ by

$$a_{n,k} = \begin{cases} \dfrac{1}{\tau} + \kappa_n c_{n,n}, & k = n, \\ -\dfrac{1}{\tau} + \kappa_n(c_{n,n-1} - c_{n,n}), & k = n-1, \\ \kappa_n(c_{n,k} - c_{n,k+1}), & n-k \geq 2. \end{cases} \tag{3.9}$$

Based on these notations, we obtain the matrix form of (3.7)

$$\big(a_{n,n}\mathbf{B} + \mathbf{C}\big)\boldsymbol{U}^{(n)} = \boldsymbol{F}^{(n)} - \sum_{k=1}^{n-1} a_{n,k}\mathbf{B}\boldsymbol{U}^{(k)} + \kappa_n c_{n,1}\boldsymbol{U}^{(0)} \tag{3.10}$$

for $1 \leq n \leq N$ where $\boldsymbol{U}^{(n)} := [U_1^n, U_2^n, \cdots, U_M^n]^\top$.

Denote

$$D_k^{(s)} = \frac{(\overline{\alpha} - \alpha_k)^s}{s!\Gamma(2-\alpha_k)\tau^{\alpha_k}}, \quad g_k^{(s)} = k^{1-\overline{\alpha}} \ln^s k - (k-1)^{1-\overline{\alpha}} \ln^s(k-1),$$

then $c_{n,k}$ in (3.2) could be expressed by

$$c_{n,k} = \sum_{s=0}^{S} D_k^{(s)} g_{n-k+1}^{(s)}.$$

Insert this expression into (3.9) to obtain that

$$a_{n,k} = \kappa_n \sum_{s=0}^{S} \Big(D_k^{(s)} g_{n-k+1}^{(s)} - D_{k+1}^{(s)} g_{n-k}^{(s)} \Big), \quad n-k \geq 2. \tag{3.11}$$

Based on this decomposition we have the following conclusion.

Theorem 3.1 *The coefficients $\{a_{n,n}\}_{n=1}^N$, $\{a_{n,n-1}\}_{n=2}^N$ and the components $\{\kappa_n\}_{n=3}^N$, $\{D_k^s\}_{k=1}^{N-2}$, $\{g_k^{(s)}\}_{k=2}^N$ for $0 \leq s \leq S$ can be computed and stored in $O(SN)$ operations and storage. Here S is given in (3.8).*

Proof We could compute and store $\{a_{n,n}\}_{n=1}^N$ and $\{a_{n,n-1}\}_{n=2}^N$ in $O(N)$ computations and storage, and $O(N)$ operations and storage are required for $\{\kappa_n\}_{n=3}^N$. We also note that $O(SN)$ operations and storage are needed for computing and storing $\{D_k^{(s)}\}_{k=1}^{N-2}$ and $\{g_k^{(s)}\}_{k=2}^N$ for $0 \le s \le S$. Thus we complete the proof. □

3.3 Fast Divide and Conquer Algorithm

Although the linear system (3.10) can be solved in $O(M)$ operations at each time step, we have to compute the right-hand side term $\sum_{k=1}^n a_{n,k}\mathbf{B}\boldsymbol{U}^{(k)}$ in $O(nM)$ operations. Therefore, totally $O(N^2M)$ operations are needed for all time steps, which leads to $O(N^2M)$ computations of solving the linear system (3.10). To derive a fast algorithm for solving (3.10) for $1 \le n \le N$, we set $\boldsymbol{U} = [\boldsymbol{U}^{(1)\top}, \boldsymbol{U}^{(2)\top}, \cdots, \boldsymbol{U}^{(N)\top}]^\top$ and $\boldsymbol{F} = [\boldsymbol{F}^{(1)\top}, \boldsymbol{F}^{(2)\top}, \cdots, \boldsymbol{F}^{(N)\top}]^\top$ to rewrite (3.10) into the following all-at-once linear system

$$\big(\mathbf{A} \otimes \mathbf{B} + \mathbf{I}_N \otimes \mathbf{C}\big)\boldsymbol{U} = \boldsymbol{F}, \tag{3.12}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix.

By (3.11), we decompose the coefficient matrix $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{A}^b + \mathbf{A}^t := \mathbf{A}^b + \mathrm{diag}(\boldsymbol{\kappa}) \sum_{s=0}^{S} \big[\mathbf{T}_p^{(s)}\mathrm{diag}(\boldsymbol{D}^{(s)}) - \mathbf{T}^{(s)}\mathrm{diag}(\boldsymbol{D}_p^{(s)})\big], \tag{3.13}$$

where $\mathbf{A}^b$ is a band matrix with $(\mathbf{A}^b)_{n,i} = a_{n,i}$ for $n-2 \le i \le n$ and zeros otherwise, $\mathrm{diag}(\boldsymbol{\kappa})$, $\mathrm{diag}(\boldsymbol{D}_p^{(s)})$ and $\mathrm{diag}(\boldsymbol{D}^{(s)})$ are diagonal matrices generated from the vectors $\boldsymbol{\kappa}$, $\boldsymbol{D}_p^{(s)}$ and $\boldsymbol{D}^{(s)}$ with $(\boldsymbol{\kappa})_n = \kappa_n$, $(\boldsymbol{D}_p^{(s)})_n = D_{n+1}^{(s)}$ and $(\boldsymbol{D}^{(s)})_n = D_n^{(s)}$, respectively, for $0 \le s \le S$. $\mathbf{T}_p^{(s)} = \mathrm{toeplitz}(\boldsymbol{t}_p^{(s)}, \mathbf{0})$ and $\mathbf{T}^{(s)} = \mathrm{toeplitz}(\boldsymbol{t}^{(s)}, \mathbf{0})$ for $0 \le s \le S$ are Toeplitz matrices with $\boldsymbol{t}_p^{(s)}$ and $\boldsymbol{t}^{(s)}$, respectively, being the first columns, where $(\boldsymbol{t}_p^{(s)})_n = (\boldsymbol{t}^{(s)})_n = 0$ for $n = 0, 1$ and $(\boldsymbol{t}_p^{(s)})_n = g_{n+1}^{(s)}$ and $(\boldsymbol{t}^{(s)})_n = g_n^{(s)}$ for $n \ge 2$.

Base on (3.13), we divide $\mathbf{A}$ into four $N/2 \times N/2$ blocks as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_3 & \mathbf{A}_2 \end{bmatrix} := \begin{bmatrix} \mathbf{A}_1^b & \mathbf{0} \\ \mathbf{A}_3^b & \mathbf{A}_2^b \end{bmatrix} + \begin{bmatrix} \mathbf{A}_1^t & \mathbf{0} \\ \mathbf{A}_3^t & \mathbf{A}_2^t \end{bmatrix}, \tag{3.14}$$

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 (3.13) is equivalent to the following two sub-linear systems

$$\begin{cases} \big(\mathbf{A}_1 \otimes \mathbf{B} + \mathbf{I}_{N/2} \otimes \mathbf{C}\big)\boldsymbol{U}_1 = \boldsymbol{F}_1, \\ \big(\mathbf{A}_2 \otimes \mathbf{B} + \mathbf{I}_{N/2} \otimes \mathbf{C}\big)\boldsymbol{U}_2 = \boldsymbol{F}_2 - (\mathbf{A}_3 \otimes \mathbf{B})\boldsymbol{U}_1. \end{cases} \tag{3.15}$$

To develop a fast algorithm, we consider the fast matrix-vector multiplication $\mathbf{A}_3\boldsymbol{v}$ for any $\boldsymbol{v} \in \mathbb{R}^{N/2}$ in the following lemma.

Lemma 3.1 *The matrix-vector multiplication $\mathbf{A}_3\boldsymbol{v}$ for any $\boldsymbol{v} \in \mathbb{R}^{N/2}$ can be carried out in $O(SN\log N)$ operations. Here S is given in (3.8).*

Proof By (3.14), we have $\mathbf{A}_3\boldsymbol{v} = \mathbf{A}_3^b\boldsymbol{v} + \mathbf{A}_3^t\boldsymbol{v}$. As $\mathbf{A}_3^b$ has only one nonzero entry on its up and right corner, $\mathbf{A}_3^b\boldsymbol{v}$ can be computed in $O(1)$ operations. We then use the structure of $\mathbf{A}^t$ in (3.13) to decompose $\mathbf{A}_3^t\boldsymbol{v}$ as

$$\mathbf{A}_3^t\boldsymbol{v} = \operatorname{diag}(\hat{\boldsymbol{\kappa}}) \sum_{s=0}^{S} \big[\hat{\mathbf{T}}_p^{(s)}\operatorname{diag}(\hat{\boldsymbol{C}}^{(s)})\boldsymbol{v} - \hat{\mathbf{T}}^{(s)}\operatorname{diag}(\hat{\boldsymbol{C}}_p^{(s)})\boldsymbol{v}\big],$$

where $\hat{\boldsymbol{\kappa}}$ is the second half of $\boldsymbol{\kappa}$, $\hat{\boldsymbol{C}}_p^{(s)}$ and $\hat{\boldsymbol{C}}^{(s)}$ are the first half of $\boldsymbol{C}_p^{(s)}$ and $\boldsymbol{C}^{(s)}$, respectively, and $\hat{\mathbf{T}}_p^{(s)}$ and $\hat{\mathbf{T}}^{(s)}$ are $N/2 \times N/2$ Toeplitz matrices given by the left and bottom parts of $\mathbf{T}_p^{(s)}$ and $\mathbf{T}^{(s)}$, respectively. Therefore, The matrix-vector multiplication $\mathbf{A}_3^t\boldsymbol{v}$ can be computed in the following steps. First, $\boldsymbol{w}_p^{(s)} = \operatorname{diag}(\hat{\boldsymbol{C}}_p^{(s)})\boldsymbol{v}$ and $\boldsymbol{w}^{(s)} = \operatorname{diag}(\hat{\boldsymbol{C}}^{(s)})\boldsymbol{v}$ are the multiplication of diagonal matrices and vectors, which need $O(SN)$ operations totally for $0 \le s \le S$. Then $\hat{\mathbf{T}}_p^{(s)}\boldsymbol{w}_p^{(s)}$ and $\hat{\mathbf{T}}^{(s)}\boldsymbol{w}^{(s)}$ can be computed in $O(N\log N)$ operations by the FFT, and hence

$$\boldsymbol{q} = \sum_{s=0}^{S} \big(\hat{\mathbf{T}}_p^{(s)}\boldsymbol{w}_p - \hat{\mathbf{T}}^{(s)}\boldsymbol{w}\big)$$

requires $O(SN\log N)$ operations. Finally, $\operatorname{diag}(\hat{\boldsymbol{\kappa}})\boldsymbol{q}$ requires $O(N)$ operations. Thus we complete the proof. □

By Lemma 3.1, $(\mathbf{A}_3 \otimes \mathbf{B})\boldsymbol{U}_1$ on the right-hand side of (3.15) can be carried out in $O(SMN)$ operations. Furthermore, as each sub-linear system in (3.15) has a similar structure as (3.12), we could repeat this splitting procedure to obtain a fDAC algorithm by recursively solving the sub-linear systems, the computational cost of which is analyzed in the following theorem.

Theorem 3.2 *The all-at-once linear system (3.12) could be solved in $O(SMN\log^2 N)$ operations by the fDAC algorithm.*

Proof Let ϱ be a positive integer satisfying $2^\varrho = N$. By Lemma 3.1, $O\big(SMN\log N\big)$ operations are needed to compute $\big(\mathbf{A}_3 \otimes \mathbf{B}\big)\boldsymbol{U}_1$ in (3.15).Concerning the repeatedly splitting procedure in the fast DAC algorithm, the total computational complexity of the right-hand side terms is

$$\begin{aligned} &O\bigg(SM\bigg[N\log N + 2\cdot\frac{N}{2}\log\frac{N}{2} + \cdots + 2^{\varrho-1}\cdot\frac{N}{2^{\varrho-1}}\log\frac{N}{2^{\varrho-1}}\bigg]\bigg) \\ &= O\bigg(SMN\log\frac{N^\varrho}{2^{\varrho(\varrho-1)/2}}\bigg) = O\bigg(SMN\log N^{\frac{\varrho+1}{2}}\bigg) = O(SMN\log^2 N), \end{aligned}$$

which completes the proof. □

Remark 3.1 In view of the proof of Lemma 3.1 and Theorem 3.2, we could indeed set $c_{n,k} = b_{n,k}$ for $0 \le n-k \le S$ rather than $0 \le n-k \le 1$, with the same accuracy and computational complexity.

4 Error Estimate

In this section we prove several auxiliary estimates to perform error estimates for the efficient finite element scheme (3.7).

4.1 Auxiliary Estimates

We first refer the estimates of $\hat{R}_n$ and R_n proved in [34].

Lemma 4.1 *Suppose* $\alpha, \kappa \in C^1[0,T]$, $u_0 \in \check{H}^4$, $f \in H^1(0,T;\check{H}^2) \bigcap H^2(0,T;L^2)$, *then the following estimates hold*

$$\|\hat{R}_n\| \le QM_0\tau,\ \|R_n\| \le QM_0 N^{\alpha^*-1} n^{-\alpha^*}$$

where $M_0 := \|u_0\|_{\check{H}^4} + \|f\|_{H^1(0,T;\check{H}^2)} + \|f\|_{H^2(0,T;L^2)}$.

We then estimate the local truncation errors $r_{n,k}$ and $\tilde{R}_n$ in the following lemma.

Lemma 4.2 *The local truncation errors* $r_{n,k}$ *in (3.3) for* $n-k \ge 1$ *could be bounded by*

$$|r_{n,k}| \le \frac{1}{2^{S+1}(S+1)!\Gamma(2-\alpha_k)\tau^{\alpha_k}} \times\Big[(n-k+1)\ln^{S+1}(n-k+1) - (n-k)\ln^{S+1}(n-k)\Big]. \tag{4.1}$$

Proof For a given constant $a \ge 1$, define a function of $\beta \in [0,1]$ as

$$y_a(\beta) = (a+1)^{1-\beta}\ln^{S+1}(a+1) - a^{1-\beta}\ln^{S+1} a.$$

Therefore, $r_{n,k}$ could be rewritten as

$$r_{n,k} = \frac{(\overline{\alpha}-\alpha_k)^{S+1}}{(S+1)!\Gamma(2-\alpha_k)\tau^{\alpha_k}} y_{n-k}(\xi_{n,k}). \tag{4.2}$$

As

$$y_a'(\beta) = -(a+1)^{1-\beta}\ln^{S+2}(a+1) + a^{1-\beta}\ln^{S+2} a < 0,$$

$y_a(\beta)$ is a monotone decreasing function with

$$y_a(\beta) \le y_a(0) = (a+1)\ln^{S+1}(a+1) - a\ln^{S+1} a.$$

We incorporate this with (4.2) under $a = n-k$ and $|\overline{\alpha}-\alpha_k| \le 1/2$ to complete the proof. □

Lemma 4.3 *If* $\alpha, \kappa \in C^1[0,T]$, $u_0 \in \check{H}^4$, $f \in H^1(0,T;\check{H}^2) \bigcap H^2(0,T;L^2)$. *Then*

$$\sum_{k=1}^{n-1}|r_{n,k}| \le 1, \quad \|\tilde{R}_n\| \le QM_0\tau, \quad 1 \le n \le N. \tag{4.3}$$

Proof We use (4.1) and the fact that $1/\Gamma(2-\alpha) \le 1.2$ for $\alpha \in [0,1]$ to get

$$\begin{aligned}\sum_{k=1}^{n-1}|r_{n,k}| &\le \frac{1.2\tau^{-\alpha^*}}{2^{(S+1)}(S+1)!}\sum_{k=1}^{n}\big[(n-k+1)\ln^{S+1}(n-k+1) \\ &\quad -(n-k)\ln^{S+1}(n-k)\big] \\ &\le \frac{1.2\times N^{\alpha^*} n\ln^{S+1} n}{2^{S+1}(S+1)!} \le \frac{1.2\times N^{1+\alpha^*}\ln^{S+1} N}{2^{S+1}(S+1)!}.\end{aligned} \tag{4.4}$$

We apply the Stirling's formula $1/(S+1)! \le e^{S+1}(S+1)^{-(S+3/2)}$ to get

$$\begin{aligned}\frac{1.2\times N^{1+\alpha^*}\ln^{S+1}N}{2^{S+1}(S+1)!} &\le \frac{1.2\times N^{1+\alpha^*}(e\ln N)^{S+1}}{2^{S+1}(S+1)^{S+3/2}}\\ &\le N^{1+\alpha^*}\Big(\frac{e\ln N}{2(S+1)}\Big)^{S+1}.\end{aligned} \tag{4.5}$$

By (3.8) we have

$$\sum_{k=1}^{n-1}|r_{n,k}| \le N^{1+\alpha^*}\Big(\frac{e}{6}\Big)^{3\ln N} = N^{1+\alpha^*}e^{3\ln N\ln\frac{e}{6}} \le N^{4+\alpha^*-3\ln 6} < 1, \tag{4.6}$$

which proves the first statement in (4.3). By (3.5) we obtain

$$\|\tilde R_n\| = \Big\|\sum_{k=1}^{n-1} r_{n,k}(u_k-u_{k-1})\Big\| \le \tau\|u\|_{C^1([0,T];L^2)}\sum_{k=1}^{n-1}|r_{n,k}| \le QM_0\tau, \tag{4.7}$$

which proves the second statement of (4.3). □

Due to the lack of the monotonicity in the coefficients $\{c_{n,k}\}_{k=1}^n$, which is key in error estimates of standard discretizations such as the L1 methods for constant-order and even variable-order fractional derivatives [23, 37], we prove the following lemma to circumvent this difficulty.

Lemma 4.4 *Under (3.8) we have*

$$\sum_{k=1}^{n-1}|c_{n,k+1}-c_{n,k}| \le c_{n,n}+Q_1, \quad 2\le n\le N. \tag{4.8}$$

where $Q_1>0$ is a constant independent of n, N and τ.

Proof For $1\le k\le n-1$, we decompose $c_{n,k+1}-c_{n,k}$ as

$$c_{n,k+1}-c_{n,k} = (c_{n,k+1}-b_{n,k+1}) + (b_{n,k+1}-b_{n,k}) + (b_{n,k}-c_{n,k}). \tag{4.9}$$

Define an intermediate variable $\hat b_{n,k}$ by

$$\hat b_{n,k} = \int_{t_{k-1}}^{t_k}\frac{ds}{\Gamma(1-\alpha_{k+1})(t_n-s)^{\alpha_{k+1}}},$$

which satisfies $b_{n,k+1} > \hat b_{n,k}$ and

$$\begin{aligned}\sum_{k=1}^{n-1}|\hat b_{n,k}-b_{n,k}| &\le \sum_{k=1}^{n-1}\Big|\int_{t_{k-1}}^{t_k}\frac{\partial}{\partial z}\Big(\frac{1}{\Gamma(1-\alpha(z))(t_n-s)^{\alpha(z)}}\Big)\Big|_{z=\iota}ds\Big|\\ &\le \sum_{k=1}^{n-1}\int_{t_{k-1}}^{t_k}\Big|\frac{\alpha'(z)\Gamma'(1-\alpha(z))-\Gamma(1-\alpha(z))\alpha'(z)\ln(t_n-s)}{\Gamma^2(1-\alpha(z))(t_n-s)^{\alpha(z)}}\Big|ds\\ &\le Q\sum_{k=1}^{n-1}\int_{t_{k-1}}^{t_k}\frac{ds}{(t_n-s)^{\frac{1+\alpha^*}{2}}} \le Q_0.\end{aligned}$$

Thus the second term on the right-hand side of (4.9) can be bounded by

$$\left|b_{n,k+1}-b_{n,k}\right| \le (b_{n,k+1}-\hat{b}_{n,k}) + \left|\hat{b}_{n,k}-b_{n,k}\right| \\ = (b_{n,k+1}-b_{n,k}) + 2\left|b_{n,k}-\hat{b}_{n,k}\right|.$$

Combining this with (4.3), we obtain

$$\sum_{k=1}^{n-1}\left|c_{n,k+1}-c_{n,k}\right| \le 2\sum_{k=1}^{n-1}\left|c_{n,k}-b_{n,k}\right| + 2\sum_{k=1}^{n}\left|\hat{b}_{n,k}-b_{n,k}\right| \\ + \sum_{k=1}^{n-1}(b_{n,k+1}-b_{n,k}) \\ \le 2 + 2Q_0 + b_{n,n} := Q_1 + c_{n,n}.$$

Thus we complete the proof. □

Let I be the identity operator and $\Pi_h : H_0^1 \to S_h$ be the Ritz projection operator defined by

$$(\mathbf{K}\nabla(g-\Pi_h g), \nabla\chi_h) = 0, \quad \forall \chi_h \in S_h, \quad \text{for } g \in H_0^1,$$

satisfying the approximation property $\|g-\Pi_h g\|_{L^2} \le Qh^2\|g\|_{H^2}$ for $g \in H^2 \cap H_0^1$ [26]. Let $u_n - U_n = \eta_n + \xi_n$ with $\eta = u - \Pi_h u$ and $\xi_n = \Pi_h u_n - U_n$, and we estimate η in the following lemma.

Lemma 4.5 *Suppose $\alpha, \kappa \in C^1[0,T]$, $u_0 \in \check{H}^4$, $f \in H^1(0,T;\check{H}^2)\bigcap H^2(0,T;L^2)$, then*

$$\|\delta_\tau \eta\|_{\hat{L}^\infty(0,T;L^2)} + \|\tilde{\delta}_\tau^\alpha \eta\|_{\hat{L}^\infty(0,T;L^2)} \le QM_0h^2, \tag{4.10}$$

where $\|g(\cdot,t)\|_{\hat{L}^\infty(0,T;L^2)} := \max_{1\le n\le N}\|g(\cdot,t_n)\|$.

Proof We use Theorem 2.1 to bound $\delta_\tau\eta$ by

$$\|\delta_\tau\eta_n\| = \frac{1}{\tau}\left\|(I-\Pi_h)\int_{t_{n-1}}^{t_n}\partial_t u dt\right\| \le QM_0h^2.$$

By (3.4), we rewrite $\tilde{\delta}_\tau^{\alpha_n}\eta_n$ in the form of

$$\tilde{\delta}_\tau^{\alpha_n}\eta_n = \partial_{t_n}^{\alpha_n}\eta_n - \sum_{k=1}^{n-1} r_{n,k}(\eta_k - \eta_{k-1}) - \hat{F}_n - F_n, \tag{4.11}$$

where $\hat{F}_n$ and F_n are given by

$$\hat{F}_n = \sum_{k=1}^{n}\int_{t_{k-1}}^{t_k}\Big(\frac{\partial_s\eta(\boldsymbol{x},s)}{\Gamma(1-\alpha(s))(t_n-s)^{\alpha(s)}} - \frac{\partial_s\eta(\boldsymbol{x},s)}{\Gamma(1-\alpha_k)(t_n-s)^{\alpha_k}}\Big)ds,$$

$$F_n = \sum_{k=1}^{n}\int_{t_{k-1}}^{t_k}\frac{1}{\tau\Gamma(1-\alpha_k)(t_n-s)^{\alpha_k}}\int_{t_{k-1}}^{t_k}\int_s^z \partial_{\theta\theta}\eta(\boldsymbol{x},\theta)d\theta dz ds.$$

We estimate the first term on the right-hand side of (4.11) by

$$\|\partial_{t_n}^{\alpha_n}\eta_n\| = \left\|(I-\Pi_h)\int_0^{t_n}\frac{\partial_s u(\boldsymbol{x},s)ds}{\Gamma(1-\alpha(s))(t_n-s)^{\alpha(s)}}\right\| \\ \le Qh^2\|u\|_{C^1([0,T];\check{H}^2)} \le QM_0h^2.$$

We use Lemma 4.3 to estimate the second term on the right-hand side of (4.11) by

$$\Big\| \sum_{k=1}^{n-1} r_{n,k}(\eta_k - \eta_{k-1}) \Big\| \le \sum_{k=1}^{n-1} |r_{n,k}| \Big\| (I - \Pi_h) \int_{t_{k-1}}^{t_k} \partial_s u(\boldsymbol{x}, s) ds \Big\| \le Q M_0 \tau h^2.$$

The third right-hand side term in (4.11) can be bounded by

$$\begin{aligned} \|\hat{F}_n\| &\le \sum_{k=1}^{n} \Big\| (I - \Pi_h) \int_{t_{k-1}}^{t_k} \frac{\partial}{\partial y} \Big(\frac{1}{\Gamma(1 - \alpha(y))(t_n - s)^{\alpha(y)}} \Big) \Big|_{y=z} \\ &\qquad \times (s - t_k) \partial_s u(\boldsymbol{x}, s) ds \Big\| \\ &\le Q \tau h^2 \|u\|_{C^1([0,T]; \check{H}^2)} \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \Big| \frac{1 + \ln(t - s)}{(t - s)^{\alpha^*}} \Big| ds \\ &\le Q M_0 \tau h^2 \int_0^{t_n} \frac{1}{(t - s)^{\frac{1+\alpha^*}{2}}} ds \le Q M_0 \tau h^2, \end{aligned}$$

where $z \in [s, t_k]$. We estimate the last term on the right-hand side of (4.11) by

$$\begin{aligned} \|F_n\| &\le \sum_{k=1}^{n} \Big\| (I - \Pi_h) \int_{t_{k-1}}^{t_k} \frac{\int_{t_{k-1}}^{t_k} \int_z^s \partial_{\theta\theta} u(\boldsymbol{x}, \theta) d\theta dz}{\tau \Gamma(1 - \alpha_k)(t_n - s)^{\alpha_k}} ds \Big\| \\ &\le \sum_{k=1}^{n} \Big\| (I - \Pi_h) \int_{t_{k-1}}^{t_k} \frac{\int_{t_{k-1}}^{t_k} \big(\partial_s u(\boldsymbol{x}, s) - \partial_z u(\boldsymbol{x}, z)\big) dz}{\tau \Gamma(1 - \alpha_k)(t_n - s)^{\alpha_k}} ds \Big\| \\ &\le 2 Q h^2 \|u\|_{C^1([0,T]; \check{H}^2)} \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \frac{ds}{\Gamma(1 - \alpha_k)(t_n - s)^{\alpha_k}} \\ &\le Q M_0 h^2 \int_0^{t_n} \frac{ds}{(t_n - s)^{\alpha^*}} \le Q M_0 h^2. \end{aligned}$$

Inserting these estimates into (4.11) completes the proof.

□

4.2 Optimal-Order Error Estimate

Theorem 4.1 *If $\alpha, \kappa \in C^1[0, T]$, $u_0 \in \check{H}^4$, $f \in H^1(0, T; \check{H}^2) \bigcap H^2(0, T; L^2)$, the following optimal-order error estimate for (3.7) holds under (3.8)*

$$\|U - u\|_{\hat{L}^\infty(0,T;L^2)} \le Q M_0 (\tau + h^2), \tag{4.12}$$

where $Q = Q(\|\alpha\|_{C^1[0,T]}, \|\kappa\|_{C^1[0,T]})$ is independent of τ, N and h.

Proof We subtract (3.7) form (3.6) with $\chi = \chi_h = \xi_n$ to obtain the following error equation

$$(\delta_\tau \xi_n, \xi_n) + \kappa_n (\tilde{\delta}_\tau^{\alpha_n} \xi_n, \xi_n) + (\mathbf{K} \nabla \xi_n, \nabla \xi_n) = -\big(G_n, \xi_n\big), \tag{4.13}$$

where $G_n = \kappa_n(\tilde{R}_n + \hat{R}_n + R_n + \tilde{\delta}_\tau^{\alpha_n} \eta_n) + E_n + \delta_\tau \eta_n$. We use $\xi_0 = \tilde{U}_0 - \Pi_h u_0 \equiv 0$ to rewrite $\tilde{\delta}_\tau^{\alpha_n} \xi_n$ as

$$\tilde{\delta}_\tau^{\alpha_n} \xi_n = c_{n,n} \xi_n - \sum_{k=1}^{n-1} (c_{n,k+1} - c_{n,k}) \xi_k,$$

and use the coercive property of $\mathbf{K}$ to reformulate (4.13) to be

$$(1+\tau\kappa_n c_{n,n})\|\xi_n\|^2 = (\xi_{n-1},\xi_n) + \tau\kappa_n \sum_{k=1}^{n-1}(c_{n,k+1}-c_{n,k})(\xi_k,\xi_n) + \tau(G_n,\xi_n).$$

We use Cauchy inequality to cancel $\|\xi_n\|$ on both sides to obtain

$$(1+\tau\kappa_n c_{n,n})\|\xi_n\| \le \|\xi_{n-1}\| + \tau\kappa_n \sum_{k=1}^{n-1}|c_{n,k+1}-c_{n,k}|\|\xi_k\| + \tau\|G_n\|. \tag{4.14}$$

For $n=1$, we have

$$\|\xi_1\| \le \tau\|G_1\| \le \tau\big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]\|G_1\|.$$

where Q_1 is given in (4.8). We assume that

$$\|\xi_m\| \le \tau\big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]^m \sum_{j=1}^{m}\|G_j\|, \quad 1\le m\le n-1. \tag{4.15}$$

Plug (4.15) with $1\le m\le n-1$ into (4.14) and use the fact that $\big[1+Q_1\tau\|\kappa\|_{C^1[0,T]}\big]^m \le \big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]^{n-1}$ to get

$$\begin{aligned}(1+\tau\kappa_n c_{n,n})\|\xi_n\| &\le \|\xi_{n-1}\| + \tau\kappa_n\Big(\tau\big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]^{n-1}\sum_{j=1}^{n-1}\|G_j\|\Big)\\ &\quad\times\sum_{k=1}^{n-1}|c_{n,k+1}-c_{n,k}| + \tau\|G_n\|\\ &\le \tau\big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]^{n-1}\big[1+\tau\kappa_n(Q_1+c_{n,n})\big]\sum_{j=1}^{n}\|G_j\|,\end{aligned}$$

then we obtain

$$\begin{aligned}\|\xi_n\| &\le \tau\big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]^{n-1}\Big(1+\frac{\tau\kappa_n Q_1}{1+\tau\kappa_n c_{n,n}}\Big)\sum_{j=1}^{n}\|G_j\|\\ &\le \tau\big[1+Q_1\tau\|\kappa\|_{C[0,T]}\big]^{n}\sum_{j=1}^{n}\|G_j\|.\end{aligned}$$

Thus (4.15) holds for $m=n$ and so for any $n\ge 2$ by mathematical induction.

By Lemmas 4.1, 4.3 and 4.5 as well as Theorem 2.2, G_n can be estimated by

$$\begin{aligned}\tau\sum_{j=1}^{n}\|G_j\| &\le \tau\|\kappa\|_{C[0,T]}\sum_{j=1}^{n}\big(\|\tilde R_j\| + \|\hat R_j\| + \|R_j\| + \|\tilde\delta_\tau^{\alpha_j}\eta_j\|\big)\\ &\quad+\tau\sum_{j=1}^{n}\|\delta_\tau\eta_j\| + \tau\sum_{j=1}^{n}\|E_j\|\\ &\le QM_0(\tau+h^2) + \sum_{j=1}^{n}\int_{t_{j-1}}^{t_j}\|u\|_{C^2([s,T];L^2)}(s-t_{j-1})ds\\ &\le QM_0(\tau+h^2) + QM_0\tau\int_0^t s^{-\alpha(0)}ds \le QM_0(\tau+h^2).\end{aligned}$$

Table 1 Errors and convergence rates of (2.6) and (3.7) for Example 1

	$d=1$		$d=2$	
N	$\|u-\hat{U}\|_{\hat{L}^\infty}$	$\|u-U\|_{\hat{L}^\infty}$	$\|u-\hat{U}\|_{\hat{L}^\infty}$	$\|u-U\|_{\hat{L}^\infty}$
2^5	1.3645e−2	1.3645e−2	9.5646e−3	9.5646e−3
2^6	6.5434e−3	6.5434e−3	4.6823e−3	4.6823e−3
2^7	3.1501e−3	3.1501e−3	2.3496e−3	2.3496e−3
2^8	1.5220e−3	1.5220e−3	1.2306e−3	1.2306e−3
Conv. rate	$\mu=1.06$	$\nu=1.06$	$\mu=0.99$	$\nu=0.99$

Invoking this estimate in (4.15) and using the fact that $\left[1+Q_1\tau\|\kappa\|_{C[0,T]}\right]^n \le Q$, we prove (4.12). □

5 Numerical Experiments

We carry out numerical experiments to investigate the performance of the fast finite element scheme (3.7) and the standard finite element scheme (2.6). We set $T=1$, $u_0=0$, $\Omega=(0,1)$ or $(0,1)^2$, $\mathbf{K}=0.01$ or $\mathrm{diag}(0.01, 0.01)$ for $d=1$ or 2, respectively. The uniform rectangular partition on Ω with mesh size $h=2^{-8}$ for $d=1$ and $h=2^{-6}$ for $d=2$ is used. As the spatial discretization is standard, we only measure the temporal convergence rates by

$$\|u-\hat{U}\|_{\hat{L}^\infty(0,T;L^2)} \le QM_0\tau^\mu, \quad \|u-U\|_{\hat{L}^\infty(0,T;L^2)} \le QM_0\tau^\nu.$$

For convenience, we use $\|\cdot\|_{\hat{L}^\infty}$ instead of $\|\cdot\|_{\hat{L}^\infty(0,T;L^2)}$. All experiments are performed on Matlab 2016b on a computer with the following configuration: Intel(R) Core(TM) i5-6500U, CPU 3.2GHz and 8.00GB RAM.

Example 1 We choose the exact solution $u(x_1,t)=t^2\sin(\pi x_1)$ for $d=1$ or $u(x_1,x_2,t)=t^2\sin(\pi x_1)\sin(\pi x_2)$ for $d=2$. The variable order $\alpha(t)$ is given by

$$\alpha(t)=0.1+\frac{1+t}{3}, \quad t\in[0,T],$$

and the right-hand side term $f(\boldsymbol{x},t)$ is computed accordingly. We present the errors and convergence rates in Table 1, from which we observe that the fast finite element scheme has the same accuracy as the standard finite element method. We also record the CPU times $Coef$ for computing the coefficients $\{b_{n,k}\}$ of (2.6), $fCoef$ for computing components of $\{c_{n,k}\}$, TS for solving (2.6) by the time stepping method and $fDAC$ for solving (3.7) by the fDAC algorithm. All these CPU times are measured in seconds and are presented in Table 2. The symbol "-" represents the CPU time is more than 2 hours and we terminate the program. Form this table we observe that the fast finite element scheme is more efficient than the standard finite element scheme. We also observe that $Coef$ and TS increase quadratically, while $fCoef$ and $fDAC$ increase almost linearly, which is highly consistent with our theoretical analysis.

Example 2 We set the right-hand side term as $f=\sin(\pi x_1)$ for $d=1$ or $f=\sin(\pi x_1)\sin(\pi x_2)$ for $d=2$. The variable order $\alpha(t)$ is given by

$$\alpha(t)=0.1+\frac{\cos(t)}{2}.$$

Table 2 CPU times of computing coefficients and solving (2.6) and (3.7) for Example 1

	$d=1$				$d=2$			
N	$Coef$	$fCoef$	TS	$fDAC$	$Coef$	$fCoef$	TS	$fDAC$
2^8	0.90	0.46	0.75	0.28	0.92	0.47	2.58	1.87
2^9	3.32	0.84	1.30	0.52	3.36	0.80	4.82	4.01
2^{10}	13.3	1.63	6.67	1.05	13.9	1.70	13.8	8.75
2^{11}	53.0	3.37	17.8	2.09	55.3	3.33	39.3	18.0
2^{12}	210	6.77	36.9	4.05	220	6.64	132	38.3
2^{13}	845	13.5	80.6	8.23	884	13.3	484	81.3
2^{14}	3400	27.1	186	17.1	3524	27.0	1904	169
2^{15}	–	54.0	–	34.7	–	53.6	–	347
2^{16}	–	216	–	92.1	–	214	–	1010
2^{17}	–	433	–	213	–	426	–	2225
2^{18}	–	867	–	420	–	853	–	4845

Table 3 Errors and convergence rates of (2.6) and (3.7) for Example 2

	$d=1$		$d=2$	
N	$\|u-\hat{U}\|_{\hat{L}^\infty}$	$\|u-U\|_{\hat{L}^\infty}$	$\|u-\hat{U}\|_{\hat{L}^\infty}$	$\|u-U\|_{\hat{L}^\infty}$
2^5	1.7550e−3	1.7550e−3	1.2688e−3	1.2688e−3
2^6	9.4872e−4	9.4872e−4	6.7856e−4	6.7856e−4
2^7	4.9952e−4	4.9952e−4	3.5181e−4	3.5181e−4
2^8	2.5761e−4	2.5761e−4	1.7645e−4	1.7645e−4
Conv. rate	$\mu=0.93$	$\nu=0.93$	$\mu=0.95$	$\nu=0.95$

Table 4 CPU times of computing coefficients and solving (2.6) and (3.7) for Example 2

	$d=1$				$d=2$			
N	$Coef$	$fCoef$	TS	$fDAC$	$Coef$	$fCoef$	TS	$fDAC$
2^8	0.93	0.48	0.05	0.16	0.88	0.44	0.74	1.81
2^9	3.53	0.85	0.15	0.26	3.54	0.86	2.32	4.05
2^{10}	14.1	1.60	0.59	0.59	13.8	1.67	8.04	8.63
2^{11}	56.0	3.26	2.35	1.18	55.4	3.40	30.0	18.4
2^{12}	224	6.45	9.55	2.50	225	6.86	114	38.5
2^{13}	941	12.9	39.8	5.19	913	13.6	476	81.7
2^{14}	3598	51.6	158	16.0	3609	27.3	1877	168
2^{15}	–	103	–	33.1	–	54.5	–	348
2^{16}	–	208	–	75.7	–	213	–	995
2^{17}	–	424	–	156	–	426	–	2219
2^{18}	–	850	–	347	–	877	–	4884

As the exact solutions are not available, we set the numerical solutions by standard finite element scheme with $\tau = 2^{-12}$, $h = 2^{-8}$ for $d = 1$ and $\tau = 2^{-12}$, $h = 2^{-6}$ for $d = 2$ as the reference solutions, respectively. The errors and convergence rates are presented in Table 3, and the CPU times are presented in Table 4. We draw the same conclusions as Example 1 that the fast finite element scheme (3.7) is more efficient than the standard finite element scheme (2.6) with the same accuracy and convergence rates.

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, and by the International Postdoctoral Exchange Fellowship Program (Talent Introduction Program) YJ20210019. All data generated or analyzed during this study are included in this article.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Adams, R.A., Fournier, J.J.F.: Sobolev Spaces. Elsevier, San Diego (2003)
2. Chen, M., Deng, W.: High order algorithms for the fractional substantial diffusion equation with truncated Lévy flights. SIAM J. Sci. Comput. **37**, A890–A917 (2015)
3. Diethelm, K., Ford, N.J., Freed, A.D., Luchko, Y.: Algorithms for the fractional calculus: a selection of numerical methods. Comput. Method. Appl. Mech. Engrg. **194**, 743–773 (2005)
4. Du, R., Sun, Z., Wang, H.: Temporal second-order finite difference schemes for variable-order time-fractional wave equations. SIAM J. Numer. Anal. (2021). https://doi.org/10.1137/19M1301230
5. Ervin, V.J.: Regularity of the solution to fractional diffusion, advection, reaction equations in weighted Sobolev spaces. J. Diff. Equ. **278**, 294–325 (2021)
6. 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)
7. 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)
8. 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)
9. Jia, J., Wang, H., Zheng, X.: A preconditioned fast finite element approximation to variable-order time-fractional diffusion equations in multiple space dimensions. Appl. Numer. Math. **163**, 15–29 (2021)
10. Jia, J., Wang, H.: Analysis of a hidden memory variably distributed-order space-fractional diffusion equation. Appl. Math. Lett. **124**, 107617 (2022)
11. 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)
12. Jin, B., Li, B., Zhou, Z.: Subdiffusion with a time-dependent coefficient: Analysis and numerical solution. Math. Comput. **88**, 2157–2186 (2019)
13. Ke, R., Ng, M., Sun, H.: A fast direct method for block triangular Toeplitz-like with tri-diagonal block systems from time-fractional partial differential equations. J. Comput. Phys. **303**, 203–211 (2015)
14. Le, K., McLean, W., Stynes, M.: Existence, uniqueness and regularity of the solution of the time-fractional Fokker-Planck equation with general forcing. Commun. Pur. Appl. Anal. **18**, 2765–2787 (2019)
15. Li, B., Luo, H., Xie, X.: Analysis of a time-stepping scheme for time fractional diffusion problems with nonsmooth data. SIAM J. Numeri. Anal. **57**, 779–798 (2019)
16. Li, B., Ma, S.: Exponential convolution quadrature for nonlinear subdiffusion equations with nonsmooth initial data. SIAM J. Numer. Anal. (2022). https://doi.org/10.1137/21M1421386
17. 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)

18. Lin, Y., Xu, C.: Finite difference/spectral approximations for the time-fractional diffusion equation. J. Comput. Phys. **225**, 1533–1552 (2007)
19. Lorenzo, C., Hartley, T.: Variable order and distributed order fractional operators. Nonlinear Dyn. **29**, 57–98 (2002)
20. Pang, H., Qin, H., Sun, H.: All-at-once method for variable-order time fractional diffusion equations. Numer. Algor. (2021). https://doi.org/10.1007/s11075-021-01178-7
21. Lu, X., Pang, H., Sun, H.: Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations. Numer. Linear Alg. Appl. **22**, 866–882 (2015)
22. Schumer, R., Benson, D.A., Meerschaert, M.M., Baeumer, B.: Fractal mobile/immobile solute transport. Water Resour. Res. **39**, 1–12 (2003)
23. 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)
24. 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)
25. Sun, Z., Wu, X.: A fully discrete difference scheme for a diffusion-wave system. Appl. Numer. Math. **56**, 193–209 (2006)
26. Thomée, V.: In: Galerkin Finite Element Methods for Parabolic Problems. Lecture Notes in Mathematics, vol. 1054. Springer-Verlag, New York (1984)
27. Wang, H., Zheng, X.: Wellposedness and regularity of the variable-order time-fractional diffusion equations. J. Math. Anal. Appl. **475**, 1778–1802 (2019)
28. Zayernouri, M., Karniadakis, G.E.: Fractional Sturm-Liouville eigen-problems: theory and numerical approximations. J. Comp. Phys. **47**, 2108–2131 (2013)
29. 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)
30. Zhang, Y., Benson, D., Reeves, D.: Time and space nonlocalities underlying fractional derivative models: distinction and literature review of field applications. Adv. Water Resources **32**, 561–581 (2009)
31. 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)
32. 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)
33. Zhao, X., Hu, X., Cai, W., Karniadakis, G.E.: Adaptive finite element method for fractional differential equations using hierarchical matrices. Comput. Meth. Appl. Mech. Eng. **325**, 56–76 (2017)
34. Zheng, X., Wang, H.: An error estimate of a numerical approximation to a hidden-memory variable-order space-time fractional diffusion equation. SIAM J. Numer. Anal. **58**, 2492–2514 (2020)
35. Zheng, X., Wang, H.: Wellposedness and smoothing properties of history-state-based variable-order time-fractional diffusion equations. Z. Agnew. Math. Phys. **71**, 34 (2020)
36. Zheng, X., Wang, H.: A hidden-memory variable-order fractional optimal control model: analysis and approximation. SIAM J. Control Optim. **59**, 1851–1880 (2021)
37. 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)