Journal of Scientific Computing (2024) 98:59
https://doi.org/10.1007/s10915-024-02454-z

A Positivity-Preserving and Robust Fast Solver for Time-Fractional Convection–Diffusion Problems

Boyang Yu[1] · Yonghai Li[1] · Jiangguo Liu[2]

Received: 12 October 2023 / Revised: 30 December 2023 / Accepted: 5 January 2024

Abstract
This paper presents a fast solver for time-fractional two-dimensional convection-diffusion problems that maintains non-negativity of numerical solutions. To this end, two new techniques are developed. (i) A three-part decomposition of the L1 discretization for Caputo derivatives is proposed and justified for fast evaluation while maintaining positivity; (ii) A positivity-correction technique is devised for both diffusive and convective fluxes. An upwinding technique for the bilinear finite volume approximation on general quadrilaterals is utilized for enabling the solver robustness in handling convection dominance. The solver attains optimal convergence rates when graded temporal meshes are used. These properties are theoretically justified and numerically illustrated.

Keywords Caputo derivatives · Fast numerical solver · Finite volume method · Positivity-preserving · Time-fractional convection-diffusion · Upwinding

Mathematics Subject Classification 65M08 · 65M12 · 76R99 · 26A33 · 35R11

1 Introduction

This paper is concerned with fast numerical solvers with certain desired properties, e.g., non-negativity of numerical solutions, for time-fractional 2-dimensional convection-diffusion boundary initial value problems prototyped as

✉ Jiangguo Liu
liu@math.colostate.edu

Boyang Yu
yuby21@mails.jlu.edu.cn

Yonghai Li
yonghai@jlu.edu.cn

[1] School of Mathematics, Jilin University, Changchun 130012, China

[2] Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA

Published online: 05 February 2024

$$\begin{cases} \partial_t^\alpha u + \nabla \cdot (\boldsymbol{b}u - A\nabla u) = f, & \text{in } \Omega \times (0, T], \\ u(x, y, t) = g_1(x, y, t), & \text{on } \partial\Omega \times (0, T], \\ u(x, y, 0) = g_2(x, y), & \text{in } \Omega, \end{cases} \tag{1.1}$$

where $\partial_t^\alpha u$, $\alpha \in (0, 1)$ is the Caputo derivative defined as

$$\partial_t^\alpha u(t) := \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial_s u(s)}{(t-s)^\alpha} ds. \tag{1.2}$$

Here, $\Omega \subset \mathbb{R}^2$ is a bounded connected open domain with a Lipschitz boundary $\partial\Omega$, $T > 0$ the final time, $\boldsymbol{b}$ a known velocity, $A > 0$ a constant for diffusivity, $u(x, y, t)$ the unknown concentration for the substance being transported, $f \geq 0$ a known source, and $g_1 \geq 0$, $g_2 \geq 0$ boundary and initial data.

Fractional order partial differential equations (PDEs) have been attracting significant research efforts, since they provide models for many problems in science and engineering [6, 16, 40], biology [17], medical and health science [24], and finance [11]. A new collection of such problems up to 2018 was presented in [38]. In particular, the equation in (1.1) can be used to model gas transport through heterogeneous reservoirs [8].

For discretization of the Caputo derivative, the L1 scheme based on linear approximation of the integrand is a popular choice [22, 27, 33, 45], whereas the L2 schemes based on quadratic approximation of the integrand [13, 30] can be used to match higher order spatial approximations. The L2-1_σ discretization provides a more delicate choice [1].

Due to the nonlocal nature of fractional order derivatives [9], their discretizations involve solution values at all spatial nodes/elements and/or all previous time steps. This results in high computational costs. Various types of techniques have been investigated for development of fast numerical solvers. Numerical methods based on the concept of nested meshes were proposed in [10, 12]. Based on the integral representations of the singular kernels, kernel compression techniques were developed in [2–4]. Based on approximation of a negative power kernel by sum-of-exponentials (SOE) [5], a new set of fast solvers have been developed recently [19, 37, 43, 51]. As demonstrated in [47], fast Poisson solvers for spatial discretization can also be utilized for time-fractional subdiffusion problems. However, the fast solvers developed for time-fractional subdiffusion problems may not be extended directly to fast solvers for time-fractional convection-diffusion problems when the positivity of numerical solutions is concerned.

Positivity or non-negativity of numerical solutions is an important aspect of PDEs. For general PDEs, there have been many mature results. In [14, 41, 46], a monotone finite volume scheme for diffusion equations on polygonal and general quadrilateral meshes was proposed. Some work on the finite element method can be found in [28]. A cut off method for the numerical computation of nonnegative solutions of parabolic PDEs is studied in [29]. However, only few work addressed such an issue of fractional order PDEs, e.g., [21] for a class of piecewise linear finite element approximations for subdiffusion equations; [49] for a maximum-principle preserving scheme for the time-fractional Allen-Cahn equation. As of our best knowledge, there is not yet a known fast solver for time-fractional convection-diffusion problems that preserves positivity, although recent developments of numerical methods can be found in [7, 18, 26, 31, 32, 34, 35, 42, 48, 50]. Some work on the meshless methods can be found in [39, 52].

This paper intends to fill such gaps. We take a comprehensive approach to develop a numerical solver for time-fractional convection-diffusion problems that has several desired properties,

e.g., preserving positivity, robust in handling convectional dominance, attaining optimal convergence rates, and being a fast solver.

(i) For discretization of the Caputo derivative, we still consider L1 discretization and approximation by sums of exponentials, but we propose a three-part decomposition (current, transition, and history terms) that will play a key role in positivity-preserving. It will be shown that a fast solver based on the conventional 2-part decomposition fails to preserve positivity. Our fast solver based on the 3-part decomposition combined with the graded temporal meshes will attain optimal temporal convergence rates.
(ii) For spatial discretization, we use bilinear finite volumes for general quadrilaterals. It has been recognized that quadrilateral meshes are equally flexible as triangular meshes for accommodating complicated domain geometry [15].
(iii) As motivated by the work [25], a new upwinding technique for bilinear finite volumes is developed, which allows the solver to handle well convectional dominance.
(iv) A positivity-correction technique is developed for both diffusive and convective fluxes. This contributes to a slightly nonlinear approximation, which is implemented via Picard iterations. It will be discussed later such nonlinearization will be worthwhile in maintaining positivity. A combination of the correction technique and the new upwinding technique ensures the optimal spatial convergence rate.

The rest of this paper is organized as follows. Section 2 reviews the L1 discretization and then proposes a 3-term decomposition for a modified fast evaluation algorithm (MFL1). Section 3 describes a new upwinding technique for the finite volume discretization on general quadrilateral meshes. Section 4 presents a positivity-correction technique for both convective and diffusive fluxes. Section 5 describes our new solver that combines MFL1 and flux-correction and its implementation based on Picard iterations. Section 6 elaborates on the positivity-preserving property and computational efficiency of this solver. Section 7 presents numerical tests to demonstrate convergence rates, positivity-preserving property, and efficiency of the solver. The paper is concluded with some remarks in Sect. 8.

2 A Modified Fast L1 Evaluation Algorithm for Caputo Derivatives

This section briefly reviews the L1 discretization for Caputo derivatives and the conventional fast evaluation algorithm based on a two-part decomposition and approximation of a negative power kernel by sums of exponentials (SOE). Then we propose a modified fast L1 algorithm (MFL1) that will play an important role in positivity-preserving.

2.1 L1 Discretization and SOE Approximation

The L1 discretization is based on a piecewise linear approximation of function $u(\cdot)$ in the integrand. Assume the time interval $[0, T]$ has a partition $t_n = T(n/N_T)^r$ for $n = 0, 1, \ldots, N_T$, where $r \geq 1$. Let $\tau_n = t_n - t_{n-1}$ and $\tau_{n,k} = t_n - t_k$ for $n \geq k \geq 0$ and $n = 1, 2, \ldots, N_T$. For convenience, we denote $u(t_n)$ as $u^{(n)}$. The piecewise linear approximant is expressed as

$$(\Pi_k u)(t) = u^{(k-1)} \frac{t_k - t}{\tau_k} + u^{(k)} \frac{t - t_{k-1}}{\tau_k}, \quad \forall t \in [t_{k-1}, t_k], \quad 1 \leq k \leq N_T. \tag{2.1}$$

Its derivative is a piecewise constant

$$(\Pi_k u)'(t) = \frac{u^{(k)} - u^{(k-1)}}{\tau_k}, \quad \forall t \in (t_{k-1}, t_k). \tag{2.2}$$

This implies that, for any t_n with $1 \le n \le N_T$,

$$\begin{aligned} \partial_t^\alpha u(t_n) &= \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} \frac{u'(s)}{(t_n - s)^\alpha} ds \approx \frac{1}{\Gamma(1-\alpha)} \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \frac{(\Pi_k u)'(s)}{(t_n - s)^\alpha} ds \\ &= \frac{1}{\Gamma(1-\alpha)} \sum_{k=1}^{n} \frac{u^{(k)} - u^{(k-1)}}{\tau_k} \int_{t_{k-1}}^{t_k} \frac{ds}{(t_n - s)^\alpha} =: \mathrm{D}_{L1}^\alpha u^{(n)}. \end{aligned} \tag{2.3}$$

Direct calculations of the above integrals yield, for $1 \le n \le N_T$,

$$\mathrm{D}_{L1}^\alpha u^{(n)} = \frac{1}{\Gamma(2-\alpha)} \sum_{k=1}^{n} \frac{u^{(k)} - u^{(k-1)}}{\tau_k} (\tau_{n,k-1}^{1-\alpha} - \tau_{n,k}^{1-\alpha}). \tag{2.4}$$

Now we rewrite the above discretization formula of the Caputo derivative as

$$\mathrm{D}_{L1}^\alpha u^{(n)} = \frac{d_{n,1}}{\Gamma(2-\alpha)} u^{(n)} - \frac{d_{n,n}}{\Gamma(2-\alpha)} u^{(0)} - \sum_{k=1}^{n-1} \frac{d_{n,k} - d_{n,k+1}}{\Gamma(2-\alpha)} u^{(n-k)}, \tag{2.5}$$

where

$$d_{n,k} := \frac{\tau_{n,n-k}^{1-\alpha} - \tau_{n,n-k+1}^{1-\alpha}}{\tau_{n-k+1}}, \quad 1 \le k \le n. \tag{2.6}$$

It is easy to prove that

$$d_{n,k} \ge 0 \ \text{ for } \ 1 \le k \le n; \qquad d_{n,k} - d_{n,k+1} \ge 0 \ \text{ for } \ 1 \le k \le n-1. \tag{2.7}$$

Remark 1 Note that in the L1 discretization (2.5), the coefficients of the history layers $u^{(k)}$ $(k = 0, \ldots, n-1)$ are negative, whereas the coefficient of the current layer $u^{(n)}$ is positive.

Computational costs for numerically solving time-fractional PDEs would be very high if the direct L1 discretization formula was used. Fast evaluation algorithms have been developed thanks to approximation by a sum of exponentials (SOE) to the negative power kernel in the definition of the Caputo derivative.

Lemma 1 *(Approximation to a negative power by SOE). For any fractional exponent $\beta \in (0, 1)$, an error tolerance $\varepsilon \in (0, e^{-1}]$, and a cut-off time $\delta \in (0, 1]$, there exist a positive integer N_{exp}, positive constants λ_j and positive weights θ_j for $j = 1, 2, \ldots, N_{exp}$, such that the relative error*

$$\left| t^{-\beta} - \sum_{j=1}^{N_{exp}} \theta_j e^{-\lambda_j t} \right| \Big/ t^{-\beta} \le \varepsilon, \quad \forall t \in [\delta, 1]. \tag{2.8}$$

Discussion of selection of N_{exp}, λ_j and θ_j can be found in [5].

2.2 A Modified Fast L1 Evaluation Algorithm

Recall the $L1$ discretization involves

$$\mathrm{D}_{L1}^{\alpha} u^{(n)} = \frac{1}{\Gamma(1-\alpha)} \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \frac{(\Pi_k u)'(s)}{(t_n - s)^{\alpha}} ds. \tag{2.9}$$

For the conventional fast algorithm based on SOE approximation [19], a common practice is to split the sum (2.9) into two parts. The current term for interval $[t_{n-1}, t_n]$ is approximated directly via L1, whereas the history term for interval $[t_0, t_{n-1}]$, which is related to the "long-tail", is approximated by SOE. However, this conventional fast algorithm fails to preserve non-negativity of numerical solutions. Here, we propose a new algorithm that splits the sum (2.9) into three parts (for $n \geq 3$) as follows.

– ***The current term***

$$I_C(t_n) = \frac{1}{\Gamma(1-\alpha)} \int_{t_{n-1}}^{t_n} \frac{(\Pi_n u)'(s)}{(t_n - s)^{\alpha}} ds; \tag{2.10}$$

– ***A transitional term***

$$I_T(t_n) = \frac{1}{\Gamma(1-\alpha)} \int_{t_{n-2}}^{t_{n-1}} \frac{(\Pi_{n-1} u)'(s)}{(t_n - s)^{\alpha}} ds; \tag{2.11}$$

– ***The history term***

$$I_H(t_n) = \frac{1}{\Gamma(1-\alpha)} \sum_{k=1}^{n-2} \int_{t_{k-1}}^{t_k} \frac{(\Pi_k u)'(s)}{(t_n - s)^{\alpha}} ds. \tag{2.12}$$

The current term $I_C(t_n)$ on $[t_{n-1}, t_n]$ can still be handled by direct L1 approximation

$$I_C(t_n) = \frac{1}{\Gamma(1-\alpha)} \int_{t_{n-1}}^{t_n} \frac{(\Pi_n u)'(s)}{(t_n - s)^{\alpha}} ds = \frac{d_{n,1}}{\Gamma(2-\alpha)} \cdot \left(u^{(n)} - u^{(n-1)} \right). \tag{2.13}$$

For the transitional term, we apply the SOE approximation Lemma 1 partially,

$$\begin{aligned}
I_T(t_n) &= \frac{1}{\Gamma(1-\alpha)} \int_{t_{n-2}}^{t_{n-1}} (\Pi_{n-1} u)'(s) \, (t_n - s)^{-\alpha} \, ds \\
&= \frac{1}{\Gamma(1-\alpha)} \int_{t_{n-2}}^{t_{n-1}} \frac{u^{(n-1)}}{\tau_{n-1}} (t_n - s)^{-\alpha} \, ds - \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \int_{t_{n-2}}^{t_{n-1}} \frac{u^{(n-2)}}{\tau_{n-1}} \left(\frac{t_n - s}{T} \right)^{-\alpha} ds \\
&\approx \frac{d_{n,2}}{\Gamma(2-\alpha)} u^{(n-1)} - \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j \int_{t_{n-2}}^{t_{n-1}} \frac{u^{(n-2)}}{\tau_{n-1}} \, e^{-\lambda_j (t_n - s)/T} ds \\
&= \frac{d_{n,2}}{\Gamma(2-\alpha)} u^{(n-1)} - \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j \frac{e^{-\lambda_j (\tau_n / T)} - e^{-\lambda_j (\tau_{n,n-2}/T)}}{\lambda_j \tau_{n-1}/T} \, u^{(n-2)}.
\end{aligned} \tag{2.14}$$

The history term $I_H(t_n)$ on $[0, t_{n-2}]$ causing the "long-tail" needs to be reformulated. Applying Lemma 1, we obtain

$$\begin{aligned} I_H(t_n) &= \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{k=1}^{n-2} \int_{t_{k-1}}^{t_k} (\Pi_k u)'(s) \Big(\frac{t_n - s}{T}\Big)^{-\alpha} ds \\ &\approx \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{k=1}^{n-2} \int_{t_{k-1}}^{t_k} (\Pi_k u)'(s) \sum_{j=1}^{N_{exp}} \theta_j e^{-\lambda_j (t_n - s)/T} ds \\ &= \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j \sum_{k=1}^{n-2} \int_{t_{k-1}}^{t_k} (\Pi_k u)'(s)\, e^{-\lambda_j (t_n - s)/T} ds. \end{aligned} \tag{2.15}$$

For convenience, we denote, for $2 \le n \le N_T$ and $1 \le j \le N_{exp}$,

$$w_j^{(n)} = \sum_{k=1}^{n-2} \int_{t_{k-1}}^{t_k} (\Pi_k u)'(s)\, e^{-\lambda_j (t_n - s)/T} ds. \tag{2.16}$$

Specifically, $w_j^{(2)} = 0$. We split the above sum and perform direct calculations to obtain

$$\begin{aligned} w_j^{(n)} &= \sum_{k=1}^{n-3} \int_{t_{k-1}}^{t_k} (\Pi_k u)'(s)\, e^{-\lambda_j (t_n - s)/T} ds + \int_{t_{n-3}}^{t_{n-2}} (\Pi_{n-2} u)'(s)\, e^{-\lambda_j (t_n - s)/T} ds \\ &= e^{-\lambda_j (\tau_n/T)} \sum_{k=1}^{n-3} \int_{t_{k-1}}^{t_k} (\Pi_k u)'(s)\, e^{-\lambda_j (t_{n-1} - s)/T} ds + \int_{t_{n-3}}^{t_{n-2}} \frac{u^{(n-2)} - u^{(n-3)}}{\tau_{n-2}} e^{-\lambda_j (t_n - s)/T} ds, \end{aligned} \tag{2.17}$$

which yields, for $3 \le n \le N_T$ and $1 \le j \le N_{exp}$,

$$w_j^{(n)} = e^{-\lambda_j (\tau_n/T)} w_j^{(n-1)} + \frac{e^{-\lambda_j (\tau_{n,n-2}/T)} - e^{-\lambda_j (\tau_{n,n-3}/T)}}{\lambda_j \tau_{n-2}/T} \big(u^{(n-2)} - u^{(n-3)}\big). \tag{2.18}$$

Shown below is our ***modified fast L1 evaluation algorithm (MFL1)*** for the Caputo derivative.

– For $n = 1, 2$, this algorithm is the direct $L1$ evaluation formula

$$\mathrm{D}_F^{\alpha} u^{(n)} = \mathrm{D}_{L1}^{\alpha} u^{(n)} = \frac{d_{n,1}}{\Gamma(2-\alpha)} u^{(n)} - \frac{d_{n,n}}{\Gamma(2-\alpha)} u^{(0)} - \sum_{k=1}^{n-1} \frac{d_{n,k} - d_{n,k+1}}{\Gamma(2-\alpha)} u^{(n-k)}. \tag{2.19}$$

– For $n = 3, 4, \cdots, N_T$, we have

$$\begin{aligned} \mathrm{D}_F^{\alpha} u^{(n)} &= \frac{d_{n,1}}{\Gamma(2-\alpha)} u^{(n)} - \frac{d_{n,1} - d_{n,2}}{\Gamma(2-\alpha)} u^{(n-1)} \\ &\quad - \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j \frac{e^{-\lambda_j (\tau_n/T)} - e^{-\lambda_j (\tau_{n,n-2}/T)}}{\lambda_j \tau_{n-1}/T} u^{(n-2)} + \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j w_j^{(n)}, \end{aligned} \tag{2.20}$$

where the auxiliary quantity $w_j^{(n)}$ satisfies a recurrence formula stated above but reformulated as follows

$$\begin{cases} w_j^{(n)} = e^{-\lambda_j (\tau_n/T)} w_j^{(n-1)} + \dfrac{e^{-\lambda_j (\tau_{n,n-2}/T)} - e^{-\lambda_j (\tau_{n,n-3}/T)}}{\lambda_j \tau_{n-2}/T} \big(u^{(n-2)} - u^{(n-3)}\big), \\ w_j^{(2)} = 0, \qquad \forall 1 \le j \le N_{exp}. \end{cases} \tag{2.21}$$

Next we show that the MFL1 algorithm maintains certain properties of the L1 discretization.

Theorem 1 *For the MFL1 algorithm,* $\mathrm{D}_F^{\alpha} u^{(n)}$ *has the following properties.*

(i) The coefficient of the current layer $u^{(n)}$ *is positive;*
(ii) The coefficients of the history layers $u^{(k)}(k = 0, \ldots, n-1)$ *are negative.*

Proof It is clear from (2.19) and (2.20) that the coefficient of the current layer $u^{(n)}$ is positive. Yes, the MFL1 algorithm satisfies Property (i).

For $n = 1, 2$, we know that $\mathrm{D}_F^{\alpha} u^{(n)} = \mathrm{D}_{L1}^{\alpha} u^{(n)}$ from (2.19). Property (ii) holds for $\mathrm{D}_F^{\alpha} u^{(n)}, n = 1, 2$. To ease presentation, for $n \geq 3$, we denote

$$v_j^{(n)} = -\frac{e^{-\lambda_j(\tau_n/T)} - e^{-\lambda_j(\tau_{n,n-2}/T)}}{\lambda_j \tau_{n-1}/T} u^{(n-2)} + w_j^{(n)}. \tag{2.22}$$

Then (2.20) can be written as

$$\mathrm{D}_F^{\alpha} u^{(n)} = \frac{d_{n,1}}{\Gamma(2-\alpha)} u^{(n)} - \frac{d_{n,1} - d_{n,2}}{\Gamma(2-\alpha)} u^{(n-1)} + \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j v_j^{(n)}. \tag{2.23}$$

According to (2.7), the coefficient of $u^{(n-1)}$ is negative.

Next, we prove that the coefficients of time layers in $v_j^{(n)}$ $(n \geq 3)$ are negative by math induction. When $n = 3$, from (2.20) and (2.21), we obtain

$$v_j^{(3)} = -\frac{e^{-\lambda_j(\tau_3/T)} - e^{-\lambda_j(\tau_{3,1}/T)}}{\lambda_j \tau_2/T} u^{(1)} + \frac{e^{-\lambda_j(\tau_{3,1}/T)} - e^{-\lambda_j(\tau_{3,0}/T)}}{\lambda_j \tau_1/T} \left(u^{(1)} - u^{(0)}\right). \tag{2.24}$$

It is easy to see that the coefficient for $u^{(0)}$ is negative. As for the coefficient of $u^{(1)}$, we fix $1 \leq j \leq N_{exp}$ and then apply the Mean Value Theorem to obtain

$$\frac{e^{-\lambda_j(\tau_{3,1}/T)} - e^{-\lambda_j(\tau_{3,0}/T)}}{\lambda_j \tau_1/T} - \frac{e^{-\lambda_j(\tau_3/T)} - e^{-\lambda_j(\tau_{3,1}/T)}}{\lambda_j \tau_2/T} < 0. \tag{2.25}$$

Thus the coefficients of time layers in $v_j^{(3)}$ are indeed negative.

By induction hypothesis, the coefficients of time layers in $v_j^{(n)}(n = 3, 4, \ldots, l-1)$ are negative. According to (2.18) and (2.22), we have

$$v_j^{(l)} = e^{\lambda_j(\tau_l/T)} v_j^{(l-1)} + \left(\frac{e^{-\lambda_j(\tau_{l,l-2}/T)} - e^{-\lambda_j(\tau_{l,l-3}/T)}}{\lambda_j \tau_{l-2}/T} - \frac{e^{-\lambda_j(\tau_l/T)} - e^{-\lambda_j(\tau_{l,l-2}/T)}}{\lambda_j \tau_{l-1}/T} \right) u^{(l-2)}. \tag{2.26}$$

Similarly, the Mean Value Theorem implies that

$$\frac{e^{-\lambda_j(\tau_{l,l-2}/T)} - e^{-\lambda_j(\tau_{l,l-3}/T)}}{\lambda_j \tau_{l-2}/T} - \frac{e^{-\lambda_j(\tau_l/T)} - e^{-\lambda_j(\tau_{l,l-2}/T)}}{\lambda_j \tau_{l-1}/T} < 0. \tag{2.27}$$

So the claim about the time layer coefficients for $v_j^{(l)}$ holds. Property (ii) holds by mathematical induction. □

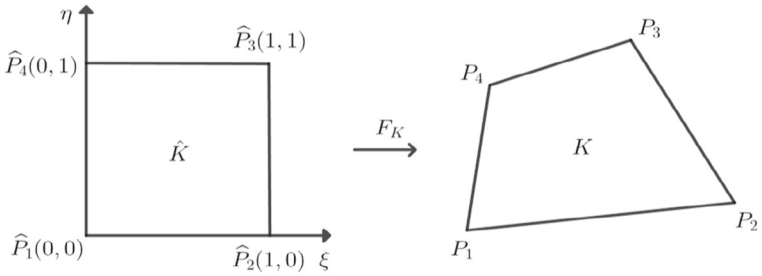


Fig. 1 A bilinear mapping F_K

Remark 2 The conventional fast L1 algorithm (CFL1) based on a two-part decomposition [19] does not satisfy Property (ii). The correction technique in (4.30) and (4.31) (to be elaborated on later) will not work under CFL1. The modified fast L1 algorithm (MFL1) based on the above three-part decomposition will play a key role in preserving positivity of numerical solutions.

3 Upwinding for Bilinear Finite Volume Discretization

Let $\mathcal{T}_h = \{K\}$ be a quadrilateral mesh on $\overline{\Omega}$, where K represents a typical quadrilateral and h denotes the mesh size. Let $\mathcal{P}_h$ be the set of all vertices and N_P be the number of vertices. Let $\hat{K} = [0,1]^2$ be the reference element with coordinates (ξ, η). We consider a typical quadrilateral K with vertices $P_i = (x_i, y_i)(i = 1, 2, 3, 4)$ ordered in the counterclockwise orientation. There exists a unique invertible bilinear mapping F_K from $\hat{K}$ to K (see Fig. 1):

$$\begin{cases} x = x_1 + a_1\xi + a_2\eta + a_3\xi\eta, \\ y = y_1 + b_1\xi + b_2\eta + b_3\xi\eta, \end{cases} \tag{3.1}$$

where

$$\begin{cases} a_1 = x_2 - x_1,\ a_2 = x_4 - x_1,\ a_3 = x_3 - x_4 - x_2 + x_1, \\ b_1 = y_2 - y_1,\ b_2 = y_4 - y_1,\ b_3 = y_3 - y_4 - y_2 + y_1. \end{cases} \tag{3.2}$$

The Jacobian matrix of the mapping F_K is

$$\mathbf{J}_K(\xi,\eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} a_1 + a_3\eta & a_2 + a_3\xi \\ b_1 + b_3\eta & b_2 + b_3\xi \end{bmatrix}. \tag{3.3}$$

Denote the Jacobian determinant as $J_K(\xi, \eta)$. By direct calculations, we get

$$\begin{aligned} \nabla\xi &= \left[\frac{\partial \xi}{\partial x}, \frac{\partial \xi}{\partial y}\right]^\top = ((1-\xi)\boldsymbol{q_{14}} + \xi\boldsymbol{q_{23}})/J_K(\xi,\eta), \\ \nabla\eta &= \left[\frac{\partial \eta}{\partial x}, \frac{\partial \eta}{\partial y}\right]^\top = ((1-\eta)\boldsymbol{q_{21}} + \eta\boldsymbol{q_{34}})/J_K(\xi,\eta), \end{aligned} \tag{3.4}$$

where $\boldsymbol{q_{ij}}$ is obtained by rotating the vector $\overrightarrow{P_iP_j}$ by $\pi/2$ clockwise (see Fig. 2). We denote

$$\boldsymbol{q_1}(\xi) = (1-\xi)\boldsymbol{q_{14}} + \xi\boldsymbol{q_{23}}, \qquad \boldsymbol{q_2}(\eta) = (1-\eta)\boldsymbol{q_{21}} + \eta\boldsymbol{q_{34}}. \tag{3.5}$$

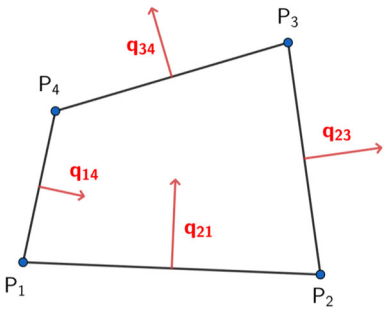


Fig. 2 The normal vectors on the edges of a quadrilateral

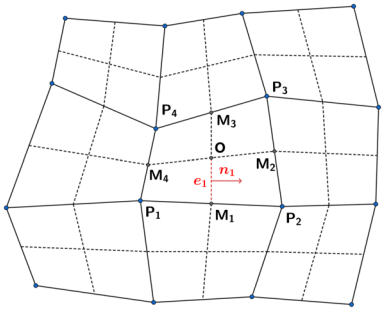


Fig. 3 The dual elements surrounding the primal vertices P_1, P_2, P_3, P_4

Then

$$\nabla\xi = \frac{\boldsymbol{q_1}(\xi)}{J_K(\xi,\eta)}, \qquad \nabla\eta = \frac{\boldsymbol{q_2}(\eta)}{J_K(\xi,\eta)}. \tag{3.6}$$

Let $\mathcal{U}_h(\hat{K})$ be the standard bilinear polynomial space on $\hat{K}$. Define the trial space as

$$\mathcal{U}_h = \{u_h \in C(\overline{\Omega}) :\ u_h|_K = \hat{u}_h \circ F_K^{-1},\ \hat{u}_h \in \mathcal{U}_h(\hat{K}),\ \forall K \in \mathcal{T}_h\} = \text{Span}\{\phi_P :\ P \in \mathcal{P}_h\}, \tag{3.7}$$

where ϕ_P represents a typical nodal basis function. For any shape function $u_h \in \mathcal{U}_h$, the reference shape function $\hat{u}_h$ corresponding to $u_h|_K$ can be expressed as

$$\hat{u}_h = u_{P_1}(1-\xi)(1-\eta) + u_{P_2}\xi(1-\eta) + u_{P_3}\xi\eta + u_{P_4}(1-\xi)\eta. \tag{3.8}$$

By combining Formulas (3.6) and (3.8), we obtain the gradient of $u_h|_K$ as follows.

$$\begin{aligned}\nabla(u_h|_K) &= \frac{\partial\hat{u}_h}{\partial\xi}\nabla\xi + \frac{\partial\hat{u}_h}{\partial\eta}\nabla\eta \\ &= (u_{P_2}-u_{P_1})(1-\eta)\frac{\boldsymbol{q_1}(\xi)}{J_K(\xi,\eta)} + (u_{P_3}-u_{P_4})\eta\frac{\boldsymbol{q_1}(\xi)}{J_K(\xi,\eta)} \\ &+ (u_{P_4}-u_{P_1})(1-\xi)\frac{\boldsymbol{q_2}(\eta)}{J_K(\xi,\eta)} + (u_{P_3}-u_{P_2})\xi\frac{\boldsymbol{q_2}(\eta)}{J_K(\xi,\eta)}.\end{aligned} \tag{3.9}$$

Let $\mathcal{T}_h^*$ be the dual mesh corresponding to the primary mesh $\mathcal{T}_h$. A dual element is a polygon centred at a given node and enclosed by zig-zag line segments that connect the midpoints of the adjacent edges and the centers of the surrounding primal volumes (see Fig. 3). We define the test function space as the space of piecewise constants on the dual mesh

$$\mathcal{V}_h = \{v_h \in L^2(\overline{\Omega}) :\ v_h|_{K_P^*} = \text{constant},\ \forall K_P^* \in \mathcal{T}_h^*\} = \text{Span}\{\psi_P :\ P \in \mathcal{P}_h\}, \tag{3.10}$$

where ψ_P is the characteristic function for K_P^*.

The finite volume bilinear form for diffusion is defined as

$$\mathcal{A}_h(u_h, v_h) = -\sum_{K_P^* \in \mathcal{T}_h^*} \int_{\partial K_P^*} A\nabla u_h \cdot \boldsymbol{n}\, v_h\, ds, \quad \forall u_h \in \mathcal{U}_h,\ \forall v_h \in \mathcal{V}_h, \tag{3.11}$$

where $\boldsymbol{n}$ is the outward unit normal vector on ∂K_P^*.

Now we introduce the ***pointwise average gradient*** on a shared edge. Let $K_1 \big| K_2$ be the common edge of two adjacent elements K_1 and K_2 in $\mathcal{T}_h$. Define the average gradient of v at $(x, y) \in K_1 \big| K_2$ as

$$\overline{\nabla} v(x, y) = \frac{1}{2}\big((\nabla v|_{K_1})(x, y) + (\nabla v|_{K_2})(x, y)\big). \tag{3.12}$$

Consider e_1 as a line segment shared by two adjacent dual elements $K_{P_1}^*$ and $K_{P_2}^*$ (see Fig. 3). The reference coordinates corresponding to e_1 are $\frac{1}{2}$ and η with $\eta \in (0, \frac{1}{2})$. The upstream point $(\hat{x}(\eta), \hat{y}(\eta))$ is defined as

$$(\hat{x}(\eta), \hat{y}(\eta)) = \begin{cases} F_K(0, \eta), & \text{if } \int_{e_1} \boldsymbol{b} \cdot \boldsymbol{n}_1 \geq 0, \\ F_K(1, \eta), & \text{if } \int_{e_1} \boldsymbol{b} \cdot \boldsymbol{n}_1 \leq 0, \end{cases} \quad \eta \in \left(0, \frac{1}{2}\right), \tag{3.13}$$

where $\boldsymbol{n}_1$ is the outward unit normal vector for $K_{P_1}^*$ with respect to edge $M_1 O$. Then we obtain the upwind approximation of u at any point $(x_0, y_0) \in e_1$ as

$$u(x_0, y_0) \approx u^{up}(\eta_0) := u(\hat{x}(\eta_0), \hat{y}(\eta_0)) + \boldsymbol{v} \cdot \overline{\nabla} u(\hat{x}(\eta_0), \hat{y}(\eta_0)), \tag{3.14}$$

where $(\frac{1}{2}, \eta_0)$ are the reference coordinates corresponding to (x_0, y_0), and $\boldsymbol{v} = [x_0 - \hat{x}(\eta_0), y_0 - \hat{y}(\eta_0)]^\top$. Especially, if $(\hat{x}, \hat{y}) \in \partial\Omega$, then we take $\overline{\nabla} u(\hat{x}, \hat{y}) = \nabla u(\hat{x}, \hat{y})$.

Accordingly, the bilinear form for convection reads as

$$\mathcal{B}_h(u_h, v_h) = \sum_{K_P^* \in \mathcal{T}_h^*} \int_{\partial K_P^*} (\boldsymbol{b} \cdot \boldsymbol{n}) u_h^{up} v_h ds, \quad \forall u_h \in \mathcal{U}_h, \quad \forall v_h \in \mathcal{V}_h. \tag{3.15}$$

Thus, our semi-discrete upwinding finite volume scheme for the time-fractional 2-dim convection-diffusion Eq. (1.1) is formulated as

$$\left(\partial_t^\alpha u_h, v_h\right) + \mathcal{A}_h(u_h, v_h) + \mathcal{B}_h(u_h, v_h) = (f, v_h), \quad \forall v_h \in \mathcal{V}_h, \tag{3.16}$$

where

$$\left(\partial_t^\alpha u_h, v_h\right) = \sum_{K_P^* \in \mathcal{T}_h^*} \iint_{K_P^*} (\partial_t^\alpha u_h) v_h\, dxdy, \quad \forall v_h \in \mathcal{V}_h, \tag{3.17}$$

$$(f, v_h) = \sum_{K_P^* \in \mathcal{T}_h^*} \iint_{K_P^*} f v_h\, dxdy, \quad \forall v_h \in \mathcal{V}_h. \tag{3.18}$$

4 Flux Correction for Bilinear Finite Volume Discretization

The semi-discrete bilinear finite volume scheme (3.16) can be rewritten as

$$\left(\partial_t^\alpha u_h, \psi_P\right) + \mathcal{A}_h(u_h, \psi_P) + \mathcal{B}_h(u_h, \psi_P) = (f, \psi_P), \quad \forall P \in \mathcal{P}_h. \tag{4.1}$$

The discrete bilinear forms $\mathcal{A}_h(u_h, \psi_P)$ and $\mathcal{B}_h(u_h, \psi_P)$ involve line integrals along the boundary segments of a typical dual element K_P^*. In this section, our flux correction technique is established for these integral terms. For ease of presentation, assume e_1 is a common boundary segment of two adjacent dual elements $K_{P_1}^*$ and $K_{P_2}^*$.

4.1 Splitting of the Diffusive Flux

Plugging the test functions ψ_{P_1}, ψ_{P_2} into (3.11), respectively, we end up with the following integrals on the line segment e_1:

$$\mathcal{F}_{P_1,e_1} = -\int_{e_1} A\nabla u_h \cdot \boldsymbol{n}_1 ds, \qquad \mathcal{F}_{P_2,e_1} = -\int_{e_1} A\nabla u_h \cdot (-\boldsymbol{n}_1) ds. \tag{4.2}$$

Obviously, $\mathcal{F}_{P_1,e_1} + \mathcal{F}_{P_2,e_1} = 0$.

For a quadrilateral element K, the normal vector $\boldsymbol{n}_1$ can be written as

$$\boldsymbol{n}_1 = \frac{\boldsymbol{q_1}(\frac{1}{2})}{|\boldsymbol{q_1}(\frac{1}{2})|}. \tag{4.3}$$

Applying Formula (3.9), we obtain the gradient of $u_h|_{e_1}$ as shown below.

$$\begin{aligned} \nabla(u_h|_{e_1}) = {} & (u_{P_2} - u_{P_1})(1-\eta)\frac{\boldsymbol{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)} + (u_{P_3} - u_{P_4})\eta\frac{\boldsymbol{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)} \\ & + (u_{P_4} - u_{P_1})\Big(1-\frac{1}{2}\Big)\frac{\boldsymbol{q_2}(\eta)}{J_K(\frac{1}{2},\eta)} + (u_{P_3} - u_{P_2})\frac{1}{2}\frac{\boldsymbol{q_2}(\eta)}{J_K(\frac{1}{2},\eta)}, \end{aligned} \tag{4.4}$$

where $\eta \in [0, \frac{1}{2}]$. Then we have

$$\begin{aligned} \mathcal{F}_{P_1,e_1} &= -\int_0^{\frac{1}{2}} A\nabla(u_h|_{e_1})(\eta)\cdot \boldsymbol{q_1}\Big(\frac{1}{2}\Big)d\eta \\ &= \int_0^{\frac{1}{2}} A(1-\eta)\frac{|\boldsymbol{q_1}(\frac{1}{2})|^2}{J_K(\frac{1}{2},\eta)}d\eta\,(u_{P_1} - u_{P_2}) + \int_0^{\frac{1}{2}} A\eta\frac{|\boldsymbol{q_1}(\frac{1}{2})|^2}{J_K(\frac{1}{2},\eta)}d\eta\,(u_{P_4} - u_{P_3}) \\ &\quad + \int_0^{\frac{1}{2}} \frac{A}{2}\frac{\boldsymbol{q_2}(\eta)\cdot\boldsymbol{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)}d\eta\,(u_{P_1} - u_{P_4}) + \int_0^{\frac{1}{2}} \frac{A}{2}\frac{\boldsymbol{q_2}(\eta)\cdot\boldsymbol{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)}d\eta\,(u_{P_2} - u_{P_3}). \end{aligned} \tag{4.5}$$

Examining the 1st term of the right-hand side of (4.5), we note that the numerical flux $\mathcal{F}_{P_1,e_1}$ demonstrates a two-point flux structure $\gamma(u_{P_1} - u_{P_2})$ with $\gamma \geq 0$. Therefore, we split the numerical flux $\mathcal{F}_{P_1,e_1}$ into two parts: ***the major part with a two-point flux structure*** and a remainder. Specifically,

$$\mathcal{F}_{P_1,e_1} = \gamma_{e_1}(u_{P_1} - u_{P_2}) + R^d_{P_1,e_1}, \tag{4.6}$$

where

$$\gamma_{e_1} = \int_0^{\frac{1}{2}} A(1-\eta)\frac{|\boldsymbol{q_1}(\frac{1}{2})|^2}{J_K(\frac{1}{2},\eta)}d\eta, \tag{4.7}$$

and

$$R^d_{P_1,e_1} = \int_0^{\frac{1}{2}} A\eta \frac{|\boldsymbol{q_1}(\frac{1}{2})|^2}{J_K(\frac{1}{2},\eta)} d\eta \left(u_{P_4} - u_{P_3}\right) + \int_0^{\frac{1}{2}} \frac{A}{2} \frac{\boldsymbol{q_2}(\eta)\cdot\boldsymbol{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)} d\eta \left(u_{P_1} - u_{P_4}\right) + \int_0^{\frac{1}{2}} \frac{A}{2} \frac{\boldsymbol{q_2}(\eta)\cdot\boldsymbol{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)} d\eta \left(u_{P_2} - u_{P_3}\right). \tag{4.8}$$

Direct calculations yield

$$\mathcal{F}_{P_2,e_1} = \gamma_{e_1}(u_{P_2} - u_{P_1}) + R^d_{P_2,e_1}, \qquad R^d_{P_2,e_1} = -R^d_{P_1,e_1}. \tag{4.9}$$

4.2 Splitting of the Convective Flux

Now we consider the convection terms expressed as integrals on the line segment e_1:

$$\mathcal{G}_{P_1,e_1} = \int_{e_1} (\boldsymbol{b}\cdot\boldsymbol{n}_1) u_h^{up} ds, \qquad \mathcal{G}_{P_2,e_1} = -\int_{e_1} (\boldsymbol{b}\cdot\boldsymbol{n}_1) u_h^{up} ds. \tag{4.10}$$

Assume that $\int_{e_1} (\boldsymbol{b}\cdot\boldsymbol{n}_1) ds \geq 0$. According to (3.13), we have

$$(\hat{x}(\eta), \hat{y}(\eta)) = F_K(0,\eta), \quad \eta \in \left(0, \frac{1}{2}\right). \tag{4.11}$$

Combining (3.14), (4.3) and (4.11), we obtain

$$\begin{aligned}
\mathcal{G}_{P_1,e_1} &= \int_0^{\frac{1}{2}} \left(\hat{\boldsymbol{b}}\left(\frac{1}{2},\eta\right)\cdot\boldsymbol{q_1}\left(\frac{1}{2}\right)\right)\left(u_h(\hat{x}(\eta),\hat{y}(\eta)) + \hat{\boldsymbol{v}}(\eta)\cdot\overline{\nabla} u_h(\hat{x}(\eta),\hat{y}(\eta))\right) d\eta \\
&= \int_0^{\frac{1}{2}} \left(\hat{\boldsymbol{b}}\left(\frac{1}{2},\eta\right)\cdot\boldsymbol{q_1}\left(\frac{1}{2}\right)\right)\left((1-\eta)u_{P_1} + \eta u_{P_4} + \hat{\boldsymbol{v}}(\eta)\cdot\overline{\nabla} u_h(\hat{x}(\eta),\hat{y}(\eta))\right) d\eta \\
&= \int_0^{\frac{1}{2}} \left(\hat{\boldsymbol{b}}\left(\frac{1}{2},\eta\right)\cdot\boldsymbol{q_1}\left(\frac{1}{2}\right)\right) d\eta\, u_{P_1} - 0\times u_{P_2} \\
&\quad + \int_0^{\frac{1}{2}} \left(\hat{\boldsymbol{b}}\left(\frac{1}{2},\eta\right)\cdot\boldsymbol{q_1}\left(\frac{1}{2}\right)\right)\left(\eta\left(u_{P_4} - u_{P_1}\right) + \hat{\boldsymbol{v}}(\eta)\cdot\overline{\nabla} u_h(\hat{x}(\eta),\hat{y}(\eta))\right) d\eta,
\end{aligned} \tag{4.12}$$

where $\hat{\boldsymbol{b}}(\xi,\eta) = \boldsymbol{b}\circ F_K(\xi,\eta)$ and $\hat{\boldsymbol{v}}(\eta) = F_K(\frac{1}{2},\eta) - F_K(0,\eta)$.

We conduct a splitting similar to that in handling the diffusion term. We choose the terms containing u_{P_1} and u_{P_2} as the major part that demonstrates a quasi two-point flux structure, since u_{P_1},u_{P_2} are respectively the upstream and downstream nodes. Accordingly, the integral can be decomposed as

$$\mathcal{G}_{P_1,e_1} = \kappa_{e_1} u_{P_1} - 0\times u_{P_2} + R^c_{P_1,e_1}, \tag{4.13}$$

where

$$\kappa_{e_1} = \int_0^{\frac{1}{2}} \left(\hat{\boldsymbol{b}}\left(\frac{1}{2},\eta\right)\cdot\boldsymbol{q_1}\left(\frac{1}{2}\right)\right) d\eta \geq 0, \tag{4.14}$$

and

$$R^c_{P_1,e_1} = \int_0^{\frac{1}{2}} \left(\hat{\boldsymbol{b}}\left(\frac{1}{2},\eta\right)\cdot\boldsymbol{q_1}\left(\frac{1}{2}\right)\right)\left(\eta(u_{P_4} - u_{P_1}) + \hat{\boldsymbol{v}}(\eta)\cdot\overline{\nabla} u_h(\hat{x}(\eta),\hat{y}(\eta))\right) d\eta. \tag{4.15}$$

Similarly, we obtain a splitting as shown below.

$$\mathcal{G}_{P_2,e_1} = 0 \times u_{P_2} - \kappa_{e_1} u_{P_1} + R^c_{P_2,e_1}, \qquad R^c_{P_2,e_1} = -R^c_{P_1,e_1}. \tag{4.16}$$

4.3 Positivity-Correction for Diffusive and Convective Fluxes

Now we explain the technique for positivity correction. First, we introduce two integral terms

$$\mathcal{I}_{P_1,e_1} = \mathcal{F}_{P_1,e_1} + \mathcal{G}_{P_1,e_1} = \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - \gamma_{e_1} u_{P_2} + R^d_{P_1,e_1} + R^c_{P_1,e_1}, \tag{4.17}$$

$$\mathcal{I}_{P_2,e_1} = \mathcal{F}_{P_2,e_1} + \mathcal{G}_{P_2,e_1} = \gamma_{e_1} u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} + R^d_{P_2,e_1} + R^c_{P_2,e_1}. \tag{4.18}$$

Setting

$$R_{e_1} = R^d_{P_1,e_1} + R^c_{P_1,e_1} = -R^d_{P_2,e_1} - R^c_{P_2,e_1}, \tag{4.19}$$

we obtain

$$\mathcal{I}_{P_1,e_1} = \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - \gamma_{e_1} u_{P_2} + R_{e_1}, \tag{4.20}$$

$$\mathcal{I}_{P_2,e_1} = \gamma_{e_1} u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - R_{e_1}. \tag{4.21}$$

Next, we denote the positive and negative parts of R_{e_1} as

$$R^+_{e_1} = \frac{|R_{e_1}| + R_{e_1}}{2}, \qquad R^-_{e_1} = \frac{|R_{e_1}| - R_{e_1}}{2}. \tag{4.22}$$

The integrals can be rewritten as

$$\mathcal{I}_{P_1,e_1} = \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - \gamma_{e_1} u_{P_2} + R^+_{e_1} - R^-_{e_1}, \tag{4.23}$$

$$\mathcal{I}_{P_2,e_1} = \gamma_{e_1} u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - R^+_{e_1} + R^-_{e_1}. \tag{4.24}$$

Let B be an empirical large positive constant. Then we have

$$\begin{aligned}\mathcal{I}_{P_1,e_1} = &\left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR^+_{e_1}}{Bu_{P_1} + h^2}\right) u_{P_1} \\ &- \left(\gamma_{e_1} + \frac{BR^-_{e_1}}{Bu_{P_2} + h^2}\right) u_{P_2} + \frac{h^2 R^+_{e_1}}{Bu_{P_1} + h^2} - \frac{h^2 R^-_{e_1}}{Bu_{P_2} + h^2},\end{aligned} \tag{4.25}$$

$$\begin{aligned}\mathcal{I}_{P_2,e_1} = &\left(\gamma_{e_1} + \frac{BR^-_{e_1}}{Bu_{P_2} + h^2}\right) u_{P_2} \\ &- \left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR^+_{e_1}}{Bu_{P_1} + h^2}\right) u_{P_1} - \frac{h^2 R^+_{e_1}}{Bu_{P_1} + h^2} + \frac{h^2 R^-_{e_1}}{Bu_{P_2} + h^2}.\end{aligned} \tag{4.26}$$

Dropping the last two terms in (4.25) and (4.26), respectively, we obtain ***nonlinear*** approximations to $\mathcal{I}_{P_1,e_1}$ and $\mathcal{I}_{P_2,e_1}$ as follows.

$$\widetilde{\mathcal{I}}_{P_1,e_1} = \left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR^+_{e_1}}{Bu_{P_1} + h^2}\right) u_{P_1} - \left(\gamma_{e_1} + \frac{BR^-_{e_1}}{Bu_{P_2} + h^2}\right) u_{P_2}, \tag{4.27}$$

$$\widetilde{\mathcal{I}}_{P_2,e_1} = \left(\gamma_{e_1} + \frac{BR_{e_1}^-}{Bu_{P_2}+h^2}\right) u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR_{e_1}^+}{Bu_{P_1}+h^2}\right) u_{P_1}. \tag{4.28}$$

Note that $\widetilde{\mathcal{I}}_{P_1,e_1} + \widetilde{\mathcal{I}}_{P_2,e_1} = 0$, which means the corrected finite volume method satisfies ***local mass conservation***. Note also that when the node is on the domain boundary, the corrected integral terms appear as

$$\begin{cases} \widetilde{\mathcal{I}}_{P_1,e_1} = \left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR_{e_1}^+}{Bu_{P_1}+h^2}\right) u_{P_1} - \gamma_{e_1} u_{P_2} - R_{e_1}^-, \quad \text{if } P_2 \in \partial\Omega, \\ \widetilde{\mathcal{I}}_{P_2,e_1} = \left(\gamma_{e_1} + \frac{BR_{e_1}^-}{Bu_{P_2}+h^2}\right) u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - R_{e_1}^+, \quad \text{if } P_1 \in \partial\Omega. \end{cases} \tag{4.29}$$

Finally, the fully discrete finite volume scheme with flux positivity-correction read as

$$\widetilde{\mathcal{A}}_h(u_h, \psi_{P_1}) + \widetilde{\mathcal{B}}_h(u_h, \psi_{P_1}) = \sum_{e \subset \partial K_{P_1}^*} \widetilde{\mathcal{I}}_{P_1,e}, \tag{4.30}$$

$$\widetilde{\mathcal{A}}_h(u_h, \psi_{P_2}) + \widetilde{\mathcal{B}}_h(u_h, \psi_{P_2}) = \sum_{e \subset \partial K_{P_2}^*} \widetilde{\mathcal{I}}_{P_2,e}. \tag{4.31}$$

5 A Positivity-Preserving Fast Solver for Time-fractional Convection-Diffusion Problems

Combining the techniques and results in Sections 2–4, we establish a novel numerical scheme (***MFL1-Correction***) for the time-fractional convection-diffusion equation in (1.1) that seeks $u_h^{(n)} \in U_h$ such that

$$\left(\mathrm{D}_F^\alpha u_h^{(n)}, \psi_P\right) + \widetilde{\mathcal{A}}_h(u_h^{(n)}, \psi_P) + \widetilde{\mathcal{B}}_h(u_h^{(n)}, \psi_P) = (f, \psi_P), \quad \forall P \in \mathcal{P}_h. \tag{5.1}$$

5.1 A Nonlinear Discrete System for the Solver

We examine the algebraic aspects of the nonlinear system resulted from (5.1).

Firstly, consider the stiffness matrix for the diffusion and convection terms combined. Let $K_{P_i}^*$ and $K_{P_j}^*$ be two adjacent dual elements sharing a common boundary e. Similar to (4.27) and (4.28), the integral terms $\widetilde{\mathcal{I}}_{P_i,e}$ and $\widetilde{\mathcal{I}}_{P_j,e}$ are represented in a nonlinear algebraic form

$$\begin{bmatrix} \widetilde{\mathcal{I}}_{P_i,e} \\ \widetilde{\mathcal{I}}_{P_j,e} \end{bmatrix} = \mathbf{K}_e(\boldsymbol{u}_h) \begin{bmatrix} u_{P_i} \\ u_{P_j} \end{bmatrix} - \mathbf{g}_e(\boldsymbol{u}_h), \tag{5.2}$$

where $\boldsymbol{u}_h = [u_{P_1}, u_{P_2}, \ldots, u_{P_{N_P}}]^\top$ with N_P being the number of nodes in the mesh,

$$\mathbf{K}_e(\boldsymbol{u}_h) = \begin{bmatrix} \beta_e + \frac{BR_e^+}{Bu_{P_i}+h^2} & -\rho_e - \frac{BR_e^-}{Bu_{P_j}+h^2} \\ -\beta_e - \frac{BR_e^+}{Bu_{P_i}+h^2} & \rho_e + \frac{BR_e^-}{Bu_{P_j}+h^2} \end{bmatrix}, \qquad \mathbf{g}_e(\boldsymbol{u}_h) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{5.3}$$

and β_e and ρ_e are positive constants. If one of the nodes P_i and P_j is on the boundary $\partial\Omega$, the scheme changes slightly. Assume that P_i is an interior node but $P_j \in \partial\Omega$. Then $\widetilde{\mathcal{I}}_{P_j,e} = 0$.

By (4.29), we have

$$\mathbf{K}_e(\boldsymbol{u}_h) = \begin{bmatrix} \beta_e + \frac{BR_e^+}{Bu_{P_i}+h^2} & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{g}_e(\boldsymbol{u}_h) = \begin{bmatrix} \rho_e u_{P_j} + R_e^- \\ 0 \end{bmatrix}. \tag{5.4}$$

Denote by $\mathbf{T}_e$ an $N_P \times 2$ matrix whose entries are 1 at positions $(i, 1)$ and $(j, 2)$. The assembly from element stiffness matrices into the global stiffness matrix is expressed as

$$\mathbf{K}(\boldsymbol{u}_h) = \sum_e \mathbf{T}_e \mathbf{K}_e(\boldsymbol{u}_h)\mathbf{T}_e^\top, \qquad \mathbf{g}(\boldsymbol{u}_h) = \sum_e \mathbf{T}_e \mathbf{g}_e(\boldsymbol{u}_h). \tag{5.5}$$

For the time-fractional derivative, we apply ***the lump-of-mass technique*** to obtain

$$\left(\mathrm{D}_F^\alpha u_h^{(n)}, \psi_P\right) = |K_P^*|\, \mathrm{D}_F^\alpha u_P^{(n)}, \tag{5.6}$$

where $|K_P^*|$ is the area of K_P^*. This implies that the mass matrix $\mathbf{M}$ is a diagonal matrix whose entries are just the areas of the dual elements.

Accordingly, the MFL1-correction time-marching solver is formulated as

– For $n = 1, 2$, the temporal discretization is handled by the direct L1 algorithm as

$$\begin{aligned} &\left(\frac{d_{n,1}}{\Gamma(2-\alpha)}\mathbf{M} + \mathbf{K}(\boldsymbol{u}_h^{(n)})\right)\boldsymbol{u}_h^{(n)} = \mathbf{f}^{(n)} + \mathbf{g}(\boldsymbol{u}_h^{(n)}) \\ &\quad + \mathbf{M}\left(\frac{d_{n,n}}{\Gamma(2-\alpha)}\boldsymbol{u}_h^{(0)} + \sum_{k=1}^{n-1}\frac{d_{n,k} - d_{n,k+1}}{\Gamma(2-\alpha)}\boldsymbol{u}_h^{(n-k)}\right). \end{aligned} \tag{5.7}$$

– For $n = 3, 4, \cdots, N_T$,

$$\begin{aligned} &\left(\frac{d_{n,1}}{\Gamma(2-\alpha)}\mathbf{M} + \mathbf{K}(\boldsymbol{u}_h^{(n)})\right)\boldsymbol{u}_h^{(n)} = \mathbf{f}^{(n)} + \mathbf{g}(\boldsymbol{u}_h^{(n)}) + \frac{d_{n,1} - d_{n,2}}{\Gamma(2-\alpha)}\mathbf{M}\boldsymbol{u}_h^{(n-1)} \\ &\quad + \mathbf{M}\left(\frac{T^{-\alpha}}{\Gamma(1-\alpha)}\sum_{j=1}^{N_{exp}}\theta_j \frac{e^{-\lambda_j(\tau_n/T)} - e^{-\lambda_j(\tau_{n,n-2}/T)}}{\lambda_j\tau_{n-1}/T}\boldsymbol{u}_h^{(n-2)} - \frac{T^{-\alpha}}{\Gamma(1-\alpha)}\sum_{j=1}^{N_{exp}}\theta_j \boldsymbol{w}_j^{(n)}\right), \end{aligned} \tag{5.8}$$

where the auxiliary quantity $\boldsymbol{w}_j^{(n)}$ satisfies a recurrence formula

$$\begin{cases} \boldsymbol{w}_j^{(n)} = e^{-\lambda_j(\tau_n/T)}\boldsymbol{w}_j^{(n-1)} + \dfrac{e^{-\lambda_j(\tau_{n,n-2}/T)} - e^{-\lambda_j(\tau_{n,n-3}/T)}}{\lambda_j\tau_{n-2}/T}\left(\boldsymbol{u}_h^{(n-2)} - \boldsymbol{u}_h^{(n-3)}\right), \\ \boldsymbol{w}_j^{(2)} = \mathbf{0}, \qquad \forall 1 \le j \le N_{exp}. \end{cases} \tag{5.9}$$

For both cases, $\mathbf{f}^{(n)}$ is the contribution from the source term.

5.2 Implementation Based on Picard Iterations

Algorithm 1 Picard iterations for the **MFL1-Correction** solver

Choose a small positive value ϵ, a large parameter B, N_T as # of time-marching steps
Determine a error tolerance of SOE approximation ε, parameters $N_{exp}, \lambda_j, \theta_j$
Let $\boldsymbol{u}_h^{(0)} = \big(g_2(P_1), g_2(P_2), \ldots, g_2(P_{N_P})\big)^\top \geq \mathbf{0}$
for $n = 1, 2$ **do**
  $[\boldsymbol{u}_h^{(n)}]^0 = \boldsymbol{u}_h^{(n-1)}$;
  for $p = 0, 1, \cdots$ **do**
    $\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\,[\boldsymbol{u}_h^{(n)}]^{p+1} = \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_1(\boldsymbol{u}_h^{(0)}, \ldots, \boldsymbol{u}_h^{(n-1)})$;
    Solve the linear system
    if $\|[\boldsymbol{u}_h^{(n)}]^{p+1} - [\boldsymbol{u}_h^{(n)}]^p\|_{\max} < \epsilon$ **then**
      $\boldsymbol{u}_h^{(n)} = [\boldsymbol{u}_h^{(n)}]^{p+1}$;
      Stop.
    end if
  end for
end for
for $n = 3, 4, \cdots, N_T$ **do**
  $[\boldsymbol{u}_h^{(n)}]^0 = \boldsymbol{u}_h^{(n-1)}$;
  Compute $\boldsymbol{w}_j^{(n)}$, $1 \leq j \leq N_{exp}$ by the recurrence formula (5.9)
  for $p = 0, 1, \cdots$ **do**
    $\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\,[\boldsymbol{u}_h^{(n)}]^{p+1} = \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_2(\boldsymbol{w}_1^{(n)}, \ldots, \boldsymbol{w}_{N_{exp}}^{(n)}, \boldsymbol{u}_h^{(n-2)}, \boldsymbol{u}_h^{(n-1)})$;
    Solve the linear system
    if $\|[\boldsymbol{u}_h^{(n)}]^{p+1} - [\boldsymbol{u}_h^{(n)}]^p\|_{\max} < \epsilon$ **then**
      $\boldsymbol{u}_h^{(n)} = [\boldsymbol{u}_h^{(n)}]^{p+1}$;
      Stop.
    end if
  end for
end for

Picard iterations can be used to solve the nonlinear systems (5.7) and (5.8), that is, for an integer $p \geq 0$,

$$\begin{aligned}\left(\frac{d_{n,1}}{\Gamma(2-\alpha)}\mathbf{M} + \mathbf{K}([\boldsymbol{u}_h^{(n)}]^p)\right)[\boldsymbol{u}_h^{(n)}]^{p+1} &= \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) \\ + \mathbf{M}\left(\frac{d_{n,n}}{\Gamma(2-\alpha)}\boldsymbol{u}_h^{(0)} + \sum_{k=1}^{n-1}\frac{d_{n,k} - d_{n,k+1}}{\Gamma(2-\alpha)}\boldsymbol{u}_h^{(n-k)}\right),\end{aligned} \tag{5.10}$$

and similarly,

$$\begin{aligned}&\left(\frac{d_{n,1}}{\Gamma(2-\alpha)}\mathbf{M} + \mathbf{K}([\boldsymbol{u}_h^{(n)}]^p)\right)[\boldsymbol{u}_h^{(n)}]^{p+1} = \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \frac{d_{n,1} - d_{n,2}}{\Gamma(2-\alpha)}\mathbf{M}\boldsymbol{u}_h^{(n-1)} \\ &+ \mathbf{M}\left(\frac{T^{-\alpha}}{\Gamma(1-\alpha)}\sum_{j=1}^{N_{exp}}\theta_j\frac{e^{-\lambda_j(\tau_n/T)} - e^{-\lambda_j(\tau_{n,n-2}/T)}}{\lambda_j\tau_{n-1}/T}\boldsymbol{u}_h^{(n-2)} - \frac{T^{-\alpha}}{\Gamma(1-\alpha)}\sum_{j=1}^{N_{exp}}\theta_j\boldsymbol{w}_j^{(n)}\right),\end{aligned} \tag{5.11}$$

where $[\boldsymbol{u}_h^{(n)}]^p$ is the approximate solution at the p-th iteration. We set

$$\mathbf{S}(\boldsymbol{u}_h^{(n)}) = \frac{d_{n,1}}{\Gamma(2-\alpha)}\mathbf{M} + \mathbf{K}(\boldsymbol{u}_h^{(n)}), \tag{5.12}$$

along with

$$\mathbf{w}_1(\boldsymbol{u}_h^{(0)},\ldots,\boldsymbol{u}_h^{(n-1)}) = \mathbf{M}\left(\frac{d_{n,n}}{\Gamma(2-\alpha)}\boldsymbol{u}_h^{(0)} + \sum_{k=1}^{n-1}\frac{d_{n,k}-d_{n,k+1}}{\Gamma(2-\alpha)}\boldsymbol{u}_h^{(n-k)}\right), \tag{5.13}$$

and

$$\begin{aligned}\mathbf{w}_2(\boldsymbol{w}_1^{(n)},\ldots,\boldsymbol{w}_{N_{exp}}^{(n)},\boldsymbol{u}_h^{(n-2)},\boldsymbol{u}_h^{(n-1)}) &= \frac{d_{n,1}-d_{n,2}}{\Gamma(2-\alpha)}\mathbf{M}\boldsymbol{u}_h^{(n-1)}\\ &+\mathbf{M}\left(\frac{T^{-\alpha}}{\Gamma(1-\alpha)}\sum_{j=1}^{N_{exp}}\theta_j\frac{e^{-\lambda_j(\tau_n/T)}-e^{-\lambda_j(\tau_{n,n-2}/T)}}{\lambda_j\tau_{n-1}/T}\boldsymbol{u}_h^{(n-2)} - \frac{T^{-\alpha}}{\Gamma(1-\alpha)}\sum_{j=1}^{N_{exp}}\theta_j\boldsymbol{w}_j^{(n)}\right).\end{aligned} \tag{5.14}$$

Note that (5.10) and (5.11) can be simplified as

$$\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\,[\boldsymbol{u}_h^{(n)}]^{p+1} = \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_1(\boldsymbol{u}_h^{(0)},\ldots,\boldsymbol{u}_h^{(n-1)}) \tag{5.15}$$

and

$$\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\,[\boldsymbol{u}_h^{(n)}]^{p+1} = \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_2(\boldsymbol{w}_1^{(n)},\ldots,\boldsymbol{w}_{N_{exp}}^{(n)},\boldsymbol{u}_h^{(n-2)},\boldsymbol{u}_h^{(n-1)}), \tag{5.16}$$

respectively.

6 Advantages of the MFL1-Correction Solver

This section elaborates on the positivity-preserving property and computational efficiency of our new solver that combines the modified fast L1 evaluation algorithm and flux correction.

6.1 Positivity-Preserving Property of the MFL1-Correction Solver

For the upwinding bilinear finite volume scheme on a general quadrilateral mesh, some off-diagonal entries of the coefficient matrix may be positive. The scheme does not guarantee positivity of the numerical solution to problem (1.1). However, our flux correction technique converts the coefficient matrix to an M-matrix. As is well known, the inverse of an M-matrix has the non-negativity property, which guarantees non-negativity of the numerical solution produced by Algorithm 1.

Theorem 2 *The solution by Algorithm 1 (**MFL1-Correction** solver) is nonnegative.*

Proof We apply mathematical induction on the time-marching step n. The claim is true for $n = 0$, since

$$\boldsymbol{u}_h^{(0)} = \big(g_2(P_1), g_2(P_2),\ldots,g_2(P_{N_P})\big)^\top \geq \mathbf{0}. \tag{6.1}$$

Fix $n \in \{1, 2\}$. Assume that $\boldsymbol{u}_h^{(k)}$ is nonnegative for $k = 0,\ldots,n-1$. By (2.7), we have

$$\mathbf{w}_1(\boldsymbol{u}_h^{(0)},\ldots,\boldsymbol{u}_h^{(n-1)}) \geq \mathbf{0}. \tag{6.2}$$

According to Algorithm 1, the iterative approximation $[\boldsymbol{u}_h^{(n)}]^0 = \boldsymbol{u}_h^{(n-1)}$ is nonnegative. Note that

$$\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\,[\boldsymbol{u}_h^{(n)}]^{p+1} = \mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_1(\boldsymbol{u}_h^{(0)}, \ldots, \boldsymbol{u}_h^{(n-1)}). \tag{6.3}$$

From (5.3),(5.4), we know $\mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) \geq \mathbf{0}$. Note matrix $\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)$ satisfies the following conditions

(i) All diagonal entries are positive;
(ii) All off-diagonal entries are non-positive;
(iii) The column sum is positive.

This implies that $\mathbf{S}^\top$ is an M-matrix and hence $\mathbf{S}^{-1} = ((\mathbf{S}^\top)^{-1})^\top$ is a nonnegative matrix. For the model problem (1.1) with $f \geq 0$, it is clear that $\mathbf{f}^{(n)} \geq \mathbf{0}$. By the induction hypothesis,

$$[\boldsymbol{u}_h^{(n)}]^{p+1} = \left(\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\right)^{-1} \left(\mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_1(\boldsymbol{u}_h^{(0)}, \ldots, \boldsymbol{u}_h^{(n-1)})\right) \geq \mathbf{0}, \quad \text{for } n \leq 2. \tag{6.4}$$

Thus $\boldsymbol{u}_h^{(n)} \geq \mathbf{0}$ for $n \in \{0, 1, 2\}$.

Similarly, fix $n \in \{3, 4, \ldots, N_T\}$. Assume that $\boldsymbol{u}_h^{(k)}$ is nonnegative for $k = 0, \ldots, n-1$. According to Theorem 1, we have

$$\frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j \frac{e^{-\lambda_j(\tau_n/T)} - e^{-\lambda_j(\tau_{n,n-2}/T)}}{\lambda_j \tau_{n-1}/T}\, \boldsymbol{u}_h^{(n-2)} - \frac{T^{-\alpha}}{\Gamma(1-\alpha)} \sum_{j=1}^{N_{exp}} \theta_j \boldsymbol{w}_j^{(n)} \geq 0. \tag{6.5}$$

Then

$$\mathbf{w}_2(\boldsymbol{w}_1^{(n)}, \ldots, \boldsymbol{w}_{N_{exp}}^{(n)}, \boldsymbol{u}_h^{(n-2)}, \boldsymbol{u}_h^{(n-1)}) \geq \mathbf{0}. \tag{6.6}$$

It is straightforward to prove that

$$[\boldsymbol{u}_h^{(n)}]^{p+1} = \left(\mathbf{S}([\boldsymbol{u}_h^{(n)}]^p)\right)^{-1} \left(\mathbf{f}^{(n)} + \mathbf{g}([\boldsymbol{u}_h^{(n)}]^p) + \mathbf{w}_2(\boldsymbol{w}_1^{(n)}, \ldots, \boldsymbol{w}_{N_{exp}}^{(n)}, \boldsymbol{u}_h^{(n-2)}, \boldsymbol{u}_h^{(n-1)})\right) \geq \mathbf{0}, \tag{6.7}$$

where $n \in \{3, 4, \ldots, N_T\}$. By mathematical induction, this implies that the numerical solution produced by Algorithm 1 is indeed nonnegative. □

6.2 Reduction in Bandwidth and Computational Complexity

A noticeable benefit of flux-correction is the reduction of stencil size and then bandwidth of the coefficient matrix of the discrete algebraic system, and accordingly savings in computational costs for each Picard iteration.

Recall discretization of convection utilize information from the upstream elements/nodes. For instance, in the case $\boldsymbol{b} \geq \mathbf{0}$, the stencil size of the original scheme on the dual element K_P^* (shown in Fig. 4a) is 15. It is clear from a comparison of (4.5), (4.12), and (4.27) that our flux correction technique can reduce the stencil size to 5, as shown in Fig. 4b.

Assume a primal mesh has N_P nodes. The MFL1-Correction solver requires $\mathcal{O}(N_P)$ operations for $w_j^{(n)}$ for each fixed n (one step in time-marching) and each fixed j (one term in the

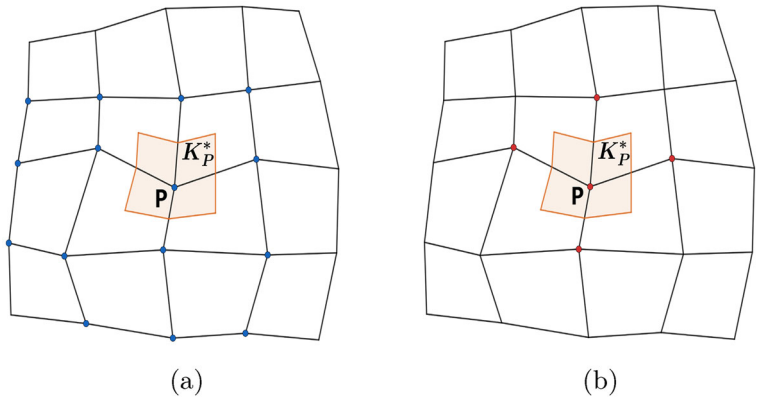


Fig. 4 Reduction in stencil size and hence bandwidth of the coefficient matrix thanks to positivity correction. **a** Before correction: 15 nodes with nonzero coefficients; **b** After correction: Only 5 nodes with nonzero coefficients

SOE approximation) in the recurrence formula. Since $N_{exp} \leq \mathcal{O}((\log N_T)^2)$ (see [5]), this solver requires only

$$\mathcal{O}(N_P N_T (\log N_T)^2), \qquad \mathcal{O}(N_P (\log N_T)^2) \tag{6.8}$$

for operations and storage, respectively.

On the other hand, the L1 discretization does satisfy the two properties in Theorem 1, it can also be combined with our flux correction technique. Such a combination is more expensive, since its requirements for operations and storage are respectively

$$\mathcal{O}(N_P N_T^2), \qquad \mathcal{O}(N_P N_T).$$

7 Numerical Experiments

This section presents numerical examples to demonstrate accuracy, efficiency, and positivity-preserving property of our fast solver. General quadrilateral meshes are used. They are generated from random perturbations of uniform rectangular meshes. For an interior node (x, y) in a uniform rectangular mesh with size h, the corresponding node of the quadrilateral mesh is

$$(\overline{x}, \overline{y}) = (x, y) + \left(-\frac{h}{4} + \frac{h}{2} * \texttt{rand}(1, 2)\right), \tag{7.1}$$

where $\texttt{rand}(1, 2)$ generates a matrix of size 1×2 with entries being random numbers in $(0, 1)$. The distortion range of node coordinates is $(-h/4, h/4)$. Such meshes (see Fig. 5) are used in Example 1 & 3. For all numerical tests, N_E denotes the number of elements and N_T denotes the number of time steps.

Example 1 **(Convergence rates)**. We consider a time-fractional convection-diffusion problem with $\Omega = (0, 1)^2$, $T = 1$, $\alpha = 0.6$, $\boldsymbol{b}(x, y) = [2, 1]^\top$, $A = 10^{-8}$, and a known exact solution

$$u(x, y, t) = (1 - e^{2x+y-\frac{3}{A}} + e^{x+y})\,(t^3 + t^\alpha), \tag{7.2}$$

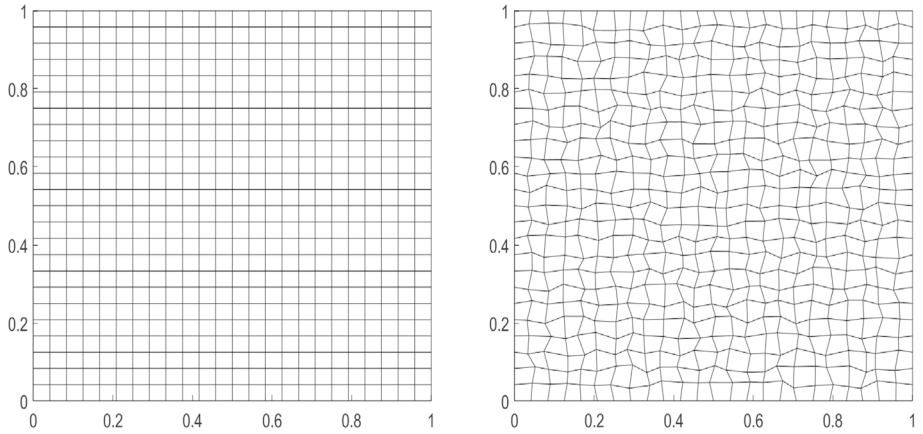


Fig. 5 A quadrilateral mesh obtained from perturbation of a rectangular mesh

which is weakly singular at $t = 0$. More specifically, there exists a constant $C > 0$ so that

$$|\partial_t u(x, y, t)| \leq C(1 + t^{\alpha-1}). \tag{7.3}$$

Such singularity may result in accumulation of errors as time-marching progresses. To address this issue, temporal graded meshes should be used. For instance, one may set $t_n = T(n/N_T)^r$ for $n = 0, 1, \ldots, N_T$. The optimal index is $r = \frac{2-\alpha}{\alpha}$, as suggested in [36], while $r = 1$ gives a uniform temporal partition. We examine two types of errors.

- The L^2-norm of the spatial errors at the final time T: $\|u(\cdot, T) - u_h^{(N_T)}(\cdot)\|_{L^2}$;
- The overall spatial-temporal errors: $\max\limits_{0\leq n\leq N_T} ||u(\cdot, t_n) - u_h^{(n)}(\cdot)||_{L^2}$.

We use quadrilateral meshes obtained from perturbation of uniform rectangular meshes. Parameter $B = 10^{10}$ is used for correction and $\epsilon = 10^{-6}$ for controlling Picard iterations.

Remark 3 Parameter B is problem-dependent, somewhat like the penalty factor for the discontinuous Galerkin finite element methods. It needs to be large enough to guarantee convergence of the numerical solutions to that of the given problem. Here we make a simple empirical choice $B = 10^{10}$ for Example 1. Parameter ϵ is also empirical. Here we expect various types of errors to reach the level of 10^{-3} and accordingly choose $\epsilon = 10^{-6}$, which is three-magnitude lower.

Our numerical solver has errors $\mathcal{O}((\Delta t)^{2-\alpha} + h^2)$, when quasi-uniform spatial meshes (with size h) and graded temporal partitions (with understanding $\Delta t = \frac{1}{N_T}$) are used. But a rigorous proof is omitted due to page limitation. With the consideration to emphasize $(\Delta t)^{2-\alpha}$, we choose $h \approx \Delta t$ (so that h^2 is a higher order infinitesimal) or equivalently $N_E \approx N_T^2$ in Tables 1 & 3. With the consideration to emphasize h^2, we choose $h^2 \approx \Delta t$ or equivalently $N_E \approx N_T$ in Table 4. Listed below are observations for the individual tables.

(i) When the optimal graded temporal mesh ($r = \frac{2-\alpha}{\alpha}$) is used and $h \approx \frac{1}{N_T}$, the overall errors are proportional to $(\Delta t)^{2-\alpha}$, as shown in Table 1;
(ii) If a uniform temporal mesh is used instead and $h \approx \frac{1}{N_T}$, the overall errors converge at a lower rate (close to α), as shown in Table 2;
(iii) When the optimal graded temporal mesh is used and $h \approx \frac{1}{N_T}$, the spatial L^2-errors at the final time T exhibit a convergence rate close to $(2 - \alpha)$, as shown in Table 3;

Table 1 Ex.1 ($\alpha = 0.6$): Overall errors for graded temporal meshes with $r = \frac{2-\alpha}{\alpha}$

		CFL1-Uncorrected solver		MFL1-Correction solver	
N_E	N_T	$\max_n \|\|u_h^{(n)} - u(t_n)\|\|_{L^2}$	Rate	$\max_n \|\|u_h^{(n)} - u(t_n)\|\|_{L^2}$	Rate
8×8	8	1.4558×10^{-1}	–	1.4514×10^{-1}	–
16×16	16	6.3456×10^{-2}	1.198	6.3683×10^{-2}	1.188
32×32	32	2.5641×10^{-2}	1.307	2.5805×10^{-2}	1.303
64×64	64	1.0117×10^{-2}	1.341	1.0118×10^{-2}	1.350
128×128	128	3.9192×10^{-3}	1.368	3.9251×10^{-3}	1.366

Table 2 Ex.1 ($\alpha = 0.6$): Overall errors for uniform temporal partitions ($r = 1$)

		CFL1-Uncorrected solver		MFL1-Correction solver	
N_E	N_T	$\max_n \|\|u_h^{(n)} - u(t_n)\|\|_{L^2}$	Rate	$\max_n \|\|u_h^{(n)} - u(t_n)\|\|_{L^2}$	Rate
8×8	8	9.7168×10^{-2}	–	9.8382×10^{-2}	–
16×16	16	9.1061×10^{-2}	0.093	9.1020×10^{-2}	0.112
32×32	32	7.3654×10^{-2}	0.306	7.3571×10^{-2}	0.307
64×64	64	5.5226×10^{-2}	0.415	5.5227×10^{-2}	0.413
128×128	128	3.9651×10^{-2}	0.478	3.9659×10^{-2}	0.477

Table 3 Ex.1 ($\alpha = 0.6$): Spatial errors of numerical solutions at $t = 1$ for graded temporal meshes with $r = \frac{2-\alpha}{\alpha}$

		CFL1-Uncorrected solver		MFL1-Correction solver	
N_E	N_T	L^2-error	Rate	L^2-error	Rate
8×8	8	1.4558×10^{-1}	–	1.4514×10^{-1}	–
16×16	16	6.3456×10^{-2}	1.198	6.3683×10^{-2}	1.188
32×32	32	2.5641×10^{-2}	1.307	2.5805×10^{-2}	1.303
64×64	64	1.0117×10^{-2}	1.341	1.0118×10^{-2}	1.350
128×128	128	3.9192×10^{-3}	1.368	3.9251×10^{-3}	1.366

(iv) When the optimal graded temporal mesh is used and $h^2 \approx \frac{1}{N_T}$, the spatial L^2-errors at the final time T behave like h^2, as shown in Table 4.

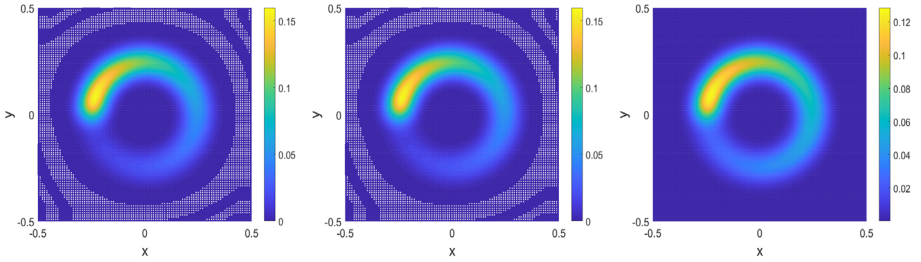
Remark 4 . It is also interesting to note that Tables 1 and 3 record the same results. This is mainly due to the weak singularity of the solution at $t = 0$. When the optimal graded temporal partition ($r = \frac{2-\alpha}{\alpha}$) is used, the following holds

$$\max_{0 \le n \le N_T} \|u(\cdot, t_n) - u_h^{(n)}\|_{L^2} = \|u(\cdot, t_{N_T}) - u_h^{(N_T)}\|_{L^2}.$$

Example 2 **(Positivity-preserving property)**. Here we consider a time-fractional convection-diffusion problem in $\Omega = (-\frac{1}{2}, \frac{1}{2})^2$ with $T = \pi$, $\mathbf{b} = [2y, -2x]^\top$, $A = 10^{-4}$, and $f = 0$.

Table 4 Ex.1 ($\alpha = 0.6$): Spatial errors of numerical solutions at $t = 1$ for graded temporal meshes with $r = \frac{2-\alpha}{\alpha}$

		CFL1-Uncorrected solver		MFL1-Correction solver	
N_E	N_T	L^2-error	Rate	L^2-error	Rate
4×4	4^2	1.5836×10^{-1}	–	1.5700×10^{-1}	–
8×8	8^2	4.2648×10^{-2}	1.892	4.4818×10^{-2}	1.808
16×16	16^2	1.0322×10^{-2}	2.046	1.0296×10^{-2}	2.121
32×32	32^2	2.5676×10^{-3}	2.007	2.5155×10^{-3}	2.033



(a) CFL1-Uncorrected (b) MFL1-Uncorrected (c) MFL1-Correction

Fig. 6 Ex.2 with $\alpha = 0.3$: Numerical solutions at $t = \pi$ by three different solvers. **a**, **b** Solutions exhibit negative values (shown as white dots) for the solvers without correction; **c** the solution remains nonnegative for the solver with correction

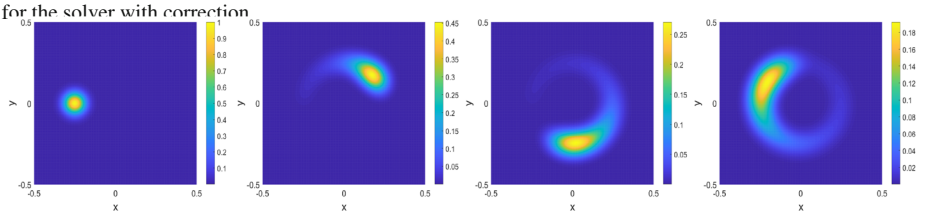


Fig. 7 Ex.2 ($\alpha = 0.95$): Numerical solutions by the MFL1-Correction solver at $t = 0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi$ (from left to right)

A Gaussian hump is specified as the initial condition

$$g_2(x, y) = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right), \quad (x_c, y_c) = (-0.25, 0), \quad \sigma = 0.0447. \tag{7.4}$$

The boundary condition is set as

$$g_1(x, y, t) = \frac{2\sigma^2}{2\sigma^2 + 4At} \exp\left(-\frac{(\cos(2t)x - \sin(2t)y - x_c)^2 + (\sin(2t)x + \cos(2t)y - y_c)^2}{2\sigma^2 + 4At}\right). \tag{7.5}$$

According to the maximum principle [23], for $\alpha \in (0, 1)$, the solution u is nonnegative on $\overline{\Omega} \times (0, T]$.

The problem was solved for $\alpha = 0.3$ by four different solvers on a 96×96 uniform rectangular mesh with a uniform time partition ($N_T = 3000$). Picard iteration control parameter is set as

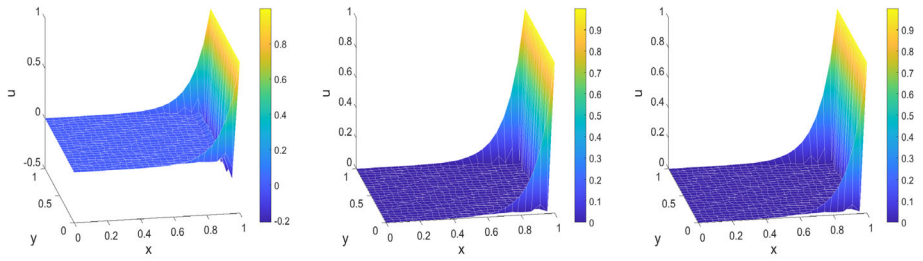


Fig. 8 Ex.3: Numerical solutions at $t = 1$ by three different solvers. **a** Nonphysical oscillations; **b**, **c** No nonphysical oscillations

$\epsilon = 10^{-5}$. Other parameters are the same as in Ex.1. The concentration profiles for the final time $T = \pi$ (see Fig. 6) show clearly negative values (marked as white dots) for the solvers without correction. For CFL1 with correction, it still produced negative solution values as early as $n = 3$ (graphics not presented though). Only the MFL1-Correction solver works and maintains non-negativity of the numerical solution as shown in Fig. 6c. It is clear that both temporal modification and spatial correction are needed.

Example 2 was also solved for $\alpha = 0.95$ by the MFL1-Correction solver. Concentration profiles at time moments $t = 0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi$ are shown in Fig. 7. The numerical solution remains nonnegative and a "long tail" is clearly observed as the counterclockwise rotation progresses.

Example 3 **(Efficiency while preserving positivity)**. We consider a quasi-2d problem with a boundary layer, which is similar to those in [20, 44]. Specifically, $\Omega = (0, 1)^2$, $T = 1$, $\alpha = 0.5$, $A = 1$, and $\mathbf{b} = [x(1 - x) + 400, 0]^\top$. The initial condition is $g_2(x, y) = 0$ and the boundary condition is

$$g_1(x, y, t) = -\frac{1 - e^{10x}}{2.20255 \times 10^4} t^2. \tag{7.6}$$

The problem is solved on a 24×24 quadrilateral mesh with a uniform temporal partition $N_T = 128$. The correction parameter is $B = 10^{10}$ and the control parameter for Picard iterations is $\epsilon = 10^{-6}$.

As shown in Table 5, there could be six solvers. But none of Solver#1,#3,#5 would guarantee non-negativity of numerical solutions, since there is no flux correction. Solver#4 does not preserve positivity either, since it is based on a conventional fast L1 algorithm, namely, a 2-part decomposition of the L1 discretization (see Section 2). Solver#2 does preserve positivity but may be slow. Solver#6 preserves positivity and is faster than Solver#2, as shown in Table 6. Figure 8 provides more details about features of the numerical solutions. Therefore, Solver#6 (**MFL1-Correction**) is the right choice.

8 Concluding Remarks

In this paper, we have developed a novel positivity-preserving fast solver for time-fractional 2-dim convection-diffusion problems. The solver is robust in handling convection dominance. It attains optimal convergence rates when graded temporal meshes are used.

Table 5 Six possible solvers

	Without flux correction	With flux correction
L1 discretization	Solver#1	Solver#2 (L1-correction)
2-part splitting	Solver#3	Solver#4 (CFL1-correction)
3-part splitting	Solver#5	Solver#6 (MFL1-correction)

Table 6 Ex.3: Comparison of #steps of Picard iteration and solver runtime

	L1-correction (solver#2)		MFL1-correction (solver#6)	
N_T	Avg. #steps for Picard iteration	CPU time (s)	Avg. #steps for Picard iteration	CPU time (s)
10,000	2.09	717.87	2.09	444.64
15,000	1.99	1251.32	1.99	632.19
20,000	1.99	1946.26	1.99	841.37

The three-part decomposition of L1 discretization of Caputo derivatives plays an important role in maintaining numerical solutions nonnegative. As discussed in Section 2, the conventional two-part splitting fails to preserve positivity of numerical solutions.

The flux-correction technique discussed in Section 4 leads to a slightly nonlinear problem that involves Picard iterations. As demonstrated in numerical experiments, only few iterations are needed for each time-marching step. But the bandwidth of the stiffness matrix is actually reduced. The numerical solution is guaranteed to be nonnegative and hence the efforts are worthwhile.

Our solver applies to the time-fractional Fokker-Planck equation also. Here we would like to comment on the differences between our work and that in [44]. The work in [44] is concerned with the discrete maximum principle by using the cell-centred finite volume method, but it does not consider fast computation. For our work, the non-negativity of numerical solutions is critical to time-fractional convection-diffusion problems. The finite volume element method was used and our positivity-correction applies to both diffusive and convective fluxes. Moreover, our solver is a fast solver.

The methodology for developing the new solver in this paper can be extended to 3-dim problems and other fractional order PDEs. Combining the higher order temporal discretizations L2 and L2-1_σ with the upwinding and flux-correction techniques in this paper will be interesting topics for further study. These will be reported in our future work.

Acknowledgements Y.Li was partially supported by the National Natural Science Foundation of China (Grant No.12071177). J.Liu was partially supported by US National Science Foundation under Grant DMS-2208590. We sincerely thank the anonymous reviewers, whose comments have helped improve the quality of this paper, and also Prof. Guangwei Yuan, with whom we have meaningful discussion about certain techniques in this paper.

Declarations

Conflict of interests All authors declare no conflict of interests.

Data Availability The data related to this manuscript will be available upon request.

References

1. Alikhanov, A.A.: A new difference scheme for the time fractional diffusion equation. J. Comput. Phys. **280**, 424–438 (2015)
2. Baffet, D.: A Gauss–Jacobi kernel compression scheme for fractional differential equations. J. Sci. Comput. **79**, 227–248 (2019)
3. Baffet, D., Hesthaven, J.S.: High-order accurate adaptive kernel compression time-stepping schemes for fractional differential equations. J. Sci. Comput. **72**, 1169–1195 (2017)
4. Baffet, D., Hesthaven, J.S.: A kernel compression scheme for fractional differential equations. SIAM J. Numer. Anal. **55**, 496–520 (2017)
5. Beylkin, G., Monzón, L.: Approximation by exponential sums revisited. Appl. Comput. Harmon. Anal. **28**(2), 131–149 (2010)
6. Bueno-Orovio, A., Teh, I., Schneider, J.E., Burrage, K., Grau, V.: Anomalous diffusion in cardiac tissue as an index of myocardial microstructure. IEEE Trans. Med. Imaging **35**(9), 2200–2207 (2016)
7. Cao, J., Xiao, A., Bu, W.: Finite difference/finite element method for tempered time fractional advection-dispersion equation with fast evaluation of Caputo derivative. J. Sci. Comput. **83**, 1–29 (2020)
8. Chang, A., Sun, H., Zheng, C., Lu, B., Lu, C., Ma, R., Zhang, Y.: A time fractional convection-diffusion equation to model gas transport through heterogeneous soil and gas reservoirs. Phys. A **502**, 356–369 (2018)
9. D'Elia, M., Du, Q., Glusa, C., Gunzburger, M., Tian, X., Zhou, Z.: Numerical methods for nonlocal and fractional models. Acta Numer. **29**, 1–124 (2020)
10. Diethelm, K., Freed, A.D.: An efficient algorithm for the evaluation of convolution integrals. Comput. Math. Appl. **51**(1), 51–72 (2006)
11. Fallahgoul, H., Focardi, S., Fabozzi, F.: Fractional calculus and fractional processes with applications to financial economics: theory and application. Academic Press, Cambridge (2016)
12. Ford, N.J., Simpson, A.C.: The numerical solution of fractional differential equations: speed versus accuracy. Numer. Algorithms **26**, 333–346 (2001)
13. Gao, G., Sun, Z., Zhang, H.: A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications. J. Comput. Phys. **259**, 33–50 (2014)
14. Gao, Y., Yuan, G., Wang, S., Hang, X.: A finite volume element scheme with a monotonicity correction for anisotropic diffusion problems on general quadrilateral meshes. J. Comput. Phys. **407**, 109143 (2020)
15. Harper, G., Liu, J., Tavener, S., Wildey, T.: Coupling Arbogast–Correa and Bernardi–Raugel elements to resolve coupled Stokes–Darcy flow problems. Comput. Methods Appl. Mech. Eng. **373**, 113469 (2021)
16. Hilfer, R.: Applications of Fractional Calculus in Physics. World Scientific, Singapore (2000)
17. Ionescu, C., Lopes, A., Copot, D., Machado, J., Bates, J.: The role of fractional calculus in modeling biological phenomena: a review. Commun. Nonlinear Sci. Numer. Simul. **51**, 141–159 (2017)
18. Jannelli, A.: Adaptive numerical solutions of time-fractional advection-diffusion-reaction equations. Commun. Nonlinear Sci. Numer. Simul. **105**, 106073 (2022)
19. 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**(3), 650–678 (2017)
20. Jiang, Y., Xu, X.: A monotone finite volume method for time fractional Fokker–Planck equations. Sci. China Math. **62**, 783–794 (2019)
21. Jin, B., Lazarov, R., Thomée, V., Zhou, Z.: On nonnegativity preservation in finite element methods for subdiffusion equations. Math. Comput. **86**, 2239–2260 (2017)
22. Jin, B., Lazarov, R., Zhou, Z.: An analysis of the L1 scheme for the subdiffusion equation with nonsmooth data. IMA J. Numer. Anal. **36**(1), 197–221 (2016)
23. Kopteva, N.: Maximum principle for time-fractional parabolic equations with a reaction coefficient of arbitrary sign. Appl. Math. Lett. **132**, 108209 (2022)
24. Kumar, D., Singh, J.: Fractional Calculus in Medical and Health Science. CRC Press, Boca Raton (2020)
25. Lan, B., Sheng, Z., Yuan, G.: A new positive finite volume scheme for two-dimensional convection-diffusion equation. Z. Angew. Math. Mech. **99**, e201800067 (2019)
26. Li, C., Wang, Z.: Numerical methods for the time-fractional convection-diffusion-reaction equation. Numer. Funct. Anal. Optim. **42**, 1115–1153 (2021)
27. Lin, Y., Xu, C.: Finite difference/spectral approximations for the time-fractional diffusion equation. J. Comput. Phys. **225**(2), 1533–1552 (2007)
28. Lu, C., Huang, W., Qiu, J.: Maximum principle in linear finite element approximations of anisotropic diffusion-convection-reaction problems. Numer. Math. **127**, 515–537 (2014)
29. Lu, C., Huang, W., Vleck, E.S.V.: The cutoff method for the numerical computation of nonnegative solutions of parabolic PDEs with application to anisotropic diffusion and Lubrication-type equations. J. Comput. Phys. **242**, 24–36 (2013)

30. Lv, C., Xu, C.: Error analysis of a high order method for time-fractional diffusion equations. SIAM J. Sci. Comput. **38**(5), A2699–A2724 (2016)
31. Ngondiep, E.: A two-level fourth-order approach for time-fractional convection-diffusion-reaction equation with variable coefficients. Commun. Nonlinear Sci. Numer. Simul. **111**, 106444 (2022)
32. Ngondiep, E.: A high-order numerical scheme for multidimensional convection-diffusion-reaction equation with time-fractional derivative. Numer. Algorithms **91**, 681–700 (2023)
33. Oldham, K.B., Spanier, J.: The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitray Order, Mathematics in Science and Engineering, vol. 111. Academic Press, Cambridge (1974)
34. Roul, P., Rohil, V.: A high-order numerical scheme based on graded mesh and its analysis for the two-dimensional time-fractional convection-diffusion equation. Comput. Math. Appl. **126**, 1–13 (2022)
35. Sahoo, S.K., Gupta, V.: A robust uniformly convergent finite difference scheme for the time-fractional singularly perturbed convection-diffusion problem. Comput. Math. Appl. **137**, 126–146 (2023)
36. Stynes, M., O'Riordan, E., Gracia, J.L.: Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. SIAM J. Numer. Anal. **55**, 1057–1079 (2016)
37. Sun, H., Cao, W.: A fast temporal second-order difference scheme for the time-fractional subdiffusion equation. Numer. Meth. PDEs **37**(3), 1825–1846 (2021)
38. Sun, H., Zhang, Y., Baleanu, D., Chen, W., Chen, Y.: A new collection of real world applications of fractional calculus in science and engineering. Commun. Nonlinear Sci. Numer. Simul. **64**, 213–231 (2018)
39. Tayebi, A., Shekari, Y., Heydari, M.: A meshless method for solving two-dimensional variable-order time fractional advection-diffusion equation. J. Comput. Phys. **340**, 655–669 (2017)
40. West, B.J., Bologna, M., Grigolini, P.: Physics of Fractal Operators. Springer, Berlin (2003)
41. Wu, J., Gao, Z.: Interpolation-based second-order monotone finite volume schemes for anisotropic diffusion equations on general grids. J. Comput. Phys. **275**, 569–588 (2014)
42. Wu, L., Zhai, S.: A new high order ADI numerical difference formula for time-fractional convection-diffusion equation. Appl. Math. Comput. **387**, 124564 (2020)
43. Yan, Y., Sun, Z., Zhang, J.: Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations: a second-order scheme. Commun. Comput. Phys. **22**(4), 1028–1048 (2017)
44. Yang, X., Zhang, H., Zhang, Q., Yuan, G., Sheng, Z.: The finite volume scheme preserving maximum principle for two-dimensional time-fractional Fokker–Planck equations on distorted meshes. Appl. Math. Lett. **97**, 99–106 (2019)
45. Yang, Z., Zeng, F.: A corrected L1 method for a time-tractional subdiffusion equation. J. Sci. Comput. **95**(3), 85 (2023)
46. Yuan, G., Sheng, Z.: Monotone finite volume schemes for diffusion equations on polygonal meshes. J. Comput. Phys. **227**(12), 6288–6312 (2008)
47. Zeng, F., Zhang, Z., Karniadakis, G.E.: Fast difference schemes for solving high-dimensional time-fractional subdiffusion equations. J. Comput. Phys. **307**, 15–33 (2016)
48. Zhai, S., Feng, X., He, Y.: An unconditionally stable compact ADI method for three-dimensional time-fractional convection-diffusion equation. J. Comput. Phys. **269**, 138–155 (2014)
49. Zhang, G., Huang, C., Alikhanov, A.A., Yin, B.: A high-order discrete energy decay and maximum-principle preserving scheme for time fractional Allen–Cahn equation. J. Sci. Comput. **96**(2), 39 (2023)
50. Zhang, J., Zhang, X., Yang, B.: An approximation scheme for the time fractional convection-diffusion equation. Appl. Math. Comput. **335**, 305–312 (2018)
51. Zhu, H., Xu, C.: A fast high order method for the time-fractional diffusion equation. SIAM J. Numer. Anal. **57**, 2829–2849 (2019)
52. Zhuang, P., Gu, Y., Liu, F., Turner, I., Yarlagadda, P.: Time-dependent fractional advection-diffusion equations by an implicit MLS meshless method. Int. J. Numer. Meth. Eng. **88**, 1346–1362 (2011)