Hindawi
Journal of Mathematics
Volume 2024, Article ID 2740720, 14 pages
https://doi.org/10.1155/2024/2740720

Research Article

A High Accuracy Numerical Method Based on Interpolation Technique for Time-Fractional Advection-Diffusion Equations

Yan Chen and Xindong Zhang

School of Mathematical Sciences, Xinjiang Normal University, Urumqi 830017, China

Correspondence should be addressed to Xindong Zhang; liaoyuan1126@163.com

Received 19 September 2023; Revised 18 December 2023; Accepted 31 December 2023; Published 29 January 2024

Academic Editor: Xian-Ming Gu

In this paper, the time-fractional advection-diffusion equation (TFADE) is solved by the barycentric Lagrange interpolation collocation method (BLICM). In order to approximate the fractional derivative under the definition of Caputo, BLICM is used to approximate the unknown function. We obtain the discrete scheme of the equation by combining BLICM with the Gauss-Legendre quadrature rule. The convergence rate for the TFADE equation of the BLICM is derived, and the accuracy of the discrete scheme can be improved by modifying the number of Gaussian nodes. To illustrate the efficiency and accuracy of the present method, a few numerical examples are presented and compared with the other existing methods.

1. Introduction

The fractional partial differential equation (FPDE) has become more widely used in recent decades and has become an important tool in many areas [1–7]. There are numerous numerical schemes for FPDE, for instance, the finite difference method [8–11], finite element method [12, 13], and the spectral method [14, 15], among others. Compared with other numerical methods, BLICM is a high-precision, high-efficiency method of numerical computation, which does not require a dense computational mesh, and its computational program is very easy to write. In recent years, BLICM has been applied in many fields. Some researches on the barycentric interpolation method can be found in [16–21], among others.

The standard advection-diffusion equation describes the changes in a concentration profile as a result of simultaneous diffusion and advection. If we replace the first-order time derivative by the fractional one, then we can obtain the time-fractional advection-diffusion equation. The fractional advection-diffusion equation (FADE) is presented as a useful approach for the description of transport dynamics in complex systems that are governed by anomalous diffusion and nonexponential relaxation patterns [22]. In recent years, several methods have been proposed for solving FADE. Liu et al. [23] considered the space-time fractional advection-diffusion equation by the difference method. Jiang et al. [24] obtained the analytical solutions for the multiterm time-space Caputo–Riesz fractional advection-diffusion equations on a finite domain. Wei et al. [25] studied the time-fractional advection-diffusion equation by the local discontinuous Galerkin method. Tayebi et al. [26] presented a meshless method for solving a two-dimensional variable-order time-fractional advection-diffusion equation. Aghdam et al. [27] developed a numerical method for solving the space-time fractional advection-diffusion equation.

To our knowledge, there are no relevant results using BLICM for solving the TFADE. Based on the aforementioned reasons, the main motivation of this paper is to introduce the BLICM for TFADE. In this paper, we investigate the following TFADE:

$$\begin{cases} {}_0^cD_t^\beta u(x,t) = P\dfrac{\partial^2 u(x,t)}{\partial x^2} - Q\dfrac{\partial u(x,t)}{\partial x} + f(x,t),\ (x,t) \in (0,R)\times(0,T], \\ u(x,0) = \psi(x), \quad x \in [0,R], \\ u(0,t) = \mu(t), \quad t \in [0,T] \\ u(R,t) = \nu(t), \quad t \in [0,T], \end{cases} \tag{1}$$

where $0<\beta<1$, P and Q are the diffusion coefficient and advection coefficient, respectively, $f(x,t)$ is a known function, u is the unknown function, and $\psi(x)$, $\mu(t)$, and $\nu(t)$ are given continuous functions. For $0<\beta<1$, the Caputo fractional derivative is defined as follows:

$${}_0^CD_t^\beta u(x,t) = \frac{1}{\Gamma(1-\beta)}\int_0^t (t-\tau)^{-\beta}\frac{\partial u(x,\tau)}{\partial\tau}\mathrm{d}\tau, \tag{2}$$

which is one of the common fractional derivatives and has been applied in many areas. Properties and more details about Caputo's fractional derivative can be sought out in [28–30].

The remainder of the paper is organised as follows. In Section 2, the basic form of the BLICM is elaborated, and based on this form, an integer-order differential form of BLICM is proposed. Then, a numerical algorithm for the Caputo fractional derivative is proposed in conjunction with the Gauss-Legendre quadrature rule. Finally, the discrete scheme for the TFADE is obtained at the end of this section. The error of the numerical algorithm is theoretically analyzed in Section 3. We provide some numerical examples in Section 4 to demonstrate the effectiveness of this scheme. Lastly, we give the summary in Section 5.

2. A High-Precision Numerical Algorithm for TFADE

2.1. BLICM with the Second Class of Chebyshev Nodes. Barycentric Lagrange interpolation (BLI) is an improvement of Lagrange interpolation. In 2004, Berrut and Trefethen proposed BLI [31].

Let x_i $(i \in O_n, O_n = \{0, 1, \ldots, n\})$ be $n+1$ different interpolation nodes. The value of $u(x)$ at point x_i is denoted by $u_i = u(x_i)$; the interpolation polynomial can be written as the famous Lagrange interpolation polynomial.

$$\begin{aligned} u_L(x) &= \sum_{i=0}^{n}\vartheta_i(x)u_i, \\ \vartheta_i(x) &= \prod_{\theta=0,\theta\neq i}^{n}\frac{x-x_\theta}{x_i-x_\theta}, \quad i \in O_n, \end{aligned} \tag{3}$$

where

$$\vartheta_i(x_a) = \delta_{ia} = \begin{cases} 1, & i=a, \\ 0, & i\neq a. \end{cases} \tag{4}$$

Let $l(x) = \prod_{i=0}^{n}(x-x_i)$ and the barycentric weight be defined as $\omega_i = \prod_{i\neq\theta}1/(x_i-x_\theta)$. This implies $\omega_i = 1/l'(x_i)$, and $\vartheta_i(x)$ can be writen as follows:

$$\vartheta_i(x) = l(x)\frac{\omega_i}{x-x_i}. \tag{5}$$

By equations (3) and (5), then we can obtain:

$$u_L(x) = \sum_{i=0}^{n}\vartheta_i(x)u_i = l(x)\sum_{i=0}^{n}\frac{\omega_i}{x-x_i}u_i. \tag{6}$$

For a fixed point x, we can deduce that

$$1 = \sum_{i=0}^{n}\vartheta_i(x) = l(x)\sum_{i=0}^{n}\frac{\omega_i}{x-x_i}. \tag{7}$$

By equations (6) and (7), we can get the barycentric Lagrange interpolation polynomial (BLIP) denoted by $u_B(x)$ such that

$$u_B(x) = \sum_{i=0}^{n}\zeta_i(x)u_i, \tag{8}$$

where

$$\zeta_i(x) = \frac{\omega_i/(x-x_i)}{\sum_{\theta=0}^{n}\omega_\theta/(x-x_\theta)}. \tag{9}$$

Similarly, for $m+1$ different interpolation nodes t_j $(j \in O_m, O_m = \{0, 1, \ldots, m\})$, we can get the BLIP of $u(t)$ denoted by $u_B(t)$ such that

$$u_B(t) = \sum_{j=0}^{m}\varphi_j(t)u_j, \tag{10}$$

where $u_j = u(t_j)$ and

$$\varphi_j(t) = \frac{\omega_j/(t-t_j)}{\sum_{\rho=0}^{m}\omega_\rho/(t-t_\rho)}, \tag{11}$$

with $\omega_j = \prod_{j\neq r}1/(t_j-t_r)$ $(j \in O_m)$.

BLI has good numerical stability as the nodes distribution density is proportional to the function $(1-x^2)^{-1/2}$. As mentioned in [31], the simplest node distribution that satisfies the above condition is the Chebyshev node family. In this paper, we choose the second class of Chebyshev nodes

$$x_i = \cos\frac{i\pi}{n}, \quad i \in O_n, \tag{12}$$

and the BLI weight of these nodes are as follows:

$$\omega_i = (-1)^i \delta_i,$$
$$\delta_i = \begin{cases} \frac{1}{2}, & i = 0 \text{ or } n, \\ 1, & \text{other values.} \end{cases} \tag{13}$$

The value of $u(x,t)$ at point x_i is $u_i(t) = u(x_i,t)$, for variable x, similar to equation (8), we can get the following BLIP:

$$u_{\mathrm{TB}}(x,t) = \sum_{i=0}^{n} \zeta_i(x) u_i(t), \tag{14}$$

in which the value of the unknown function $u_i(t)$ on nodes t_j is denoted as $u_i(t_j) = u(x_i,t_j) = u_{ij}$. Then, the BLIP of $u_i(t)$ can be represented as follows:

$$u_i(t) = \sum_{j=0}^{m} \varphi_j(t) u_{ij}. \tag{15}$$

Combining equations (14) with (15), the BLIP of $u(x,t)$ at nodes $(x_i,t_j)(i \in O_n, j \in O_m)$ can be obtained as follows:

$$u_{\mathrm{TB}}(x,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} \zeta_i(x)\varphi_j(t) u_{ij}. \tag{16}$$

2.2. Differential Matrix of BLI. By equations (8) and (10), we get

$$\begin{aligned} u_B'(x) &= \sum_{i=0}^{n} \zeta_i'(x) u_i, \\ u_B''(x) &= \sum_{i=0}^{n} \zeta_i''(x) u_i, \\ u_B'(t) &= \sum_{j=0}^{m} \varphi_j'(t) u_j, \\ u_B''(t) &= \sum_{j=0}^{m} \varphi_j''(t) u_j. \end{aligned} \tag{17}$$

By equation (16), we can obtain the following ones:

$$\begin{aligned} \frac{\partial u_{\mathrm{TB}}(x,t)}{\partial x} &= \sum_{i=0}^{n}\sum_{j=0}^{m} \zeta_i'(x)\varphi_j(t) u_{ij}, \\ \frac{\partial^2 u_{\mathrm{TB}}(x,t)}{\partial x^2} &= \sum_{i=0}^{n}\sum_{j=0}^{m} \zeta_i''(x)\varphi_j(t) u_{ij}, \end{aligned} \tag{18}$$

$$\begin{aligned} \frac{\partial u_{\mathrm{TB}}(x,t)}{\partial t} &= \sum_{i=0}^{n}\sum_{j=0}^{m} \zeta_i(x)\varphi_j'(t) u_{ij}, \\ \frac{\partial^2 u_{\mathrm{TB}}(x,t)}{\partial t^2} &= \sum_{i=0}^{n}\sum_{j=0}^{m} \zeta_i(x)\varphi_j''(t) u_{ij}. \end{aligned} \tag{19}$$

It follows by equation (9) that

$$\zeta_i(x)\sum_{i=0}^{n} \frac{\omega_i}{x-x_i} = \frac{\omega_i}{x-x_i}. \tag{20}$$

For above equation, if multiply $x - x_a \quad (a \in O_n)$ on both sides simultaneously, then we can get that

$$\zeta_i(x)\sum_{i=0}^{n} \frac{\omega_i(x-x_a)}{x-x_i} = \frac{\omega_i(x-x_a)}{x-x_i}. \tag{21}$$

Let

$$\sigma_a(x) = \sum_{i=0}^{n} \frac{\omega_i(x-x_a)}{x-x_i}, \tag{22}$$

by equation (21), the following equations can be obtained:

$$\begin{aligned} \zeta_i(x)\sigma_a'(x) + \zeta_i'(x)\sigma_a(x) &= \left(\frac{\omega_i(x-x_a)}{x-x_i}\right)', \\ \zeta_i(x)\sigma_a''(x) + 2\zeta_i'(x)\sigma_a'(x) + \zeta_i''(x)\sigma_a(x) &= \left(\frac{\omega_i(x-x_a)}{x-x_i}\right)''. \end{aligned} \tag{23}$$

Fix a node x_a, we observe that

$$\begin{aligned} \sigma_a(x_a) &= \omega_a, \\ \sigma_a'(x_a) &= \sum_{i\neq a} \frac{\omega_i}{x_a-x_i}, \\ \sigma_a''(x_a) &= -2\sum_{i\neq a} \frac{\omega_i}{(x_a-x_i)^2}. \end{aligned} \tag{24}$$

In order to get the differential matrix of BLI, we present some results below which will be applied in our arguments. We will discuss in two cases.

Case 1. $a \neq i$.

Since $\zeta_i(x_a) = 0 \quad (a \neq i)$, we can achieve

$$\begin{aligned} \zeta_i'(x_a) &= \frac{\omega_i/\omega_a}{x_a - x_i}, \\ \zeta_i''(x_a) &= -2\frac{\omega_i/\omega_a}{x_a - x_i}\left(\sum_{k\neq a}\frac{\omega_k/w_a}{(x_a - x_k)^2} + \frac{1}{x_a - x_i}\right). \end{aligned} \tag{25}$$

Case 2. $a = i$.

It is pointed out that $\zeta_i(x)$ is also barycentic Lagrange interpolation basis function satisfying the property $\sum_{i=0}^{n}\zeta_i(x) = 1$, thus we can get that

$$\sum_{i=0}^{n}\zeta_i^{(\mu)}(x) = 0 \Leftrightarrow \zeta_a^{(\mu)}(x_a) = -\sum_{i\neq a}\zeta_i^{(\mu)}(x_a), \tag{26}$$

where $\zeta_i^{(\mu)}(x)$ denotes the μ-order $(\mu \in \mathbb{N}^+)$ derivative of function $\zeta_i(x)$.

Thus, the BLIP for the μ-order derivative of $u(x)$ on nodes $x_a \quad (a \in O_n)$ can be obtained as follows:

$$\begin{aligned} u_B^{(\mu)}(x_a) &= \sum_{i=0}^{n}\zeta_i^{(\mu)}(x_a)u_i \\ &= \sum_{i=0}^{n}D_{ai}^{(\mu)}u_i. \end{aligned} \tag{27}$$

Similarly, we can obtain the BLIP for the μ-order derivative of $u(t)$ on nodes $t_b \quad (b \in O_m)$, that is

$$\begin{aligned} u_B^{(\mu)}(t_b) &= \sum_{j=0}^{m}\varphi_j^{(\mu)}(t_b)u_j \\ &= \sum_{j=0}^{m}C_{bj}^{(\mu)}u_j. \end{aligned} \tag{28}$$

Finally, the BLIP for the μ-order derivative of $u(x,t)$ on nodes $(x_a, t_b)(a \in O_n, b \in O_m)$ can be obtained in the following form:

$$\begin{aligned} \frac{\partial^{(\mu)}u_{\mathrm{TB}}(x_a, t_b)}{\partial x^{\mu}} &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i^{(\mu)}(x_a)\varphi_j(t_b)u_{ij} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m}D_{ai}^{(\mu)}\varphi_j(t_b)u_{ij}, \end{aligned} \tag{29}$$

$$\begin{aligned} \frac{\partial^{(\mu)}u_{\mathrm{TB}}(x_a, t_b)}{\partial t^{\mu}} &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x_a)\varphi_j^{(\mu)}(t_b)u_{ij} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x_a)C_{bj}^{(\mu)}u_{ij}, \end{aligned} \tag{30}$$

where the μ-order differential matrices of BLI have the following forms, which will be needed in Subsection 2.4.

$$\begin{cases} D_{ai}^{(\mu)} = \mu\left(D_{aa}^{(\mu-1)}D_{ai}^{(1)} - \dfrac{D_{ai}^{(\mu-1)}}{x_a - x_i}\right), & a \neq i, \\ D_{aa}^{(\mu)} = -\displaystyle\sum_{i=0, i\neq a}^{n}D_{ai}^{(\mu)}, \\ C_{bj}^{(\mu)} = \mu\left(C_{bb}^{(\mu-1)}C_{bj}^{(1)} - \dfrac{C_{bj}^{(\mu-1)}}{t_b - t_j}\right), & b \neq j, \\ C_{bb}^{(\mu)} = -\displaystyle\sum_{j=0, j\neq b}^{n}C_{bj}^{(\mu)}. \end{cases} \tag{31}$$

2.3. Calculation Scheme of Caputo Fractional Derivative.

We ponder the numerical scheme of Caputo fractional derivative in this subsection. By equation (2), we can infer that

$${}_{0}^{C}D_t^{\beta}u(x,t) = \frac{1}{\Gamma(2-\beta)}\frac{\partial u(x,0)}{\partial \tau}t^{1-\beta} + \frac{1}{\Gamma(2-\beta)}\int_0^t (t-\tau)^{1-\beta}\frac{\partial^2 u(x,\tau)}{\partial \tau^2}\mathrm{d}\tau. \tag{32}$$

In equation (32), if we replace $\partial u(x,0)/\partial\tau$, $\partial^2 u(x,\tau)/\partial\tau^2$ by the terms of equation (30) and discretize the domain by $n+1\,(0 = x_0 < x_1 < \cdots < x_n = 1)$ nodes in space and $m+1\,(0 = t_0 < t_1 < \cdots < t_m = 1)$ nodes in time, then the following holds:

$${}_{0}^{C}D_t^{\beta}u(x,t) \doteq \frac{t^{1-\beta}}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j'(t_0)u_{ij} + \frac{1}{\Gamma(2-\beta)}\int_0^t (t-\tau)^{1-\beta}\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j''(\tau)u_{ij}\mathrm{d}\tau. \tag{33}$$

By using Gauss-Legendre quadrature rule [32], we can get the following formula:

$$\begin{aligned}\int_0^t (t-\tau)^{1-\beta}\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j''(\tau)u_{ij}\mathrm{d}\tau &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\int_0^t (t-\tau)^{1-\beta}\varphi_j''(\tau)u_{ij}\mathrm{d}\tau \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\sum_{r=1}^{p}\left(t-\tau_r\right)^{1-\beta}\varphi_j''(\tau_r)u_{ij}W_r \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{r=1}^{p}\left(t-\tau_r\right)^{1-\beta}\zeta_i(x)\varphi_j''(\tau_r)u_{ij}W_r,\end{aligned} \tag{34}$$

where τ_r is integral point, W_r and p are integral weight and the number of point for Gauss–Legendre quadrature rule, respectively. Combining above two discrete formats, we can obtain the discrete scheme of Caputo fractional derivative as follows:

$$ {}_0^C D_t^\beta u(x,t) = \frac{1}{\Gamma(2-\beta)}\left(t^{1-\beta}\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j'(t_0)u_{ij} + \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{r=1}^{p}\left(t-\tau_r\right)^{1-\beta}\zeta_i(x)\varphi_j''(\tau_r)u_{ij}W_r\right). \tag{35}$$

2.4. Discrete Scheme of TFADE. In this subsection, the BLICM is used to approximate equation (1). By equations (18), (19), and (35), we present the following equation:

$$\begin{aligned}&\frac{t^{1-\beta}}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j'(t_0)u_{ij} + \frac{1}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{r=1}^{p}\left(t-\tau_r\right)^{1-\beta}\zeta_i(x)\varphi_j''(\tau_r)u_{ij}W_r \\ &\quad = P\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i''(x)\varphi_j(t)u_{ij} - Q\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i'(x)\varphi_j(t)u_{ij} + f(x,t).\end{aligned} \tag{36}$$

Let equation (36) hold at nodes $(x_a, t_b)\,(a \in O_n, b \in O_m)$, then

$$\begin{aligned}&\frac{t_b^{1-\beta}}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x_a)\varphi_j'(t_0)u_{ij} + \frac{1}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{r=1}^{p}\left(t_b-\tau_r\right)^{1-\beta}\zeta_i(x_a)\varphi_j''(\tau_r)u_{ij}W_r \\ &\quad = P\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i''(x_a)\varphi_j(t_b)u_{ij} - Q\sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i'(x_a)\varphi_j(t_b)u_{ij} + f(x_a,t_b).\end{aligned} \tag{37}$$

Let $K_j(t_b) = \sum_{r=1}^{p}(t_b-\tau_r)^{1-\beta}\varphi_j''(\tau_r)W_r$, combining $\zeta_i(x_a) = \delta_{ia}$ and $\varphi_j(t_b) = \delta_{jb}$, then equation (37) can be written as follows:

$$\frac{t_b^{1-\beta}}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}\delta_{ia}C_{0j}^{(1)}u_{ij}+\frac{1}{\Gamma(2-\beta)}\sum_{i=0}^{n}\sum_{j=0}^{m}K_j(t_b)\delta_{ia}u_{ij}$$
$$=P\sum_{i=0}^{n}\sum_{j=0}^{m}D_{ai}^{(2)}\delta_{jb}u_{ij}-Q\sum_{i=0}^{n}\sum_{j=0}^{m}D_{ai}^{(1)}\delta_{jb}u_{ij}+f(x_a,t_b). \tag{38}$$

Writing equation (38) in the matrix form, we get

$$\frac{t_b^{1-\beta}}{\Gamma(2-\beta)}\begin{bmatrix}\sum_{j=0}^{m}\delta_{0a}C_{0j}^{(1)}u_{0j}\\ \vdots\\ \sum_{j=0}^{m}\delta_{na}C_{0j}^{(1)}u_{nj}\end{bmatrix}+\frac{1}{\Gamma(2-\beta)}\begin{bmatrix}\sum_{j=0}^{m}\delta_{0a}K_j(t_b)u_{0j}\\ \vdots\\ \sum_{j=0}^{m}\delta_{na}K_j(t_b)u_{nj}\end{bmatrix}$$
$$=P\begin{bmatrix}\sum_{j=0}^{m}\delta_{jb}D_{a0}^{(2)}u_{0j}\\ \vdots\\ \sum_{j=0}^{m}\delta_{jb}D_{an}^{(2)}u_{nj}\end{bmatrix}-Q\begin{bmatrix}\sum_{j=0}^{m}\delta_{jb}D_{a0}^{(1)}u_{0j}\\ \vdots\\ \sum_{j=0}^{m}\delta_{jb}D_{an}^{(1)}u_{nj}\end{bmatrix}+\begin{bmatrix}f(x_0,t_b)\\ \vdots\\ f(x_n,t_b)\end{bmatrix}. \tag{39}$$

Taking all values of $a\in O_n$ and $b\in O_m$, by equation (39), we can be obtained the following matrix form:

$$\frac{1}{\Gamma(2-\beta)}\left(\mathbf{C}^{(1)}\otimes\mathbf{I}_N\right)\begin{bmatrix}\mathbf{u}_0\\ \vdots\\ \mathbf{u}_m\end{bmatrix}+\frac{1}{\Gamma(2-\beta)}\left(\mathbf{K}\otimes\mathbf{I}_N\right)\begin{bmatrix}\mathbf{u}_0\\ \vdots\\ \mathbf{u}_m\end{bmatrix}$$
$$=P\left(\mathbf{I}_M\otimes\mathbf{D}^{(2)}\right)\begin{bmatrix}\mathbf{u}_0\\ \vdots\\ \mathbf{u}_m\end{bmatrix}-Q\left(\mathbf{I}_M\otimes\mathbf{D}^{(1)}\right)\begin{bmatrix}\mathbf{u}_0\\ \vdots\\ \mathbf{u}_m\end{bmatrix}+\begin{bmatrix}\mathbf{f}_0\\ \vdots\\ \mathbf{f}_m\end{bmatrix}, \tag{40}$$

where

$$\mathbf{C}^{(1)}=\begin{bmatrix}t_0^{1-\beta}C_{00}^{(1)} & \cdots & t_0^{1-\beta}C_{0m}^{(1)}\\ \vdots & \ddots & \vdots\\ t_m^{1-\beta}C_{00}^{(1)} & \cdots & t_m^{1-\beta}C_{0m}^{(1)}\end{bmatrix},$$
$$\mathbf{K}=\begin{bmatrix}t_0K_0(t_0) & \cdots & t_0K_m(t_0)\\ \vdots & \ddots & \vdots\\ t_mK_0(t_m) & \cdots & t_mK_m(t_m)\end{bmatrix},$$
$$\mathbf{D}^{(2)}=\begin{bmatrix}D_{00}^{(2)} & \cdots & D_{0n}^{(2)}\\ \vdots & \ddots & \vdots\\ D_{n0}^{(2)} & \cdots & D_{nn}^{(2)}\end{bmatrix},$$
$$\mathbf{D}^{(1)}=\begin{bmatrix}D_{00}^{(1)} & \cdots & D_{0n}^{(1)}\\ \vdots & \ddots & \vdots\\ D_{n0}^{(1)} & \cdots & D_{nn}^{(1)}\end{bmatrix},$$
$$\mathbf{u}_c=\begin{bmatrix}u_{0c}\\ \vdots\\ u_{nc}\end{bmatrix},$$
$$\mathbf{f}_c=\begin{bmatrix}f(x_0,t_c)\\ \vdots\\ f(x_n,t_c)\end{bmatrix}, \tag{41}$$

$\mathbf{I}_M$ and $\mathbf{I}_N$ are identity matrices of order $m+1$ and $n+1$, respectively, and $c\in O_m$.

Let $\mathbf{U}=[\mathbf{u}_0,\ldots,\mathbf{u}_m]^T$ and $\mathbf{F}=[\mathbf{f}_0,\ldots,\mathbf{f}_m]^T$, then the discrete form of equation (1) can be expressed as follows:

$$\left[\frac{1}{\Gamma(2-\beta)}\left(\left(\mathbf{C}^{(1)}\otimes\mathbf{I}_N\right)+\left(\mathbf{K}\otimes\mathbf{I}_N\right)\right)-P\left(\mathbf{I}_M\otimes\mathbf{D}^{(2)}\right)+Q\left(\mathbf{I}_M\otimes\mathbf{D}^{(1)}\right)\right]\mathbf{U}=\mathbf{F}, \tag{42}$$

and the discrete formats of the initial value conditions are as follows:

$$
\begin{aligned}
u_{\mathrm{TB}}(x,0) &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j(0)u_{ij} = \psi(x),\\
u_{\mathrm{TB}}(0,t) &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(0)\varphi_j(t)u_{ij} = \mu(t),\\
u_{\mathrm{TB}}(R,t) &= \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(R)\varphi_j(t)u_{ij} = \nu(t).
\end{aligned}
\tag{43}
$$

3. Error Analysis

The error estimates of BLICM based on the second class of Chebyshev nodes are presented in this section. We first give the following definitions and lemmas for future applications in our arguments.

Definition 3. Let $\mathscr{J}_1$ be the function space consisting of the interpolated basis functions $\{\zeta_i\}_{i=0}^{n}$ defined by equation (9), and $\mathscr{J}_2$ be the function space consisting of the interpolated basis functions $\{\varphi_j\}_{j=0}^{m}$ defined by equation (11).

Definition 4. For $u(x), u(t) \in C[-1,1]$, define $\mathfrak{B}_{x,n}$: $C[-1,1] \longrightarrow \mathscr{J}_1$ and $\mathfrak{B}_{t,m}$: $C[-1,1] \longrightarrow \mathscr{J}_2$, they are interpolation operators for x and t, and satisfying the following equation:

$$
\begin{aligned}
\mathfrak{B}_{x,n}u(x) &= \sum_{i=0}^{n}\zeta_i(x)u_i,\\
\mathfrak{B}_{t,m}u(t) &= \sum_{j=0}^{m}\varphi_j(t)u_j.
\end{aligned}
\tag{44}
$$

Similarly, let $\mathscr{J} = \mathscr{J}_1 \cup \mathscr{J}_2$, we can define $\mathfrak{B}_{x,n}\mathfrak{B}_{t,m}$: $C([-1,1]\times[-1,1]) \longrightarrow \mathscr{J}$, it satisfies the following equation:

$$
\mathfrak{B}_{x,n}\mathfrak{B}_{t,m}u(x,t) = \sum_{i=0}^{n}\sum_{j=0}^{m}\zeta_i(x)\varphi_j(t)u_{ij}. \tag{45}
$$

It is obvious that $u_{\mathrm{TB}} \coloneqq \mathfrak{B}_{x,n}\mathfrak{B}_{t,m}u$. Let $\alpha_1, \alpha_2 \in \mathbb{R}$. Since $\mathfrak{B}_{x,n}(\alpha_1 u(x) + \alpha_2 v(x)) = \alpha_1\sum_{i=0}^{n}\zeta_i(x)u_i + \alpha_2\sum_{i=0}^{n}\zeta_i(x)v_i = \alpha_1\mathfrak{B}_{x,n}u(x) + \alpha_2\mathfrak{B}_{x,n}v(x)$, by the definition of linear operator, we conclude that $\mathfrak{B}_{x,n}$ is a linear operator. Similarly, $\mathfrak{B}_{t,m}$ and $\mathfrak{B}_{x,n}\mathfrak{B}_{t,m}$ are all linear operators.

Definition 5 (see [33]) (Lebesgue constant). $\boldsymbol{\lambda}_n = \|\mathfrak{B}_{x,n}\|_\infty = \max_{x\in[-1,1]}\sum_{i=0}^{n}|\zeta_i(x)|$.

Lemma 6 (see [34]). *Assuming that interpolation nodes $x_0, x_1, \ldots, x_n$ in the interval $[-1,1]$, $u^{(n+1)}(x) \in C[-1,1]$, then, for all $x \in [-1,1]$, there exists $\xi \in [-1,1]$, ξ related to x, such that*

$$
u - U = \frac{u^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i), \tag{46}
$$

where U is the interpolation polynomial of u.

Lemma 7 (see [35]). *When the BLICM at the second class of Chebyshev nodes, its Lebesgue constant $\boldsymbol{\lambda}_n$ satisfies the following equation:*

$$
\boldsymbol{\lambda}_n \le \frac{2}{\pi}\log(n+1) + 1. \tag{47}
$$

Theorem 8. *Let $u_x^{(n+1)}(x,t)$, $u_t^{(m+1)}(x,t) \in C([0,R]\times[0,T])$, and let $u = u(x_a, t_b)$ be the exact solution of equation (1) and $u_{TB} = u_{TB}(x_a, t_b)$ be the numerical solution of equation (38) at the nodes (x_a, t_b) with $a \in O_n$ and $b \in O_m$, then*

$$
\|u - u_{\mathrm{TB}}\|_\infty \le \left\{\frac{n}{2^{n-1}(n+1)!} + \left(\frac{2}{\pi}\log(n+1) + 1\right)\frac{m}{2^{m-1}(m+1)!}\right\}\left\|u^{(*)}\right\|_\infty, \tag{48}
$$

where $\|u^{(*)}\|_\infty$ is the maximum one of $\|u_x^{(n+1)}\|_\infty$ and $\|u_t^{(m+1)}\|_\infty$.

Proof. Using the triangle inequality, we achieve that

$$
\begin{aligned}
\|u - u_{\mathrm{TB}}\|_\infty &= \|u - \mathfrak{B}_{x,n}\mathfrak{B}_{t,m}u\|_\infty\\
&\le \|u - \mathfrak{B}_{x,n}u\|_\infty + \|\mathfrak{B}_{x,n}u - \mathfrak{B}_{x,n}\mathfrak{B}_{t,m}u\|_\infty.
\end{aligned}
\tag{49}
$$

By Lemma 6, we obtain

$$
\|u - \mathfrak{B}_{x,n}u\|_\infty = \frac{\left\|u_x^{(n+1)}(\xi,t)\right\|_\infty}{(n+1)!}\left|\prod_{i=0}^{n}(x - x_i)\right|. \tag{50}
$$

Applying $|\prod_{i=0}^{n}(x - x_i)| \le n2^{1-n} = n/2^{n-1}$ (see [36]), the above equation enables us to write the following equation:

$$
\|u - \mathfrak{B}_{x,n}u\|_\infty \le \frac{n}{2^{n-1}(n+1)!}\left\|u_x^{(n+1)}(\xi,t)\right\|_\infty. \tag{51}
$$

Similarly, we can get

$$
\|u - \mathfrak{B}_{t,m}u\|_\infty \le \frac{m}{2^{m-1}(m+1)!}\left\|u_t^{(m+1)}(x,\xi)\right\|_\infty. \tag{52}
$$

Since $\mathfrak{B}_{x,n}$ and $\mathfrak{B}_{t,m}$ are linear operators, combining Definition 5 and equation (52), we have

$$
\begin{aligned}
\|\mathfrak{B}_{x,n}u - \mathfrak{B}_{x,n}\mathfrak{B}_{t,m}u\|_\infty &\le \left\|\mathfrak{B}_{x,n}\left(u - \mathfrak{B}_{t,m}u\right)\right\|_\infty\\
&\le \|\mathfrak{B}_{x,n}\|_\infty\|u - \mathfrak{B}_{t,m}u\|_\infty\\
&\le \boldsymbol{\lambda}_n\frac{m}{2^{m-1}(m+1)!}\left\|u_t^{(m+1)}\right\|_\infty.
\end{aligned}
\tag{53}
$$

Combining these with Lemma 7 we deduce that

$$\left\|\mathfrak{B}_{x,n}u - \mathfrak{B}_{x,n}\mathfrak{B}_{t,m}u\right\|_{\infty} \le \left(\frac{2}{\pi}\log(n+1)+1\right)\frac{m}{2^{m-1}(m+1)!}\left\|u_t^{(m+1)}\right\|_{\infty}. \tag{54}$$

Let $\|u^{(*)}\|_{\infty} = \max\left\{\|u_x^{(n+1)}\|_{\infty}, \|u_t^{(m+1)}\|_{\infty}\right\}$, by equations (51) and (54) we conclude that

$$\begin{aligned}\left\|u - u_{\mathrm{TB}}\right\|_{\infty} &\le \frac{n}{2^{n-1}(n+1)!}\left\|u_x^{(n+1)}\right\|_{\infty} + \left(\frac{2}{\pi}\log(n+1)+1\right)\frac{m}{2^{m-1}(m+1)!}\left\|u_t^{(m+1)}\right\|_{\infty}\\ &\le \left\{\frac{n}{2^{n-1}(n+1)!} + \left(\frac{2}{\pi}\log(n+1)+1\right)\frac{m}{2^{m-1}(m+1)!}\right\}\left\|u^{(*)}\right\|_{\infty}.\end{aligned} \tag{55}$$

This completes the proof.

Analysis similar to that in the proof of Theorem 8 in [36], we can get the following error estimate of Caputo fractional derivative. □

Theorem 9. *Let $(x,t) \in [-1,1] \times [-1,1]$, then the following error estimate for the BLICM of Caputo fractional derivative holds:*

$$\left\|{}_0^C D_t^{\beta} u(x,t) - {}_0^C D_t^{\beta} u_{\mathrm{TB}}(x,t)\right\|_{\infty} \le \left\|u^{(*)}\right\|_{\infty}\left\{C_1\left(\frac{\mathrm{e}h_x}{2n}\right)^{n} + C_2\left(\frac{\mathrm{e}h_t}{2(m-\beta)}\right)^{m-\beta}\right\}, \tag{56}$$

where $\|u^{(*)}\|_{\infty}$ is the maximum one of $\|u_x^{(n+1)}\|_{\infty}$ and $\|u_t^{(m+1)}\|_{\infty}$, C_1 and C_2 are constants independent of n and m, and h_x, h_t are represented as the lengths of the interval in two dimensions.

4. Numerical Examples

This section demonstrates the superiority of BLICM in solving TFADE through some examples. All numerical results are implemented on the AMD Ryzen 5 5600H Windows 10 system by using MATLAB R2022b. The space-time discrete scheme equation (42) is a system of linear algebraic equations AU $= F$, which can be solved as $U = A\backslash F$ ("\" is the built-in function in MATLAB). By the way, the space-time discrete linear system equation (42) is very related to the so-called all-at-once linear system, which can be solved by the parallel iterative method [37] in order to improve the computational effectiveness. But that is outside the scope of this paper and we shall not pursue that here.

The absolute error and the relative error in all examples are defined as follows:

$$\begin{aligned}E_a &= \left|u\left(x_i,t_j\right) - u_{\mathrm{TB}}\left(x_i,t_j\right)\right|,\\ E_r &= \frac{\max\left\{\left|u\left(x_i,t_j\right) - u_{\mathrm{TB}}\left(x_i,t_j\right)\right|\right\}}{\max\left\{\left|u\left(x_i,t_j\right)\right|\right\}},\end{aligned} \tag{57}$$

where $u(x_i,t_j)$ and $u_{\mathrm{TB}}(x_i,t_j)$ denote the exact value and numerical value on (x_i,t_j), respectively. The convergence order is defined as $\log(\mathrm{Er1}/\mathrm{Er2})/\log(2)$, where Er2 is the current error and Er1 is the previous error.

Example 1. Let $R = T = 1$, $P = 1$, $Q = 0$, $\mu(t) = 0$, $\nu(t) = 0$, $\psi(x) = \sin(\pi x)$ and the forcing function is $f(x,t) = 2\sin(\pi x)t^{2-\beta}/\Gamma(3-\beta) + \pi^2(t^2+1)\sin(\pi x)$ in equation (1). The exact solution is $u(x,t) = (t^2+1)\sin(\pi x)$.

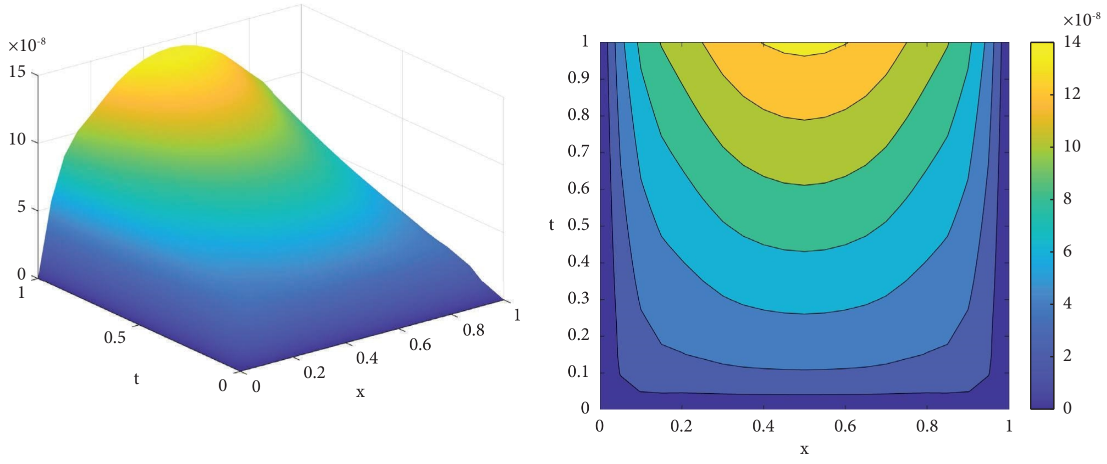
Taking 900 Gaussian nodes and using equation (42) to solve the Example 1. Table 1 shows the relative error and convergence order of Example 1 for $\beta = 0.3$,0.5 and 0.7, respectively. Moreover, the maximum relative error with 11×11 nodes by our method is about 10^{-10}. As $m = n = 11$ and $\beta = 0.3$, Figures 1 and 2 show the absolute error under the different types of nodes. From Figures 1 and 2, we can find that the second class of Chebyshev nodes are generally more accurate than equidistant nodes, which shows that Chebyshev nodes are more suitable for BLICM.

Example 2. Let $R = T = 1$, $P = 1$, $Q = 0$, $\mu(t) = 0$, $\nu(t) = 0$, $\psi(x) = 0$, and the forcing function is $f(x,t) = 2/\Gamma(3-\beta)t^{2-\beta}\sin(2\pi x) + 4\pi^2 t^2\sin(2\pi x)$ in equation (1). The exact solution is $u(x,t) = t^2\sin(2\pi x)$.

By equation (42), the results of Example 2 are obtained in Table 2 and Figure 3. Table 2 shows the relative error and convergence order of Example 2 for $\beta = 0.3$, 0.5 and 0.7 with

Table 1: The results of Example 1 with 900 Gaussian nodes.

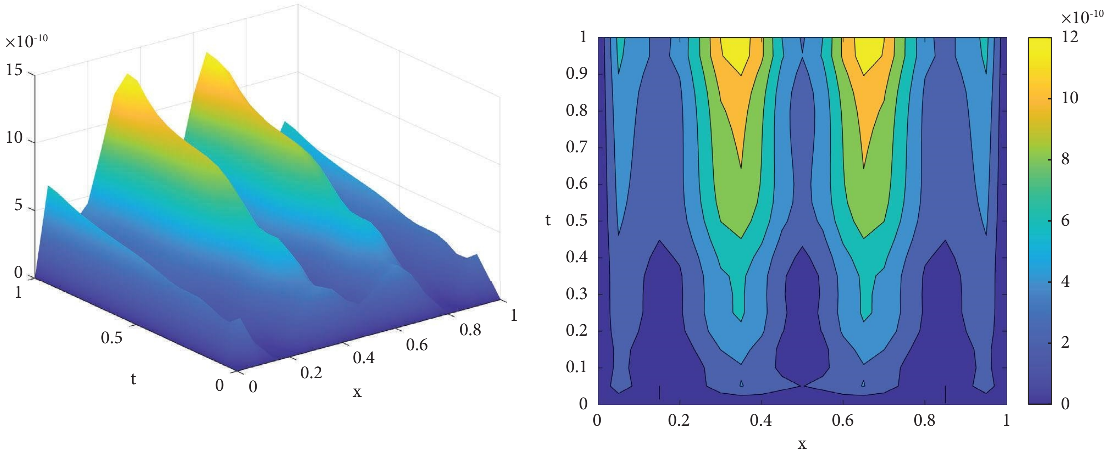
	$n \times m$	E_r	order	CPU times (s)
$\beta = 0.3$	5×5	7.3273×10^{-3}	—	0.0678
	7×7	2.5836×10^{-5}	8.1478	0.0748
	9×9	2.1571×10^{-7}	6.9041	0.1093
	11×11	6.7406×10^{-10}	8.3220	0.1442
$\beta = 0.5$	5×5	7.5890×10^{-3}	—	0.0587
	7×7	2.6879×10^{-5}	8.1413	0.0684
	9×9	2.3353×10^{-7}	6.8467	0.0951
	11×11	4.8861×10^{-10}	8.9007	0.1268
$\beta = 0.7$	5×5	7.9797×10^{-3}	—	0.0511
	7×7	2.8917×10^{-5}	8.1083	0.0907
	9×9	2.7703×10^{-7}	6.7057	0.1012
	11×11	4.9982×10^{-10}	9.1144	0.1708



(a)

(b)

Figure 1: Absolute error of Example 1 for equidistant nodes. (a) Absolute errors with $\beta = 0.3$. (b) Contour plot of absolute errors with $\beta = 0.3$.



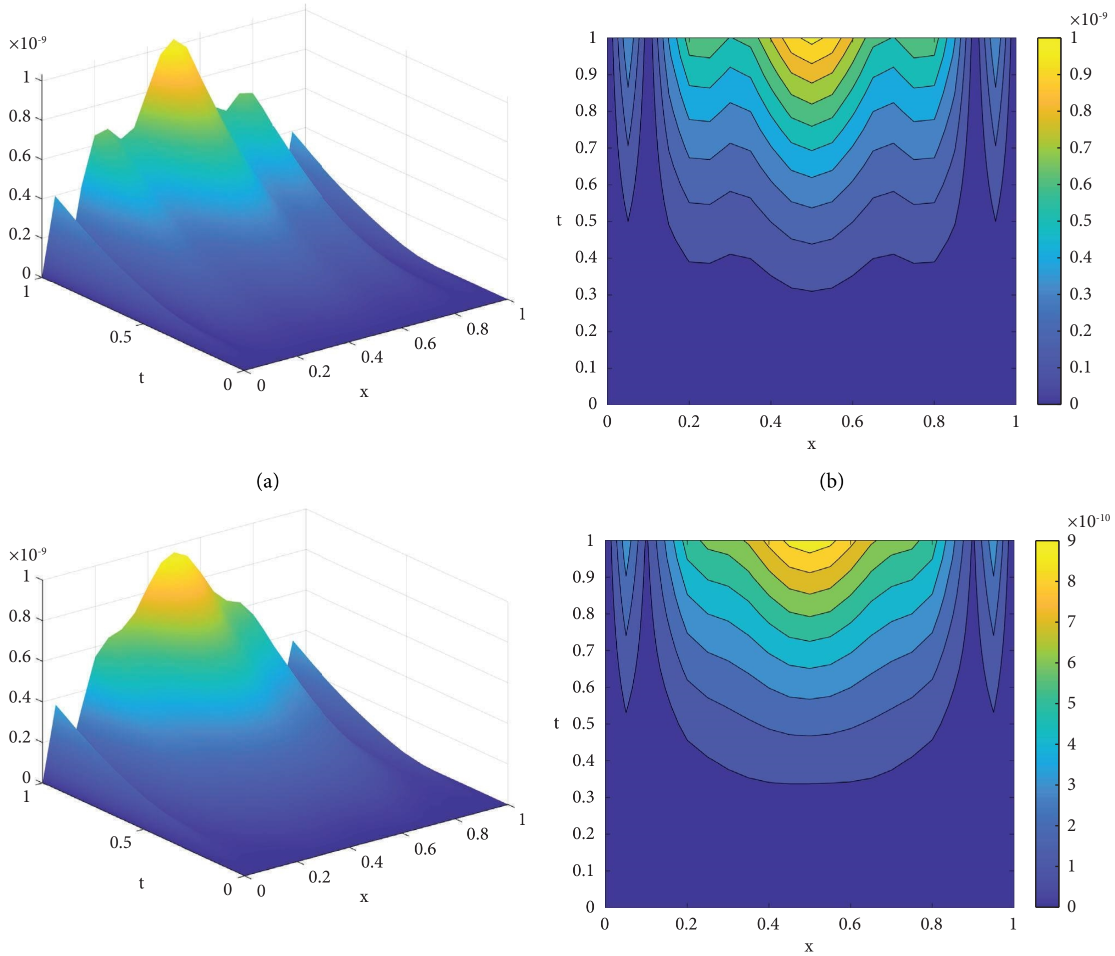
(a)

(b)

Figure 2: Absolute error of Example 1 for second class of Chebyshev nodes (a) Absolute errors with $\beta = 0.3$. (b) Contour plot of absolute errors with $\beta = 0.3$.

Table 2: The results of Example 2 with 1000 Gaussian nodes.

	$n\times m$	E_r	order	CPU times (s)
$\beta = 0.3$	8×8	4.4550×10^{-4}	—	0.1285
	10×10	8.0476×10^{-6}	5.7907	0.1242
	12×12	8.9659×10^{-8}	6.4880	0.1861
	14×14	9.6759×10^{-10}	6.5339	0.2002
$\beta = 0.5$	8×8	4.4480×10^{-4}	—	0.0897
	10×10	8.0344×10^{-6}	5.7908	0.1118
	12×12	8.9647×10^{-8}	6.4858	0.1429
	14×14	9.5702×10^{-10}	6.5946	0.2076
$\beta = 0.7$	8×8	4.4409×10^{-4}	—	0.0978
	10×10	8.0209×10^{-6}	5.7909	0.1396
	12×12	8.9565×10^{-8}	6.4847	0.1561
	14×14	8.7046×10^{-10}	6.6850	0.1932



(a)

(b)

(c)

(d)

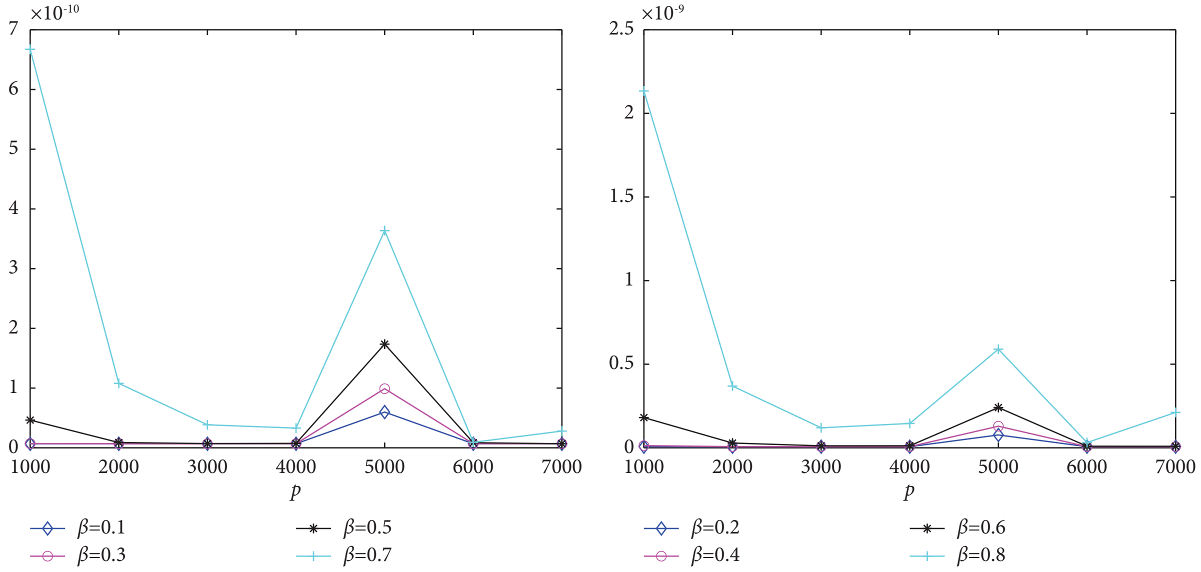
Figure 3: Absolute errors of Example 2 with $m = n = 14$ and $p = 1000$ for different β. (a) Absolute errors with $\beta = 0.3$. (b) Contour plot of absolute errors with $\beta = 0.3$. (c) Absolute errors with $\beta = 0.7$. (d) Contour plot of absolute errors with $\beta = 0.7$.

Table 3: The results of Example 3 with 6000 Gaussian nodes.

	$n \times m$	E_r	order	CPU times (s)
$\beta = 0.3$	3×3	0.0449	—	0.8851
	5×5	9.1899×10^{-6}	12.2535	1.0206
	7×7	1.0160×10^{-8}	9.8210	1.2445
	9×9	6.8431×10^{-12}	10.5359	1.5697
$\beta = 0.5$	3×3	0.0484	—	0.8719
	5×5	9.1913×10^{-6}	12.3630	1.0231
	7×7	1.0155×10^{-8}	9.8220	1.1891
	9×9	8.5369×10^{-12}	10.2161	1.4915
$\beta = 0.7$	3×3	0.0551	—	0.8813
	5×5	9.1904×10^{-6}	12.5491	1.0134
	7×7	1.0152×10^{-8}	9.8222	1.3287
	9×9	9.2665×10^{-12}	10.0974	1.4871

Table 4: Comparison of L^{∞}-errors for Example 3 with $p = 600$ and $\beta = 0.01$.

Present method		Method in [38]	
$n \times m$	L^{∞}-Err	$n \times m$	L^{∞}-Err
4×4	2.4967×10^{-5}	4×50	8.1931×10^{-5}
6×6	2.7631×10^{-8}	8×100	3.9105×10^{-6}
8×8	1.8857×10^{-11}	16×200	1.4467×10^{-7}
10×10	2.4464×10^{-12}	32×400	5.1049×10^{-9}



(a)

(b)

Figure 4: Relative errors of Example 3 with $m = n = 9$ for different β and p. (a) $\beta = 0.1$, 0.3, 0.5 and 0.7. (b) $\beta = 0.2$, 0.4, 0.6 and 0.8.

$p = 1000$. As $m = n = 14$, $\beta = 0.3$, 0.7, and $p = 1000$, the absolute errors of Example 2 are shown in Figure 3. By Figure 3 we can find that the fluctuation of error size and distribution is small when β is taken at different values, which shows the excellent stability of BLICM.

Example 3. Let $R = T = 1$, $P = 1$, $Q = 1$, $\mu(t) = t^3$, $\nu(t) = et^3$, $\psi(x) = 0$ and the forcing function is $f(x,t) = \Gamma(4)/\Gamma(4-\beta)t^{3-\beta}e^x$ in equation (1). The exact solution is $u(x,t) = t^3 e^x$.

This example can be found in [38]. The results of Example 3 are as follows. Table 3 shows the relative errors for different β with $p = 6000$. Table 4 reports the relative errors for $\beta = 0.01$ and compare the present results with the results obtained by the method in [38] (see Example 2 in [38]). We perceive from this table that the results obtained by the

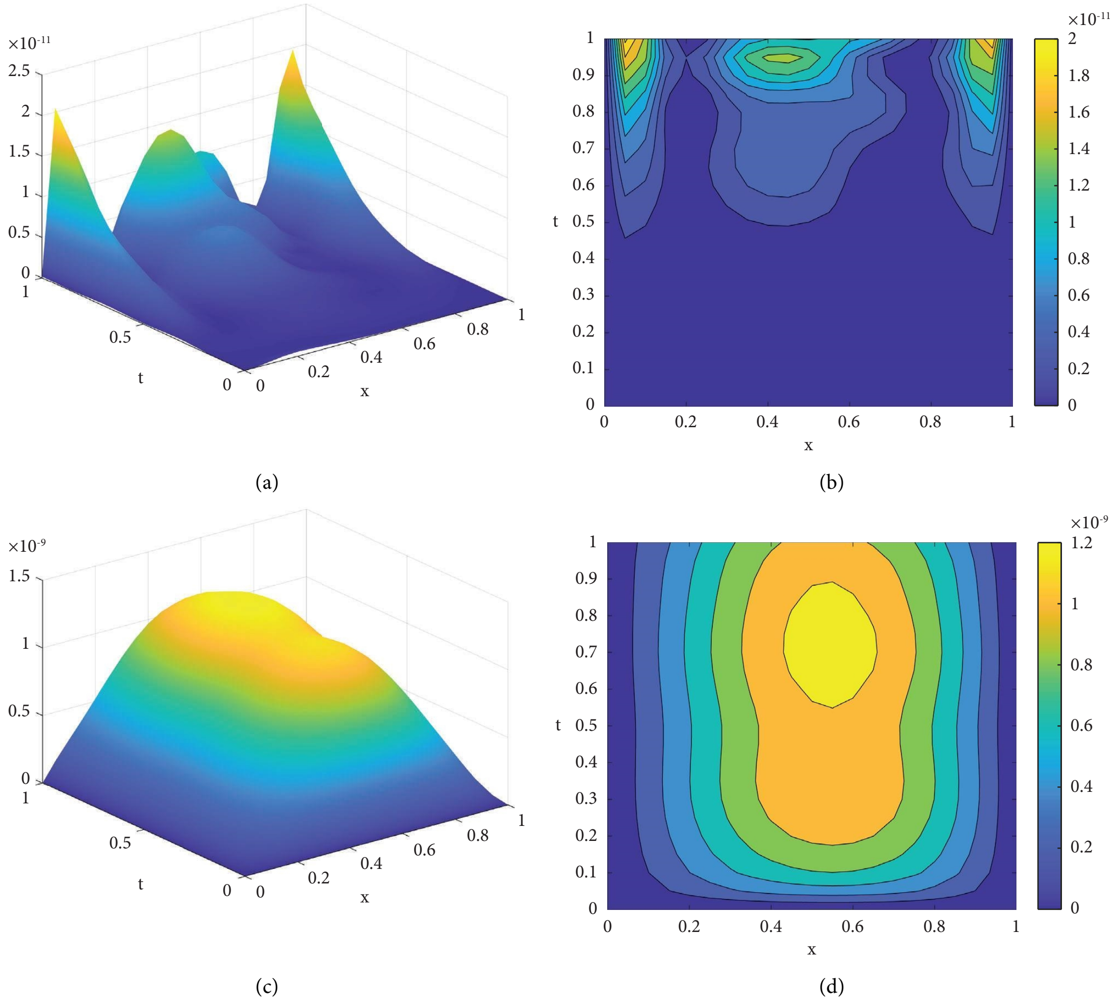


Figure 5: Absolute errors of Example 3 with $m = n = 9$ and $p = 6000$ for different β. (a) Absolute errors with $\beta = 0.1$. (b) Contour plot of absolute errors with $\beta = 0.1$. (c) Absolute errors with $\beta = 0.9$. (d) Contour plot of absolute errors with $\beta = 0.9$.

Table 5: The results of Example 4 with 1000 Gaussian nodes.

	$n \times m$	E_r	order	CPU times (s)
$\beta = 0.1$	4×4	8.4551×10^{-2}	—	0.0502
	5×5	5.0093×10^{-3}	4.0771	0.0629
	6×6	6.0879×10^{-4}	3.0406	0.0694
	7×7	3.0452×10^{-5}	4.3213	0.0853
$\beta = 0.9$	4×4	8.4482×10^{-2}	—	0.0498
	5×5	5.0906×10^{-3}	4.0527	0.0675
	6×6	6.3304×10^{-4}	3.0075	0.0860
	7×7	8.7959×10^{-5}	2.8474	0.0946

proposed method are more accurate than the results in [38]. For different β, Figure 4 shows the variability of relative error for Example 3 with different number of Gaussian nodes, it implies that the stability of the relative errors becomes better with the increasing of the number of nodes. For $\beta = 0.1$ and $\beta = 0.9$, the absolute errors for Example 3 are shown in Figure 5 with $m = n = 9$ and $p = 6000$. The numerical results of the third example also confirm the theoretical prediction and verify the effectiveness of the proposed method.

Example 4. Let $R = T = 1$, $P = 256/\Gamma(2-\beta)$, $Q = 128/\Gamma(2-\beta)$, $\mu(t) = 0$, $\nu(t) = 0$, $\psi(x) = 0$, and the forcing function is $f(x,t) = \Gamma(1+\beta)\sin(\pi x) + 256\pi^2/\Gamma(2-\beta) t^{\beta}\sin(\pi x) + 128\pi/\Gamma(2-\beta)t^{\beta}\cos(\pi x)$ in equation (1). The exact solution is $u(x,t) = t^{\beta}\sin(\pi x)$.

In this example, we want to test the problem of which the exact solution is not smooth enough. The results of Example 4 are shown in Table 5 for different β. The main purpose of this example is to verify the effectiveness of the proposed method. We can find that the performance of the proposed method will get worse for given problem, but it is still effective.

5. Conclusion

In this paper, we investigate the numerical algorithm for solving TFADE by using BLICM. Discrete scheme of TFADE is given by combining BLICM with Gauss-Legendre quadrature rule. Theoretical analysis and numerical results show that the discrete scheme constructed in our paper has high numerical convergence speed and accuracy. A comparison of the obtained results with exact solutions and other existing methods reveals that our method is more accurate and efficient for TFADE. The proposed method can be extended to solve problems of integer and noninteger orders in high dimensions. In our future work, we will include the problems of high dimensions and nonlinear fractional partial differential equations.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Xinjiang Uygur Autonomous Region Graduate Student Research Innovation Program (Grant no. XJ2023G231), Natural Science Foundation of Xinjiang Uygur Autonomous Region (Grant nos. 2022D01E13 and 2022D01B113), NSFC (Grant no. 11861068), and the Scientific Research Foundation for Outstanding Young Teachers of Xinjiang Normal University (Grant no. XJNU202112).

References

[1] Y. H. Ding and Y. X. Li, “Finite-approximate controllability of impulsive ψ Caputo fractional evolution equations with nonlocal conditions,” *Fractional Calculus and Applied Analysis*, vol. 26, no. 3, pp. 1326–1358, 2023.

[2] K. Shah, T. Abdeljawad, and A. Ali, “Mathematical analysis of the Cauchy type dynamical system under piecewise equations with Caputo fractional derivative,” *Chaos, Solitons and Fractals*, vol. 161, Article ID 112356, 2022.

[3] C. P. Li, Z. G. Zhao, and Y. Q. Chen, “Numerical approximation of nonlinear fractional differential equations with subdiffusion and superdiffusion,” *Computers and Mathematics with Applications*, vol. 62, no. 3, pp. 855–875, 2011.

[4] Q. Liu, Y. T. Gu, P. Zhuang, F. Liu, and Y. F. Nie, “An implicit RBF meshless approach for time fractional diffusion equations,” *Computational Mechanics*, vol. 48, pp. 1–12, 2011.

[5] F. H. Zeng, C. P. Li, F. W. Liu, and I. Turner, “The use of finite difference/element approaches for solving the time-fractional subdiffusion equation,” *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A2976–A3000, 2013.

[6] A. M. Nagy and A. A. El-Sayed, “An accurate numerical technique for solving two-dimensional time fractional order diffusion equation,” *International Journal of Modelling and Simulation*, vol. 39, no. 3, pp. 214–221, 2019.

[7] Y. Gu and H. G. Sun, “A meshless method for solving three-dimensional time fractional diffusion equation with variable-order derivatives,” *Applied Mathematical Modelling*, vol. 78, pp. 539–549, 2020.

[8] S. Taha Abdulazeez and M. Modanli, “Solutions of fractional order pseudo-hyperbolic telegraph partial differential equations using finite difference method,” *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 12443–12451, 2022.

[9] J. Liu, J. Zhang, and X. D. Zhang, “Semi-discretized numerical solution for time fractional convection-diffusion equation by RBF-FD,” *Applied Mathematics Letters*, vol. 128, Article ID 107880, 2022.

[10] M. Saqib, H. Hanif, T. Abdeljawad, I. Khan, S. Shafie, and K. Sooppy Nisar, “Heat transfer in mhd flow of Maxwell fluid via fractional cattaneo-friedrich model: a finite difference approach,” *Computers, Materials and Continua*, vol. 65, no. 3, pp. 1959–1973, 2020.

[11] J. X. Cao, C. P. Li, and Y. Q. Chen, “High-order approximation to Caputo derivatives and Caputo-type advection-diffusion equations (II),” *Fractional calculus and Applied analysis*, vol. 18, no. 3, pp. 735–761, 2015.

[12] C. P. Li and Z. Wang, “The local discontinuous Galerkin finite element methods for Caputo-type partial differential equations: numerical analysis,” *Applied Numerical Mathematics*, vol. 140, pp. 1–22, 2019.

[13] F. H. Zeng, C. P. Li, F. W. Liu, and I. Turner, “Numerical algorithms for time-fractional subdiffusion equation with second-order accuracy,” *SIAM Journal on Scientific Computing*, vol. 37, no. 1, pp. A55–A78, 2015.

[14] K. Shah, F. Jarad, and T. Abdeljawad, “Stable numerical results to a class of time-space fractional partial differential equations via spectral method,” *Journal of Advanced Research*, vol. 25, pp. 39–48, 2020.

[15] M. L. Zheng, F. W. Liu, I. Turner, and V. Anh, “A novel high order space-time spectral method for the time fractional Fokker-Planck equation,” *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A701–A724, 2015.

[16] E. S. Shoukralla and B. M. Ahmed, “Barycentric Lagrange interpolation methods for evaluating singular integrals,” *Alexandria Engineering Journal*, vol. 69, pp. 243–253, 2023.

[17] M. M. Yang, W. T. Ma, and Y. B. Ge, “A meshless collocation method with barycentric Lagrange interpolation for solving the Helmholtz equation,” *Computer Modeling in Engineering and Sciences*, vol. 126, no. 1, pp. 25–54, 2021.

[18] H. Y. Liu, J. Huang, W. Zhang, and Y. Y. Ma, “Meshfree approach for solving multi-dimensional systems of Fredholm integral equations via barycentric Lagrange interpolation,” *Applied Mathematics and Computation*, vol. 346, pp. 295–304, 2019.

[19] J. Li, X. N. Su, and K. Y. Zhao, “Barycentric interpolation collocation algorithm to solve fractional differential equations,” *Mathematics and Computers in Simulation*, vol. 205, pp. 340–367, 2023.

[20] J. Li and Y. L. Cheng, “Barycentric rational interpolation method for solving time-dependent fractional convection-diffusion equation,” *Electronic Research Archive*, vol. 31, no. 7, pp. 4034–4056, 2023.

[21] J. Li, “Barycentric rational collocation method for fractional reaction-diffusion equation,” *AIMS Mathematics*, vol. 8, no. 4, pp. 9009–9026, 2023.

[22] R. Metzler and J. Klafter, “The random walk’s guide to anomalous diffusion: a fractional dynamics approach,” *Physics Reports*, vol. 339, pp. 1–77, 2000.

[23] F. W. Liu, P. Zhuang, V. Anh, I. Turner, and K. Burrage, “Stability and convergence of the difference methods for the space-time fractional advection-diffusion equation,” *Applied Mathematics and Computation*, vol. 191, no. 1, pp. 12–20, 2007.

[24] H. Jiang, F. W. Liu, I. Turner, and K. Burrage, “Analytical solutions for the multi-term time-space Caputo-Riesz fractional advection-diffusion equations on a finite domain,” *Journal of Mathematical Analysis and Applications*, vol. 389, no. 2, pp. 1117–1127, 2012.

[25] L. Wei, X. D. Zhang, and Y. N. He, “Analysis of a local discontinuous Galerkin method for time-fractional advection-diffusion equations,” *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 23, no. 4, pp. 634–648, 2013.

[26] A. Tayebi, Y. Shekari, and M. H. Heydari, “A meshless method for solving two-dimensional variable-order time fractional advection-diffusion equation,” *Journal of Computational Physics*, vol. 340, pp. 655–669, 2017.

[27] Y. E. Aghdam, H. Mesgrani, M. Javidi, and O. Nikan, “A computational approach for the space-time fractional advection-diffusion equation arising in contaminant transport through porous media,” *Engineering with Computers*, vol. 37, no. 4, pp. 3615–3627, 2021.

[28] B. Guo, X. Pu, and F. Huang, *Fractional Partial Differential Equations and Their Numerical Solutions*, World Scientific, Singapore, 2015.

[29] R. Hilfer, *Applications of Fractional Calculus in Physics*, World Scientific, Singapore, 2000.

[30] I. Podlubny, *Fractional Differential Equations*, Academic Press, New York, NY, USA, 1999.

[31] J. P. Berrut and L. N. Trefethen, “Barycentric Lagrange interpolation,” *SIAM Review*, vol. 46, no. 3, pp. 501–517, 2004.

[32] D. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software*, Prentice-Hall, Inc, Upper Saddle River, NJ, USA, 1989.

[33] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, UK, 1996.

[34] R. H. Wang, *Numerical Approximation*, Higher Education Press, Beijing, China, 1999.

[35] N. J. Higham, “The numerical stability of barycentric Lagrange interpolation,” *IMA Journal of Numerical Analysis*, vol. 24, no. 4, pp. 547–556, 2004.

[36] S. C. Yi and L. Q. Yao, “A steady barycentric Lagrange interpolation method for the 2D higher-order time-fractional telegraph equation with nonlocal boundary condition with error analysis,” *Numerical Methods for Partial Differential Equations*, vol. 35, no. 5, pp. 1694–1716, 2019.

[37] X. M. Gu and S. L. Wu, “A parallel-in-time iterative algorithm for Volterra partial integro-differential problems with weakly singular kernel,” *Journal of Computational Physics*, vol. 417, Article ID 109576, 2020.

[38] W. H. Luo, C. P. Li, T. Z. Huang, X. M. Gu, and G. C. Wu, “A high-order accurate numerical scheme for the Caputo derivative with applications to fractional diffusion problems,” *Numerical Functional Analysis and Optimization*, vol. 39, no. 5, pp. 600–622, 2018.