

An efficient differential quadrature method for fractional advection–diffusion equation

X. G. Zhu · Y. F. Nie · W. W. Zhang

Received: 8 November 2016 / Accepted: 21 August 2017 / Published online: 8 September 2017
© Springer Science+Business Media B.V. 2017

Abstract This article studies a direct numerical approach for fractional advection–diffusion equations (ADEs). Using a set of cubic trigonometric B-splines as test functions, a differential quadrature (DQ) method is firstly proposed for the 1D and 2D time-fractional ADEs of order $(0, 1]$. The weighted coefficients are determined, and with them, the original equation is transformed into a group of general ordinary differential equations (ODEs), which are discretized by an effective difference scheme or Runge–Kutta method. The stability is investigated under a mild theoretical condition. Secondly, based on a set of cubic B-splines, we develop a new Crank–Nicolson type DQ method for the 2D space-fractional ADEs without advection. The DQ approximations to fractional derivatives are introduced, and the values of the fractional derivatives of B-splines are computed by deriving explicit formulas. The presented DQ methods are evaluated on five benchmark problems and the simulations of the unsteady propaga-

tion of solitons and Gaussian pulse. In comparison with the algorithms in the open literature, numerical results finally illustrate the validity and accuracy.

Keywords Differential quadrature (DQ) · Cubic trigonometric B-splines · Cubic B-splines · Fractional advection–diffusion equation

1 Introduction

The differential equations with a fractional derivative serve as superior models in subjects as diverse as astrophysics, chaotic dynamics, fractal network, signal processing, continuum mechanics, turbulent flow and wave propagation [29, 34, 40, 51]. This type of equations admit the non-local memory effects in mathematical mechanism, thereby filling in a big gap that the classical models cannot work well for some of the natural phenomena like anomalous transport. In general, the exact solutions can seldom be represented as closed-form expressions by using elementary functions that presents a tough challenge to derive a sufficiently valid method concerned with analytic approximations, so a keen interest has been attracted to design robust algorithms to investigate them in numerical perspectives.

In this article, we aim to construct an efficient method to numerically solve the general problems:

X. G. Zhu · Y. F. Nie (✉) · W. W. Zhang
Department of Applied Mathematics, Northwestern
Polytechnical University, Xi'an 710129, Shaanxi,
People's Republic of China
e-mail: yfnie@nwpu.edu.cn

X. G. Zhu
e-mail: zhuxg590@yeah.net

W. W. Zhang
e-mail: wwzhang@nwpu.edu.cn

(I) 1D time-fractional ADEs

$$\frac{\partial^\alpha u(x, t)}{\partial t^\alpha} + \kappa \frac{\partial u(x, t)}{\partial x} - \varepsilon \frac{\partial^2 u(x, t)}{\partial x^2} = f(x, t), \quad (1)$$

with $0 < \alpha \leq 1$, $\kappa, \varepsilon \geq 0$, $a \leq x \leq b$, $t > 0$, and the initial and boundary conditions

$$u(x, 0) = \psi(x), \quad a \leq x \leq b, \quad (2)$$

$$u(a, t) = g_1(t), \quad u(b, t) = g_2(t), \quad t > 0; \quad (3)$$

(II) 2D time-fractional ADEs

$$\begin{aligned} \frac{\partial^\alpha u(x, y, t)}{\partial t^\alpha} + \kappa_x \frac{\partial u(x, y, t)}{\partial x} + \kappa_y \frac{\partial u(x, y, t)}{\partial y} \\ - \varepsilon_x \frac{\partial^2 u(x, y, t)}{\partial x^2} - \varepsilon_y \frac{\partial^2 u(x, y, t)}{\partial y^2} = f(x, y, t), \end{aligned} \quad (4)$$

with $0 < \alpha \leq 1$, $\kappa_x, \kappa_y, \varepsilon_x, \varepsilon_y \geq 0$, $(x, y) \in \Omega$, $t > 0$, and the initial and boundary conditions

$$u(x, y, 0) = \psi(x, y), \quad (x, y) \in \Omega, \quad (5)$$

$$u(x, y, t) = g(x, y, t), \quad (x, y) \in \partial\Omega, \quad t > 0, \quad (6)$$

where $\Omega = \{(x, y) : a \leq x \leq b, c \leq y \leq d\}$ and $\partial\Omega$ denotes its boundary;

(III) 2D space-fractional ADEs without advection

$$\begin{aligned} \frac{\partial u(x, y, t)}{\partial t} - \varepsilon_x \frac{\partial^{\beta_1} u(x, y, t)}{\partial x^{\beta_1}} \\ - \varepsilon_y \frac{\partial^{\beta_2} u(x, y, t)}{\partial y^{\beta_2}} = f(x, y, t), \end{aligned} \quad (7)$$

with $1 < \beta_1, \beta_2 < 2$, $\varepsilon_x, \varepsilon_y \geq 0$, $(x, y) \in \Omega$, $t > 0$, and the initial and boundary conditions

$$u(x, y, 0) = \psi(x, y), \quad (x, y) \in \Omega, \quad (8)$$

$$u(x, y, t) = 0, \quad (x, y) \in \partial\Omega, \quad t > 0, \quad (9)$$

where Ω and $\partial\Omega$ are given as above.

In Eqs. (1), (4), the time-fractional derivatives are defined in Caputo sense, i.e.,

$$\frac{\partial^\alpha u(x, t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x, \xi)}{\partial \xi} \frac{d\xi}{(t-\xi)^\alpha},$$

while in Eq. (7), the space-fractional derivatives are defined in Riemann–Liouville sense, i.e.,

$$\begin{aligned} \frac{\partial^{\beta_1} u(x, y, t)}{\partial x^{\beta_1}} &= \frac{1}{\Gamma(2-\beta_1)} \frac{\partial^2}{\partial x^2} \int_a^x \frac{u(\xi, y, t) d\xi}{(x-\xi)^{\beta_1-1}}, \\ \frac{\partial^{\beta_2} u(x, y, t)}{\partial y^{\beta_2}} &= \frac{1}{\Gamma(2-\beta_2)} \frac{\partial^2}{\partial y^2} \int_c^y \frac{u(x, \xi, t) d\xi}{(y-\xi)^{\beta_2-1}}, \end{aligned}$$

and $\frac{\partial^\alpha u(x, y, t)}{\partial t^\alpha}$ is an analog of $\frac{\partial^\alpha u(x, t)}{\partial t^\alpha}$, where $\Gamma(\cdot)$ is the Gamma function. It is noted that Eqs. (1)–(3), (4)–(6), and (7)–(9) reduce into the classical 1D or 2D ADEs if $\alpha = 1$, $\beta_1 = \beta_2 = 2$ are fixed.

In recent decades, fractional ADEs have been notable subjects of intense research. Except for a few analytic solutions, various numerical methods have been done for Eqs. (1)–(3) without advection, covering implicit difference method [57], high-order finite element method (FEM) [17], Legendre wavelets and spectral Galerkin methods [11, 23], direct discontinuous Galerkin method [13], quadratic spline collocation method [25], cubic B-spline collocation method (CBCM) [38], orthogonal spline collocation method [50], pseudo-spectral method [9], high-order compact difference method [16], implicit radial basis function (RBF) meshless method [24], nonpolynomial and polynomial spline methods [12]. In [3, 5, 37, 56], the algorithms based on shifted fractional Jacobi polynomials, Sinc functions and shifted Legendre polynomials, Haar wavelets and the third kind Chebyshev wavelets functions were well developed via the integral operational matrix or collocation strategy for Eqs. (1)–(3) with variable coefficients. The Gegenbauer polynomial spectral collocation method was proposed in [14] for the same type of equations, and a Sinc-Haar collocation method can be found in [33]. Uddin and Haq considered a radial basis interpolation approach [48]. Cui established a high-order compact exponential difference scheme [6]. Razminia et al. proposed a DQ method for time-fractional diffusion equations by using Lagrangian interpolation polynomials as test functions [35]. Shirzadi et al. solved the 2D time-fractional ADEs with a reaction term via a local Petrov–Galerkin meshless method [39]. Gao and Sun derived two different three-point combined compact alternating direction implicit (ADI) schemes for Eqs. (4)–(6) [10], both of which own high accuracy. High-dimensional space-fractional ADEs are challenging topics in whether analytic or numerical aspects due to

the complexity and huge computing burden. The application of an numerical method to Eqs. (7)–(9) did not have large diffusion; for the conventional algorithms, we refer the readers to [15, 28, 36, 44, 52, 54] and references therein.

The trigonometric B-splines are a class of piecewise-defined functions constructed from algebraic trigonometric spaces, which have got recognition since 1964. They are preferred to the familiar polynomial B-splines since they often yield less errors when served as basis functions in interpolation theory. Nevertheless, using these basis splines to set up numerical algorithms is in its infancy and the related works are limited [1, 30]. In this study, a DQ method for the general ADEs is developed with its weighted coefficients calculated based on cubic trigonometric B-spline (CTB) functions. The basis splines are slightly modified for brevity and a few advantages. Difference schemes and Runge–Kutta Gill’s method are introduced to discretize the resulting ODEs. The condition ensuring the stability of the time-stepping DQ method is discussed and found to be rather mild. Also, we propose a new cubic B-splines-based DQ method for the 2D space-fractional diffusion equations by introducing the DQ approximations to fractional derivatives. The weights are determined by deriving explicit formulas for the fractional derivatives of B-splines through a recursive technique of partial integration. The approaches in presence are straight forward to apply and simple in implementation on computers; numerical results highlight the superiority over some previous algorithms.

The remainder is as follows. In Sect. 2, we outline some basic definitions and the cubic spline functions that are useful hereinafter. In Sect. 3, how to determine the weighted coefficients based on these CTB functions is studied and a time-stepping DQ method is constructed for Eqs. (1)–(3) and Eqs. (4)–(6). Section 4 elaborates on its stable analysis. In Sect. 5, we suggest a spline-based DQ method for Eqs. (7)–(9) based on a set of cubic B-splines by explicitly computing the values of their fractional derivatives at sampling points. A couple of numerical examples are included in Sect. 6, which manifest the effectiveness of our methods. The last section devotes to a conclusion.

2 Preliminaries

Let $M, N \in \mathbb{Z}^+$ and a time–space lattice be

$$\begin{aligned} \Omega_\tau &= \{t_n : t_n = n\tau, 0 \leq n \leq N\}, \\ \Omega_h &= \{x_i : x_i = a + ih, 0 \leq i \leq M\}, \end{aligned}$$

with $\tau = T/N, h = (b-a)/M$ on $(0, T] \times [a, b]$. Then, some auxiliary results are introduced for preliminaries.

2.1 Fractional derivatives and their discretizations

Given a good enough $f(x, t)$, the formulas

$$\begin{aligned} {}_0^C D_t^\alpha f(x, t) &= \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{\partial^m f(x, \xi)}{\partial \xi^m} \frac{d\xi}{(t-\xi)^{1+\alpha-m}}, \\ {}_0^{\text{RL}} D_t^\alpha f(x, t) &= \frac{1}{\Gamma(m-\alpha)} \frac{\partial^m}{\partial t^m} \int_0^t \frac{f(x, \xi) d\xi}{(t-\xi)^{1+\alpha-m}}, \end{aligned}$$

define the α -th Caputo and Riemann–Liouville derivatives, respectively, where $m - 1 < \alpha < m, m \in \mathbb{Z}^+$, and particularly, in the case of $\alpha = m$, both of them degenerate into the m -th integer-order derivative.

The two frequently-used fractional derivatives are equivalent with exactness to an additive factor, i.e.,

$${}_0^C D_t^\alpha f(x, t) = {}_0^{\text{RL}} D_t^\alpha f(x, t) - \sum_{l=0}^{m-1} \frac{f^{(l)}(x, 0) t^{l-\alpha}}{\Gamma(l+1-\alpha)}; \tag{10}$$

see [21, 34] for references. Utilizing ${}_0^{\text{RL}} D_t^\alpha t^l = \frac{\Gamma(l+1) t^{l-\alpha}}{\Gamma(l+1-\alpha)}$ and a proper scheme to discretize the Riemann–Liouville derivatives on the right side of Eq. (10), a difference scheme for Caputo derivative can be

$$\begin{aligned} {}_0^C D_t^\alpha f(x, t_n) &\cong \frac{1}{\tau^\alpha} \sum_{k=0}^n \omega_k^\alpha f(x, t_{n-k}) \\ &\quad - \frac{1}{\tau^\alpha} \sum_{l=0}^{m-1} \sum_{k=0}^n \frac{\omega_k^\alpha f^{(l)}(x, 0) t_{n-k}^l}{l!}, \end{aligned} \tag{11}$$

with several valid alternatives of the discrete coefficients $\{\omega_k^\alpha\}_{k=0}^n$ [4]. Typically, we have

$$\omega_k^\alpha = (-1)^k \binom{\alpha}{k} = \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}, \quad k \geq 0, \tag{12}$$

whose truncated error is $\mathcal{R}_\tau = \mathcal{O}(\tau)$, and

$$\omega_k^\alpha = \left(\frac{11}{6}\right)^\alpha \sum_{p=0}^k \sum_{q=0}^p \mu^q \bar{\mu}^{p-q} l_q^\alpha l_{p-q}^\alpha l_{k-p}^\alpha, \quad k \geq 0, \tag{13}$$

with $\mu = \frac{4}{7+\sqrt{39i}}$, $\bar{\mu} = \frac{4}{7-\sqrt{39i}}$, $i = \sqrt{-1}$, and

$$l_0^\alpha = 1, \quad l_k^\alpha = \left(1 - \frac{\alpha + 1}{k}\right) l_{k-1}^\alpha, \quad k \geq 1, \tag{14}$$

in which case, the truncated error fulfills $\mathcal{R}_\tau = \mathcal{O}(\tau^3)$. Actually, Eq. (14) is the recursive relation of Eq. (12). Moreover, the coefficients $\{\omega_k^\alpha\}_{k=0}^n$ in Eq. (12) satisfy

- (i) $\omega_0^\alpha = 1, \quad \omega_k^\alpha < 0, \quad \forall k \geq 1,$
- (ii) $\sum_{k=0}^\infty \omega_k^\alpha = 0, \quad \sum_{k=0}^{n-1} \omega_k^\alpha > 0.$

These properties are easily obtained from [34].

Reset $0 < \alpha < 1$, (11) thus turns into

$${}_0^C D_t^\alpha f(x, t_n) = \frac{1}{\tau^\alpha} \sum_{k=0}^{n-1} \omega_k^\alpha f(x, t_{n-k}) - \frac{1}{\tau^\alpha} \sum_{k=0}^{n-1} \omega_k^\alpha f(x, 0) + \mathcal{R}_\tau. \tag{15}$$

It is noteworthy that Eq. (15) gives a smooth transition to the classic schemes when $\alpha = 1$, for instance, Eq. (15) would be the four-point backward difference scheme if $\alpha = 1$ and $\{\omega_k^\alpha\}_{k=0}^n$ are chosen to be the ones in Eq. (13), because these coefficients also fulfill $\sum_{k=0}^\infty \omega_k^\alpha = 0$ and vanish apart from $\omega_0^\alpha, \omega_1^\alpha, \omega_2^\alpha$ and ω_3^α .

2.2 Cubic spline functions

Let $x_{-i} = a - ih, x_{M+i} = b + ih, i = 1, 2, 3$ be the six ghost knots outside $[a, b]$. Then the desirable CTB basis functions $\{CTB_m(x)\}_{m=-1}^{M+1}$ are defined as [30, 49]

$$CTB_m(x) = \frac{1}{\chi} \begin{cases} \phi_1(x), & x \in [x_{m-2}, x_{m-1}) \\ \phi_2(x), & x \in [x_{m-1}, x_m) \\ \phi_3(x), & x \in [x_m, x_{m+1}) \\ \phi_4(x), & x \in [x_{m+1}, x_{m+2}) \\ 0, & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} \phi_1(x) &= p^3(x_{m-2}), \\ \phi_2(x) &= q(x_{m+2})p^2(x_{m-1}) + p^2(x_{m-2})q(x_m) \\ &\quad + p(x_{m-2})p(x_{m-1})q(x_{m+1}), \\ \phi_3(x) &= p(x_{m-2})q^2(x_{m+1}) + q^2(x_{m+2})p(x_m) \\ &\quad + p(x_{m-1})q(x_{m+1})q(x_{m+2}), \\ \phi_4(x) &= q^3(x_{m+2}), \end{aligned}$$

with the notations

$$\begin{aligned} p(x_m) &= \sin\left(\frac{x - x_m}{2}\right), \\ q(x_m) &= \sin\left(\frac{x_m - x}{2}\right), \\ \chi &= \sin\left(\frac{h}{2}\right) \sin(h) \sin\left(\frac{3h}{2}\right). \end{aligned}$$

The values of $CTB_m(x)$ at each knot are given by

$$CTB_m(x_i) = \begin{cases} \sin^2\left(\frac{h}{2}\right) \csc(h) \csc\left(\frac{3h}{2}\right), & i = m \pm 1 \\ \frac{2}{1 + 2 \cos(h)}, & i = m \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

and the values of $CTB'_m(x)$ at each knot are given by

$$CTB'_m(x_i) = \begin{cases} \frac{3}{4} \csc\left(\frac{3h}{2}\right), & i = m - 1 \\ -\frac{3}{4} \csc\left(\frac{3h}{2}\right), & i = m + 1 \\ 0. & \text{otherwise} \end{cases} \tag{17}$$

Using the same grid information, the cubic B-spline basis functions $\{B_m(x)\}_{m=-1}^{M+1}$ are defined by

$$B_m(x) = \frac{1}{h^3} \begin{cases} \varphi_1(x), & x \in [x_{m-2}, x_{m-1}) \\ \varphi_2(x), & x \in [x_{m-1}, x_m) \\ \varphi_3(x), & x \in [x_m, x_{m+1}) \\ \varphi_4(x), & x \in [x_{m+1}, x_{m+2}) \\ 0, & \text{otherwise} \end{cases}$$

with the piecewise functions

$$\varphi_1(x) = (x - x_{m-2})^3,$$

$$\begin{aligned} \varphi_2(x) &= (x - x_{m-2})^3 - 4(x - x_{m-1})^3, \\ \varphi_3(x) &= (x_{m+2} - x)^3 - 4(x_{m+1} - x)^3, \\ \varphi_4(x) &= (x_{m+2} - x)^3. \end{aligned}$$

Both $\{CTB_m(x)\}_{m=-1}^{M+1}$ and $\{B_m(x)\}_{m=-1}^{M+1}$ are locally compact and twice continuously differentiable. Since the knots x_{-1}, x_{M+1} lie beyond $[a, b]$ and the weights in relation to the B-splines center at both knots do not participate in practical computation, hereunder, as in [26] for cubic B-splines, we modify the CTBs by

$$\begin{cases} MTB_0(x) = CTB_0(x) + 2CTB_{-1}(x), \\ MTB_1(x) = CTB_1(x) - CTB_{-1}(x), \\ MTB_m(x) = CTB_m(x), \quad m = 2, 3, \dots, M - 2, \\ MTB_{M-1}(x) = CTB_{M-1}(x) - CTB_{M+1}(x), \\ MTB_M(x) = CTB_M(x) + 2CTB_{M+1}(x), \end{cases} \tag{18}$$

for simplicity, which will result in a strictly tri-diagonal algebraic system after discretization on the uniform grid. $\{MTB_m(x)\}_{m=0}^M$ are also linearly independent and constitute a family of basis elements of a spline space.

3 Description of CTB-based DQ method

On a 2D domain $[a, b] \times [c, d]$, letting $M_x, M_y \in \mathbb{Z}^+$, we add a spatial lattice with equally spaced grid points with spacing of $h_x = (b - a)/M_x$ in x -axis and $h_y = (d - c)/M_y$ in y -axis, i.e.,

$$\begin{aligned} \Omega_x &= \{x_i : x_i = a + ih_x, 0 \leq i \leq M_x\}, \\ \Omega_y &= \{y_j : y_j = c + jh_y, 0 \leq j \leq M_y\}. \end{aligned}$$

DQ method is understood as a numerical technique for finding the approximate solutions of differential equations that reduces the original problem to those of solving a system of algebraic or ordinary differential equations via replacing the spatial partial derivatives by the representative weighted combinations of the functional values at certain grid points on the whole domain [2]. The key procedure of such method lies in the determination of its weights and the selection of the test functions whose derivative values are explicit at the

prescribed discrete grid points. As requested, we let

$$\frac{\partial^s u(x_i, t)}{\partial x^s} \cong \sum_{j=0}^M a_{ij}^{(s)} u(x_j, t), \quad 0 \leq i \leq M, \tag{19}$$

while for 2D problems, we let

$$\frac{\partial^s u(x_i, y_j, t)}{\partial x^s} \cong \sum_{m=0}^{M_x} a_{im}^{(s)} u(x_m, y_j, t), \tag{20}$$

$$\frac{\partial^s u(x_i, y_j, t)}{\partial y^s} \cong \sum_{m=0}^{M_y} b_{jm}^{(s)} u(x_i, y_m, t), \tag{21}$$

where $s \in \mathbb{Z}^+, 0 \leq i \leq M_x, 0 \leq j \leq M_y$ and $a_{ij}^{(s)}, a_{im}^{(s)}, b_{jm}^{(s)}$ are the weighted coefficients allowing us to approximate the s -th derivatives or partial derivatives at the given grid points in the DQ methods.

3.1 The calculation of weighted coefficients

In the sequel, we apply $\{MTB_m(x)\}_{m=0}^M$ to calculate the 1D, 2D unknown weights. Putting $s = 1$ and substituting these basis splines into Eq. (19), we get

$$\frac{\partial MTB_m(x_i)}{\partial x} = \sum_{j=0}^M a_{ij}^{(1)} MTB_m(x_j), \quad 0 \leq i, m \leq M,$$

with the weighted coefficients of the first-order derivative $a_{ij}^{(1)}, 0 \leq i, j \leq M$, yet to be determined. In view of (18) and the properties (16)–(17), some manipulations on the above equations yield the matrix–vector forms

$$\begin{cases} \mathbf{Aa}_0^{(1)} = \mathbf{Z}_0, \\ \mathbf{Aa}_1^{(1)} = \mathbf{Z}_1, \\ \vdots \\ \mathbf{Aa}_M^{(1)} = \mathbf{Z}_M, \end{cases} \tag{22}$$

where \mathbf{A} is the $(M + 1) \times (M + 1)$ coefficient matrix

$$\mathbf{A} = \begin{pmatrix} A_0 + 2A_1 & A_1 & & & & & \\ & 0 & A_0 & A_1 & & & \\ & & A_1 & A_0 & A_1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & A_1 & A_0 & 0 \\ & & & & & & A_1 & A_0 + 2A_1 \end{pmatrix},$$

$$A_0 = \frac{2}{1 + 2 \cos(h)},$$

$$A_1 = \sin^2\left(\frac{h}{2}\right) \csc(h) \csc\left(\frac{3h}{2}\right),$$

$\mathbf{a}_k^{(1)}$, $0 \leq k \leq M$, are the weighted coefficient vectors at x_k , i.e., $\mathbf{a}_k^{(1)} = [a_{k0}^{(1)}, a_{k1}^{(1)}, \dots, a_{kM}^{(1)}]^T$, and the right-side vectors \mathbf{Z}_k at x_k , $0 \leq k \leq M$, are as follows

$$\mathbf{Z}_0 = \begin{pmatrix} -2z \\ 2z \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{Z}_1 = \begin{pmatrix} -z \\ 0 \\ z \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \dots,$$

$$\mathbf{Z}_{M-1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -z \\ 0 \\ z \end{pmatrix}, \quad \mathbf{Z}_M = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -2z \\ 2z \end{pmatrix},$$

with $z = \frac{3}{4} \csc(\frac{3h}{2})$, respectively. Thus, $a_{ij}^{(1)}$ are obtained by solving Eqs. (22) for each point x_i . There are two different way to derive the weighted coefficients $a_{ij}^{(2)}$ of the second-order derivative: (i) do a similar fashion as above by putting $s = 2$ in Eq. (19) and solve an algebraic system for each grid point; (ii) find the weighted coefficients $a_{ij}^{(s)}$, $s \geq 2$, corresponding to the high-order derivatives in a recursive style [41], i.e.,

$$a_{ij}^{(s)} = s \left(a_{ii}^{(s-1)} a_{ij}^{(1)} - \frac{a_{ij}^{(s-1)}}{x_i - x_j} \right), \quad i \neq j, \quad 0 \leq i \leq M,$$

$$a_{ii}^{(s)} = - \sum_{j=0, j \neq i}^M a_{ij}^{(s)}, \quad i = j,$$

which includes $s = 2$ as a special case. The former would be less efficient since the associated equations have to be solved as priority, so the latter one will be selected during our entire computing process. Proceeding as before via replacing Ω_h by Ω_x, Ω_y leads to a 2D generalization to get $a_{im}^{(1)}, b_{jm}^{(1)}$ of the first-order partial derivatives with regard to variables x, y in Eqs. (20)–(21) and by them, the following relationships can further be applied, i.e.,

$$a_{im}^{(s)} = s \left(a_{ii}^{(s-1)} a_{im}^{(1)} - \frac{a_{im}^{(s-1)}}{x_i - x_m} \right), \quad i \neq m, \quad 0 \leq i \leq M_x,$$

$$a_{ii}^{(s)} = - \sum_{m=0, m \neq i}^{M_x} a_{im}^{(s)}, \quad i = m,$$

$$b_{jm}^{(s)} = s \left(b_{jj}^{(s-1)} b_{jm}^{(1)} - \frac{b_{jm}^{(s-1)}}{y_j - y_m} \right), \quad j \neq m, \quad 0 \leq j \leq M_y,$$

$$b_{jj}^{(s)} = - \sum_{m=0, m \neq j}^{M_y} b_{jm}^{(s)}, \quad j = m,$$

to calculate $a_{im}^{(s)}, b_{jm}^{(s)}$ with $s \geq 2$.

A point worth noticing is that $A_0, A_1 > 0$, when $0 < h < 1, 0 < h_x, h_y < 1$. Since A_0, A_1 can be deemed to be the functions of h , we obtain their derivatives

$$A'_0 = \frac{4 \sin(h)}{(1 + 2 \cos(h))^2},$$

$$A'_1 = \frac{\sec(\frac{h}{2}) \tan(\frac{h}{2})(5 + 6 \cos(h))}{4(1 + 2 \cos(h))^2}.$$

On letting $0 < h < 1$, both are proved to be larger than zero, i.e., A_0, A_1 are the increasing functions with respect to h . On the other hand, there exist $A_0(0) = 0.6667, A_1(1) = 0.2738$. Then, it suffices to show

$$\frac{2}{1 + 2 \cos(h)} > 2 \sin^2\left(\frac{h}{2}\right) \csc(h) \csc\left(\frac{3h}{2}\right),$$

which implies $A_0 > 2A_1$, and thus \mathbf{A} is a strictly diagonally dominant tri-diagonal matrix. Hence, Thomas algorithm can be applied to tackle the algebraic equations as Eqs. (22), which simply requires the arithmetic operation cost $\mathcal{O}(M + 1)$ and would greatly economize on the memory and computing time in practice.

3.2 Construction of CTB-based DQ method

In this subpart, a DQ method based on $\{MTB_m(x)\}_{m=0}^M$ (MCTB-DQM) is constructed for Eqs. (1)–(3) and Eqs. (4)–(6). Let $s = 1, 2$. The substitution of the weighted sums (19), (20)–(21) into the main equations gives

$$\frac{\partial^\alpha u(x_i, t)}{\partial t^\alpha} + \kappa \sum_{j=0}^M a_{ij}^{(1)} u(x_j, t) - \varepsilon \sum_{j=0}^M a_{ij}^{(2)} u(x_j, t) = f(x_i, t),$$

with $i = 0, 1, \dots, M$, and

$$\frac{\partial^\alpha u(x_i, y_j, t)}{\partial t^\alpha} + \kappa_x \sum_{m=0}^{M_x} a_{im}^{(1)} u(x_m, y_j, t) + \kappa_y \sum_{m=0}^{M_y} b_{jm}^{(1)} u(x_i, y_m, t) - \varepsilon_x \sum_{m=0}^{M_x} a_{im}^{(2)} u(x_m, y_j, t) - \varepsilon_y \sum_{m=0}^{M_y} b_{jm}^{(2)} u(x_i, y_m, t) = f(x_i, y_j, t),$$

with $i = 0, 1, \dots, M_x, j = 0, 1, \dots, M_y$, which are indeed a group of α -th general ODEs associated with the boundary constraints (3), (6), and involve $\alpha \in (0, 1)$ and $\alpha = 1$ as two separate cases. In what follows, we employ the notations

$$u_i^n = u(x_i, t_n), \quad u_{ij}^n = u(x_i, y_j, t_n), \\ f_i^n = f(x_i, t_n), \quad f_{ij}^n = f(x_i, y_j, t_n), \\ g_1^n = g_1(t_n), \quad g_2^n = g_2(t_n), \quad g_{ij}^n = g(x_i, y_j, t_n),$$

for the ease of exposition, where $n = 0, 1, \dots, N$.

3.2.1 The case of fractional order

Discretizing the ODEs above by the difference scheme (15) and imposing boundary constraints, we have

$$\begin{cases} \omega_0^\alpha U_i^n + \kappa \tau^\alpha \sum_{j=1}^{M-1} a_{ij}^{(1)} U_j^n - \varepsilon \tau^\alpha \sum_{j=1}^{M-1} a_{ij}^{(2)} U_j^n \\ = - \sum_{k=1}^{n-1} \omega_k^\alpha U_i^{n-k} + \sum_{k=0}^{n-1} \omega_k^\alpha U_i^0 + \tau^\alpha G_i^n, \end{cases} \quad (23)$$

with $i = 1, 2, \dots, M - 1$ and

$$G_i^n = f_i^n - \kappa (a_{i0}^{(1)} g_1^n + a_{iM}^{(1)} g_2^n) + \varepsilon (a_{i0}^{(2)} g_1^n + a_{iM}^{(2)} g_2^n),$$

for Eqs. (1)–(3), and the following scheme

$$\begin{cases} \omega_0^\alpha U_{ij}^n + \kappa_x \tau^\alpha \sum_{m=1}^{M_x-1} a_{im}^{(1)} U_{mj}^n + \kappa_y \tau^\alpha \sum_{m=1}^{M_y-1} b_{jm}^{(1)} U_{im}^n \\ - \varepsilon_x \tau^\alpha \sum_{m=1}^{M_x-1} a_{im}^{(2)} U_{mj}^n - \varepsilon_y \tau^\alpha \sum_{m=1}^{M_y-1} b_{jm}^{(2)} U_{im}^n \\ = - \sum_{k=1}^{n-1} \omega_k^\alpha U_{ij}^{n-k} + \sum_{k=0}^{n-1} \omega_k^\alpha U_{ij}^0 + \tau^\alpha G_{ij}^n, \end{cases} \quad (24)$$

with $i = 1, 2, \dots, M_x - 1, j = 1, 2, \dots, M_y - 1$, and

$$G_{ij}^n = f_{ij}^n - \kappa_x (a_{i0}^{(1)} g_{0j}^n + a_{iM_x}^{(1)} g_{M_x j}^n) - \kappa_y (b_{j0}^{(1)} g_{i0}^n + b_{jM_y}^{(1)} g_{iM_y}^n) + \varepsilon_x (a_{i0}^{(2)} g_{0j}^n + a_{iM_x}^{(2)} g_{M_x j}^n) + \varepsilon_y (b_{j0}^{(2)} g_{i0}^n + b_{jM_y}^{(2)} g_{iM_y}^n),$$

for Eqs. (4)–(6). Eqs. (23)–(24) can further be rewritten in matrix–vector forms, for instance, letting

$$\mathbf{U}^n = [U_{11}^n, \dots, U_{M_x-1,1}^n, U_{12}^n, \dots, U_{M_x-1, M_y-1}^n]^T, \\ \mathbf{G}^n = [G_{11}^n, \dots, G_{M_x-1,1}^n, G_{12}^n, \dots, G_{M_x-1, M_y-1}^n]^T,$$

for Eqs. (24), we have

$$\omega_0^\alpha \mathbf{U}^n + \tau^\alpha \mathbf{K} \mathbf{U}^n = - \sum_{k=1}^{n-1} \omega_k^\alpha \mathbf{U}^{n-k} + \sum_{k=0}^{n-1} \omega_k^\alpha \mathbf{U}^0 + \tau^\alpha \mathbf{G}^n, \quad (25)$$

where

$$\mathbf{K} = \kappa_x \mathbf{I}_y \otimes \mathbf{W}_x^1 + \kappa_y \mathbf{W}_y^1 \otimes \mathbf{I}_x - \varepsilon_x \mathbf{I}_y \otimes \mathbf{W}_x^2 - \varepsilon_y \mathbf{W}_y^2 \otimes \mathbf{I}_x,$$

with $\mathbf{I}_x, \mathbf{I}_y$ being the identity matrices in x - and y -axis, “ \otimes ” being Kronecker product, and

$$W_z^c = \begin{pmatrix} w_{11}^{(c)} & w_{12}^{(c)} & \cdots & w_{1,M_z-1}^{(c)} \\ w_{21}^{(c)} & w_{22}^{(c)} & \cdots & w_{2,M_z-1}^{(c)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M_z-1,1}^{(c)} & w_{M_z-1,2}^{(c)} & \cdots & w_{M_z-1,M_z-1}^{(c)} \end{pmatrix}, \quad c = 1, 2,$$

in which, $z = x$ if $w = a$ while $z = y$ if $w = b$. The initial states are got from Eqs. (2), (5). As a result, the approximate solutions are obtained via performing the iteration in Eqs. (23)–(24) until the last time level by rewriting them in matrix–vector forms first.

3.2.2 The case of integer order

When $\alpha = 1$, despite $\omega_0^1 = 1.8333$, $\omega_1^1 = -3$, $\omega_2^1 = 1.5$, $\omega_3^1 = -0.3333$, being the coefficients of the four-point backward difference scheme, the initial values with the errors of the same convergent rate are generally necessary to start Eqs. (23)–(24). However, this situation would not happen if $\{\omega_k^\alpha\}_{k=0}^n$ in Eq. (12) are applied. In such a case, to make the algorithm to be more cost-effective, we use Runge–Kutta Gill’s method to handle those ODEs instead, which is explicit and fourth-order convergent. Rearrange the ODEs in a unified form

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{F}(\mathbf{u}), \tag{26}$$

then the DQ method is constructed as follow

$$\begin{aligned} \mathbf{U}^n &= \mathbf{U}^{n-1} + \frac{1}{6} \left[K_1 + (2 - \sqrt{2})K_2 + (2 + \sqrt{2})K_3 + K_4 \right], \\ K_1 &= \tau \mathbf{F}(t_{n-1}, \mathbf{U}^{n-1}), \\ K_2 &= \tau \mathbf{F} \left(t_{n-1} + \frac{\tau}{2}, \mathbf{U}^{n-1} + \frac{K_1}{2} \right), \\ K_3 &= \tau \mathbf{F} \left(t_{n-1} + \frac{\tau}{2}, \mathbf{U}^{n-1} + \frac{\sqrt{2}-1}{2}K_1 + \frac{2-\sqrt{2}}{2}K_2 \right), \\ K_4 &= \tau \mathbf{F} \left(t_{n-1} + \tau, \mathbf{U}^{n-1} - \frac{\sqrt{2}}{2}K_2 + \frac{2+\sqrt{2}}{2}K_3 \right), \end{aligned} \tag{27}$$

where \mathbf{u} , \mathbf{U}^n , $n = 1, 2, \dots, N$, are the unknown vectors and $\mathbf{F}(\cdot)$ stands for the matrix–vector system corresponding to the weighted sums in ODEs and contains $a_{ij}^{(s)}$ or $a_{im}^{(s)}$, $b_{jm}^{(s)}$, $s = 1, 2$, as its elements. Meanwhile, the boundary constraints (3), (6) must be imposed on $\mathbf{F}(\cdot)$ in the way as they are done for the fractional cases before we can fully run the procedures for Eqs. (27).

4 Stability analysis

This part makes an attempt to study the matrix stability of Eqs. (26) and the numerical stability of Eqs. (23)–(24). When $\alpha = 1$, we rewrite Eqs. (26) by

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{K}\mathbf{u} + \mathbf{Q}, \tag{28}$$

where \mathbf{Q} is a vector containing the right-hand part and the boundary conditions, and \mathbf{K} is the weighted matrix mentioned before. We discuss the homogeneous case. The numerical stability of an algorithm for the ODEs generated by a DQ method relies on the stability of the ODEs themselves. Only when their solutions are stable can a well-known method such as Runge–Kutta Gill’s method yield convergent solutions. It is enough to show their stability that the real parts of the eigenvalues of weighted matrix $-\mathbf{K}$ are all non-positive. Denote the row vector consisting of the eigenvalues of \mathbf{W}_z^c by λ_z^c , with $z = x, y$ and $c = 1, 2$. In view of the properties of Kronecker product, the eigenvalues of $\mathbf{W}_y^c \otimes \mathbf{I}_x$, $\mathbf{I}_y \otimes \mathbf{W}_x^c$ are $\lambda_y^c \otimes \mathbf{e}_x$ and $\mathbf{e}_y \otimes \lambda_x^c$ (see [22]), respectively, and therefore, we have the eigenvalues of $-\mathbf{K}$ in Eq. (28), i.e.,

$$\lambda = -\kappa_x \mathbf{e}_y \otimes \lambda_x^1 - \kappa_y \lambda_y^1 \otimes \mathbf{e}_x + \varepsilon_x \mathbf{e}_y \otimes \lambda_x^2 + \varepsilon_y \lambda_y^2 \otimes \mathbf{e}_x,$$

where \mathbf{e}_x , \mathbf{e}_y are the row vectors of sizes $M_x + 1$ and $M_y + 1$, respectively, with all of their components being 1. The exact solution of ODEs is related to λ and the condition $\text{Re}\{\lambda\} \leq 0$ is easy to meet because λ_z^2 are always verified to be real and negative while λ_z^1 be complex with their real parts being very close to zero; see Fig. 1 for example. More than that, we notice that the foregoing analysis is also valid for the 1D cases and the phenomena appearing in Fig. 1 would be enhanced as the grid numbers increase. Hence, we come to a conclusion that the ODEs are stable in most cases.

The discussion about the numerical stability of a fully discrete DQ method is difficult and still sparse [42,43]. In the sequel, we show the conditionally stable nature of Eqs. (23)–(24) in the context of L_2 -norm $\|\cdot\|$ and the analysis is not just applicable to the fractional case. Without loss of generality, consider the 2D cases and the discrete coefficients $\{\omega_k^\alpha\}_{k=0}^n$ in Eq. (12). Let $\tilde{\mathbf{U}}^0$ be the approximation of initial values \mathbf{U}^0 . Then

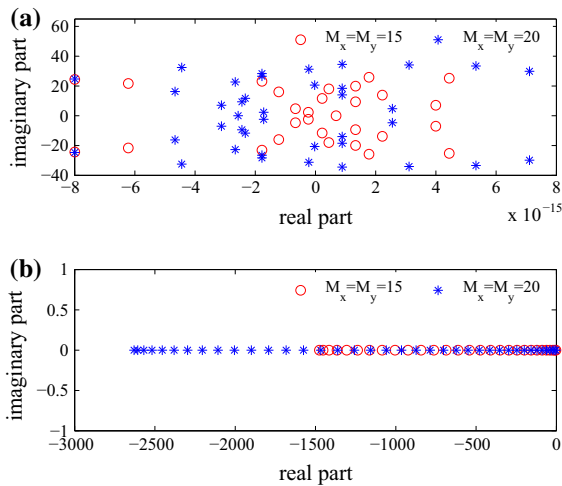


Fig. 1 The eigenvalues of the weighted matrices generated by DQ method when $a = c = 0, b = d = 2$: (a) W_z^1 ; (b) W_z^2

$$\tilde{U}^n + \tau^\alpha \mathbf{K} \tilde{U}^n = - \sum_{k=1}^{n-1} \omega_k^\alpha \tilde{U}^{n-k} + \sum_{k=0}^{n-1} \omega_k^\alpha \tilde{U}^0 + \tau^\alpha \mathbf{G}^n. \tag{29}$$

On subtracting Eq. (29) from (25) and letting $e^n = U^n - \tilde{U}^n$, we have the perturbation equation

$$e^n = - \sum_{k=1}^{n-1} \omega_k^\alpha (\mathbf{I} + \tau^\alpha \mathbf{K})^{-1} e^{n-k} + \sum_{k=0}^{n-1} \omega_k^\alpha (\mathbf{I} + \tau^\alpha \mathbf{K})^{-1} e^0, \tag{30}$$

where \mathbf{I} is the identity matrix in the same size of \mathbf{K} . To prove $\|e^n\| \leq \|e^0\|$, we make the assumption

$$\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\| \leq 1. \tag{31}$$

When $n = 1$, by taking $\|\cdot\|$ on both sides of Eq. (30), $\|e^1\| \leq \|e^0\|$ is trivial due to $\omega_0^\alpha = 1$. Let

$$\|e^m\| \leq \|e^0\|, \quad m = 1, 2, \dots, n - 1.$$

Using mathematical induction, it thus follows from the properties of $\{\omega_k^\alpha\}_{k=0}^n$ stated in Sect. 2 that

$$\begin{aligned} \|e^n\| &= \left\| - \sum_{k=1}^{n-1} \omega_k^\alpha (\mathbf{I} + \tau^\alpha \mathbf{K})^{-1} e^{n-k} + \sum_{k=0}^{n-1} \omega_k^\alpha (\mathbf{I} + \tau^\alpha \mathbf{K})^{-1} e^0 \right\| \\ &\leq \left(1 - \sum_{k=0}^{n-1} \omega_k^\alpha + \sum_{k=0}^{n-1} \omega_k^\alpha \right) \|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\| \max_{0 \leq m \leq n-1} \|e^m\| \\ &= \|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\| \max_{0 \leq m \leq n-1} \|e^m\| \leq \|e^0\|. \end{aligned}$$

Hereinafter, we proceed with a full numerical investigation on the assumption (31) to explore the potential factors which may lead to $\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\| > 1$. At first, if τ^α continuously varies from 1 to 0, there holds $\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\| \rightarrow 1$. However, this process can affect the maximal ratio of the coefficients of advection and diffusivity to keep (31); we leave this case to the end of the discussion. To be more representative, we take $\tau = 1.0 \times 10^{-3}, \alpha = 0.5$, and $M_x = M_y = 5$, unless otherwise stated. The main procedures are divided into three steps: (i) fixing $\varepsilon_x, \varepsilon_y$, and Ω , let κ_x, κ_y vary and the values of $\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\|$ as the function of κ_x, κ_y are plotted in (a), (b) of Fig. 2; (ii) fixing κ_x, κ_y , and Ω , let M_x, M_y vary and the results are plotted in (c) of Fig. 2, where $\kappa_x = \kappa_y = 500$; (iii) fixing $\kappa_x, \kappa_y, \varepsilon_x, \varepsilon_y$, let $a = c = 0$ and b, d vary, and the corresponding results are presented in (d) of Fig. 2. It is worthy to note that Ω is the unit square except the case of (iii), and the parameters of the same types in x - and y - axis are used as the same, for example, $\varepsilon_x = \varepsilon_y$. Now, we consider the influence brought by τ . Resetting $\tau = 1.0 \times 10^{-10}$, let $\varepsilon_x = \varepsilon_y = 1$ and κ_x, κ_y vary. The behavior of objective quantity is plotted in subfigure (e), from which, we see that the critical ratio between κ_x, κ_y and $\varepsilon_x, \varepsilon_y$ to maintain (31) is about 40, far less than the case of (i), and can further be improved by increasing M_x, M_y .

From the foregoing discussion and figures, we summarize the conclusions as follows: (i) if $\varepsilon_x, \varepsilon_y$ are not small, the tolerant ranges of κ_x, κ_y to guarantee (31) are quite loose and when $\varepsilon_x, \varepsilon_y \rightarrow \infty, \|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\|$ can be very close to zero; (ii) if κ_x, κ_y are larger than $\varepsilon_x, \varepsilon_y$ and $\varepsilon_x, \varepsilon_y$ themselves are small, $\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\|$ can be larger than 1; however, such issue can be remedied by increasing the grid numbers; (iii) in general, the larger M_x, M_y , the smaller $\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\|$; (iv) when the computational domain expands, $\|(\mathbf{I} + \tau^\alpha \mathbf{K})^{-1}\|$ grows at a speed, which may result in the invalidation of (31) if $\varepsilon_x, \varepsilon_y$ and M_x, M_y remain unchanged; (v) when $\tau \rightarrow 0$, the critical ratio between κ_x, κ_y and $\varepsilon_x, \varepsilon_y$ to maintain this assumption appears to decrease, but it would be enhanced as the grid is refined.

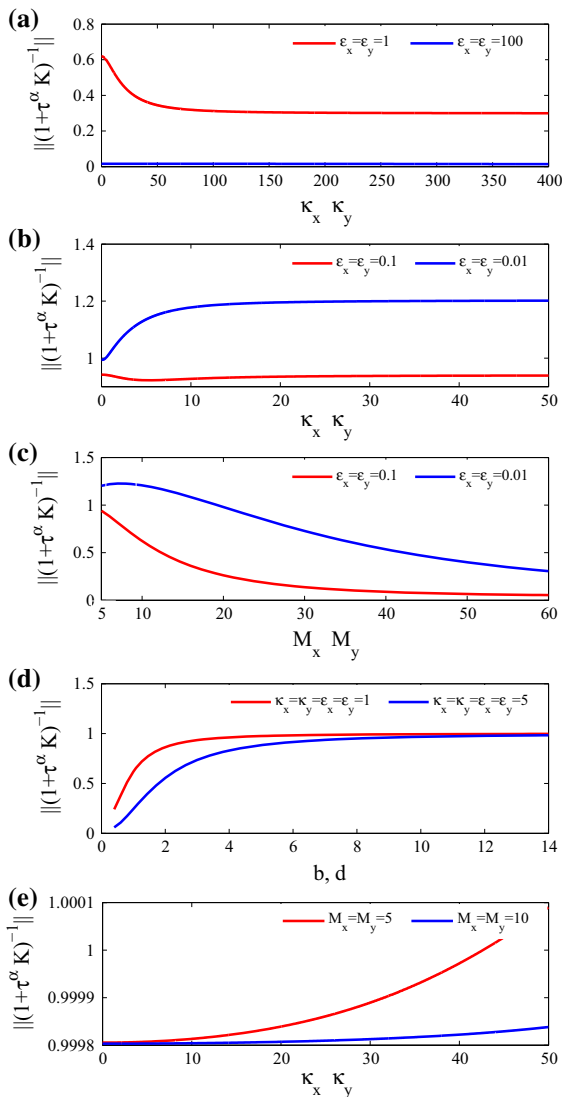


Fig. 2 The values of $\|(I + \tau^\alpha K)^{-1}\|$ versus the variation of various factors: $\kappa_x, \kappa_y, \varepsilon_x, \varepsilon_y, M_x, M_y$, and b, d

Consequently, the assumption is meaningful and essentially a mild theoretical restriction in practise.

5 Description of cubic B-spline DQ method

In this section, a robust DQ method (MCB-DQM) based on the modified cubic B-splines $\{MB_m(x)\}_{m=0}^M$

is established for Eqs. (7)–(9) by introducing the DQ approximations to fractional derivatives. In the light of the essence of traditional DQ methods, we consider

$$\frac{\partial^{\beta_1} u(x_i, y_j, t)}{\partial x^{\beta_1}} \cong \sum_{m=0}^{M_x} a_{im}^{(\beta_1)} u(x_m, y_j, t), \tag{32}$$

$$\frac{\partial^{\beta_2} u(x_i, y_j, t)}{\partial y^{\beta_2}} \cong \sum_{m=0}^{M_y} b_{jm}^{(\beta_2)} u(x_i, y_m, t), \tag{33}$$

for fractional derivatives, in constructing the DQ algorithm, where $0 \leq i \leq M_x, 0 \leq j \leq M_y$ and the weighted coefficients $a_{im}^{(\beta_1)}, b_{jm}^{(\beta_2)}$ satisfy

$$\frac{\partial^{\beta_1} MB_k(x_i)}{\partial x^{\beta_1}} = \sum_{m=0}^{M_x} a_{im}^{(\beta_1)} MB_k(x_m), \quad 0 \leq i, k \leq M_x, \tag{34}$$

$$\frac{\partial^{\beta_2} MB_k(y_j)}{\partial y^{\beta_2}} = \sum_{m=0}^{M_y} b_{jm}^{(\beta_2)} MB_k(y_m), \quad 0 \leq j, k \leq M_y. \tag{35}$$

The validation of Eqs. (32)–(33) is ensured by the linear properties of fractional derivatives. $a_{im}^{(\beta_1)}, b_{jm}^{(\beta_2)}$ are then determined by tackling the resulting algebraic problems from the above equations for each axis if the values of the fractional derivatives of B-splines $\{MB_m(x)\}_{m=0}^M$ at all sampling points are known.

5.1 The explicit formulas of fractional derivatives

It is the weakly singular integral structure that makes it difficult to calculate the values of the fractional derivatives for a function as B-spline at a sampling point. In the text that follows, we concentrate on the explicit expressions of the β -th ($1 < \beta < 2$) Riemann–Liouville derivative of B-splines $\{B_m(x)\}_{m=-1}^{M+1}$ with a recursive technique of integration by parts. Since these basis splines are piecewise and locally compact on four consecutive subintervals, we have

$${}^{RL}_{x_0} D_x^\beta B_m(x) = \begin{cases} 0, & x \in [x_0, x_{m-2}) \\ {}^{RL}_{x_{m-2}} D_x^\beta \varphi_1(x), & x \in [x_{m-2}, x_{m-1}) \\ {}^{RL}_{x_{m-2}} D_x^\beta \varphi_1(x) \\ \quad + {}^{RL}_{x_{m-1}} D_x^\beta \varphi_2(x), & x \in [x_{m-1}, x_m) \\ {}^{RL}_{x_{m-2}} D_x^\beta \varphi_1(x) \\ \quad + {}^{RL}_{x_{m-1}} D_x^\beta \varphi_2(x) \\ \quad + {}^{RL}_{x_m} D_x^\beta \varphi_3(x), & x \in [x_m, x_{m+1}) \\ {}^{RL}_{x_{m-2}} D_x^\beta \varphi_1(x) \\ \quad + {}^{RL}_{x_{m-1}} D_x^\beta \varphi_2(x) \\ \quad + {}^{RL}_{x_m} D_x^\beta \varphi_3(x) \\ \quad + {}^{RL}_{x_{m+1}} D_x^\beta \varphi_4(x), & x \in [x_{m+1}, x_{m+2}) \\ {}^{RL}_{x_{m-2}} D_x^\beta \varphi_1(x) \\ \quad + {}^{RL}_{x_{m-1}} D_x^\beta \varphi_2(x) \\ \quad + {}^{RL}_{x_m} D_x^\beta \varphi_3(x) \\ \quad + {}^{RL}_{x_{m+1}} D_x^\beta \varphi_4(x), & x \in [x_{m+2}, x_M] \end{cases}$$

with $2 \leq m \leq M - 2$. The compact supports of $B_{M-1}(x)$, $B_M(x)$, and $B_{M+1}(x)$ partially locate on the outside of $[x_0, x_M]$, so do $B_{-1}(x)$, $B_0(x)$, and $B_1(x)$; nevertheless, $B_{M-1}(x)$, $B_M(x)$, and $B_{M+1}(x)$ can be thought of as the special cases of the aforementioned argument, so are omitted here. Further, we have

$${}^{RL}_{x_0} D_x^\beta B_{-1}(x) = \begin{cases} {}^{RL}_{x_0} D_x^\beta \varphi_4(x), & x \in [x_0, x_1] \\ {}^{RL}_{x_0} D_{x_1}^\beta \varphi_4(x), & x \in [x_1, x_M] \end{cases}$$

$${}^{RL}_{x_0} D_x^\beta B_0(x) = \begin{cases} {}^{RL}_{x_0} D_x^\beta \varphi_3(x), & x \in [x_0, x_1] \\ {}^{RL}_{x_0} D_{x_1}^\beta \varphi_3(x) \\ \quad + {}^{RL}_{x_1} D_x^\beta \varphi_4(x), & x \in [x_1, x_2] \\ {}^{RL}_{x_0} D_{x_1}^\beta \varphi_3(x) \\ \quad + {}^{RL}_{x_1} D_{x_2}^\beta \varphi_4(x), & x \in [x_2, x_M] \end{cases}$$

$${}^{RL}_{x_0} D_x^\beta B_1(x) = \begin{cases} {}^{RL}_{x_0} D_x^\beta \varphi_2(x), & x \in [x_0, x_1] \\ {}^{RL}_{x_0} D_{x_1}^\beta \varphi_2(x) \\ \quad + {}^{RL}_{x_1} D_x^\beta \varphi_3(x), & x \in [x_1, x_2] \\ {}^{RL}_{x_0} D_{x_1}^\beta \varphi_2(x) \\ \quad + {}^{RL}_{x_1} D_{x_2}^\beta \varphi_3(x) \\ \quad + {}^{RL}_{x_2} D_x^\beta \varphi_4(x), & x \in [x_2, x_3] \\ {}^{RL}_{x_0} D_{x_1}^\beta \varphi_2(x) \\ \quad + {}^{RL}_{x_1} D_{x_2}^\beta \varphi_3(x) \\ \quad + {}^{RL}_{x_2} D_{x_3}^\beta \varphi_4(x), & x \in [x_3, x_M] \end{cases}$$

On the other hand, as the integrands of the integration in fractional derivatives, $\varphi_i(x)$, $i = 1, 2, 3, 4$, are cubic polynomials, for which, the order shrinks by one

each time integration by parts is applied. Being aware of this, we can eliminate the weakly singular integrations by repeating integration by parts four times for each $\varphi_i(x)$ to derive the fully explicit formulas. The derivation processes are lengthy and tedious; we therefore outline the specific expressions of B-splines ${}^{RL}_{x_0} D_x^\beta B_m(x)$, $-1 \leq m \leq M + 1$, in ‘‘Appendix’’.

5.2 Construction of cubic B-spline DQ method

Use the early notations for brevity. On using DQ approximations (32)–(33) to handle fractional derivatives, Eq. (7) is transformed into a set of first-order ODEs

$$\begin{aligned} \frac{\partial u(x_i, y_j, t)}{\partial t} - \varepsilon_x \sum_{m=0}^{M_x} a_{im}^{(\beta_1)} u(x_m, y_j, t) \\ - \varepsilon_y \sum_{m=0}^{M_y} b_{jm}^{(\beta_2)} u(x_i, y_m, t) = f(x_i, y_j, t), \end{aligned} \tag{36}$$

with $i = 0, 1, \dots, M_x$, $j = 0, 1, \dots, M_y$. Imposing the boundary constraint (9) on Eq. (36) and applying the Crank–Nicolson scheme in time, we thus obtain the following spline-based DQ scheme

$$\begin{aligned} \left\{ \begin{aligned} & U_{ij}^n - \frac{\tau \varepsilon_x}{2} \sum_{m=1}^{M_x-1} a_{im}^{(\beta_1)} U_{mj}^{n-1} - \frac{\tau \varepsilon_y}{2} \sum_{m=1}^{M_y-1} b_{jm}^{(\beta_2)} U_{im}^{n-1} \\ & = U_{ij}^{n-1} + \frac{\tau \varepsilon_x}{2} \sum_{m=1}^{M_x-1} a_{im}^{(\beta_1)} U_{mj}^{n-1} \\ & \quad + \frac{\tau \varepsilon_y}{2} \sum_{m=1}^{M_y-1} b_{jm}^{(\beta_2)} U_{im}^{n-1} + \tau f_{ij}^{n-1/2}, \end{aligned} \right. \tag{37}$$

where $i = 1, 2, \dots, M_x - 1$, $j = 1, 2, \dots, M_y - 1$. It is visible that DQ methods are truly meshless and convenient in implementation. Due to the insensitivity to dimensional changes, (37) can easily be generalized to the higher-dimensional space-fractional problems, but do not cause the rapid increase of computing burden.

6 Illustrative examples

In this section, a couple of numerical examples are carried out to gauge the practical performance of MCTB-

DQM and new MCB-DQM. In order to check their accuracy, we compute the errors by using the norms

$$e_\infty(M) \cong \max_i |u_i^n - U_i^n|,$$

$$e_2(M) \cong \sqrt{\frac{1}{M} \sum_{i=1}^{M-1} |u_i^n - U_i^n|^2},$$

$$e_N(M) \cong \sqrt{\frac{\sum_{i=1}^{M-1} |u_i^n - U_i^n|^2}{\sum_{i=1}^{M-1} |U_i^0|^2}},$$

$$e_\infty(M_x, M_y) \cong \max_{i,j} |u_{ij}^n - U_{ij}^n|,$$

$$e_2(M_x, M_y) \cong \sqrt{\frac{1}{M_x M_y} \sum_{i=1}^{M_x-1} \sum_{j=1}^{M_y-1} |u_{ij}^n - U_{ij}^n|^2},$$

where $e_N(M)$ is termed by a normalized L_2 -norm. As to $\{\omega_k^\alpha\}_{k=0}^n$ in the schemes (23)–(24), we use (12) in the first and fifth examples and (13) in the others but not the last two ones. In the computation, our algorithms are implemented on MATLAB platform in a Lenovo PC with Intel(R) Pentium(R) G2030 3.00GHz CPU and 4 GB RAM except the fourth example. The obtained results are comparatively discussed with the early works available in the open literature.

Example 6.1 Let $\kappa = 1, \varepsilon = 2$; Eqs. (1)–(3) with $\psi(x) = \exp(x), g_1(t) = E_\alpha(t^\alpha), g_2(t) = \exp(1)E_\alpha(t^\alpha)$ and homogeneous forcing term are considered on $[0, 1]$, where $E(t^\alpha)$ is the Mittag-Leffler function

$$E_\alpha(z) = \sum_{k=0}^\infty \frac{z^k}{\Gamma(\alpha k + 1)}, \quad 0 < \alpha < 1.$$

It is verified that its solution is $u(x, t) = \exp(x)E(t^\alpha)$. In order to show the convergence of MCTB-DQM, we fix $\tau = 1.0 \times 10^{-5}$ so that the temporal errors are negligible as compared to spatial errors. The numerical results at $t = 0.1$ for various α are displayed in Table 1; the convergent rate is shortly written as ‘‘Cov. rate’’. As one sees, our method is pretty stable and convergent with almost spatial second-order for this problem.

Table 1 The numerical results at $t = 0.1$ with $\tau = 1.0 \times 10^{-5}$ for Example 6.1

α	M	$e_2(M)$	Cov. rate	$e_\infty(M)$	Cov. rate
0.2	8	2.4430e-03	–	3.5200e-03	–
	16	6.3696e-04	1.9394	9.2142e-04	1.9337
	32	1.6272e-04	1.9688	2.4362e-04	1.9192
	64	4.1425e-05	1.9738	6.2649e-05	1.9592
	128	1.0765e-05	1.9441	1.5906e-05	1.9777
0.5	8	1.2489e-03	–	1.8283e-03	–
	16	3.2655e-04	1.9352	4.8198e-04	1.9235
	32	8.3679e-05	1.9644	1.2771e-04	1.9160
	64	2.1466e-05	1.9628	3.3008e-05	1.9520
	128	5.7295e-06	1.9056	8.4114e-06	1.9724
0.8	8	9.7261e-04	–	1.4452e-03	–
	16	2.5487e-04	1.9321	3.8329e-04	1.9147
	32	6.5378e-05	1.9629	1.0160e-04	1.9156
	64	1.6774e-05	1.9625	2.6330e-05	1.9481
	128	4.4723e-06	1.9072	6.7183e-06	1.9705

Example 6.2 In this test, we solve a diffusion equation on $[0, 1]$ with $\varepsilon = 1, \psi(x) = 4x(1 - x)$, zero boundary condition and right side. Its true solution has the form

$$u(x, t) = \frac{16}{\pi^3} \sum_{k=1}^\infty \frac{1}{k^3} E_\alpha(-k^2 \pi^2 t^\alpha) (1 - (-1)^k) \sin(k\pi x).$$

For comparison of the numerical results given by FDS-D I, FDS-D II [53] and the semi-discrete FEM [18], we choose the same time stepsize $\tau = 1.0 \times 10^{-4}$. Letting $\alpha = 0.1, 0.5$ and 0.95 , the corresponding results of these four methods at $t = 1$ are tabulated side by side in Table 2, from which, we conclude that MCTB-DQM is accurate and produces very small errors as the other three methods as the grid number M increases.

Example 6.3 Let $\kappa = 0, \varepsilon = 1$; we solve Eqs. (1)–(3) with homogeneous initial and boundary values, and

$$f(x, t) = \frac{2t^{2-\alpha} \sin(2\pi x)}{\Gamma(3 - \alpha)} + 4\pi^2 t^2 \sin(2\pi x),$$

on $[0, 1]$. The true solution is $u(x, t) = t^2 \sin(2\pi x)$. The algorithm is first run with $\alpha = 0.8, \tau = 2.0 \times 10^{-2}$ and $M = 50$. In Fig. 3, we plot the approximate solution and a point to point error distribution at $t =$

Table 2 A comparison of $e_N(M)$ at $t = 1$ with $\tau = 1.0 \times 10^{-4}$ for Example 6.2

α	M	FDS-D I [53]	FDS-D II [53]	FEM [18]	MCTB-DQM
0.1	8	9.98e-04	1.00e-03	5.23e-04	2.7445e-04
	16	2.44e-04	2.53e-04	1.29e-04	3.6823e-05
	32	5.36e-05	6.33e-05	3.21e-05	5.0524e-06
	64	5.89e-06	1.55e-05	8.01e-06	9.6533e-07
	128	6.08e-06	3.62e-06	2.00e-06	4.6423e-07
0.5	8	7.13e-04	7.13e-04	3.37e-04	1.9258e-04
	16	1.79e-04	1.79e-04	8.31e-04	2.6563e-05
	32	4.46e-05	4.44e-05	2.07e-05	4.5588e-06
	64	1.07e-05	1.06e-05	5.17e-06	1.7866e-06
	128	2.23e-06	2.12e-06	1.30e-06	1.4534e-06
0.95	8	1.11e-04	1.11e-04	4.84e-05	2.8976e-05
	16	2.83e-05	2.82e-05	1.21e-05	4.0937e-06
	32	7.09e-06	7.08e-06	3.05e-06	8.7766e-07
	64	1.76e-06	1.75e-06	7.93e-07	4.7682e-07
	128	4.29e-07	4.23e-07	2.32e-07	4.2766e-07

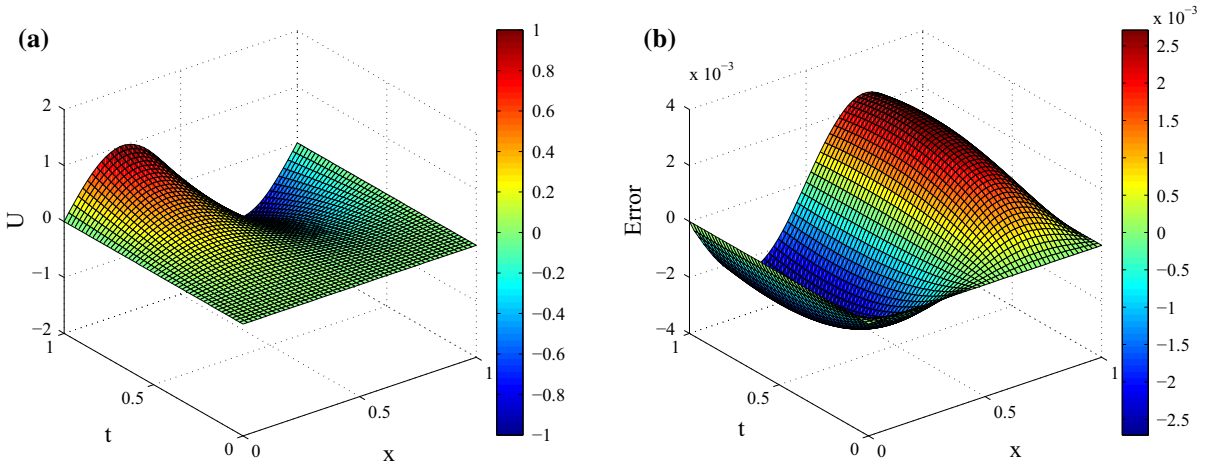


Fig. 3 The approximate solution and error distribution at $t = 1$ with $\alpha = 0.8$ for Example 6.3

1, where good accuracy is observed. In Table 3, we then report a comparison of $e_2(M)$, $e_\infty(M)$ at $t = 1$ between MCTB-DQM and CBCM [38], when $\alpha = 0.3$. Here, MCTB-DQM uses $\tau = 5.0 \times 10^{-3}$ while CBCM chooses $\tau = 1.25 \times 10^{-3}$. As expected, our approach generates the approximate solutions with a better accuracy than those obtained by CBCM.

Example 6.4 We consider a 2D diffusion equation on $[-1, 1] \times [-1, 1]$ with $\varepsilon_x = \varepsilon_y = 1$, which is referred to by Zhai and Feng as a test of a block-centered finite dif-

Table 3 A comparison of $e_2(M)$, $e_\infty(M)$ at $t = 1$ with $\alpha = 0.3$ for Example 6.3

M	CBCM [38]		MCTB-DQM	
	$e_2(M)$	$e_\infty(M)$	$e_2(M)$	$e_\infty(M)$
8	3.4134e-02	4.8273e-02	9.4300e-03	1.5762e-02
16	8.7334e-03	1.2351e-02	1.1924e-03	2.1670e-03
32	2.1955e-03	3.1048e-03	1.5040e-04	2.8541e-04
64	5.4957e-04	7.7721e-04	1.8925e-05	3.6701e-05
128	1.3739e-04	1.9430e-04	2.3752e-06	4.6559e-06

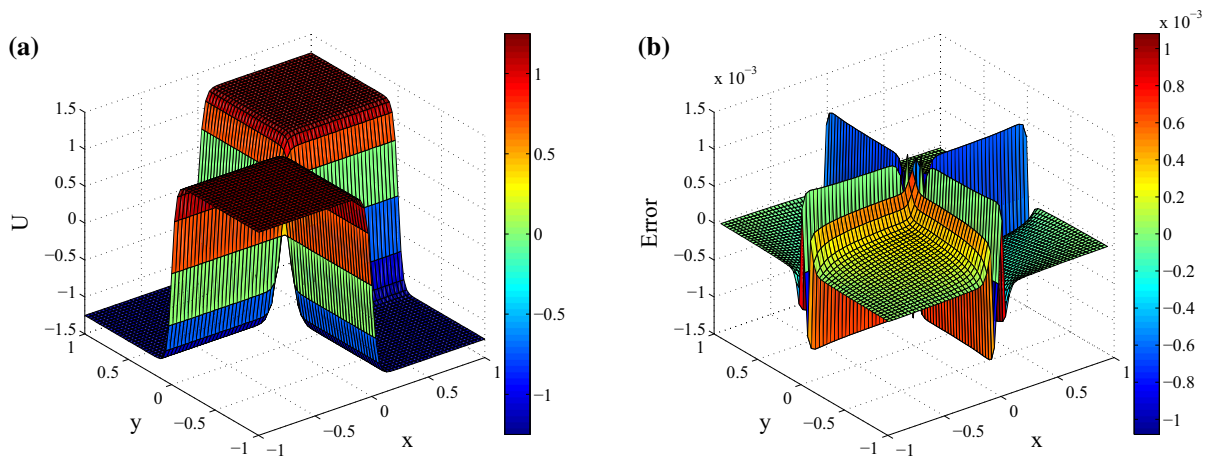


Fig. 4 The approximate solution and error distribution at $t = 0.5$ with $\alpha = 0.5$ for Example 6.4

Table 4 A comparison of $e_\infty(M_x, M_y)$ at $t = 0.5$ with $\alpha = 0.5$ for Example 6.4

M_x, M_y	BCFDM [55]		MCTB-DQM	
	$e_\infty(M_x, M_y)$	Cov. rate	$e_\infty(M_x, M_y)$	Cov. rate
12	8.75e-02	–	3.3376e-01	–
24	2.73e-02	1.68	4.6331e-03	6.1707
48	8.26e-03	1.73	3.4566e-04	3.7446
96	2.24e-03	1.88	1.8605e-05	4.2156

ference method (BCFDM) on nonuniform grids [55]. The forcing function is specified to enforce

$$u(x, y, t) = (1 + t^2) \tanh(20x) \tanh(20y).$$

Under $\tau = 1.0 \times 10^{-2}$, $M_x = M_y = 60$ and $\alpha = 0.5$, we first plot the approximate solution and a point to point error distribution at $t = 0.5$ in Fig. 4. Then, we compare MCTB-DQM and BCFDM in term of $e_\infty(M_x, M_y)$ at $t = 0.5$ in Table 4. It is obvious that MCTB-DQM produces significantly smaller errors than BCFDM as the grid number increases despite a smaller time stepsize $\tau = 2.5 \times 10^{-3}$ and the nonuniform grids BCFDM adopts; moreover, MCTB-DQM provides more than quadratic rate of convergence for this problem.

Example 6.5 In this test, we simulate the solitons propagation and collision governed by the following time-fractional nonlinear Schrödinger equation (NLS):

$$i \frac{\partial^\alpha u}{\partial t^\alpha} + \frac{\partial^2 u}{\partial x^2} + \beta |u|^2 u = 0, \quad x \in (-\infty, +\infty),$$

with $i = \sqrt{-1}$ and β being a real constant, subjected to the initial values of two Gaussian types:

(i) mobile soliton

$$\psi(x) = \text{sech}(x) \exp(2ix); \tag{38}$$

(ii) double solitons collision

$$\psi(x) = \sum_{j=1}^2 \text{sech}(x - x_j) \exp(ip(x - x_j)). \tag{39}$$

When $\alpha = 1$ and $\beta = 2$, the NLS with Eq. (38) has the soliton solution $u(x, t) = \text{sech}(x - 4t) \exp(i(2x - 3t))$. As the solutions would generally decay to zero as $|x| \rightarrow \infty$, we truncate the system into a bounded interval $\Omega = [a, b]$ with $a \ll 0$ and $b \gg 0$, and enforce periodic or homogeneous Dirichlet boundary conditions. Letting $u(x, t) = U(x, t) + iV(x, t)$. Then, the original equation can be recast as a coupled diffusion system

$$\begin{aligned} \frac{\partial^\alpha U}{\partial t^\alpha} + \frac{\partial^2 V}{\partial x^2} + \beta(U^2 + V^2)V &= 0, \\ \frac{\partial^\alpha V}{\partial t^\alpha} - \frac{\partial^2 U}{\partial x^2} - \beta(U^2 + V^2)U &= 0. \end{aligned}$$

After applying the scheme (23), nevertheless, a nonlinear system has to be solved at each time step. In such

Table 5 The numerical results in term of $e_2(M)$ at $t = 0.1$ for Example 6.5

α	Method	Real part	Imaginary part	CPU time
0.2	Scheme (23)	7.4588e-04	7.3396e-04	10.355 (s)
0.5	Scheme (23)	3.9328e-03	3.7620e-03	10.819 (s)
0.8	Scheme (23)	4.5245e-03	4.6714e-03	10.775 (s)
1.0	Scheme (23)	2.2229e-03	2.2153e-03	1.7436 (s)
	Scheme (26)	8.0954e-04	8.1843e-04	0.1521 (s)

a case, the Newton’s iteration is utilized to treat it and terminated by reaching a solution with tolerant error 1.0×10^{-12} if $\alpha = 1$, for which, the Jacobian matrix is

$$J = \begin{pmatrix} 2UV & U^2 + 3V^2 \\ -3U^2 - V^2 & -2UV \end{pmatrix}.$$

When $\alpha \neq 1$, because the analytic solutions still remain unknown and the Newton’s procedure relies heavily on its initial values, we instead employ the trust-region-dogleg algorithm built into MATLAB to improve the convergence of iteration. At first, taking $\tau = 2.0 \times 10^{-3}$, $M = 100$, $\beta = 2$, and $\Omega = [-10, 10]$, the mean square errors at $t = 0.1$ with the initial condition (38) for various α are reported in Table 5, where the solutions computed by using the coefficients (13) on a very fine time–space lattice, i.e., $\tau = 2.5 \times 10^{-4}$, $M = 400$, are adopted as reference solutions ($\alpha \neq 1$). As seen from Table 5, our methods are convergent and applicable to nonlinear coupled problems; besides, the scheme (26) is clearly more efficient than (23) since an

extra Newton’s outer loop is avoided. Then, retaking $M = 200$ and $\Omega = [-20, 20]$, we display the evolution of the amplitude of the mobile soliton created by (23) for $\alpha = 0.98$ and 1.0 in Fig. 5, respectively. Using the same discrete parameters, we consider the double solitons collision for $\alpha = 0.96$ and 1.0 with $x_1 = -6$, $x_2 = 6$, and $p = \pm 2$ in Fig. 6. It is easily drawn from these figures that the width and height of the solitons have been significantly changed by the fractional derivative. In particular, when $\alpha = 1$, a collision of double solitons without any reflection, transmission, trapping and creation of new solitary waves is exhibited, which says that it is elastic, while in fractional cases, the shapes of the solitons may not be retained after they intersect each other.

Example 6.6 In this test, we simulate an unsteady propagation of a Gaussian pulse governed by a classical 2D advection-dominated diffusion equation on a square domain $[0, 2] \times [0, 2]$ by using the scheme (26), which has been extensively studied [19, 27, 31, 45]. The Gaussian pulse solution is expressed as

$$u(x, y, t) = \frac{1}{1 + 4t} \times \exp\left(-\frac{(x - \kappa_x t - 0.5)^2}{\varepsilon_x(1 + 4t)} - \frac{(y - \kappa_y t - 0.5)^2}{\varepsilon_y(1 + 4t)}\right),$$

and the initial Gaussian pulse and boundary values are taken from the pulse solution. Letting $\kappa_x = \kappa_y = 0.8$, $\varepsilon_x = \varepsilon_y = 0.01$, we display its true solution at $t = 1.25$ with $M_x = M_y = 50$ and the used lattice points on problem domain in Fig. 7, which describe a pulse cen-

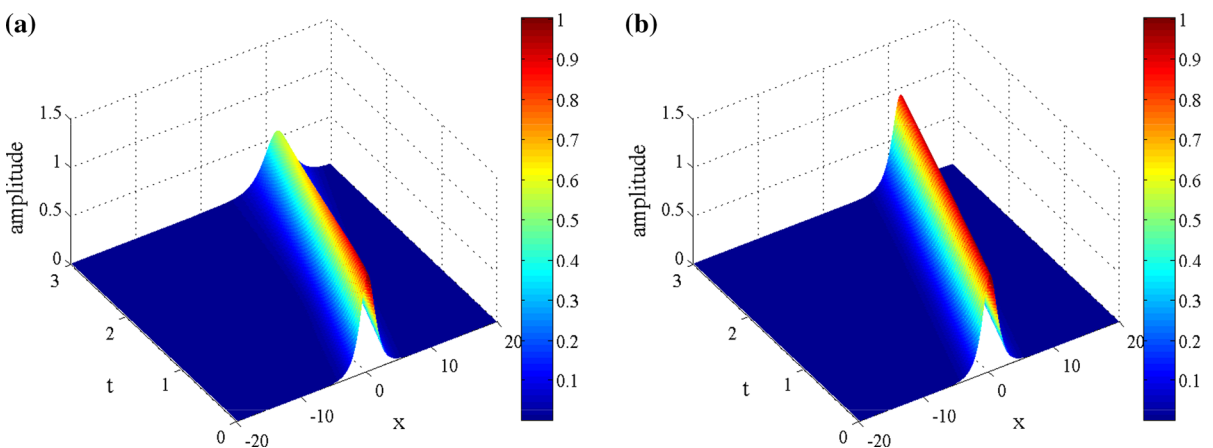


Fig. 5 The single soliton propagation for $\alpha = 0.98, 1.0$ with $\tau = 2.0 \times 10^{-3}$ and $M = 200$

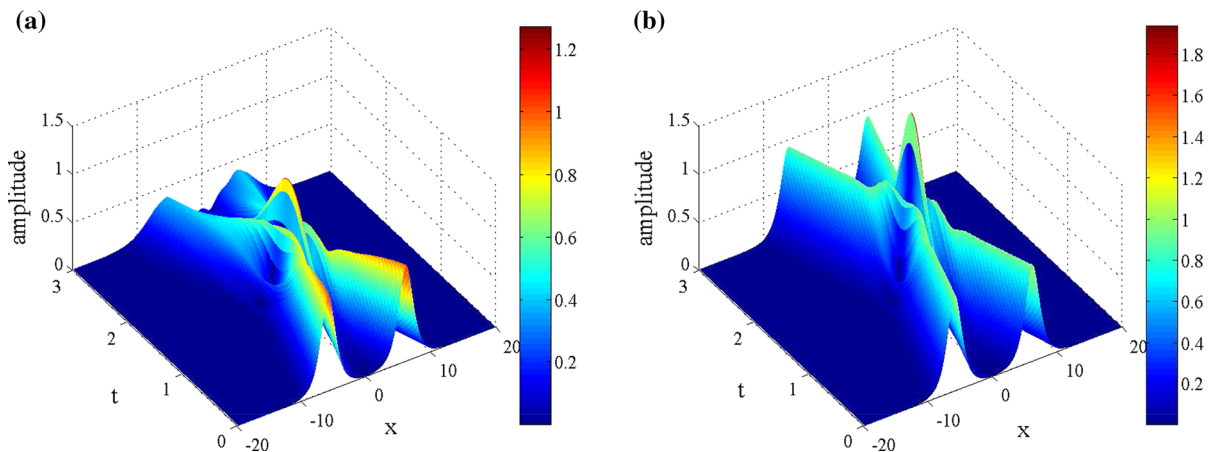


Fig. 6 The interaction of double solitons for $\alpha = 0.96, 1.0$ with $\tau = 2.0 \times 10^{-3}$ and $M = 200$

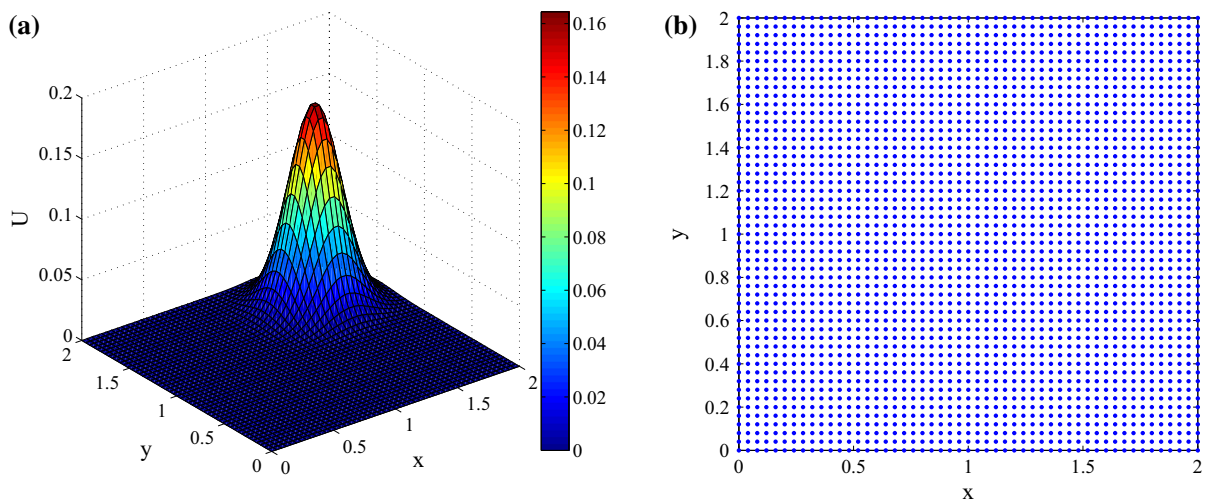


Fig. 7 The true solution at $t = 1.25$ and spatial lattice points for Example 6.6

tered at $(1.5, 1.5)$ with a pulse height of $1/6$. Using the same grid number together with $\tau = 5.0 \times 10^{-3}$, we present the contour plots of the approximate solutions at $t = 0, 0.25, 0.75, 1.25$ created by MCTB-DQM in Fig. 8. As the graph shows, the pulse is initially centered at $(0.5, 0.5)$ with a pulse height of 1, then it moves toward a position centered at $(1.5, 1.5)$; during this process, its width and height appear to be continuously varying as the time goes by. Besides, the last contour plot in Fig. 8 coincides with the true solution plotted in Fig. 7. Retaking $\tau = 6.25 \times 10^{-3}$ and $M_x = M_y = 80$, we compare our results with those obtained by some previous algorithms as nine-point high-order compact

(HOC) schemes [19,31], Peaceman–Rachford ADI scheme (PR-ADI) [32], HOC-ADI scheme [20], exponential HOC-ADI scheme (EHOC-ADI) [46], HOC boundary value method (HOC-BVM) [7], compact integrated RBF ADI method (CIRBF-ADI) [45], coupled compact integrated RBF ADI method (CCIRBF-ADI) [47], and the Galerkin FEM combined with the method of characteristics (CGFEM) [8], at $t = 1.25$ in Table 6. We implement CGFEM on a quasi-uniform triangular mesh with the meshsize 2.5×10^{-2} by using both Lagrangian P1 and P2 elements. Also, average absolute errors are added as supplements to evaluate and compare their accuracy. As seen from Table 6,

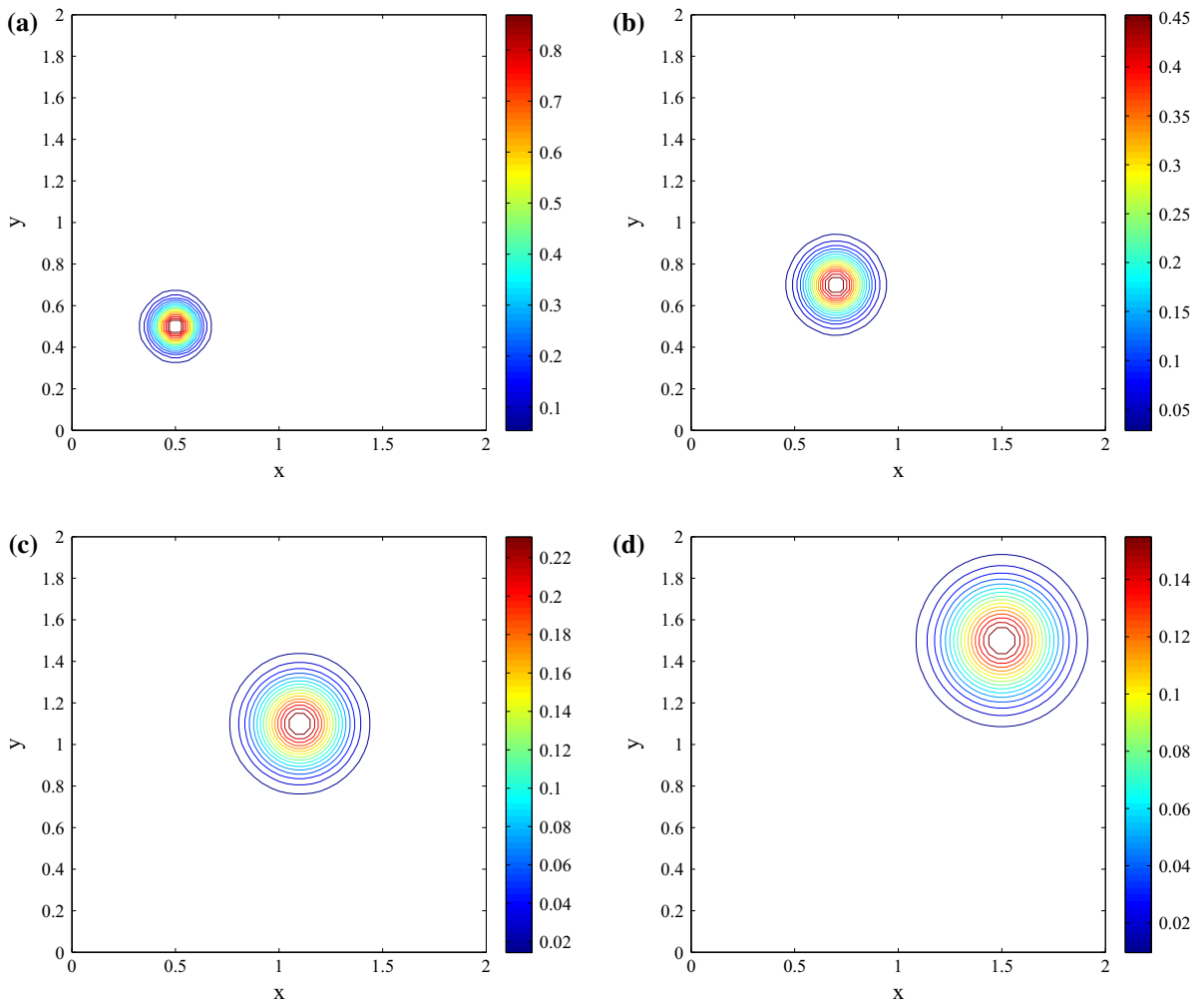


Fig. 8 The contour plots of Gaussian pulse at $t = 0, 0.25, 0.75, 1.25$ with $\tau = 5.0 \times 10^{-3}$ and $M_x = M_y = 50$

all of these methods are illustrated to be very accurate to capture the Gaussian pulse except the PR-ADI scheme; besides, our method reaches a better accuracy than the others and even shows promise in treating the advection–diffusion equations in the high Péclet number regime.

Example 6.7 In the last test, we consider the 2D space-fractional Eqs. (7)–(9) on $[0, 1] \times [0, 1]$ with $\varepsilon_x = \varepsilon_y = 1$, $\psi(x, y) = x^2(1 - x)^2y^2(1 - y)^2$, and homogeneous boundary values. The source term is manufactured as

$$\begin{aligned}
 f(x, y, t) = & -e^{-t}x^2(1 - x)^2y^2(1 - y)^2 \\
 & - \frac{2e^{-t}x^{2-\beta_1}y^2(1 - y)^2}{\Gamma(3 - \beta_1)} \left(1 - \frac{6x}{3 - \beta_1} + \frac{12x^2}{(3 - \beta_1)(4 - \beta_1)} \right) \\
 & - \frac{2e^{-t}x^2(1 - x)^2y^{2-\beta_2}}{\Gamma(3 - \beta_2)} \left(1 - \frac{6y}{3 - \beta_2} + \frac{12y^2}{(3 - \beta_2)(4 - \beta_2)} \right)
 \end{aligned}$$

to enforce the analytic solution $u(x, y, t) = e^{-t}x^2(1 - x)^2y^2(1 - y)^2$. Letting $\beta_1 = 1.1$, $\beta_2 = 1.3$, and $\tau = 2.5 \times 10^{-4}$, we solve the problem via the FEM proposed by [52] and MCB-DQM, and compare their numerical results at $t = 0.2$ in Table 7, where the P1

Table 6 A comparison of global errors at $t = 1.25$ with $\tau = 6.25 \times 10^{-3}$ and $M_x = M_y = 80$ for Example 6.6.

Method	Average error	$e_2(M_x, M_y)$	$e_\infty(M_x, M_y)$
CIRBF-ADI [45]	6.742e-06	2.197e-05	1.703e-04
CCIRBF-ADI [47]	5.989e-06	1.904e-05	1.427e-04
Noye and Tan [31]	1.971e-05	1.280e-04	6.509e-04
Kalita et al. [19]	1.597e-05	1.024e-04	4.477e-04
PR-ADI [32]	3.109e-04	2.025e-03	7.778e-03
HOC-ADI [20]	9.218e-06	5.931e-05	2.500e-04
EHOC-ADI [46]	9.663e-06	6.194e-05	2.664e-04
HOC-BVM [7]	9.4931e-06	–	2.4766e-04
CGFEM P1	6.3746e-05	1.8849e-04	1.5731e-03
CGFEM P2	1.5667e-05	5.4061e-05	5.8044e-04
MCTB-DQM	9.1512e-07	5.6996e-06	2.2830e-05

element and structured meshes are adopted. The data indicate that DQ method converges toward the analytic solution as the grid numbers increase and admits slightly better results than FEM. More importantly, the implemental CPU times of MCB-DQM are far less than those of FEM, which confirm its high computing efficiency.

7 Conclusion

The ADEs are the subjects of active interest in mathematical physics and the related areas of research. In this work, we have proposed an effective DQ method for

such equations involving the derivatives of fractional orders in time and space. Its weighted coefficients are calculated by making use of modified CTBs and cubic B-splines as test functions. The stability of DQ method for the time-fractional ADEs in the context of L_2 -norm is performed. The theoretical condition required for the stable analysis is numerically surveyed at length. We test the codes on several benchmark problems and the outcomes have demonstrated that it outperforms some of the previously reported algorithms such as BCFDM and FEM in term of overall accuracy and efficiency.

In a linear space, spanned by a set of proper basis functions as B-splines, any function can be represented by a weighted combination of these basis functions. While all basis functions are defined, the function remains unknown because the coefficients on the front of basis functions are still unknown. However, when all basis functions satisfy Eqs. (34)–(35), by virtue of linearity, it can be examined that the objective function satisfies Eqs. (34)–(35) as well. This is the essence of DQ methods, which guarantees their convergence.

Despite the error bounds are difficult to determine, the numerical results illustrate that the spline-based DQ method admits the convergent results for the fractional ADEs. The presented approach can be generalized to the higher-dimensional and other complex model problems arising in material science, structural and fluid mechanics, heat conduction, biomedicine, differential dynamics, and so forth. High computing efficiency, low memory requirement, and the ease of programming are its main advantages.

Table 7 A comparison of global errors at $t = 0.2$ with $\tau = 2.5 \times 10^{-4}$, $\beta_1 = 1.1$, and $\beta_2 = 1.3$ for Example 6.7

Method	M_x, M_y	$e_2(M_x, M_y)$	Cov. rate	$e_\infty(M_x, M_y)$	Cov. rate	CPU time
FEM [52]	10	6.5781e-05	–	1.5632e-04	–	63.750(s)
	15	3.0082e-05	1.9296	7.2433e-05	1.8972	199.37(s)
	20	1.7376e-05	1.9079	4.1321e-05	1.9511	494.98(s)
	25	1.1404e-05	1.8872	2.6599e-05	1.9741	1062.7(s)
MCB-DQM	10	5.4217e-05	–	1.4763e-04	–	0.3726(s)
	15	2.6606e-05	1.7556	6.9553e-05	1.8562	1.1785(s)
	20	1.5559e-05	1.8650	4.0088e-05	1.9153	4.0598(s)
	25	1.0207e-05	1.8892	2.6163e-05	1.9124	11.267(s)

Acknowledgements The authors are very grateful to the reviewers for their valuable comments and suggestions. This research was supported by National Natural Science Foundations of China (Nos. 11471262 and 11501450).

Appendix: The explicit formulas of the fractional derivatives of cubic B-splines

The fractional derivatives center at x_{-1} , x_0 , and x_1 :

$$\begin{aligned}
 {}^{\text{RL}}_{x_0} D_x^\beta B_{-1}(x) &= \begin{cases} \frac{(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} - \frac{3(x-x_0)^{1-\beta}}{\Gamma(2-\beta)h} + \frac{6(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} - \frac{6(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_0, x_1) \\ \frac{(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} - \frac{3(x-x_0)^{1-\beta}}{\Gamma(2-\beta)h} + \frac{6(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} - \frac{6(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{6(x-x_1)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_1, x_M] \end{cases} \\
 {}^{\text{RL}}_{x_0} D_x^\beta B_0(x) &= \begin{cases} \frac{4(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} - \frac{12(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} + \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_0, x_1) \\ \frac{4(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} - \frac{12(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} + \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_1)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_1, x_2) \\ \frac{4(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} - \frac{12(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} + \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_1)^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{6(x-x_2)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_2, x_M] \end{cases} \\
 {}^{\text{RL}}_{x_0} D_x^\beta B_1(x) &= \begin{cases} \frac{(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} + \frac{3(x-x_0)^{1-\beta}}{\Gamma(2-\beta)h} + \frac{6(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} - \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_0, x_1) \\ \frac{(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} + \frac{3(x-x_0)^{1-\beta}}{\Gamma(2-\beta)h} + \frac{6(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} - \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_1)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_1, x_2) \\ \frac{(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} + \frac{3(x-x_0)^{1-\beta}}{\Gamma(2-\beta)h} + \frac{6(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} - \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_1)^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_2)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_2, x_3) \\ \frac{(1-\beta)(x-x_0)^{-\beta}}{\Gamma(2-\beta)} + \frac{3(x-x_0)^{1-\beta}}{\Gamma(2-\beta)h} + \frac{6(x-x_0)^{2-\beta}}{\Gamma(3-\beta)h^2} - \frac{18(x-x_0)^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_1)^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_2)^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{6(x-x_3)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_3, x_M] \end{cases}
 \end{aligned}$$

The fractional derivatives center at x_m with $2 \leq m \leq M + 1$:

$${}^{\text{RL}}_{x_0} D_x^\beta B_m(x) = \begin{cases} 0, & x \in [x_0, x_{m-2}) \\ \frac{6(x-x_{m-2})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{m-2}, x_{m-1}) \\ \frac{6(x-x_{m-2})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{m-1})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{m-1}, x_m) \\ \frac{6(x-x_{m-2})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{m-1})^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_m)^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_m, x_{m+1}) \\ \frac{6(x-x_{m-2})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{m-1})^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_m)^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{m+1})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{m+1}, x_{m+2}) \\ \frac{6(x-x_{m-2})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{m-1})^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_m)^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{m+1})^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{6(x-x_{m+2})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{m+2}, x_M] \end{cases}$$

which contain ${}^{\text{RL}}_{x_0} D_x^\beta B_{M-1}(x)$, ${}^{\text{RL}}_{x_0} D_x^\beta B_M(x)$, and ${}^{\text{RL}}_{x_0} D_x^\beta B_{M+1}(x)$ as special cases:

$$\begin{aligned}
 {}^{\text{RL}}_{x_0} D_x^\beta B_{M-1}(x) &= \begin{cases} 0, & x \in [x_0, x_{M-3}) \\ \frac{6(x-x_{M-3})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{M-3}, x_{M-2}) \\ \frac{6(x-x_{M-3})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{M-2})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{M-2}, x_{M-1}) \\ \frac{6(x-x_{M-3})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{M-2})^{3-\beta}}{\Gamma(4-\beta)h^3} + \frac{36(x-x_{M-1})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{M-1}, x_M] \end{cases} \\
 {}^{\text{RL}}_{x_0} D_x^\beta B_M(x) &= \begin{cases} 0, & x \in [x_0, x_{M-2}) \\ \frac{6(x-x_{M-2})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{M-2}, x_{M-1}) \\ \frac{6(x-x_{M-2})^{3-\beta}}{\Gamma(4-\beta)h^3} - \frac{24(x-x_{M-1})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{M-1}, x_M] \end{cases} \\
 {}^{\text{RL}}_{x_0} D_x^\beta B_{M+1}(x) &= \begin{cases} 0, & x \in [x_0, x_{M-1}) \\ \frac{6(x-x_{M-1})^{3-\beta}}{\Gamma(4-\beta)h^3}, & x \in [x_{M-1}, x_M] \end{cases}
 \end{aligned}$$

References

1. Abbas, M., Majid, A.A., Ismail, A.I.M., Rashid, A.: The application of cubic trigonometric B-spline to the numerical solution of the hyperbolic problems. *Appl. Math. Comput.* **239**, 74–88 (2014)
2. Bellman, R., Casti, J.: Differential quadrature and long-term integration. *J. Math. Anal. Appl.* **34**(2), 235–238 (1971)
3. Bhrawy, A., Zaky, M.: A fractional-order Jacobi Tau method for a class of time-fractional PDEs with variable coefficients. *Math. Method Appl. Sci.* **39**(7), 1765–1779 (2016)
4. Chen, M.H., Deng, W.H.: Fourth order difference approximations for space Riemann–Liouville derivatives based on weighted and shifted Lubich difference operators. *Commun. Comput. Phys.* **16**(2), 516–540 (2014)
5. Chen, Y.M., Wu, Y.B., Cui, Y.H., Wang, Z.Z., Jin, D.M.: Wavelet method for a class of fractional convection–diffusion equation with variable coefficients. *J. Comput. Sci.* **1**(3), 146–149 (2010)
6. Cui, M.R.: A high-order compact exponential scheme for the fractional convection–diffusion equation. *J. Comput. Appl. Math.* **255**, 404–416 (2014)
7. Dehghan, M., Mohebbi, A.: High-order compact boundary value method for the solution of unsteady convection–diffusion problems. *Math. Comput. Simulat.* **79**(3), 683–699 (2008)
8. Douglas Jr., J., Russell, T.F.: Numerical methods for convection–dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.* **19**(5), 871–885 (1982)
9. Esmaeili, S., Garrappa, R.: A pseudo-spectral scheme for the approximate solution of a time-fractional diffusion equation. *Int. J. Comput. Math.* **92**(5), 980–994 (2015)
10. Gao, G.H., Sun, H.W.: Three-point combined compact alternating direction implicit difference schemes for two-dimensional time-fractional advection–diffusion equations. *Commun. Comput. Phys.* **17**(2), 487–509 (2015)
11. Heydari, M.H.: Wavelets Galerkin method for the fractional subdiffusion equation. *J. Comput. Nonlinear Dyn.* **11**(6), 061014 (2016)
12. Hosseini, S.M., Ghaffari, R.: Polynomial and nonpolynomial spline methods for fractional sub-diffusion equations. *Appl. Math. Model.* **38**(14), 3554–3566 (2014)
13. Huang, C.B., Yu, X.J., Wang, C., Li, Z.Z., An, N.: A numerical method based on fully discrete direct discontinuous Galerkin method for the time fractional diffusion equation. *Appl. Math. Comput.* **264**, 483–492 (2015)
14. Izadkhah, M.M., Saberi-Nadjafi, J.: Gegenbauer spectral method for time-fractional convection–diffusion equations with variable coefficients. *Math. Method Appl. Sci.* **38**(15), 3183–3194 (2015)
15. Jia, J.H., Wang, H.: A fast finite volume method for conservative space-fractional diffusion equations in convex domains. *J. Comput. Phys.* **310**, 63–84 (2016)
16. Ji, C.C., Sun, Z.Z.: A high-order compact finite difference scheme for the fractional sub-diffusion equation. *J. Sci. Comput.* **64**(3), 959–985 (2015)
17. Jiang, Y.J., Ma, J.T.: High-order finite element methods for time-fractional partial differential equations. *J. Comput. Appl. Math.* **235**(11), 3285–3290 (2011)
18. Jin, B.T., Lazarov, R., Zhou, Z.: Error estimates for a semidiscrete finite element method for fractional order parabolic equations. *SIAM J. Numer. Anal.* **51**(1), 445–466 (2013)
19. Kalita, J.C., Dalal, D.C., Dass, A.K.: A class of higher order compact schemes for the unsteady two-dimensional convection–diffusion equation with variable convection coefficients. *Int. J. Numer. Methods Fluids* **38**(12), 1111–1131 (2002)
20. Karaa, S., Zhang, J.: High order ADI method for solving unsteady convection–diffusion problems. *J. Comput. Phys.* **198**(1), 1–9 (2004)
21. Kilbas, A.A., Srivastava, H.M., Trujillo, J.J.: *Theory and Applications of Fractional Differential Equations*. Elsevier, Amsterdam (2006)
22. Laub, A.J.: *Matrix Analysis For Scientists & Engineers*. Society for Industrial and Applied Mathematics, Philadelphia (2005)
23. Lin, Y.M., Xu, C.J.: Finite difference/spectral approximations for the time-fractional diffusion equation. *J. Comput. Phys.* **225**(2), 1533–1552 (2007)
24. Liu, Q., Gu, Y.T., Zhuang, P., Liu, F., Nie, Y.F.: An implicit RBF meshless approach for time fractional diffusion equations. *Comput. Mech.* **48**(1), 1–12 (2011)
25. Luo, W.H., Huang, T.Z., Wu, G.C., Gu, X.M.: Quadratic spline collocation method for the time fractional subdiffusion equation. *Appl. Math. Comput.* **276**, 252–265 (2016)
26. Mittal, R.C., Jain, R.K.: Numerical solutions of nonlinear Burgers’ equation with modified cubic B-splines collocation method. *Appl. Math. Comput.* **218**(15), 7839–7855 (2012)
27. Ma, Y.B., Sun, C.P., Haake, D.A., Churchill, B.M., Ho, C.M.: A high-order alternating direction implicit method for the unsteady convection-dominated diffusion problem. *Int. J. Numer. Methods Fluids* **70**(6), 703–712 (2012)
28. Meerschaert, M.M., Scheffler, H.P., Tadjeran, C.: Finite difference methods for two-dimensional fractional dispersion equation. *J. Comput. Phys.* **211**(1), 249–261 (2006)
29. Mainardi, F.: *Fractals and Fractional Calculus Continuum Mechanics*. Springer, Berlin (1997)
30. Nazir, T., Abbas, M., Ismail, A.I.M., Majid, A.A., Rashid, A.: The numerical solution of advection–diffusion problems using new cubic trigonometric B-splines approach. *Appl. Math. Model.* **40**(7–8), 4586–4611 (2016)
31. Noye, B.J., Tan, H.H.: Finite difference methods for solving the two-dimensional advection–diffusion equation. *Int. J. Numer. Methods Fluids* **9**(1), 75–98 (1989)
32. Peaceman, D.W., Rachford Jr., H.H.: The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math.* **3**(1), 28–41 (1955)
33. Pirkhedri, A., Javadi, H.H.S.: Solving the time-fractional diffusion equation via Sinc–Haar collocation method. *Appl. Math. Comput.* **257**, 317–326 (2015)
34. Podlubny, I.: *Fractional Differential Equations*. Academic Press, San Diego (1999)
35. Razminia, K., Razminia, A., Baleanu, D.: Investigation of the fractional diffusion equation based on generalized integral quadrature technique. *Appl. Math. Model.* **39**(1), 86–98 (2015)

36. Roop, J.P.: Computational aspects of FEM approximation of fractional advection dispersion equations on bounded domains in \mathbb{R}^2 . *J. Comput. Appl. Math.* **193**(1), 243–268 (2006)
37. Saadatmandi, A., Dehghan, M., Azizi, M.R.: The Sinc–Legendre collocation method for a class of fractional convection–diffusion equations with variable coefficients. *Commun. Nonlinear Sci. Numer. Simul.* **17**(11), 4125–4136 (2012)
38. Sayevand, K., Yazdani, A., Arjang, F.: Cubic B-spline collocation method and its application for anomalous fractional diffusion equations in transport dynamic systems. *J. Vib. Control* **22**(9), 2173–2186 (2016)
39. Shirzadi, A., Ling, L., Abbasbandy, S.: Meshless simulations of the two-dimensional fractional-time convection–diffusion–reaction equations. *Eng. Anal. Bound. Elem.* **36**(11), 1522–1527 (2012)
40. Shlesinger, M.F., West, B.J., Klafter, J.: Lévy dynamics of enhanced diffusion: application to turbulence. *Phys. Rev. Lett.* **58**(11), 1100–1103 (1987)
41. Shu, C., Richards, B.E.: Application of generalized differential quadrature to solve two-dimensional incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids* **15**(7), 791–798 (1992)
42. Tomasiello, S.: Stability and accuracy of the iterative differential quadrature method. *Int. J. Numer. Methods Eng.* **58**(9), 1277–1296 (2003)
43. Tomasiello, S.: Numerical stability of DQ solutions of wave problems. *Numer. Algor.* **57**(3), 289–312 (2011)
44. Tian, W.Y., Zhou, H., Deng, W.H.: A class of second order difference approximations for solving space fractional diffusion equations. *Math. Comput.* **84**(294), 1703–1727 (2015)
45. Thai-Quang, N., Mai-Duy, N., Tran, C.-D., Tran-Cong, T.: High-order alternating direction implicit method based on compact integrated-RBF approximations for unsteady/steady convection–diffusion equations. *CMES Comput. Model. Eng. Sci.* **89**(3), 189–220 (2012)
46. Tian, Z.F., Ge, Y.B.: A fourth-order compact ADI method for solving two-dimensional unsteady convection–diffusion problems. *J. Comput. Appl. Math.* **198**(1), 268–286 (2007)
47. Tien, C.M.T., Thai-Quang, N., Mai-Duy, N., Tran, C.-D., Tran-Cong, T.: A three-point coupled compact integrated RBF scheme for second-order differential problems. *CMES Comput. Model. Eng. Sci.* **104**(6), 425–469 (2015)
48. Uddin, M., Haq, S.: RBFs approximation method for time fractional partial differential equations. *Commun. Nonlinear Sci. Numer. Simul.* **16**(11), 4208–4214 (2011)
49. Walz, G.: Identities for trigonometric B-splines with an application to curve design. *BIT Numer. Math.* **37**(1), 189–201 (1997)
50. Yang, X.H., Zhang, H.X., Xu, D.: Orthogonal spline collocation method for the two-dimensional fractional subdiffusion equation. *J. Comput. Phys.* **256**, 824–837 (2014)
51. Zaslavsky, G.M., Stevens, D., Weitzner, H.: Self-similar transport in incomplete chaos. *Phys. Rev. E.* **48**(3), 1683–1694 (1993)
52. Zhu, X.G., Nie, Y.F., Wang, J.G., Yuan, Z.B.: A numerical approach for the Riesz space-fractional Fisher’ equation in two-dimensions. *Int. J. Comput. Math.* **94**(2), 296–315 (2017)
53. Zeng, F.H., Li, C.P., Liu, F.W., Turner, I.: The use of finite difference/element approaches for solving the time-fractional subdiffusion equation. *SIAM J. Sci. Comput.* **35**(6), A2976–A3000 (2013)
54. Zeng, F.H., Liu, F.W., Li, C.P., Burrage, K., Turner, I., Anh, V.: A Crank–Nicolson ADI spectral method for a two-dimensional Riesz space fractional nonlinear reaction–diffusion equation. *SIAM J. Numer. Anal.* **52**(6), 2599–2622 (2014)
55. Zhai, S.Y., Feng, X.L.: A block-centered finite-difference method for the time-fractional diffusion equation on nonuniform grids. *Numer. Heat Transf. Part B Fundam.* **69**(3), 217–233 (2016)
56. Zhou, F.Y., Xu, X.Y.: The third kind Chebyshev wavelets collocation method for solving the time-fractional convection diffusion equations with variable coefficients. *Appl. Math. Comput.* **280**, 11–29 (2016)
57. Zhuang, P.H., Liu, F.W.: Implicit difference approximation for the time fractional diffusion equation. *J. Appl. Math. Comput.* **22**(3), 87–99 (2006)