Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2012.m1107

Vol. **5**, No. 3, pp. 333-358
August 2012

High-Order Accurate Runge-Kutta (Local) Discontinuous Galerkin Methods for One- and Two-Dimensional Fractional Diffusion Equations

Xia Ji[1,*] and Huazhong Tang[2]

[1] *LSEC, Institute of Computational Mathematics, Chinese Academy of Sciences, P. O. Box 2719, Beijing 100190, China.*
[2] *HEDPS, CAPT and LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, China.*

Received 2 April 2011; Accepted (in revised version) 6 June 2011

Available online 3 July 2012

Abstract. As the generalization of the integer order partial differential equations (PDE), the fractional order PDEs are drawing more and more attention for their applications in fluid flow, finance and other areas. This paper presents high-order accurate Runge-Kutta local discontinuous Galerkin (DG) methods for one- and two-dimensional fractional diffusion equations containing derivatives of fractional order in space. The Caputo derivative is chosen as the representation of spatial derivative, because it may represent the fractional derivative by an integral operator. Some numerical examples show that the convergence orders of the proposed local P^k–DG methods are $O(h^{k+1})$ both in one and two dimensions, where P^k denotes the space of the real-valued polynomials with degree at most k.

AMS subject classifications: 35R11, 65M60, 65M12

Key words: Discontinuous Galerkin method, Runge-Kutta time discretization, fractional derivative, Caputo derivative, diffusion equation.

1. Introduction

Fractional calculus is a natural extension of the integer order calculus [28, 30]. Recently many problems in physics [2], finance [31] and hydrology [1] have been formulated on fractional partial differential equations (PDE), containing derivatives of fractional order in space, time or both. For example, anomalous diffusion is a possible mechanism underlying plasma transport in magnetically confined plasmas, and the fractional order space derivative operators can be used to model such transport mechanism.

In recent years the numerical solutions of the fractional PDEs have attracted a considerable interest both in mathematics and in applications. An intrinsical difference between

*Corresponding author. *Email addresses:* jixia@lsec.cc.ac.cn (X. Ji), hztang@pku.edu.cn (H. Tang)

the behaviors of integer and fractional order derivatives is that the integer order derivatives depend only on the local behavior of a function or solution, while the fractional derivatives are non-local, i.e., they depend on the entire function or solution. Thus, new difficulties and challenges appear in deriving numerical methods for this kind of equations.

The fractional derivatives of order $\alpha > 0$, $\frac{\partial^\alpha u}{\partial x^\alpha}$, are usually represented by the Riemann-Liouville formula [28,30]

$$\frac{\partial^\alpha u}{\partial x^\alpha}(x,t) = \frac{1}{\Gamma(n-\alpha)}\frac{\partial^n}{\partial x^n}\int_a^x u(\xi,t)(x-\xi)^{n-\alpha-1}\,d\xi, \tag{1.1}$$

where $\Gamma(\cdot)$ is the Gamma function, $x \in [a,b]$, $-\infty \le a < b \le \infty$, $n-1 < \alpha < n$, $n \in \mathbb{Z}^+$. The fractional derivatives are also frequently defined by the Grünwald-Letnikov formula

$$\frac{\partial^\alpha u}{\partial x^\alpha}(x,t) = \lim_{\Delta x \to 0}\frac{1}{\Delta x^\alpha}\sum_{\ell=0}^{[\frac{x-a}{\Delta x}]}(-1)^\ell \begin{pmatrix} \alpha \\ \ell \end{pmatrix} u(x-\ell\Delta x, t), \tag{1.2}$$

where $[\frac{x-a}{\Delta x}]$ denotes the integer part of $\frac{x-a}{\Delta x}$. If $u(\xi,\cdot) \in C^n[a,x]$, the Riemann-Liouville formula is equivalent to the Grünwald-Letnikov. However, the discrete approximations of the latter present some limitations: frequently numerical approximations based on this formula originate unstable numerical methods and henceforth in many cases a shifted Grünwald-Letnikov formula is used; the order of accuracy of such approaches is never higher than one.

Another way to represent the fractional derivative is by the Caputo formula

$$\frac{\partial^\alpha u}{\partial x^\alpha} = \frac{1}{\Gamma(n-\alpha)}\int_a^x \frac{\partial^n u(\xi,t)}{\partial \xi^n}(x-\xi)^{n-\alpha-1}d\xi. \tag{1.3}$$

This formula has some advantages over the Riemann-Liouville formula. The Laplace transform method is very frequently used for solving fractional differential equations, the Laplace transform of the Riemann-Liouville derivatives leads to boundary conditions involving the limit value of the Riemann-Liouville derivatives at the lower terminal $x = a$. Although technically such problems can be solved, there is no physical interpretation. On the other hand, the Laplace transform of the Caputo derivative imposes boundary conditions involving integer order derivatives which usually are more acceptable and physical. Another advantage is that the Caputo derivative of a constant is zero, while for the Riemann-Liouville it's not.

During the past decade, numerical methods of the fractional PDEs have been increasingly appearing in literatures. Lynch et al. [24] studied the numerical properties of the PDEs of fractional order $\alpha \in (1,2)$. Shen and Liu [35] gave error analysis of an explicit finite difference approximation for the space fractional diffusion equation with insulated ends. Chen et al. [3] proved the stability and convergence of an implicit difference approximation scheme of the fractional diffusion equation describing anomalous slow diffusion (sub-diffusion) by using a Fourier method. Liu et al. [21] discussed stability and convergence of the difference methods for the space-time fractional advection-diffusion equation.

Shen et al. [36] studied the fundamental solution and the finite difference approximations of the Riesz factional advection-dispersion equation. Cui [11] developed a compact finite difference method for the fractional diffusion equation by using the Grünwald-Letnikov discretization of the Riemann-Liouville derivative. Yuste [43] developed weighted average finite difference methods for fractional diffusion equations.

Most of the existing finite difference methods developed for the PDEs of fractional order $\alpha \in (1,2)$ are only first order accurate and restricted to one-dimensional problem. A second order accurate finite difference method for one- and two-dimensional fractional diffusion equations can be found in [39, 40], where the Richardson extrapolation is applied to achieve second-order accuracy. Xu and his coworkers [19, 20] considered spectral approximations for the time fractional diffusion equation, this equation can be used to describe the anomalous sub-diffusion. Deng [13] studied finite element methods for the space and time fractional Fokker-Planck equation. In the time fractional equation, the solution at a time t_k depends on the solutions at all previous time levels $t < t_k$. This makes the storage very expensive and challenges the algorithm design. There are several ways to discretize the time fractional derivative and speed its computation [12, 19, 23], McLean et al. studied the convergence analysis of a discontinuous Galerkin method for the time fractional differential equation, the non-uniform time steps are used in their methods due to the singularity of derivatives at $t = 0$ [25, 26].

The aim of this paper is to develope high-order accurate local discontinuous Galerkin (DG) approximations of a fractional diffusion model containing derivatives of fractional order in space. The DG methods discussed here are a class of the finite element methods, which adopt completely discontinuous piecewise polynomial space for the numerical solutions and the test functions in the spatial variables, coupled with explicit and nonlinearly stable high order Runge-Kutta time discretization. The original DG was introduced in 1973 by Reed and Hill to solve the neutron transport equation [33], which was first analyzed by Lesaint and Raviart [17] in 1974. Since then, it has been widely used in the numerical simulation of elliptic equation [32], Dirac equation [38], convection-diffusion equation [34], shallow water equation [18], MHD equation [42], the time fractional diffusion equation [25], etc. It was first developed for hyperbolic conservation laws containing first derivatives by Cockburn et al. in a series of papers [5–8]. The natural features of the the Runge-Kutta DG methods are their formal high order accuracy, their nonlinear stability, their ability to capture the discontinuous or strong gradients of the exact solution without producing spurious oscillations, and their excellent parallel efficiency. For a detailed description of the methods, we refer the readers to several substantial review papers and lecture notes [4, 9] as well as the paper [22, 41] on the extensions and applications of the DG methods.

The paper is organized as follows. Section 2 develops numerical schemes for one-dimensional fractional differential equation. Their two-dimensional extensions are given in Section 3. Section 4 conducts some numerical experiments to demonstrate the accuracy and capability of the present methods. Several concluding remarks are given in Section 5.

2. Numerical schemes in one dimension

In one-dimensional case, we consider the fractional diffusion equation

$$\frac{\partial u}{\partial t} = d(x)\frac{\partial^\alpha u}{\partial x^\alpha} + p(x,t), \quad 0 < x < L, \tag{2.1}$$

subject to the following initial condition and (Dirichlet) boundary conditions

$$u(x,0) = u_0(x),\ u(0,t) = u_L(t), \quad u(L,t) = u_R(t), \quad 0 < x < L, \quad t \geq 0, \tag{2.2}$$

with $1 < \alpha < 2$ and $d(x) \geq 0$. This model is often used to describe the super-diffusion [37]. An example of the super-diffusion is a continuous random walk with steps obeying a Levy distribution [29]. Models for anomalous slow diffusion (sub-diffusion) can be found in [27]. The super-diffusion is a form of diffusion in which the random walk of the molecules contains occasional very long steps, while the sub-diffusion is the tendency of particles in a fluid not to diffuse due to random trapping.

This section will construct high-order local DG spatial discretizations for (2.1) on a finite interval $[0, L]$ with the Caputo representation (1.3) for $1 < \alpha < 2$, i.e.

$$\frac{\partial^\alpha u}{\partial x^\alpha} = \frac{1}{\Gamma(2-\alpha)}\int_0^x \frac{\partial^2 u(\xi,t)}{\partial x^2}(x-\xi)^{1-\alpha} d\xi =: \mathscr{K}\left[\frac{\partial^2 u(x,t)}{\partial x^2}\right], \tag{2.3}$$

and two auxiliary variables $q(x,t)$ and $g(x,t)$, which are defined by

$$q(x,t) := \frac{\partial u(x,t)}{\partial x}, \tag{2.4}$$

$$g(x,t) := \frac{\partial q(x,t)}{\partial x} \equiv \frac{\partial^2 u(x,t)}{\partial x^2}. \tag{2.5}$$

Using the auxiliary variables $q(x,t)$ and $g(x,t)$, (2.1) may be rewritten as follows

$$\frac{\partial u}{\partial t} = d(x)\mathscr{K}[g(x,t)] + p(x,t). \tag{2.6}$$

Multiplying (2.4)-(2.6) by the test functions v_u, v_q, v_g, respectively, integrating them over the finite element I_j, which is defined in the following text, and using a simple formal integration by parts, we can get

$$\int_{I_j} \frac{\partial u}{\partial t} v_u \, dx = \int_{I_j} d(x)\mathscr{K}[g] v_u \, dx + \int_{I_j} p v_u \, dx, \quad t \geq 0, \tag{2.7}$$

$$\int_{I_j} q v_q \, dx = -\int_{I_j} u\frac{dv_q}{dx} \, dx + u(x_j,t)v_q(x_j) - u(x_{j-1},t)v_q(x_{j-1}), \quad t \geq 0, \tag{2.8}$$

$$\int_{I_j} g v_g \, dx = -\int_{I_j} g\frac{dv_y}{dx} \, dx + q(x_j,t)v_g(x_j) - q(x_{j-1},t)v_g(x_{j-1}), \quad t \geq 0. \tag{2.9}$$

The initial condition in (2.2) becomes

$$\int_{I_j} u(x,0)v_u\,dx = \int_{I_j} u_0(x)v_u\,dx. \tag{2.10}$$

Those are the weak formulations of (2.4)-(2.6) as well as (2.2), in which will be used to construct corresponding local DG approximations of the problem (2.1)-(2.2).

2.1. Local DG spatial discretization

This subsection starts to present high-order accurate local DG methods approximating the initial-boundary value problem (2.1)-(2.2). Divide the domain $\Omega = [0,L]$ into N elements arbitrarily, i.e. $\Omega = \bigcup_{j=1}^{N} I_j$, where the jth element is $I_j = [x_{j-1}, x_j]$, $j = 1, \cdots, N$, and $0 = x_0 < \cdots < x_{j-1} < x_j \cdots < x_N = L$. Our computations will be restricted to the uniform partition of the domain for the sake of simplicity, Δx is the step size.

Following the idea of Cockburn and Shu [10], define the finite dimensional space

$$\mathscr{V}_h^k = \left\{ v \in \mathscr{L}^2(0,L) \middle|\ v(x) \in P^k(I_j),\ \text{if } x \in I_j,\ j = 1, \cdots, N \right\}, \tag{2.11}$$

where $P^k(I_j)$ denotes the space of the real-valued polynomials of degree at most k over the element I_j.

Replacing the exact solutions $(u(x,t), q(x,t), g(x,t))$ by their approximations

$$(u_h(x,t), q_h(x,t), g_h(x,t)) \in (\mathscr{V}_h^k)^3$$

in (2.7)-(2.10), and approximating the fluxes on the boundary of the element I_j, $\{u(x_j,t)$, $u(x_{j-1},t)$, $q(x_j,t)$, $q(x_{j-1},t)\}$, by the corresponding numerical fluxes, $\{\hat{h}_{q,j}$, $\hat{h}_{q,j-1}$, $\hat{h}_{g,j}$, $\hat{h}_{g,j-1}\}$, respectively, in (2.8)-(2.9), then we have the following local DG formulations for the problem (2.1)-(2.2): find $u_h \in \mathscr{V}_h^k, q_h \in \mathscr{V}_h^k, g_h \in \mathscr{V}_h^k$ such that for all $I_j, j = 1, \cdots, N$ we have

$$\int_{I_j} \frac{\partial u_h}{\partial t} v_{u,h}\,dx = \int_{I_j} d(x)\mathscr{K}[g_h]v_{u,h}dx + \int_{I_j} p v_{u,h}\,dx, \tag{2.12}$$

$$\int_{I_j} q_h v_{q,h}\,dx = -\int_{I_j} u_h \frac{dv_{q,h}}{dx}\,dx + \hat{h}_{q,j} v_{q,h}(x_j^-) - \hat{h}_{q,j-1} v_{q,h}(x_{j-1}^+), \tag{2.13}$$

$$\int_{I_j} g_h v_{g,h}\,dx = -\int_{I_j} q_h \frac{dv_{g,h}}{dx}\,dx + \hat{h}_{g,j} v_{g,h}(x_j^-) - \hat{h}_{g,j-1} v_{g,h}(x_{j-1}^+), \tag{2.14}$$

$$\int_{I_j} u_h(x,0)v_{u,h}\,dx = \int_{I_j} u_0(x)v_{u,h}\,dx, \tag{2.15}$$

for any $v_{u,h} \in \mathscr{V}_h^k$, $v_{q,h} \in \mathscr{V}_h^k$, $v_{g,h} \in \mathscr{V}_h^k$. Here $\int_{I_j} d(x)\mathscr{K}[g_h]v_{u,h}\,dx$ is defined as

$$\begin{aligned}\int_{I_j} d(x)\mathscr{K}[g_h]v_{u,h}\,dx =& \frac{1}{\Gamma(2-\alpha)}\int_{I_j}[v_{u,h}d(x)\int_0^x g_h(\xi,t)(x-\xi)^{1-\alpha}\,d\xi]\,dx\\ =& \frac{1}{\Gamma(2-\alpha)}\int_{I_j} v_{u,h}d(x)\Big\{[\sum_{\ell=1}^{j-1}\int_{I_\ell} g_h(\xi,t)(x-\xi)^{1-\alpha}\,d\xi\\ &+\int_{x_{j-1}}^x g_h(\xi,t)(x-\xi)^{1-\alpha}\,d\xi]\Big\}\,dx. \end{aligned} \tag{2.16}$$

Since the approximate solution $\{u,q,g\}_h(x_j,t)$ is discontinuous at the points x_j, $j = 0,\cdots,N$, we must choose $\{\hat{h}_{q,j},\hat{h}_{g,j}\}$ carefully in order to derive a stable scheme. The choice of the numerical fluxes is quite delicate as it can affect the stability and accuracy of the method, as well as the sparsity and symmetry of the stiffness matrix [10]. Usually $\{\hat{h}_{q,j},\hat{h}_{g,j}\}$ is taken as a two-point numerical flux, which may be represented in the following general form

$$\begin{aligned}\widehat{h}_{q,j} :=& \widehat{h}_q\big(u_h(x_j-0,t),u_h(x_j+0,t)\big), \quad \widehat{h}_q(u,u)=u,\\ \widehat{h}_{g,j} :=& \widehat{h}_g\big(q_h(x_j-0,t),q_h(x_j+0,t)\big), \quad \widehat{h}_g(q,q)=q.\end{aligned}$$

Inspired by the mixed formulation for the heat equation, we use the "alternating principle" [44] to choose the numerical fluxes in (2.13)-(2.14)

$$\hat{h}_{q,j} = u_h(x_j+0,t), \quad \hat{h}_{g,j} = q_h(x_j-0,t), \tag{2.17}$$

or

$$\hat{h}_{q,j} = u_h(x_j-0,t), \quad \hat{h}_{g,j} = q_h(x_j+0,t), \tag{2.18}$$

on the interior boundaries, i.e., $j = 1,2,\cdots,N-1$. On the domain boundaries $x = 0$ or $x = L$, the numerical fluxes are taken as

$$\hat{h}_{q,0} = u_L(t), \quad \hat{h}_{q,N} = u_R(t), \quad \hat{h}_{g,0/N} = q_h(x = 0/L,t). \tag{2.19}$$

For any $x \in [0,L]$ and $t \in [0,T]$, we now express our approximation solutions (u_h,q_h,g_h) as follows

$$u_h(x,t) = \sum_{\ell=1}^{k+1} u_j^{(\ell)}(t)\phi_j^{(\ell)}(x), \quad q_h(x,t) = \sum_{\ell=1}^{k+1} q_j^{(\ell)}(t)\phi_j^{(\ell)}(x), \tag{2.20a}$$

$$g_h(x,t) = \sum_{\ell=1}^{k+1} g_j^{(\ell)}(t)\phi_j^{(\ell)}(x), \quad x \in I_j, \tag{2.20b}$$

where $\{\phi_j^{(\ell)}(x), \ell = 1,\cdots,k+1\}$ is a basis of the local polynomial space $P^k(I_j)$ for $x \in I_j$. We also define the local mass matrix $\mathbf{M}_j$, the local stiff matrix $\mathbf{M}_j^x$, the convolution related

matrix $\mathbf{M}^{j,i}$, the source related vector $\vec{S}_j$ and the initial value related vector $\vec{U}_j^0$, whose elements are defined as

$$(\mathbf{M}_j)_{m,\ell} = \int_{I_j} \phi_j^{(\ell)} \phi_j^{(m)}\, dx, \quad (\mathbf{M}_j^x)_{m,\ell} = \int_{I_j} \phi_j^{(\ell)} \frac{d\phi_j^{(m)}}{dx}\, dx, \tag{2.21a}$$

$$(\vec{S}_j)_m = \int_{I_j} p(x,t)\phi_j^{(m)}\, dx, \quad (\mathbf{M}^{j,i})_{m,\ell} = \int_{I_j} d(x)\phi_j^{(m)} \mathscr{K}[\phi_i^{(\ell)}]\, dx, \tag{2.21b}$$

$$(\vec{U}_j^0)_m = \int_{I_j} u_0(x)\phi_j^{(m)}\, dx, \tag{2.21c}$$

and introduce the following local vector notations

$$\vec{u}_j = (u_j^{(1)}, \cdots, u_j^{(k+1)})^T, \qquad \vec{q}_j = (q_j^{(1)}, \cdots, q_j^{(k+1)})^T, \tag{2.22a}$$

$$\vec{g}_j = (g_j^{(1)}, \cdots, g_j^{(k+1)})^T, \qquad \vec{\phi}_{j,R} = \left(\phi_j^{(1)}(1), \cdots, \phi_j^{(k+1)}(1)\right)^T, \tag{2.22b}$$

$$\vec{\phi}_{j,L} = \left(\phi_j^{(1)}(-1), \cdots, \phi_j^{(k+1)}(-1)\right)^T. \tag{2.22c}$$

By using the projections in (2.20), the notations in (2.21)-(2.22), the numerical fluxes e.g. (2.17), and taking the test functions $v_{u,h}, v_{q,h}, v_{g,h}$ as the basis functions $\phi_j^{(\ell)}(x)$, $\ell = 1, 2, \cdots, k+1$, the semi-discrete local DG scheme (2.12)-(2.17) can be recasted into the following local matrix-vector forms.

The numerical schemes for $\vec{q}_j$ are

$$\mathbf{M}_j\vec{q}_j = -\mathbf{M}_j^x\vec{u}_j + \vec{\phi}_{j,R}\vec{\phi}_{j,L}^T\vec{u}_{j+1} - \vec{\phi}_{j,L}u_L, \qquad j = 1, \tag{2.23a}$$

$$\mathbf{M}_j\vec{q}_j = \left(-\mathbf{M}_j^x - \vec{\phi}_{j,L}\vec{\phi}_{j,L}^T\right)\vec{u}_j + \vec{\phi}_{j,R}\vec{\phi}_{j,L}^T\vec{u}_{j+1}, \qquad j = 2, \cdots, N-1, \tag{2.23b}$$

$$\mathbf{M}_j\vec{q}_j = \left(-\mathbf{M}_j^x - \vec{\phi}_{j,L}\vec{\phi}_{j,L}^T\right)\vec{u}_j + \vec{\phi}_{j,R}u_R, \qquad j = N. \tag{2.23c}$$

For $\vec{g}_j$, we have

$$\mathbf{M}_j\vec{g}_j = \left(-\mathbf{M}_j^x - \vec{\phi}_{j,L}\vec{\phi}_{j,L}^T + \vec{\phi}_{j,R}\vec{\phi}_{j,R}^T\right)\vec{q}_j, \qquad j = 1, \tag{2.24a}$$

$$\mathbf{M}_j\vec{g}_j = (-\mathbf{M}_j^x + \vec{\phi}_{j,R}\vec{\phi}_{j,R}^T)\vec{q}_j - \vec{\phi}_{j,L}\vec{\phi}_{j,R}^T\vec{q}_{j-1}, \qquad j = 2, \cdots, N, \tag{2.24b}$$

and the numerical schemes for $\vec{u}_j$ are

$$\mathbf{M}_j\frac{d\vec{u}_j}{dt} = \sum_{i=1}^{j}\mathbf{M}^{j,i}\vec{g}_i + \vec{S}_j, \quad j = 1, \cdots, N. \tag{2.25}$$

Similarly, (2.15) becomes

$$\mathbf{M}_j\vec{u}_j(0) = \vec{U}_j^0. \tag{2.26}$$

Remark 2.1 In practice, we will choose the Legendre polynomials $L_\ell(\xi)$ as basis functions of the space $P^k(I_j)$, i.e.,

$$\phi_j^{(\ell)}(x) := L_{\ell-1}(\xi_j(x)), \quad \xi_j(x) = \frac{2(x-x_{j-1})}{\Delta x} - 1, \quad \ell = 1, \cdots, k+1, \quad x \in I_j, \quad (2.27)$$

and then we may exploit sufficiently their L^2–orthogonality:

$$\int_{-1}^{1} L_m(\xi) L_\ell(\xi)\, d\xi = \frac{2}{2m+1}\delta_{m,\ell}, \quad m \le \ell, \quad (2.28)$$

to obtain a diagonal mass matrix so that we may construct an explicit local DG method approximating the fractional diffusion equation (2.1). In this case, let $x = x_{i-1} + \hat{x}(x_j - x_{j-1})$, then $\mathscr{K}[\phi_i^{(\ell)}](x)$ in the matrix element $(\mathbf{M}^{j,i})_{m,\ell}$ can be represented as

$$\mathscr{K}[\phi_i^{(\ell)}](x) = \begin{cases} 0, & j < i, \\ \dfrac{\Delta x^{2-\alpha}}{\Gamma(2-\alpha)} \displaystyle\int_0^1 (\hat{x}-s)^{1-\alpha} L_{\ell-1}(2s-1)\, ds, & j > i, \\ \dfrac{\Delta x^{2-\alpha}}{\Gamma(2-\alpha)} \displaystyle\int_0^{\hat{x}} (\hat{x}-s)^{1-\alpha} L_{\ell-1}(2s-1)\, ds, & j = i, \end{cases} \quad (2.29)$$

whose analytical formula can be gotten by using the symbolic toolbox of *Matlab*. We give the expressions for $\alpha = 1.2$ in Appendix I.

2.2. Time discretizations

To discretize the ODE system (2.25) with the initial data (2.26), we use the high-order Runge-Kutta time discretizations. As shown by Cockburn in [4], when the polynomials of degree k, a higher-order accurate Runge-Kutta method must be used in order to guarantee that the scheme is stable. In this paper we use a fourth-order non-TVD Runge-Kutta scheme [15]. Numerical experiments demonstrate its numerical stability.

The ODE system (2.25) is first rewritten into

$$\frac{d}{dt}\vec{U}(t) = \vec{F}(t, \vec{U}),$$

then the fourth-order accurate non-TVD Runge-Kutta scheme is given as

$$\vec{k}_1 = \vec{F}(t_n, \vec{U}_n), \quad (2.30a)$$

$$\vec{k}_2 = \vec{F}(t_n + \frac{\Delta t}{2}, \vec{U}_n + \frac{\Delta t}{2}\vec{k}_1), \quad (2.30b)$$

$$\vec{k}_3 = \vec{F}(t_n + \frac{\Delta t}{2}, \vec{U}_n + \frac{\Delta t}{2}\vec{k}_2), \quad (2.30c)$$

$$\vec{k}_4 = \vec{F}(t_n + \Delta t, \vec{U}_n + \Delta t\vec{k}_3), \quad (2.30d)$$

$$\vec{U}^{(n+1)} = \vec{U}^n + \frac{\Delta t}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4), \quad (2.30e)$$

where Δt is the time step size, $\vec{U}^n$ is an approximation of the solution vector $\vec{U}$ at time $t = t_n$.

3. Numerical schemes in two dimensions

This section extends the above high-order accurate local DG methods to the two-dimensional fractional diffusion equation

$$\frac{\partial u}{\partial t} = d_1(x,y)\frac{\partial^\alpha u}{\partial x^\alpha} + d_2(x,y)\frac{\partial^\beta u}{\partial y^\beta} + p(x,y,t), \tag{3.1}$$

on a finite rectangular domain $\Omega = \{(x,y)|\ 0 < x < L_x, 0 < y < L_y\}$, where $1 < \alpha, \beta < 2$, $d_1(x,y) \geq 0$, and $d_2(x,y) \geq 0$. The initial condition and the boundary conditions are specified as

$$u(x,y,0) = u_0(x,y), \quad u(0,y,t) = u_W(y,t), \quad u(L_x,y,t) = u_E(y,t), \tag{3.2a}$$
$$u(x,0,t) = u_S(x,t), \quad u(x,L_y,t) = u_N(x,t). \tag{3.2b}$$

To obtain the local DG methods, we will still employ the Caputo representation (1.3) of the fractional derivatives in x- and y-directions in (3.1), i.e.

$$\frac{\partial^\alpha u}{\partial x^\alpha} = \frac{1}{\Gamma(2-\alpha)}\int_0^x \frac{\partial^2 u(\xi,y,t)}{\partial x^2}(x-\xi)^{1-\alpha}\,d\xi =: \mathscr{K}^x\left[\frac{\partial^2 u(x,y,t)}{\partial x^2}\right], \tag{3.3}$$
$$\frac{\partial^\beta u}{\partial y^\beta} = \frac{1}{\Gamma(2-\beta)}\int_0^y \frac{\partial^2 u(x,\eta,t)}{\partial y^2}(y-\eta)^{1-\beta}\,d\eta =: \mathscr{K}^y\left[\frac{\partial^2 u(x,y,t)}{\partial y^2}\right], \tag{3.4}$$

and introduce four auxiliary variables $q^x(x,t)$, $q^y(x,t)$, $g^x(x,t)$ and $g^y(x,t)$ as follows

$$q^x(x,t) := \frac{\partial u(x,y,t)}{\partial x}, \quad g^x(x,t) := \frac{\partial q^x(x,y,t)}{\partial x} = \frac{\partial^2 u(x,y,t)}{\partial x^2}, \tag{3.5}$$
$$q^y(x,t) := \frac{\partial u(x,y,t)}{\partial y}, \quad g^y(x,t) := \frac{\partial q^y(x,y,t)}{\partial y} = \frac{\partial^2 u(x,y,t)}{\partial y^2}, \tag{3.6}$$

so that the fractional diffusion equation (3.1) can be rewritten as

$$\frac{\partial u}{\partial t} = d_1(x,y)\mathscr{K}^x\left[g^x(x,y,t)\right] + d_2(x,y)\mathscr{K}^y\left[g^y(x,y,t)\right] + p(x,y,t). \tag{3.7}$$

Now we start to present high-order accurate local DG methods approximating (3.1) from (3.5)-(3.7). Divide the domain $\Omega = [0,L_x]\times[0,L_y]$ into $N_x \times N_y$ elements uniformly, and define $I_{i,j} = [(i-1)\Delta x, i\Delta x]\times[(j-1)\Delta y, j\Delta y]$, where Δx and Δy are step sizes in x- and y-directions. Define the finite element space as

$$\mathscr{V}_h = \{v \in L^2([0,L_x]\times[0,L_y])\big|\ v|_{I_{i,j}} \in P(I_{i,j}), i = 1,\cdots,N_x, j = 1,\cdots,N_y\}, \tag{3.8}$$

where $P(I_{i,j})$ is any finite dimensional space of the two-dimensional smooth functions (e.g. polynomials for our local DG methods) defined on the element $I_{i,j}$.

Similarly, multiplying (3.5)-(3.7) by the test function $v_u, v_{q^x}, v_{q^y}, v_{g^x}, v_{g^y}$, respectively, and integrating by parts over the element $I_{i,j}$, we get

$$\int_{I_{i,j}} \frac{\partial u}{\partial t} v_u \, dxdy = \int_{I_{i,j}} \left(d_1(x,y)\mathscr{K}^x[g^x] + d_2(x,y)\mathscr{K}^y[g^y]\right) v_u \, dxdy + \int_{I_{i,j}} p v_u \, dxdy, \tag{3.9}$$

$$\int_{I_{i,j}} q^x v_{q^x} \, dxdy = -\int_{I_{i,j}} u \frac{\partial v_{q^x}}{\partial x} \, dxdy + \int_{\partial I_{i,j}} u v_{q^x} n_x \, ds, \tag{3.10}$$

$$\int_{I_{i,j}} g^x v_{g^x} \, dxdy = -\int_{I_{i,j}} q^x \frac{\partial v_{g^x}}{\partial x} \, dxdy + \int_{\partial I_{i,j}} q^x v_{g^x} n_x \, ds, \tag{3.11}$$

$$\int_{I_{i,j}} q^y v_{q^y} \, dxdy = -\int_{I_{i,j}} u \frac{\partial v_{q^y}}{\partial y} \, dxdy + \int_{\partial I_{i,j}} u v_{q^y} n_y \, ds, \tag{3.12}$$

$$\int_{I_{i,j}} g^y v_{g^y} \, dxdy = -\int_{I_{i,j}} q^y \frac{\partial v_{g^y}}{\partial y} \, dxdy + \int_{\partial I_{i,j}} q^y v_{g^y} n_y \, ds, \tag{3.13}$$

where $\partial I_{i,j}$ denotes the boundary of the element $I_{i,j}$, and (n_x, n_y) is the outward unit normal vector on the element boundary $\partial I_{i,j}$. The initial condition in (3.2) becomes

$$\int_{I_{i,j}} u(x,y,0) v_u \, dxdy = \int_{I_{i,j}} u_0(x,y) v_u \, dxdy. \tag{3.14}$$

These equations form the weak formulations of the initial-boundary value problem (3.1)-(3.2) which we shall use to define the local DG methods.

Following the same procedure in one-dimensional case, the two-dimensional semi-discrete local DG methods are given as follows: Find $u_h \in \mathscr{V}_h$, $q_h^x \in \mathscr{V}_h$, $g_h^x \in \mathscr{V}_h$, $q_h^y \in \mathscr{V}_h$, and $g_h^y \in \mathscr{V}_h$ such that for all $I_{i,j}, i = 1, \cdots, N_x, j = 1, \cdots, N_y$, we have

$$\int_{I_{i,j}} \frac{\partial u_h}{\partial t} v_{u,h} \, dxdy = \int_{I_{i,j}} \left(d_1(x,y)\mathscr{K}^x[g_h^x] + d_2(x,y)\mathscr{K}^y[g_h^y]\right) v_{u,h} \, dxdy + \int_{I_{i,j}} p v_{u,h} \, dxdy, \tag{3.15}$$

$$\int_{I_{i,j}} q_h^x v_{q^x,h} \, dxdy = -\int_{I_{i,j}} u_h \frac{\partial v_{q^x,h}}{\partial x} \, dxdy + \int_{\partial I_{i,j}} \hat{h}_q^x v_{q^x,h} n_x \, ds, \tag{3.16}$$

$$\int_{I_{i,j}} g_h^x v_{g^x,h} \, dxdy = -\int_{I_{i,j}} q_h^x \frac{\partial v_{g^x,h}}{\partial x} \, dxdy + \int_{\partial I_{i,j}} \hat{h}_g^x v_{g^x,h} n_x \, ds, \tag{3.17}$$

$$\int_{I_{i,j}} q_h^y v_{q^y,h}\,dxdy = -\int_{I_{i,j}} u_h \frac{\partial v_{q^y,h}}{\partial y}\,dxdy + \int_{\partial I_{i,j}} \hat{h}_q^y u_{q^y} n_y\,ds, \tag{3.18}$$

$$\int_{I_{i,j}} g_h^y v_{g^y,h}\,dxdy = -\int_{I_{i,j}} q_h^y \frac{\partial v_{g^y,h}}{\partial y}\,dxdy + \int_{\partial I_{i,j}} \hat{h}_g^y v_{g^y} n_y\,ds, \tag{3.19}$$

$$\int_{I_{i,j}} u_h(x,y,0) v_{u,h}\,dxdy = \int_{I_{i,j}} u_0(x,y) v_{u,h}\,dxdy. \tag{3.20}$$

For numerical computations, the second integrations at the right-hand sides of (3.16)-(3.19) are calculated by the higher-order accurate Gaussian quadrature rule, and the numerical fluxes $\hat{h}_q^x$, $\hat{h}_g^x$, $\hat{h}_q^y$ and $\hat{h}_g^y$ in (3.16)-(3.19) are appropriate approximations to u, q^x and q^y at the points of the element boundary $\partial I_{i,j}$ corresponding to the nodes of gaussian quadrature. In our computations, we take them as

$$\hat{h}_q^x = \overline{u_h} + \frac{\langle u_h\rangle}{2} n_x,\ \hat{h}_q^y = \overline{u_h} + \frac{\langle u_h\rangle}{2} n_y,\ \hat{h}_g^x = \overline{q_h^x} - \frac{\langle q_h^x\rangle}{2} n_x,\ hath_g^y = \overline{q_h^y} - \frac{\langle q_h^y\rangle}{2} n_y, \tag{3.21}$$

for the interior boundaries of the element, and

$$\hat{h}_q^x = u_B, \quad \hat{h}_q^y = u_B, \quad \hat{h}_g^x = q_h^x, \quad \hat{h}_g^y = q_h^y, \tag{3.22}$$

for those element boundaries in accordance with part of the domain boundary, where $\overline{a} := \frac{a^{\text{out}}+a^{\text{in}}}{2}$, $\langle a\rangle := a^{\text{out}} - a^{\text{in}}$, a^{in} and a^{out} are the limits taken from the interior and exterior of the element $I_{i,j}$ along the outward unit normal vector (n_x, n_y) of $\partial I_{i,j}$, respectively, and u_B denotes the node value of u_W, u_E, u_S, or u_N, which are given in (3.2).

Suppose $\{\phi_{i,j}^{(\ell)}(x,y), \ell = 1,2,\cdots,k\}$ are the basis functions of the local function space $P(I_{i,j})$, then we may represent the approximate solutions $u_h, q_h^x, g_h^x, q_h^y, g_h^y$ as

$$\begin{aligned}
u_h(x,y,t) &= \sum_{\ell=1}^{k} u_{i,j}^{(\ell)}(t)\phi_{i,j}^{(\ell)}(x,y), & q_h^x(x,y,t) &= \sum_{\ell=1}^{k} q_{i,j}^{x,(\ell)}(t)\phi_{i,j}^{(\ell)}(x,y),\\
g_h^x(x,y,t) &= \sum_{\ell=1}^{k} g_{i,j}^{x,(\ell)}(t)\phi_{i,j}^{(\ell)}(x,y), & q_h^y(x,y,t) &= \sum_{\ell=1}^{k} q_{i,j}^{y,(\ell)}(t)\phi_{i,j}^{(\ell)}(x,y),\\
g_h^y(x,y,t) &= \sum_{\ell=1}^{k} g_{i,j}^{y,(\ell)}(t)\phi_{i,j}^{(\ell)}(x,y),
\end{aligned} \tag{3.23}$$

For convenience, we introduce some notations

$$\begin{aligned}
\vec{u}_{i,j} &= (u_{i,j}^{(1)},\cdots,u_{i,j}^{(k)})^T, & \vec{q}_{i,j}^{\,x} &= (q_{i,j}^{x,(1)},\cdots,q_{i,j}^{x,(k)})^T,\\
\vec{g}_{i,j}^{\,x} &= (g_{i,j}^{x,(1)},\cdots,g_{i,j}^{x,(k)})^T, & \vec{q}_{i,j}^{\,y} &= (q_{i,j}^{y,(1)},\cdots,q_{i,j}^{y,(k)})^T,\\
\vec{g}_{i,j}^{\,y} &= (g_{i,j}^{y,(1)},\cdots,g_{i,j}^{y,(k)})^T,
\end{aligned} \tag{3.24}$$

and define the local mass matrix $\mathbf{M}_{i,j}$, the local stiff matrices $\mathbf{M}_{i,j}^x, \mathbf{M}_{i,j}^y$, the sources related vector $\vec{S}_{i,j}$, the element boundary related matrices $\mathbf{W}_{i,j}$, $\mathbf{E}_{i,j}$, $\mathbf{S}_{i,j}$, $\mathbf{N}_{i,j}$, $\tilde{\mathbf{W}}_{i,j}$, $\tilde{\mathbf{E}}_{i,j}$, $\tilde{\mathbf{S}}_{i,j}$, $\tilde{\mathbf{N}}_{i,j}$,

the initial value related vector $\vec{U}^0_{i,j}$, and the element boundary related vectors $\vec{B}^W_{i,j}$, $\vec{B}^E_{i,j}$, $\vec{B}^S_{i,j}$, $\vec{B}^N_{i,j}$, whose elements are defined by

$$
\begin{aligned}
&(\mathbf{M}_{i,j})_{m,\ell} = \int_{I_{i,j}} \phi^{(\ell)}_{i,j}\phi^{(m)}_{i,j}\,dxdy, && (\mathbf{M}^x_{i,j})_{m,\ell} = \int_{I_{i,j}} \phi^{(\ell)}_{i,j}\frac{\partial \phi^{(m)}_{i,j}}{\partial x}\,dxdy,\\
&(\mathbf{M}^y_{i,j})_{m,\ell} = \int_{I_{i,j}} \phi^{(\ell)}_{i,j}\frac{\partial \phi^{(m)}_{i,j}}{\partial y}\,dxdy, && (\vec{S}_{i,j})_m = \int_{I_{i,j}} p(x,y)\phi^{(m)}_{i,j}\,dxdy,\\
&(\mathbf{W}_{i,j})_{m,\ell} = \int_{\partial I^W_{i,j}} \phi^{W,(\ell)}_{i,j}\phi^{W,(m)}_{i,j}\,dy, && (\tilde{\mathbf{W}}_{i,j})_{m,\ell} = \int_{\partial I^W_{i,j}} \phi^{W,(\ell)}_{i,j}\phi^{E,(m)}_{i,j}\,dy,\\
&(\mathbf{E}_{i,j})_{m,\ell} = \int_{\partial I^E_{i,j}} \phi^{E,(\ell)}_{i,j}\phi^{E,(m)}_{i,j}\,dy, && (\tilde{\mathbf{E}}_{i,j})_{m,\ell} = \int_{\partial I^E_{i,j}} \phi^{E,(\ell)}_{i,j}\phi^{W,(m)}_{i,j}dy,\\
&(\mathbf{S}_{i,j})_{m,\ell} = \int_{\partial I^S_{i,j}} \phi^{S,(\ell)}_{i,j}\phi^{S,(m)}_{i,j}\,dx, && (\tilde{\mathbf{S}}_{i,j})_{m,\ell} = \int_{\partial I^S_{i,j}} \phi^{S,(\ell)}_{i,j}\phi^{N,(m)}_{i,j}dx,\\
&(\mathbf{N}_{i,j})_{m,\ell} = \int_{\partial I^N_{i,j}} \phi^{N,(\ell)}_{i,j}\phi^{N,(m)}_{i,j}\,dx, && (\tilde{\mathbf{N}}_{i,j})_{m,\ell} = \int_{\partial I^N_{i,j}} \phi^{N,(\ell)}_{i,j}\phi^{S,(m)}_{i,j}dx,\\
&(\vec{B}^W_{i,j})_m = \int_{\partial I^W_{i,j}} \phi^{W,(m)}_{i,j}u_W(y,t)\,dy, && (\vec{B}^E_{i,j})_m = \int_{\partial I^E_{i,j}} \phi^{E,(m)}_{i,j}u_E(y,t)\,dy,\\
&(\vec{B}^S_{i,j})_m = \int_{\partial I^S_{i,j}} \phi^{S,(m)}_{i,j}u_S(y,t)\,dx, && (\vec{B}^N_{i,j})_m = \int_{\partial I^N_{i,j}} \phi^{N,(m)}_{i,j}u_N(y,t)\,dx,\\
&(\vec{U}^0_{i,j})_m = \int_{I_{i,j}} u_0(x,y)\phi^{(m)}_{i,j}\,dxdy,
\end{aligned}
\tag{3.25}
$$

where $\partial I^W_{i,j}$, $\partial I^E_{i,j}$, $\partial I^S_{i,j}$ and $\partial I^N_{i,j}$ denote the left, right, bottom and upper parts of $\partial I_{i,j}$, respectively, $\phi^{W,(\ell)}_{i,j}$, $\phi^{E,(\ell)}_{i,j}$, $\phi^{S,(\ell)}_{i,j}$ and $\phi^{N,(\ell)}_{i,j}$ are the values of $\phi^{(\ell)}_{i,j}$ at the points of the element boundaries $\partial I^W_{i,j}$, $\partial I^E_{i,j}$, $\partial I^S_{i,j}$ and $\partial I^N_{i,j}$ corresponding to the nodes of gaussian quadrature, respectively, $\ell = 1,\cdots,k$.

By using the projections in (3.23), the notations in (3.24)-(3.25), the numerical fluxes e.g. (3.21), and taking the test functions $v_{u,h}, v_{q^x,h}, v_{q^y,h} v_{g^x,h}, v_{g^y,h}$, as the basis functions $\phi^{(\ell)}_{i,j}$, $\ell = 1,2,\cdots,k$, the semi-discrete local DG scheme (3.15)-(3.20) can also be rewritten into the following local matrix-vector forms.

The numerical schemes for $\vec{q}^x_{i,j}$ are, for all $i = 1,\cdots,N_x$

$$
\begin{aligned}
&\mathbf{M}_{i,j}\vec{q}^x_{i,j} = (-\mathbf{M}^x_{i,j} - \mathbf{W}_{i,j})\vec{u}_{i,j} + \tilde{\mathbf{W}}_{i,j}\vec{u}_{i,j+1}, && j = 2,\cdots,N_y-1,\\
&\mathbf{M}_{i,j}\vec{q}^x_{i,j} = -\mathbf{M}^x_{i,j}\vec{u}_{i,j} - \vec{B}^W_{i,j} + \tilde{\mathbf{W}}_{i,j}\vec{u}_{i,j+1}, && j = 1,\\
&\mathbf{M}_{i,j}\vec{q}^x_{i,j} = (-\mathbf{M}^x_{i,j} - \mathbf{W}_{i,j})\vec{u}_{i,j} + \vec{B}^E_{i,j}, && j = N_y.
\end{aligned}
\tag{3.26}
$$

For $\vec{q}^y_{i,j}$ and all $j = 1,\cdots,N_y$, we have

$$
\begin{aligned}
&\mathbf{M}_{i,j}\vec{q}^y_{i,j} = (-\mathbf{M}^y_{i,j} - \mathbf{S}_{i,j})\vec{u}_{i,j} + \tilde{\mathbf{S}}_{i,j}\vec{u}_{i+1,j}, && i = 2,\cdots,N_x-1,\\
&\mathbf{M}_{i,j}\vec{q}^y_{i,j} = -\mathbf{M}^y_{i,j}u_{i,j} - \vec{B}^S_{i,j} + \tilde{\mathbf{S}}_{i,j}\vec{u}_{i+1,j}, && i = 1,\\
&\mathbf{M}_{i,j}\vec{q}^y_{i,j} = (-\mathbf{M}^y_{i,j} - \mathbf{S}_{i,j})\vec{u}_{i,j} + \vec{B}^N_{i,j}, && i = N_x.
\end{aligned}
\tag{3.27}
$$

For $\vec{g}^x_{i,j}$ and all $i = 1,\cdots,N_x$, we have

$$
\begin{aligned}
&\mathbf{M}_{i,j}\vec{g}^x_{i,j} = (-\mathbf{M}^x_{i,j} + \mathbf{E}_{i,j})\vec{q}^x_{i,j} - \tilde{\mathbf{E}}_{i,j}\vec{q}^x_{i,j-1}, && j = 2,\cdots,N_y,\\
&\mathbf{M}_{i,j}\vec{g}^x_{i,j} = (-\mathbf{M}^x_{i,j} + \mathbf{E}_{i,j} - \mathbf{W}_{i,j})\vec{q}^x_{i,j}, && j = 1.
\end{aligned}
\tag{3.28}
$$

For $\vec{g}^{y}_{i,j}$ and all $j=1,\cdots,N_y$, we have

$$\begin{array}{ll}\mathbf{M}_{i,j}\vec{g}^{y}_{i,j}=(-\mathbf{M}^{y}_{i,j}+\mathbf{N}_{i,j})\vec{q}^{y}_{i,j}-\tilde{\mathbf{N}}_{i,j}\vec{q}^{y}_{i-1,j}, & i=2,\cdots,N_x,\\ \mathbf{M}_{i,j}\vec{g}^{y}_{i,j}=(-\mathbf{M}^{y}_{i,j}+\mathbf{N}_{i,j}-\mathbf{S}_{i,j})\vec{q}^{y}_{i,j}, & i=1.\end{array} \tag{3.29}$$

The scheme for $\vec{u}_{i,j}$ is, for all $i=1,\cdots,N_x, j=1,\cdots,N_y$

$$\mathbf{M}_{i,j}\frac{d\vec{u}_{i,j}}{dt}=\sum_{r=1}^{j}\mathbf{M}^{1,i,j,r}\vec{g}^{x}_{i,r}+\sum_{s=1}^{i}\mathbf{M}^{2,i,j,s}\vec{g}^{y}_{s,j}+\vec{S}_{i,j}, \tag{3.30}$$

where $\mathbf{M}^{1,i,j,r}$ and $\mathbf{M}^{2,i,j,s}$ are defined as

$$(\mathbf{M}^{1,i,j,r})_{m,\ell}:=\int_{I_{i,j}}d_1(x,y)\phi^{(m)}_{i,j}\mathscr{K}^{x}[\phi^{(\ell)}_{i,r}]\,dxdy,$$
$$(\mathbf{M}^{2,i,j,s})_{m,\ell}:=\int_{I_{i,j}}d_2(x,y)\phi^{(m)}_{i,j}\mathscr{K}^{y}[\phi^{(\ell)}_{s,j}]\,dxdy.$$

Similar to one-dimensional case, the analytical expressions of $\mathscr{K}^{x}[\phi^{(\ell)}_{r,s}]$ and $\mathscr{K}^{y}[\phi^{(\ell)}_{r,s}]$ may be obtained by using the symbolic toolbox of *Matlab*.

The matrix (3.30) is a system of ordinary differential equations and will be solved by the fourth-order accurate non-TVD Runge-Kutta method (2.30).

The discrete initial condition is

$$\mathbf{M}_{i,j}\vec{u}_{i,j}(0)=\vec{U}^{0}_{i,j}. \tag{3.31}$$

Remark 3.1 In practice, the basis functions of the space $P(I_{i,j})$ may be obtained by mapping the basis functions of the space $P(I)$ by the transformation $\Psi:(\xi,\eta)\in I\mapsto(x,y)\in I_{i,j}$, see e.g., [22], where $I=[-1,1]\times[-1,1]$ denotes the standard rectangular element. For example, if we choose a set of basis functions for $P(I)$ as

$$\phi^{(\ell)}(\xi,\eta):=L_{\ell_1}(\xi)L_{\ell_2}(\eta),\ \ 0\le\ell_1,\ell_2\le\hat{k},\ \ -1\le\xi,\eta\le1, \tag{3.32}$$

where $L_{\ell_1}(\cdot)$ and $L_{\ell_2}(\cdot)$ are the Legendre polynomials of order ℓ_1 and ℓ_2 respectively, and $k=(\hat{k}+2)(\hat{k}+1)/2$, then a set of the basis functions for the space $P(I_{i,j})$ can be taken as

$$\phi^{(\ell)}_{i,j}(x,y):=\phi^{(\ell)}\big(\xi(x,y),\eta(x,y)\big),\ell=1,2,\cdots,k,\ (x,y)\in I_{i,j}.$$

4. Numerical results

This section conducts some numerical experiments to show the accuracy and the performance of the present local DG methods for the fractional diffusion equations (2.1) and (3.1). The time step size of the Runge-Kutta local P^k-DG methods is taken as $\Delta t=\left(\frac{\Delta x}{k^2}\right)^{\alpha}$ for one-dimensional problems and $dt=0.5(\frac{h}{k^2})^{\max\{\alpha,\beta\}}$ for two-dimensional problems, respectively, $h=\min\{\Delta x,\Delta y\}$, which is in agreement with the case of the integer values of α [16].

Table 1: Example 4.1: Numerical errors and convergence rates of the local P^k-DG methods at $t=1$. Here $\alpha=1.2$ and $d(x)$ is a constant function.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
1	2^5	6.09e-04	–	7.97e-05	–	1.40e-04	–
	2^6	1.45e-04	2.06	1.89e-05	2.08	3.34e-05	2.06
	2^7	3.60e-05	2.02	4.63e-06	2.03	8.26e-06	2.01
2	2^3	3.07e-04	–	5.24e-05	–	7.69e-05	–
	2^4	3.78e-05	3.02	6.20e-06	3.08	9.19e-06	3.06
	2^5	4.75e-06	2.99	7.63e-07	3.02	1.14e-06	3.02
3	4	9.08e-05	–	2.07e-05	–	2.83e-05	–
	8	9.64e-07	4.17	1.15e-06	4.17	1.61e-06	4.14
	12	2.61e-8	4.09	2.15e-07	4.14	3.07e-07	4.09
4	2	3.77e-05	–	1.07e-05	–	1.48e-05	–
	3	4.56e-06	5.21	1.37e-06	5.06	1.81e-06	5.18
	4	9.21e-07	5.56	3.57e-07	4.67	4.34e-07	4.96

4.1. 1D numerical results

Example 4.1 (Diffusive coefficient is constant)**.** Consider the initial-boundary value problem (2.1)-(2.2) on the domain $\Omega=[0,1]$ with initial condition $u(x,0)=x^5$ and the boundary conditions $u(0,t)=0$ and $u(1,t)=e^{-t}$. Let

$$d(x)=\frac{\Gamma(6-\alpha)}{\Gamma(6)},\quad p(x,t)=-e^{-t}(x^5+x^{5-\alpha}),\quad \alpha\in(1,2). \tag{4.1}$$

In this case, the exact solution is $u(x,t)=e^{-t}x^5$, which does not depend on the value of α and will be used to estimate the numerical errors in our computations.

Table 1 shows the numerical L^1-, L^2-, and L^∞-errors and the convergence rates of the local P^k-DG methods at $t=1$ for the case $\alpha=1.2$ with $k=1,2,3,4$, where N denotes the number of elements. From there we see that the local P^k-DG methods can achieve the accuracy of order $k+1$.

Example 4.2 (Diffusive coefficient is nonconstant)**.** Consider the initial-boundary value problem (2.1)-(2.2) on the domain $\Omega=[0,1]$ with initial condition $u(x,0)=x^5$ and the boundary conditions $u(0,t)=0$ and $u(1,t)=e^{-t}$. Let

$$d(x)=\frac{\Gamma(6-\alpha)}{\Gamma(6)}x^\alpha,\quad p(x,t)=-2e^{-t}x^5,\quad \alpha\in(1,2). \tag{4.2}$$

Now the exact solution is given by $e^{-t}x^5$, which does not depend on the value of α either.

We estimate the numerical L^1-, L^2-, and L^∞-errors and the numerical convergence rates of the local P^k-DG methods at $t=1$ for the cases $\alpha=1.01$, 1.2 and 1.99, $k=1,2,3,4$. Those results are given in Tables 2, 3 and 4, and show that the local P^k-DG methods are $(k+1)$st order accurate in both cases.

Table 2: Example 4.2: Numerical errors and convergence rates of the local P^k-DG methods at $t=1$. Here $\alpha=1.01$ and $d(x)$ is a function of x.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
1	2^5	1.29e-03	–	1.44e-04	–	2.68e-04	–
	2^6	3.23e-04	2.00	3.45e-05	2.06	6.46e-05	2.05
	2^7	7.98e-05	2.02	8.00e-06	2.11	1.52e-05	2.09
2	2^3	1.08e-03	–	1.30e-04	–	2.25e-04	–
	2^4	1.36e-04	2.99	1.22e-05	3.41	2.14e-05	3.39
	2^5	1.26e-05	3.43	9.08e-07	3.75	1.62e-06	3.72
3	4	2.63e-04	–	7.24e-05	–	9.79e-05	–
	8	1.87e-05	3.82	4.15e-06	4.13	5.42e-06	4.17
	12	3.46e-06	4.16	7.59e-07	4.19	1.00e-06	4.16
4	2	1.50e-04	–	3.44e-05	–	5.05e-05	–
	3	1.56e-05	5.58	5.55e-06	4.50	7.19e-06	4.81
	4	5.04e-06	3.93	1.32e-06	4.99	1.79e-06	4.84

Example 4.3 (Diffusive coefficient is nonconstant)**.** This example is used to test the convergence when increasing the order of the finite element approximation. Solve numerically the initial-boundary value problem (2.1)-(2.2) within the domain $\Omega=[0,1]$ with the initial condition

$$u(x,0)=(x-1)^{16}x^{16}10^{10}+(x-0.4)^2, \tag{4.3}$$

and the boundary conditions $u(0,t)=0.16e^t, u(1,t)=0.36e^t$. Here $d(x)$ and $p(x,t)$ are taken as

$$d(x)=x^{1/5}\Gamma(0.8), \tag{4.4}$$

and

$$\begin{aligned}p(x,t)=&-10^{10}/6080241545153194351130616254 76e^{-t}x^{15}(6080241545153194351130616254 76x^{17}\\&+3540510620429609825907724446113 4x^{16}-6220928886768962897880805541616 30x^{15}\\&+4670634487774233633025986026842815x^{14}-2134324954285871870286229424788 3680x^{13}\\&+6728705615146745769898803646835833 2x^{12}-1557997887116294930642310575813131 92x^{11}\\&+2745122193193351342941272537154920 60x^{10}+2274865280859856238834133904722645 82x^6\\&+4628575440138535892341981537254594 0x^4-1341278265599326983352739940181788 0x^3\\&+2697560382746457054641828709304884x^2-1175783907643812330461240129978836 80x^5\\&-3757135636393747420607076898488738 80x^9-3420034644996646772092048124641256 92x^7\\&+4038142954303987189794528762740858 10x^8-3362990226815898389511174071245 24x\\&+1957033580886199081802368164062 5)+e^t\left((x-0.4)^2-\frac{2028240960365167138960136684156 25}{8112963841460668169578900514406 4}x\right),\end{aligned} \tag{4.5}$$

respectively, where $p(x,t)$ is generated by *Matlab* in order to have the exact solution $u(x,t)=e^{-t}(x-1)^{16}x^{16}10^{10}+e^t(x-0.4)^2$.

Fig. 1 gives the exact solution u and the numerical solutions u_h of the local P^k-DG methods at $t=1$, $k=2,3,4$. Here $\alpha=1.2$, and 4 elements are used, 20 samples points

Table 3: Same as Table 2, except for $\alpha = 1.2$.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
	2^5	6.12e-04	–	8.08e-05	–	1.41e-04	–
1	2^6	1.46e-04	2.07	1.90e-05	2.09	3.35e-05	2.07
	2^7	3.60e-05	2.02	4.64e-06	2.03	8.28e-06	2.02
	2^3	3.11e-04	–	5.41e-05	–	7.89e-05	–
2	2^4	3.78e-05	3.04	6.30e-06	3.10	9.28e-06	3.09
	2^5	4.75e-06	2.99	7.70e-07	3.03	1.14e-06	3.02
	4	9.35e-05	–	2.95e-05	–	3.95e-05	–
3	8	5.78e-06	4.01	1.60e-06	4.20	2.14e-06	4.20
	12	1.20e-06	3.88	3.04e-07	4.10	4.11e-07	4.08
	2	4.17e-05	–	1.03e-05	–	1.62e-05	–
4	3	5.81e-06	4.87	1.60e-06	4.60	2.27e-06	4.84
	4	1.33e-06	5.13	4.08e-07	4.75	5.52e-07	4.92

Table 4: Same as Table 2, except for $\alpha = 1.99$.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
	2^5	5.55e-04	–	7.40e-05	–	1.32e-04	–
1	2^6	1.40e-04	1.98	1.83e-05	2.01	3.28e-05	2.01
	2^7	3.53e-05	1.99	4.56e-06	2.01	8.18e-06	2.00
	2^3	2.81e-04	–	4.97e-05	–	7.32e-05	–
2	2^4	3.66e-05	2.94	6.09e-06	3.03	9.04e-06	3.02
	2^5	4.67e-06	2.97	7.56e-07	3.01	1.12e-06	3.01
	4	6.68e-05	–	1.55e-05	–	2.30e-05	–
3	8	4.36e-06	3.94	9.45e-07	4.03	1.41e-06	4.02
	12	8.76e-07	3.96	1.85e-07	4.02	2.78e-07	4.01
	2	2.57e-05	–	7.23e-06	–	1.13e-05	–
4	3	3.37e-06	5.01	1.25e-06	4.34	1.70e-06	4.68
	4	8.00e-07	5.00	3.29e-07	4.63	4.24e-07	4.83

are taken in each element to get the plots. We see that the solutions of the local P^k-DG methods are more and more accurate with the increasing of the approximation order k.

4.2. 2D numerical results

Example 4.4 (Diffusive coefficients are constant)**.** Consider the two-dimensional initial-boundary value problem (3.1)-(3.2) on $\Omega = \{(x,y)\big|\ 0 < x < 1, 0 < y < 1\}$. The initial condition and the boundary conditions are specified as

$$\begin{aligned} &u(x,y,0) = x^5y^5,\ \ u(x,0,t) = u(0,y,t) = 0,\\ &u(x,1,t) = e^{-t}x^5,\ \ u(1,y,t) = e^{-t}y^5, \end{aligned} \tag{4.6}$$

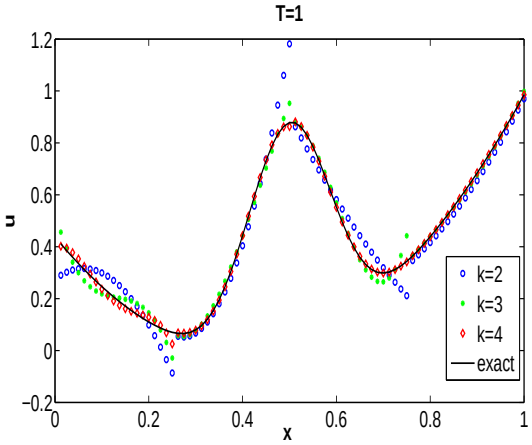

Figure 1: Example 4.3: The exact and numerical solutions of the local P^k-DG methods at $t=1$, $k=2,3,4$. Here $\alpha=1.2$.

and the functions $d_1(x,y)$, $d_2(x,y)$, and $p(x,y,t)$ are

$$d_1(x,y)=\frac{\Gamma(6-\alpha)}{2\Gamma(6)},\quad d_2(x,y)=\frac{\Gamma(6-\beta)}{2\Gamma(6)}, \tag{4.7}$$

$$p(x,y,t)=\frac{1}{2}\Big[-e^{-t}(x^5+x^{5-\alpha})-e^{-t}(x^5+x^{5-\beta})\Big],\quad \alpha\in(1,2),\quad \beta\in(1,2). \tag{4.8}$$

The exact solution of this initial-boundary value problem is $u(x,y,t)=e^{-t}x^5y^5$, independent on the parameters α and β.

We estimate the numerical L^1-, L^2-, and L^∞-errors and the numerical convergence orders of the local P^k-DG methods at $t=1$ for the case $(\alpha,\beta)=(1.2,1.8)$ with $k=1,2,3,4$. Numerical results are presented in Table 5 and show that the local P^k-DG methods can achieve the accuracy of order $k+1$, where N denotes the number of elements.

Example 4.5 (Diffusive coefficients are nonconstant)**.** Solve the initial-boundary value problem (3.1)-(3.2) with non-constant diffusive coefficients

$$d_1(x,y)=\frac{\Gamma(6-\alpha)}{2\Gamma(6)}x^\alpha,\quad d_2(x,y)=\frac{\Gamma(6-\beta)}{2\Gamma(6)}x^\beta, \tag{4.9}$$

with initial condition $u(x,y,0)=x^5y^5, 0<x<1, 0<y<1$ and the boundary conditions

$$u(x,0,t)=u(0,y,t)=0,\ \ u(x,1,t)=e^{-t}x^5,\ \ u(1,y,t)=e^{-t}y^5. \tag{4.10}$$

If the source term in (3.1) is taken as

$$p(x,y,t)=-2e^{-t}x^5y^5, \tag{4.11}$$

then the exact solution may be given by $u(x,y,t)=e^{-t}x^5y^5$, which does not depend on α and β.

Table 5: Example 4.4: Numerical errors and convergence rates of the local P^k-DG methods at $t=1$. Here $(\alpha,\beta)=(1.2,1.8)$ and $d_1(x,y)$ and $d_2(x,y)$ are two constant functions.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
1	$2^3\times 2^3$	2.86e-02	–	5.52e-04	–	2.02e-03	–
	$2^4\times 2^4$	7.93e-03	1.85	1.10e-04	2.33	4.00e-04	2.33
	$2^5\times 2^5$	2.15e-03	1.88	2.33e-05	2.24	8.44e-05	2.25
2	$2^2\times 2^2$	1.90e-02	–	5.05e-04	–	1.36e-03	–
	$2^3\times 2^3$	2.93e-03	2.70	4.87e-05	3.38	1.49e-04	3.19
	$2^4\times 2^4$	4.33e-04	2.76	5.59e-06	3.12	1.79e-05	3.05
3	4×4	2.86e-03	–	8.61e-05	–	1.95e-04	–
	6×6	7.29e-04	3.37	1.43e-05	4.43	3.65e-05	4.14
	8×8	2.54e-04	3.66	4.27e-06	4.19	1.14e-05	4.06
4	3×3	1.14e-03	–	3.80e-05	–	7.35e-05	–
	4×4	3.22e-04	4.41	8.55e-06	5.19	1.73e-05	5.02
	5×5	1.16e-04	4.56	2.71e-06	5.19	5.62e-06	5.04

Table 6: Example 4.5: Numerical errors and convergence rates of the local P^k-DG methods at $t=1$. Here $(\alpha,\beta)=(1.2,1.8)$ and $d_1(x,y)$ and $d_2(x,y)$ depend on the independent variables x and y, N is the number of elements.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
1	$2^3\times 2^3$	2.86e-02	–	5.38e-04	–	1.98e-03	–
	$2^4\times 2^4$	7.93e-03	1.85	1.12e-04	2.26	4.04e-04	2.29
	$2^5\times 2^5$	2.14e-03	1.89	2.40e-05	2.23	8.57e-05	2.24
2	$2^2\times 2^2$	1.95e-02	–	5.08e-04	–	1.44e-03	–
	$2^3\times 2^3$	2.92e-03	2.75	5.23e-05	3.28	1.54e-04	3.22
	$2^4\times 2^4$	4.32e-04	2.76	5.77e-06	3.18	1.83e-05	3.08
3	4×4	2.93e-03	–	8.20e-05	–	2.03e-04	–
	6×6	7.35e-04	3.41	1.43e-05	4.30	3.73e-05	4.17
	8×8	2.55e-04	3.67	4.26e-06	4.22	1.15e-05	4.09
4	3×3	1.15e-03	–	3.33e-0	–	7.42e-05	–
	4×4	3.25e-04	4.38	7.46e-06	5.20	1.74e-05	5.05
	5×5	1.17e-04	4.59	2.35e-06	5.17	5.56e-06	5.10

We also estimate the numerical L^1-, L^2-, and L^∞-errors and the convergence orders of the local P^k-DG methods at $t=1$ of the local P^k-DG methods at $t=1$ for the cases $(\alpha,\beta)=(1.2,1.8)$ and $(1.99,1.99)$ with $k=1,2,3,4$. Those results in Tables 6 and 7 show that the local P^k-DG methods are $(k+1)$st order accurate in both cases.

Example 4.6 (Diffusive coefficients are nonconstant)**.** This example is to solve the initial-boundary value problem (3.1)-(3.2) with slightly more complicated diffusive coefficients

$$d_1(x,y)=\frac{399}{3125}x^{6/5}\Gamma(0.8),\quad d_2(x,y)=\frac{44}{3125}y^{9/5}\Gamma(0.2). \tag{4.12}$$

The initial condition and the boundary conditions are

$$u(x,y,0)=10^4(x-0.5)(x-1)^2x^2(y-0.5)(y-1)^2y^2, \tag{4.13}$$

Table 7: Same as Table 6, except for $(\alpha,\beta)=(1.99,1.99)$.

k	N	L^∞–error	order	L^1–error	order	L^2–error	order
1	$2^3\times 2^3$	1.40e-02	–	5.38e-04	–	1.03e-03	–
	$2^4\times 2^4$	4.18e-03	1.74	1.12e-04	2.26	2.40e-04	2.10
	$2^5\times 2^5$	1.15e-03	1.86	2.40e-05	2.23	5.78e-05	2.05
2	$2^2\times 2^2$	1.35e-02	–	5.08e-04	–	1.08e-03	–
	$2^3\times 2^3$	2.37e-03	2.51	5.23e-05	3.28	1.32e-04	3.02
	$2^4\times 2^4$	3.55e-04	2.74	5.77e-06	3.18	1.63e-05	3.02
3	4×4	2.93e-03	–	8.20e-05	–	1.35e-04	–
	6×6	7.35e-04	3.41	1.43e-05	4.30	2.76e-05	3.91
	8×8	2.55e-04	3.67	4.26e-06	4.22	8.92e-06	3.93
4	3×3	7.13e-04	–	3.33e-0	–	4.84e-05	–
	4×4	2.07e-04	4.31	7.46e-06	5.20	1.19e-05	4.89
	5×5	7.65e-05	4.46	2.35e-06	5.17	3.96e-06	4.92

and $u(0,y,t)=u(1,y,t)=u(x,0,t)=u(x,1,t)=0$, respectively. The function $p(x,y,t)$ in (3.1) are given as

$$\begin{aligned}p(x,y,t)=-2e^{-t}x^2y^2(&-8364x-8356y+1825+15000x^3y^3-33000x^3y^2\\&+23520x^3y-5352x^3-34500x^2y^3+75000x^2y^2-52800x^2y+\\&11880x^2+25320xy^3-54300xy^2-5798y^3+12245y^2+37680xy),\end{aligned}\tag{4.14}$$

in order to have the exact solution $u(x,y,t)=10^4e^{-t}(x-0.5)^2(x-1)^2x^2(y-0.5)^2(y-1)^2y^2$. Fig. 2 shows the solution of the local P^4-DG method at $t=1$, Fig. 3 gives the exact solution and the numerical solutions of the local P^k-DG methods at $t=1$ along the line $y=0.25$, where $k=2,3,4$. In these numerical experiments, we take $(\alpha,\beta)=(1.2,1.8)$,

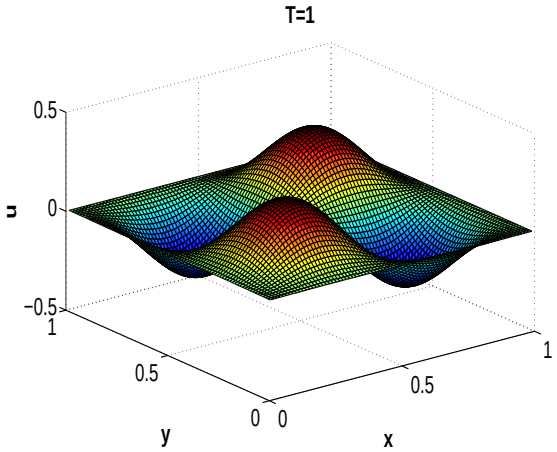

Figure 2: Example 4.6: The numerical solution of the local P^4-DG method at $t=1$.

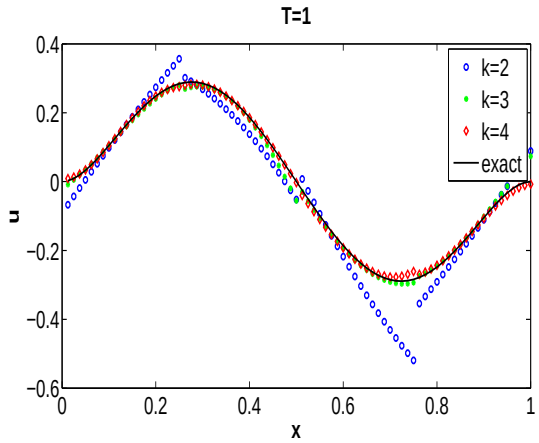

Figure 3: Example 4.6: The exact solution and the numerical solutions of the local P^k-DG methods at $t=1$ along the line $y=0.25$.

4×4 elements and 20×20 samples points in each element to get the plots. The results show that the approximation is also more and more accurate with the increasing of k. Fig. 4 further gives the evolution of the relative energy errors of the local P^k-DG methods in time, $k=2,3,4$, where the relative error is defined by

$$\left|\int_0^1 \frac{u_h^2(x,t)}{2}dx-\int_0^1 \frac{u^2(x,t)}{2}dx\right| \Big/ \int_0^1 \frac{u^2(x,t)}{2}dx, \tag{4.15}$$

where u_h is the numerical solution of the local P^k-DG method, and u is the exact solution.

Example 4.7 (Diffusive coefficients are nonconstant)**.** The last example is used to further evaluate the proposed local P^k-DG methods. We solve the initial-boundary value problem

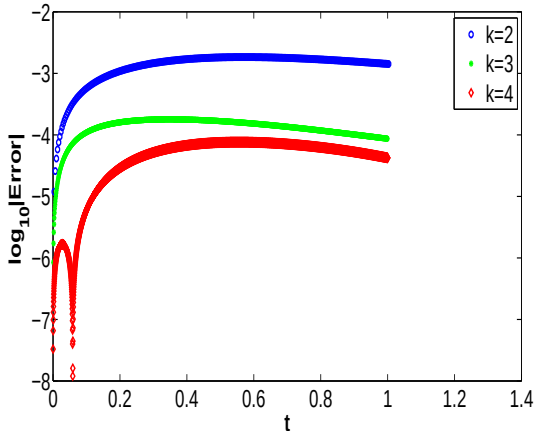

Figure 4: Example 4.6: The relative energy errors of the local P^k-DG methods from $t=0$ to 1.

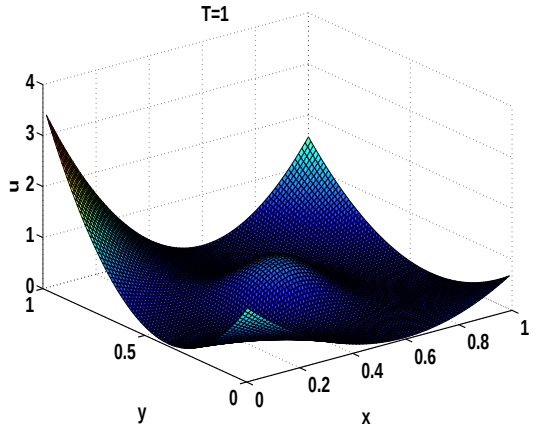

Figure 5: Example 4.7: The numerical solution of the local P^4-DG method at $t=1$.

(3.1)-(3.2) with the initial condition

$$u(x,y,0)=10^{10}(x-1)^8x^8(y-1)^8y^8+10(x-0.6)^2(y-0.4)^2,\quad 0\le x,y\le 1, \tag{4.16}$$

and the boundary conditions

$$u(x,0,t)=1.6e^t(x-0.6)^2,\quad u(0,y,t)=3.6e^t(y-0.4)^2, \tag{4.17}$$
$$u(x,1,t)=3.6e^t(x-0.6)^2,\quad u(1,y,t)=1.6e^t(y-0.4)^2. \tag{4.18}$$

The diffusive coefficients $d_1(x,y)$ and $d_2(x,y)$ and the source function $p(x,y,t)$ are chosen as

$$d_1(x,y)=10^{-5}x^{1/5}\Gamma(0.8),\quad d_2(x,y)=10^{-5}y^{4/5}\Gamma(0.2), \tag{4.19}$$

and

$$\begin{aligned}
p(x,y,t)&=10^{10}e^{-t}(-(x-1)^8x^8(y-1)^8y^8-1/100000x^{1/5}(y-1)^8y^8\\
&(-762939453125/3203176788x^{69/5}+683593750/1339481x^{44/5}+390625/28101x^{34/5}\\
&-15625000/121771x^{39/5}-976562500/852397x^{49/5}+30517578125/39791016x^{64/5}\\
&+12207031250/7671573x^{54/5}+3814697265625/118517541156x^{74/5}\\
&-48828125000/34817139x^{59/5})-1/100000y^{4/5}(x-1)^8x^8(-6835937500/380029y^{46/5}\\
&+170898437500/6460493y^{51/5}-1525878906250/333460831y^{66/5}\\
&+15258789062500/23675719001y^{71/5}-7812500/4433y^{36/5}+1367187500/181753y^{41/5}\\
&+781250/4433y^{31/5}+427246093750/30314621y^{61/5}-12207031250/496961y^{56/5}))\\
&+10e^t((x-3/5)^2(y-2/5)^2\\
&-3245185536584267422336218694 65/1298074214633706907132624082305024x(y-2/5)^2\\
&-2028240960365167113619036909101/2028240960365167042394725128601600y(x-3/5)^2),
\end{aligned} \tag{4.20}$$

which is generated by *Matlab*. Now the exact solution of the problem is $u(x,y,t)=e^{-t}10^{10}(x-1)^8x^8(y-1)^8y^8+10e^t(x-0.6)^2(y-0.4)^2$. Fig. 5 shows the solution of the

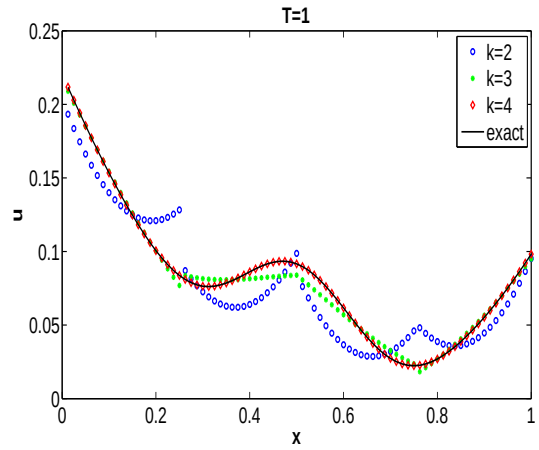

Figure 6: Example 4.7: The exact solution and the numerical solutions of the local P^k-DG methods at $t=1$ along the line $y=0.5$.

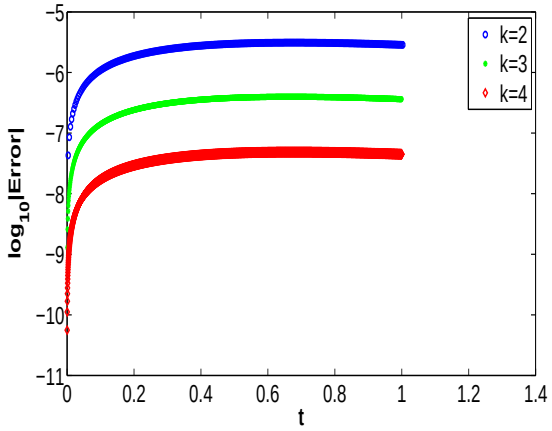

Figure 7: Example 4.7: The relative energy errors of the local P^k-DG methods from $t=0$ to 1.

local P^4-DG method at $t=1$, Fig. 6 gives the exact solution and the numerical solutions of the local P^k-DG methods at $t=1$ along the line $y=0.5$, where $k=2,3,4$. In the numerical experiments, we take $(\alpha,\beta)=(1.2,1.8)$, 4×4 elements and 20×20 samples points in each element to get the plots. The results show that the approximation is also more and more accurate with the increasing of k. Fig. 7 further gives the evolution of the relative energy errors of the local P^k-DG methods in time, $k=2,3,4$, where the relative error is defined by (4.15).

5. Conclusions

Fractional calculus is a natural extension of the integer order calculus. In recent years the numerical solutions of the fractional PDEs have become considerably interesting both

in mathematics and physics as well as in applications. Based on two (or four) auxiliary variables in one dimension (or two dimensions) and the Caputo derivative as the spatial derivative, this paper developed the high-order accurate Runge-Kutta local discontinuous Galerkin (DG) methods for one- and two-dimensional space-fractional diffusion equations with the variable diffusive coefficients. No any additional boundary condition is required. The motivation to choose the Caputo derivative is that it could represent the fractional derivative by an integral operator. Numerical examples were given to estimate the numerical errors and evaluate the numerical resolutions of the solutions. Our results showed that the convergence orders of the proposed local P^k–DG methods are $O(h^{k+1})$ both in one- and two-dimensions, where P^k denotes the space of the real-valued polynomials of degree at most k. It is worth mentioning that an alternative work on the DG method is given in [14] for the one-dimensional fractional diffusion equation with a constant diffusive coefficient and a more restrict assumption on the left boundary condition: $u(a,t)=0$. It is a challenge to develop the high-order accurate Runge-Kutta local discontinuous Galerkin (DG) methods on unstructured meshes and the adaptive refinement meshes.

Acknowledgments Xia Ji is partially supported by the National Natural Science Foundation of China for the Youth (No. 10901157/ A0117) and the National Basic Research Program of China (973 Program 2012CB025904). Huazhong Tang was partially supported by the National Basic Research Program under the Grant 2005CB321703, the National Natural Science Foundation of China (No. 10925101, 10828101), the Program for New Century Excellent Talents in University (NCET-07-0022), and the Doctoral Program of Education Ministry of China (No. 20070001036).

Appendix: Alternate method for equation (2.29)

The standard Legendre polynomials with $\ell = 0,1,2,3,4$ defined on $[-1,1]$ are

$$L_0(x)=1, \qquad L_1(x)=x,$$

$$L_2(x)=\frac{1}{2}(3x^2-1), \tag{A.1}$$

$$L_3(x)=\frac{1}{2}(5x^3-3x), \tag{A.2}$$

$$L_4(x)=\frac{1}{8}(35x^4-30x^2+3). \tag{A.3}$$

Using the symbolic toolbox of Matlab, we can get the integrals in (2.29) as follows

$$\int_0^1 (x-s)^{-0.2}L_0(2s-1)ds=\frac{5}{4}x^{\frac{4}{5}}-\frac{5}{4}(x-1)^{\frac{4}{5}}, \tag{A.4}$$

$$\int_0^x (x-s)^{-0.2}L_0(2s-1)ds=\frac{5}{4}x^{\frac{4}{5}}, \tag{A.5}$$

$$\int_0^1 (x-s)^{-0.2} L_1(2s-1)ds = -\frac{5}{4}x^{\frac{4}{5}} + \frac{25}{18}x^{\frac{9}{5}} + \frac{5}{36}(x-1)^{\frac{4}{5}} - \frac{25}{18}(x-1)^{\frac{4}{5}}x, \tag{A.6}$$

$$\int_0^x (x-s)^{-0.2} L_1(2s-1)ds = -\frac{5}{4}x^{\frac{4}{5}} + \frac{25}{18}x^{\frac{9}{5}}, \tag{A.7}$$

$$\int_0^1 (x-s)^{-0.2} L_2(2s-1)ds = \frac{5}{4}x^{\frac{4}{5}} - \frac{25}{6}x^{\frac{9}{5}} + \frac{125}{42}x^{\frac{14}{5}} - \frac{5}{84}(x-1)^{\frac{4}{5}} + \frac{25}{14}(x-1)^{\frac{4}{5}}x - \frac{125}{42}(x-1)^{\frac{4}{5}}x^2, \tag{A.8}$$

$$\int_0^x (x-s)^{-0.2} L_2(2s-1)ds = \frac{5}{4}x^{\frac{4}{5}} - \frac{25}{6}x^{\frac{9}{5}} + \frac{125}{42}x^{\frac{14}{5}}, \tag{A.9}$$

$$\int_0^1 (x-s)^{-0.2} L_3(2s-1)ds = -\frac{5}{4}x^{\frac{4}{5}} + \frac{25}{3}x^{\frac{9}{5}} - \frac{625}{42}x^{\frac{14}{5}} + \frac{3125}{399}x^{\frac{19}{5}} + \frac{55}{1596}(x-1)^{\frac{4}{5}} - \frac{275}{133}(x-1)^{\frac{4}{5}}x + \frac{6875}{798}(x-1)^{\frac{4}{5}}x^2 - \frac{3125}{399}x^3(x-1)^{\frac{4}{5}}, \tag{A.10}$$

$$\int_0^x (x-s)^{-0.2} L_3(2s-1)ds = -\frac{5}{4}x^{\frac{4}{5}} + \frac{25}{3}x^{\frac{9}{5}} - \frac{625}{42}x^{\frac{14}{5}} + \frac{3125}{399}x^{\frac{19}{5}}, \tag{A.11}$$

$$\int_0^1 (x-s)^{-0.2} L_4(2s-1)ds = \frac{5}{4}x^{\frac{4}{5}} - \frac{125}{9}x^{\frac{9}{5}} + \frac{625}{14}x^{\frac{14}{5}} - \frac{3125}{57}x^{\frac{19}{5}} - \frac{55}{2394}(x-1)^{\frac{4}{5}} + \frac{15625}{684}x^4(x^{\frac{4}{5}} - (x-1)^{\frac{4}{5}}) + \frac{2750}{1197}(x-1)^{\frac{4}{5}}x - \frac{6875}{399}(x-1)^{\frac{4}{5}}x^2 + \frac{6250}{171}(x-1)^{\frac{4}{5}}x^3, \tag{A.12}$$

$$\int_0^x (x-s)^{-0.2} L_4(2s-1)ds = \frac{5}{4}x^{\frac{4}{5}} - \frac{125}{9}x^{\frac{9}{5}} + \frac{625}{14}x^{\frac{14}{5}} - \frac{3125}{57}x^{\frac{19}{5}} + \frac{15625}{684}x^{\frac{19}{5}}, \tag{A.13}$$

here we choose $\alpha = 1.2$.

References

[1] B. Baeumer, D. A. Benson, M. M. Meerschaert and S. W. Wheatcraft, *Subordinated advection-dispersion equation for contaminant transport*, Water Resour. Res., 37 (2001), pp. 1543–1550.

[2] E. Barkai, R. Metzler and J. Klafter, *From continuous time random walks to the fractional Fokker-Planck equation*, Phys. Rev. E, 61 (2000), pp. 132–138.

[3] C. M. Chen, F. Liu, I. Turner and V. Anh, *A Fourier method for the fractional diffusion equation describing sub-diffusion*, J. Comput. Phys., 227 (2007), pp. 886–897.

[4] B. Cockburn, *Discontinuous Galerkin methods for convection-dominated problems*, in High-Order Methods for Computational Physics, T. J. Barth and H. Deconinck, editors, Lecture Notes in Computational Science and Engineering, Vol. 9, Springer, 1999, pp. 69–224.

[5] B. Cockburn and C. W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework*, Math. Comp., 52 (1989), pp. 411-435.

[6] B. Cockburn, S. Y. Lin and C. W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems*, J. Comput. Phys., 84 (1989), pp. 90–113.
[7] B. Cockburn, S. Hou and C. W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case*, Math. Comp., 54 (1990), pp. 545–581.
[8] B. Cockburn and C. W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, J. Comput. Phys., 141 (1998), pp. 199–224.
[9] B. Cockburn and C. W. Shu, *Runge-Kutta discontinuous Galerkin method for convection-dominated problems*, J. Sci. Comput., 16 (2001), pp. 173–261.
[10] B. Cockburn and C. W. Shu, *The local discontinuous Galerkin finite element method for convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463.
[11] M. R. Cui, *Compact finite difference method for the fractional diffusion equation*, J. Comput. Phys., 228 (2009), pp. 7792–7804.
[12] W. H. Deng, *Short memory principle and a predictor-corrector approach for fractional differential equations*, J. Comput. Appl. Math., 206 (2007), pp. 174–188.
[13] W. H. Deng, *Finite element methods for the space and time fractional Fokker-Planck equation*, SIAM J. Numer. Anal., 47 (2008), pp. 204–226.
[14] W. H. Deng and J. S. Hesthaven, *Discontinuous Galerkin methods for fractional diffusion equations*, preprint, 2010.
[15] S. Gottlieb and C. W. Shu, *Total variation diminishing Runge-Kutta schemes*, Math. Comp., 67 (1998), pp. 73–85.
[16] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Springer-Verlag, 2008.
[17] P. Lesaint and P. A. Raviart, *On a finite element method for solving the neutron transport equation*, in Mathematical Aspects of Finite Elements in Partial Differential Equations, C. deBoor, ed., Academic Press, New York, 1974, pp. 89–123.
[18] H. Li and R. X. Liu, *The discontinuous Galerkin finite element method for the 2D shallow water equations*, Mathematics and Computers in Smulation, 56 (2001), pp. 223–233.
[19] X. J. Li and C. J. Xu, *A space-time spectral method for the time fractional diffusion equation*, SIAM J. Numer. Anal., 47 (2009), pp. 2108–2131.
[20] Y. M. Lin and C. J. Xu, *Finite difference/spectral approximations for the time-fractional diffusion equation*, J. Comput. Phys., 225 (2007), pp. 1533–1552.
[21] F. 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*, Appl. Math. Comput., 191 (2006), pp. 12–20.
[22] T. Lu, P. W. Zhang and W. Cai, *Discontinuous Galerkin methods for dispersive and lossy Maxwell's equations and PML boundary conditions*, J. Comput. Phys., 200 (2004), pp. 549–580.
[23] C. Lubich, *Discretized fractional calculus*, SIAMJ. Math. Anal., 17 (1986), pp. 704–719.
[24] V. E. Lynch, B. A. Carreras, D. Del-Castillo-Negrete, K. M. Ferreira-Mejias and H. R. Hicks, *Numerical methods for the solution of partial differential equations of fractional order*, J. Comput. Phys., 192 (2003), pp. 406–421.
[25] W. McLean and K. Mustapha, *Convergence analysis of a discontinuous Galerkin method for a sub-diffusion equation*, Numer. Algor., 52 (2009), pp. 69–88.
[26] K. Mustapha and W. McLean, *Piecewise-linear, discontinuous Galerkin method for a fractional diffusion equation*, Numer. Algor., 56 (2011), pp. 159–184.
[27] R. Metzler and J. Klafter, *The random walk's guide to anomalous diffusion: a fractional dynamics approach*, Phys. Rep., 339 (2000), pp. 1–77.

[28] K. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, Wiley, 1993.
[29] E. W. Montroll and M. F. Shlesinger, *On the wonderful world of random walks*, in Studies in Statistical Mechanics, edited by J. Lebowitz and E. Montroll, North-Holland, Amsterdam, 1984, pp. 1–121.
[30] I. Podlubny, *Fractional Differential Equations*, Academic Press, 1999.
[31] M. Raberto, E. Scalas and F. Mainardi, *Waiting-times and returns in high-frequency financial data: an empirical study*, Physica A, 314 (2002), pp. 749–755.
[32] P. A. Raviart and J. M. Thomas,*A finite element method for second order elliptic problems in mathematical aspects of the finite element method*, Lecture Noets in Math, Springer-Verlag, Berlin, 1977.
[33] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport for a scalar hyperbolic equation*, Research Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[34] G. R. Ritcher, *The discontinuous Galerkin method with diffusion*, Math. Comp., 58 (1992), pp. 631–643.
[35] S. Shen and F. Liu, *Error analysis of an explicit finite difference approximation for the space fractional diffusion equation with insulated ends*, ANZIAM J., 46 (2005), pp. C871–C887.
[36] S. Shen, F. Liu, V. Anh and I. Turner, *The fundamental solution and numerical solution of the Riesz fractional advection-dispersion equation*, IMA J. Appl. Math., 73 (2008), pp. 850–872.
[37] M. F. Shlesinger, G. M. Zaslavsky and J. Klafter, *Strange kinetic*, Nature, 363 (1993), pp. 31–37.
[38] S. H. Shao and H. Z. Tang, *Higher-order accurate Runge-Kutta discontinuous Galerkin methods for a nonlinear Dirac model*, Discrete and Continuous Dynamical Systems, Series B, 6 (2006), pp. 623–640.
[39] C. Tadjeran, M. M. Meerschaert and H-P Scheffler, *A second-order accurate numerical approximation for the fractional diffusion equation*, J. Comput. Phys., 213 (2006), pp. 205–213.
[40] C. Tadjeran and M. M. Meerschaert, *A second-order accurate numerical method for the two-dimensional fractional diffusion equation*, J. Comput. Phys., 220 (2007), pp. 813–823.
[41] H. Z. Tang and G. Warnecke, *A Runge-Kutta discontinuous Galerkin method for the Euler equations*, Computers Fluids, 34 (2005), pp. 375–398.
[42] T. C. Warburton, I. Lomtev, R. M. Kirby and G. E. Karniadakis, *A discontinuous Galerkin method for viscous MHD equations*, J. Comput. Phys., 152 (1999), pp. 1–34.
[43] S. B. Yuste, *Weighted average finite difference methods for fractional diffusion equations*, J. Comput. Phys., 216 (2006), pp. 264–274.
[44] J. Yan and C. W. Shu, *Local discontinuous Galerkin methods for partial differential equations with higher order derivatives*, J. Sci. Comput., 17 (2002), pp. 27–47.