Copyright © 2014 Tech Science Press CMES, vol.97, no.4, pp.299-322, 2014

Differential Quadrature and Cubature Methods for Steady-State Space-Fractional Advection-Diffusion Equations

Guofei Pang[1], **Wen Chen**[1,2] **and K.Y. Sze**[3]

Abstract: Space-fractional advection-diffusion equation is a promising tool to describe the solute anomalous transport in underground water, and it has been extended to multi-dimensions with the help of weighted, fractional directional diffusion operator [Benson, Wheatcraft and Meerschaert (2000)]. Due to the nonlocal property of the space-fractional derivative, it is always a challenge to develop an efficient numerical solution method. The present paper extends the polynomial-based differential quadrature and cubature methods to the solution of steady-state spatial fractional advection-diffusion equations on a rectangular domain. An improved differential cubature method is proposed which accelerates the solution process considerably. Owing to the global interpolation nature these methods are more accurate and efficient than the finite element method. Numerical convergence is investigated thru one- and two- dimensional benchmark problems. The convergence can be improved after well-organized explicit formulas for weighting coefficients are obtained.

Keywords: space-fractional, differential quadrature and cubature, advection-diffusion, finite element.

1 Introduction

Replacement of the second order derivative in the conventional advection-diffusion equation (ADE) by a derivative of order $\alpha \in (1,2)$ yields the fractional diffusion term and thus a fractional advection-diffusion equation (FADE). The super-diffusive flow feature of FADE is suitable for describing contaminant transport in groundwater [Benson, Wheatcraft and Meerschaert (2000); Benson, Schumer,

[1] Department of Engineering Mechanics, Hohai University, Nanjing 210098, China.

[2] Corresponding author. Email: chenwen@hhu.edu.cn

[3] Professor, Department of Mechanical Engineering, The University of Hong Kong, Pokfulam, Hong Kong.

Meerschaert and Wheatcraft (2001); Zhang, Benson and Reeves (2009)]. For 2D and 3D FADE, Meerschaert et al [Meerschaert, Benson and Baumer (1999)] introduced a directionally-weighted fractional diffusion operator D_M^α which is an extension of the well-known fractional Laplacian with M designating a directional probability measure [Chen and Holm (2004); Pang, Chen, and Sze (2013); Duan (2005); Yang, Liu and Turner (2010); Katzav and Adda-bedia (2008)]. The inaccessibility of closed-form analytical solutions of FADE fosters the development of the relevant numerical solution techniques.

Although numerical methods for time-fractional derivative equations are extensively investigated [Zeng, Li, Liu and Turner (2013); Li, Chen and Ye (2011); Wang, Meng, Ma and Wu (2013); Chen, Sun, Li and Fu (2013)], numerical solution of space-fractional derivative equations, especially in high-dimensional case, is less reported. Finite difference method (FDM) [Yang, Liu, and Turner (2010)], finite element method (FEM) [Huang, Huang and Zhan (2008); Zheng, Li and Zhao (2010)], finite volume method [Zhang, Crawford, Deeks, Stutter, Bengough and Young (2005)], matrix transform technique (MTT) [Ilic, Liu, rner and Anh (2005); Ilic, Liu, Turner and Anh (2006)] and spectral methods [Hanert (2010);Li and Xu (2010)] have been employed to solve 1D FADE. In solving FADEs in high spatial dimensions, FDMs [Meerschaert, Scheffler and Tadjeran (2006); Tadjeran and Meerschaert (2007); Wang and Wang (2011)] are commonly used. Unfortunately, they are limited to the fractional differential operators in coordinate directions whilst the generalized operator D_M^α involves the fractional derivative along all directions.

To the best of our knowledge, FEM [Roop (2006)], vector Grünwald formula (VGF) [Meerschaert and Mortenson (2004)] and MTT [Yang, Turner, Liu, and Ilic (2011)] are the available methods for approximating D_M^α Based on the variational statements in [Ervin and Roop (2006)], Roop presented a finite element scheme for fractional advection-diffusion equation with D_M^α and the convergence rate was proven to be equal to or exceed two, i.e., $O(h^d)$ where d ≥2 and h is the grid or element size. Despite the relatively high accuracy, the spatial nonlocality of space-fractional derivative leads to element stiffness matrices which couple all nodal dof. Accordingly, the global stiffness matrix is non-sparse even though the trial functions are local. These significantly increase the assembling and solution time of FEM. The VGF is actually the multi-variate extension of the standard one-variate Grünwald-Letnikov formula [Yang, Liu and Turner (2010)] for fractional derivative and the method has the first-order accuracy, i.e., $O(h)$. MTT is widely employed for approximating fractional Laplacian (-Δ)$^{\alpha/2}$ but may not be a good choice for approximating D_M^α as they have different forms of definitions: the former is defined by the spectral decomposition while the latter is defined by a vector integral.

Pitfalls of the three afore-mentioned methods motivate us to seek more accurate and efficient alternatives Differential quadrature method (DQM) [Shu (2000); Chen, Zhong and Shu (1998)] is well-known for its high accuracy and low computational effort. In the current literature, DQM has only been applied to 1D diffusion equations with time fractional derivative and the DQ formula was confined to approximating the spatial integer-order derivatives [Mokhtazri (2011)]. In this paper we consider the 1D FADE with space-fractional derivative and apply DQ formulas to both spatial integer-order and fractional derivatives. Furthermore we shall apply the differential cubature method (DCM) to solve 2D FADE and go further in developing an improved DCM with better efficiency.

In the integration quadrature method, a weighted sum of the integrands at the quadrature points is taken to approximate an integral. Similarly, in the DQM, the derivative of a given function at each sampling point in a coordinate direction is expressed in terms of a weighted sum of the functional values of all the sampling points along the same direction [Bert and Malik (1996)]. Lagrange interpolants are most commonly used as the test functions of DQM to derive the weighting coefficients yet the Fourier and radial basis functions can also be employed [Shu, Ding and Yeo (2003); Shu and Chew (1997)]. This paper only considers the polynomial test functions. The distribution of sampling points and the accuracy of weighting coefficients computed are critical in the accuracy of the DQM. When the sampling points are Gauss-Chebyshev or Gauss-Legendre-Lobatto points, DQM is equivalent to Chebyshev or Legendre collocation method [Lui (2011)]. Though explicit expressions of the DQ weighting coefficients for integer-order derivatives are available [Quan and Chang (1989); Shu and Richards (1992)], those for fractional derivatives are difficult, if not impossible, to derive. Here, they are obtained numerically by matrix inversion. For a large number of sampling points, the matrix becomes ill-conditioned and the accuracy as well as the convergence of the DQM may be adversely affected It will be seen, however, from the numerical tests that matrix inversion remains to be an acceptable expedient.

Unlike DQM which employs only the sampling points along a coordinate direction, DCM uses the weighted sum of functional values at the sampling points in both coordinate and non-coordinate directions to approximate the derivatives. This method is originally developed for numerical quadrature of multi-dimensional functions and subsequently used to solve high-dimensional differential equations. A comparative study of DQM and DCM for solving conventional advection-diffusion-reaction problems can be found in [Malik and Civan (1995)].

The rest of paper is organized as follows. In Section 2, the DQ, DC and improved DC formulas for fractional directional derivatives are presented. Note that a weighted sum of fractional directional derivatives constitutes the directionally-weighted fractional diffusion operator D_M^α (see Section 3 below). Section 3 briefly reviews the definition of D_M^α, followed by D_M^α's DQ and DC approximations. Section 4 presents some preliminary numerical results for steady-state 1D and 2D FADEs. Section 5 gives some conclusions

2 Approximation for fractional derivatives in a given direction

In this section, DQ and DC formulas are employed to approximate fractional directional derivatives of one- and two-variate functions, respectively. An improved DC formula is developed for a fast computation of weighting coefficients of DCM

2.1 DQ formula

In DQ formula for integer-order derivative, the n-th order derivative is approximated by a linear weighted sum of functional values at sampling points, i.e.

$$D^n u(x)\,|_{x=x_i} = \sum_{k=1}^{N_x} A_{ik}^{(n)} u_k, i=1,...,N_x, x\in[a,b], n=1,2,... \tag{1}$$

where x_i is the sampling point, $u_k = u(x_k)$ and $A_{ik}^{(n)}$ is the weighting coefficient. The sampling points can be equally or unequally spaced [Bert and Malik (1996)]:

$$x_i = \begin{cases} a+\dfrac{i-1}{N_x-1}(b-a) & \text{(equal)} \\ \dfrac{b+a}{2}-\dfrac{(b-a)\cos\left[(i-1)\pi/(N_x-1)\right]}{2} & \text{(unequal)} \end{cases}. \tag{2}$$

Now, consider the fractional-order counterpart of (1), i.e., for $\alpha\in$ $(n$-1$,n]$ and $x\in[ab]$,

$${}^*D_\theta^\alpha u(x)\,|_{x=x_i} = \sum_{k=1}^{N_x} A_{ik}^{(\alpha,\theta)} u_k,\ i=1,2,...N_x, \theta\text{=}0\ or\ \pi \tag{3}$$

In particular, $\theta = 0$ and $\theta = \pi$ define respectively the left and right Caputo fractional differential operators [Podlubny (1999); Samko, Kilbas and Marichey (1993)], i.e.

$$\begin{cases} {}^*D_0^\alpha u(x) = \dfrac{1}{\Gamma(n-\alpha)} \displaystyle\int_0^{x-a} \varsigma^{n-\alpha-1} \frac{d^n u(\tau)}{d\tau^n}\,|_{\tau=x-\varsigma}\, d\varsigma, \\ {}^*D_\pi^\alpha u(x) = \dfrac{1}{\Gamma(n-\alpha)} \displaystyle\int_0^{b-x} \varsigma^{n-\alpha-1} \frac{d^n u(\tau)}{d\tau^n}\,|_{\tau=x+\varsigma}\, d\varsigma. \end{cases} \tag{4}$$

It is trivial that ${}^*D_0^n = \partial^n/\partial x^n, {}^*D_\pi^n = (-1)^n \partial^n/\partial x^n$ [Kilbas, Srivastava and Trujillo (2006)] and $A_{ik}^{(n,0)} = A_{ik}^{(n)}$. In compliance with the fractional differential order in FADE, α falls into the interval (1, 2]. There are two common approaches to derive $A_{ik}^{(n)}$ in (1):

- Numerical approach — numerically solve a Vandermonde system of linear equations formed by replacing u in (1) with monomials;

- Analytical approach — use the Lagrange interpolation of u to derive the explicit formulas.

Using the former approach, the convergence of the DQM is adversely affected by the inherently ill-conditioned Vandermonde matrix as the number of sampling points goes up. The latter approach does not suffer the same pitfall and therefore is always employed for deriving $A_{ik}^{(n)}$ where n is an integer To obtain $A_{ik}^{(\alpha,\theta)}$ for fractional-order α in (3), we first consider the analytical approach. The Lagrange interpolation of u gives

$$\hat{u}(x) = \sum_{k=1}^{N_x} u_k l_k(x) \tag{5}$$

where

$$l_k(x) = \frac{\prod\limits_{j=1, j\neq k}^{N_x} (x - x_j)}{\prod\limits_{j=1, j\neq k}^{N_x} (x_k - x_j)} \tag{6}$$

Differentiation of (5) leads to

$${}^*D_\theta^\alpha \hat{u}\,|_{x=x_i} = \sum_{k=1}^{N_x} u_k {}^*D_\theta^\alpha l_k(x)\,|_{x=x_i}, i = 1, 2..., N_x, \alpha \in (1,2] \tag{7}$$

Comparison of (7) and (3) gives the explicit form of the weighting coefficients:

$$A_{ik}^{(\alpha,\theta)} = {}^*D_\theta^\alpha l_k(x)\,|_{x=x_i} \tag{8}$$

For the trivial case α = 2 and θ = 0, the explicit expression of (8) has been given in [Shu and Richards (1992)]. Since the fractional Leibniz differential rule is complicated [Podlubny (1999)], the deduction of (8) for α∈(1,2) is cumbersome.

A seemingly alternative for determining (8) is the termwise differentiation on $l_k(x)$ which reads

$$ {}^{*}D_{\theta}^{\alpha} l_k(x) = \sum_{p=1}^{N_x} c_p^{(k)}\,{}^{*}D_{\theta}^{\alpha} x^{p-1} \tag{9}$$

where $c_p^{(k)}$ is the coefficient before the monomial x^{p-1} in the expansion of $l_k(x)$. As ${}^{*}D_{\theta}^{\alpha} x^{p-1}$ can be determined easily, the remaining work is to derive the coefficients $c_p^{(k)}$. This can be done by the Matlab function *poly*() which returns the coefficients of a polynomial with zeros $\{x_i\}$. Despite this convenience, our numerical results on 1D problems show that the solution accuracy of DQM based on (9) drops substantially when the number of equispaced sampling points grows, i.e. N_x exceeds 18 (or larger). The convergence is deteriorated by the numerical difficulty of a very large $c_p^{(k)}$ times a very small x^{p-1} for a large N_x in (9).

Noting the limitation of the direct analytical approach, we consider the numerical approach The weighting coefficients are determined by the following linear systems (α∈(1,2))

$$ {}^{*}D_{0}^{\alpha}(x-a)^{p-1}\,|_{x=x_i} = \sum_{q=1}^{N_x} A_{iq}^{(\alpha,0)}(x_q-a)^{p-1},\ i,p = 1,2,\ldots,N_x \tag{10}$$

$$ {}^{*}D_{\pi}^{\alpha}(b-x)^{p-1}\,|_{x=x_i} = \sum_{q=1}^{N_x} A_{iq}^{(\alpha,\pi)}(b-x_q)^{p-1},\ i,p = 1,2,\ldots,N_x \tag{11}$$

Using the basic properties of Caputo fractional derivative the lefthand side of the above equations can be explicitly determined as follows [Podlubny (1999)]

$$ {}^{*}D_{0}^{\alpha}(x-a)^{p-1}\,|_{x=x_i} = \begin{cases} 0,\ p=1,2 \\ \frac{\Gamma(p)}{\Gamma(p-\alpha)}(x_i^{-}a)^{p-\alpha-1}, p\geq 3 \end{cases}, \tag{12}$$

$$ {}^{*}D_{\pi}^{\alpha}(b-x)^{p-1}\,|_{x=x_i} = \begin{cases} 0,\ p=1,2 \\ \frac{\Gamma(p)}{\Gamma(p-\alpha)}(b-x_i)^{p-\alpha-1}, p\geq 3 \end{cases}. \tag{13}$$

From our numerical tests, the weighting coefficients of (3) obtained by solving the linear systems lead to higher convergence than evaluating (9) directly. Thus, the numerical approach will be adopted in the rest of the paper to solve the weighting coefficients.

2.2 DC formula

To consider a rectangular domain Ω=[a,b]$\times$[c,e], (4) is extended to two dimensions as:

$$^{*}D_{\theta}^{\alpha}u(x,y)=I_{\theta}^{2-\alpha}D_{\theta}^{2}u(x,y),\alpha\in(1,2),\theta\in[0,2\pi) \tag{14}$$

in which the fractional directional integral is defined as [Samko, Kilbas, and-Marichey (1993); Pang, Chen, and Sze (2013)], for σ ∈(0,1),

$$I_{\theta}^{\sigma}v(x,y)=\frac{1}{\Gamma(\sigma)}\int_{0}^{d}\varsigma^{\sigma-1}v(x-\varsigma\cos\theta,y-\varsigma\sin\theta)d\varsigma. \tag{15}$$

Here, $D_{\theta}^{2}=(\cos\theta\cdot\partial_x+\sin\theta\cdot\partial_y)^2$ is the second-order directional derivative. The upper integration limit of the integral in (15), i.e. d, is the "backward distance" of internal node (x,y) to the boundary of the problem domain Ω (see Fig.1). This quantity is determined by the node location (xy), the direction vector θ, and the shape of the computational domain.

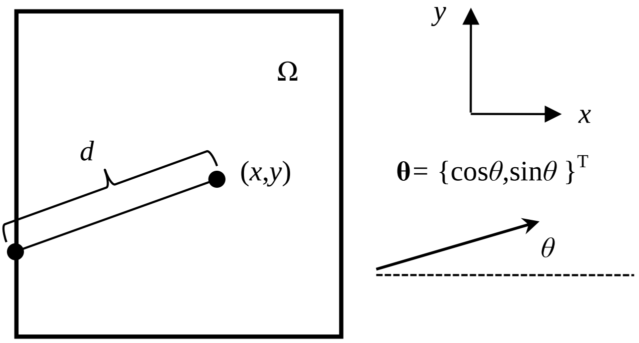

Figure 1: The "backward distance" d of the internal node (x,y) to the boundary of domain Ω in the direction θ.

The DC formula of fractional directional derivative can be written as

$$\begin{aligned}&^{*}D_{\theta}^{\alpha}u(x,y)\left|_{x=x_i,y=y_i}\right.=\sum_{j=1}^{N_xN_y}C_{ij}^{(\alpha,\theta)}u_j,\ i=1,2,...N_xN_y,\\&(x,y)\in[a,b]\times[c,e],\theta\in[0,2\pi),\end{aligned} \tag{16}$$

and the corresponding DC weighting coefficients are determined through solving the following linear system: for $\theta \in (0,2\pi)$,

$$
\begin{aligned}
&{}^*D_\theta^\alpha x^{v-1} y^{\mu-1}|_{x=x_i,y=y_i} = \sum_{j=1}^{N_xN_y} C_{ij}^{(\alpha,\theta)} x_j^{v-1} y_j^{\mu-1}, \\
&i = 1,2,...,N_xN_y; v = 1,2,...,N_x; \mu = 1,2,...,N_y
\end{aligned}
\tag{17}
$$

The subscripts i, j in (16) and (17) are both global subscripts which correspond to a list of grid (or sampling) points $(x_k y_k)$ sorted by a given order. The lefthand side of (17) can be computed by the Gauss-Lobatto-type quadrature rules discussed in our previous work [Pang, Chen and Sze (2013)].

2.3 Improved DC formula

To obtain the weighting coefficients $C_{ij}^{(\alpha,\theta)}$ in (17), one needs to solve N_xN_y linear systems of dimension N_xN_y. Here, we introduce a more efficient procedure which only needs to solve one linear system of dimension N_x or N_y .

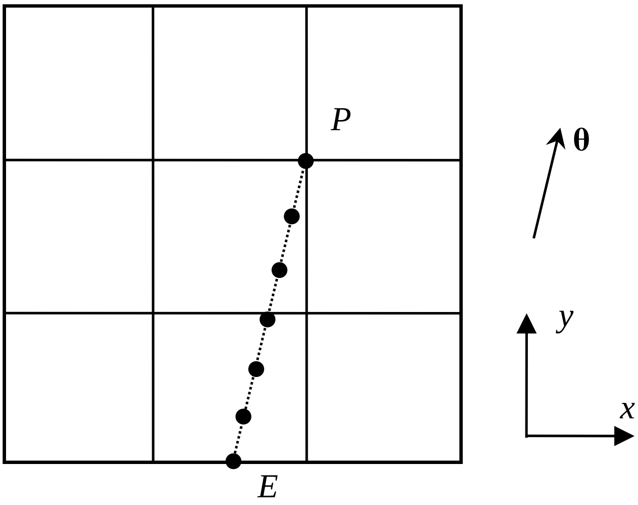

Figure 2: The DQ sampling points ("•") contributing to the approximation of fractional directional derivative evaluated at point P

For simplicity, we consider a rectangular domain modeled by a 4×4 grid as shown in Fig.2 and the backward distance of point P starts from the point E on the lower boundary in direction θ. Along EP, there are N_{DQ} sampling points $(\bar{x}_k, \bar{y}_k)$s which will be termed as the DQ sampling points. The fractional directional derivative at point P is first approximated by a weighted linear sum of functional values $\bar{u}_k$s at the DQ sampling points:

$$^*D_\theta^\alpha u_P = \sum_{k=1}^{N_{DQ}} B_k^{(\alpha,\theta)} \bar{u}_k \quad , \tag{18}$$

where $B_k^{(\alpha,\theta)}$ is the DQ weighting coefficients. Using Lagrange interpolation for a $N_x \times N_y$ grid, the highest order monomial term is $x^{Nx-1}y^{Ny-1}$. Along a straight line such as EP, x and y are linearly related and the highest order term can be expressed as $x^{Nx+Ny-2}$ or $y^{Nx+Ny-2}$ which can be exactly interpolated at $N_x + N_y$-1 DQ sampling points. It thus requires N_{DQ} = 7 for the grid as shown in Fig.2 where $N_x = N_y$ = 4. $B_k^{(\alpha,\theta)}$ can be obtained by $B_k^{(\alpha,\theta)} = l_\theta^{-\alpha} \tilde{A}_k^{(\alpha,\theta)}$ where l_θ = |EP| and the reference DQ weighting coefficient $\tilde{A}_k^{(\alpha,\theta)}$ is solved from (10) by letting a= 0, $N_x = N_{DQ}$ and introducing the normalized coordinate $\tilde{x}_k \in [0,1]$ for the reference sampling points. The ordinates of the reference DQ sampling points along EP are related to the normalized coordinate $\tilde{x}_k \in [0,1]$ through $\bar{y}_k = y_E + (y_P - y_E)\tilde{x}_i$ where y_E and y_P are the ordinates of point E and point P, respectively, as seen in Fig.3.

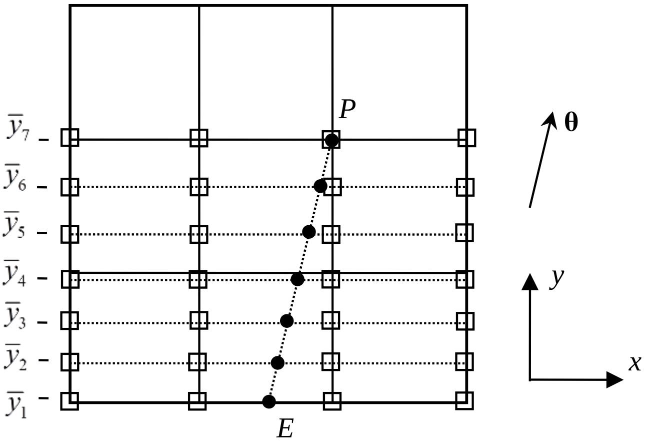

Figure 3: Auxiliary points "□" onto which the DQ weighting coefficients are distributed. "•" denotes the reference sampling point.

Conventionally, one can relate the values of u at the DQ sampling points to those at the grid points by the 2D Lagrange interpolation which involves N_xN_y terms. A more efficient procedure proposed here is to use a two-step 1D interpolation. Considering the i-th DQ sampling point with ordinate $\bar{y}_i$ in Fig.3 as an example, the auxiliary points are the points with the same ordinate $\bar{y}_i$ on the N_x vertical grid lines. In the first step, the values of the u at auxiliary points are first obtained from those at the N_y grid points on the same vertical grid line by the 1D N_y-point Lagrange interpolation. In the second step, the values of u at the N_x auxiliary points are interpolated to the i-th DQ sampling point using the 1D N_x-point Lagrange interpolation.

Finally, the DQ weighting coefficients at sampling points on the line PE are distributed onto all the grid points thru the two-step Lagrange interpolation, thus leading to the modified DC weighting coefficients $\bar{C}_{ij}^{(\alpha,\theta)}$ which are similar to $C_{ij}^{(\alpha,\theta)}$ in (16). In fact, because $C_{ij}^{(\alpha,\theta)}$ and $\bar{C}_{ij}^{(\alpha,\theta)}$ can both be represented in terms of the fractional directional derivative of Lagrange interpolation $l_i(x)l_j(y)$, these two types of weighting coefficients are identical in theory, and therefore the quadrature solution-s of DCM and improved DCM should have been completely identical. However, numerical errors enter differently to $\bar{C}_{ij}^{(\alpha,\theta)}$ and $C_{ij}^{(\alpha,\theta)}$, the pertinent quadrature solutions are different especially when the number of grid points is large. This point will be illustrated in Section/Example 4.2.

Fig.4 shows the auxiliary points when the starting point of the backward distance is on different portion of the domain boundary. In subsequent descriptions, the improved DCM will be abbreviated as iDCM.

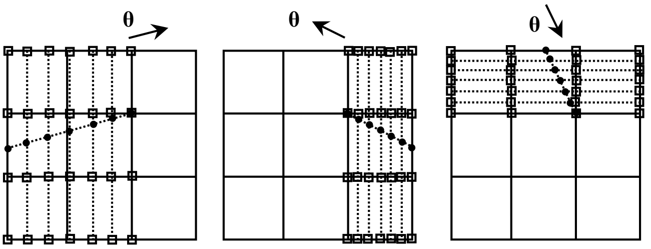

Figure 4: Selections of auxiliary points for other cases of $\theta\in[0,2\pi)$

3 DQ (-DC) formulas for fractional diffusion term

The 2D steady-state FADE can be written as [Meerschaert, Benson and Baumer (1999); Roop (2006); Pang, Chen and Sze (2013)]

$$\begin{aligned} &-cD_M^{\alpha}u(x,y)+\mathbf{b}\cdot\nabla u(x,y)=f(x,y),(x,y)\in\Omega, \\ &u(x,y)\,|_{\partial\Omega}=u_b(x,y),(x,y)\in\partial\Omega \end{aligned} \tag{19}$$

with fractional diffusion operator D_M^α, constant diffusion factor c, constant medium velocity $\mathbf{b}$, source term f and the rectangular domain Ω. D_M^α in R^2 is defined as [Meerschaert, Benson and Baumer (1999)]

$$D_M^\alpha(\cdot) = \int_0^{2\pi} D_\theta^\alpha(\cdot) M(d\theta)(1 < \alpha < 2) \tag{20}$$

with the probability measure $M(d\theta)$ which can be discrete or continuous [Roop (2006); Pang, Chen and Sze (2013)]:

$$D_M^\alpha(\cdot) = \sum_i D_{\theta_i}^\alpha(\cdot) m_i\,,\ \ 1 < \alpha < 2,\ \sum_i m_i = 1 \tag{21}$$

$$D_M^\alpha(\cdot) = \int_0^{2\pi} D_\theta^\alpha(\cdot) m(\theta) d\theta,\ \ 1 < \alpha < 2,\ \int_0^{2\pi} m(\theta) d\theta = 1. \tag{22}$$

The directional differential operator $D_\theta^\alpha := D_\theta^2 I_\theta^{2-\alpha}$ here belongs to the Riemann-Liouville type. For computational convenience, we consider its Caputo counterpart ${}^*D_\theta^\alpha := I_\theta^{2-\alpha} D_\theta^2$. Accordingly, we replace the fractional diffusion operator in (19) with the following operator

$${}^*D_M^\alpha(\cdot) = \int_0^{2\pi} {}^*D_\theta^\alpha(\cdot) M(d\theta)(1 < \alpha < 2). \tag{23}$$

For a one-dimensional operator, (21) reduces to

$${}^*D_M^\alpha u(x) = (m_1 {}^*D_0^\alpha + (1 - m_1) {}^*D_\pi^\alpha)\, u(x), m_1 \in [0,1], \tag{24}$$

and from (3) its DQ formula is given by

$$\begin{array}{l} {}^*D_M^\alpha u(x)\,|_{x=x_i} = \sum\limits_{k=1}^{N_x} (m_1 A_{ik}^{(\alpha,0)} + (1 - m_1) A_{ik}^{(\alpha,\pi)})\, u_k, \\ i = 1, 2, \ldots, N_x \end{array} \quad . \tag{25}$$

For a two-dimensional continuous operator in (22), numerical quadrature can be used to approximate (23), i.e.

$${}^*D_M^\alpha u(x,y) = \sum_{k=0}^{N-1} w_k m(\theta_k) {}^*D_{\theta_k}^\alpha u(x,y) \tag{26}$$

where θ_k and w_k are the quadrature point and weight, respectively. The corresponding DC and improved DC formulas are

$$^{*}D_{M}^{\alpha}u(x,y)\,|_{x=x_i,y=y_i}=\begin{cases}\sum\limits_{k=0}^{N-1}w_k m(\theta_k)\sum\limits_{j=1}^{N_xN_y}C_{ij}^{(\alpha,\theta_k)}u_j\\ \sum\limits_{k=0}^{N-1}w_k m(\theta_k)\sum\limits_{j=1}^{N_xN_y}\bar{C}_{ij}^{(\alpha,\theta_k)}u_j\end{cases}\tag{27}$$

where i=1,2,...N_xN_y. Similar formulas can also be found for discrete operator in (21). For convenience, we use compound trapezoidal rule here by letting w_k =2π/N and θ_k= 2$k\pi$/N where N is the number of integration intervals with respect to θ

4 Numerical results and discussions

4.1 One-dimensional advection-diffusion problem

$$-^{*}D_{M}^{1.5}u(x)+\frac{du(x)}{dx}=f(x),x\in[0,1],\tag{28}$$

with exact solution

(a) $u(x)=x^8x^5$+3x^2+5 ;

(b) $u(x)=e^x\cos(x)+x^3$

Here, $^{*}D_{M}^{1.5}$ is prescribed to be $0.5(^{*}D_{0}^{1.5}+{}^{*}D_{\pi}^{1.5})$ and the Dirichlet boundary condition is taken; the source term $f(x)$ is obtained by the 10-point Gauss-Jacobi-Lobatto quadrature as given in [Pang, Chen and Sze (2013)]. The first-order derivative in (28) is approximated by conventional DQ formula, i. e.

$$\frac{du(x)}{dx}\,|_{x=x_i}=\sum_{i=1}^{N_x}A_{ik}^{(1,0)}u_k\tag{29}$$

where the weighting coefficients are given by [Shu and Richards (1992)] :

$$\begin{cases}A_{ik}^{(1,0)}=\dfrac{\prod\limits_{v=1,v\neq i}^{N_x}(x_i-x_v)}{(x_i-x_k)\prod\limits_{v=1,v\neq k}^{N_x}(x_k-x_v)}\\ for\ i,k=1,2,...,N_x\ \text{ and }\ k\neq i\ \text{(off-diagonal)},\\ A_{ii}^{(1,0)}=-\sum\limits_{v=1,v\neq i}^{N_x}A_{iv}^{(1,0)}\ \text{ for }\ i=1,2,...,N_x\ \text{(diagonal).}\end{cases}\tag{30}$$

We use DQ formula (25) to approximate ${}^{*}D_{M}^{1.5}u(x)$ choose five test points z=0.1,0.3,0.5,0.7 and 0.9, and define the normalized error as

$$Err = \frac{\sqrt{\sum\limits_{t=1}^{5}\left(\tilde{u}_i - u(z_i)\right)^2}}{\sqrt{\sum\limits_{t=1}^{5}u(z_i)^2}} \tag{31}$$

where the approximant $\tilde{u}_i$ at z_i is determined by Lagrange interpolation, i.e. (5). Two types of sampling points (i=1,2,..., N_x) are considered in the present example:

(1) Equally spaced points $x_i = \frac{i-1}{N_x-1}$;

(2) The unequally spaced Gauss-Chebyshev-Lobatto points x_i=$\frac{1}{2} - \frac{1}{2}\cos\left(\frac{i-1}{N_x-1}\pi\right)$.

Figs.5 and 6 plot the errors for cases (a) and (b), respectively.

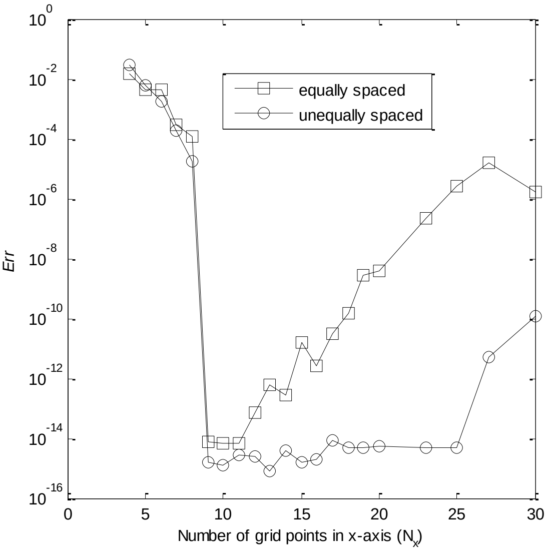

Figure 5: Convergence of the DQ solutions of the Example 4.1 (a)

Same as in integer-order derivative problems, it can be seen that DQM can achieve the spectral accuracy for fractional derivative problem and unequally spaced sampling points can delay the numerical errors. This is because the Vandermenda matrix system for determining DQ weighting coefficients generated by non-uniform sampling points is less ill-conditioned than that generated by uniform sampling

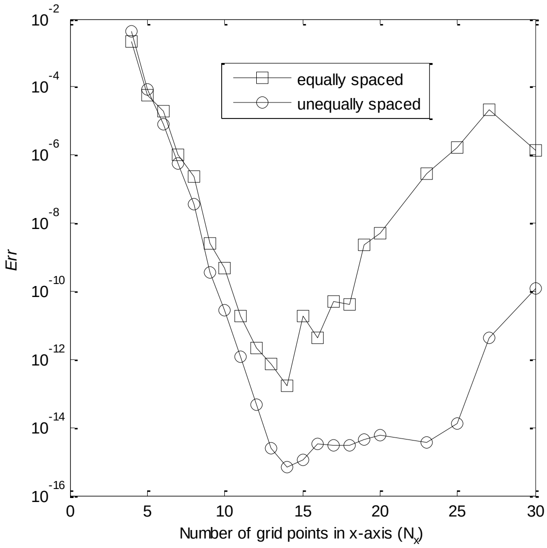

Figure 6: Convergence of the DQ solutions of Example 4.1 (b)

points. Despite this advantage, the machine precision $\sim 10^{-15}$ for double precision starts to lose at $N_x \approx 25$. On the other hand, for a numerical solution method having spectral accuracy, a less dense grid is often adequate for most problems..

4.2 Two-dimensional homogenous diffusion problem

$$-\frac{1}{2\pi}\int_{0}^{2\pi} {}^{*}D_{\theta}^{1.5}u(x,y)d\theta = f(x,y), (x,y) \in \Omega = [0,1]^2 \tag{32}$$

with exact solution

(a) $u(xy)$=$x^2(1$-$x)^2y^2(1$-$y)^2$

(b) $u(xy)$=$x^3\cos(x^2+y^2)+e^{x+2y}$

and source term ($v = D_{\theta}^{2}u$)

$$f(x,y) = -\frac{1}{2\pi}\int_{0}^{2\pi}\left[\frac{1}{\Gamma(0.5)}\int_{0}^{d}\varsigma^{-0.5}v(x-\varsigma\cos\theta, y-\varsigma\sin\theta)d\varsigma\right]d\theta \tag{33}$$

The Dirichlet boundary condition is taken. To compute the value of f at each grid point, integration quadrature is implemented. The backward distance d in (33) can be derived by [Roop (2006)]:

$$d(x,y,\theta)=\begin{cases}\dfrac{x}{\cos\theta},\theta\in\left[0,\arctan\left(\dfrac{y}{x}\right)\right),\\ \dfrac{y}{\sin\theta},\theta\in\left[\arctan\left(\dfrac{y}{x}\right),\pi+\arctan\left(\dfrac{y}{x-1}\right)\right),\\ \dfrac{x-1}{\cos\theta},\theta\in\left[\pi+\arctan\left(\dfrac{y}{x-1}\right),\pi+\arctan\left(\dfrac{y-1}{x-1}\right)\right),\\ \dfrac{y-1}{\sin\theta},\theta\in\left[\pi+\arctan\left(\dfrac{y-1}{x-1}\right),2\pi+\arctan\left(\dfrac{y-1}{x}\right)\right),\\ \dfrac{x}{\cos\theta},\theta\in\left[2\pi+\arctan\left(\dfrac{y-1}{x}\right),2\pi\right).\end{cases}\tag{34}$$

The integral with respect to θ is partitioned into five sub-integrals according to (34). High-order Gauss-Jacobi-type quadrature rules [Pang, Chen and Sze (2013)] are used to evaluate each sub-integral for a highly accurate source term f.

Here, 81 test points with coordinates (x_iy_j)=(i/10, j/10), i,j=1,2,...,9 are employed to quantify the normalized error which is defined as

$$Err=\frac{\sqrt{\sum\limits_{t=1}^{9}\sum\limits_{j=1}^{9}(\tilde{u}_{ij}-u(x_i,y_j))^2}}{\sqrt{\sum\limits_{t=1}^{9}\sum\limits_{j=1}^{9}u(x_i,y_j)^2}}\tag{35}$$

where the approximant $\tilde{u}_{ij}$ are derived by 2D Lagrange interpolation at the gird points.

We first consider the case (a) with exact solution $x^2(1\text{-}x)^2y^2(1\text{-}y)^2$. Roop's FEM in [Roop (2006)] is taken as a comparison. Four-node rectangular elements and uniformly spaced nodes are employed In iDCM, if the grid points are equally (unequally) distributed, the DQ sampling points would also be equally (unequally) distributed. Other combinations of grid point and DQ sampling point distributions are possible but the numerical solutions are plagued by numerical errors more readily as the grid density increases. Fig.7 compares the accuracy of FEM, DCM and iDCM. Let N=64 in (26) for DCM and iDCM. Fig.8 compares the CPU time (using an Intel i5 dual-core CPU and 2G RAM) for computing the weighting coefficient matrix of (improved) DCM with the CPU time for generating the global stiffness matrix in FEM. Here, $N_x=N_y=M$ which vary from 4 to 20. Note that the inherent ill-conditioning of the system matrix in (17) simply allows a less dense grid (i.e., a small N_x and N_y) when one needs acceptable solution accuracy.

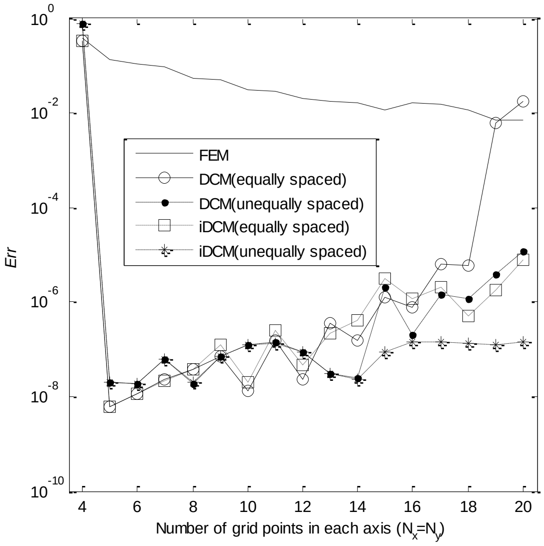

Figure 7: Convergence of the FEM, DCM and iDCM (N_{DQ}= 12) solutions of Example 4.2(a)

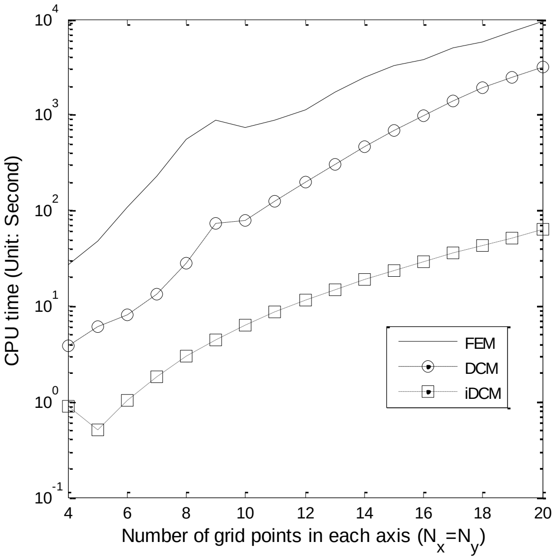

Figure 8: CPU time for generating discretized matrices of FEM, DCM and iDCM (N_{DQ}= 12) in Example 4.2(a)

It can be seen from Fig.7 that DCM and iDCM at small N possess the spectral convergence which is much higher than that of FEM. Moreover, the non-uniform grid yields more stable solutions. As discussed in Section 2.3, for small N_x or N_y, DCM and iDCM yield identical solutions; while for large N_x or N_y, their solutions exhibit obvious distinction. The computational efficiency of the improved DCM over DCM can be seen in Fig.8

It has been discussed in Section 2.3 that for a $N_x \times N_y$ grid, the number of DQ sampling points, i.e., N_{DQ}, in the iDCM is sensible to take $N_x + N_y$-1 based on the interpolation consideration. On the other hand, if N_{DQ} is too large, the Vandermenda matrix may be ill-conditioned and adversely affect the accuracy and convergence. With reference to Example 4.2(a), the exact solution can be exactly interpolated by a 5×5 grid when $N_{DQ} \geq 9$. This can be seen by an obvious increase of the accuracy for N_x $(= N_y) \geq 5$ and $N_{DQ} \geq 9$ as shown in Fig.9.

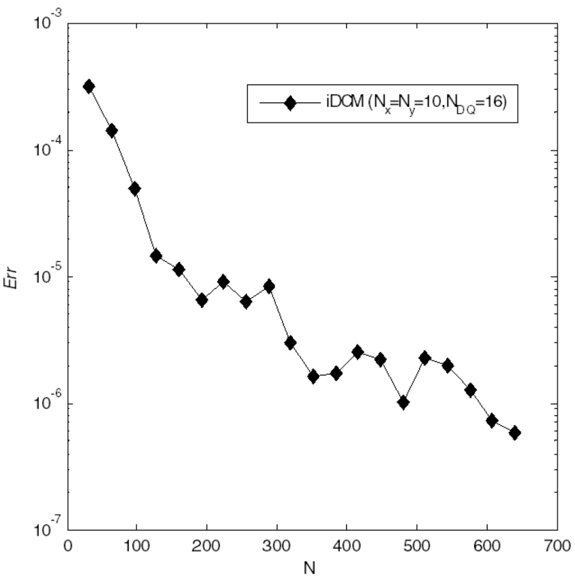

Figure 9: Convergence of the improved DCM for different N_{DQ} with unequally spaced gird points in Example 4.2(a)

Next, we consider case (b) with exact solution $x^3\cos(x^2 + y^2) + e^{x+2y}$. Since the finite element scheme for the 2D FADE (19) is only developed for a problem with homogenous Dirichlet boundary condition [Roop (2006)], we will not consider the FEM solution for (b) which does not vanish identically over the domain boundary. The comparison of the solutions from the DCM and the iDCM (N_{DQ}=12) for N=

64 in (26) is given in Fig.10. It can be seen that accurate and stable solutions can still be secured by the iDCM with a unequispaced grid.

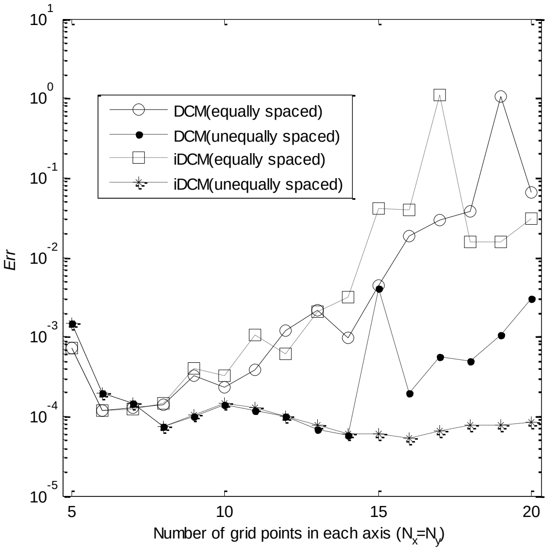

Figure 10: Convergence of the DCM and iDCM (N_{DQ}=12) solutions of Example 4.2(b)

Now discuss the influence of N_{DQ} on the solution accuracy of the iDCM. Fig.11 displays the variation of the normalized error of the iDCM with the N_{DQ}. The observations show that the influence of N_{DQ} can only be clearly seen for a larger N_x or N_y and an increasing N_{DQ} will not obviously improve the solution accuracy. We conjecture that two factors, i.e., (a) the inherently ill-conditioning of the linear system that determines the DQ weighting coefficients and (b) the numerical errors accumulated in the two-step 1D Lagrange interpolation deteriorate the solution convergence. From the interpolation consideration in Section 2.3, a possible way to find an optimal N_{DQ} is to let N_{DQ} vary with the gird size, namely, $N_{DQ} = N_x + N_y$-1. The corresponding convergence curve is shown in Fig.12, from which one can see the poor accuracy for a larger N_x or N_y. Comparison of Figs. 11 and 12, however, indicates that using a constant N_{DQ} is usually more suitable than using a variable one, but for a smaller N_x or N_y, both cases are available. Considering the influence of constant N_{DQ} as shown in Fig.11, we suggest a constant $N_{DQ} \in$[7 min{18,$N_x + N_y$-1}] for $N_xN_y \geq$4

It is worth noting that the integration quadrature accuracy in (26) also plays a crucial role in the final solution accuracy. This will be true for both the case (a) and

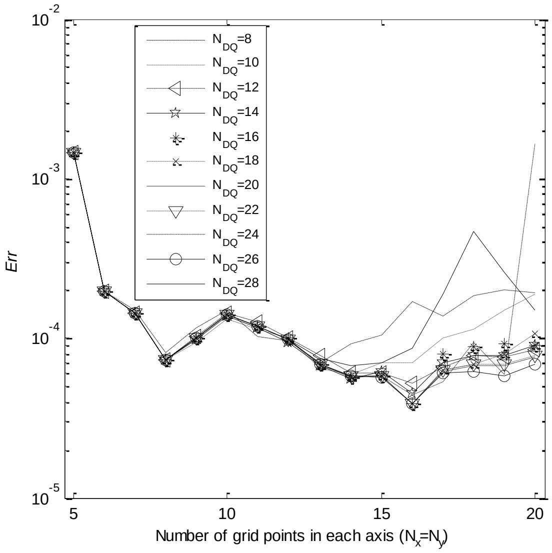

Figure 11: Convergence of iDCM for constant N_{DQ} with unequally spaced grid points in Example 4.2(b)

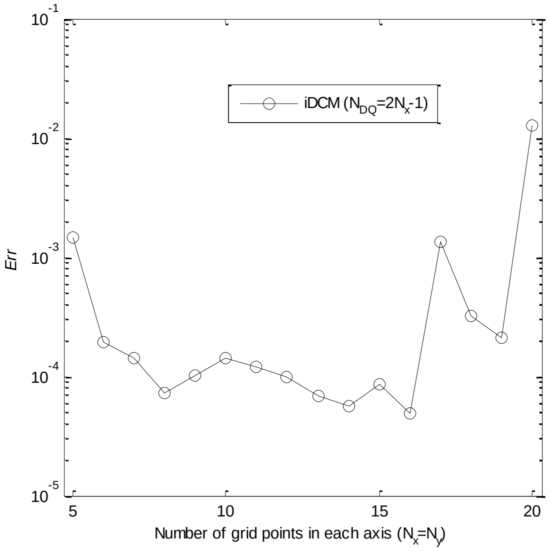

Figure 12: Convergence of iDCM for variable $N_{DQ} = N_x + N_y$-1 with unequally spaced grid points in Example 4.2(b).

case (b). For instance, the variation of normalized error *Err* with the interval number N in compound trapezoidal rule for case (b) where $N_x = N_y$= 10 and N_{DQ}=16 $\in$ [7, 18] is illustrated in Fig.13. The solution accuracy will gradually increase with the increasing N. Consideration of both the accuracy and the CPU time leads to an intermediate but acceptable N. Here, we took N= 64 just as mentioned before

Thru numerical tests, it is interesting to find that using higher-order quadrature rules such as compound Simpson and even the (piecewise) Gauss quadrature rules will not obviously improve the convergence in Fig.13. The is also the reason why we use compound trapezoidal rule instead of high-order rules to discretize the angle integral in (26) and (27).

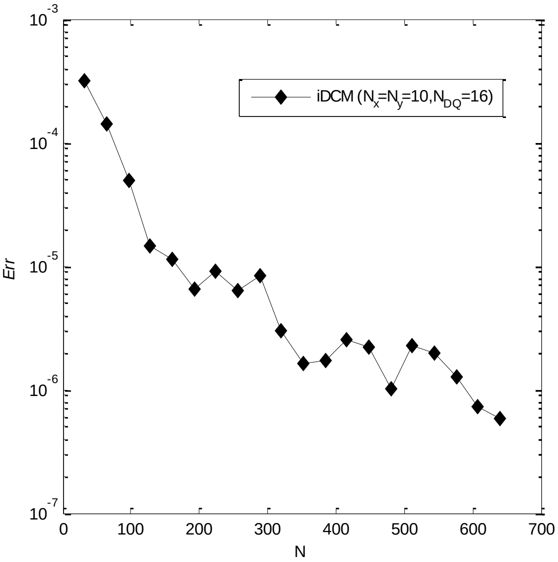

Figure 13: Variation of the normalized error with the number of the integration interval in compound trapezoidal rule using nonuniform grid for Example 4.2(b)

5 Conclusions

This paper presents differential quadrature and cubature methods for numerical solution of steady-state advection-diffusion equations with directionally-weighted fractional diffusion operator. The present methods feature spectral convergence and enjoy lower computational effort compared to the finite element method. An improved version of the original differential cubature method is proposed to reduce the computational cost considerably On the other hand, the weighting coefficients derived from solving the inherently ill-conditioned linear system fail these methods in large-scale problems. A partial antidote to this problem is to deduce explicit expressions for DQ/DC weighting coefficients that take the similar form to (30), which is under study now.

Acknowledgement: The research is financially supported by the Hong Kong Research Grant Council (GRF grant HKU 7173 09E), the 111 project under grant B12032, the National Basic Research Program of China (973 Project No. 2010CB832702), and the National Science Funds for Distinguished Young Scholars (11125208).

References

Benson, D. A.; Schumer, R.; Meerschaert, M. M.; Wheatcraft, S. W. (2001): Fractional diffusion, Levy motions, and the MADE tracer tests. *Transp. Porous Media*, vol. 42, pp. 211-240.

Benson, D. A.; Wheatcraft, S.; Meerschaert, M. M. (2000): Application of a fractional advection-diffusion equation. *Water Resour. Res.*, vol. 36, pp. 132-138.

Bert, C. W.; Malik, M. (1996): Differential quadrature method in computational mechanics: A review. *Appl. Mech. Rev.*, vol. 49, pp. 1-28.

Chen, W.; Holm, S. (2004): Fractional Laplacian time-space models for linear and nonlinear lossy media exhibiting arbitrary frequency power-law dependency. *J. Acoust.Soc.Am.*, vol.115, pp. 1424-1430.

Chen, Y.; Sun, L.; Li, X.; Fu, X. (2013): Numerical solution of nonlinear fractional integral differential equations by using the second kind Chebyshev wavelets. *CMES: Computer Modeling in Engineering & Sciences*, vol. 90, pp. 359-378.

Chen, W.; Zhong, T.; Shu, C. (1998): A Lyapunov formulation for efficient solution of the Poisson and convection-diffusion equations by the differential quadrature method, *J. Comput. Phys.*, vol. 141, pp. 78-84.

Duan, J. S. (2005): Time- and space-fractional partial differential equations. *J. Math. Phys.*, vol.46, pp. 13504-13511.

Ervin, V. J.; Roop, J. P. (2006): Variational solution of fractional advection diffusion equation on bounded domains in R^d. *Numer. Meth. Part Differ. Equ.*, vol. 23, pp. 256-281.

Hanert, E. (2010): A comparison of three Eulerian numerical methods for fractional-order transport models. *Environmental Fluid Mechanics*, vol. 10, pp. 7-20.

Huang, Q. Z; Huang, G. H.; Zhan, H. B. (2008): A finite element solution for the fractional advection-diffusion equation. *Advances in Water Resources*, vol. 31, pp. 1578-1589.

Ilic, M.; Liu, F. W.; Turner, I.; Anh, V. (2005): Numerical approximation of a fractional-in-space diffusion equation, I. *Fractional Calculus & Applied Analysis*, vol. 8, pp. 323-341.

Ilic, M.; Liu, F. W.; Turner, I.; Anh, V. (2006): Numerical approximation of a fractional-in-space diffusion equation (II) – with nonhomogeneous boundary conditions. *Fractional Calculus & Applied Analysis*, vol. 9, pp. 333-349.

Katzav, E.; Adda-bedia, M. (2008): The spectrum of the fractional Laplacian and First-Passage-Time statistics. *Europhysics Letters*, vol.83, pp.30006.

Kilbas, A. A.; Srivastava, H. M.; Trujillo, J. J. (2006): *Theory and applications of fractional differential equations*. Elsevier B.V..

Li, C. P.; Chen, A.; Ye, J. J. (2011): Numerical approaches to fractional calculus and fractional ordinary differential equation. *Journal of Computational Physics*, vol. 230, pp. 3352-3368.

Li, X. J.; Xu, C. J. (2010): Existence and uniqueness of the weak solution of the space-time fractional diffusion equation and spectral method approximation. *Commun. Comput. Phys.*, vol. 8, pp. 1016-1051.

Lui, S. H. (2011): *Numerical analysis of partial differential equations*. WELEY.

Malik, M.; Civan, F. (1995): A comparative study of differential quadrature and cubature methods vis-à-vis some conventional techniques in context of convection-diffusion-reaction problems. *Chemical Engineering Science*, vol. 50, pp. 531-547.

Meerschaert, M. M.; Benson, D. A.; Baumer, B. (1999): Multidimensional advection and fractional diffusion. *Phys. Rev. E*, vol.59, pp.5026-5028.

Meerschaert, M. M.; Mortenson, J. (2004): Vector Grünwald formula for fractional derivatives. *Fract. Calc. Appl. Anal.*, vol. 7, pp. 61-81.

Meerschaert, M. M.; Scheffler, H. P.; Tadjeran, C. (2006): Finite difference methods for two-dimensional fractional diffusion equation. *J. Comput. Phys.*, vol. 211, pp. 249-261.

Mokhtari, R. (2011): GDQM for solving the fractional diffusion equation, *Communications in Fractional Calculus*, vol. 2, pp. 1-13.

Pang, G. F.; Chen, W.; Sze, K. Y. (2013): Gauss-Jacobi-type quadrature rules for fractional directional integrals. *Computation and Mathematics with Applications*, vol. 66, pp. 597-607.

Podlubny, I. (1999): *Fractional differential equations: An introduction to Fractional Derivatives, Fractional Differential Equations, to methods of their Solution* *and some of their Applications*. ACADEMIC PRESS.

Quan, J. R.; Chang, C. T. (1989): New insights in solving distributed system equations by the quadrature method-I. Analysis. *Computers Chem. Engng.*, vol. 13, pp. 779-788.

Roop, J. P. (2006): Computational aspects of FEM approximation of fractional advection diffusion equations on boundary domains in R^2. *J. Comput. Appl. Math.*, vol. 193, pp. 243-268.

Samko, S. G.; Kilbas, A. A.; Marichev, O. I. (1993): *Fractional integrals and derivatives: Theory and Applications*. Gordon and Breach, Newark, N.J..

Shu, C. (2000): *Differential quadrature and its application in engineering*. Springer-Verlag London Limited.

Shu, C.; Chew, Y. T. (1997): Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems. *Commun. Numer. Meth. Engrg.*, vol. 13, pp. 643-653.

Shu, C.; Ding, H.; Yeo, K. S. (2003): Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, vol. 192, pp. 941-954.

Shu, C.; Richards, B. E. (1992): Application of generalized differential quadrature to solving two-dimensional incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, vol. 15, pp. 791-798.

Shu, C.; Yao, Q.; Yeo, K. S. (2002): Block-marching in time with DQ discretization: an efficient method for time-dependent problem. *Comput. Methods Appl. Mech. Engrg.*, vol. 191, pp. 4587-4597.

Tadjeran, C.; Meerschaert, M. M. (2007): A second-order accurate numerical method for the two-dimensional fractional diffusion equation. *J. Comput. Phys.*, vol. 220, pp. 813-823.

Wang, L.; Meng, Z.; Ma, Y.; Wu, Z. (2013): Numerical solution of fractional partial differential equations using Haar wavelets. *CMES: Computer Modeling in Engineering & Sciences*, vol. 91, pp. 269-287.

Wang, H.; Wang, K. X. (2011): An O ($N\log^{2N}$) alternating-direction finite difference method for two-dimensional fractional diffusion equations. *J. Comput. Phys.*, vol. 230, pp. 7830-7839.

Yang, Q. Q.; Liu, F. W.; Turner, I. (2010): Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. *Applied Mathematical Modeling*, vol.34, pp.200-218.

Yang, Q. Q.; Turner, I.; Liu, F. W.; Ilic, M. (2011): Novel numerical methods for solving the time-space fractional diffusion equation in two dimensions *SIAM J.SCI. COMPUT.*, vol. 33, pp. 1159-1180.

Zhang, Y.; Benson, D. A.; Reeves, D. M. (2009): Time and space nonlocalities underlying fractional-derivatives models: Distinction and literature review of field applications *Advances in Water Resources*, vol. 32, pp. 561-581.

Zhang, X.; Crawford, J. W.; Deeks, L. K.; Stutter, M. I.; Bengough, A. G.; Young, I. M. (2005): A mass balance based numerical method for the fractional advection diffusion equation: theory and application. *Water Resour. Res.*, vol. 41, pp. W07029.

Zeng, F. H.; Li, C. P.; Liu, F. W.; Turner, I. (2013): The use of finite difference /element approaches for solving the time-fractional subdiffusion equation. *SIAM J. Sci. Comput.*, vol. 35, pp. A2976-A3000.

Zheng, Y. Y.; Li, C. P.; Zhao, Z. G. (2010): A note on the finite element method for the space-fractional advection diffusion equation. *Computers and Mathematics with Applications*, vol. 59, pp. 1718-1726.