Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **8**, No. 1, pp. 166-186

DOI: 10.4208/aamm.2014.m693
February 2016

A Comparative Study of Finite Element and Finite Difference Methods for Two-Dimensional Space-Fractional Advection-Dispersion Equation

Guofei Pang[1], Wen Chen[1,*] and Kam Yim Sze[2]

[1] *Institute of Soft Matter Mechanics, Department of Engineering Mechanics, Hohai University, Nanjing 210098, China*
[2] *Department of Mechanical Engineering, The University of Hong Kong, Pokfulam, Hong Kong*

Received 13 July 2014; Accepted (in revised version) 19 January 2015

Abstract. The paper makes a comparative study of the finite element method (FEM) and the finite difference method (FDM) for two-dimensional fractional advection-dispersion equation (FADE) which has recently been considered a promising tool in modeling non-Fickian solute transport in groundwater. Due to the non-local property of integro-differential operator of the space-fractional derivative, numerical solution of FADE is very challenging and little has been reported in literature, especially for high-dimensional case. In order to effectively apply the FEM and the FDM to the FADE on a rectangular domain, a backward-distance algorithm is presented to extend the triangular elements to generic polygon elements in the finite element analysis, and a variable-step vector Grünwald formula is proposed to improve the solution accuracy of the conventional finite difference scheme. Numerical investigation shows that the FEM compares favorably with the FDM in terms of accuracy and convergence rate whereas the latter enjoys less computational effort.

AMS subject classifications: 26A33, 65D32, 65N06, 65N30

Key words: Space-fractional derivative, advection-dispersion, finite element, finite difference.

1 Introduction

Fractional derivative models have been extensively investigated in recent decades for describing the anomalous diffusion or dispersion [1, 2], energy dissipation of vibration and wave [3, 4], and dynamic system [5], with fewer parameters than the classical models of integer-order derivative. A state-of-the-art review of applications of fractional calculus on solid and fluid mechanics can be found in monographs [6, 7]. Due to the

*Corresponding author.
Email: chenwen@hhu.edu.cn (W. Chen)

 ©2016 Global Science Press

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

integro-differential and convolution expression of the fractional derivative, the analytical solution of fractional derivative equations are not always obtainable, especially for high-dimensional irregular computing domain and complex boundary conditions.

Recent decade has witnessed fast growing developments of numerical methods for fractional derivative equations. In comparison with scalar time-fractional derivative, space-fractional derivative, particularly the fractional Laplacian [8, 9], is of more difficulty for discretization due to its vector integral expression. It is stressed that the differentiation directions of the space-fractional derivative are usually confined to coordinate axes in most of the existing literature [10–12]. However, for the FADE model of our interest in describing the multidimensional non-Fickian solution transport little has been reported on the consideration of the non-coordinate differentiation directions of the space-fractional derivative. It is significant to consider the non-coordinate derivative in the FADE in order to generate a full family of multivariable Lévy stable laws that underlie particle random walks with occasional large jumps [13, 14]. The motivation of this study is to seek effective numerical methods for discretizing the space-fractional derivative having non-coordinate differentiation directions.

FEM and FDM have long been considered well-known mesh-based approximation methods for solving a tremendous amount of engineering and scientific problems. Great effort has been made to apply these two methods to fractional models with coordinate-directed space-fractional derivative [10–12, 15–18]. Nevertheless, special care should be taken to the FADE with non-coordinate-directed derivative due to the convolution characteristic and the direction dependence of the space-fractional derivative. Roop [19] has presented for the FADE a finite element scheme based on the variational statements derived in [20], but the scheme is confined to the use of triangular elements and will be difficult to extend to rectangular elements because of the complicated mathematical analysis arising from the non-coordinate derivative. Note that rectangular elements are usually preferred for rectangular computing domains. Meerschaert et al. [21] proposed a vector Grünwald formula (VGF), a type of finite difference scheme, for discretizing the space-fractional derivative on an infinite domain. But due to taking fixed spatial steps, this formula will become inaccurate for a finite computing domain.

In this study, we make a comparative study of the FEM and the FDM for two-dimensional FADE. A backward-distance algorithm is presented to diversify the types of elements that can be used in the finite element analysis, and a variable-step VGF is proposed to improve the solution accuracy of the conventional VGF. Unless otherwise stated, we call the backward-distance-algorithm based FEM the FEM and the variable-step VGF the FDM for simplicity. In the FEM, the backward-distance algorithm is intended for all types of polygon elements, such as triangle, rectangle, and parallelogram. The algorithm can automatically derive the desired distance irrespective of the relative locations of the quadrature point and the finite element. The FDM adopts variable spatial steps to guarantee more grid points are selected in arbitrary differentiation directions of the space-fractional derivative. This avoids the accuracy decrease due to very few computing points taken along certain differentiation direction.

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

The rest of the paper is organized as follows. Section 2 introduces the FADE and the space-fractional derivative therein. The underlying ideas of the backward distance algorithm and the variable-step method are elaborated in Sections 3 and 4, respectively. Tested examples are given and discussed in Section 5 to show the merits and drawbacks of the FEM and the FDM. Concluding remarks are given in the final section.

2 Fractional advection-dispersion equation

Consider the two-dimensional FADE of the form [19,22]

$$\frac{\partial u(x,y,t)}{\partial t} - cD_M^{\alpha} u(x,y,t) + \mathbf{b}\cdot\nabla u(x,y,t) = f(x,y,t), \quad (x,y)\in\Omega, \tag{2.1}$$

subject to the initial value condition $u(x,y,0)=u_0(x,y)$ and the homogeneous Dirichlet boundary condition $u(x,y,t)\equiv 0$ for $(x,y)\in\partial\Omega$. Here u is the concentration, c the dispersion coefficient, $\mathbf{b}$ the medium velocity, ∇ the gradient operator, f the source term, and Ω a rectangular domain. The fundamental solution of the Eq. (2.1) is the probability density function of the multivariable Lévy stable distribution, and thus the equation can describe the particle random walks with occasional large jumps which has long been observed in the non-Fickian solute transport in groundwater [13, 14]. The homogeneous boundary condition is taken assuming that the solute have not yet reached the domain boundary during the observation. The space-fractional derivative D_M^{α} $(\alpha\in(1,2])$ can be considered the weighted sum of the fractional directional derivative D_{θ}^{α} as follows [19]

$$D_M^{\alpha}u(x,y) = \begin{cases} \displaystyle\int_0^{2\pi} {D_{\theta}}^{\alpha} u(x,y)m(\theta)d\theta, & \text{(continuous or non-coordinate case)},\\ \displaystyle\sum_i {D_{\theta_i}}^{\alpha} u(x,y)m_i, & \text{(discrete or coordiante case)}. \end{cases} \tag{2.2}$$

The θ or θ_i indicates the differential direction, and as θ or θ_i equals 0 and π, the directional derivative D_{θ}^{α} or $D_{\theta_i}^{\alpha}$ recovers to the left-sided and right-sided Riemann-Liouville fractional derivative [24], respectively. The weight $m(\theta)$ or m_i represents the probability of a dispersion occurring in the direction θ or θ_i. The fractional directional derivative takes the form of

$$D_{\theta}^{\alpha}u(x,y) = D_{\theta}^2 I_{\theta}^{2-\alpha}u(x,y), \tag{2.3}$$

where the second-order directional derivative reads

$$D_{\theta}^2 := \left(\cos\theta\frac{\partial}{\partial x} + \sin\theta\frac{\partial}{\partial y}\right)^2, \tag{2.4}$$

and the fractional directional integral is given by

$$I_{\theta}^{\sigma}v(x,y) = \frac{1}{\Gamma(\sigma)}\int_0^{d(x,y,\theta)} \varsigma^{\sigma-1}v(x-\varsigma\cos\theta, y-\varsigma\sin\theta)d\varsigma, \quad \sigma\in(0,1), \tag{2.5}$$

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

where v is a sufficiently smooth function. The integration limit $d(x,y,\theta)$ is called the backward distance from the internal node (x,y) to the boundary $\partial\Omega$, along the direction $(-\cos\theta,-\sin\theta)$. For instance, for a unit square $[0,1]^2$, $d(x,y,0)=x$, $d(x,y,\pi/2)=y$, $d(x,y,\pi)=1-x$, and $d(x,y,3\pi/2)=1-y$. It can be proven via integration by parts that $I_\theta^\sigma u(x,y)\to u(x,y)$ as 0, and therefore as α equals 2, D_θ^α recovers to D_θ^2. The space-fractional derivative D_M^α can be related to the fractional Laplacian by the following equation [23]

$$D_M^\alpha=\frac{\sqrt{\pi}}{\Gamma\left(\frac{1-\alpha}{2}\right)\Gamma\left(\frac{2+\alpha}{2}\right)}(-\Delta)^{\alpha/2},\quad \alpha\in(1,2], \tag{2.6}$$

if D_M^α and $(-\Delta)^{\alpha/2}$ are both defined on $\mathbb{R}^2$ and $m(\theta)$ is a constant, namely $1/2\pi$. It is trivial that $D_M^2=0.5\Delta$ on $\mathbb{R}^2$. Here $\Gamma(\cdot)$ is the gamma function.

It is worthy of noting that in practical computation we consider the counterpart of D_θ^α, i.e., the Caputo-form operator

$${}^*D_\theta^\alpha u(x,y)=I_\theta^{2-\alpha}D_\theta^2 u(x,y), \tag{2.7}$$

for convenience. One can still obtain the numerical results for D_θ^α by using the relationship between D_θ^α and ${}^*D_\theta^\alpha$, which is an analogue to the relationship between Riemann-Liouville- and Caputo-fractional derivatives. In fact, from the homogeneous Dirichlet boundary, these two types of fractional derivatives are equivalent [24].

3 Finite element method

Due to the direction dependence of the non-coordinate-directed space-fractional derivative, the calculation of the backward distance plays a crucial role in the computation of stiffness matrix. The section first revisits the basic idea behind the finite element scheme, and then presents the backward distance algorithm that can calculate the backward distances efficiently.

3.1 Finite element scheme

A rectangular domain Ω modeled by N_e four-node rectangular elements and M active nodes is considered here. Using the finite element interpolation, the interpolated field function reads:

$$u(x,y,t)=\sum_{j=1}^{M}\phi_j(x,y)u_j(t)=\{\phi_1(x,y),\cdots,\phi_M(x,y)\}\begin{Bmatrix}u_1(t)\\ \vdots\\ u_M(t)\end{Bmatrix}=\mathbf{\Phi u}, \tag{3.1}$$

in which ϕ_j is non-zero only on the elements that contain node j. The interpolation vector $\mathbf{\Phi}=\{\phi_j\}$ and vector of nodal coefficients $\mathbf{u}=\{u_j\}$ are self-defined. The weak form of

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

Eq. (2.1) takes the form of [19]

$$\mathbf{M}\frac{d\mathbf{u}}{dt}+c\int_0^{2\pi}\mathbf{A}(\theta)m(\theta)d\theta\mathbf{u}-\mathbf{B}\mathbf{u}=\mathbf{F}, \tag{3.2}$$

where

$$\begin{cases}\mathbf{M}=\displaystyle\int_\Omega \mathbf{\Phi}^T\mathbf{\Phi}d\Omega,\\ \mathbf{B}=\displaystyle\int_\Omega \mathbf{\Phi}^T(\mathbf{b}\cdot\nabla\mathbf{\Phi})d\Omega,\\ \mathbf{F}=\displaystyle\int_\Omega f\mathbf{\Phi}^T d\Omega,\\ \mathbf{A}(\theta)=\displaystyle\int_\Omega ({}^*D_\theta^{\alpha-1}\mathbf{\Phi})^T(D_\theta\mathbf{\Phi})d\Omega.\end{cases} \tag{3.3}$$

Within the element domain Ω_e, u can be expressed as:

$$u^e(x,y,t)=\{N_1^e(x,y),N_2^e(x,y),N_3^e(x,y),N_4^e(x,y)\}\begin{Bmatrix}u_1^e(t)\\ \vdots\\ u_4^e(t)\end{Bmatrix}=\mathbf{N}^e\mathbf{u}^e,$$

in which u_1^e through u_4^e are the values of u at the four element nodes. The global equation arising from (3.2) is formed by following the pseudo-code:

for $e_1=1,2,\cdots,N_e$,

- compute the 4×4 element matrices

$$\mathbf{M}^e=\int_{\Omega^{e_1}}(\mathbf{N}^{e_1})^T\mathbf{N}^{e_1}d\Omega \quad \text{and} \quad \mathbf{B}^e=\int_{\Omega^{e_1}}(\mathbf{N}^{e_1})^T(\mathbf{b}\cdot\nabla\mathbf{N}^{e_1})d\Omega,$$

- assemble the matrices $\mathbf{M}^e$, $\mathbf{B}^e$ to form $\mathbf{M}$, $\mathbf{B}$

  for $e_2=1,2,\cdots,N_e$,

  - compute the 4×4 element matrices

$$\mathbf{A}^e(\theta)=\int_{\Omega^{e_2}}({}^*D_\theta^{\alpha-1}\mathbf{N}^{e_1})^T(D_\theta\mathbf{N}^{e_2})d\Omega$$

    and the element stiffness matrix $\int_0^{2\pi}\mathbf{A}^e(\theta)m(\theta)d\theta$,
  - assemble the local matrix $\int_0^{2\pi}\mathbf{A}^e(\theta)m(\theta)d\theta$ to form the global stiffness matrix $\int_0^{2\pi}\mathbf{A}(\theta)m(\theta)d\theta$.

  end

end

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

In particular, **M**, **B** and $\mathbf{A}^e(\theta)$ are evaluated by Gaussian quadrature while the integration of $\mathbf{A}(\theta)m(\theta)$ with respect to θ is evaluated by the trapezoidal rule as suggested in [19]

$$\int_0^{2\pi}\mathbf{A}(\theta)m(\theta)d\theta\approx\frac{2\pi}{M_t}\sum_{k=1}^{M_t}\mathbf{A}(\theta_k)m(\theta_k),\quad \theta_k=\frac{2\pi(k-1)}{M_t},\tag{3.4}$$

where M_t is the number of the quadrature points.

For the integer-order case, namely taking $\alpha=2$, the above "for" statement in the inner loop vanishes, since the local matrix $\mathbf{A}^e(\theta)$ vanishes when elements e_1 and e_2 do not coincide. This case only needs a single loop to assemble the global stiffness matrix. However, for the fractional-order case $\alpha\in(1,2)$, the $\mathbf{A}^e(\theta)$ does not always vanish even if the elements e_1 and e_2 are distant from each other. This is due to the nonlocal property of the fractional directional differentiation ${}^*D_\theta^{\alpha-1}$. As a result, double loops are required for fractional-order case. The nonlocality of ${}^*D_\theta^{\alpha-1}$ will be further explained in the next subsection.

Moreover, the implicit Crank-Nicolson scheme is employed for the temporal discretization of (3.2).

3.2 Backward distance algorithm

Consider 4×4 matrix $\mathbf{A}^e(\theta)=\int_{\Omega^{e_2}}({}^*D_\theta^{\alpha-1}\mathbf{N}^{e_1})^T(D_\theta\mathbf{N}^{e_2})d\Omega$, whose entry is computed by

$$\int_{\Omega^{e_2}}({}^*D_\theta^{\alpha-1}N_r^{e_1})(D_\theta N_s^{e_2})d\Omega=\sum_k w_k\,{}^*D_\theta^{\alpha-1}N_r^{e_1}(\mathbf{P}_k)D_\theta N_s^{e_2}(\mathbf{P}_k),\tag{3.5}$$

for $r,s=1,2,3,4$, where $\mathbf{P}_k=(x_k,y_k)$ is the Gauss quadrature point in Ω^{e_2} as shown in Fig. 1, and w_k the corresponding weight factor. For interpolant of the form N(x, y)=a+bx+cy+dxy, from definition (2.7), its fractional directional derivative ${}^*D_\theta^{\alpha-1}N_r^{e_1}(x,y)$ is evaluated as

$${}^*D_\theta^{\alpha-1}N_r^{e_1}(x,y)=I_\theta^{2-\alpha}D_\theta N_r^{e_1}(x,y)=\tilde{a}I_\theta^{2-\alpha}\{1\}+\tilde{b}I_\theta^{2-\alpha}\{x\}+\tilde{c}I_\theta^{2-\alpha}\{y\},\tag{3.6}$$

where $\tilde{a}$, $\tilde{b}$ and $\tilde{c}$ are the coefficients of the 1, x and y terms in $D_\theta N_r^{e_1}(x,y)$. It follows from the definition of the fractional directional integral (2.5) that the explicit expression of each integral in (3.6) is obtainable. For instance, it holds that

$$I_\theta^{2-\alpha}\{x_k\}=\frac{1}{\Gamma(2-\alpha)}\int_{d_1}^{d_2}\zeta^{1-\alpha}(x_k-\zeta\cos\theta)d\zeta=\frac{(d_2^{2-\alpha}-d_1^{2-\alpha})x_k}{\Gamma(3-\alpha)}-\frac{(d_2^{3-\alpha}-d_1^{3-\alpha})\cos\theta}{\Gamma(2-\alpha)(3-\alpha)},\tag{3.7}$$

in which d_1 and d_2 are the backward distances of Gauss point $\mathbf{P}_k$ to the boundary of element Ω^{e_1} in the direction $-\boldsymbol{\theta}$, as shown in Fig. 1. It is seen from Eqs. (3.6) and (3.7) provided that the backward distances d_1 and d_2 exist as Fig. 1 shows, the fractional directional derivative of the shape function $N^{e_1}(x,y)$ at Gauss point $\mathbf{P}_k$ does not vanishes even

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

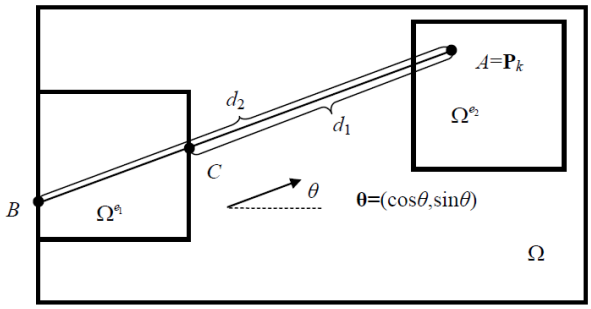


Figure 1: Backward distances d_1 and d_2 form the point A to the element Ω^{e_1}.

though the $\mathbf{P}_k$ is outside the element e_1. Note that this phenomenon will never occur for integer-order derivative which is locally defined. It is seen that the nonlocality of space-fractional derivative manifests itself through the definition of the fractional directional integral, i.e., the definition (2.5).

To compute the backward distances such as d_1 and d_2 in Fig. 1, Roop [19] presented an algorithm based on element transform and triple product of vectors that join the Gauss quadrature point and the triangular element vertices. This algorithm will become cumbersome when applied to rectangular elements since the reference plane needs to be extremely partitioned. In this study, based on the relative locations of point and line, we propose an alternative algorithm, backward distance algorithm, intended for generic polygon elements.

The backward distance algorithm consists of two steps:

1. Obtain the information of the intersection of the direction ray AB in Fig. 1 and each boundary segment of the element e_1,
2. Analyze the intersection information of the whole boundary, and determine the backward distances of the target point, namely the Gauss quadrature point $\mathbf{P}$ and the locations of the backward intersection points.

Step 1 Obtain the backward intersection point arising from $\mathbf{P}$ along $-\boldsymbol{\theta}$, which is point B or C in Fig. 1.

Consider the different relative locations of the target point $\mathbf{P}$ (x_P,y_P) and the straight line V_1V_2 on which the boundary segment V_1V_2 lies. Points V_1 (x_{v_1},y_{v_1}) and V_2 (x_{v_2},y_{v_2}) are the endpoints of the boundary segment. The differentiation direction is $\boldsymbol{\theta}=(\cos\theta,\sin\theta)$. All the possible relative locations are shown in Figs. 2 and 3. The backward intersection point can be the point C in Fig. 2, the point $\mathbf{P}$, or numerous points on line segment PV_1, PV_2, or V_1V_2 in Fig. 3. Unless otherwise stated, we always use point C to represent the backward intersection point.

To illustrate the algorithm procedure of Step 1, we use the following notations:

- Point Q is (not) on the line EF: $Q\in EF$ ($Q\notin EF$);

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

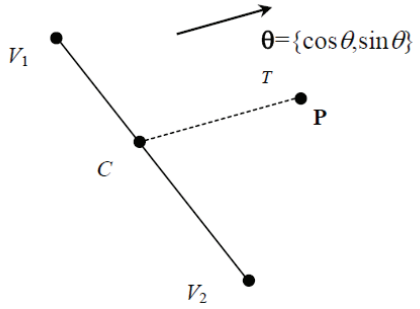


Figure 2: Backward intersection of the direction ray PC and the boundary segment V_1V_2 (**P** is not on the line V_1V_2, C is called the backward intersection point).

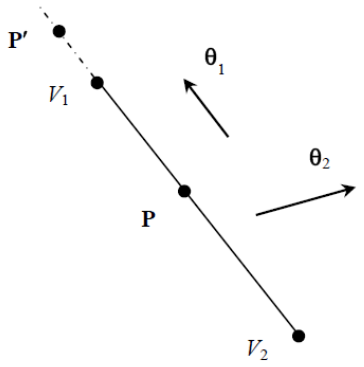


Figure 3: Backward intersection of the point **P** and boundary segment V_1V_2 (**P** can either be on the line segment V_1V_2 or be on the line V_1V_2 but outside the segment V_1V_2 (see **P**′); direction $\boldsymbol{\theta}$ is parallel or not parallel to the line V_1V_2).

- Direction $\boldsymbol{\theta}$ is (not) parallel to line EF: $\boldsymbol{\theta} \parallel EF$ ($\boldsymbol{\theta} \nparallel EF$);
- Point Q is (not) on the line segment EF: $Q \in \overline{EF}$ ($Q \notin \overline{EF}$).

The output variables from Step 1 are the number of backward intersection points r, the backward distance t, and the backward intersection point C (x_C, y_C). The graphic illustration of the Step 1 procedure is shown in Fig. 4 below. The specific pseudo-code is given in the Appendix.

Step 2 Obtain the backward distances d_1, d_2 of **P** to the boundary of element e_1.

- Input the locations of element vertexes, target point **P** and the direction $\boldsymbol{\theta}$.
- Output the backward distances d_1, d_2 and the location of backward intersection point p_C.
- Pseudo-code of the procedure is given in Appendix.

It is straightforward to extend the present algorithm to other polygon elements, such as the triangle and the parallelogram, because the algorithm only depends on the rela-

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

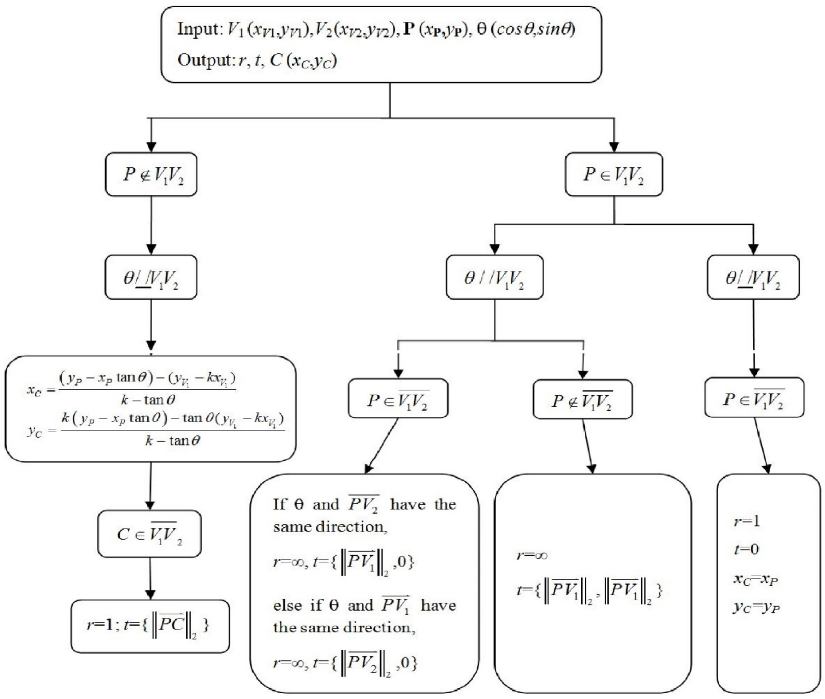


Figure 4: Graphic illustration of the Step 1 of the backward distance algorithm.

tive locations of the target point and the boundary segment irrespective of the boundary shape.

4 Finite difference method

The standard vector Grünwald formula defined on a grid with size $l^{\alpha+1}$ takes the form of [21]:

$$ {}^{*}D_{\theta}^{\alpha}v(x,y) = l^{-\alpha}\sum_{k=0}^{\infty}(-1)^k\begin{pmatrix}\alpha\\k\end{pmatrix}v\big(\chi\{x-kl\cos\theta, y-kl\sin\theta\}\big)+\mathcal{O}(l), \quad (x,y)\in\Omega, \quad (4.1)$$

where l is the sampling step along all differentiation directions, and $\chi_{x,y}$ denotes the nearest grid point to the point (x,y). Formula (4.1) works well in theory for an infinite domain Ω, but would fail for a finite domain. This is because, for a finite domain, the sampling step l is generally much larger than the grid size (e.g., $l > l^{1+\alpha} = 0.03125$ when $l = 0.25$, $\alpha = 1.5$). Thus fewer computing points are taken along certain differentiation directions, leading to lower accurate results.

To involve more grid points in approximation, we replace the original sampling points $\mathbf{x}_k = \{x-kl\cos\theta, y-kl\sin\theta\}$ by the intersection points of the direction ray and the

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

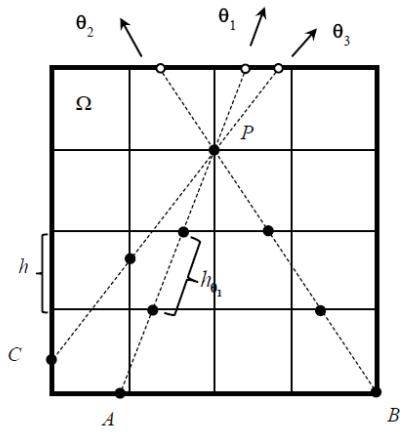


Figure 5: Sampling point of the FDM on 4×4 uniform grid.

grid lines. The grid lines are those parallel to the boundary segment on which the backward intersection point lays, i.e., the solid points in Fig. 5. The direction ray here can be referred to as PA, PB or PC. It should be noted that when the backward intersection point of P coincides with the vertex of Ω, e.g., direction ray PB and backward intersection point B, we select as many sampling points as possible.

To keep the formula numerically stable [25], we further adopt the right-shifted scheme

$$ {}^*D_\theta^\alpha v(P) \approx \sum_{k=0}^{m+1} c_k(\theta,\alpha) v\big(\chi\{x_P-(k-1)h_\theta\cos\theta, y_P-(k-1)h_\theta\sin\theta\}\big), \tag{4.2} $$

where the horizontal or vertical distance of P to $\partial\Omega$ is mh, and

$$ c_k(\theta,\alpha) = h_\theta^{-\alpha}(-1)^k \binom{\alpha}{k} = \frac{\Gamma(k-\alpha)}{h_\theta^\alpha \Gamma(-\alpha)\Gamma(k+1)}. \tag{4.3} $$

The variable spatial step h_θ equals $1/m$ times the backward distance of P to the boundary, and the backward distance can be computed using the forgoing backward distance algorithm or the explicit formulas like the Eq. (34) in [19]. Note that the hollow points in Fig. 5 are the shifted sampling points. Numerical results indicate that, the accuracy of (4.2) can reach $\mathcal{O}(h_\theta)$ when all the sampling points coincide with the grid points. Otherwise, the accuracy of the formula (4.2) would be slightly reduced due to the approximation in $\chi(\cdot)$. Denote the difference matrix of ${}^*D_\theta^\alpha$ by $\tilde{\mathbf{A}}(\theta)$ and the discretization matrix of ${}^*D_M^\alpha$ is accordingly $\int_0^{2\pi}\tilde{\mathbf{A}}(\theta)m(\theta)d\theta$. The discretization of the previous integral is the same as (3.4). Also, the implicit Crank-Nicolson and the central difference schemes are applied to the temporary discretization and the discretization of the first spatial derivative in the FADE (2.1), respectively.

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

5 Numerical results and discussions

Three tested examples are given in this section to show the validity and utility of the present methods. The first two problems are steady-state diffusion problems with known exact solutions and the last problem is transient advection-dispersion problems having no analytical solution which simulates the non-Fickian solution transport in groundwater.

Example 5.1 (Steady-state diffusion problem in coordinate directions)**.** For

$$\begin{cases} -(m_1{}^*D_0^{1.5}+m_2{}^*D_{\pi/2}^{1.5}+m_3{}^*D_{\pi}^{1.5}+m_4{}^*D_{3\pi/2}^{1.5})u=f, \quad (x,y)\in\Omega=[0,1]^2, \\ u\big|_{\partial\Omega}=0, \end{cases}$$

Case A: $\mathbf{m}=\{m_j\}=\left\{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right\}$,

Case B: $\mathbf{m}=\left\{\frac{1}{8},\frac{1}{8},\frac{1}{12},\frac{2}{3}\right\}$.

Rectangular domains of different sizes can be normalized to unit square Ω here. The exact solution is assumed to be $u(x,y)=x^2(1-x)^2y^2(1-y)^2$. The source term f can be derived with the aids of symbolic computational software. In Case A, four types of elements are tested. Ω is partitioned into $(L+1)\times(L+1)$ grid with size $h=1/L$. If necessary, each square grid is further partitioned into two triangles. Three-node triangular, six-node triangular, four-node rectangular and eight-node rectangular elements are tested. Define the normalized error as

$$Err=\frac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}=\frac{\|u(x,y)-\sum_j u_j^h\phi_j(x,y)\|_{L^2(\Omega)}}{\|u(x,y)\|_{L^2(\Omega)}}$$
$$=\frac{\sqrt{\sum_e\int_{\Omega^e}\left(u(x,y)-\sum_j u_j^h\phi_j(x,y)\right)^2d\Omega}}{\sqrt{\sum_e\int_{\Omega^e}u(x,y)^2d\Omega}}.$$

Table 1 shows the normalized errors of four types of elements in Case A. The values in the parenthesis are the convergence rate defined by

$$r=\frac{\log\left(\|u-u^{h_1}\|_{L^2(\Omega)}/\|u-u^{h_2}\|_{L^2(\Omega)}\right)}{\log(h_1/h_2)}.$$

It is seen from Table 1 that the numerical solutions of FEM possess the convergence rate of at least two-order. In terms of computation cost, the four-node rectangle is most economic since it enjoys the least sum of element and node numbers for a given gird size h. It is stressed that using the backward distance algorithm we succeed in realizing the finite element analysis based on the rectangular elements. However, this would not be easily done by using the finite element scheme presented in [19].

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

Table 1: Normalized errors of four types of elements in Case A of Example 5.1.

Err	$h=1/4$	$h=1/8$	$h=1/16$	$h=1/32$
3−node triangle	$1.461e-1$	$3.460e-2(2.08)$	$7.481e-3(2.21)$	$1.631e-3(2.20)$
6−node triangle	$3.141e-2$	$4.849e-3(2.70)$	$7.093e-4(2.77)$	$1.068e-4(2.73)$
4−node rectangle	$1.152e-1$	$2.799e-2(2.04)$	$6.325e-3(2.15)$	$1.402e-3(2.17)$
8−node rectangle	$1.886e-2$	$2.407e-3(2.97)$	$3.101e-4(2.96)$	$4.061e-5(2.93)$

Table 2: Normalized errors of FDM in Case A of Example 5.1.

	$h=1/8$	$h=1/16$	$h=1/32$	$h=1/64$
Err	$7.823e-2$	$5.441e-2(0.52)$	$3.093e-2(0.81)$	$1.633e-2(0.92)$

The normalized errors of solutions of FDM are given in Table 2. For comparison convenience, we approximate the error on a rectangular element by the mean value of the errors on four vertices of the element, which reads

$$Err=\frac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}=\frac{\|u(x,y)-u^h(x,y)\|_{L^2(\Omega)}}{\|u(x,y)\|_{L^2(\Omega)}}=\frac{\sqrt{\sum_e\left(\frac{1}{4}\sum_{i_e=1}^4|u_{i_e}-u_{i_e}^h|\right)^2\Delta\Omega^e}}{\sqrt{\sum_e\left(\frac{1}{4}\sum_{i_e=1}^4|u_{i_e}|\right)^2\Delta\Omega^e}}.$$

Table 2 shows that the FDM is lower in accuracy and slower in convergence than the FEM, which supports the fact that the former has at most one-order accuracy [21] whereas the latter has at least two-order accuracy [19].

Fig. 6 plots the normalized error of the Case B for the FEM and the FDM. The similar comparison results have been observed as the Case A, which indicates the direction weights $\{mi\}$ before the fractional directional derivatives have no effect on the performance of the FEM and the FDM.

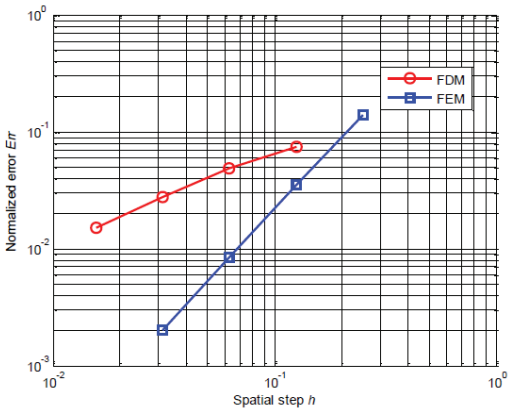


Figure 6: Normalized errors of the FEM and the FDM for Case B of Example 5.1.

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

Example 5.2 (Steady-state diffusion problem in non-coordinate directions)**.** For

$$\begin{cases} -\dfrac{1}{2\pi}\displaystyle\int_0^{2\pi} {}^*D_\theta^{1.5}u(x,y)d\theta = f(x,y), \quad (x,y)\in\Omega=[0,1]^2, \\ u\big|_{\partial\Omega}=0. \end{cases}$$

The exact solution still takes $u(x,y)=x^2(1-x)^2y^2(1-y)^2$. The source term f is expressed by the double integration and has no closed-form expression generally. In this case, integration quadrature is employed. As discussed in [23], the source term f is computed by using 10-point Gauss-Jacobi-Lobatto rule and 10-point piecewise Gauss-Legendre rule for the inner and outer integrations, respectively.

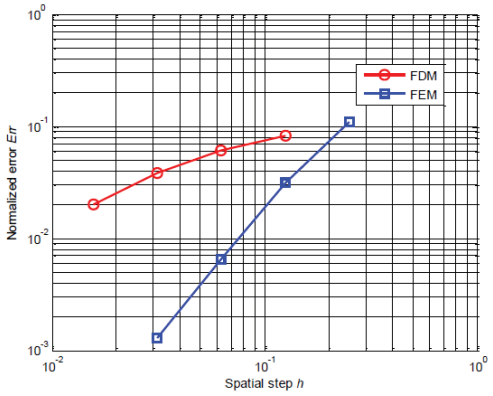


Figure 7: Normalized errors of FEM, FDM for Example 5.2.

We consider a regular $(L+1)\times(L+1)$ grid with spatial step $h=1/L$. The number of operations in computation of stiffness matrix of the FEM is expressed by $16L^2n_1M_t$, where 16 is the number of entries of the element stiffness matrix, n_1 is number of the Gauss-Legendre quadrature point assigned in each rectangular element (here we take $n_1=4$), and M_t is the number of quadrature points in trapezoidal rule (3.4). For the FDM, the operation number is smaller and takes the form of $(L-1)^2M_t$, where $(L-1)^2$ is the number of the internal grid points. To save the computing time, we use a small number of trapezoidal quadrature points, namely $M_t=8$, to evaluate the stiffness matrix $\int_0^{2\pi}\mathbf{A}(\theta)d\theta$ of the FEM and the difference matrix $\int_0^{2\pi}\tilde{\mathbf{A}}(\theta)d\theta$ of the FDM. Fig. 7 displays the normalized errors of the two methods. It is observed that the FEM compares favorably with the FDM in terms of solution accuracy and the convergence rate. Due to higher computational cost of the FEM, we only compute the FEM solution on a coarser grid, namely, with a larger spatial step h. The lower accuracy of the FDM is mainly led to by the approximation formula of the fractional directional derivative up to first-order accuracy.

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

Example 5.3 (Transient advection-dispersion problem in non-coordinate directions)**.** For

$$\begin{cases} \dfrac{\partial u}{\partial t}-\dfrac{0.3}{2\pi}\displaystyle\int_0^{2\pi} D_\theta^{1.1}u d\theta+0.12\dfrac{\partial u}{\partial x}=0, \quad \mathbf{x}\in\Omega=[-20,300]\times[-40,40], \quad t\in[0,328], \\ u\big|_{\partial\Omega}=0, \end{cases}$$

with initial-value function

$$u(x,y,0)=\begin{cases} 55610\times 2^{1-(1-(x-0.5)^2/64-(y-0.5)^2/64)^{-1}}, & 1-(x-0.5)^2/64-(y-0.5)^2/64>0, \\ 0, & \text{otherwise.} \end{cases}$$

This example simulates the dispersion and advection of a plume of contaminant particles in groundwater, and is analogue to the computational experiment discussed in [19]. In that reference, the FEM solution was obtained by using triangular elements. In this study, however, we use rectangular elements based on the backward distance algorithm. The temporal step takes $\Delta t=0.1$. The FEM solutions on 81×21 grid for normal (integer-order derivative, $\alpha=2$) and anomalous (fractional derivative, $\alpha=1.1$) dispersions are first given and compared in Fig. 8. The contours of $u=4,4^2,4^3,4^4$ and 4^5 are plotted to clearly show the concentration field.

It is seen from Fig. 8(b) that unlike the previous example, a smaller M_t no longer ensures accurate solutions. A larger M_t is needed to make the computed contours smoother. Comparison of Fig. 8(a) and Fig. 8(c) shows the super-diffusive features the fractional derivative model can produce, which agrees with the results of some hydrological experiments [2,13,14]. It should be noted that the above FEM solutions agree well with those given in [19] using triangular elements.

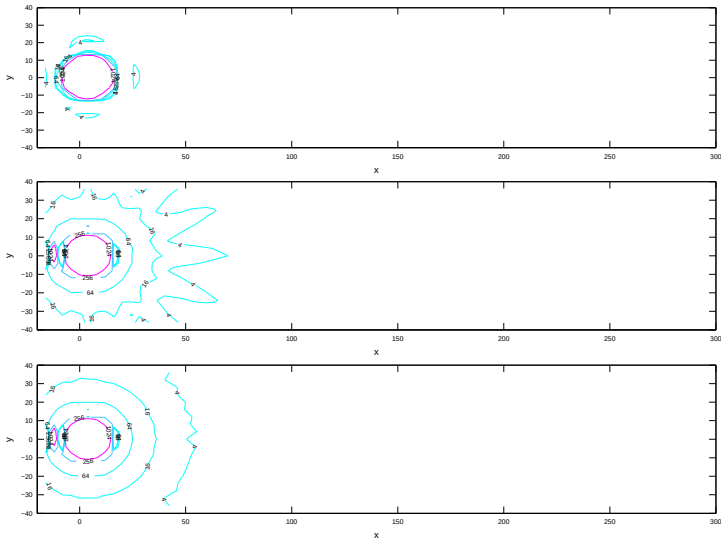


Figure 8: The solutions of FEM on 81×21 grid in Example 5.3. (a) $t=27$, $\alpha=2$, (b) $t=27$, $\alpha=1.1$, $M_t=16$, (c) $t=27$, $\alpha=1.1$, $M_t=32$.

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

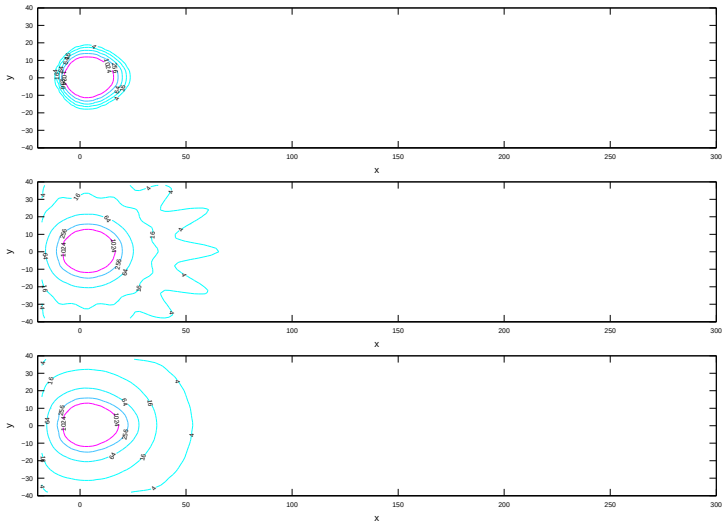


Figure 9: The solutions of the FDM (161×41 grid) in Example 5.3. (a) $t=27$, $\alpha=2$, (b) $t=27$, $\alpha=1.1$, $M_t=16$, (c) $t=27$, $\alpha=1.1$, $M_t=32$.

The solutions of the FDM are given in Fig. 9. A refined grid is employed in order to derive more accurate concentration field. As discussed in the preceding example, the computational cost of the FDM is lower than that of the FEM. The former only takes around fifty minutes to compute the difference matrix of the ${}^*D_M^\alpha$ for Fig. 9(c) whereas the latter costs roughly 15 hours to form the global stiffness matrix for Fig. 9(c). The FDM is thus more appealing for large-scale simulation. Despite the distinction of the computational effort, the FEM and the FDM can both be utilized to simulate the non-Fickian solute transport. Comparing Fig. 9(c) with the Fig. 8(c), one sees the two methods can produce the similar concentration fields. Note that the contour of the FDM is smoother than that of the FEM since the former uses a refined grid.

6 Concluding remarks

The study shows that the FEM compares favorably with the FDM in terms of the solution accuracy and the convergence rate. Also, the FEM can be easily applied to irregular domain problems whereas the FDM cannot. The disadvantage of the FEM is its large computational cost due to the double elemental loops in assembling the global stiffness matrix. This problem would be resolved by using the high-order elements or the differential quadrature elements [26], which reduces the number of the elements in the double loops. The lower accuracy of the FDM could be improved if using high-order Grünwald formula as given in [24]. The potential advantage of the FDM over the FEM is its flexibility for the FADE with different types of boundary conditions such as the Neumann and even the mixed boundary conditions. The present FEM was only proposed and intended

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

for the FADE with Dirichlet boundary condition. Up to now, little has been reported on the FEM for the FADE with Neumann or other complex boundary conditions. This is an open issue to be investigated.

On the other hand, the FEM and the FDM are essentially the local mesh-based approximation methods, depending on the finite element and the local difference approximation, respectively. The nonlocal non-coordinate derivative, namely the continuous form of the ${}^*D^{\alpha}_{M}$, in the FADE is not naturally discretized in these traditional mesh-based methods, owing to the direction ray arising from the definition of the fractional directional integral. The direction ray can be referred to as the ray PA in Fig. 1. In the FEM, even though two elements are distant from each other, the element stiffness matrix does not vanish provided that the direction ray sending from the quadrature point in one element intersects with another element. This leads to double elemental traversals in assembling the global stiffness matrix which makes the FEM time-consuming. In the FDM, the direction ray does not always pass through the grid points, which enforces one to use a nearest-grid-point strategy. This strategy reduces the accuracy of the FDM in some cases, particularly for the case of large M_t in trapezoidal rule (3.4). In fact, some global meshless methods such as the radial basis function methods [27,28] may be more suitable for discretizing the non-local non-coordinate derivative because of being globally defined (free of elemental traversal in the FEM), meshfree, and differentiation-direction independent (free of nearest-grid-point strategy in the FDM). The work along this line is under intensive investigation and will be reported in subsequent reports.

Appendix A: Pseudo-code of the Step 1

$r=0$, $t=0$, $x_C=[\ ]$, $y_C=[\ ]$. $[\ ]$ means a null value.

if **P** is not on the line V_1V_2 as shown in Fig. 2

if $\overrightarrow{V_1V_2}$ is not parallel to $\boldsymbol{\theta}$

Now get the intersection point of the direction ray PC and line V_1V_2

$$x_C=\frac{(y_P-x_P\tan\theta)-(y_{V_1}-kx_{V_1})}{k-\tan\theta},\quad y_C=\frac{k(y_P-x_P\tan\theta)-\tan\theta(y_{V_1}-kx_{V_1})}{k-\tan\theta},$$

k is the slope of line V_1V_2. We consider three cases where the line V_1V_2 is horizontal, vertical and oblique. The above equations will be modified for horizontal and vertical cases.

The following judgment sees whether the computed C belongs to the boundary segment V_1V_2

if $\overrightarrow{PC}\cdot\boldsymbol{\theta}<0$ and $\overrightarrow{V_1C}\cdot\overrightarrow{V_2C}\leq 0$

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

If so, find one backward intersection point, and the backward distance is the l_2-norm of vector PC

$$r=1, \quad t=\{\|\overrightarrow{PC}\|_2\}.$$

end

end

else if **P** is on the line V_1V_2 as shown in Fig. 3

P can be on the line V_1V_2 but outside the segment V_1V_2

if $\overrightarrow{V_1V_2}$ is parallel to $\boldsymbol{\theta}$

In this case, if point C exist, there must be numerous point C.

If $\boldsymbol{\theta}\cdot\overrightarrow{PV_1}<0$ and $\boldsymbol{\theta}\cdot\overrightarrow{PV_2}\geq 0$

P is inside the boundary segment V_1V_2, and the farthest backward intersection point is V_1

$$r=\infty, \quad t=\{\|\overrightarrow{PV_1}\|_2,0\}.$$

else if $\boldsymbol{\theta}\cdot\overrightarrow{PA}\geq 0$ and $\boldsymbol{\theta}\cdot\overrightarrow{PB}<0$

P is inside the boundary segment V_1V_2, and the farthest backward intersection point is V_2

$$r=\infty, \quad t=\{\|\overrightarrow{PV_2}\|,0\}.$$

else if $\boldsymbol{\theta}\cdot\overrightarrow{PV_1}<0$ and $\boldsymbol{\theta}\cdot\overrightarrow{PV_2}<0$

P is outside the boundary segment V_1V_2, and the nearest and the farthest backward intersection points are V_1 (or V_2) and V_2 (or V_1), respectively,

$$r=\infty, \quad t=\{\|\overrightarrow{PV_1}\|_2,\|\overrightarrow{PV_2}\|_2\}.$$

end

$\overrightarrow{V_1V_2}$ is not parallel to $\boldsymbol{\theta}$

If $\overrightarrow{PV_1}\cdot\overrightarrow{PV_2}\leq 0$

P is inside the boundary segment V_1V_2.

$$r=1, \quad t=0.$$

end

end

end

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

Appendix B: Pseudo-code of the Step 2

$$d_1=0,\quad d_2=0,\quad pC=[\ ].$$

Invoke the procedure in Step 1 to obtain the intersection information on four boundary segments of element e_1, namely r_i, t_i, $[x_{C_i},y_{C_i}]$, $i=1,2,3,4$. For a specific intersection as shown in Fig. 10, we have from Step 1 that

$$\begin{aligned}
&r_1=r_{12}=1, && t_1=t_{12}=\|\overrightarrow{PA}\|, && (x_{C_1},y_{C_1})=(x_A,y_A),\\
&r_2=r_{23}=1, && t_2=t_{23}=\|\overrightarrow{PB}\|, && (x_{C_2},y_{C_2})=(x_B,y_B),\\
&r_3=r_{34}=0, && t_3=t_{34}=0, && (x_{C_3},y_{C_3})=[\ ],\\
&r_4=r_{41}=1, && t_4=t_{41}=\|\overrightarrow{PA}\|, && (x_{C_4},y_{C_4})=(x_A,y_A).
\end{aligned}$$

Let the vector $\mathbf{r}=r_1,r_2,r_3,r_4$, $\mathbf{t}=t_1,t_2,t_3,t_4$, $\mathbf{x}=\{x_{C_1},x_{C_2},x_{C_3},x_{C_4}\}$, $\mathbf{y}=\{y_{C_1},y_{C_2},y_{C_3},y_{C_4}\}$ and denote by the scalar a the subscript of the component of $\mathbf{r}$ which equals ∞ and by vector $\mathbf{b}$ the subscripts of the components of $\mathbf{r}$, which equal one.

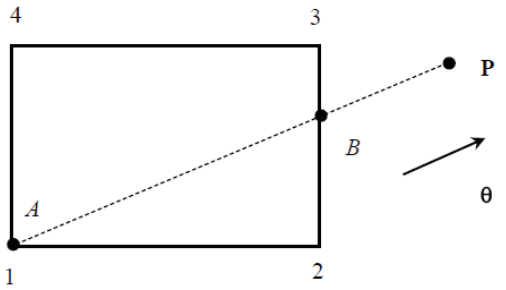


Figure 10: Backward intersection of the Gauss quadrature point **P** and the rectangular element e_1.

if a exists

In this case, point **P** is on the line where one of the boundary segments lays, and the direction θ is parallel to the line.

Numerous backward intersection points now exist, so keep pC null.

$$d_2=\max(\mathbf{t}(a:a+1))$$

$\mathbf{t}(a:a+1)$ takes the a-th and the $(a+1)$-th components of vector $\mathbf{t}$, and the function $\max(\cdot)$ takes the larger one of the previous components

$$\begin{aligned}
&d_1=\min(\mathbf{t}(a:a+1)),\\
&pC=[\ ].
\end{aligned}$$

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

else if the length of **b** is 1

In this case, the target point **P** is inside the element and there is only one backward intersection point pC.

$$d_2 = \mathbf{t}(\mathbf{b}),$$

$\mathbf{t}(\mathbf{b})$ takes the $\mathbf{b}(1)$-th, $\mathbf{b}(2)$-th, $\cdots$, $\mathbf{b}(l)$-th components of vector **t**, where l is the length of vector **b** (here $l=1$).

$$\begin{aligned} &d_1 = 0, \\ &pC = (\mathbf{x}(\mathbf{b}), \mathbf{y}(\mathbf{b})), \end{aligned}$$

else if the length of **b** is 2, and **P** is inside the element

In this case, the backward intersection point pC must be a vertex of the element.

$$d_2 = \mathbf{t}(\mathbf{b}(1)) \quad (\text{or } \ d_2 = \mathbf{t}(\mathbf{b}(2)))$$

Note that $\mathbf{t}(\mathbf{b}(1)) = \mathbf{t}(\mathbf{b}(l))$ because points $C_{\mathbf{b}(1)}$ and $C_{\mathbf{b}(2)}$ coincides

$$d_1 = 0, \quad pC = (\mathbf{x}(\mathbf{b}(1)), \mathbf{y}(\mathbf{b}(1)))$$

else if the length of **b** exceeds 1

In this case, **P** is outside the element (as given in Figs. 1 and 10) or on its boundary.

$$\begin{aligned} &[d_2, index_\max] = \max(\mathbf{t}(\mathbf{b})), \\ &[d_1, index_\min] = \min(\mathbf{t}(\mathbf{b})). \end{aligned}$$

"$index_\max$" denotes the subscript of the larger component of $\mathbf{t}(\mathbf{b})$.

Now there are two backward intersection points, namely the point A and B in Fig. 10

$$pC = \begin{bmatrix} \mathbf{x}(index_\max), \mathbf{y}(index_\max) \\ \mathbf{x}(index_\min), \mathbf{y}(index_\min) \end{bmatrix}.$$

end

Acknowledgements

The research is financially supported by the Hong Kong Research Grant Council (GRF grant HKU717309E), the National Science Funds for Distinguished Young Scholars of China (Grant No. 11125208), the National Science Funds of China (Grant No. 11372097), and the 111 Project (Grant No. B12032).

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

References

[1] H. G. SUN, W. CHEN, H. SHENG AND Y. Q. CHEN, *On mean square displacement behaviors of anomalous diffusions with variable and random orders,* Phys. Lett. A, 374 (2010), pp. 906–910.

[2] H. G. SUN, Y. ZHANG, W. CHEN AND D. M. REEVES, *Use of a variable-index fractional-derivative model to capture transient dispersion in heterogeneous media,* J. Contam. Hydrol., 157 (2014), pp. 47–58.

[3] L. GAUL, *The influence of damping on waves and vibrations,* Mech. Syst. Signal Proc., 13 (1999), pp. 1–30.

[4] W. CHEN AND S. HOLM, *Modified Szabo's wave equation models for lossy media obeying frequency power law,* J. Acoust. Soc. Am., 114 (2003), pp. 2570–2574.

[5] C. A. MONJE, Y. Q. CHEN, B. M. VINAGRE, D. XUE AND V. FELIU-BATLLE, Fractional-order Systems and Controls: Fundamentals and Applications, Springer, 2010.

[6] T. M. ATANACKOVIC, S. PILIPOVIC, B. STANKOVIC AND D. ZORICA, Fractional Calculus with Applications in Mechanics: Vibrations and Diffusion Processes, Wiley, 2014.

[7] T. M. ATANACKOVIC, S. PILIPOVIC, B. STANKOVIC AND D. ZORICA, Fractional Calculus with Applications in Mechanics: Wave Propagation, Impact, and Variational Principles, Wiley, 2014.

[8] W. CHEN AND S. HOLM, *Fractional Laplacian time-space models for linear and nonlinear lossy media exhibiting arbitrary frequency power-law dependency,* J. Acoust. Soc. Am., 115 (2004), pp. 1424–1430.

[9] W. CHEN, *An intuitive study of fractional derivative modeling and fractional quantum in soft matter,* J. Vib. Control, 14 (2008), pp. 1651–1657.

[10] F. W. LIU, V. ANH AND I. TURNER, *Numerical solution of the space fractional Fokker-Planck equation,* J. Comput. Appl. Math., 166 (2004), pp. 209–219.

[11] Q. Q. YANG, F. W. LIU AND I. TURNER, *Numerical methods for fractional partial differential equations with Riesz space fractional derivatives,* Appl. Math. Model., 34 (2010), pp. 200–218.

[12] M. M. MEERSCHAERT, H. P. SCHEFFLER AND C. TADJERAN, *Finite difference methods for two-dimensional fractional dispersion equation,* J. Comput. Phys., 211 (2006), pp. 249–261.

[13] D. A. BENSON, S. WHEATCRAFT AND M. M. MEERSCHAERT, *Application of a fractional advection-diffusion equation,* Water Resour. Res., 36 (2000), pp. 132–138.

[14] D. A. BENSON, R. SCHUMER, M. M. MEERSCHAERT AND S. W. WHEATCRAFT, *Fractional diffusion, Levy motions, and the made tracer tests,* Transp. Porous Media, 42 (2001), pp. 211–240.

[15] Y. Y. ZHENG, C. P. LI AND Z. G. ZHAO, *A note on the finite element method for the space-fractional advection diffusion equation,* Comput. Math. Appl., 59 (2010), pp. 1718–1726.

[16] Y. J. CHOI AND S. K. CHUNG, *Finite element solutions for the space fractional diffusion equation with a nonlinear source term,* Abstract Appl. Anal., 2012 (2012), pp. 596184.

[17] K. BURRAGE, N. HALE AND D. KAY, *An efficient implicit FEM scheme for fractional-in-space reaction-diffusion equations,* SIAM J. Sci. Comput., 34 (2012), pp. A2145.

[18] J. J. ZHAO, J. Y. XIAO AND Y. XU, *A finite element method for the multiterm time-space Riesz fractional advection-diffusion equations in finite domain,* Abstract Appl. Anal., 2013 (2013), pp. 868035.

[19] J. P. ROOP, *Computational aspects of FEM approximation of fractional advection diffusion equations on boundary domains in* $\mathbb{R}^2$, J. Comput. Appl. Math., 193 (2006), pp. 243–268.

[20] V. J. ERVIN AND J. P. ROOP, *Variational solution of fractional advection diffusion equation on bounded domains in* $\mathbb{R}^d$, Numer. Meth. Part Differ. Equ., 23 (2006), pp. 256–281.

[21] M. M. MEERSCHAERT AND J. MORTENSON, *Vector Grünwald formula for fractional derivatives,*

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press

Fract. Calc. Appl. Anal., 7 (2004), pp. 61–81.
[22] M. M. MEERSCHAERT, D. A. BENSON AND B. BAUMER, *Multidimensional advection and fractional diffusion,* Phys. Rev. E, 59 (1999), pp. 5026–5028.
[23] G. F. PANG, W. CHEN AND K. Y. SZE, *Gauss-Jacobi-type quadrature rules for fractional directional integrals,* Comput. Math. Appl., 66 (2013), pp. 597–607.
[24] I. PODLUBNY, Fractional differential equations: An introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications, Academic Press, 1999.
[25] C. TADJERAN, M. M. MEERSCHAERT AND H. P. SCHFFLER, *A second-order accurate numerical method for the fractional diffusion equation,* J. Comput. Phys., 213 (2006), pp. 205–213.
[26] X. W. WANG AND H. Z. GU, *Static analysis of frame structures by the differential quadrature element method,* Int. J. Numer. Methods Eng., 40 (1997), pp. 759–772
[27] E. J. KANSA, *Multiquadrics–A scattered data approximation scheme with applications to computational fluid-dynamics–II solutions to parabolic, hyperbolic and elliptic partial differential equations,* Comput. Math. Appl., 19 (1990), pp. 147–161.
[28] E. J. KANSA, *Multiquadrics–A scattered data approximation scheme with applications to computational fluid-dynamics–I surface approximations and partial derivative estimates,* Comput. Math. Appl., 19 (1990), pp. 127–145.

https://doi.org/10.4208/aamm.2014.m693 Published online by Cambridge University Press