Stable Lévy diffusion and related model fitting

Paramita Chakraborty[a,*], Xu Guo[b], Hong Wang[c]

[a]*Department of Statistics, University of South Carolina, Columbia, SC, USA*
[b]*Department of Mathematics, Hong Kong Baptist University, Hong Kong*
[c]*Department of Mathematics, University of South Carolina, Columbia, SC, USA*

chakrabp@stat.sc.edu (P. Chakraborty), guoxu1014@hkbu.edu.hk (X. Guo),
hwang@math.sc.edu (H. Wang)

Abstract   A fractional advection-dispersion equation (fADE) has been advocated for heavy-tailed flows where the usual Brownian diffusion models fail. A stochastic differential equation (SDE) driven by a stable Lévy process gives a forward equation that matches the space-fractional advection-dispersion equation and thus gives the stochastic framework of particle tracking for heavy-tailed flows. For constant advection and dispersion coefficient functions, the solution to such SDE itself is a stable process and can be derived easily by least square parameter fitting from the observed flow concentration data. However, in a more generalized scenario, a closed form for the solution to a stable SDE may not exist. We propose a numerical method for solving/generating a stable SDE in a general set-up. The method incorporates a discretized finite volume scheme with the characteristic line to solve the fADE or the forward equation for the Markov process that solves the stable SDE. Then we use a numerical scheme to generate the solution to the governing SDE using the fADE solution. Also, often the functional form of the advection or dispersion coefficients are not known for a given plume concentration data to start with. We use a Levenberg–Marquardt (L-M) regularization method to estimate advection and dispersion coefficient function from the observed data (we present the case for a linear advection) and proceed with the SDE solution construction described above.

Keywords   Stable Lévy Diffusion, fractional diffusion, fractional advection-dispersion, heavy-tailed particle tracking

---

*Corresponding author.

1 Introduction

The usual hydrological model for contamination/tracer transport through a porous media is given by a second order advection-dispersion equation (ADE) of the form $\frac{\partial c}{\partial t} = -\frac{\partial}{\partial x}[v(x)c] + \frac{\partial^2}{\partial x^2}[D(x)c]$, where $c(x, t)$ is the tracer concentration at time t, location x, v is the drift velocity and D is related to the diffusivity of the media [4, 15]. The probabilistic approach to describe this flow from a mesoscopic view is given by the hypothesis that the path of a randomly chosen tracer particle is a Markov process that solves a stochastic differential equation (SDE) driven by Brownian motion. The basis of this hypothesis is that the conditional probability density function of this Markov process solves a forward equation of the same form as the ADE [7, 6].

However, for some heavy-tailed flows, the second order diffusion model can be inadequate. For such cases a model called the fractional advection-dispersion equation or fADE of the following form has been proposed [27]:

$$\frac{\partial c}{\partial t} = -\frac{\partial}{\partial x}\big[v(x)c\big] + \frac{\partial^{\alpha-1}}{\partial x^{\alpha-1}}\bigg[D(x)\frac{\partial c}{\partial x}\bigg]. \tag{1.1}$$

For the current discussion, we will consider a one-dimensional concentration $c(x, t)$ where x denotes the distance from the origin of the plume, v is the drift velocity and D is a function that changes with the diffusion of the tracer. The fractional differentiation order $\alpha \in (1, 2)$ controls the tail of the flow. Here the fractional derivative of order α for any function f is defined as in [3] by:

$$\frac{d^\alpha f(x)}{dx^\alpha} = \frac{-1}{\Gamma(1-\alpha)} \int\limits_0^\infty \big[f(x-u) - f(x) + uf'(x)\big]\alpha y^{-1-\alpha}du.$$

The negative fractional derivative is given by $\frac{d^\alpha f(x)}{d(-x)^\alpha} = \frac{d^\alpha g(x)}{dx^\alpha}$, where $g(x) = f(-x)$.

If we assume that the diffusion coefficient D is location invariant, then the fADE can be associated with an SDE driven by an α-stable Lévy process X_t [1] of the form:

$$dY_t = a(Y_t)dt + b(Y_t)dX_t. \tag{1.2}$$

Following the Brownian diffusion argument, in the heavy-tailed diffusion model, we assume a random particle's position at time t is given by the process Y_t that solves SDE (1.2). It can be shown that Y_t is a Markov process [1]. Let us denote the transition probability density function of Y_t by $p_{y_0}(y, t)$, i.e. $P(Y_t \in A|Y_0 = y_0) = \int_A p_{y_0}(y, t)dy$. For the initial distribution $\mu(u)$ the pdf of Y_t is given by

$$p(y, t) = \int p_u(y, t)\mu(u)du, \tag{1.3}$$

e.g., for a ground water tracer concentration modeling it is reasonable to assume that all particles start at location y_0 at time $t = 0$ hence $\mu(u) = I(u = y_0)$. If $1 < \alpha < 2$,

---

[1] Following the parametrization in [22] $X_t \sim Stable(\alpha, \beta, \mu, \sigma)$ where α is the index of stability, β is the skewness parameter, $\mu = 0$ is the location parameter and $\sigma = 1$ is the scale parameter.

it can be shown [10] that $p(y, t)$ solves the forward equation:

$$\frac{\partial p(y, t)}{\partial t} = -\frac{\partial}{\partial y}\left[a(y)p(y, t)\right] + \frac{(1 + \beta)}{2}\left[-\cos\left(\frac{\pi\alpha}{2}\right)\right]^{-1} \cdot \frac{\partial^\alpha}{\partial y^\alpha}\left[b^\alpha(y)p(y, t)\right]$$
$$+ \frac{(1 - \beta)}{2}\left[-\cos\left(\frac{\pi\alpha}{2}\right)\right]^{-1} \cdot \frac{\partial^\alpha}{\partial(-y)^\alpha}\left[b^\alpha(y)p(y, t)\right]. \tag{1.4}$$

Typically the coefficient functions a and b depend on t via Y_t as in (1.2) hence the terms in the forward equation are expressed as $a(y)$ and $b(y)$, where y is supposed to be in the range space of Y_t. However, without loss of generality, the functions can also be considered as a mapping from $T \times \mathbb{R} \to \mathbb{R}$ as $a(y, t)$ and $b(y, t)$ and the associated forward equation will remain the same.

Thus for a location invariant D, the fADE in (1.1) can be written as:

$$\frac{\partial C}{\partial t} = -\frac{\partial}{\partial x}\left[v(x, t)C\right] + D(t)\frac{\partial^\alpha}{\partial x^\alpha}\left[\frac{\partial C}{\partial x}\right], \tag{1.5}$$

which essentially has the same form as the forward equation given in (1.4) with $\beta = 1$, $a(x, t) = v(x, t)$, $[-\cos(\frac{\pi\alpha}{2})]^{-1}b^\alpha(x, t) \equiv D(t)$. This shows that in a heavy-tailed plume that follows fADE (1.5), the position Y_t of a randomly chosen tracer particle at time t solves SDE (1.2). Choosing $\beta \neq 1$ will allow the particle transition in forward or backward direction.

In most practical tracking scenarios, the plume data consists of observed tracer concentration values $c(x, t)$ over a range of locations at certain time points. We write $p(y, t) = K_t c(y, t)$, where K_t is a suitable scale parameter that adjusts the total mass for $c(x, t)$ so that $\int_\mathbb{R} p(y, t)dy = 1$ for each t.

In case $v(\cdot)$ and $D(\cdot)$ are constants, (1.2) reduces to $Y_t = at + bX_t$ with $a = v$, $b = [-\cos(\frac{\pi\alpha}{2})D]^{1/\alpha}$ and consequently, Y_t itself is a stable process. The parameters of this stable process then can be estimated using the observed data (see [11] for details). However, when the coefficient functions are not constants, the solution to the SDE in (1.2) may not have a closed form.

In this paper, we present a numerical method to solve Y_t through the fitted fADE equation using the observed tracer concentration data. Also, often for a groundwater contamination modeling problem the functional form of the advection or dispersion coefficients are not known to start with. In that case, the proper advection and dispersion coefficient functions are needed to be estimated. This presents an inverse problem. We formulate this inverse problem as an optimization problem and develop a Levenberg–Marquardt (L-M) regularization method to obtain the proper advection and dispersion coefficient function from the observed data (the case for a linear drift function is presented here). Then we proceed to generate Y_t using these fitted coefficient functions. The detailed methodology is described in Section 2. Section 3 provides an illustrative example. Section 4 includes concluding remarks.

2 Methodology

We propose a method for generating a weak solution for the SDE given in (1.2). The core technique is the standard probability integral transformation. Suppose the

probability density function (pdf) of the solution process Y_t at a given time point t is given by $p(y, t)$ and the cumulative distribution function (CDF) is defined by $F_{Y_t}(x) = \int_{-\infty}^{x} p(y, t) dy$. Since $F(Y_t) \sim Uniform(0, 1)$ [21], a random observation of Y_t, say y, can be generated by $F_{Y_t}^{-1}(u) = y$ where u is a random observation from $Uniform(0, 1)$. For the present problem, closed form of $p(t, y)$ is not available, and we use point-wise numerical approximations to estimate $p(y, t)$ for any fixed t and fixed y. Therefore we can only use a numerical scheme to approximate F_{Y_t} and $F_{Y_t}^{-1}$.

2.1 Simulating a solution process for an SDE driven by a heavy-tailed stable process

For a heavy-tailed plume where the particle path can be modeled by the SDE in (1.2), the forward equation is of the form (1.5). To generate the solution to this SDE we can consider two scenarios: (i) we may want to solve an SDE with a given form of coefficient functions $v(\cdot)$ and $D(\cdot)$ and driven by a given α-stable process; (ii) a more practical problem when only the tracer concentrations at different time points over a range of locations are observed from a plume, while the coefficient functions need to be estimated along with the parameters of the driving process X_t in (1.2) from the observed concentration values.

In either case, first, we find $p(t, y)$ that solves the forward equation (1.4) with given or estimated coefficient functions and then generate Y_t using these $p(t, y)$'s.

Y_t **Simulation Steps:**

Step 1: Numerical Solution for $p(t, y)$: Assuming the probability density $p(y, t) = K_t c(y, t)$, we estimate K_t, α and β from the observed data that assumed to satisfy fADE (1.5). $p(y, t)$ can be solved numerically using a discretization scheme from the forward equation (1.4). In case the coefficient functions in (1.4) are not known, an inverse problem optimization technique can be used for estimating the parameters. See Section 2.2 for details of this step.

In either case the discretization method provides only point-wise approximation of $p(y, t)$ values for given t and y and not a closed functional form.

Step 2: Generating Y_t: To generate Y_t for a fixed t using an inverse CDF transformation, we present a straightforward scheme with a trapezoidal rule to approximate F_{Y_t} and $F_{Y_t}^{-1}$. Approximate Y_t's can be simulated using $p(y, t)$'s from step 1 as follows:

(a) Choose suitable grid $y_0 < y_1 < \cdots < y_n$ and solve for $p(y_i, t)$ as mentioned in step 1. The endpoints y_0 and y_n can be chosen so that $p(y, t)$ is small or negligible for $y < y_0$ or $y > y_n$. The fitted coefficient and parameters from the observed data in step 1 can be used for any y_i's.

(b) Approximate $F_{Y_t}(y_i)$ by $\hat{F}_{Y_t}(y_i)$ for $i = 1, 2, \ldots, n$, using the trapezoidal rule:

$$\hat{F}_{Y_t}(y_1) = (y_1 - y_0) \frac{p(y_1, t) + p(y_0, t)}{2},$$

$$\hat{F}_{Y_t}(y_2) = \hat{F}_{Y_t}(y_1) + (y_2 - y_1) \frac{p(y_2, t) + p(y_1, t)}{2},$$

$$\vdots$$

$$\hat{F}_{Y_t}(y_r) = \hat{F}_{Y_t}(y_{r-1}) + (y_r - y_{r-1})\frac{p(y_r, t) + p(y_{r-1}, t)}{2},$$

$$\vdots$$

$$\hat{F}_{Y_t}(y_n) = \hat{F}_{Y_t}(y_{n-1}) + (y_n - y_{n-1})\frac{p(y_n, t) + p(y_{n-1}, t)}{2}.$$

(c) For a randomly generated *Uniform*(0, 1) observation u, find its placement in the grid, i.e. y_r, y_{r+1} such that $\hat{F}_{Y_t}(y_r) \leq u < \hat{F}_{Y_t}(y_{r+1})$, for some $r \in \{0, 1, \ldots, n\}$. A way of generating associated random observation of Y_t or $F_{Y_t}^{-1}(u)$ (approximated) as $\hat{y}$ is described below:

$$\hat{y} = \begin{cases} y_r, & \text{if } u = y_r; \\ y_r + \dfrac{u - \hat{F}_{Y_t}(y_r)}{p(t, y_r)}, & \text{if } y_r < u < y_{r+1}. \end{cases}$$

In case $u < \hat{F}_{Y_t}(y_0)$ set $\hat{y} = y_0$ and if $u \geq \hat{F}_{Y_t}(y_n)$ set $y = y_n$.

(d) To generate N i.i.d approximated random observations $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N$ of Y_t, start with randomly generated independent *Uniform*(0, 1) observations $u_1, u_2, \ldots u_N$ and then repeat step (c). Note that the efficiency of this approximation will depend on the spacing of the chosen grid since we are actually generating sample from distribution $\hat{F}_{Y_t}$ here and $\hat{F}_{Y_t} \to F_{Y_t}$ as $\max_{0 \leq i \leq n} |y_{i+1} - y_i| \to 0$ following the convergence property of the standard trapezoidal rule.

2.2 A formulation for a numerical solution of the forward equation

In this subsection, we will present a finite volume scheme with the characteristic line for solving a forward equation of the form (1.4). To incorporate a discretized finite volume scheme with proper boundary condition, we re-parameterize the forward equation with fractional order $2 - \lambda$ with $0 < \lambda < 1$ (consistent with the construction described in [25]) as follows:

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial x}a(x)p - \frac{\partial}{\partial x}\left[b(t)\left(\gamma_{x_l}D_x^{-\lambda} + (1 - \gamma)_x D_{x_r}^{-\lambda}\right)Dp\right] = 0, \quad x_l < x < x_r,$$

$$p(x_l, t) = 0, \qquad p(x_r, t) = 0, \quad 0 < t < T.$$

$$\text{(2.1)}$$

Essentially, compared to (1.4) the fractional derivative order $\alpha = 2 - \lambda$, while $\gamma = \frac{1+\beta}{2}$ indicates the relative weight of forward versus backward transition. $_{x_l}D_x^{-\lambda}$ and $_x D_{x_r}^{-\lambda}$ represent the left and right fractional integral operators defined as

$$_{x_l}D_x^{-\lambda} f(x) := \frac{1}{\Gamma(\beta)} \int_{x_l}^x (x - s)^{\lambda-1} f(s)\mathrm{d}s,$$

$$_x D_{x_r}^{-\lambda} f(x) := \frac{1}{\Gamma(\beta)} \int_x^{x_r} (s - x)^{\lambda-1} f(s)\mathrm{d}s.$$

$$\text{(2.2)}$$

To comply with the boundary conditions, $x = x_l$ and $x = x_r$ are set to be the inflow and outflow boundaries respectively [25], while $a(x_l)$ and $a(x_r)$ are assumed to be non-negative.

2.2.1 The discretized finite volume scheme and the accumulation terms

Let us define a partition $\mathcal{P}_t$ on the time interval $[0, T]$, and a partition $\mathcal{P}_x$ on the space interval $[x_l, x_r]$

$$
\begin{aligned}
\mathcal{P}_t &: 0 = t_0 < t_1 < \cdots < t_N = T, \\
\mathcal{P}_x &: x_l = x_0 < x_1 < \cdots < x_I = x_r,
\end{aligned}
\tag{2.3}
$$

with $\Delta t_n = t_n - t_{n-1}$ for $n = 1, \ldots, N$ and $\Delta x_i = x_i - x_{i-1}$ for $i = 1, \ldots, I$. Let $\bar{\mathcal{P}}_x$ be the dual partition of $\mathcal{P}_x$ defined by $x_{i-1/2} = \frac{1}{2}(x_{i-1} + x_i)$ with $\Delta x_{i+1/2} = x_{i+1/2} - x_{i-1/2}$ for $i = 1, \ldots, I - 1$ as well as $\Delta x_{1/2} = x_{1/2} - x_0$ and $\Delta x_{I+1/2} = x_I - x_{I-1/2}$.

Let $y = r(\theta; \bar{x}, \bar{t})$ be a continuous and piecewise-smooth curve that passes through the point $\bar{x}$ at time $\bar{t}$, such that $r(\theta; x_{i-1/2}, t_n)$ and $r(\theta; x_{i+1/2}, t_n)$ do not meet each other during the time period $[t_{n-1}, t_n]$. We define a space-time control volume Ω_i^n by extending the cell $[x_{i-1/2}, x_{i+1/2}]$ along the curve $r(t; x, t_n)$ from $t = t_n$ to $t = t_{n-1}$

$$
\Omega_i^n = \big\{ (x, t) : r(t; x_{i-1/2}, t_n) < x < r(t; x_{i+1/2}, t_n), \quad t_{n-1} < t < t_n \big\}. \tag{2.4}
$$

Assuming the prism Ω_i^n does not intersect the boundary $x = x_l$ and $x = x_r$ of the domain during the time period $[t_{n-1}, t_n]$, let $x^* = r(t_{n-1}; x, t_n)$ be the foot at time t_{n-1} of the curve with head x at time t_n.

Integrating equation (2.1) over the control volume Ω_i^n we get

$$
\int_{\Omega_i^n} \frac{\partial p}{\partial t} dx dt + \int_{t_{n-1}}^{t_n} \left(a(x)p - b(t) \frac{\partial^{1-\lambda}}{\partial x^{1-\lambda}} p \right) \bigg|_{r(t; x_{i-1/2}, t_n)}^{r(t; x_{i+1/2}, t_n)} dt = 0. \tag{2.5}
$$

Without loss of generality, the accumulation term in (2.4) can be evaluated by assuming

$$
x_{i-1/2}^* < x_{i-1/2} < x_{i+1/2}^* < x_{i+1/2}, \tag{2.6}
$$

and accumulation term can be re-written as follows:

$$
\int_{\Omega_i^n} \frac{\partial p}{\partial t} dx dt = \int_{x_{i-1/2}^*}^{x_{i-1/2}} \left[\int_{t_{n-1}}^{t(x; x_{i-1/2}, t_n)} \frac{\partial p}{\partial t} dt \right] dx
$$
$$
+ \int_{x_{i-1/2}}^{x_{i+1/2}^*} \left[\int_{t_{n-1}}^{t_n} \frac{\partial p}{\partial t} dt \right] dx + \int_{x_{i+1/2}^*}^{x_{i+1/2}} \left[\int_{t(x; x_{i+1/2}, t_n)}^{t_n} \frac{\partial p}{\partial t} dt \right] dx. \tag{2.7}
$$

Here the notation $t(x; x_{i-1/2}, t_n)$ represents the time instant that the curve $r(t; , x_{i-1/2}, t_n) = x$. The notation $t(x; x_{i+1/2}, t_n)$ is defined similarly.

A simple calculation of the second term on the right-hand side of (2.7) yields

$$
\int_{x_{i-1/2}}^{x_{i+1/2}^*} \left[\int_{t_{n-1}}^{t_n} \frac{\partial p}{\partial t} dt \right] dx = \int_{x_{i-1/2}}^{x_{i+1/2}^*} p(x, t_n) dx - \int_{x_{i-1/2}}^{x_{i+1/2}^*} p(x, t_{n-1}) dx. \tag{2.8}
$$

The first and third terms on the right-hand side of (2.7) are integrated as follows

$$\int_{x_{i-1/2}^*}^{x_{i-1/2}} \left[\int_{t_{n-1}}^{t(x;x_{i-1/2},t_n)} \frac{\partial p}{\partial t} \, dt \right] dx$$

$$= \int_{t_{n-1}}^{t_n} p\big(r(t; x_{i-1/2}, t_n), t\big) \frac{\partial r(t; x_{i-1/2}, t_n)}{\partial t} \, dt - \int_{x_{i-1/2}^*}^{x_{i-1/2}} p(x, t_{n-1}) \, dx, \quad (2.9)$$

and

$$\int_{x_{i+1/2}^*}^{x_{i+1/2}} \left[\int_{t(x;x_{i+1/2},t_n)}^{t_n} \frac{\partial p}{\partial t} \, dx \right] dt$$

$$= \int_{x_{i+1/2}^*}^{x_{i+1/2}} p(x, t_n) \, dx - \int_{t_{n-1}}^{t_n} p\big(r(t; x_{i+1/2}, t_n), t\big) \frac{\partial r(t; x_{i+1/2}, t_n)}{\partial t} \, dt. \quad (2.10)$$

Incorporating equations (2.8)–(2.18) into (2.7) we get

$$\int_{\Omega_i^n} \frac{\partial p}{\partial t} \, dx dt = \int_{x_{i-1/2}}^{x_{i+1/2}} p(x, t_n) dx - \int_{x_{i-1/2}^*}^{x_{i+1/2}^*} p(x, t_{n-1}) dx$$

$$+ \int_{t_{n-1}}^{t_n} p\big(r(t; x_{i-1/2}, t_n), t\big) \frac{\partial r(t; x_{i-1/2}, t_n)}{\partial t} \, dt \quad (2.11)$$

$$- \int_{t_{n-1}}^{t_n} p\big(r(t; x_{i+1/2}, t_n), t\big) \frac{\partial r(t; x_{i+1/2}, t_n)}{\partial t} \, dt.$$

The derivation shows that (2.11) does not depend on the assumption (2.4).

Now we just set the space-time boundaries $r(t; x_{i\pm1/2}, t_n)$ of the control volume Ω_i^n to be the characteristic curves, which are defined by the initial value problem of the ordinary differential equation

$$\frac{dr}{dt} = a(r, t), \qquad r(t; x, t_n)\,|_{t=t_n} = x. \quad (2.12)$$

That is, because the characteristics $r(t; x_{i\pm1/2}, t_n)$ are assumed to be tracked exactly and hence the residual advection term vanishes naturally.

Substituting (2.11) for the accumulation term in (2.5), we obtain a locally conservative reference equation on an interior space-time control volume Ω_i^n as follows:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} p(x, t_n) \, dx + \int_{t_{n-1}}^{t_n} F(p, b)\big(r(t; x_{i+1/2}, t_n), t\big) \, dt$$

$$- \int_{t_{n-1}}^{t_n} F(p, b)\big(r(t; x_{i-1/2}, t_n), t\big) \, dt \quad (2.13)$$

$$= \int_{x_{i-1/2}^*}^{x_{i+1/2}^*} p(x, t_{n-1}) \, dx,$$

which can be approximated as

$$\int_{x_{i-1/2}}^{x_{i+1/2}} p(x, t_n) \, dx + \Delta t \big[F(p, b)(x_{i+1/2}, t_n) - F(p, b)(x_{i-1/2}, t_n) \big]$$
$$= \int_{x_{i-1/2}^*}^{x_{i+1/2}^*} p(x, t_{n-1}) \, dx, \tag{2.14}$$

with the Lagrangian interface fluxes

$$F(p, b)\big(r(t; x_{i\pm1/2}, t_n), t\big) := -b \, \frac{\partial^{1-\lambda}}{\partial x^{1-\lambda}} p\big(r(t; x_{i\pm1/2}, t_n), t\big). \tag{2.15}$$

Note that the interface fluxes $F(p, b)(r(t; x_{i\pm1/2}, t_n), t)$ are defined across the space-time boundary $r(t; x_{i-1/2}, t_n)$ and $r(t; x_{i+1/2}, t_n)$ of the space-time control volume Ω_i^n.

To discretize equation (2.1), we further let $\{\phi_i\}_{i=1}^I$ be a set of hat functions such that $\phi_i(x_i) = 1$ and $\phi_i(x_j) = 0$ for $j \neq i$. Hence the finite volume approximation p_h to the true solution p can be expressed as

$$p_h(x) = \sum_{j=1}^{I-1} p_j \phi_j(x).$$

and the finite volume scheme on the interval $[x_{i-1/2}, x_{i+1/2}]$ for $i = 1, \ldots, I-1$ has the form

$$\int_{x_{i-1/2}}^{x_{i+1/2}} p(x, t_n) \, dx + \Delta t \sum_{j=1}^{I-1} p_j^n z_{i,j} = \int_{x_{i-1/2}^*}^{x_{i+1/2}^*} p(x, t_{n-1}) \, dx, \tag{2.16}$$

where, $[z_{i,j}]_{i,j=1}^{I-1}$ is the coefficient matrix of the fractional term, and is given by

$$z_{i,j} = b\big[\big(\gamma_0 D_{x_{i-1/2}}^{-\lambda} + (1 - \gamma)_{x_{1-1/2}} D_1^{-\lambda}\big) D\phi_j(x_{i-1/2,t_n})$$
$$- \big(\gamma_0 D_{x+1/2}^{-\lambda} + (1 - \gamma)_{x_{1+1/2}} D_1^{-\lambda}\big) D\phi_j(x_{i+1/2,t_n})\big]. \tag{2.17}$$

Assuming the trial functions $p(x, t_n)$ are chosen to be piecewise linear functions on $[x_l, x_r]$ with respect to the fixed spatial partition $\mathcal{P}_x$, we evaluate the accumulation term at time step t_n in the finite volume scheme (2.16) analytically as

$$\int_{x_{i-1/2}}^{x_{i+1/2}} p(x, t_n) \, dx = \frac{1}{8} \big[\Delta x_i (p_{i-1}^n + 3p_i^n) + \Delta x_{i+1}(3p_i^n + p_{i+1}^n)\big]. \tag{2.18}$$

In addition, we can compute $x_{i-1/2}^*$ and $x_{i+1/2}^*$ at time step t_{n-1} via a backward approximate characteristic tracking. Since the trial function $p(x, t_{n-1})$ is also piecewise linear with respect to the fixed spatial partition $\mathcal{P}_x$ at time t_{n-1}, we can evaluate the accumulation term at time step t_{n-1} analytically. Because the accumulation term at time step t_{n-1} affects only the right-hand side of the finite volume scheme (2.16), the scheme retains a symmetric and positive-definite coefficient matrix. Furthermore, the finite volume scheme (2.16) is locally conservative, even if the characteristics are computed approximately.

2.2.2 The fractional diffusion term and the stiffness matrix

The finite volume scheme (2.16) appears similar to the one for the canonical second-order diffusion equation, but has a fundamental difference. Although the hat functions ϕ_j have local support, (2.2) reveals that $_{x_l}D_x^{-\lambda}\phi_j$ and $_xD_{x_r}^{-\lambda}\phi_j$ have global support. Therefore, the stiffness matrix $Z := [z_{i,j}]_{i,j=1}^{I-1}$ is a full matrix, which requires $O(N^2)$ of memory to store. Numerical schemes for space-fractional differential equations were traditionally solved by the Gaussian type direct solvers that require $O(N^3)$ of computations [13, 14, 18, 19]. In recent years, there have been some other notable developments in methods for solving the algebraic linear systems arising from discretization of fractional-order problems, especially for space dimension higher than one (see [9]); we plan to explore them in our future work.

To simplify the computation of the diffusion term, we have to explore the structure of the stiffness matrix Z [17, 26]. The entries of the stiffness matrix Z are presented below.

Its diagonal entries are given by:

$$
\begin{aligned}
z_{i,i} &= \frac{1}{\Gamma(\lambda+1)2^\lambda}b\gamma h^{\lambda-1} + \frac{1}{\Gamma(\lambda+1)}b(1-\gamma)h^{\lambda-1}\left(2\left(\frac{1}{2}\right)^\lambda - \left(\frac{3}{2}\right)^\lambda\right) \\
&\quad + \frac{1}{\Gamma(\lambda+1)}b\gamma h^{\lambda-1}\left(2\left(\frac{1}{2}\right)^\lambda - \left(\frac{3}{2}\right)^\lambda\right) \\
&\quad + \frac{1}{\Gamma(\lambda+1)2^\lambda}b(1-\gamma)h^{\lambda-1} \\
&= \frac{1}{\Gamma(\lambda+1)}bh^{\lambda-1}\left(2^{-\lambda} + 2\left(\frac{1}{2}\right)^\lambda - \left(\frac{3}{2}\right)^\lambda\right),
\end{aligned}
\tag{2.19}
$$

for $1 \le i \le I-1$.

Whereas, the sub-triangular entries of the matrix Z are given by:

$$
\begin{aligned}
z_{i,i-1} &= -\frac{1}{\Gamma(\lambda+1)}b\gamma h^{\lambda-1}\left(2\left(\frac{1}{2}\right)^\lambda - \left(\frac{3}{2}\right)^\lambda\right) - \frac{1}{\Gamma(\lambda+1)2^\lambda}b(1-\gamma)h^{\lambda-1} \\
&\quad - \frac{1}{\Gamma(\lambda+1)}b\gamma h^{\lambda-1}\left(\left(\frac{5}{2}\right)^\lambda - 2\left(\frac{3}{2}\right)^\lambda + \left(\frac{1}{2}\right)^\lambda\right) \\
&= \frac{1}{\Gamma(\lambda+1)}bh^{\lambda-1}\left(3\left(\frac{3}{2}\right)^\lambda\gamma - 3\left(\frac{1}{2}\right)^\lambda\gamma - \left(\frac{5}{2}\right)^\lambda\gamma - 2^{-\lambda}(1-\gamma)\right),
\end{aligned}
\tag{2.20}
$$

for $2 \le i \le I-1$, and

$$
\begin{aligned}
z_{i,j} &= \frac{1}{\Gamma(\lambda+1)}b\gamma h^{\lambda-1}\left(\left(i-j+\frac{1}{2}\right)^\lambda - 2\left(i-j-\frac{1}{2}\right)^\lambda + \left(i-j-\frac{3}{2}\right)^\lambda\right) \\
&\quad - \frac{1}{\Gamma(\lambda+1)}b\gamma h^{\lambda-1}\left(\left(i-j+\frac{3}{2}\right)^\lambda - 2\left(i-j+\frac{1}{2}\right)^\lambda + \left(i-j-\frac{1}{2}\right)^\lambda\right)
\end{aligned}
\tag{2.21}
$$

for $3 \le i \le I-1$ and $1 \le j \le i-2$.

The super-triangular entries of matrix Z can be also derived as

$$z_{i,i+1} = -\frac{1}{\Gamma(\lambda+1)}b(1-\gamma)h^{\lambda-1}\left(\left(\frac{1}{2}\right)^{\lambda} - 2\left(\frac{3}{2}\right)^{\lambda} + \left(\frac{5}{2}\right)^{\lambda}\right)$$

$$-\frac{1}{\Gamma(\lambda+1)2^{\lambda}}b\gamma h^{\lambda-1} - \frac{1}{\Gamma(\lambda+1)}b(1-\gamma)h^{\lambda-1}\left(2\left(\frac{1}{2}\right)^{\lambda} - \left(\frac{3}{2}\right)^{\lambda}\right)$$

$$= \frac{1}{\Gamma(\lambda+1)}bh^{\lambda-1}\left(3\left(\frac{3}{2}\right)^{\lambda}(1-\gamma) - 3\left(\frac{1}{2}\right)^{\lambda}(1-\gamma)\right.$$

$$\left. -\left(\frac{5}{2}\right)^{\lambda}(1-\gamma) - 2^{-\lambda}\gamma\right), \tag{2.22}$$

for $1 \leq i \leq I - 2$, and

$$z_{i,j} = \frac{1}{\Gamma(\lambda+1)}b(1-\gamma)h^{\lambda-1}\left(2\left(j-i+\frac{1}{2}\right)^{\lambda}\right.$$

$$\left. -\left(j-i-\frac{1}{2}\right)^{\lambda} - \left(j-i+\frac{3}{2}\right)^{\lambda}\right)$$

$$-\frac{1}{\Gamma(\lambda+1)}b(1-\gamma)h^{\lambda-1}\left(2\left(j-i-\frac{1}{2}\right)^{\lambda}\right.$$

$$\left. -\left(j-i-\frac{3}{2}\right)^{\lambda} - \left(j-i+\frac{1}{2}\right)^{\lambda}\right) \tag{2.23}$$

for $1 \leq j \leq I - 3$ and $j + 2 \leq i \leq I - 1$.

2.2.3 Estimation of coefficient function

We present the case where the drift coefficient function in fADE (1.5) (and therefore in equation (2.1)) is assumed to be a piecewise linear function of x,

$$a(x) = \begin{cases} a_0 - a_1 x, & \text{if } x_l \leq x \leq x_m, \\ a_2 - a_3 x, & \text{if } x_m < x \leq x_r, \end{cases} \tag{2.24}$$

where parameters a_0, a_1, a_2 and a_3 are to be estimated from the observed concentration data. The main idea on this part is to obtain certain measurements through physical or mechanical experiments, and then use the data to calibrate these parameters in the fADE (see [16]). This is an inverse problem: based on the initial guess p_0 of the equation (2.1), and certain observation (or desired) data such as values of the state variable **g** at the final time, we attempt to seek for the constant parameters a_0, a_1, a_2 and a_3 from the governing differential equation (2.1).

We formulate the inverse problem as an optimization and develop a Levenberg–Marquardt (L-M) regularization method (see, [12, 20, 24]) to iteratively identify the parameter. It is known that the inverse problem usually requires multiple runs of the forward problem. Considering the computational cost of the forward problem is already high, the inverse problem could become infeasible. Hence we propose an optimization algorithm for the parameter estimation.

Here we only present the details of the fitting of a_0 and a_1. The other two parameters a_2 and a_3 can be estimated similarly. The parameter identification of a_0 and a_1

can be formulated as follows: let $\boldsymbol{\alpha} := \{a_0, a_1\}$, then to find α_{inv} that satisfies

$$\alpha_{inv} = \arg \min_{\boldsymbol{\alpha}} \; \mathcal{G}(\boldsymbol{\alpha}) := \frac{1}{2} \sum_{i=1}^{M} \left[p(x_i, t; \boldsymbol{\alpha}) - \hat{p}(x_i, t) \right]^2, \qquad (2.25)$$

where, $\hat{p}(x_i, t) = \hat{K}_t c(x_i, t)$ for observed concentration c at location x_i, time t.

In case the data is available for time points $t_1, t_2, \ldots, t_R$ we rewrite equation (2.25) as

$$\alpha_{inv} = \arg \min_{\boldsymbol{\alpha}} \; \mathcal{G}(\boldsymbol{\alpha})$$
$$= \frac{1}{2} \sum_{k=1}^{R} \sum_{i=1}^{M_k} w_k \left[p(x_{t_k,i}, t_k; \boldsymbol{\alpha}) - \hat{p}(x_{t_k i}, t_k) \right]^2, \qquad (2.26)$$

here $x_{t_k,i}$ is the ith observed location at time t_k and w_k is the weight assigned for the sample set available for the same time point. (For example the data set used in Section 3 has $R = 3$).

An iteration algorithm such as the Newton method with line searching could be employed to find the solution to (2.26). Basically, the Newton algorithm for minimizing (2.26) uses the first and second derivatives of the objective function $\mathcal{G}(\boldsymbol{\alpha})$:

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \frac{\mathcal{G}'(\boldsymbol{\alpha}_k)}{\mathcal{G}''(\boldsymbol{\alpha}_k)}, \qquad (2.27)$$

where k represents the kth iteration. It is easy to check that (2.27) is equivalent to solve

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \left(\mathbf{J}_k^T \mathbf{J}_k \right)^{-1} \mathbf{J}_k^T \mathbf{r}_k, \qquad (2.28)$$

where

$$\mathbf{J}_k = \left(\mathbf{J}^1; \mathbf{J}^2; \ldots; \mathbf{J}^R \right)^T, \quad \text{with}$$
$$\mathbf{J}^i = \left(\frac{\partial p(x_{t_i,1}, t_i; \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}, \ldots, \frac{\partial p(x_{t_i, M_i}, t_i; \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \right), \qquad (2.29)$$

and

$$\mathbf{r}_k = \left(\mathbf{r}^1; \mathbf{r}^2; \ldots; \mathbf{r}^R \right)^T, \quad \text{with}$$
$$\mathbf{r}^i = (r_{i,1}, \ldots, r_{i,M_i}), \quad r_{i,j} = p(x_{t_i,j}, t_i; \boldsymbol{\alpha}) - \hat{p}(x_{t_i,j}, t_i), \qquad (2.30)$$

for $j = 1, \ldots, M_i$, and $i = 1, 2, \ldots, R$. Note that in practice, we always use the finite difference

$$\frac{p(x_{t_i,j}, t_i; \boldsymbol{\alpha} + \delta) - p(x_{t_i,j}, t_i; \boldsymbol{\alpha})}{\delta}$$

with a small enough δ to approximate the derivatives in (2.29).

However, the Newton method may fail to work because of $\mathbf{J}_k^T \mathbf{J}_k$ may be nearly zero. Therefore, the search direction $d_k := -\mathbf{J}_k^T \mathbf{r}_k / \mathbf{J}_k^T \mathbf{J}_k$ may not point in a decent direction.

A common technique to overcome this kind of problem is the L-M algorithm (or Levenberg algorithm since a single parameter case is considered in this paper), which modifies (2.27) by the following formulation

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \left(\mathbf{J}_k^T \mathbf{J}_k + \varrho_k I_2 \right)^{-1} \mathbf{J}_k^T \mathbf{r}_k, \qquad (2.31)$$

where ϱ_k is a positive penalty parameter, and I_2 is a 2×2 identity matrix. The method coincides with the Newton algorithm when $\varrho_k = 0$; and it gives a step closer to the gradient descent direction when ϱ_k is large.

Algorithm A: Parameter Identification Algorithm:

Given the observation data **g** and an initial guess $\boldsymbol{\alpha}_0 = \{a_0, a_1\}_0$, choose $\rho \in (0, 1)$, $\sigma \in (0, 1/2)$, $\varrho_0 > 0$, and δ small enough. For $k = 0, 1, \ldots, M$, (let $M = M_1 + M_2 + M_3$),

(1) Solve the equation (2.1) corresponding to $\boldsymbol{\alpha}_k$ and $\boldsymbol{\alpha}_k + \delta$ respectively to obtain $p(\cdot, \cdot, \boldsymbol{\alpha}_k)$ and $p(\cdot, \cdot, \boldsymbol{\alpha}_k + \delta)$.

(2) Compute $\mathbf{J}_k$ and $\mathbf{r}_k$, and update the search direction

$$\mathbf{d}_k := -\left(\mathbf{J}_k^T \mathbf{J}_k + \varrho_k I_2\right)^{-1} \mathbf{J}_k^T \mathbf{r}_k.$$

(3) Determine the search step ρ^m by Armijo rule:

$$\mathcal{G}\left(\boldsymbol{\alpha}_k + \rho^m \mathbf{d}_k\right) \leq \mathcal{G}(\boldsymbol{\alpha}_k) + \sigma \rho^m \mathbf{d}_k \mathbf{J}_k^T \mathbf{r}_k$$

where m is the smallest non-negative integer.

(4) If $|\rho^m \mathbf{d}_k| \leq$ Tolerance level, then stop and let $\alpha_{inv} := \boldsymbol{\alpha}_k$. Otherwise, update

$$\boldsymbol{\alpha}_{k+1} := \boldsymbol{\alpha}_k + \rho^m \mathbf{d}_k, \qquad \varrho_{k+1} := \varrho_k/2,$$

and go to the first step for the iteration again.

Algorithm A summarizes the proposed parameter estimation steps through the inverse problem approach which includes the details of the L-M method. In particular, the Armijo rule [2] in the third step, known as one of the inexact line search techniques, is imposed to ensure the objective function $\mathcal{G}$ has sufficient descent. Other rules and the related convergence theory can be found in [24].

3   An illustrative example

In this section, we use a heavy-tailed groundwater tracer concentration data to illustrate the methodology described in Section 2. The data comes from natural-gradient tracer tests conducted at the MacroDispersion Experimental (MADE) site at Columbus Air Force Base in northeastern Mississippi, precisely the MADE-2 tritium plume data [8]. The data consists of the maximum concentration measured in vertical slices perpendicular to the direction of plume travel, at day 27, day 132, day 224, and day 328 days after injection (see Figure 4 in [5]).

In [11], an fADE with constant drift and diffusion coefficient was fitted to the same data. With constant coefficient functions, the resulting solution process Y_t in the SDE (1.2) itself is a stable process. The parameters of the fitted stable process were estimated using the least-square method.

Here we are considering an fADE that may have variable coefficients. For the fits presented in this section, the diffusion coefficient is still assumed to be location invariant, but we use a piecewise linear drift function of the form (2.24). The choice of x_l, x_m and x_r are subjective but should comply with the boundary conditions. $x_l = 0$ (starting location), $x_r = 300$ (beyond the observed location range) and $x_m = 9.375$ (a rough mid-location of the observed range) were set for the current simulation. We used the fitted parameter values from the constant coefficient model [11] as initial values for the simulation; these parameter values are included in Appendix A. Then we used the data to estimate the linear drift and to obtain $p(x, t)$ that solves (2.1) by the methodology presented in Section 2. $c(x, t) = Kp(x, t)$ gives the fitted concentration that solves fADE associated with (2.1).

The concentrations fitted by the proposed method for the MADE site data at day 224 and day 328 are included here.

Fitted parameters for the MADE site data at day 224:
$a_0 = 0.110$, $a_1 = 0.00032$, $a_2 = 0.0003$, $a_3 = 0.00019$,
$b = 0.1859783$, $\lambda = 0.80$, $\gamma = 0.9999$, $K = 56778.24$.

Table 1. Observed and fitted tracer concentrations for MADE data at day 224

Day 224		
x = location, C = tracer concentration		
x	Observed C	Fitted C
2.1000	3378.0000	2871.7248
2.9000	1457.0000	3187.0093
3.6000	6494.0000	3529.6230
6.0000	1335.0000	4027.3618
6.6000	7705.0000	3321.0305
6.8000	2206.0000	3086.6518
6.9000	1291.0000	2969.4625
7.2000	4515.0000	2734.3773
8.0000	3598.0000	2361.1656
8.7000	2447.0000	2083.3103
9.7000	2831.0000	1809.0524
10.8000	2208.0000	1586.8821
12.7000	849.0000	1273.2312
13.5000	2213.0000	1169.6987
16.0000	1485.0000	935.2569
27.5000	443.0000	442.0004
31.8000	165.0000	356.0560
40.3000	291.0000	245.6054
48.8000	237.0000	178.6647
57.5000	76.0000	134.2397
66.3000	54.0000	103.7457
74.8000	137.0000	82.8644
83.4000	37.0000	67.3318
100.7000	28.0000	46.5417
117.0000	51.0000	34.4212
166.6000	18.0000	16.5498
183.5000	28.0000	13.9355
218.5000	9.0000	11.4791
268.7000	6.0000	10.1389

Fitted parameters for day 328 MADE site data:
$a_0 = 0.105, a_1 = 0.00030, a_2 = 0.0005, a_3 = 0.00018,$
$b = 0.2233695, \lambda = 0.79, \gamma = 0.9999, K = 37195.05.$

Table 2. Observed and fitted tracer concentrations for MADE data at day 328

Day 328		
x = location, C = tracer concentration		
x	Observed C	Fitted C
2.1000	1762.0000	2001.9586
3.6000	4157.0000	3264.5572
5.5000	2018.0000	1982.1889
6.0000	1295.0000	1653.7130
6.6000	2742.0000	1436.3243
6.8000	798.0000	1405.9525
6.9000	661.0000	1390.7665
7.2000	4721.0000	1323.8183
8.0000	2877.0000	1132.2689
8.7000	946.0000	1050.1787
9.7000	3140.0000	920.6741
10.8000	2075.0000	819.8101
12.7000	1636.0000	689.5556
13.5000	1569.0000	646.4434
16.0000	869.0000	536.8714
16.3000	1825.0000	525.8232
17.0000	286.0000	501.7569
18.0000	597.0000	470.2962
27.5000	127.0000	281.0554
31.8000	49.0000	231.1571
40.3000	108.0000	163.9576
48.8000	75.0000	121.3509
57.5000	95.0000	92.2235
66.3000	38.0000	71.8197
74.8000	126.0000	57.6513
83.4000	54.0000	47.0086
100.7000	19.0000	32.6278
117.0000	46.0000	24.1752
166.6000	25.0000	11.6326
183.5000	22.0000	9.4787
218.5000	11.0000	7.0942
268.7000	8.0000	6.0752

Comments:

1. The fit above uses piecewise linear drift velocity and location invariant diffusion coefficient as opposed to constant drift but same diffusion coefficient used in [11]. The fitted curves in Figures 2 and 4 show a better tail fit here compared to similar plots in [11]. The fractional order α for the fADE with constant coefficient was fitted as 1.09 and 1.05 for day 224 and day 328 respectively in [11]. Here the fitted fractional order $\alpha (= 2 - \lambda)$ are 1.2 and 1.19 for day 224 and day 328 respectively. Since the new fit improves on the fit in [11], this indicates that a better drift velocity fit may lead to an adjustment of the fractional order to improve the tail fit in the fADE. Ideally, the drift velocity function might be mechanically estimated from a properly designed field experiment while the
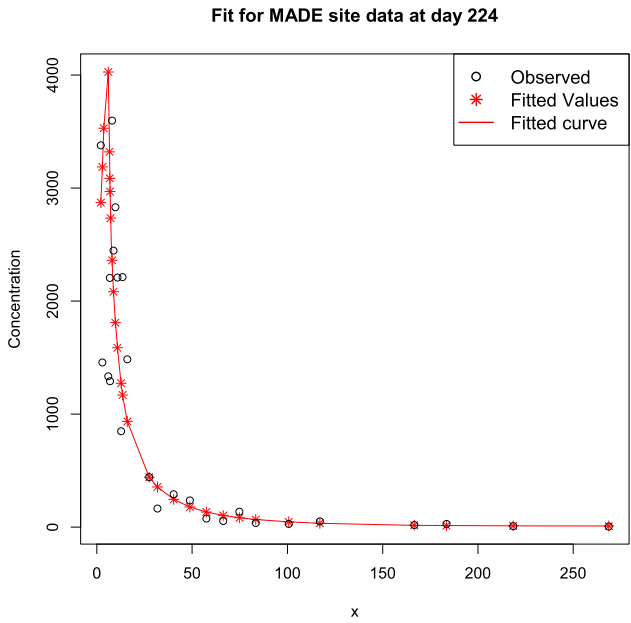
Fit for MADE site data at day 224

Fig. 1. The fitted and observed concentration at day 224 for MADE site data

Log-Log Plot for MADE site data at day 224
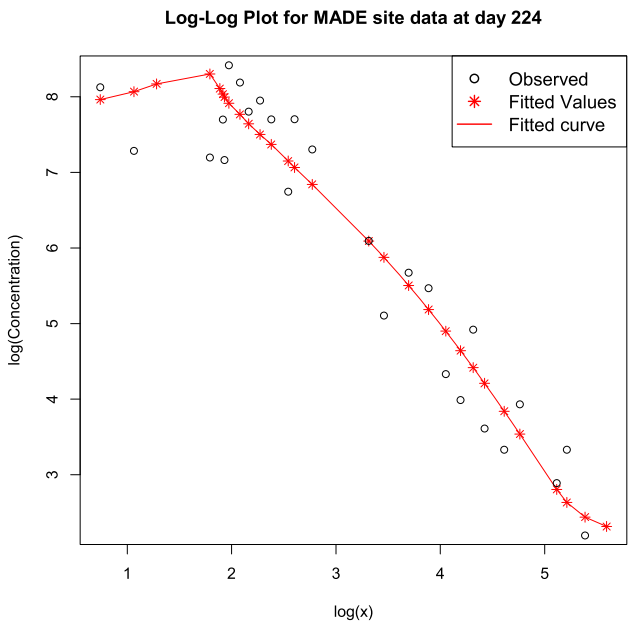
Fig. 2. The fitted and observed concentration with log scale at day 224 for MADE site data

Fit for MADE site data at day 328
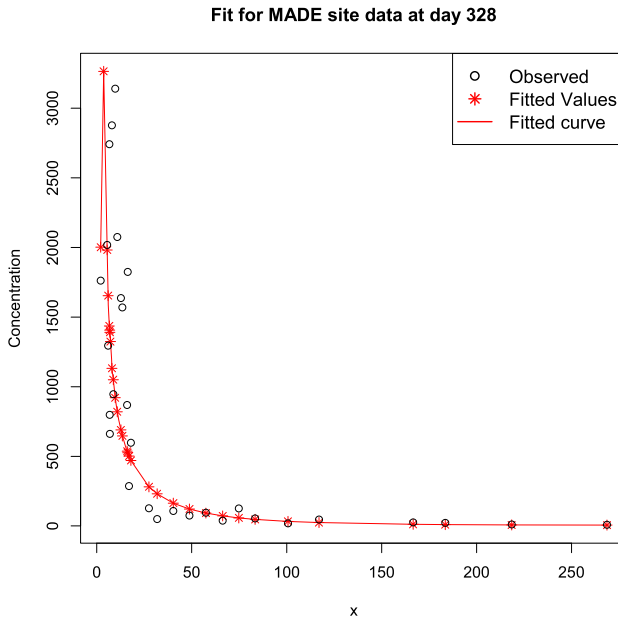
Fig. 3. The fitted and observed concentration at day 328 for MADE site data

Log-Log Plot for MADE site data at day 328
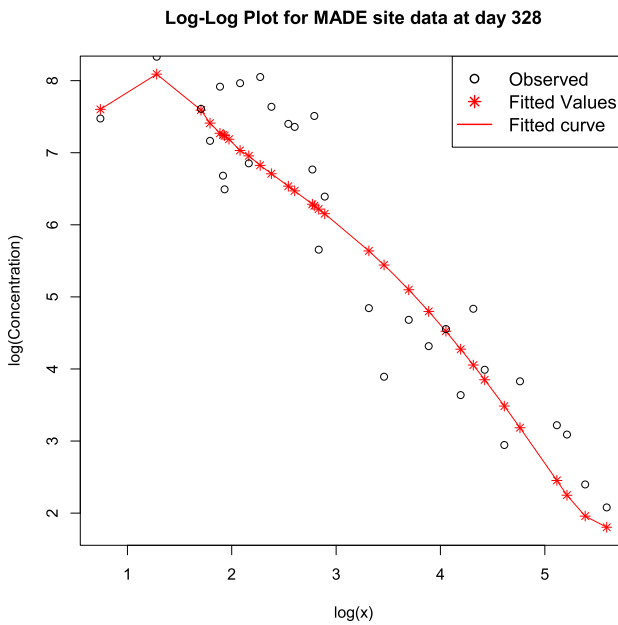
Fig. 4. Fitted and observed concentration with log scale at day 328 for MADE site data

diffusion coefficient and the fractional order can be estimated as described in this paper for an even better fADE fit.

2. Using simulation step 2, iid sample of size 1000 for Y_t was generated at $t =$ day 224 and $t =$ day 328. Histograms for the generated samples in Figures 5 and 6 are fairly consistent with the observed concentration (or observed density) data. These figures indicate that although we had adopted a numerical approach to generate Y_t's, the simulation method is efficient enough to follow the underlying probability distribution.
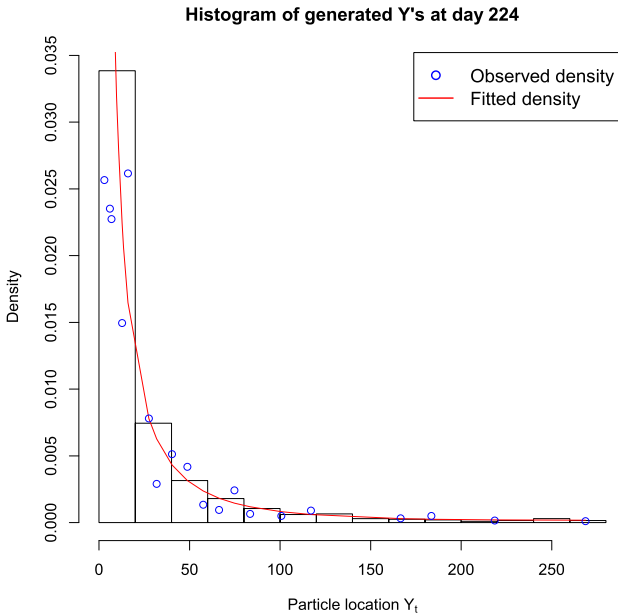
Histogram of generated Y's at day 224

Fig. 5. Histogram of 1000 Y_t values generated by the proposed method at $t = 224$. The red line shows the fitted pdf from (2.1) and blue dots show observed density = observed concentration/K (data estimates $K = 56778.24$)

4 Discussion

The trapezoidal rule provides a very simple and quick numerical approximation of the CDF of Y_t in (1.2) and for its inverse in the proposed simulation step 2. A more sophisticated numerical method like Simpson's rule or other higher order Newton–Cotes formula or quadrature rules [23] can be used to obtain a more accurate $\hat{F}_{Y_t}$, but the procedure will be much more computationally taxing.

The method described here can be used to simulate a large number of observations (approximated) from Y_t. With the usual particle tracking insights, these simulated values can also be used to build empirical confidence intervals for the fitted concentration values.
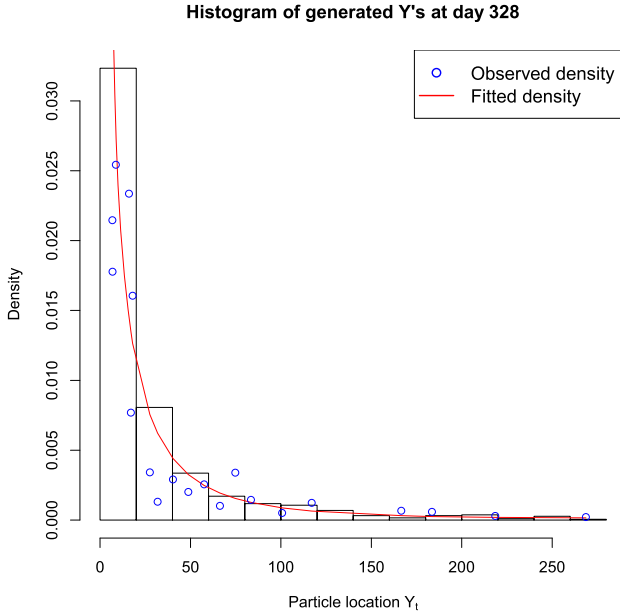
Histogram of generated Y's at day 328

Fig. 6. Histogram of 1000 Y_t values generated by the proposed method at $t = 328$. The red line shows the fitted pdf from (2.1) and blue dots show observed density = observed concentration $/K$ (data estimates $K = 37195.05$)

Let us consider $P[Y_t \in (a, b)] = F_{Y_t}(b) - F_{Y_t}(a)$. This is the probability that a randomly chosen tracer particle will be in a given interval (a, b) at a time point t. For example, in groundwater pollution modeling this expression can be an important one that estimates the chance that the pollution will reach a certain area after a certain time. Using i.i.d. random simulations $Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(N)}$ of Y_t, an empirical estimate for this probability can be given by $\hat{P} = \frac{1}{N} \sum_{i=1}^{N} I[Y_t^{(i)} \in (a, b)]$. Then by the central limit theorem $\hat{P}$ asymptotically follows the normal distribution with mean $P[Y_t \in (a, b)]$ and variance $\dfrac{P[Y_t \in (a, b)](1 - P[Y_t \in (a, b)])}{N}$. Hence for large N's, an asymptotic $(1 - \alpha)100\%$ confidence interval for $P[Y_t \in (a, b)]$ can be given by

$$\hat{P} \mp z_{\alpha/2} \sqrt{\frac{\hat{P}(1 - \hat{P})}{N}} \tag{4.1}$$

Now noting that density $p(y, t)$ can be approximated by $\frac{1}{b-a} P[Y_t \in (a, b)]$ with a small interval (a, b), the simulated Y_t values can be used to calculate the confidence intervals for the fitted densities. See [11] for detailed asymptotic confidence interval construction steps and related results associated with the empirical density function that is calculated using the generated Y_t's.

Assuming that the observed concentrations are scaled versions of density $p(y, t)$ of Y_t, one can repeat simulation step 2 described in Section 2.1 without solving the

forward equation to generate Y_t's. However, the grid used for the simulation will be limited only to the fixed observed data location points. Since Y_t's are simulated through numerical approximation the performance of this simulation depends on the grid spacing used. Thus the fitted fADE in simulation step 1 not only gives us the better understanding of the underlying process but also is essential for generating a good approximate process that resembles Y_t by enabling the use of a finer grid for the numerical approximation. Further, modeling the underlying fADE process facilitates better prediction than mere extrapolation from an observed sample set.

On the other hand, an fADE can be fitted to the observed concentration by simulation step 1 only, without describing the underlying stochastic process. But the stochastic diffusion description is useful for understanding the mesoscopic flow and particle tracking methods. Further, generating the solution to the SDE associated with the fADE can be used to construct confidence intervals for the fADE fits that account for the error of estimation by a sample set.

In conclusion, the fADE and the associated α-stable SDE are essential tools to model heavy-tailed diffusion. However, the model fitting part may get complicated if we consider any scenario other than constant drift and diffusion coefficients. A numerical scheme for solving the fADE and the related SDE with linear drift is presented here. The general ideas used in the simulation steps can be replicated for a more complicated form of the drift. We plan to explore such models in our future work along with the convergence and the efficiency of the numerical approximations applied here.

A Appendix

In [11] the tracer trajectory Y_t of the MADE site data was assumed to follow an SDE of the form (1.2) with constant parameters and hence itself was a stable process with index of stability α, skewness parameter β, location parameter v (same as the assumed constant drift velocity in (1.5)) and scale parameter σ, where $\sigma^\alpha = Dt|\cos(\pi\alpha/2)|$ with constant diffusion coefficient function D in (1.5).

The fitted values in [11] are:

Day 224 fitted values:
$\sigma = 5.137167, \mu = 43.915430;$
$\beta = 0.99;$
$\alpha = 1.0915;$
$v = 0.196051 m/day$
$D = 0.1859783 m^\alpha/day$
$K = 56778.24.$

Day 328 fitted values:
$\sigma = 5.380654; \mu = 74.677975;$
$\beta = 0.99;$
$\alpha = 1.050998;$
$v = 0.2276768 m/day$
$D = 0.2233695 m^\alpha/day$
$K = 37195.05$

These were used as initial values for the iterations in Section 3.

References

[1] Applebaum, D.: Lévy Processes and Stochastic Calculus. Cambridge University Press (2004). MR2072890. https://doi.org/10.1017/CBO9780511755323

[2] Armijo, L.: Minimization of functions having Lipschitz continuous partial derivatives. Pac. J. Math. **16**, 1–3 (1966). MR0191071

[3] Baeumer, B., Meerschaert, M.M.: Stochastic solutions for fractional Cauchy problems. Fract. Calc. Appl. Anal. **4**, 481–500 (2001). MR1874479

[4] Bear, J.: Dynamics of Fluids in Porous Media. Dover, Mineola, New York (1972)

[5] Benson, D.A., Schumer, R., Meerschaert, M.M., Wheatcraft, S.W.: Fractional dispersion, Lévy motion, and the MADE tracer tests. Transp. Porous Media **42**(1–2), 211–240 (2001). MR1948593. https://doi.org/10.1023/A:1006733002131

[6] Bhattacharya, R.N., Gupta, V.K.: A theoretical explanation of solute dispersion in saturated porous media at the Darcy scale. Water Resour. Res. **19**(4), 938–944 (1983)

[7] Bhattacharya, R.N., Gupta, V.K., Sposito, G.: On the stochastic foundation of the theory of water flow through unsaturated soil. Water Resour. Res. **12**(3) (1976)

[8] Boggs, J.M., Beard, L.M., Long, S.E., McGee, M.P.: Database for the second macro dispersion experiment (MADE-2). EPRI report TR-102072 (1993)

[9] Breiten, T., Simoncini, V., Stoll, M.: Low-rank solvers for fractional differential equations. Electron. Trans. Numer. Anal. (ETNA) **45**, 107–132 (2016). MR3498143

[10] Chakraborty, P.: A stochastic differential equation model with jumps for fractional advection and dispersion. J. Stat. Phys. **136**(3), 527–551 (2009). MR2529683. https://doi.org/10.1007/s10955-009-9794-1

[11] Chakraborty, P., Meerschaert, M.M., Lim, C.Y.: Parameter estimation for fractional transport: A particle tracking approach. Water Resour. Res. **45**(10), 10415 (2009)

[12] Chavent, G.: Nonlinear Least Squares for Inverse Problems: Theoretical Foundations and Step-by-Step Guide for Applications. Springer, Netherlands (2009). MR2554448

[13] del-Castillo-Negrete, D., Carreras, B.A., Lynch, V.E.: Fractional diffusion in plasma turbulence. Phys. Plasmas **11**, 3854 (2004)

[14] Ervin, V.J., Roop, J.P.: Variational formulation for the stationary fractional advection dispersion equation. Numer. Methods Partial Differ. Equ. **22**, 558–576 (2005). MR2212226. https://doi.org/10.1002/num.20112

[15] Fetter, C.W.: Applied Hydrogeology. Prentice Hall, New York (2001)

[16] Fu, H., Wang, H., Wang, Z.: POD/DEIM reduced-order modeling of time-fractional partial differential equations with applications in parameter identification. J. Sci. Comput. **74**, 220–243 (2018). MR3742877. https://doi.org/10.1007/s10915-017-0433-8

[17] Jia, J., Wang, H.: A preconditioned fast finite volume scheme for a fractional differential equation discretized on a locally refined composite mesh. J. Comput. Phys. **299**, 842–862 (2015). MR3384754. https://doi.org/10.1016/j.jcp.2015.06.028

[18] Liu, F., Anh, V., Turner, I.: Numerical solution of the space fractional Fokker-Plank equation. J. Comput. Appl. Math. **166**, 209–219 (2004). MR2057973. https://doi.org/10.1016/j.cam.2003.09.028

[19] Meerschaert, M., Tadjeran, C.: Finite difference approximations for fractional advection-dispersion flow equations. J. Comput. Appl. Math. **172**, 65–77 (2004). MR2091131. https://doi.org/10.1016/j.cam.2004.01.033

[20] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York, USA (2006). MR2244940

[21] Resnick, S.: A Probability Path. Birkhäuser, Boston (2003). MR1664717

[22] Samorodnitsky, G., Taqqu, M.S.: Stable Non-Gaussian Random Processes, Stochastic Models with Infinite Variance. Chapman & Hall, New York (1994). MR1280932

[23] Scarborough, J.B.: Numerical Mathematical Analysis. Johns Hopkins Press (1966). MR0198651

[24] Sun, W., Yuan, Y.: Optimization Theory and Methods: Nonlinear Programming. Springer, New York, USA (2006). MR2232297

[25] Wang, H., Al-Lawatia, M.: A locally conservative Eulerian-Lagrangian control-volume method for transient advection-diffusion equations. Numer. Methods Partial Differ. Equ. **22**, 577–599 (2006). MR2212227. https://doi.org/10.1002/num.20106

[26] Wang, H., Du, N.: A superfast-preconditioned iterative method for steady-state space-fractional diffusion equations. J. Comput. Phys. **240**, 49–57 (2013). MR3039244. https://doi.org/10.1016/j.jcp.2012.07.045

[27] Zhang, Y., Benson, D., Meerschaert, M.M., LaBolle, E.: Space-fractional advection-dispersion equations with variable parameters: Diverse formulas, numerical solutions, and application to the MADE-site data. Water Resour. Res. **43**(5), 05439 (2007)