Numerical inversions of a source term in the FADE with a Dirichlet boundary condition using final observations

Guangsheng Chi a,b, Gongsheng Li a,∗, Xianzheng Jia a

a Institute of Applied Mathematics, Shandong University of Technology, Zibo, 255049, PR China
b Shandong Kaiwen College of Science and Technology, Jinan, 250200, PR China

ARTICLE INFO

Article history:
Received 23 October 2009
Received in revised form 11 October 2010
Accepted 21 February 2011

Keywords:
Fractional advection–dispersion equation (FADE)
Source term inversion
Optimal perturbation regularization algorithm
Numerical inversion

ABSTRACT

This paper deals with an inverse problem of determining a source term in the one-dimensional fractional advection–dispersion equation (FADE) with a Dirichlet boundary condition on a finite domain, using final observations. On the basis of the shifted Grünwald formula, a finite differences scheme for the forward problem of the FADE is given, by means of which the source magnitude depending upon the space variable is reconstructed numerically by applying an optimal perturbation regularization algorithm. Numerical inversions with noisy data are carried out for the unknowns taking three functional forms: polynomials, trigonometric functions and index functions. The reconstruction results show that the inversion algorithm is efficient for the inverse problem of determining source terms in a FADE, and the algorithm is also stable for additional data having random noises.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

It is critical to model solute transport behaviors in porous media for environmental protection and prediction. The common models are the ordinary advection–dispersion equation (ADE) and the fractional advection–dispersion equation (FADE). It is known that the ADE has played an important role in modeling solute transport processes during the past forty years, and quite a few research works on inverse problems for the ADE have appeared since the 1980s (see [1–5], for instance). However, some research indicated that the ADE was inadequate for simulating many real situations, where a particle plume spreads faster or slower than predicted by the classical model. For example, in paper [6], an anomalous diffusion with a systematic mass loss was revealed by spatial moments analysis for a natural gradient tracer experiment in a heterogeneous aquifer. So, the FADE, where the usual second derivative in space is replaced by a fractional derivative of order $0 < \alpha < 2$, as a model for simulating solute transport in porous media has been applied in both laboratory and field-scale experiments (see [7–12], for instance).

Since the FADE was first applied to model the anomalous diffusion process, quite a few methods have been developed for solving it numerically. In papers [13,14], a computational method of lines was proposed by which the space FADE was transformed into a system of ordinary differential equations that was solved by using back–forward differentiation formulas. In paper [15], a space–time FADE was considered and a solution in terms of Green functions was obtained by applying Fourier–Laplace transforms. In paper [16], an approximation of the Levy–Feller advection–dispersion process was discussed using a random walk and finite difference scheme. Two discrete schemes of Cauchy problems for the FADE with $0 < \alpha < 1$ and $1 < \alpha < 2$ were presented in paper [17], and a numerical example evaluating the theoretical analysis was given. Three numerical methods for a fractional partial differential equation with Riesz space fractional derivatives on a finite domain with homogeneous Dirichlet boundary conditions were discussed in paper [18], and a finite element method was utilized to solve the FADE in paper [19]. In paper [20], on the basis of a shifted Grünwald approximation to the fractional
derivative, explicit and implicit finite difference schemes for the FADE with mixed boundary conditions on a finite domain were discussed, and it was proved that the implicit scheme is consistent and unconditionally stable.

However, research on inverse problems for anomalous diffusion in porous media described by the FADE has not been paid much attention. In paper [21], a semi-analytical inversion method for solving the FADE was developed and the corresponding inversion program for parameter estimation was presented. Recently, in paper [22], an inverse problem of determining the order of the time fractional derivative and a spatially varying coefficient was investigated, and the uniqueness of the inverse problem was proved theoretically by means of additional boundary data. It is noticeable that most of the research on the FADE has focused on numerical methods of solving the forward problem, and inverse problems for the FADE have become fruitful topics that offer great potential.

In this paper, we will deal with an inverse problem of determining a source magnitude function in the one-dimensional FADE with homogeneous Dirichlet boundary conditions on a finite domain. Although the inverse problem of the FADE with zero Dirichlet boundary conditions discussed here can be transformed into a typical problem with overdetermination by the method of the analytic semigroup, we will focus on its numerical methods, and we will present efficient numerical inversions. On solving the forward problem of the FADE numerically, we will utilize the implicit difference scheme given in paper [20], although we will cope with different boundary conditions. As for the inversion of the source magnitude, we will apply the optimal perturbation regularization algorithm developed in papers [23,24]. We will show that the inversion algorithm is not only efficient for source parameter identification for the ADE, but also suitable for inverse problems of determining source terms for the FADE.

The paper is arranged as follows.

In Section 2, an inverse problem of determining a source term in the FADE is put forward, and an implicit difference scheme for solving the forward problem of the FADE is discussed. Section 3 gives an optimal perturbation regularization algorithm for determining the unknown source term. In Section 4, numerical simulations are carried out and several factors having important impacts on the inversion results are discussed. Finally, two concluding remarks are given.

2. The inverse problem and the difference scheme for the forward problem

Consider the following initial–boundary value problem for the fractional advection–dispersion equation (FADE):

\[
\begin{aligned}
\frac{\partial c(x, t)}{\partial t} &= -c_v \frac{\partial c(x, t)}{\partial x} + D \frac{\partial^\alpha c(x, t)}{\partial x^\alpha} + \lambda(t) f(x), \quad 0 < x < \pi, \ 0 < t < T, \\
c(x, 0) &= g(x), \quad 0 \leq x \leq \pi, \\
c(0, t) &= 0, \quad c(\pi, t) = 0, \quad 0 \leq t \leq T,
\end{aligned}
\]

(1)

where \(1 < \alpha \leq 2\) is the order of the fractional derivative, \(v\) is the velocity and \(D\) is the dispersion coefficient; here \(v\) and \(D\) are both constants, and the functions \(f(x)\) and \(g(x)\) are assumed to be continuous on \([0, \pi]\) with \(g(0) = 0\), and \(\lambda(t)\) is continuous on \([0, T]\).

Problem (1) is the forward problem when all the functions \(f(x)\), \(g(x)\) and \(\lambda(t)\), and the parameters \(v\) and \(D\) are given appropriately. The inverse source problem here is to determine the source magnitude function \(f(x)\) based on problem (1) and some additional information on the solution. In this paper we will utilize final observations as additional data:

\(c(x, T) = \theta(x), \quad 0 \leq x \leq \pi.\)

(2)

As a result, an inverse problem of determining the source term \(f(x)\) is formulated as problem (1) with additional condition (2).

Before performing inversion algorithms, we should give for the forward problem a suitable solving method.

Let the space step be \(h = l/M\), the time step be \(r = T/N\), with \(x_i = ih\) and \(t_n = nr\) for \(i = 0, 1, 2, \ldots, M\) and \(n = 0, 1, 2, \ldots, N\).

On the basis of the shifted Grünwald formula as used in [19], we can get a difference system:

\[
\begin{aligned}
\frac{c_{i+1}^{n+1} - c_i^n}{r} &= -v \frac{c_{i+1}^{n+1} - c_{i-1}^{n+1}}{2h} + \left(1 + \frac{\Gamma(-\alpha)}{\Gamma(k-\alpha)} \right) \frac{1}{h^\alpha} \sum_{k=0}^{i+1} \Gamma(k-\alpha) c_{i-k+1}^{n+1} + f_i^{n+1} \lambda_j^{n+1}, \\
c_0^n &= 0, \quad c_M^n = 0, \quad c_i^0 = g(x_i).
\end{aligned}
\]

(3)

Define \(g_m = \frac{\Gamma(m-\alpha)}{\Gamma(-\alpha)} (\pi/m)^{-\alpha} \), \(p = \frac{vr}{\pi}\) and \(q = \frac{vn}{\pi}\); then this can be rearranged to yield

\[
-q g_0 c_{i+1}^{n+1} + (1 + p - q g_1) c_{i+1}^{n+1} - (p + q g_2) c_{i-1}^{n+1} - \sum_{m=3}^{i+1} g_m c_{i-m+1}^{n+1} = c_i^n + \gamma f_i^{n+1} \lambda_j^{n+1}.
\]

(4)

Let \(c_{i+1}^{n+1} = [c_1^{n+1}, c_2^{n+1}, \ldots, c_M^{n+1}]^T\), and \(A = (a_{ij})\) be a matrix of coefficients where the coefficients \(a_{ij}\) for \(i = 1, 2, \ldots, M - 2\) and \(j = 1, 2, \ldots, M - 1\) are defined as follows:

\[
a_{ij} = \begin{cases} 0, & \text{if } j > i + 1, \\ -q g_0, & \text{if } j = i + 1, \\ 1 + p - q g_1, & \text{if } j = i, \\ -p - q g_2, & \text{if } j = i - 1, \\ -q g_{i-j+1}, & \text{if } j < i - 1, 
\end{cases}
\]

(5)
and $a_{M-1,j} = -q_{M-j}$ for $i = M - 1, j = 1, 2, \ldots, M - 3$, and $a_{M-1,M-2} = -p - q_{2}, a_{M-1,M-1} = 1 + p - q_{1}$. Then Eq. (4) is transformed to an implicit difference scheme given as follows:

$$
Ac^{n+1} = c^{n} + rf^{n+1}c^{n+1}.
$$

(6)

As for the above scheme, we can get the following conclusion as proved in paper [20] by which numerical solution of the forward problem (1) can be obtained stably.

**Proposition 1** ([20]). The implicit difference scheme (6) for problem (1) with $1 < \alpha \leq 2$ is consistent and unconditionally stable.

In the following, numerical simulations for determining the source function $f = f(x)$ will be carried out based on inverse problems (1)–(2) and the above difference scheme. Therefore, a suitable inversion algorithm is needed.

### 3. The Inversion Algorithm

As we know, solving an inverse problem can be reduced to minimizing an error functional of the unknown function between the computational output data and the measured data. So, some optimal methods, such as least square methods and Gauss–Newton methods, are often utilized to solve inverse problems. However, considering data errors, model errors, and computer errors, regularization strategies are also necessary to cut down these noises and errors and get a useful solution. It is noticeable that an optimal perturbation regularization algorithm has been attested to be effective for identifying computer errors, regularization strategies are also necessary to cut down these noises and errors and get a useful solution.

Gauss–Newton methods, are often utilized to solve inverse problems. However, considering data errors, model errors, and computer errors, regularization strategies are also necessary to cut down these noises and errors and get a useful solution.

Suppose that $(\varphi_{k}(x))_{k=1}^{\infty}$ is a group of basis functions for the space of unknowns; then we have

$$
f(x) = \sum_{k=1}^{\infty} a_{k}\varphi_{k}(x),
$$

(7)

and, by approximation, we have

$$
f(x) = \sum_{k=1}^{K} a_{k}\varphi_{k}(x),
$$

(8)

where $a_{k}$ ($k = 1, 2, \ldots, K$) are expansion coefficients, and $K \geq 1$ is the truncated level of $f(x)$. Denote as $c(x, t; f)$ the unique solution of the forward problem (1) for any prescribed source function $f(x)$ given by (8); then a feasible way to solve the inverse problem here is to solve the following minimization problem:

$$
\min \|c(x, T; f) - \theta(x)\|_{2}^{2} + \mu \|f\|_{2}^{2},
$$

(9)

where $\mu$ is the regularization parameter, $c(x, T; f)$ is the computational output for any given $f = f(x)$, and $\theta(x)$ is the additional final observation given by (2).

Now, for any initial iteration $f_{s}$, suppose that

$$
f_{s+1} = f_{s} + \delta f_{s}, \quad s = 0, 1, 2, \ldots,
$$

(10)

where $\delta f_{s}$ is called the perturbation function. Thus in order to get $f_{s+1}$ from the given $f_{s}$, we only need to compute an optimal perturbation $\delta f_{s}$. In the follows, for convenience of writing, $f_{s}$ and $\delta f_{s}$ are abbreviated as $f$ and $\delta f$ respectively.

Thanks to (8), it is convenient to set

$$
\delta f(x) = \sum_{k=1}^{K} \delta a_{k}\varphi_{k}(x),
$$

(11)

and then we only need to determine the perturbation vector

$$
\delta a = (\delta a_{1}, \delta a_{2}, \ldots, \delta a_{K})^{T}.
$$

(12)

Taking Taylor’s expansion for $c(x, T; f + \delta f)$ at $f$, and ignoring higher order terms, we can get

$$
c(x, T; f + \delta f) \approx c(x, T; f) + \nabla^{T}c(x, T; f) \cdot \delta f.
$$

(13)

With the help of (13), we define an error functional for the perturbation as follows:

$$
F(\delta f) = \|\nabla^{T}c(x, T; f) \cdot \delta f - [\theta(x) - c(x, T; f)]\|_{2}^{2} + \mu \|f\|_{2}^{2}.
$$

(14)

Now, discretizing the domain $[0, \pi]$ with $0 = x_{1} < x_{2} < \cdots < x_{L} = \pi$, where $L$ denotes the number of grids, the above $L_{2}$ norm can then be reduced to the discrete Euclid norm.
Furthermore, define
\[ B = (b_{ik})_{i \times K}; \quad b_{ik} = \frac{c(x_i, T; f + \tau \varphi_k(x)) - c(x_i, T; f)}{\tau}; \quad i = 1, 2, \ldots, L, \]
\[ \xi = (c(x_1, T; f), c(x_2, T; f), \ldots, c(x_L, T; f))^T; \quad \eta = (\theta(x_1), \theta(x_2), \ldots, \theta(x_L))^T, \]
where \( \tau \) is called the numerical differentiation step.

Through the minimization of (14), we can get the following normal equation (see [26], for instance):
\[ (\mu I + B^T B)\delta a = B^T (\eta - \xi). \]

By the above discussions, we know that the best perturbation \( \delta a \) can be solved by using Eq. (16) via
\[ \delta a = (\mu I + B^T B)^{-1}B^T (\eta - \xi), \]
and then the optimal coefficient vector \( a \) can be approximated by iteration procedure (10) as long as the perturbation satisfies a given precision. In the following, several numerical simulations will be carried out to illustrate the above inversion algorithm for the inverse problems (1)–(2).

4. Numerical inversions

In the concrete computations, we will take polynomial space as the basis functions space. That is to say, we have \( \varphi_k(x) = x^{k-1}, k = \frac{1}{T}, K \), and the unknown space can be approximated as \( \phi = \text{span}\{x^{k-1}, k = \frac{1}{T}, K\} \). Nevertheless, we will choose \( K = 4 \); then the source function \( f(x) \) has an approximate expansion of
\[ f(x) = a_1 + a_2 x + a_3 x^2 + a_4 x^3. \]

In other words, where there is no confusion, we will write \( f = (a_1, a_2, a_3, a_4) \).

In addition, the hydraulic parameters are chosen as \( D = 0.25 \) and \( v = 0.01 \), the time function in the source is given as \( \lambda(t) = \exp(-t) \), and the initial function is given as \( g(x) = x(\pi - x) \); the final time is chosen as \( T = 1 \), the convergence criterion for the inversion algorithm is given as \( ||\delta a|| \leq 10^{-4} \), and the initial iteration is chosen as \( f_0 = 0 \). As for the difference scheme, we will take \( M = 100 \) and \( N = 20 \) in solving the forward problem.

It is noticeable that for \( M = 100 \), we can get the coefficient matrix \( A \) for the difference Eq. (6), whose condition number is \( \text{Cond}(A) = 34.064 \), and the maximum eigenvalue of the inverse of \( A \) is \( \lambda_{\text{max}}(A^{-1}) = 0.9882 \) which shows that the implicit difference scheme given in Section 2 is of numerical stability.

Example 1. Set the true source function as \( f(x) = 1 - x \); by solving the forward problem, we get the final data \( \theta(x) = c(x, 1; f) \) as additional data. Noting computational errors, we will add some noises to the additional data, i.e. the additional condition that we will utilize in the inversion algorithm is represented as
\[ \theta^\delta(x) = \theta(x) + \varepsilon \sigma, \]
where \( \varepsilon \) is the noise level, and \( \sigma \) is a random vector in the range \([-1, 1]\). All computations are performed using a Dell computer Studio 540, and each inversion result is an average value of ten computations.

In this example, let us investigate the impacts of fractional orders and the noise level in the case of taking the regularization parameter as \( \mu = 0.01 \) and the numerical differential step as \( \tau = 0.1 \). By applying the inversion algorithm, the source function can be reconstructed with noisy data with \( \varepsilon = 0.01 \) and \( \varepsilon = 0.05 \). The computational results are listed in Tables 1 and 2, where \( \alpha \) denotes the fractional order, \( f^* = (a^*_1, a^*_2, a^*_3, a^*_4) \) denotes the computational reconstruction solution, \( f = (a_1, a_2, a_3, a_4) \) is the true solution, \( I \) represents the average number of iterations, and the solution error is expressed as
\[ \text{Err} = \sqrt{\int_0^\pi (f(x) - f^*(x))^2 dx}. \]

Furthermore, in the case of \( \alpha = 1.9 \), the computational reconstruction solution and the true solution are plotted in Figs. 1 and 2, respectively.

From Table 1, we can see that the fractional order \( \alpha \) has little impact on the inversion results. From Figs. 1 and 2, it can be seen that the reconstructed source data are in good agreement with the true solution.

Example 2. Take \( f(x) = \sin x \) as a true source solution. In the space of \( \phi = \text{span}\{1, x, x^2, x^3\} \), the true source can be expressed as \( f = (0, 1, -1/2, 0) \). With a method similar to that used in Example 1, the source function can be reconstructed also by choosing zero initial iteration. In this example, we will investigate the impacts of fractional orders, regularization parameters and noisy data on the inversion algorithm. The computational results are listed in Tables 3–5. Furthermore, in the case of \( \alpha = 1.9, \mu = 0.01 \), and \( \tau = 0.1 \), the computational reconstruction source solution and the true solution with noisy data with \( \varepsilon = 0.01 \) and \( \varepsilon = 0.05 \) are plotted in Figs. 3 and 4, respectively.
Table 1
Impacts of fractional orders on the inversion results for $\varepsilon = 0.01$.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$f^*$</th>
<th>$f^*$</th>
<th>$\text{Err}$</th>
<th>$l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>(0.9928, -0.9829, -0.0114, 0.0022)</td>
<td>0.9928</td>
<td>-0.9829</td>
<td>-0.0114</td>
</tr>
<tr>
<td>1.5</td>
<td>(0.9944, -0.9879, -0.0075, 0.0014)</td>
<td>0.9944</td>
<td>-0.9879</td>
<td>-0.0075</td>
</tr>
<tr>
<td>1.7</td>
<td>(1.0162, -1.0362, 0.0230, -0.0044)</td>
<td>1.0162</td>
<td>-1.0362</td>
<td>0.0230</td>
</tr>
<tr>
<td>1.9</td>
<td>(1.0085, -1.0197, 0.0128, -0.0025)</td>
<td>1.0085</td>
<td>-1.0197</td>
<td>0.0128</td>
</tr>
</tbody>
</table>

Table 2
Impacts of noisy data on the inversion algorithm.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$f^*$</th>
<th>$f^*$</th>
<th>$\text{Err}$</th>
<th>$l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>(1.0085, -1.0197, 0.0128, -0.0025)</td>
<td>1.0085</td>
<td>-1.0197</td>
<td>0.0128</td>
</tr>
<tr>
<td>0.05</td>
<td>(1.0732, -1.1726, 0.1137, -0.0222)</td>
<td>1.0732</td>
<td>-1.1726</td>
<td>0.1137</td>
</tr>
</tbody>
</table>

Fig. 1. The reconstruction solution and true solution for $\varepsilon = 0.01$.

Fig. 2. The reconstruction solution and true solution for $\varepsilon = 0.05$.

Table 3
Impacts of fractional orders on the inversion results for $\varepsilon = 0.01$ in Example 2.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$f^*$</th>
<th>$\text{Err}$</th>
<th>$l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>(-0.0875, 1.3836, -0.4520, 0.0043)</td>
<td>0.0368</td>
<td>8</td>
</tr>
<tr>
<td>1.5</td>
<td>(-0.0967, 1.3969, -0.4561, 0.0044)</td>
<td>0.0401</td>
<td>10</td>
</tr>
<tr>
<td>1.7</td>
<td>(-0.0889, 1.3760, -0.4403, 0.0009)</td>
<td>0.0405</td>
<td>15</td>
</tr>
<tr>
<td>1.9</td>
<td>(-0.0748, 1.3410, -0.4137, -0.0052)</td>
<td>0.0417</td>
<td>23</td>
</tr>
</tbody>
</table>
Table 4
Impacts of regularization parameters on the inversion results for $\varepsilon = 0.01$ in Example 2.

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$f^*$</th>
<th>$\text{Err}$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>(-0.0825, 1.3565, -0.4245, -0.0028)</td>
<td>0.0418</td>
<td>137.1</td>
</tr>
<tr>
<td>0.01</td>
<td>(-0.0748, 1.3410, -0.4137, -0.0052)</td>
<td>0.0417</td>
<td>23</td>
</tr>
<tr>
<td>0.001</td>
<td>(-0.0899, 1.3741, -0.4561, -0.0005)</td>
<td>0.0425</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5
Impacts of noisy data on the inversion algorithm in Example 2.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$f^*$</th>
<th>$\text{Err}$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>(-0.0748, 1.3410, -0.4137, -0.0052)</td>
<td>0.0417</td>
<td>23</td>
</tr>
<tr>
<td>0.05</td>
<td>(-0.0234, 1.2157, -0.3312, -0.0211)</td>
<td>0.0461</td>
<td>22.8</td>
</tr>
</tbody>
</table>

Fig. 3. The reconstruction solution and true solution for $\varepsilon = 0.01$ in Example 2.

Fig. 4. The reconstruction solution and true solution for $\varepsilon = 0.05$ in Example 2.

From Tables 3 and 4, we can observe that the fractional order $\alpha$ and regularization parameter $\mu$ have little impact on the inversion results. However, regularization parameters have some influence on the number of iteration times. From Figs. 3 and 4, it can be seen that the inversion solutions basically coincide with the true solution except for the endpoints.

**Example 3.** In this example, we will take $f(x) = \exp(-x)$ as a true solution. In the polynomial space $\phi = \text{span}\{1, x, x^2, x^3\}$, the true source magnitude can be expressed as $f = (1, -1, 1/2, 1/6)$. Also choosing the initial iteration as $f_0 = (0, 0, 0, 0)$, we get the computational reconstruction results which are listed in Table 6. Furthermore, in the case of $\alpha = 1.9$, $\mu = 0.01$
Table 6

<table>
<thead>
<tr>
<th>( \varepsilon )</th>
<th>( f^* )</th>
<th>( \text{Err} )</th>
<th>( I )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>(0.9755, -0.8740, 0.3070, -0.0399)</td>
<td>0.0113</td>
<td>22</td>
</tr>
<tr>
<td>0.05</td>
<td>(1.0227, -0.9863, 0.3815, -0.0546)</td>
<td>0.0226</td>
<td>22.4</td>
</tr>
</tbody>
</table>

Fig. 5. The reconstruction solution and true solution for \( \varepsilon = 0.01 \) in Example 3.

Fig. 6. The reconstruction solution and true solution for \( \varepsilon = 0.05 \) in Example 3.

and \( \tau = 0.1 \), the reconstruction source solution and the true solution with noise levels of \( \varepsilon = 0.01 \) and \( \varepsilon = 0.05 \) are plotted in Figs. 5 and 6, respectively.

From Table 6, we can see that the inversion solutions with noisy data basically agree with the true solution; however, there still remain some deviations at the endpoints, as seen in Figs. 5 and 6. Furthermore, in the case of taking the noise level \( \varepsilon = 0.01 \), ten computations for different fractional orders and the corresponding average results with \( \tau = 0.1 \) and \( \mu = 0.01 \) are listed in Table 7.

From Table 7, it can be seen that the inversion algorithm is also stable, and the solution errors undergo little change when the fractional order takes different values.

We give two concluding remarks to end the paper.

(1) In this paper, an inverse problem of determining a linear source magnitude function in a FADE using final observations is investigated by applying an optimal perturbation regularization algorithm with noisy data, and the numerical inversion results seem to be satisfactory. If the source function takes a polynomial form, the inversion reconstruction data are in good agreement with the true solution, and they basically coincide with the true solution in the case of the source taking trigonometric or index function forms. There are some computational deviations occurring at the endpoints of the domain.
Table 7

<table>
<thead>
<tr>
<th>Times</th>
<th>$\alpha = 1.3$</th>
<th>$\alpha = 1.5$</th>
<th>$\alpha = 1.7$</th>
<th>$\alpha = 1.9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0085</td>
<td>0.0089</td>
<td>0.0125</td>
<td>0.0107</td>
</tr>
<tr>
<td>2</td>
<td>0.0154</td>
<td>0.0097</td>
<td>0.0094</td>
<td>0.0108</td>
</tr>
<tr>
<td>3</td>
<td>0.0099</td>
<td>0.0088</td>
<td>0.0098</td>
<td>0.0159</td>
</tr>
<tr>
<td>4</td>
<td>0.0104</td>
<td>0.0121</td>
<td>0.0143</td>
<td>0.0105</td>
</tr>
<tr>
<td>5</td>
<td>0.0089</td>
<td>0.0135</td>
<td>0.0097</td>
<td>0.0120</td>
</tr>
<tr>
<td>6</td>
<td>0.0126</td>
<td>0.0140</td>
<td>0.0097</td>
<td>0.0102</td>
</tr>
<tr>
<td>7</td>
<td>0.0102</td>
<td>0.0107</td>
<td>0.0117</td>
<td>0.0110</td>
</tr>
<tr>
<td>8</td>
<td>0.0130</td>
<td>0.0092</td>
<td>0.0125</td>
<td>0.0103</td>
</tr>
<tr>
<td>9</td>
<td>0.0085</td>
<td>0.0111</td>
<td>0.0096</td>
<td>0.0103</td>
</tr>
<tr>
<td>10</td>
<td>0.0120</td>
<td>0.0093</td>
<td>0.0119</td>
<td>0.0115</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0109</td>
<td>0.0107</td>
<td>0.0111</td>
<td>0.0113</td>
</tr>
</tbody>
</table>

(2) From the inversion computations, we find that the FADE ($1 < \alpha < 2$) has some different properties as compared with the ADE ($\alpha = 2$). The numerical inversion results are more accurate and stable in the case of $\alpha \in (1, 2)$ than for $\alpha = 2$ with noisy data. We also find that regularization parameters should be chosen carefully in performing the inversion algorithm. The inversion accuracy could be worse and the number of iterations become larger if the regularization parameters are chosen too large, and the inversion algorithm may fail if the regularization parameter is too small.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 11071148, No. 10926194, No. 10471080), and the Natural Science Foundation of Shandong Province (No. Y2007A29).

References