

Research Article

A Numerical Method for Delayed Fractional-Order Differential Equations

Zhen Wang^{1,2}

¹ College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

² State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Correspondence should be addressed to Zhen Wang; wangzhen.sd@gmail.com

Received 3 February 2013; Accepted 14 April 2013

Academic Editor: Theodore E. Simos

Copyright © 2013 Zhen Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A numerical method for nonlinear fractional-order differential equations with constant or time-varying delay is devised. The order here is an arbitrary positive real number, and the differential operator is with the Caputo definition. The general Adams-Bashforth-Moulton method combined with the linear interpolation method is employed to approximate the delayed fractional-order differential equations. Meanwhile, the detailed error analysis for this algorithm is given. In order to compare with the exact analytical solution, a numerical example is provided to illustrate the effectiveness of the proposed method.

1. Introduction

Fractional calculus is an old mathematical problem and has been always thought of as a pure mathematical problem for nearly three centuries [1–3]. Though having a long history, it was not applied to physics and engineering for a long period of time. However, in the last few decades, fractional calculus began to attract increasing attention of scientists from an application point of view [4–6]. In the fields of continuous-time modeling, many scholars have pointed out that fractional derivatives are very helpful in describing linear viscoelasticity, acoustics, rheology, polymeric chemistry, and so forth. Moreover, fractional derivatives have proved to be a very suitable tool for the description of memory and hereditary properties of various materials and processes. Nowadays the mathematical theories and practical applications of these operators are well established, and their applicabilities to science and engineering are being considered as attractive topics. In modeling dynamical processes, equations described with fractional derivatives have appeared in various fields such as physics, chemistry, and engineering [7, 8], and therefore, fractional-order differential equations (FDEs) are becoming the focus of many studies.

The development of effective methods for numerically solving FDEs has received increasing attention over the last

few years. Several methods based on Caputo or Riemann-Liouville definition have been proposed and analyzed [9–24]. For instance, based on the predictor-corrector scheme, Diethelm et al. introduced Adams-Bashforth-Moulton algorithm [11–13], and meanwhile some error analysis and an extension of Richardson extrapolation were presented to improve the numerical accuracy. These techniques elaborated in [11–13] take advantage of the fact that the FDEs can be reduced to Volterra type integral equations. And therefore, one can apply the numerical schemes for Volterra type integral equations to find the solution of FDEs; one can refer to [21, 25–28] and the literature cited therein for more details. The technique presented in [11–13] has been further analyzed and extended for multiterm [15] and multiorder systems [17]. Li and Tao [20] studied the error analyses of the fractional Adams method for the fractional-order ordinary differential equations for more general cases. Deng [24] obtained a good numerical approximation by combining the short memory principle with the predictor-corrector approach.

However, in real world systems, delay is very often encountered in many practical systems, such as automatic control, biology and hydraulic networks, economics, and long transmission lines. Delayed differential equations are correspondingly used to describe such dynamical systems. In recent years, delayed FDEs begin to arouse the attention

TABLE 1: Error behavior at time $t = T$ with analytical value with $\alpha = 0.2, \tau = 0.1$.

| T | E | $h = 1/10$ | $h = 1/20$ | $h = 1/40$ | $h = 1/80$ | $h = 1/160$ | $h = 1/320$ |
|-----|-------|------------|------------|--------------|---------------|---------------|---------------|
| 5 | E_A | 0.8037 | 0.2881 | 0.1074 | 0.0411 | 0.0160 | 0.0062 |
| | E_R | 0.0402 | 0.0144 | 0.0054 | 0.0021 | $7.980e - 4$ | $3.090e - 4$ |
| 10 | E_A | 1.659 | 0.5964 | 0.2234 | 0.0861 | 0.0338 | 0.0134 |
| | E_R | 0.0184 | 0.0066 | 0.0025 | $9.562e - 4$ | $3.757e - 4$ | $1.494e - 5$ |
| 50 | E_A | 8.161 | 2.953 | 1.114 | 0.4322 | 0.1714 | 0.0690 |
| | E_R | 0.0033 | 0.0012 | $4.545e - 4$ | $1.7643e - 4$ | $6.9940e - 5$ | $2.8171e - 5$ |

of many researchers [29–31]. To simulate these equations is an important technique in the research, accordingly, finding effective numerical methods for the delayed FDEs is a necessary process. However, to the best of our knowledge, there are very few works devoted in the literature to such work so far. The aim of this paper is to generalize the Adams-Bashforth-Moulton algorithm introduced in [11–13] to the delayed FDEs.

The rest of this paper is organized as follows. In Section 2, the numerical scheme is devised based on the Adams-Bashforth-Moulton method, and meanwhile the predictor-corrector scheme is developed in the case of constant and time-varying delay, respectively. The detailed error analysis of the newly proposed algorithm is given in Section 3. Numerical simulations are performed in Section 4 in order to illustrate the effectiveness of the proposed scheme. Finally, some concluding remarks are reported in Section 5.

2. Formulation of the Numerical Method for Delayed FDEs

There are three commonly used definitions for fractional derivatives [2], that is, Riemann-Liouville, Grünwald-Letnikov, and Caputo definitions. In pure mathematics, Riemann-Liouville derivative is more commonly used than Caputo derivative. However, as we all know, only the Caputo definition has the same form as integer-order differential equations in the initial conditions. And therefore this paper is based on Caputo definition.

2.1. Caputo Fractional Derivatives

Definition 1 (see [2]). The Caputo fractional derivative of order α of a continuous function $f : R^+ \rightarrow R$ is defined as follows:

$$D_t^\alpha y(t) = \begin{cases} \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{f^{(m)}(\tau)}{(t-\tau)^{\alpha-m+1}} d\tau, & m-1 < \alpha < m, \\ \frac{d^m}{dt^m} f(t), & \alpha = m, \end{cases} \quad (1)$$

where Γ is Γ -function, and

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, \quad \Gamma(z+1) = z\Gamma(z). \quad (2)$$

2.2. The Procedure of Adams-Bashforth-Moulton Algorithm. The main aim of this paper is to study a numerical scheme for the approximate solution of delayed FDEs. For this purpose, we consider delayed FDEs described by

$$D_t^\alpha y(t) = f(t, y(t), y(t-\tau)), \quad t \geq 0, \quad m-1 < \alpha \leq m, \quad (3)$$

$$y(t) = \varphi(t), \quad t \leq 0, \quad (4)$$

where α is the order of the differential equation, $\varphi(t)$ is the initial value, and m is an integer.

Before introducing the new algorithm for the delayed FDEs, we first briefly recall the idea of the classical one-step Adams-Bashforth-Moulton algorithm for the following ODE:

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(0) = y_0. \quad (5)$$

Let $t \in [0, T]$, $t_j = jh$, $j = 0, 1, \dots, N$, and $h = T/N$ be the time step. Assume that $y_j = y(t_j)$, $j = 1, \dots, n$, then the previous equation is equivalent to

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s)) ds. \quad (6)$$

Since the integral in the right-hand side of (6) can be approximated by trapezoidal quadrature formula,

$$\int_{t_n}^{t_{n+1}} f(s, y(s)) ds \approx \frac{h}{2} (f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))). \quad (7)$$

Thus, we can get the approximation to y_{n+1} as follows:

$$y_{n+1} = y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1})], \quad (8)$$

which is an implicit equation and cannot be solved for y_{n+1} directly; however, one can adopt another numerical method to approximate y_{n+1} in the right-hand side preliminarily.

The preliminary approximation y_{n+1}^p is called predictor and can be obtained by forward Euler method or any other explicit method; here we take forward Euler method as an example; that is,

$$y_{n+1}^p = y_n + hf(t_n, y_n). \quad (9)$$

Then, the so-called one-step Adams-Bashforth-Moulton method is

$$y_{n+1} = y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1}^p)], \quad (10)$$

TABLE 2: Error behavior at time $t = T$ with analytical value with $\alpha = 0.9, \tau = 0.1$.

| T | E | $h = 1/10$ | $h = 1/20$ | $h = 1/40$ | $h = 1/80$ | $h = 1/160$ | $h = 1/320$ |
|-----|-------|---------------|---------------|---------------|----------------|----------------|----------------|
| 5 | E_A | 0.0534 | 0.0163 | 0.0038 | $-7.5742e - 4$ | -0.0026 | -0.0033 |
| | E_R | 0.0027 | $8.1293e - 4$ | $1.9072e - 4$ | $-3.7871e - 5$ | $-1.2853e - 4$ | $-1.6663e - 4$ |
| 10 | E_A | 0.0865 | 0.0247 | 0.0061 | $-5.1223e - 5$ | -0.0022 | -0.0031 |
| | E_R | $9.6095e - 4$ | $2.7404e - 4$ | $6.7297e - 5$ | $-5.6910e - 7$ | $-2.4824e - 5$ | $-3.4191e - 5$ |
| 50 | E_A | 0.3610 | 0.0955 | 0.0248 | 0.0051 | $-5.3786e - 4$ | -0.0023 |
| | E_R | $1.4736e - 4$ | $3.8962e - 5$ | $1.0103e - 5$ | $2.0997e - 6$ | $-7.1954e - 7$ | $-9.2911e - 7$ |

TABLE 3: Error behavior at time $t = T$ with analytical value with $\alpha = 0.2, \tau = 0.01 \exp(-t)$.

| T | E | $h = 1/10$ | $h = 1/20$ | $h = 1/40$ | $h = 1/80$ | $h = 1/160$ | $h = 1/320$ |
|-----|-------|---------------|---------------|---------------|---------------|---------------|---------------|
| 5 | E_A | 0.0425 | 0.0218 | 0.0133 | 0.0097 | 0.0081 | 0.0074 |
| | E_R | 0.0021 | 0.0011 | $6.6680e - 4$ | $4.8592e - 4$ | $4.0585e - 4$ | $3.6960e - 4$ |
| 10 | E_A | 0.0411 | 0.0217 | 0.0138 | 0.0104 | 0.0089 | 0.0082 |
| | E_R | $4.5679e - 4$ | $2.4130e - 4$ | $1.5319e - 4$ | $1.1553e - 4$ | $9.8859e - 5$ | $9.1315e - 5$ |
| 50 | E_A | -0.0351 | 0.0184 | 0.0116 | 0.0087 | 0.0074 | 0.0052 |
| | E_R | $1.4317e - 5$ | $7.5036e - 6$ | $4.7294e - 6$ | $3.5467e - 6$ | $3.0242e - 6$ | $2.4322e - 6$ |

in which (9) is called the predictor, and (10) is called the corrector. It is well known that this method is convergent; that is,

$$\max_{j=0,1,\dots,N} |y(t_j) - y_j| = O(h^2). \quad (11)$$

2.3. The Representation of Algorithm for Delayed FDEs. Having introduced the Adams-Bashforth-Moulton algorithm, now the key problem is to establish approximation to the delayed term $y(t - \tau)$, which consists of the following two cases.

Case I (when τ is constant). It can be seen that, for any positive constant τ , $t_j - \tau$ may not be a grid point t_n for any n . Suppose that $(m - \delta)h = \tau$ and $0 \leq \delta < 1$. When $\delta = 0$, $y(t_n - \tau)$ can be approximated by

$$y(t_n - \tau) \approx \begin{cases} y_{n-m}, & \text{if } n > m, \\ \varphi_n, & \text{if } n \leq m, \end{cases} \quad (12)$$

and when $0 < \delta < 1$, $y(t_n - \tau)$ cannot be calculated directly.

For $(m - 1)h < \tau < mh$, let v_{n+1} be the approximation to $y(t_{n+1} - \tau)$; we interpolate it by the two nearest points; that is,

$$v_{n+1} = \delta y_{n-m+2} + (1 - \delta) y_{n-m+1}. \quad (13)$$

It can be seen from (13) that if $m > 1$, the numerical equation is explicit and thus can be computed directly. However, when $m = 1$ and $\delta \neq 0$, that is, $\tau < h$, then the first term in the right-hand side of the previous equation is δy_{n+1} . It still needs to predict, in this case, that is

$$v_{n+1} = \delta y_{n+1}^p + (1 - \delta) y_n. \quad (14)$$

Case II (when τ is time varying). If τ is time varying, that is, $\tau = \tau(t)$, then the approximation in this case is more complicated. Let v_{n+1} be the approximation to $y(t_{n+1} - \tau)$,

and the linear interpolation of $\{y_j\}$ at point $t = t_{n+1} - \tau(t_{n+1})$ is employed to approximate the delay term. Let $\tau(t_{n+1}) = (m_{n+1} - \delta_{n+1})h$, where m_{n+1} is a positive integer and $\delta_{n+1} \in [0, 1]$, then

$$v_{n+1} = \delta_{n+1} y_{n-m_{n+1}+2} + (1 - \delta_{n+1}) y_{n-m_{n+1}+1}. \quad (15)$$

It should be noted that when τ is constant, for given h and τ , we can judge whether $m = 1$ or $m > 1$ holds at the beginning of the programme. However, when τ is time varying, m is also time varying; that is, at one moment it is equal to 1, and at another moment it may greater than 1. If $m_{n+1} = 1$, the first term in the right-hand side of (15) still needs to predict; otherwise, when $m_{n+1} > 1$, it needs not. So in each step of the computing, it needs to test whether $m_n = 1$ or not firstly, then computation of the first term in the right-hand side of (15) depends on whether it requires predicting or not in different steps.

Now we begin the numerical algorithm for the delayed FDEs: (3)-(4).

From the theory introduced in [10-14], we know that the delayed initial value problem (3)-(4) is equivalent to Volterra integral equation as follows:

$$y(t) = \sum_{k=0}^{m-1} \varphi(t) \frac{t^k}{k!} + \frac{1}{\Gamma(\alpha)} \int_0^t \frac{f(s, y(s), y(s - \tau))}{(t - s)^{1-\alpha}} ds. \quad (16)$$

Just as the method used in [13], the remaining problem is to compute the integral term in (16). Firstly the product trapezoidal quadrature formula is applied to replace the integrals term, where nodes t_j ($j = 0, 1, \dots, n + 1$) are taken with respect to the weight function $(t_{n+1} - \cdot)^{\alpha-1}$. That is, we employ the approximation

$$\int_0^{t_{n+1}} (t_{k+1} - z)^{\alpha-1} g(z) dz \approx \int_0^{t_{n+1}} (t_{n+1} - z)^{\alpha-1} \tilde{g}_{n+1}(z) dz, \quad (17)$$

TABLE 4: Error behavior at time $t = T$ with analytical value with $\alpha = 0.9$, $\tau = 0.01 \exp(-t)$.

| T | E | $h = 1/10$ | $h = 1/20$ | $h = 1/40$ | $h = 1/80$ | $h = 1/160$ | $h = 1/320$ |
|-----|-------|-------------|-------------|-------------|-------------|-------------|-------------|
| 5 | E_A | 0.0031 | 0.0016 | 0.0012 | 0.0011 | 0.0011 | 0.0010 |
| | E_R | $1.5375e-4$ | $8.1279e-5$ | $6.1399e-5$ | $5.5480e-5$ | $5.3282e-5$ | $5.2136e-5$ |
| 10 | E_A | 0.0022 | $9.1466e-4$ | $5.8474e-4$ | $4.9919e-4$ | $4.7680e-4$ | $4.7115e-4$ |
| | E_R | $2.4189e-5$ | $1.0163e-5$ | $6.4971e-6$ | $5.5465e-6$ | $5.2300e-6$ | $5.2350e-6$ |
| 50 | E_A | 0.0016 | $4.9082e-4$ | $1.9954e-4$ | $1.2510e-4$ | $1.0616e-4$ | $7.0303e-4$ |
| | E_R | $6.6311e-7$ | $2.0033e-7$ | $8.1447e-8$ | $5.1062e-8$ | $4.3330e-8$ | $2.7081e-8$ |

where $\tilde{g}_{n+1}(\cdot)$ is the piecewise linear interpolation for $g(\cdot)$ with nodes and knots chosen at t_j , $j = 0, 1, \dots, n+1$. Using the standard technique of quadrature theory, it can be found that we can write the integrals on the right-hand side of (17) as

$$\int_0^{t_{n+1}} (t_{n+1} - z)^{\alpha-1} \tilde{g}_{n+1}(z) dz = \sum_{j=0}^{n+1} a_{j,n+1} g(t_j), \quad (18)$$

where

$$\begin{aligned} a_{j,n+1} &= \frac{h^\alpha}{\alpha(\alpha+1)} \\ &\times \begin{cases} n^{\alpha+1} - (n-\alpha)(n+1)^\alpha, & j=0, \\ (n-j+2)^{\alpha+1} + (n-j)^{\alpha+1} - 2(n-j+1)^{\alpha+1}, & 1 \leq j \leq n, \\ 1, & j=n+1. \end{cases} \end{aligned} \quad (19)$$

Consequently, the numerical scheme for FDEs (3)-(4) can be depicted as:

$$\begin{aligned} y_{n+1} &= \sum_{k=0}^{m-1} \varphi_k \frac{t_{n+1}^k}{k!} + \frac{h^\alpha}{\Gamma(\alpha+2)} f(t_{n+1}, y_{n+1}^p, v_{n+1}) \\ &+ \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n a_{j,n+1} f(t_j, y_j, v_j), \\ v_{n+1} &= \begin{cases} \delta y_{n-m+2} + (1-\delta) y_{n-m+1}, & \text{if } m > 1, \\ \delta y_{n+1}^p + (1-\delta) y_n, & \text{if } m = 1, \end{cases} \\ y_{n+1}^p &= \sum_{k=0}^{m-1} \varphi_k \frac{t_{n+1}^k}{k!} + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n b_{j,n+1} f(t_j, y_j, v_j), \end{aligned} \quad (20)$$

where

$$b_{j,n+1} = \frac{h^\alpha}{\alpha} ((n-j+1)^\alpha - (n-j)^\alpha). \quad (21)$$

3. Error Analysis of the Newly-Proposed Algorithm

Before giving the main results, we firstly introduce the following two lemmas, which will be used in the proof of the theorem.

Lemma 2 (see [13]). (i) Let $z \in C^1[0, T]$, then

$$\begin{aligned} &\left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} z(t) dt - \sum_{j=0}^n b_{j,n+1} z(t_j) \right| \\ &\leq \frac{1}{\alpha} \|z'\|_\infty t_{n+1}^\alpha h. \end{aligned} \quad (22)$$

(ii) Let $z(t) \in C^2[0, T]$, then

$$\begin{aligned} &\left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} z(t) dt - \sum_{j=0}^{n+1} a_{j,n+1} z(t_j) \right| \\ &\leq C_\alpha^{Tr} \|z''\|_\infty t_{n+1}^\alpha h^2. \end{aligned} \quad (23)$$

In this paper, we further assume that $f(\cdot)$ in (3) satisfies the following Lipschitz conditions with respect to its variables as follows:

$$\begin{aligned} |f(t, y_1, u) - f(t, y_2, u)| &\leq L_1 |y_1 - y_2|, \\ |f(t, y, u_1) - f(t, y, u_2)| &\leq L_2 |u_1 - u_2|, \end{aligned} \quad (24)$$

where L_1, L_2 are positive constants.

Theorem 3. Suppose the solution $y(t) \in C^2[0, T]$ of the initial value problem satisfies the following two conditions:

$$\begin{aligned} &\left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} D_t^\alpha y(t) dt - \sum_{j=0}^n b_{j,n+1} D_t^\alpha y(t_j) \right| \\ &\leq C t_{n+1}^{\gamma_1} h^{\delta_1}, \\ &\left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} D_t^\alpha y(t) dt - \sum_{j=0}^n a_{j,n+1} D_t^\alpha y(t_j) \right| \\ &\leq C t_{n+1}^{\gamma_2} h^{\delta_2}, \end{aligned} \quad (25)$$

with some $\gamma_1, \gamma_2 \geq 0$, and $\delta_1, \delta_2 > 0$; then for some suitable $T > 0$, we have

$$\max_{0 \leq j \leq N} |y(t_j) - y_j| \leq Ch^q, \quad (26)$$

where $q = \min\{\delta_1 + \alpha, \delta_2\}$, $N = [T/h]$, and C is a positive constant.

Proof. We will prove the result by virtue of mathematical induction. Suppose that the conclusion is true for $j = 0, 1, \dots, n$, then we have to prove that the inequality also holds for $j = n + 1$. To do this, we first consider the error of the predictor y_{n+1}^p .

From the assumptions (24), we have

$$\begin{aligned} & |f(t_j, y(t_j), y(t_j - \tau)) - f(t_j, y_j, v_j)| \\ & \leq |f(t_j, y(t_j), y(t_j - \tau)) - f(t_j, y_j, y(t_j - \tau))| \\ & \quad + |f(t_j, y_j, y(t_j - \tau)) - f(t_j, y_j, v_j)| \\ & \leq L_1 h^q + L_2 h^q = (L_1 + L_2) h^q. \end{aligned} \quad (27)$$

So

$$\begin{aligned} & |y(t_{n+1}) - y_{n+1}^p| \\ & = \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} f(t, y(t), y(t - \tau)) dt \right. \\ & \quad \left. - \sum_{j=0}^n b_{j,n+1} f(t_j, y_j, v_j) \right| \\ & \leq \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} D_t^\alpha y(t) dt - \sum_{j=0}^n b_{j,n+1} D_t^\alpha y(t_j) \right| \\ & \quad + \frac{1}{\Gamma(\alpha)} \\ & \quad \times \sum_{j=0}^n b_{j,n+1} \{ |f(t_j, y(t_j), y(t_j - \tau)) \\ & \quad - f(t_j, y_j, y(t_j - \tau))| \\ & \quad + |f(t_j, y_j, y(t_j - \tau)) - f(t_j, y_j, v_j)| \} \\ & \leq \frac{Ct_{n+1}^{\gamma_1}}{\Gamma(\alpha)} h^{\delta_1} + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n b_{j,n+1} (L_1 + L_2) Ch^q, \end{aligned} \quad (28)$$

and from

$$\sum_{j=0}^n b_{j,n+1} = \sum_{j=0}^n \int_{jh}^{(j+1)h} (t_{n+1} - t)^{\alpha-1} dt = \frac{1}{\alpha} t_{n+1}^\alpha \leq \frac{1}{\alpha} T^\alpha, \quad (29)$$

we have

$$|y(t_{n+1}) - y_{n+1}^p| \leq \frac{CT^{\gamma_1}}{\Gamma(\alpha)} h^{\delta_1} + \frac{CT^\alpha}{\Gamma(\alpha+1)} h^q. \quad (30)$$

Since

$$\begin{aligned} \sum_{j=0}^n a_{j,n+1} & \leq \sum_{j=0}^n \frac{h^\alpha}{\alpha(\alpha+1)} \\ & \quad \times \{ [(k-j+2)^{\alpha+1} - (k-j+1)^{\alpha+1}] \\ & \quad - [(k-j+1)^{\alpha+1} - (k-j)^{\alpha+1}] \} \\ & = \left[\int_0^{t_{n+1}} (t_{n+2} - t)^\alpha dt - \int_0^{t_{n+1}} (t_{n+1} - t)^\alpha dt \right] \\ & = \frac{1}{\alpha} \int_0^{t_{n+1}} [(t_{n+1} - t)^\alpha]'(t) dt = \frac{1}{\alpha} t_{n+1}^\alpha \leq T^\alpha, \end{aligned} \quad (31)$$

we have

$$\begin{aligned} & |y(t_{n+1}) - y_{n+1}| \\ & = \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} f(t, y(t), y(t - \tau)) \right. \\ & \quad \left. y(t - \tau) dt - \sum_{j=0}^n a_{j,n+1} f(t_j, y_j, v_j) \right. \\ & \quad \left. - a_{n+1,n+1} f(t_{n+1}, y_{n+1}^p, v_{n+1}) \right| \\ & \leq \frac{1}{\Gamma(\alpha)} \left\{ \left| \int_0^{t_{n+1}} (t_{n+1} - t)^{\alpha-1} f(t, y(t), y(t - \tau)) \right. \right. \\ & \quad \left. - \sum_{j=0}^{n+1} a_{j,n+1} f(t_j, y(t_j), y(t_j - \tau)) \right| \\ & \quad + \sum_{j=0}^n a_{j,n+1} |f(t_j, y(t_j), y(t_j - \tau)) - f(t_j, y_j, v_j)| \\ & \quad + a_{n+1,n+1} |f(t_{n+1}, y(t_{n+1}), y(t_{n+1} - \tau)) \\ & \quad \quad \quad - f(t_{n+1}, y_{n+1}^p, v_{n+1})| \} \\ & \leq \frac{1}{\Gamma(\alpha)} \left[Ct_{n+1}^{\gamma_2} h^{\delta_2} + Ch^q \sum_{j=0}^n a_{j,n+1} \right. \\ & \quad \left. + a_{n+1,n+1} L \left(h^q + \frac{CT^{\gamma_1}}{\Gamma(\alpha)} h^{\delta_1} + \frac{CT^\alpha}{\Gamma(\alpha+1)} h^q \right) \right] \\ & \leq \frac{1}{\Gamma(\alpha)} \left[Ct_{n+1}^{\gamma_2} h^{\delta_2} + \frac{Lh^{q+h}}{\alpha(\alpha+1)} \right. \\ & \quad \left. + \frac{CLT^{\gamma_1}}{\Gamma(\alpha+2)} + \frac{CT^\alpha h^{\alpha+q}}{\alpha\Gamma(\alpha+2)} \right] = Ch^q, \end{aligned} \quad (32)$$

which completes the proof. \square

When $\delta_1 = 1$, $\delta_2 = 2$, applying Theorem 3, it is not difficult to derive the following result on the error bound.

Theorem 4. Suppose that $\alpha > 0$ and $f(t, y(t), z(t)) \in C'[0, T]$, then

$$\max_{0 \leq j \leq N} |y(t_j) - y_j| \leq Ch^q, \quad (33)$$

where $q = \min \{1 + \alpha, 2\}$, and C is a positive constant.

Remark 5. We can see from Cases I and II that if the solution $y(t)$ is not smooth, then the truncation error of y_{n+1} , which approximates to $y(t - \tau)$, will not be $O(h^2)$. Therefore, by the error analysis elaborated in Section 3, we cannot yield the same conclusion as in Theorem 4. We will consider such a problem in our future work.

4. A Numerical Example

In this section, the following delayed FDE is considered:

$$\begin{aligned} D_t^\alpha y(t) &= \frac{2}{\Gamma(3-\alpha)} t^{2-\alpha} - \frac{1}{\Gamma(2-\alpha)} t^{1-\alpha} + 2\tau t - \tau^2 \\ &\quad - \tau - y(t) + y(t - \tau), \quad \alpha \in (0, 1), \\ y(t) &= 0, \quad t \leq 0. \end{aligned} \quad (34)$$

Notice that the exact solution to this equation is

$$y(t) = t^2 - t. \quad (35)$$

In accordance with delay τ being constant or time varying, the numerical results are displayed in Tables 1, 2, 3, and 4. respectively, where E_A denotes the absolute numerical error and E_R denotes the relative numerical error. From the numerical results we can see that the computing errors are in general acceptable for engineering.

5. Conclusions

In this paper, a numerical algorithm is formulated based on Adams-Bashforth-Moulton method for fractional-order differential equations with time delay. The error analysis of the numerical scheme is carried out, meanwhile, a numerical example with constant delay and time-varying delay is proposed to testify the effectiveness of the scheme. This algorithm can be used not only in the simulation of delayed fractional-order differential equations but also in the simulation of delayed fractional-order control systems.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61004078, 61273012, the Foundation of State Key Laboratory for Novel Software Technology of Nanjing University, the China Postdoctoral Science Foundation Funded Project, and the Special Funds for Postdoctoral Innovative Projects of Shandong Province.

References

- [1] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, John Wiley & Sons, New York, NY, USA, 1993.
- [2] I. Podlubny, *Fractional Differential Equations*, vol. 198 of *Mathematics in Science and Engineering*, Academic Press, San Diego, Calif, USA, 1999.
- [3] P. L. Butzer and U. Westphal, "An introduction to fractional calculus," in *Applications of Fractional Calculus in Physics*, pp. 1-85, World Scientific Publishing, River Edge, NJ, USA, 2000.
- [4] Z. Wang, X. Huang, Y.-X. Li, and X.-N. Song, "A new image encryption algorithm based on fractional-order hyperchaotic Lorenz system," *Chinese Physics B*, vol. 22, Article ID 010504, 2013.
- [5] K. B. Oldham and J. Spanier, *The Fractional Calculus: Theory and Application of Differentiation and Integration to Arbitrary Order*, vol. 111 of *Mathematics in Science and Engineering*, Academic Press, New York, NY, USA, 1974.
- [6] R. Hilfer, Ed., *Applications of Fractional Calculus in Physics*, World Scientific Publishing, River Edge, NJ, USA, 2000.
- [7] P. J. Torvik and R. L. Bagley, "On the appearance of the fractional derivatives in the behaviour of real materials," *Journal of Applied Mechanics*, vol. 51, no. 2, pp. 294-298, 1984.
- [8] F. Mainardi, "Fractional relaxation-oscillation and fractional diffusion-wave phenomena," *Chaos, Solitons & Fractals*, vol. 7, no. 9, pp. 1461-1477, 1996.
- [9] N. J. Ford and A. C. Simpson, "The numerical solution of fractional differential equations: speed versus accuracy," *Numerical Algorithms*, vol. 26, no. 4, pp. 333-346, 2001.
- [10] V. Daftardar-Gejji and A. Babakhani, "Analysis of a system of fractional differential equations," *Journal of Mathematical Analysis and Applications*, vol. 293, no. 2, pp. 511-522, 2004.
- [11] K. Diethelm and N. J. Ford, "Analysis of fractional differential equations," *Journal of Mathematical Analysis and Applications*, vol. 265, no. 2, pp. 229-248, 2002.
- [12] K. Diethelm, N. J. Ford, and A. D. Freed, "A predictor-corrector approach for the numerical solution of fractional differential equations," *Nonlinear Dynamics*, vol. 29, no. 1-4, pp. 3-22, 2002.
- [13] K. Diethelm, N. J. Ford, and A. D. Freed, "Detailed error analysis for a fractional Adams method," *Numerical Algorithms*, vol. 36, no. 1, pp. 31-52, 2004.
- [14] K. Diethelm, N. J. Ford, A. D. Freed, and Yu. Luchko, "Algorithms for the fractional calculus: a selection of numerical methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 6-8, pp. 743-773, 2005.
- [15] K. Diethelm and N. J. Ford, "Multi-order fractional differential equations and their numerical solution," *Applied Mathematics and Computation*, vol. 154, no. 3, pp. 621-640, 2004.
- [16] Z. Odibat, "Approximations of fractional integrals and Caputo fractional derivatives," *Applied Mathematics and Computation*, vol. 178, no. 2, pp. 527-533, 2006.
- [17] J. T. Edwards, N. J. Ford, and A. C. Simpson, "The numerical solution of linear multi-term fractional differential equations; systems of equations," *Journal of Computational and Applied Mathematics*, vol. 148, no. 2, pp. 401-418, 2002.
- [18] K. Diethelm, "An investigation of some nonclassical methods for the numerical approximation of Caputo-type fractional derivatives," *Numerical Algorithms*, vol. 47, no. 4, pp. 361-390, 2008.

- [19] K. Diethelm, "An improvement of a nonclassical numerical method for the computation of fractional derivatives," *Journal of Vibration and Acoustics*, vol. 131, no. 1, Article ID 014502, 4 pages, 2009.
- [20] C. Li and C. Tao, "On the fractional Adams method," *Computers & Mathematics with Applications*, vol. 58, no. 8, pp. 1573–1588, 2009.
- [21] R. Garrappa, "On some explicit Adams multistep methods for fractional differential equations," *Journal of Computational and Applied Mathematics*, vol. 229, no. 2, pp. 392–399, 2009.
- [22] L. Galeone and R. Garrappa, "Fractional Adams-Moulton methods," *Mathematics and Computers in Simulation*, vol. 79, no. 4, pp. 1358–1367, 2008.
- [23] L. Galeone and R. Garrappa, "Explicit methods for fractional differential equations and their stability properties," *Journal of Computational and Applied Mathematics*, vol. 228, no. 2, pp. 548–560, 2009.
- [24] W. Deng, "Short memory principle and a predictor-corrector approach for fractional differential equations," *Journal of Computational and Applied Mathematics*, vol. 206, no. 1, pp. 174–188, 2007.
- [25] C. Li and G. Peng, "Chaos in Chen's system with a fractional order," *Chaos, Solitons & Fractals*, vol. 22, no. 2, pp. 443–450, 2004.
- [26] X. Wu, J. Li, and G. Chen, "Chaos in the fractional order unified system and its synchronization," *Journal of the Franklin Institute*, vol. 345, no. 4, pp. 392–401, 2008.
- [27] M. S. Tavazoei and M. Haeri, "Chaotic attractors in incommensurate fractional order systems," *Physica D*, vol. 237, no. 20, pp. 2628–2637, 2008.
- [28] S. Dadras and H. R. Momeni, "Control of a fractional-order economical system via sliding mode," *Physica A*, vol. 389, no. 12, pp. 2434–2442, 2010.
- [29] Y. Chen and K. L. Moore, "Analytical stability bound for a class of delayed fractional-order dynamic systems," *Nonlinear Dynamics*, vol. 29, no. 1–4, pp. 191–200, 2002.
- [30] P. Lanusse, H. Benlaoukli, D. Nelson-Gruel, and A. Oustaloup, "Fractional-order control and interval analysis of SISO systems with time-delayed state," *IET Control Theory & Applications*, vol. 2, no. 1, pp. 16–23, 2008.
- [31] X. Zhang, "Some results of linear fractional order time-delay system," *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 407–411, 2008.

