

RESEARCH

Open Access



# High order algorithms for numerical solution of fractional differential equations

Mohammad Shahbazi Asl<sup>1\*</sup>, Mohammad Javidi<sup>1</sup> and Yubin Yan<sup>2</sup>

\*Correspondence:

[mshahbazia@yahoo.com](mailto:mshahbazia@yahoo.com)

<sup>1</sup>Department of Mathematics,  
University of Tabriz, Tabriz, Iran  
Full list of author information is  
available at the end of the article

## Abstract

In this paper, two novel high order numerical algorithms are proposed for solving fractional differential equations where the fractional derivative is considered in the Caputo sense. The total domain is discretized into a set of small subdomains and then the unknown functions are approximated using the piecewise Lagrange interpolation polynomial of degree three and degree four. The detailed error analysis is presented, and it is analytically proven that the proposed algorithms are of orders 4 and 5. The stability of the algorithms is rigorously established and the stability region is also achieved. Numerical examples are provided to check the theoretical results and illustrate the efficiency and applicability of the novel algorithms.

**MSC:** 26A33; 65D05; 65D30

**Keywords:** Numerical algorithm; Fractional differential equation; Caputo fractional derivative; Stability analysis; Error estimates

## 1 Introduction

The subject of fractional calculus (theory of integration and differentiation of arbitrary order) can be considered as an old yet novel topic. It has been an ongoing topic for more than 300 years; however, since the 1970s, it has been gaining increasing attention [1]. Firstly, there were almost no practical applications of fractional calculus (FC), and it was considered by many as an abstract area containing only mathematical manipulations of little or no use [2–4]. Recently, FC has been widely used in various applications in almost every field of science, engineering, and mathematics, and it has gained considerable importance due to its frequent applications in fluid flow, polymer rheology, economics, biophysics, control theory, psychology, and so on [5, 6].

The main reason that fractional differential equations (FDEs) are being used to model real phenomena is that they are nonlocal in nature, that is, a realistic model of a physical phenomenon depends not only on the time instant but also on the previous time history [7]. In other words, fractional derivative (FD) provides a perfect tool when it is used to describe the memory and hereditary properties of various materials and processes [8, 9]. Some of the other main differences between fractional calculus and classical calculus are: (i) FDEs are, at least, as stable as their integer order counterparts [10, 11]; (ii) using FDEs

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

can help to reduce the errors arising from the neglected parameters in modeling real-life phenomena [12, 13]; (iii) in some situations, the FDE models seem more consistent with the real phenomena than the integer order models [14, 15]; (iv) modeling of real-life processes with FD is more reliable and accurate than the classical derivative [16, 17]; (v) fractional order models are more general [18] and in the limit results obtained from FC coincide with those obtained from classical calculus [19], and so on.

The wide applicability of FC in the field of science and engineering motivates researchers to try to find out the analytical or numerical solutions for FDEs. It is well known that the analytical and closed solutions of FDEs cannot generally be obtained and if luckily obtained always contain some infinite series (such as Mittag-Leffler function), which makes evaluation very expensive [20, 21]. For this reason, necessarily, one may need an efficient approximate and numerical technique for the solution of FDEs [22].

Odibat et al. constructed a numerical scheme for the numerical solution of FDEs based on the modified trapezoidal rule and the fractional Euler method [23]. To present a numerical solution scheme for the fractional differential equations, authors of [24] divided total time into a set of small intervals and utilized quadratic interpolation polynomial between two successive intervals to approximate unknown functions. Cao and Xu applied quadratic interpolation polynomial to construct a high order scheme based on the so-called block-by-block approach for the fractional ordinary differential equations [25]. The convergence order of this scheme is  $3 + \alpha$  for  $0 < \alpha \leq 1$  and 4 for  $\alpha > 1$ . Diethelm proposed an implicit numerical algorithm for solving FDEs by using piecewise linear interpolation polynomials to approximate the Hadamard finite-part integral [26]. Yan et al. designed a high order numerical scheme for solving a linear fractional differential equation defined in terms of the Riemann–Liouville fractional derivative [27]. This method is based on a direct discretisation of the fractional differential operator, and the order of convergence of the method is  $O(h^{3-\alpha})$ . A high order fractional Adams-type method for solving nonlinear FDEs is also obtained in this paper. Asl et al. [28] applied the piecewise cubic interpolation polynomial to construct a numerical scheme for solving linear FDEs. The convergence order of this scheme is  $O(h^{4-\alpha})$  for  $0 < \alpha \leq 1$ .

In recent years a lot of attention has been paid to finding effective numerical methods to simulate the fractional partial differential equations, such as meshless methods [29–31] and homotopy analysis method [32]. Dehghan et al. presented a fourth order numerical method for solving a fractional partial integro-differential equation [33]. Authors of [34] proposed a high accuracy numerical method to simulate a nonlinear fractional partial integro-differential equation.

The present study proposes to develop two numerical algorithms for the numerical approximation of a class of FDEs which are expressed in terms of Caputo-type fractional derivatives. In these algorithms the properties of the Caputo derivative are used to reduce the FDE into a Volterra-type integral equation of the second kind. The outline of the paper is as follows. Numerical algorithms are presented in Sect. 2 by using the piecewise Lagrange interpolation polynomial of degree three and degree four. Section 3 deals with the error analysis of the presented algorithms, and stability analysis of these algorithms is given on Sect. 4. Linear stability analysis of the presented schemes is given in Sect. 5 to achieve stability region of these methods. To demonstrate the effectiveness and high accuracy of the proposed methods, some numerical examples are provided in Sect. 6. The

experimental orders of convergence are also investigated in this section. Finally, some concluding remarks are given in Sect. 7.

## 2 Numerical algorithms

The present paper further discusses numerical algorithms for the initial value problem for a FDE in the form

$$\begin{cases} {}^C_0D_t^\alpha y(t) = f(t, y(t)) = \beta y(t) + g(t), & 0 \leq t \leq T, \\ y^{(k)}(t_0) = y_0^{(k)}, & k = 0, 1, \dots, \lceil \alpha \rceil - 1, \end{cases} \tag{1}$$

where  ${}^C_0D_t^\alpha$  denotes the Caputo fractional derivative(FD). The reason for choosing Caputo derivative in (1) is partly due to the fact that the Caputo derivative of a constant is zero. More importantly, the Caputo FD determines an ordinary derivative followed by a fractional integral to obtain the preferred order of FDs. That means the Caputo FD allows us to couple FDEs with traditional (local) boundary and initial conditions. These conditions have clear physical interpretation and are familiar to us [8, 35, 36]. Motivated by these advantages, the Caputo FD has been applied to model various physical and real-life problems such as epidemic model for HIV transmission [8, 9], symbiosis system [16], multi-pulse splitting process [17], reaction–diffusion problems [37, 38], and tumor–host models [39]. Presented numerical algorithms are based on the analytical property that the initial value problem (1) is equivalent to the Volterra integral equation [40, 41]

$$y(t) = h(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau, \tag{2}$$

where  $h(t) = \sum_{j=0}^{\lceil \alpha \rceil - 1} \frac{t^j}{j!} y^{(j)}(0)$ , in the sense that a continuous function solves (2) if and only if it solves (1). The piecewise Lagrange interpolation polynomials of degree three and degree four are used to approximate the integral in (2). For an integer  $N$  and the given time  $T$ , the interval  $[0, T]$  is divided into  $t_j = jh, j = 0, \dots, N$ , where  $h = T/N$  is the step length. The numerical solution of Eq. (1) at the point  $t_j$  is denoted by  $y_j$ . For notational convenience, let  $F(\tau) = f(\tau, y(\tau))$  and  $F_j = f(t_j, y_j) = \beta y_j + g(t_j)$ .

### 2.1 Numerical algorithm I

We start with computing the value of  $y(t)$  at  $t_1, t_2$ , and  $t_3$ , simultaneously. Consider the following integral for the first three steps ( $k = 0, 1, 2$ ):

$$\begin{aligned} I_{k+1} &= \int_0^{t_{k+1}} (t_{k+1} - \tau)^{\alpha-1} F(\tau) d\tau = \sum_{j=0}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} F(\tau) d\tau \\ &\approx \sum_{j=0}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \tilde{F}(\tau) d\tau = \sum_{j=0}^3 d_j^{k+1} F(t_j), \end{aligned} \tag{3}$$

where  $\tilde{F}(\tau)$  is chosen to be the piecewise Lagrange cubic interpolation polynomial of  $F(\tau)$  associated with the nodes  $t_0, t_1, t_2,$  and  $t_3$ . In this way, we have

$$\hat{d}_j^1 = \begin{cases} 6\alpha^3 + 25\alpha^2 + 23\alpha, & j = 0, \\ 6(3\alpha^2 + 10\alpha + 6), & j = 1, \\ -9\alpha^2 - 21\alpha, & j = 2, \\ -2\alpha^2 - 4\alpha, & j = 3, \end{cases}, \quad \hat{d}_j^2 = \begin{cases} 2^{\alpha+1}\alpha(\alpha + 2)(3\alpha + 1), & j = 0, \\ 3 \times 2^{\alpha+2}(3\alpha^2 + 5\alpha), & j = 1, \\ 3 \times 2^{\alpha+1}(-3\alpha^2 + \alpha + 6), & j = 2, \\ 2^{\alpha+2}\alpha^2 - \alpha & j = 3, \end{cases} \tag{4}$$

$$\hat{d}_j^3 = \begin{cases} 3^{\alpha+1}(2\alpha^3 + \alpha^2 + 3\alpha), & j = 0, \\ 2 \times 3^{\alpha+3}\alpha^2, & j = 1, \\ -3^{\alpha+3}(\alpha^2 - 3\alpha), & j = 2, \\ 2 \times 3^{\alpha+1}(\alpha^2 - 4\alpha + 6) & j = 3, \end{cases}, \quad d_j^{k+1} = \frac{h^\alpha}{6\alpha(\alpha + 1)(\alpha + 2)(\alpha + 3)} \hat{d}_j^{k+1}.$$

To explain further, for  $y_1$ , we have

$$y_1 = h(t_1) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^3 \int_{t_0}^{t_1} (t_1 - \tau)^{\alpha-1} L_{3,j}(\tau) F_j d\tau = h(t_1) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^3 d_j^1 F_j,$$

where

$$L_{3,j}(\tau) = \prod_{\substack{i=0 \\ i \neq j}}^3 \frac{\tau - t_i}{t_j - t_i}, \quad d_j^1 = \int_{t_0}^{t_1} (t_1 - \tau)^{\alpha-1} L_{3,j}(\tau) d\tau, \quad j = 0, 1, 2, 3.$$

For this reason, after some elementary calculations,  $y_{k+1}$  for the first three steps  $k = 0, 1, 2$  can be approximated as follows:

$$y_{k+1} = h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^3 d_j^{k+1} (\beta y_j + g(t_j)),$$

$$\left( \frac{\Gamma(\alpha) - \beta d_{k+1}^{k+1}}{\Gamma(\alpha)} \right) y_{k+1} = h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{\substack{j=0 \\ j \neq k+1}}^3 d_j^{k+1} F_j + \frac{d_{k+1}^{k+1}}{\Gamma(\alpha)} g(t_{k+1}), \quad k = 0, 1, 2. \tag{5}$$

As it is mentioned above, the first three step solutions  $y_1, y_2,$  and  $y_3$  are coupled in (5), thus need to be solved simultaneously. This procedure leads to a linear system of equations, which can be solved explicitly by means of back-substitution.

To construct the scheme for the next steps,  $I_{k+1}, k \geq 3,$  is discretized as follows:

$$I_{k+1} \approx \left[ \sum_{j=0}^2 \int_{t_j}^{t_{j+1}} \tilde{F}(\tau) + \sum_{j=3}^k \int_{t_j}^{t_{j+1}} \tilde{F}_{j+1}(\tau) \right] (t_{k+1} - \tau)^{\alpha-1} d\tau = \sum_{j=0}^{k+1} d_j^{k+1} F(t_j), \tag{6}$$

in which, like in (3), for the first three integrals ( $j = 0, 1, 2, 3$ ),  $\tilde{F}$  is the piecewise Lagrange cubic interpolation polynomial of  $F(\tau)$  associated with the nodes  $t_0, t_1, t_2,$  and  $t_3$ . For the remaining integrals ( $j = 3, 4, \dots, k + 1$ ),  $\tilde{F}_{j+1}$  is chosen to be the piecewise Lagrange cubic

interpolation polynomial of  $F(\tau)$  associated with the nodes  $t_{j-2}, t_{j-1}, t_j$ , and  $t_{j+1}$ . In this way, for  $k \geq 3$ , we have

$$\begin{aligned} \hat{d}_0^{k+1} &= (k+1)^\alpha [6\alpha^3 + 25\alpha^2 + 23\alpha + 12\alpha k^2 - k(11\alpha^2 + 31\alpha + 6(k-2)(k-1))] \\ &\quad + 2P(1)(k-2)^{\alpha+1}, \\ \hat{d}_1^{k+1} &= 6(k+1)^{\alpha+1} [3k(k-3) - 5k\alpha + 3\alpha^2 + 10\alpha + 6] + \frac{6}{3}P(2)(k-3)^{\alpha+1} \\ &\quad - \frac{24}{3}P(1)(k-2)^{\alpha+1}, \\ \hat{d}_2^{k+1} &= -3(k+1)^{\alpha+1} [6k(k-2) - 8k\alpha + 3\alpha^2 + 7\alpha] + 2P(3)(k-4)^{\alpha+1} \\ &\quad - \frac{24}{3}P(2)(k-3)^{\alpha+1} + 12P(1)(k-2)^{\alpha+1}, \\ \hat{d}_3^{k+1} &= 2(k+1)^{\alpha+1} [\alpha^2 + 2\alpha + 3k^2 - 3\alpha k - 3k] + 2P(4)(k-5)^{\alpha+1} - 8P(3)(k-4)^{\alpha+1} \\ &\quad + 12P(2)(k-3)^{\alpha+1} - 8P(1)(k-2)^{\alpha+1}, \\ \hat{d}_j^{k+1} &= 2P(j-3)(k-j+2)^{\alpha+1} - 8P(j-2)(k-j+1)^{\alpha+1} + 12P(j-1)(k-j)^{\alpha+1} \\ &\quad - 8P(j)(k-j-1)^{\alpha+1} + 2P(j+1)(k-j-2)^{\alpha+1}, \quad 4 \leq j \leq k-2, \\ \hat{d}_{k-1}^{k+1} &= 6\hat{d}_{k+1}^{k+1} - 2^{\alpha+4}\varphi + 2 \times 3^{\alpha+1} [\alpha^2 + 14\alpha + 60], \quad \hat{d}_k^{k+1} = 2^{\alpha+2}\varphi - 4\hat{d}_{k+1}^{k+1}, \\ \hat{d}_{k+1}^{k+1} &= 2\alpha^2 + 16\alpha + 36, \end{aligned}$$

in which  $d_j^{k+1} = \frac{h^\alpha}{6\alpha(\alpha+1)(\alpha+2)(\alpha+3)} \hat{d}_j^{k+1}$  and

$$\varphi = \alpha^2 + 11\alpha + 36, \quad P(j) = \alpha(\alpha + 2) + 3j^2 - 3j(\alpha + 2k + 1) + 3k^2 + 3(\alpha + 1)k.$$

The following special cases should be excluded:

$$\begin{cases} \hat{d}_2^4 = -3[4^{\alpha+1}(3\alpha^2 - 17\alpha + 18) - 2\hat{d}_{k+1}^{k+1}], \\ \hat{d}_3^4 = 2[4^{\alpha+1}(\alpha^2 - 7\alpha + 18) - 2\hat{d}_{k+1}^{k+1}], \end{cases} \tag{7}$$

$$\hat{d}_3^5 = 2[5^{\alpha+1}(\alpha^2 - 10\alpha + 36) - 2^{\alpha+3}\varphi + 3\hat{d}_{k+1}^{k+1}]. \tag{8}$$

In this way, after some explicit calculations,  $y_{k+1}$  for  $k \geq 3$  can be approximated as follows:

$$\begin{aligned} y_{k+1} &= h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{k+1} d_j^{k+1} (\beta y_j + g(t_j)), \\ \left( \frac{\Gamma(\alpha) - \beta d_{k+1}^{k+1}}{\Gamma(\alpha)} \right) y_{k+1} &= h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \left( \sum_{j=0}^k d_j^{k+1} F_j + d_{k+1}^{k+1} g(t_{k+1}) \right), \quad k \geq 3. \end{aligned} \tag{9}$$

Summing up the above arguments, we have the following novel scheme:

$$\left( \frac{\Gamma(\alpha) - \beta d_{k+1}^{k+1}}{\Gamma(\alpha)} \right) y_{k+1} = \begin{cases} h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{\substack{j=0 \\ j \neq k+1}}^3 d_j^{k+1} F_j + \frac{d_{k+1}^{k+1}}{\Gamma(\alpha)} g(t_{k+1}) & \text{if } k = 0, 1, 2, \\ h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} (\sum_{j=0}^k d_j^{k+1} F_j + d_{k+1}^{k+1} g(t_{k+1})) & \text{if } k \geq 3. \end{cases} \tag{10}$$

### 2.2 Numerical algorithm II

Consider the following integral for the first four steps ( $k = 0, 1, 2, 3$ ):

$$\begin{aligned}
 I_{k+1} &= \int_0^{t_{k+1}} (t_{k+1} - \tau)^{\alpha-1} F(\tau) d\tau = \sum_{j=0}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} F(\tau) d\tau \\
 &= \sum_{j=0}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \tilde{F}(\tau) d\tau = \sum_{j=0}^4 b_j^{k+1} F(t_j),
 \end{aligned} \tag{11}$$

where  $\tilde{F}(\tau)$  is the piecewise Lagrange interpolation polynomial of degree four associated with the nodes  $t_0, t_1, t_2, t_3,$  and  $t_4$ . Therefore, one can achieve the following weights:

$$\begin{cases}
 \hat{b}_0^1 = \alpha [12\alpha^3 + 95\alpha^2 + 230\alpha + 165], \\
 \hat{b}_1^1 = 4 [12\alpha^3 + 82\alpha^2 + 157\alpha + 72], \\
 \hat{b}_2^1 = -6\alpha [6\alpha^2 + 35\alpha + 47], \\
 \hat{b}_3^1 = 4\alpha [4\alpha^2 + 22\alpha + 27], \\
 \hat{b}_4^1 = -\alpha [3\alpha^2 + 16\alpha + 19], \\
 \\
 \hat{b}_0^2 = 2^{\alpha+1} \alpha [6\alpha^3 + 35\alpha^2 + 55\alpha + 20], \\
 \hat{b}_1^2 = 2^{\alpha+5} \alpha [3\alpha^2 + 14\alpha + 14], \\
 \hat{b}_2^2 = 3 \times 2^{\alpha+3} (\alpha + 1)(\alpha + 3)(4 - 3\alpha), \\
 \hat{b}_3^2 = 2^{\alpha+5} \alpha [\alpha^2 + 2\alpha - 2], \\
 \hat{b}_4^2 = 2^{\alpha+1} \alpha [-3\alpha^2 - 5\alpha + 4], \\
 \\
 \hat{b}_0^3 = 3^{\alpha+1} \alpha [4\alpha^3 + 15\alpha^2 + 20\alpha + 15], \\
 \hat{b}_1^3 = 4 \times 3^{\alpha+2} \alpha [4\alpha^2 + 10\alpha + 3], \\
 \hat{b}_2^3 = 2 \times 3^{\alpha+3} \alpha [-2\alpha^2 + \alpha + 9], \\
 \hat{b}_3^3 = 4 \times 3^{\alpha+1} [4\alpha^3 - 6\alpha^2 - \alpha + 24], \\
 \hat{b}_4^3 = -3^{\alpha+2} \alpha [\alpha^2 - 2\alpha + 3], \\
 \\
 \hat{b}_0^4 = 4^{\alpha+1} \alpha^2 [3\alpha^2 + 5\alpha + 20], \\
 \hat{b}_1^4 = 4^{\alpha+3} \alpha [3\alpha^2 + \alpha + 4], \\
 \hat{b}_2^4 = 3 \times 4^{\alpha+2} \alpha [-3\alpha^2 + 11\alpha - 4], \\
 \hat{b}_3^4 = 4^{\alpha+3} \alpha [\alpha^2 - 5\alpha + 12], \\
 \hat{b}_4^4 = -4^{\alpha+1} (\alpha - 2)(3\alpha^2 - 11\alpha + 36),
 \end{cases} \tag{12}$$

here  $b_j^{k+1} = \frac{h^\alpha}{12\alpha(\alpha+1)(\alpha+2)(\alpha+3)(\alpha+4)} \hat{b}_j^{k+1}$ . Hence  $y_{k+1}$  for the first four steps  $k = 0, 1, 2, 3$  can be determined as follows:

$$\begin{aligned}
 y_{k+1} &= h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^4 b_j^{k+1} (Ay_j + g(t_j)), \\
 \left( \frac{\Gamma(\alpha) - \beta b_{k+1}^{k+1}}{\Gamma(\alpha)} \right) y_{k+1} &= h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{\substack{j=0 \\ j \neq k+1}}^4 b_j^{k+1} F_j + \frac{b_{k+1}^{k+1}}{\Gamma(\alpha)} g(t_{k+1}), \quad 0 \leq k \leq 3.
 \end{aligned} \tag{13}$$

It is obvious that the first four step solutions  $y_1, y_2, y_3$ , and  $y_4$  are coupled in (13), thus need to be solved simultaneously. This procedure leads to a linear system of equations, which can be solved explicitly by means of back-substitution.

To design the scheme for the next steps,  $I_{k+1}, k \geq 4$  is discretized as follows:

$$I_{k+1} = \left[ \sum_{j=0}^3 \int_{t_j}^{t_{j+1}} \tilde{F}(\tau) + \sum_{j=4}^k \int_{t_j}^{t_{j+1}} \tilde{F}_{j+1}(\tau) \right] (t_{k+1} - \tau)^{\alpha-1} d\tau = \sum_{j=0}^{k+1} b_j^{k+1} F(t_j), \tag{14}$$

in which, as in (11), for the first four integrals ( $j = 0, 1, 2, 3$ ),  $\tilde{F}$  is a piecewise Lagrange interpolation polynomial of degree four associated with the nodes  $t_0, t_1, t_2, t_3$ , and  $t_4$ . For the remaining integrals ( $j = 4, 5, \dots, k + 1$ ),  $\tilde{F}_{j+1}$  is the piecewise Lagrange interpolation polynomial of degree four associated with the nodes  $t_{j-3}, t_{j-2}, t_{j-1}, t_j$ , and  $t_{j+1}$ . In this way, for  $k \geq 4$ , we have the following weights:

$$\begin{aligned} \hat{b}_0^{k+1} &= (k + 1)^\alpha [12\alpha^4 + 5\alpha^2(7k^2 - 31k + 46) - 5\alpha(k - 3)(6k^2 - 13k + 11) \\ &\quad + 5\alpha^3(19 - 5k) + 12k(k - 3)(k - 2)(k - 1)] - P(2)(k - 3)^{\alpha+1}, \\ \hat{b}_1^{k+1} &= -4(k + 1)^{\alpha+1} [h_2(-2) - 15(\alpha + 4)(\alpha^2 + 4\alpha + 3k^2 - \alpha k + 3k + 6)] \\ &\quad - P(3)(k - 4)^{\alpha+1} + 5P(2)(k - 3)^{\alpha+1}, \\ \hat{b}_2^{k+1} &= 6(k + 1)^{\alpha+1} [h_2(-2) - (\alpha + 4)(9\alpha^2 + 37\alpha + 42k^2 - 8\alpha k + 60k + 72)] \\ &\quad - P(4)(k - 5)^{\alpha+1} + 5P(3)(k - 4)^{\alpha+1} - 10P(2)(k - 3)^{\alpha+1}, \\ \hat{b}_3^{k+1} &= -4(k + 1)^{\alpha+1} [h_2(-2) - (\alpha + 4)(7\alpha^2 + 32\alpha + 39k^2 - 3\alpha k + 69k + 72)] \\ &\quad - P(5)(k - 6)^{\alpha+1} + 5P(4)(k - 5)^{\alpha+1} - 10P(3)(k - 4)^{\alpha+1} + 10P(2)(k - 3)^{\alpha+1}, \\ \hat{b}_4^{k+1} &= (k + 1)^{\alpha+1} [h_2(-2) - 6(\alpha + 4)(\alpha^2 + 5\alpha + 6k^2 + 12k + 12)] - P(6)(k - 7)^{\alpha+1} \\ &\quad + 5P(5)(k - 6)^{\alpha+1} - 10P(4)(k - 5)^{\alpha+1} + 10P(3)(k - 4)^{\alpha+1} - 5P(2)(k - 3)^{\alpha+1}, \\ \hat{b}_j^{k+1} &= -P(j + 2)(-j + k - 3)^{\alpha+1} + P(j - 3)(-j + k + 2)^{\alpha+1} - 5P(j - 2)(-j + k + 1)^{\alpha+1} \\ &\quad + 10P(j - 1)(k - j)^{\alpha+1} + 5P(j + 1)(-j + k - 2)^{\alpha+1} - 10P(j)(-j + k - 1)^{\alpha+1}, \\ &\quad 5 \leq j \leq k - 3, \\ \hat{b}_{k-2}^{k+1} &= 2^{\alpha+2} [5\psi_2 + 2^\alpha (3\alpha^3 + 71\alpha^2 + 674\alpha + 2520)] - 5 \times 3^{\alpha+2} \psi_1 - 10\hat{b}_{k+1}^{k+1}, \\ \hat{b}_{k-1}^{k+1} &= 10[\hat{b}_{k+1}^{k+1} - 2^\alpha \psi_2] + 3^{\alpha+2} \psi_1, \quad \hat{b}_k^{k+1} = 2^{\alpha+1} \psi_2 - 5\hat{b}_{k+1}^{k+1}, \\ \hat{b}_{k+1}^{k+1} &= 3\alpha^3 + 38\alpha^2 + 173\alpha + 288, \end{aligned}$$

where  $b_j^{k+1} = \frac{h^\alpha}{12\alpha(\alpha+1)(\alpha+2)(\alpha+3)(\alpha+4)} \hat{b}_j^{k+1}$  and

$$\begin{aligned} P(j) &= 3\alpha^3 + \alpha(18j^2 - 36jk - 41j + 18k^2 + 41k + 19) + \alpha^2(-11j + 11k + 16) \\ &\quad - 12(j - k - 2)(j - k - 1)(j - k), \\ \psi_1 &= \alpha^3 + 20\alpha^2 + 157\alpha + 480, \quad \psi_2 = 3\alpha^3 + 49\alpha^2 + 304\alpha + 720. \end{aligned}$$

The following special cases should be excluded:

$$\begin{cases} \hat{b}_2^5 = -10[3 \times 5^\alpha(6\alpha^3 - 41\alpha^2 + 91\alpha - 96) + \hat{b}_{k+1}^{k+1}], \\ \hat{b}_3^5 = 10[5^\alpha(8\alpha^3 - 68\alpha^2 + 278\alpha - 288) + \hat{b}_{k+1}^{k+1}], \\ \hat{b}_4^5 = -5[5^\alpha(3\alpha^3 - 28\alpha^2 + 143\alpha - 288) + \hat{b}_{k+1}^{k+1}], \end{cases} \tag{15}$$

$$\begin{cases} \hat{b}_3^6 = 10(2^{\alpha+1}\psi_2 - \hat{b}_{k+1}^{k+1}) + 2^{\alpha+5}3^{\alpha+1}(\alpha^3 - 12\alpha^2 + 68\alpha - 120), \\ \hat{b}_4^6 = 10(-2^\alpha\psi_2 + \hat{b}_{k+1}^{k+1}) - 2^{\alpha+1}3^{\alpha+2}(\alpha^3 - 13\alpha^2 + 88\alpha - 240), \end{cases} \tag{16}$$

$$\hat{b}_4^7 = 7^{\alpha+1}(-3\alpha^3 + 50\alpha^2 - 421\alpha + 1440) - 5 \times 3^{\alpha+2}\psi_1 + 5 \times 2^{\alpha+2}\psi_2 - 10\hat{b}_{k+1}^{k+1}. \tag{17}$$

Therefore  $y_{k+1}$  for  $k \geq 4$  can be approximated as follows:

$$y_{k+1} = h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{k+1} b_j^{k+1}(\beta y_j + g(t_j)),$$

$$\left(\frac{\Gamma(\alpha) - \beta b_{k+1}^{k+1}}{\Gamma(\alpha)}\right) y_{k+1} = h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \left(\sum_{j=0}^k b_j^{k+1} F_j + b_{k+1}^{k+1} g(t_{k+1})\right), \quad k \geq 4. \tag{18}$$

Thus a new numerical algorithm II is described by (13) and (18) with the weights  $b_j^{k+1}$  defined as above.

### 3 Error analysis

For numerical algorithm I, the truncation error at step  $k + 1$  is defined by [25]

$$r_{k+1}(h) := y(t_{k+1}) - \tilde{y}_{k+1}, \tag{19}$$

where  $\tilde{y}_{k+1}$  is an approximation to  $y(t_{k+1})$ , evaluated by using algorithm I (10) with exact previous solutions, i.e., for  $k \geq 3$ ,

$$\tilde{y}_{k+1} = h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{k+1} d_j^{k+1} F(t_j). \tag{20}$$

For numerical algorithm II (18), the definition of truncation error is the same as (19), where  $\tilde{y}_{k+1}$  for  $k \geq 4$  is as follows:

$$\tilde{y}_{k+1} = h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{k+1} b_j^{k+1} F(t_j). \tag{21}$$

**Theorem 1** *Let  $r_{k+1}(h)$  be the truncation error defined in (19). If  $F(\tau) \in C^4[0, T]$  for some suitable chosen  $T$ , then for numerical algorithm I (10) there exists a positive constant  $C > 0$ , independent of  $h$ , such that*

$$|r_{k+1}(h)| \leq Ch^4.$$

*Proof* By the definition of truncation error (19) we have

$$|r_{k+1}(h)| = |y(t_{k+1}) - \tilde{y}_{k+1}|.$$

We compute  $y(t_{k+1})$  and  $\tilde{y}_{k+1}$  from (2) and (20), respectively, which yields

$$|r_{k+1}(h)| = \left| h(t_{k+1}) + \frac{1}{\Gamma(\alpha)} \int_0^{t_{k+1}} (t_{k+1} - \tau)^{\alpha-1} F(\tau) d\tau - h(t_{k+1}) - \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{k+1} d_j^{k+1} F(t_j) \right|.$$

We have, by (6),

$$\begin{aligned} |r_{k+1}(h)| &\leq \frac{1}{\Gamma(\alpha)} \left| \sum_{j=0}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} F(\tau) d\tau - \sum_{j=0}^2 \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \tilde{F}(\tau) d\tau - \sum_{j=3}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \tilde{F}_{j+1}(\tau) d\tau \right| \\ &\leq \frac{1}{\Gamma(\alpha)} \left[ \sum_{j=0}^2 \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} |F(\tau) - \tilde{F}(\tau)| d\tau + \sum_{j=3}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} |F(\tau) - \tilde{F}_{j+1}(\tau)| d\tau \right], \end{aligned}$$

where  $\tilde{F}(\tau)$  and  $\tilde{F}_{j+1}$  are defined by (6). Thus, according to the property of cubic interpolation polynomials,

$$\begin{aligned} |r_{k+1}(h)| &\leq \frac{1}{\Gamma(\alpha)} \left[ \sum_{j=0}^2 \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \left| \frac{F^{(4)}(\xi_1(\tau))}{4!} (\tau - t_0)(\tau - t_1)(\tau - t_2)(\tau - t_3) \right| d\tau \right. \\ &\quad \left. + \sum_{j=3}^k \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \left| \frac{F^{(4)}(\xi_j(\tau))}{4!} (\tau - t_{j-2})(\tau - t_{j-1})(\tau - t_j)(\tau - t_{j+1}) \right| d\tau \right], \end{aligned}$$

where  $\xi_1(\tau) \in [t_0, t_3]$  and  $\xi_j(\tau) \in [t_{j-2}, t_{j+1}]$ .

$$\begin{aligned} |r_{k+1}(h)| &\leq \frac{1}{\Gamma(\alpha)} \left[ \sum_{j=0}^2 \frac{\|F^{(4)}\|_\infty}{4!} |(\tilde{\tau}_j - t_0)(\tilde{\tau}_j - t_1)(\tilde{\tau}_j - t_2)(\tilde{\tau}_j - t_3)| \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} d\tau \right. \\ &\quad \left. + \sum_{j=3}^k \frac{\|F^{(4)}\|_\infty}{4!} |(\tilde{\tau}_j - t_{j-2})(\tilde{\tau}_j - t_{j-1})(\tilde{\tau}_j - t_j)(\tilde{\tau}_j - t_{j+1})| \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} d\tau \right]. \end{aligned}$$

Here,  $\tilde{\tau}_j \in [t_j, t_{j+1}]$  and the second integral mean value theorem is used. By the definition of  $t_j = jh$  and after evaluating the integrals in the above inequality, we get

$$|r_{k+1}(h)| \leq \frac{1}{\Gamma(\alpha)} \left[ \frac{\|F^{(4)}\|_\infty}{4!} (3h)^4 \sum_{j=0}^2 \frac{1}{\alpha} [(t_{k+1} - t_j)^\alpha - (t_{k+1} - t_{j+1})^\alpha] \right]$$

$$\begin{aligned}
 & + \frac{\|F^{(4)}\|_\infty}{4!} (3h)^4 \sum_{j=3}^k \frac{1}{\alpha} \left[ (t_{k+1} - t_j)^\alpha - (t_{k+1} - t_{j+1})^\alpha \right], \\
 |r_{k+1}(h)| & \leq \frac{1}{\Gamma(\alpha)} \frac{1}{\alpha} \frac{\|F^{(4)}\|_\infty}{4!} (3h)^4 \sum_{j=0}^k \left[ (t_{k+1} - t_j)^\alpha - (t_{k+1} - t_{j+1})^\alpha \right].
 \end{aligned}$$

The series in the above expression can be easily summed up, which yields

$$|r_{k+1}(h)| \leq \frac{3^4 \|F^{(4)}\|_\infty}{4! \Gamma(\alpha + 1)} h^4 (t_{k+1} - t_0)^\alpha = \left( \frac{3^4 \|F^{(4)}\|_\infty T^\alpha}{4! \Gamma(\alpha + 1)} \right) h^4. \quad \square$$

**Theorem 2** *Let  $r_{k+1}(h)$  be the truncation error defined in (19). If  $F(\tau) \in C^5[0, T]$  for some suitable chosen  $T$ , then for numerical algorithm II (13) and (18) there exists a positive constant  $C > 0$ , independent of  $h$ , such that*

$$|r_{k+1}(h)| \leq Ch^5.$$

*Proof* The proof is similar to the proof of Theorem 1. We omit the proof here. □

#### 4 Stability analysis

The stability of a numerical scheme mainly refers to that if there is a perturbation in the initial condition, then the small change causes small errors in the numerical solution [40, 41]. Suppose that  $y_{k+1}$  and  $\tilde{y}_{k+1}$  are numerical solutions in (10), and the initial conditions are given by  $y_0^{(i)}$  and  $\tilde{y}_0^{(i)}$  respectively. If there exists a positive constant  $C$  independent of  $h$  such that

$$|y_{k+1} - \tilde{y}_{k+1}| \leq C_{\alpha,T} \|y_0 - \tilde{y}_0\|_\infty, \tag{22}$$

then we conclude that scheme (10) is stable [42]. It is similar to defining the numerical stability for numerical algorithm II (13) and (18). Assume that  $F(\tau)$  is sufficiently smooth and  $C_\alpha > 0$  is independent of all discretization parameters. Firstly, we introduce two lemmas which will be used in stability analysis.

**Lemma 1** *For the weights of novel scheme (10), we have*

$$\sum_{j=0}^{k+1} |d_j^{k+1}| \leq C_\alpha T^\alpha, \tag{23}$$

where  $C_\alpha$  only depends on  $\alpha$ .

*Proof* For  $d_0^{k+1}$ , we have

$$\begin{aligned}
 |d_0^{k+1}| & = \left| \sum_{j=0}^2 \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} \frac{\tau - t_1}{t_0 - t_1} \frac{\tau - t_2}{t_0 - t_2} \frac{\tau - t_3}{t_0 - t_3} d\tau \right| \\
 & \leq \sum_{j=0}^2 \int_{t_j}^{t_{j+1}} |(t_{k+1} - \tau)^{\alpha-1}| \left| \frac{\tau - t_1}{t_0 - t_1} \frac{\tau - t_2}{t_0 - t_2} \frac{\tau - t_3}{t_0 - t_3} \right| d\tau
 \end{aligned}$$

$$\begin{aligned} &\leq \sum_{j=0}^2 \left| \frac{\tilde{t}_j - t_1}{t_0 - t_1} \frac{\tilde{t}_j - t_2}{t_0 - t_2} \frac{\tilde{t}_j - t_3}{t_0 - t_3} \right| \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} d\tau, \quad \tilde{t}_j \in [t_j, t_{j+1}], \\ |d_0^{k+1}| &\leq \frac{1}{6h^3} |(\xi_1 h - h)(\xi_1 h - 2h)(\xi_1 h - 3h)| \frac{1}{\alpha} [(t_{k+1})^\alpha - (t_{k+1} - t_1)^\alpha] \\ &\quad + \frac{1}{6h^3} |(\xi_2 h - h)(\xi_2 h - 2h)(\xi_2 h - 3h)| \frac{1}{\alpha} [(t_{k+1} - t_1)^\alpha - (t_{k+1} - t_2)^\alpha] \\ &\quad + \frac{1}{6h^3} |(\xi_3 h - h)(\xi_3 h - 2h)(\xi_3 h - 3h)| \frac{1}{\alpha} [(t_{k+1} - t_2)^\alpha - (t_{k+1} - t_3)^\alpha], \end{aligned}$$

where  $j - 1 \leq \xi_j \leq j, j = 1, 2, 3$ . Therefore we have

$$|d_0^{k+1}| \leq \frac{1}{6h^3} \frac{1}{\alpha} t_{k+1}^\alpha (6h^3 + 2h^3 + 2h^3) \leq \frac{5}{3\alpha} T^\alpha.$$

Using similar analysis it can be shown that for  $j = 1, 2, 3, k - 1, k, k + 1$  there exists  $C_\alpha$ , which has dissimilar values at each case, such that the following inequality holds:

$$|d_j^{k+1}| \leq C_\alpha T^\alpha, \quad j = 1, 2, 3, k - 1, k, k + 1. \tag{24}$$

For  $j = 4, 5, \dots, k - 2$ , we have

$$\begin{aligned} \sum_{j=4}^{k-2} |d_j^{k+1}| &\leq \sum_{j=4}^{k-2} \left[ \int_{t_{j-1}}^{t_j} |(t_{k+1} - \tau)^{\alpha-1}| \left| \frac{\tau - t_{j-3}}{t_j - t_{j-3}} \frac{\tau - t_{j-2}}{t_j - t_{j-2}} \frac{\tau - t_{j-1}}{t_j - t_{j-1}} \right| d\tau \right. \\ &\quad + \int_{t_j}^{t_{j+1}} |(t_{k+1} - \tau)^{\alpha-1}| \left| \frac{\tau - t_{j-2}}{t_j - t_{j-2}} \frac{\tau - t_{j-1}}{t_j - t_{j-1}} \frac{\tau - t_{j+1}}{t_j - t_{j+1}} \right| d\tau \\ &\quad + \int_{t_{j+1}}^{t_{j+2}} |(t_{k+1} - \tau)^{\alpha-1}| \left| \frac{\tau - t_{j-1}}{t_j - t_{j-1}} \frac{\tau - t_{j+1}}{t_j - t_{j+1}} \frac{\tau - t_{j+2}}{t_j - t_{j+2}} \right| d\tau \\ &\quad \left. + \int_{t_{j+2}}^{t_{j+3}} |(t_{k+1} - \tau)^{\alpha-1}| \left| \frac{\tau - t_{j+1}}{t_j - t_{j+1}} \frac{\tau - t_{j+2}}{t_j - t_{j+2}} \frac{\tau - t_{j+3}}{t_j - t_{j+3}} \right| d\tau \right], \\ \sum_{j=4}^{k-2} |d_j^{k+1}| &\leq \sum_{j=4}^{k-2} \left[ \left| \frac{\tilde{t}_1 - t_{j-3}}{3h} \frac{\tilde{t}_1 - t_{j-2}}{2h} \frac{\tilde{t}_1 - t_{j-1}}{h} \right| \int_{t_{j-1}}^{t_j} (t_{k+1} - \tau)^{\alpha-1} d\tau \right. \\ &\quad + \left| \frac{\tilde{t}_2 - t_{j-2}}{2h} \frac{\tilde{t}_2 - t_{j-1}}{h} \frac{\tilde{t}_2 - t_{j+1}}{-h} \right| \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{\alpha-1} d\tau \\ &\quad + \left| \frac{\tilde{t}_3 - t_{j-1}}{h} \frac{\tilde{t}_3 - t_{j+1}}{-h} \frac{\tilde{t}_3 - t_{j+2}}{-2h} \right| \int_{t_{j+1}}^{t_{j+2}} (t_{k+1} - \tau)^{\alpha-1} d\tau \\ &\quad \left. + \left| \frac{\tilde{t}_4 - t_{j+1}}{-h} \frac{\tilde{t}_4 - t_{j+2}}{-2h} \frac{\tilde{t}_4 - t_{j+3}}{-3h} \right| \int_{t_{j+2}}^{t_{j+3}} (t_{k+1} - \tau)^{\alpha-1} d\tau \right], \end{aligned}$$

where  $\tilde{t}_1 \in [t_{j-1}, t_j], \tilde{t}_2 \in [t_j, t_{j+1}], \tilde{t}_3 \in [t_{j+1}, t_{j+2}],$  and  $\tilde{t}_4 \in [t_{j+2}, t_{j+3}].$  Hence, the above equation has the simplified form

$$\sum_{j=4}^{k-2} |d_j^{k+1}| \leq \left| \frac{6h^3}{6h^3} \right| \frac{1}{\alpha} \sum_{j=4}^{k-2} [(t_{k+1} - t_{j-1})^\alpha - (t_{k+1} - t_j)^\alpha] + \sum_{j=4}^{k-2} [(t_{k+1} - t_j)^\alpha - (t_{k+1} - t_{j+1})^\alpha]$$

$$\begin{aligned}
 & + \sum_{j=4}^{k-2} [(t_{k+1} - t_{j+1})^\alpha - (t_{k+1} - t_{j+2})^\alpha] + \sum_{j=4}^{k-2} [(t_{k+1} - t_{j+2})^\alpha - (t_{k+1} - t_{j+3})^\alpha] \\
 & = \frac{1}{\alpha} [ [(t_{k+1} - t_3)^\alpha - (t_{k+1} - t_{k-2})^\alpha] + [(t_{k+1} - t_4)^\alpha - (t_{k+1} - t_{k-1})^\alpha] \\
 & \quad + [(t_{k+1} - t_5)^\alpha - (t_{k+1} - t_k)^\alpha] + (t_{k+1} - t_6)^\alpha ], \\
 \sum_{j=4}^{k-2} |d_j^{k+1}| & \leq \frac{1}{\alpha} (t_{k+1} - t_3)^\alpha + (t_{k+1} - t_4)^\alpha + (t_{k+1} - t_5)^\alpha + (t_{k+1} - t_6)^\alpha - [t_3^\alpha + t_2^\alpha + t_1^\alpha] \\
 & \leq \frac{4}{\alpha} t_{k+1}^\alpha.
 \end{aligned}$$

Combining all the above results, by choosing sufficiently large  $C_\alpha$  and also sufficiently small  $T$ , one can get (23) to complete the proof of the lemma. □

**Lemma 2** For the weights of the novel scheme (18), we have

$$\sum_{j=0}^{k+1} |d_j^{k+1}| \leq C_\alpha T^\alpha, \tag{25}$$

where  $C_\alpha$  only depends on  $\alpha$ .

*Proof* The idea of the proof is similar to that of Lemma 1, so it is omitted. □

**Theorem 3** Assume that  $y_j$  ( $j = 1, 2, \dots, k$ ) are the solutions of scheme (10). Then the presented scheme (10) is stable.

*Proof* Suppose that  $y_{k+1}$  and  $\tilde{y}_{k+1}$  are numerical solutions in (10) and the initial conditions are given by  $y_0^{(i)}$  and  $\tilde{y}_0^{(i)}$  respectively. We shall use mathematical induction. Assume that

$$|y_j - \tilde{y}_j| \leq C_{\alpha,T} \|y_0 - \tilde{y}_0\|_\infty \tag{26}$$

is true for ( $j = 0, 1, \dots, k$ ). We must prove that this also holds for  $j = k + 1$ . Note that, by assumptions of the given initial conditions, the induction basis ( $j = 0$ ) is presupposed.

$$\begin{aligned}
 |y_{k+1} - \tilde{y}_{k+1}| & \leq \sum_{i=0}^{[\alpha]-1} \frac{t_{k+1}^i}{i!} |y_0^{(i)} - \tilde{y}_0^{(i)}| + \frac{1}{\Gamma(\alpha)} \left( \sum_{j=0}^k |d_j^{k+1}| |f(t_j, y_j) - f(t_j, \tilde{y}_j)| \right. \\
 & \quad \left. + |d_{k+1}^{k+1}| |f(t_{k+1}, y_{k+1}) - f(t_{k+1}, \tilde{y}_{k+1})| \right) \\
 & \leq C_1 \|y_0 - \tilde{y}_0\|_\infty + \frac{|\beta|}{\Gamma(\alpha)} \left( \sum_{j=0}^k |d_j^{k+1}| |y_j - \tilde{y}_j| + |d_{k+1}^{k+1}| |y_{k+1} - \tilde{y}_{k+1}| \right).
 \end{aligned}$$

Utilizing Lemma 1, one can get

$$|y_{k+1} - \tilde{y}_{k+1}| \leq C_1 \|y_0 - \tilde{y}_0\|_\infty + \frac{|\beta|}{\Gamma(\alpha)} \left( C_{1,\alpha} T^\alpha \max_{0 \leq j \leq k} |y_j - \tilde{y}_j| + C_{k+1,\alpha} T^\alpha |y_{k+1} - \tilde{y}_{k+1}| \right)$$

$$\leq \frac{1}{1 - (|\beta|C_{k+1,\alpha}T^\alpha)/\Gamma(\alpha)} \left( C_1 \|y_0 - \tilde{y}_0\|_\infty + \frac{|\beta|}{\Gamma(\alpha)} C_{1,\alpha} T^\alpha \max_{0 \leq j \leq k} |y_j - \tilde{y}_j| \right).$$

Now, for sufficiently small  $T$ , one can end the proof using mathematical induction (26) and by choosing constant  $C_{\alpha,T}$  sufficiently large. □

**Theorem 4** Assume that  $y_j$  ( $j = 1, 2, \dots, k$ ) are the solutions of algorithm II (13) and (18). Then the presented algorithm II is stable.

*Proof* The proof is similar to Theorem 3. □

### 5 Linear stability analysis

Following the ideas of [43, 44], consider the following test problem to investigate stability region of the presented methods:

$${}^C_0D_t^\alpha y(t) = \lambda y(t), \quad y(t_0) = y_0, \quad 0 < \alpha < 1, \tag{27}$$

where  $\lambda \in \mathbb{C}$  is a complex number. The new method (13) and (18) gives the following iteration formula for solving (27):

$$y_{k+1} = y_0 + \frac{1}{\Gamma(\alpha)} \sum_{i=0}^{k+1} \frac{h^\alpha}{12\alpha(\alpha + 1)(\alpha + 2)(\alpha + 3)(\alpha + 4)} \hat{b}_j^{k+1} \lambda y_j. \tag{28}$$

Denoting  $z = \lambda h^\alpha$ , we get

$$z = 12\Gamma(\alpha + 5) \frac{y_{k+1} - y_0}{\sum_{i=0}^{k+1} \hat{b}_j^{k+1} y_j}. \tag{29}$$

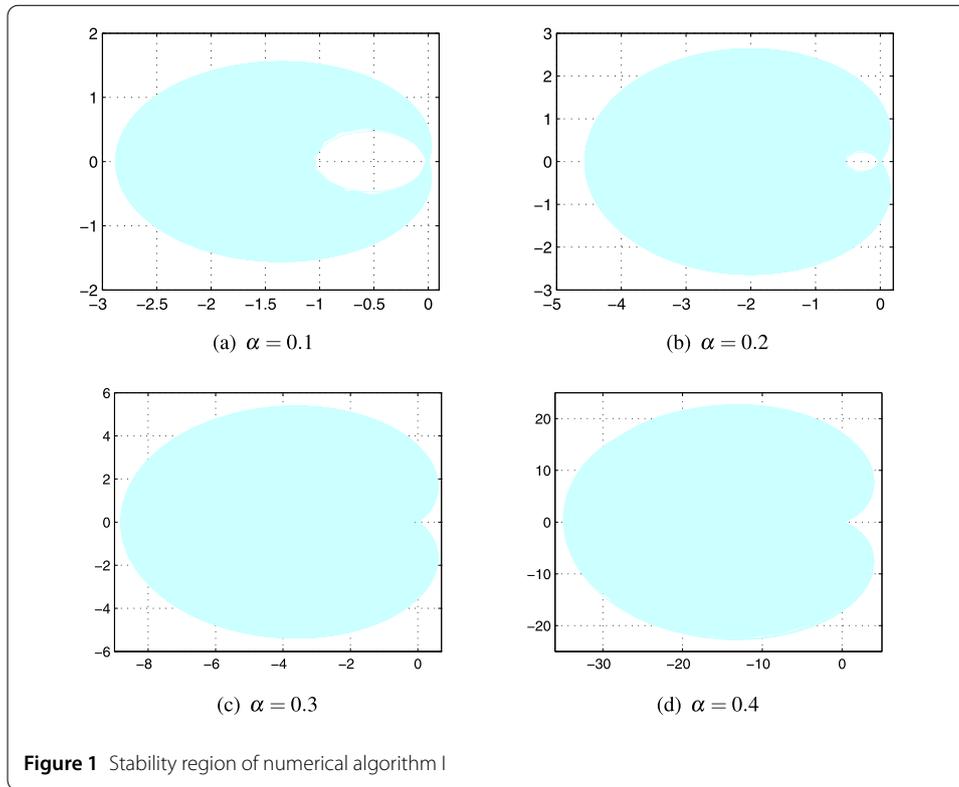
Let  $y_j = \xi^j$ , then by assuming  $\xi = e^{i\theta}$  with  $0 \leq \theta \leq 2\pi$  we get the following stability region for scheme (10):

$$S = \left\{ z : z = 12\Gamma(\alpha + 5) \frac{\xi^{k+1} - \xi^0}{\sum_{j=0}^{k+1} \hat{b}_{k+1-j}^{k+1} \xi^j} \right\}. \tag{30}$$

The stability region of algorithm I (10) can be achieved in a quite similar way. The stability region of numerical algorithm I is obtained in Figs. 1 and 2 by choosing  $k = 2000$ , and that of numerical algorithm II is shown in Figs. 3 and 4 by choosing  $k = 500$ . The stability region in Figs. 1 and 3 is inside of the boundary, and it is outside of the boundary in Figs. 2 and 4.

### 6 Numerical results

To check the numerical errors between the exact and the numerical solution, numerical experiments are carried out in this section. The presented examples have exact solutions and also have been solved by other numerical methods from literature. This allows one to compare the numerical results obtained by the presented schemes with the analytical solutions or those obtained by other methods. We also aim to graphically investigate the



**Figure 1** Stability region of numerical algorithm I

convergence order of the presented algorithms using the idea of [27] and [28]. The experimental orders of convergence (EOC) of the algorithms are measured by

$$EOC = \log_2 \left( \frac{E(h)}{E(h/2)} \right),$$

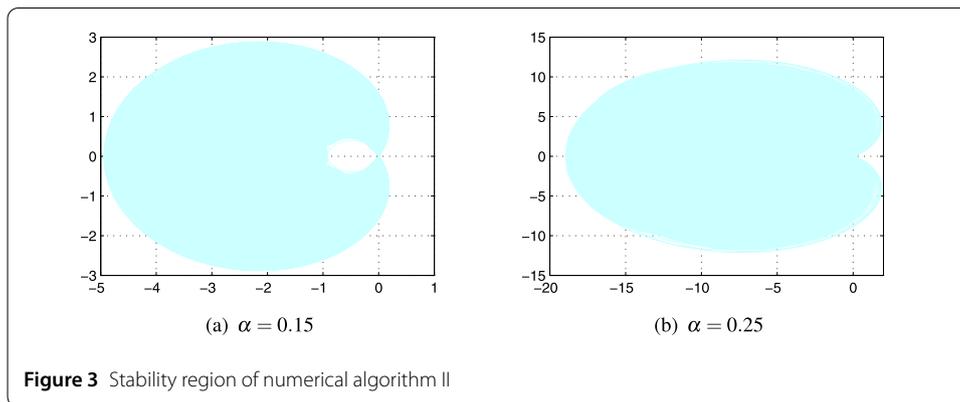
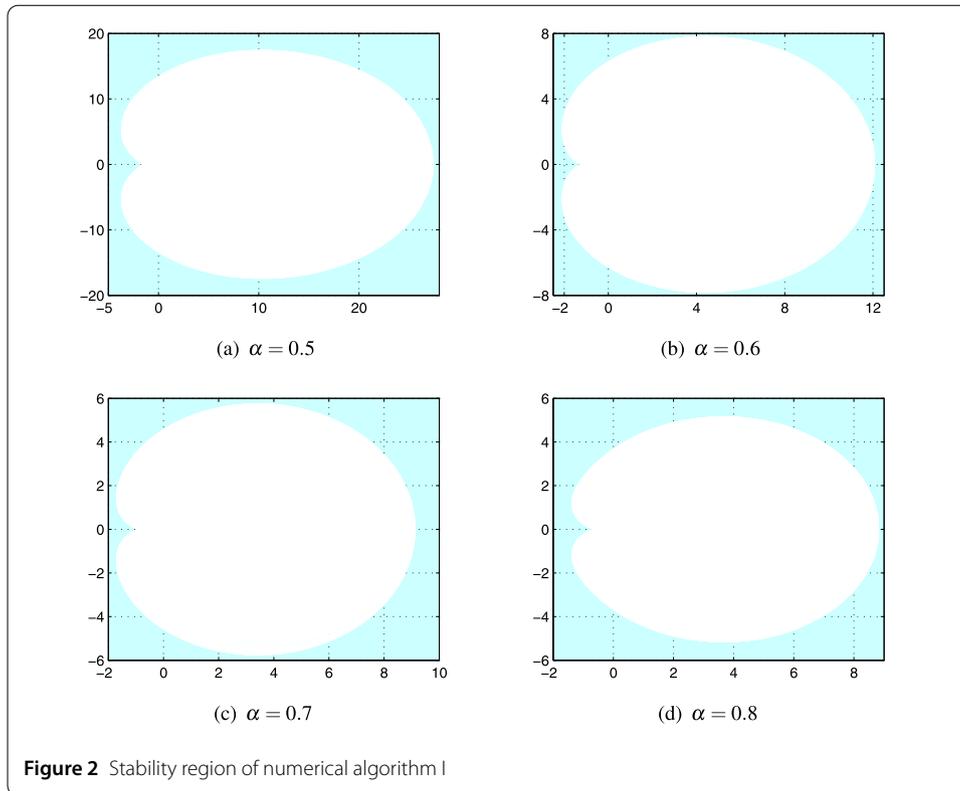
where  $E(h)$  is the absolute error  $|y(t_j) - y_j|$  at  $t = 1$  for the step size  $h$ . By the results of Theorems 1 and 2, we have

$$|E(h)| = |y(t_j) - y_j| \leq Ch^{\text{order}} \Rightarrow \log_2(|E(h)|) = \log_2(C) + (\text{order}) \times \log_2(h),$$

where (*order*) is 4 for the presented algorithm (10) and is 5 for the presented algorithm (13) and (18). Before proceeding, we first define some notations. Let  $Y = \log_2(|E(h)|)$  and  $X = \log_2(h)$ , also we use algorithm I to denote the presented algorithm (10) and algorithm II to denote the presented algorithm (13) and (18). We use  $E_I$  and  $E_{II}$  to denote the absolute error of algorithm I and algorithm II respectively. To graphically examine the order of convergence, we shall plot the straight line  $y = (\text{order}) \times x$  and also the function  $Y = Y(X)$ . The parallelism of these two lines can confirm the results of Theorems 1 and 2. Numerical computations are carried out with the help of MATLAB R2014a on an Intel(R) Core(TM)2 Duo CPU (2.67 GHz, 4 GB RAM) Windows 7 system.

*Example 1* Consider the following fractional differential equation:

$${}_0^C D_t^\alpha y(t) = \beta y(t) + g(t), \quad y(0) = 0, \tag{31}$$

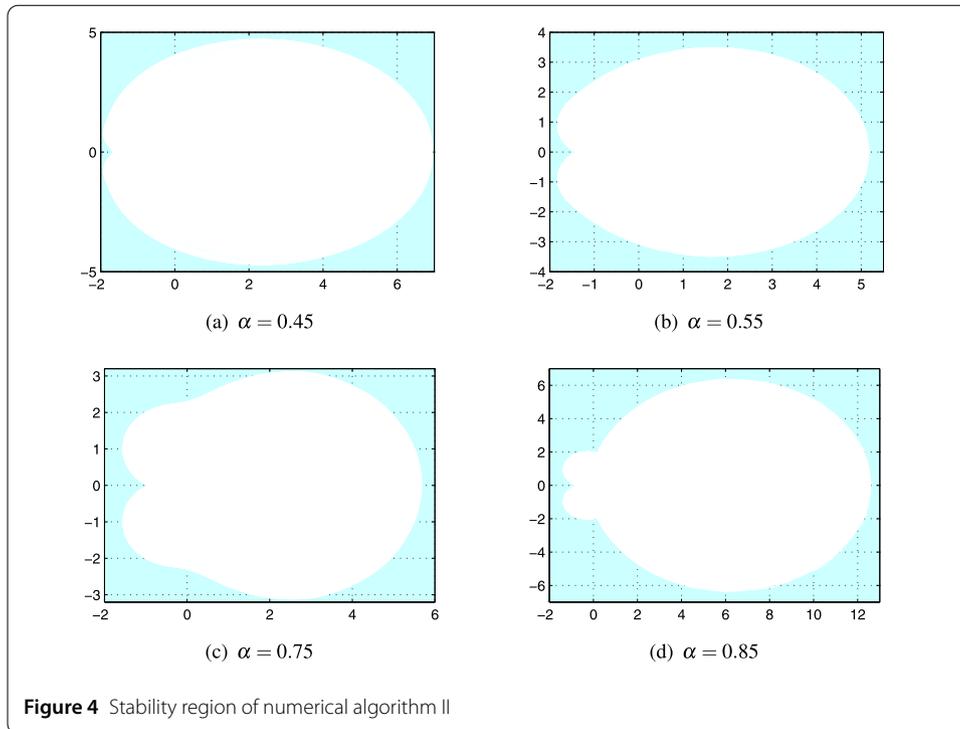


where

$$\beta = -1, g(t) = \frac{\Gamma(\nu + 1)}{\Gamma(\nu + 1 - \alpha)} t^{\nu - \alpha} + t^\nu, \quad \nu > 0.$$

The exact solution is  $y(t) = t^\nu$ . At the time  $t = 1$ , for different step sizes  $h$  and different  $\alpha$ , the approximate solutions for equation (31) are obtained for two values of  $\nu = 4$  and  $\nu = 3.45$  by using the presented algorithms.

First, we choose  $\nu = 4$ , which for this case  $y(t)$  is a sufficiently smooth function. The absolute errors of the presented algorithms and of the method reported in Ref. [45] are shown in Table 1. This table shows that novel schemes are valid methods in solving a fractional differential equation. Although the convergence ratios in Table 1 are a bit slower, they are not too far from our analytical results. This can be mainly due to some round-off



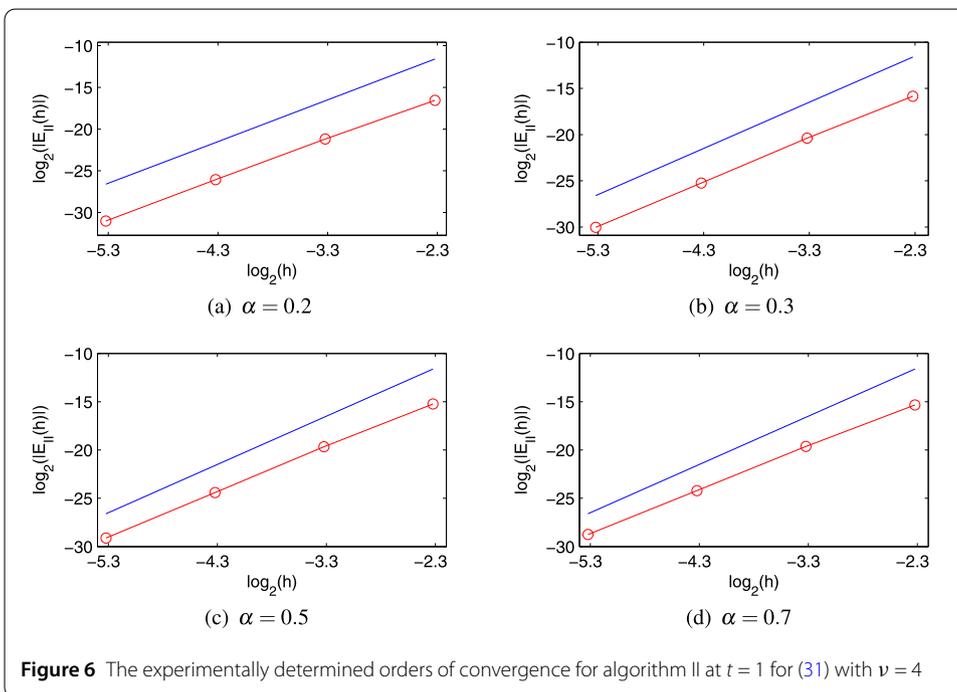
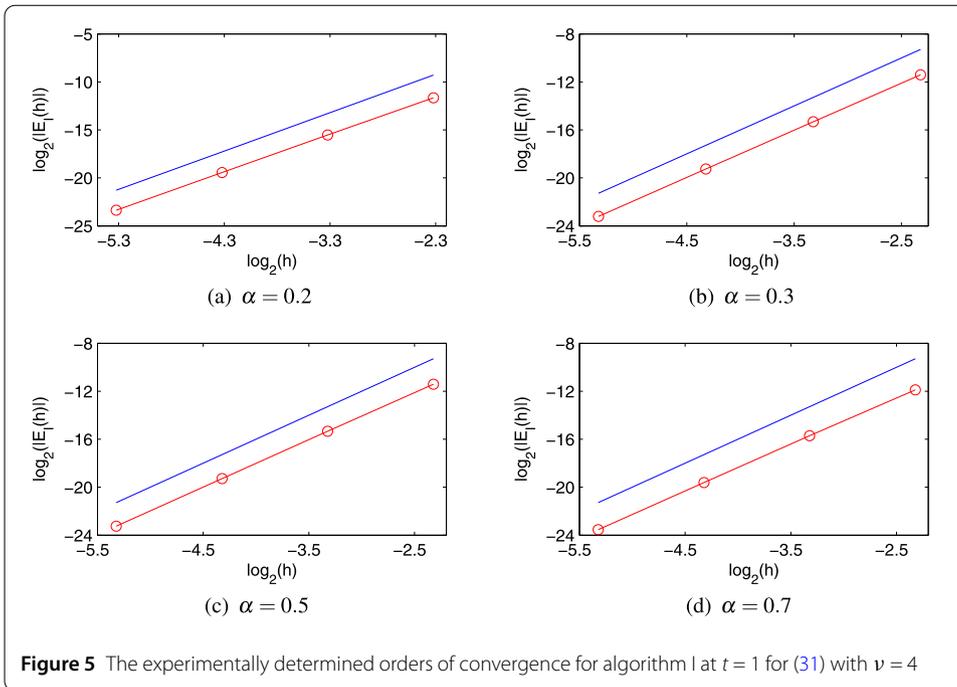
**Figure 4** Stability region of numerical algorithm II

**Table 1** The absolute errors of the presented algorithm I ( $E_I$ ), algorithm II ( $E_{II}$ ), and the numerical method of [45] for (31) with  $\alpha = 0.1, \nu = 4$

$h$	Algorithm II			Algorithm I			Method of [45]
	$E_{II}$	Ratio	CPU	$E_I$	Ratio	CPU	$E$
1/5	3.5631e-06	–	0.0021	1.9302e-04	–	0.0015	–
1/10	1.3548e-07	4.71	0.0028	1.3809e-05	3.81	0.0044	3.64e-01
1/20	4.7060e-09	4.85	0.0062	9.4607e-07	3.87	0.0067	1.70e-01
1/40	1.9210e-10	4.61	0.0199	6.3530e-08	3.90	0.0180	7.13e-02

errors during the computations of the algorithms. In Figs. 5–6, we plot the experimentally determined orders of convergence (EOC) for different values of  $\alpha$  at the time  $t = 1$ . Figure 5 shows the straight line  $y = 4x$  and the function  $Y = Y(X)$ , where  $E(h)$  is computed by algorithm I. One can see that these two lines are exactly parallel for different values of  $\alpha$ , which implies that the EOC of algorithm I indeed is  $O(h^4)$ . Figure 6 shows the straight line  $y = 5x$  and the function  $Y = Y(X)$ , where  $E(h)$  is computed by algorithm II. One can see that these two lines are parallel for different values of  $\alpha$ , which implies that the EOC of algorithm II indeed is  $O(h^5)$ .

Second, we choose  $\nu = 3.45$ , which for this case  $y(t) \in C^3[0, 1]$ , so it is not a sufficiently smooth function. This case enables us to check the efficiency of the presented algorithms when the solution is not sufficiently smooth, namely  $y(t) \notin C^4[0, 1]$  for algorithm I and  $y(t) \notin C^5[0, 1]$  for algorithm II. The absolute errors of the presented algorithms for  $\alpha = 0.2$  are presented in Table 2. From Table 2 we see that for this case the method has still high accuracy and has acceptable convergence ratios. Figures 7–8 show the EOC of the algorithms for different values of  $\alpha$  at the time  $t = 1$ . As explained in the later case ( $\nu = 4$ ), from Figs. 7–8, again we can confirm the results of Theorems 1 and 2.



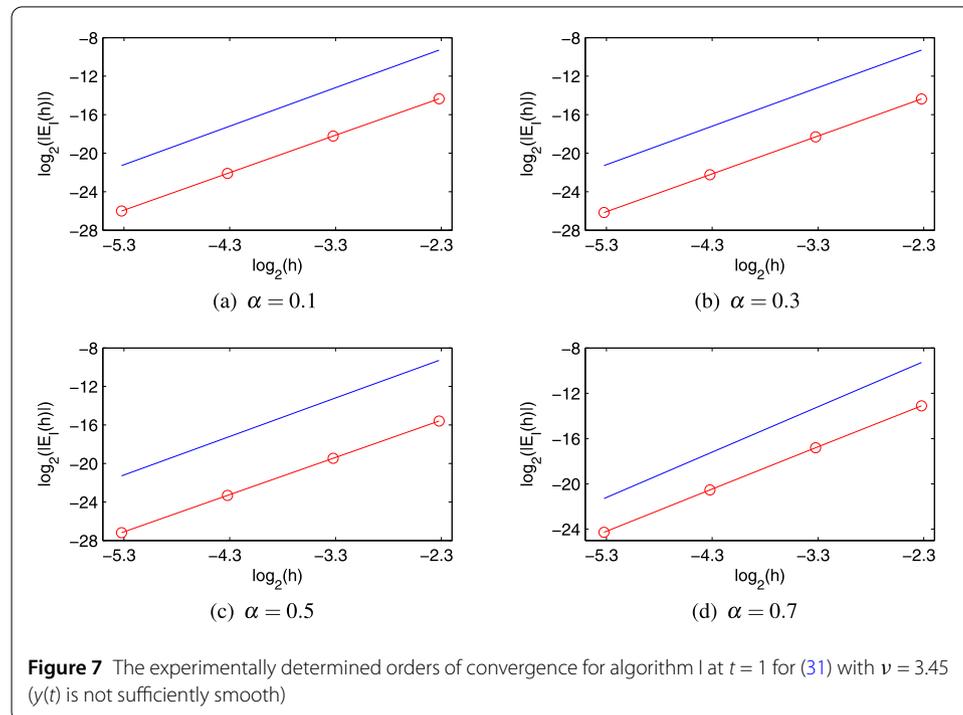
*Example 2* Consider the following fractional differential equation:

$${}_0^C D_t^\alpha y(t) = \frac{24}{\Gamma(5-\alpha)} t^{4-\alpha} - \frac{3}{\Gamma(4-\alpha)} t^{3-\alpha} - \frac{1}{2} t^3 - y(t) + t^4, \quad y(0) = 0. \tag{32}$$

The exact solution is  $y(t) = t^4 - \frac{1}{2} t^3$ . Table 3 shows the absolute errors of the presented schemes and the method reported in Ref. [46] at the time  $t = 1$ . From this table it is observed that the error of the presented method has decreased significantly. In Figs. 9–10,

**Table 2** The absolute errors of the presented algorithm I ( $E_I$ ) and algorithm II ( $E_{II}$ ) for (31) with  $\alpha = 0.2, \nu = 3.45$

$h$	Algorithm II			Algorithm I		
	$E_{II}$	Ration	CPU time(s)	$E_I$	Ration	CPU time(s)
1/5	6.8126e-06	–	0.0021	5.9853e-05	–	0.0012
1/10	2.7965e-07	4.61	0.0032	3.9525e-06	3.92	0.0021
1/20	9.9336e-09	4.82	0.0065	2.6190e-07	3.92	0.0051
1/40	3.3460e-10	4.89	0.0195	1.7182e-08	3.93	0.0171

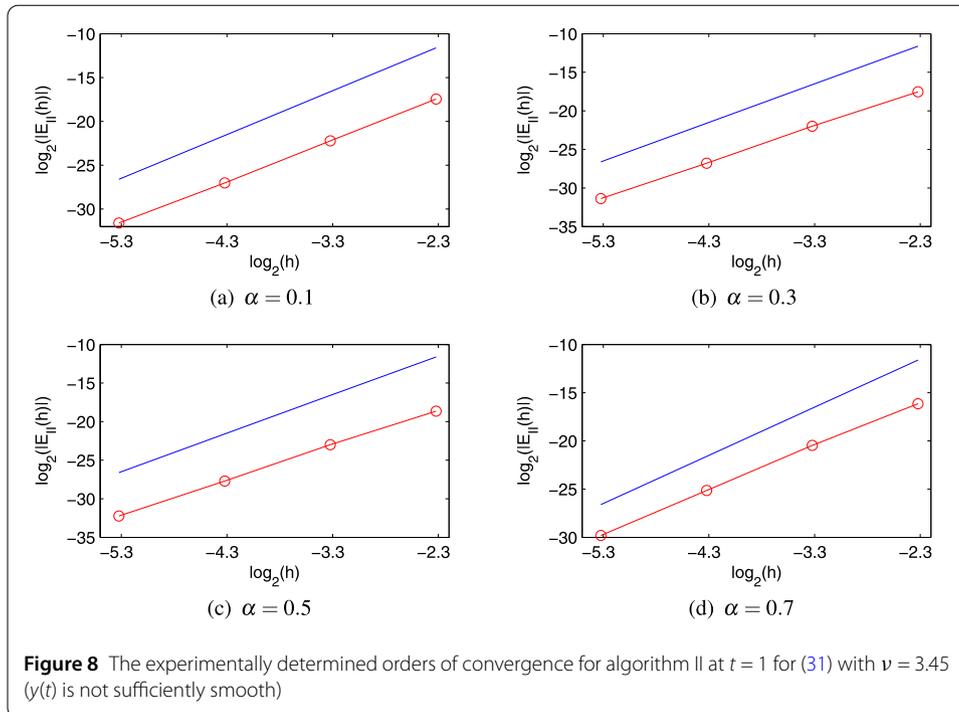


**Figure 7** The experimentally determined orders of convergence for algorithm I at  $t = 1$  for (31) with  $\nu = 3.45$  ( $y(t)$  is not sufficiently smooth)

we plot the EOC for different values of  $\alpha$  at the time  $t = 1$ . Figure 9 examines the EOC of algorithm I. The details of this figure are explained in Example 1 (Figure 5). We can see that the lines  $y = 4x$  and  $Y = Y(X)$  are exactly parallel, which verifies the results of Theorem 1. Figure 10 examines the EOC of algorithm II. The details of this figure are explained in Example 1 (Figure 6). We can see that the lines  $y = 5x$  and  $Y = Y(X)$  are parallel, which verifies the results of Theorem 2.

**Example 3** In this example we aim to examine the validity and efficiency of the presented methods when they are applied to solve a fractional order system of differential equations. Consider the following initial value problem:

$$\begin{cases} {}^C_0D_t^\alpha x(t) = -x(t), \\ {}^C_0D_t^\alpha y(t) = 2x(t) + y(t) - 9z(t), \\ {}^C_0D_t^\alpha z(t) = 3x(t) + 6y(t) + z(t). \end{cases} \quad (33)$$



The exact solution of this system is [47]

$$\begin{aligned}
 \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} &= c_1 \begin{pmatrix} 58 \\ -31 \\ 6 \end{pmatrix} E_\alpha(-t^\alpha) \\
 &+ \frac{c_2}{2} \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} (E_\alpha((1 + 3\sqrt{6})t^\alpha) + E_\alpha((1 - 3\sqrt{6})t^\alpha)) \\
 &- \frac{c_2}{2i} \begin{pmatrix} 0 \\ \sqrt{6} \\ 0 \end{pmatrix} (E_\alpha((1 + 3\sqrt{6})t^\alpha) - E_\alpha((1 - 3\sqrt{6})t^\alpha)) \\
 &+ \frac{c_3}{2} \begin{pmatrix} 0 \\ \sqrt{6} \\ 0 \end{pmatrix} (E_\alpha((1 + 3\sqrt{6})t^\alpha) + E_\alpha((1 - 3\sqrt{6})t^\alpha)) \\
 &+ \frac{c_3}{2i} \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} (E_\alpha((1 + 3\sqrt{6})t^\alpha) - E_\alpha((1 - 3\sqrt{6})t^\alpha)).
 \end{aligned}$$

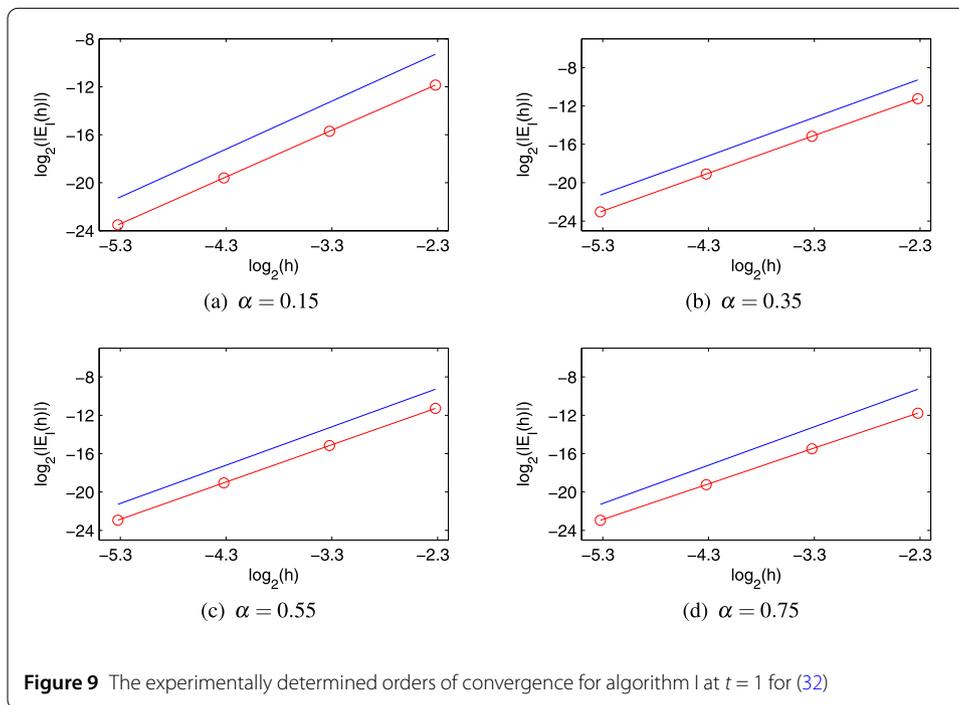
Under the initial conditions  $(x(0), y(0), z(0)) = (-3, 5, 0)$ , we have

$$(c_1, c_2, c_3) = (-0.05172, 0.15517, 1.38664).$$

We solve system (33) using the presented algorithms numerically. Figure 11 shows the difference between the curves of exact and approximate solutions for  $\alpha = 0.925$  with  $h =$

**Table 3** Absolute errors of the present presented algorithm I ( $E_I$ ), algorithm II ( $E_{II}$ ), and the numerical methods of [46], with  $\alpha = 0.3$  for (32)

$h$	Algorithm II			Algorithm I			Method of [46]
	$E_{II}$	Ratio	CPU	$E_I$	Ratio	CPU	$E$
1/5	2.0268e-05	–	0.0020	3.9338e-04	–	0.0013	–
1/10	8.8773e-07	4.51	0.0029	2.6193e-05	3.91	0.0022	1.4571e-04
1/20	3.0045e-08	4.88	0.0064	1.7205e-06	3.93	0.0057	2.3118e-05
1/40	1.0533e-09	4.83	0.0205	1.1167e-07	3.95	0.0176	3.6127e-06

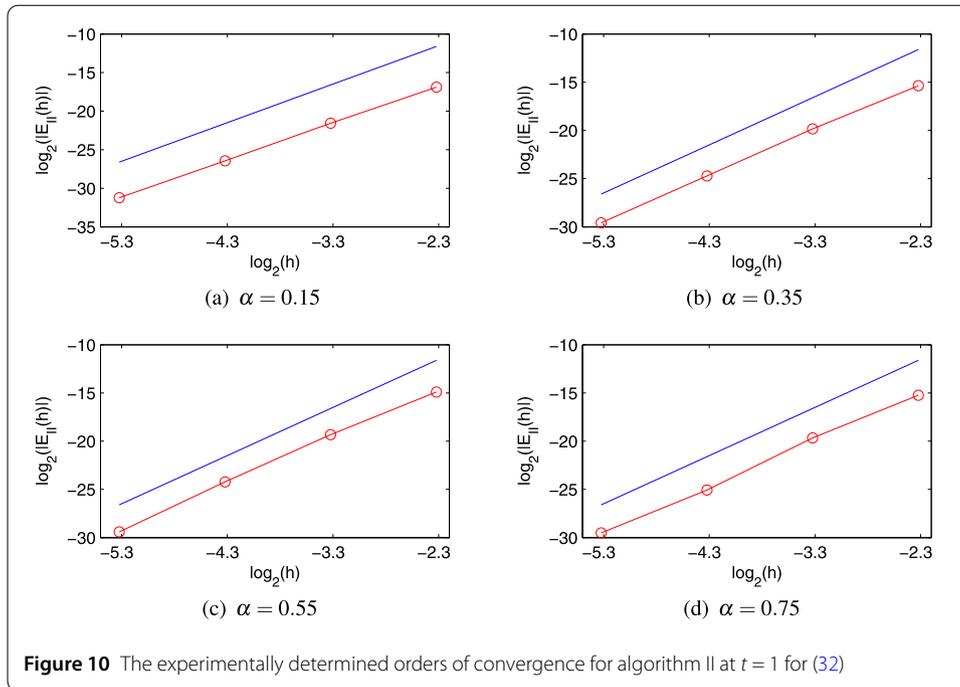


**Figure 9** The experimentally determined orders of convergence for algorithm I at  $t = 1$  for (32)

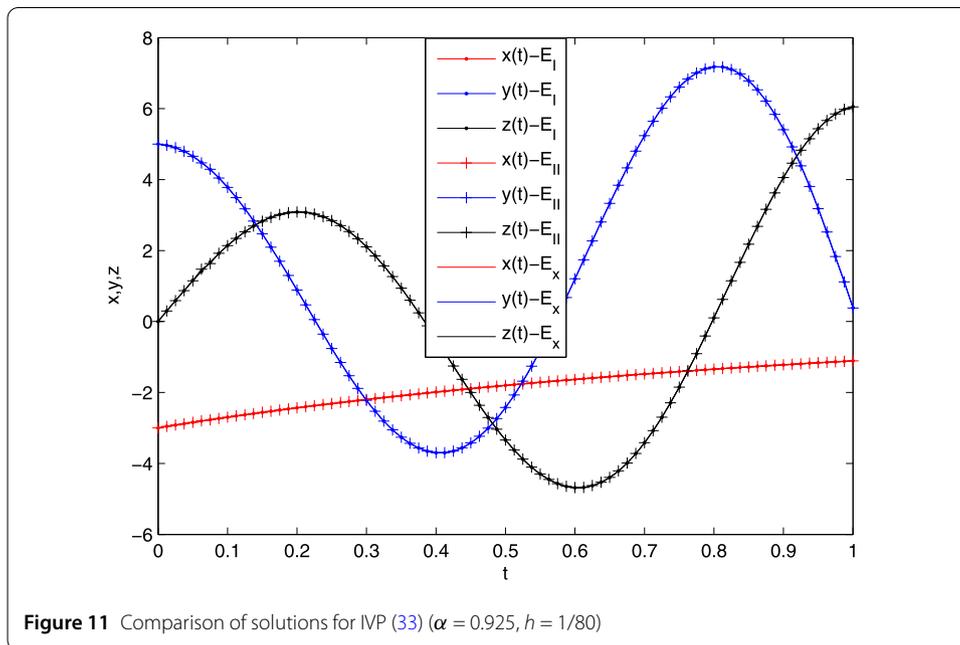
1/80. It is observed that the numerical solutions obtained by algorithm I and algorithm II are in excellent agreement with the exact solution. Figure 12(a) shows the exact and approximate solutions of system (33) obtained by algorithm I for  $t = 50$  with  $\alpha = 0.95$ ,  $h = 1/40$ . Figure 12(b) shows the exact and approximate solutions of system (33) obtained by algorithm II for  $t = 50$  with  $\alpha = 0.975$ ,  $h = 1/40$ . From Fig. 12 it is observed that the numerical solutions are at very good agreement with the exact solution and the presented algorithms are still valid and capable of solving problems in a large scale computational domain.

**7 Conclusion**

This paper provides two high order numerical schemes with theoretically proved convergence order of 4 and 5 for solving FDEs. The properties of the Caputo derivative are used to reduce FDEs into a Volterra integral equation. After dividing the total domain into a set of grid points, the piecewise Lagrange interpolation polynomials of degree three and degree four are utilized to approximate unknown functions. The stability and error estimate of the methods are investigated. Moreover, graphical illustrations for the stability region of the schemes are derived. In the numerical results, the accuracy and convergence orders

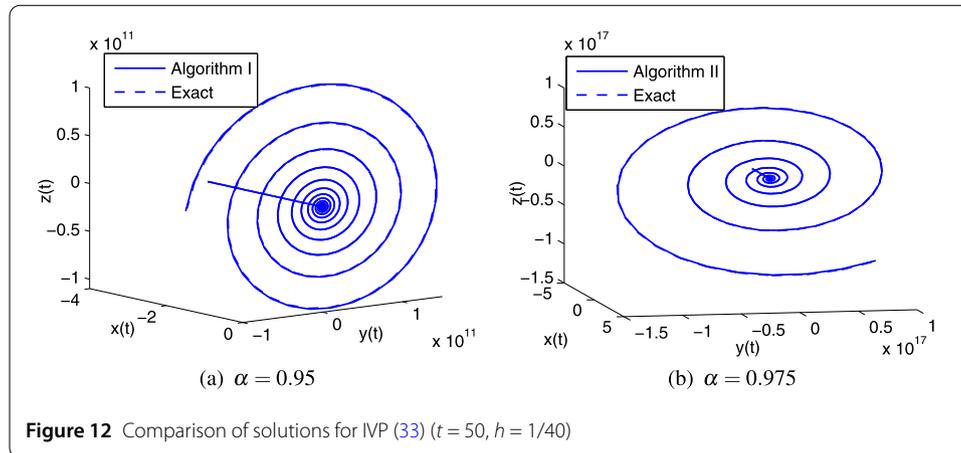


**Figure 10** The experimentally determined orders of convergence for algorithm II at  $t = 1$  for (32)



**Figure 11** Comparison of solutions for IVP (33) ( $\alpha = 0.925, h = 1/80$ )

of the developed algorithms have been studied. Numerical results confirm the theoretical results of the algorithms. Furthermore, we extended the presented algorithms to solve a 3D fractional order system of equations. It is shown that the algorithms are valid and efficient for solving a linear fractional system, and it is observed that the algorithms are capable of solving large domain problems. In the future, we shall try to follow this idea to construct higher order schemes for solving nonlinear FDEs.



### Acknowledgements

Not applicable.

### Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

### Availability of data and materials

All the data and materials of this paper are available.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

The authors have equally made contributions to this paper. All authors read and approved the final manuscript.

### Author details

<sup>1</sup>Department of Mathematics, University of Tabriz, Tabriz, Iran. <sup>2</sup>Department of Mathematics, University of Chester, CH1 4BJ, Chester, UK.

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 23 September 2020 Accepted: 2 February 2021 Published online: 17 February 2021

### References

- Simpson, R., Jaques, A., Nuñez, H., Ramirez, C., Almonacid, A.: Fractional calculus as a mathematical tool to improve the modeling of mass transfer phenomena in food processing. *Food Eng. Rev.* **5**(1), 45–55 (2013)
- Gepreel, K.A., Nofal, T.A.: Optimal homotopy analysis method for nonlinear partial fractional differential equations. *Math. Sci.* **9**(1), 47–55 (2015)
- Herzallah, M.A., Gepreel, K.A.: Approximate solution to the time–space fractional cubic nonlinear Schrödinger equation. *Appl. Math. Model.* **36**(11), 5678–5685 (2012)
- Gepreel, K.A., Nofal, T.A., Alotaibi, F.M.: Numerical solutions for the time and space fractional nonlinear partial differential equations. *J. Appl. Math.* **2013**, Article ID 482419 (2013)
- Roohi, M., Aghababa, M.P., Haghghi, A.R.: Switching adaptive controllers to control fractional-order complex systems with unknown structure and input nonlinearities. *Complexity* **21**(2), 211–223 (2015)
- Roohi, M., Zhang, C., Chen, Y.: Adaptive model-free synchronization of different fractional-order neural networks with an application in cryptography. *Nonlinear Dyn.* **100**, 3979–4001 (2020)
- Ahmad, B., Ntouyas, S.K., Alsaedi, A.: A study of nonlinear fractional differential equations of arbitrary order with Riemann–Liouville type multistrip boundary conditions. *Math. Probl. Eng.* **2013**, Article ID 320415 (2013)
- Naik, P.A., Zu, J., Owolabi, K.M.: Global dynamics of a fractional order model for the transmission of HIV epidemic with optimal control. *Chaos Solitons Fractals* **138**, 109826 (2020)
- Naik, P.A., Owolabi, K.M., Yavuz, M., Zu, J.: Chaotic dynamics of a fractional order HIV-1 model involving AIDS-related cancer cells. *Chaos Solitons Fractals* **140**, 110272 (2020)
- Guo, C., Fang, S.: Stability and approximate analytic solutions of the fractional Lotka–Volterra equations for three competitors. *Adv. Differ. Equ.* **2016**(1), 219 (2016)
- Asl, M.S., Javidi, M.: A new numerical method for solving system of FDEs: applied in plankton system. *Dyn. Contin. Discrete Impuls. Syst., Ser. B, Appl. Algorithms* **26**(4), 221–242 (2019)
- Arafa, A., Rida, S., Khalil, M.: A fractional-order model of HIV infection with drug therapy effect. *J. Egypt. Math. Soc.* **22**(3), 538–543 (2014)
- Kuroda, L.K.B., Gomes, A.V., Tavoni, R., de Arruda Mancera, P.F., Varalta, N., de Figueiredo Camargo, R.: Unexpected behavior of Caputo fractional derivative. *Comput. Appl. Math.* **36**, 1173–1183 (2017)

14. Rihan, F.A.: Numerical Modeling of Fractional-Order Biological Systems. Abstract and Applied Analysis, vol. 2013. Hindawi Publishing Corporation, New York (2013)
15. Rihan, F., Lakshmanan, S., Hashish, A., Rakkiyappan, R., Ahmed, E.: Fractional-order delayed predator–prey systems with Holling type-II functional response. *Nonlinear Dyn.* **80**(1–2), 777–789 (2015)
16. Owolabi, K.M.: Behavioural study of symbiosis dynamics via the Caputo and Atangana–Baleanu fractional derivatives. *Chaos Solitons Fractals* **122**, 89–101 (2019)
17. Owolabi, K.M., Karaagac, B.: Dynamics of multi-pulse splitting process in one-dimensional Gray–Scott system with fractional order operator. *Chaos Solitons Fractals* **136**, 109835 (2020)
18. Damor, R., Kumar, S., Shukla, A.: Numerical simulation of fractional bioheat equation in hyperthermia treatment. *J. Mech. Med. Biol.* **14**(02), 1450018 (2014)
19. Kumar, K., Pandey, R.K., Sharma, S.: Comparative study of three numerical schemes for fractional integro-differential equations. *J. Comput. Appl. Math.* **315**, 287–302 (2017)
20. Owolabi, K.M.: Robust and adaptive techniques for numerical simulation of nonlinear partial differential equations of fractional order. *Commun. Nonlinear Sci. Numer. Simul.* **44**, 304–317 (2017)
21. Bushnaq, S., Momani, S., Zhou, Y.: A reproducing kernel Hilbert space method for solving integro-differential equations of fractional order. *J. Optim. Theory Appl.* **156**(1), 96–105 (2013)
22. Chakraverty, S., Behera, D.: Dynamic responses of fractionally damped mechanical system using homotopy perturbation method. *Alex. Eng. J.* **52**(3), 557–562 (2013)
23. Odibat, Z.M., Momani, S.: An algorithm for the numerical solution of differential equations of fractional order. *J. Appl. Math. Inform.* **26**, 15–27 (2008)
24. Kumar, P., Agrawal, O.P.: An approximate method for numerical solution of fractional differential equations. *Signal Process.* **86**(10), 2602–2610 (2006)
25. Cao, J., Xu, C.: A high order schema for the numerical solution of the fractional ordinary differential equations. *J. Comput. Phys.* **238**, 154–168 (2013)
26. Diethelm, K.: An algorithm for the numerical solution of differential equations of fractional order. *Electron. Trans. Numer. Anal.* **5**(1), 1–6 (1997)
27. Yan, Y., Pal, K., Ford, N.J.: Higher order numerical methods for solving fractional differential equations. *BIT Numer. Math.* **54**(2), 555–584 (2014)
28. Asl, M.S., Javidi, M., Yan, Y.: A novel high-order algorithm for the numerical estimation of fractional differential equations. *J. Comput. Appl. Math.* **342**, 180–201 (2018)
29. Dehghan, M., Abbaszadeh, M.: Two meshless procedures: moving Kriging interpolation and element-free Galerkin for fractional PDEs. *Appl. Anal.* **96**(6), 936–969 (2017)
30. Abbaszadeh, M., Dehghan, M.: The Crank–Nicolson/interpolating stabilized element-free Galerkin method to investigate the fractional Galilei invariant advection-diffusion equation. *Math. Methods Appl. Sci.* **44**(4), 2752–2768 (2021)
31. Abbaszadeh, M., Dehghan, M.: Numerical and analytical investigations for solving the inverse tempered fractional diffusion equation via interpolating element-free Galerkin (IEFG) method. *J. Therm. Anal. Calorim.*, 1–17 (2020)
32. Dehghan, M., Manafian, J., Saadatmandi, A.: Solving nonlinear fractional partial differential equations using the homotopy analysis method. *Numer. Methods Partial Differ. Equ.* **26**(2), 448–479 (2010)
33. Dehghan, M., Abbaszadeh, M., Deng, W.: Fourth-order numerical method for the space–time tempered fractional diffusion-wave equation. *Appl. Math. Lett.* **73**, 120–127 (2017)
34. Dehghan, M., Abbaszadeh, M.: Spectral element technique for nonlinear fractional evolution equation, stability and convergence analysis. *Appl. Numer. Math.* **119**, 51–66 (2017)
35. Manafian, J., Lakestani, M.: A new analytical approach to solve some of the fractional-order partial differential equations. *Indian J. Phys.* **91**(3), 243–258 (2017)
36. Saadatmandi, A., Dehghan, M.: A new operational matrix for solving fractional-order differential equations. *Comput. Math. Appl.* **59**(3), 1326–1336 (2010)
37. Owolabi, K.M., Karaagac, B.: Chaotic and spatiotemporal oscillations in fractional reaction-diffusion system. *Chaos Solitons Fractals* **141**, 110302 (2020)
38. Owolabi, K.M.: Numerical simulation of fractional-order reaction–diffusion equations with the Riesz and Caputo derivatives. *Neural Comput. Appl.* **32**(8), 4093–4104 (2020)
39. Owolabi, K.M., Shikongo, A.: Fractional operator method on a multi-mutation and intrinsic resistance model. *Alex. Eng. J.* **2020**, 1999–2013 (2020)
40. Asl, M.S., Javidi, M.: Novel algorithms to estimate nonlinear FDEs: applied to fractional order nutrient-phytoplankton–zooplankton system. *J. Comput. Appl. Math.* **339**, 193–207 (2018)
41. Asl, M.S., Javidi, M., Ahmad, B.: New predictor-corrector approach for nonlinear fractional differential equations: error analysis and stability. *J. Appl. Anal. Comput.* **9**(4), 1527–1557 (2019)
42. Li, C., Chen, A., Ye, J.: Numerical approaches to fractional calculus and fractional ordinary differential equation. *J. Comput. Phys.* **230**(9), 3352–3368 (2011)
43. Nwajeri, K.U.: Stability regions of numerical methods for solving fractional differential equations. PhD thesis, University of Chester (2012)
44. Garrappa, R.: On linear stability of predictor–corrector algorithms for fractional differential equations. *Int. J. Comput. Math.* **87**(10), 2281–2290 (2010)
45. Deng, W., Li, C.: Numerical schemes for fractional ordinary differential equations. In: Miidla, P. (ed.) *Numerical Modelling*, pp. 355–374 (2012)
46. Pal, K., Liu, F., Yan, Y.: Numerical solutions of fractional differential equations by extrapolation. In: Dimov, I., Faragó, I., Vulkov, L. (eds.) *Finite Difference Methods, Theory and Applications*, pp. 299–306 (2015)
47. Odibat, Z.M.: Analytical study on linear systems of fractional differential equations. *Comput. Math. Appl.* **59**(3), 1171–1183 (2010)