Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(1): 804–820.
DOI:10.3934/math.2021048
Received: 17 August 2021
Accepted: 06 October 2021
Published: 18 October 2021

Research article

A collocation methods based on the quadratic quadrature technique for fractional differential equations

Sunyoung Bu*

Department of Liberal arts, Hongik University, Sejong, 30016, South Korea

* **Correspondence:** Email: syboo@hongik.ac.kr; Tel: +82448602121; Fax: +82448632648.

Abstract: In this paper, we introduce a mixed numerical technique for solving fractional differential equations (FDEs) by combining Chebyshev collocation methods and a piecewise quadratic quadrature rule. For getting solutions at each integration step, the fractional integration is calculated in two intervals-all previous time intervals and the current time integration step. The solution at the current integration step is calculated by using Chebyshev interpolating polynomials. To remove a singularity which belongs originally to the FDEs, Lagrangian interpolating technique is considered since the Chebyshev interpolating polynomial can be rewritten as a Lagrangian interpolating form. Moreover, for calculating the fractional integral on the whole previous time intervals, a piecewise quadratic quadrature technique is applied to get higher accuracy. Several numerical experiments demonstrate the efficiency of the proposed method and show numerically convergence orders for both linear and nonlinear cases.

Keywords: fractional differential equations; quadratic quadrature; interpolation; Chebyshev collocation; Lagrangian interpolation
Mathematics Subject Classification: 34A08, 65L05, 65L20

1. Introduction

In this paper, we discuss the numerical solution of the fractional differential equations (FDEs) initial value problem

$$D_*^{\alpha} y(t) = f(t, y(t)), \quad 0 \le t \le T, \tag{1.1}$$

$$y^{(j)}(0) = y_0^{(j)}, \quad j = 0, 1, \cdots, \lceil \alpha \rceil - 1, \tag{1.2}$$

where D_*^{α} is fractional Caputo derivative of order α and defined as

$$D_*^{\alpha} y(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-\tau)^{n-\alpha-1} y^{(n)}(\tau) d\tau, \tag{1.3}$$

where $n = \lceil \alpha \rceil$ is the smallest integer greater than or equal to α and the right hand side function $f(t, y)$ is assumed to be continuous with respect to two variables t and y. Here, we note that there are several definitions of fractional derivatives such as Caputo, Grunwald-Letnikov, Marchaud, Riemann-Liouville, Weyl, etc. In this study, we just focus on the Caputo type of fractional derivative.

Fractional calculus is a research topic in many areas of science and engineering, such as signal processing, control engineering, electromagnetism, bioscience, fluid mechanics, electrochemistry, diffusion processes, continuum and statistical mechanics and propagation of spherical flames. Due to this reason, during the past few decades, mathematical theories and numerical analysis of fractional differential equations have received lots of attention and several numerical methods have been developed. For example, Deithem et al. [8, 11] introduced the theory and numerical schemes for the predictor-corrector type of Adams methods. In [15], a non-polynomial collocation method was proposed for fractional equations with having non-smooth solutions. Yan et al. [30] developed a higher order predictor-corrector methods using quadratic quadrature techniques based on fractional Adam-type methods [11] for both linear and nonlinear cases. In [28], authors proposed the usage of a suitable truncated series expressed in terms of fractional powers of the independent variable for ordinary fractional differential equation. Besides, there are relevant researches [8–10, 13, 14, 22, 23, 25–28, 32].

The main contribution of this paper is, unlike other numerical schemes described above, to split two different time intervals-one is the sum of the whole previous time interval and the other one is the current time interval. Then, different numerical schemes are provided to the two time intervals to obtain desired solutions at the current time interval-piecewise quadratic quadrature techniques are applied to the whole previous time intervals and Lagrangian collocation methods are cast for the current time step. There are few attempts to use the mixed numerical schemes. For example, in [31], a new product integration scheme is introduced by using the idea of local Fourier expansion and several types of quadrature rules. Inspired by this idea, in this paper, a new version of the mixed numerical scheme using the piecewise quadratic quadrature techniques and the Lagrangian collocation methods are introduced.

The procedure of this mixed numerical method can be described as follows: Since the IVP (1.1) is equivalent to the following Volterra integral equation of the second kind, Eq (1.1) is rewritten as the Volterra integral equation :

$$y(t) = \sum_{k=0}^{\lceil \alpha \rceil - 1} y_0^{(k)} \frac{t^k}{k!} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau)) d\tau. \tag{1.4}$$

First of all, a given time interval $[t_0, t_{final}]$ is divided into several sub-intervals. In each sub-interval $[t_i, t_{i+1}]$, we estimate new solutions by using solutions calculated in all previous intervals $[t_0, t_i]$. However, it is not easy to calculate the integral equations directly because FDEs originally have a singularity at the end point of integral interval.

To hurdle this drawback, the Lagrangian interpolating formulation is applied so that the beta function property can be used for the fractional integration, and it enables to eliminate the singularity by removing the fractional integral. For this, we introduce Chebyshev node points in the sub-interval $[t_i, t_{i+1}]$ and calculate solutions at the collocation points. Moreover, solutions at all collocation points through the whole previous intervals are accumulated. Based on all these accumulated solutions, we can easily calculate the fractional integral by using any quadrature rule in the whole previous time interval $[t_0, t_i]$. Here, for the higher accuracy, a piecewise quadratic quadrature rule is introduced.

Note that it has been already known [6, 20] that multi-stage methods such as collocation methods have lots of good properties in the sense of stability and accuracy, compared with multi-step methods.

This paper is organized as follows. In Section 2, we briefly review the basic background such as Lagrangian interpolations. In Section 3, a Lagrangian interpolation technique is applied to calculate the fractional integral in a certain time sub-interval. Moreover, a piecewise quadratic interpolation polynomial is introduced to approximate an integral with all known solutions calculated in all previous time intervals. In Section 4, to examine the effectiveness and efficiency of the proposed scheme, several numerical results are presented and shown numerically a convergence order of the propose method. Finally in Section 5, we summarize our results and discuss possibilities to increase the efficiency of the propose scheme for solving other types of fractional differential equations, such as multi-term FDEs or fractional partial differential equations, etc.

2. Preliminaries

In this section, we briefly review the numerical techniques required to develop the numerical methods to solve the FDE initial value problems (IVPs).

2.1. Lagrangian scheme

Suppose that $s_0 < s_1 < \cdots < s_n$ are $n+1$ Chebyshev-Gauss-Lobatto (CGL) node points on $[0, 1]$, where $s_0 = 0$ and $s_n = 1$. We usually discretize the whole time interval into several sub-intervals to solve IVP (1.1) and solve the IVP in each sub-interval. Suppose y_m is approximated for the exact value $y(t_m)$ at time t_m. Based on the approximated solutions y_m, we need to approximate the solution at time t_{m+1}. At first, $n+1$ CGL-points on $[t_m, t_{m+1}]$ are required through the following linear transformation

$$t_{m,k} = t_m + hs_k,$$

for $k = 0, 1, \cdots, n$, where $h = t_{m+1} - t_m$. Once the numerical solutions $y_{m,1}, y_{m,2}, ..., y_{m,n}$ at the nodes $t_{m,1}, t_{m,2}, ..., t_{m,n}$ are obtained, we can write the solution $y(t)$ and the function function $f(t, y(t))$ in Lagrange interpolation form as follows:

$$y(t) = \sum_{k=0}^{n} y_{m,k} L_k(t), \tag{2.1}$$

$$f(t, y(t)) = \sum_{k=0}^{n} f_{m,k} L_k(t), \tag{2.2}$$

where $L_k(t)$ us the Lagrange interpolation polynomial of order n, given as

$$L_k(t) = \prod_{i=0, i\neq k}^{n} \frac{t - t_{m,i}}{t_{m,k} - t_{m,i}},$$

$$f_k(t) = f(t_{m,k}, y(t_{m,k})).$$

Also, $L_k(t)$ should be transformed into following expressions :

$$L_k(t) = \sum_{j=0}^{n} c_{j,k} t^j, \quad k = 0, 1, \cdots, n,$$

where the coefficients $c_{j,k}$ can be computed by it matrix form,

$$\begin{pmatrix} t_{m,0}^0 & t_{m,0}^1 & \cdots & t_{m,0}^n \\ t_{m,1}^0 & t_{m,1}^1 & \cdots & t_{m,1}^n \\ \cdots & \cdots & \cdots & \cdots \\ t_{m,n}^0 & t_{m,n}^1 & \cdots & t_{m,n}^n \end{pmatrix} \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n,0} & c_{n,1} & \cdots & c_{n,n} \end{pmatrix} = I_{n+1}, \tag{2.3}$$

where I_{n+1} is the identity matrix of order $n+1$. Moreover, the fractional integral containing $L_k(t)$ can be easily rewritten by using the Beta function property as follows:

$$\begin{aligned} \frac{1}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1}L_k(\tau)d\tau &= \frac{1}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1}\sum_{j=0}^{n} c_{j,k}\tau^j d\tau, \\ &= \sum_{j=0}^{n} c_{j,k}\frac{1}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1}\tau^j d\tau, \\ &= \sum_{j=0}^{n} c_{j,k}\frac{\Gamma(j+1)}{\Gamma(j+1+\alpha)}t^{j+\alpha}. \end{aligned} \tag{2.4}$$

As seen above, it turns out that the Lagrangian polynomial can remove the fractional integral, so the singularity in FDE can be resolved .

3. Development of the proposed algorithm

In this section, we introduce a mixed numerical method to solve the FDEs for $0 < \alpha < 2$ since the case for $\alpha > 2$ is not our primary practical concern [12]. Note that we formulate all equations in terms of the Caputo sense.

3.1. Chebyshev interpolation

The fractional IVP (1.1) is equivalent to the following Volterra integral equation

$$y(t) = y_0 + y_0' t + \frac{1}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))d\tau, \tag{3.1}$$

with $y(0) = y_0$. The other condition $y'(0) = y_0'$ is needed only for $1 < \alpha < 2$, so, for $0 < \alpha < 1$, $y'(0) = y_0'$ is not necessary, so we set $y_0' = 0$ for $0 < \alpha < 1$. Here, we suppose the the function $f(\tau)$ satisfies the Lipschitz condition.

First of all, we discretize the given time interval $[0, T]$ into N sub-intervals equally and the step size $h = \frac{T}{N}$. Based on all the solutions calculated in all previous intervals $[0, t_m]$, we approximate the solution at t_{m+1} as follows:

$$\begin{aligned} y(t_{m+1}) &= y_0 + y_0' t_{m+1} + \frac{1}{\Gamma(\alpha)}\int_0^{t_{m+1}} (t_{m+1}-\tau)^{\alpha-1} f(\tau, y(\tau))d\tau, \\ &= y_0 + y_0' t_{m+1} + \frac{1}{\Gamma(\alpha)}\int_0^{t_m} (t_{m+1}-\tau)^{\alpha-1} f(\tau, y(\tau))d\tau \end{aligned}$$

$$+\frac{1}{\Gamma(\alpha)}\int_{t_m}^{t_{m+1}}(t_{m+1}-\tau)^{\alpha-1}f(\tau,y(\tau))d\tau. \tag{3.2}$$

Note that the integral $\int_0^{t_m}(t_{m+1}-\tau)^{\alpha-1}f(\tau,y(\tau))d\tau$, the third term of right hand side of Eq (3.2) can be calculated numerically since the approximated values of $y(t_i)$ in $t_i \in [0,t_m]$ are calculated in the previous intervals $[t_i,t_{i+1}]$ for $i = 0,\cdots,m-1$. The details of calculation for the integration will be described in the following subsection. Instead, we discuss the calculation of the last term $\int_{t_m}^{t_{m+1}}(t_{m+1}-\tau)^{\alpha-1}f(\tau,y(\tau))d\tau$ of right hand side of Eq (3.2). Since the first three terms of the right hand side of Eq (3.2) are known values, we just let the three terms a constant C_m, so the Eq (3.2) is rewritten as

$$y(t_{m+1}) = C_m + \frac{1}{\Gamma(\alpha)}\int_{t_m}^{t_{m+1}}(t_{m+1}-\tau)^{\alpha-1}f(\tau,y(\tau))d\tau, \tag{3.3}$$

where

$$C_m = y_0 + y_0' t_{m+1} + \int_0^{t_m}(t_{m+1}-\tau)^{\alpha-1}f(\tau,y(\tau))d\tau.$$

To discretize the integral equation Eq (3.3) in a time interval $[t_m,t_{m+1}]$, we introduce Lagrangian interpolation described in Section 2.1. With $h = t_{m+1}-t_m$, we let $\tau = hs + t_m, \quad 0 \le s \le 1$. Here, we use Gauss-Legendre-Lobatto nodes for s described in Section 2.1.

Eq (3.3) leads to

$$\begin{aligned} y(t_{m+1}) &= C_m + \frac{1}{\Gamma(\alpha)}\int_0^1 (h-hs)^{\alpha-1}f(hs+t_m,y(hs+t_m))h\,ds \\ &= C_m + \frac{1}{\Gamma(\alpha)}\int_0^1 h^\alpha(1-s)^{\alpha-1}f(hs+t_m,y(hs+t_m))ds. \end{aligned} \tag{3.4}$$

Using the notations of Lagrangian interpolation

$$y(t) = \sum_{k=0}^{n} y_k L_k(t), \quad f(t,y(t)) = \sum_{k=0}^{n} f_k L_k(t), \quad t \in [t_m,t_{m+1}],$$

where

$$f_k = f(t_k,y_k), \quad \text{and} \quad L_k(t) = \sum_{j=0}^{n} c_{j,k}t^j, \quad k = 0,1,\cdots,n,$$

Eq (3.4) can be rewritten as

$$\begin{aligned} y(t_{m+1}) &= C_m + \frac{h^\alpha}{\Gamma(\alpha)}\int_0^1 (1-s)^{\alpha-1}\sum_{k=0}^{n} f_k L_k(hs+t_m)ds, \\ &= C_m + \frac{h^\alpha}{\Gamma(\alpha)}\int_0^1 (1-s)^{\alpha-1}\sum_{k=0}^{n} f_k \sum_{j=0}^{n} c_{j,k}(hs+t_m)^j ds, \\ &= C_m + \frac{h^\alpha}{\Gamma(\alpha)}\sum_{k=0}^{n} f_k \sum_{j=0}^{n} c_{j,k}\int_0^1 (1-s)^{\alpha-1}\sum_{i=0}^{j}\binom{j}{i}h^i s^i t_m^{j-i} ds, \end{aligned}$$

$$= \quad C_m + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{k=0}^{n} f_k \sum_{j=0}^{n} c_{j,k} \sum_{i=0}^{j} \binom{j}{i} h^i t_m^{j-i} \int_0^1 (1-s)^{\alpha-1} s^i ds, \tag{3.5}$$

where $\binom{j}{i}$ denotes a combination of j objects taken i. Using the definition of the beta distribution described in Eq (2.4) and the derivation of Eq (3.5), the following formula can be derived,

$$y(t) = C_m + \sum_{k=0}^{n} f_k \sum_{j=0}^{n} c_{j,k} \sum_{i=0}^{j} \frac{\Gamma(i+1)}{\Gamma(\alpha+i+1)} \binom{j}{i} (t-t_m)^{\alpha+i} t_m^{j-i}, \tag{3.6}$$

where t is a collocation point contained in $[t_m, t_{m+1}]$ defined above. Using the notation of the Lagrangian interpolation for $y(t)$, it produces a nonlinear system for y_k,

$$C_m - \sum_{k=0}^{n} y_k \sum_{j=0}^{n} c_{j,k} t^j + \sum_{k=0}^{n} f_k \sum_{j=0}^{n} c_{j,k} \sum_{i=0}^{j} \frac{\Gamma(i+1)}{\Gamma(\alpha+i+1)} \binom{j}{i} (t-t_m)^{\alpha+i} t_m^{j-i} = 0. \tag{3.7}$$

By solving the nonlinear system Eq (3.7) for $y_k, k = 0, \cdots, n$, we can finally approximate the solution $y(t_{m+1})$.

Theorem 3.1. *Let $p_N = \sum_{k=0}^{n} f_k L_k(x)$ defined above. Then, for $f \in C^N([0,T])$,*

$$\|f - p_N\| \leq \bar{c} h^{N+1} \|f^{(N+1)}\|, \tag{3.8}$$

for some positive constant $\bar{c}$.

Proof. The details of the proof can be found in [1, 16, 18]. □

3.2. Quadrature rule

In this subsection, we explain how to calculate C_m in Eq (3.7) using a piecewise quadratic quadrature rules. There are several well-developed quadrature rules [3–5, 17, 19, 24]. For example, [4] presented explicit quadrature rules for spaces of quintic splines with uniform knot sequences over finite domains by using only 2 quadrature points per element. In [17, 19], an efficient ruless for NURBS-based isogeometric analysis was presented for spaces arising in the calculation, for which the number of quadrature points in an optimal rule is almost equal to half the number of degrees-of-freedom. This idea was extended to the practical computation of quadrature rules for univariate non-uniform splines up to any precision. Despite various choices of the quadrature rules, in this work, we simply use a piecewise quadratic quadrature for calculation of C_m in Eq (3.7), since the quadrature efficiency is not the main focus of this work. Later, we will apply various higher order quadrature rules to increase accuracy of the proposed scheme and report it in the future.

Remind that the C_m is

$$C_m = y_0 + y_0' t_{m+1} + \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau.$$

To calculate the integration in C_m, we equally discretize the given whole integration interval $[0, t_m]$ into N subintervals. As described in subsection 3.1, in each subtime interval $[t_i, t_{i+1}]$, we use p-Gaussian node points to approximate the solutions. That is, there are p-points in the subinterval $[t_i, t_{i+1}]$:

$$t_i = t_{i,0} < t_{i,1} < \cdots < t_{i,p} = t_{i+1}.$$

By Eq (3.7) with a notation of Lagrangian interpolation, solutions at p node points in the interval $[t_i, t_{i+1}]$ can be calculated. Finally, at the time interval $[t_m, t_{m+1}]$, we can have estimated solutions in mp node points in whole previous time interval $[0, t_m]$. To denote the points explicitly, we define $\hat{t}$ by

$$\hat{t}_{ip+j} = t_{i,j}, \quad 0 \le i \le m-1, \quad 0 \le j \le p. \tag{3.9}$$

With having all calculated solutions in the whole previous time interval set $[0, t_m]$ and a notation in Eq (3.9), we consider a sub-interval $[\hat{t}_{2i}, \hat{t}_{2i+2}]$ having 3 time points $(\hat{t}_{2i}, \hat{t}_{2i+1}, \hat{t}_{2i+2})$, and we can write the integral as the 3-point quadrature polynomial to calculate the fractional integration. That is, the given function f is replaced by the quadratic polynomial:

$$\begin{aligned} f(\tau) \approx P(\tau) \;&=\; \frac{(\tau-\hat{t}_{2i+1})(\tau-\hat{t}_{2i+2})}{(\hat{t}_{2i}-\hat{t}_{2i+1})(\hat{t}_{2i}-\hat{t}_{2i+2})} f(\hat{t}_{2i}) \\ &\quad + \frac{(\tau-\hat{t}_{2i})(\tau-\hat{t}_{2i+2})}{(\hat{t}_{2i+1}-\hat{t}_{2i})(\hat{t}_{2i+1}-\hat{t}_{2i+2})} f(\hat{t}_{2i+1}) \\ &\quad + \frac{(\tau-\hat{t}_{2i})(\tau-\hat{t}_{2i+1})}{(\hat{t}_{2i+2}-\hat{t}_{2i})(\hat{t}_{2i+2}-\hat{t}_{2i+1})} f(\hat{t}_{2i+2}). \end{aligned} \tag{3.10}$$

Therefore the fractional integration in the sub-interval $[\hat{t}_{2i}, \hat{t}_{2i+2}]$ can be calculated as

$$\int_{\hat{t}_{2i}}^{\hat{t}_{2i+2}} (t_{m+1}-\tau)^{\alpha-1} f(\tau)\,d\tau \approx \omega_1 f(\hat{t}_{2i}) + \omega_2 f(\hat{t}_{2i+1}) + \omega_3 f(\hat{t}_{2i+2}), \tag{3.11}$$

where ω_1, ω_2 and ω_3 are easily obtained by calculation of the integration (3.10). Based on the calculation in sub-interval, we can extend this calculation to the whole time interval $[0, t_m]$. We summarize the calculation as the following remark.

Remark 3.2. *For* $0 < \alpha < 1$*, the piecewise quadratic quadrature for fractional integration is*

$$\int_0^{t_m} (t_{m+1}-\tau)^{\alpha-1} f(\tau)\,d\tau = \sum_{i=0}^{mp} \omega_{i,m+1} f(\hat{t}_i), \tag{3.12}$$

where

$$\omega_{i,m+1} = \begin{cases} \omega^1_{0,m+1}, & i = 0, \\ \omega^2_{k,m+1}, & i = 2k+1, \quad k = 0, 1, \cdots \frac{mp}{2} - 1 \\ \omega^1_{k,m+1} + \omega^3_{k-1,m+1}, & i = 2k, \quad k = 1, \cdots \frac{mp}{2} - 1 \\ \omega^3_{mp-2,m+1}, & i = mp, \end{cases}$$

$$\begin{aligned} \omega^1_{k,m+1} \;&=\; -\frac{P_{\alpha+2,k} - P_{\alpha+1,k}(2t_{m+1} - \hat{t}_{2k+2} - \hat{t}_{2k+1})}{(\hat{t}_{2k+2}-\hat{t}_{2k})(\hat{t}_{2k+1}-\hat{t}_{2k})} \\ &\quad - \frac{P_{\alpha,k}(t_{m+1}-\hat{t}_{2k+2})(t_{m+1}-\hat{t}_{2k+1})}{(\hat{t}_{2k+2}-\hat{t}_{2k})(\hat{t}_{2k+1}-\hat{t}_{2k})}, \end{aligned} \tag{3.13}$$

$$\begin{aligned} \omega^2_{k,m+1} \;&=\; \frac{P_{\alpha+2,k} - P_{\alpha+1,k}(2t_{m+1} - \hat{t}_{2k+2} - \hat{t}_{2k})}{(\hat{t}_{2k+1}-\hat{t}_{2k})(\hat{t}_{2k+2}-\hat{t}_{2k+1})} \\ &\quad + \frac{P_{\alpha,k}(t_{m+1}-\hat{t}_{2k+2})(t_{m+1}-\hat{t}_{2k})}{(\hat{t}_{2k+1}-\hat{t}_{2k})(\hat{t}_{2k+2}-\hat{t}_{2k+1})}, \end{aligned} \tag{3.14}$$

$$\omega^3_{k,m+1} \quad = \quad -P_{\alpha,k} - \omega^1_{k,m+1} - \omega^2_{k,m+1}, \tag{3.15}$$

with

$$P_{\alpha,k} = \frac{(t_{m+1} - \hat{t}_{2k+2})^\alpha - (t_{m+1} - \hat{t}_{2k})^\alpha}{\alpha}.$$

Note that to calculate Eq (3.12), we apply the quadrature rule in $\frac{mp}{2}$ sub-intervals. In each sub-interval, 3 w_i should be calculated and each w_i is obtained from one of $w^i_{k,m+1}$ defined in Eqs (3.13)–(3.15). Thus, a computational cost in the interval $[0, t_m]$ is $\frac{3mp}{2}$. Overall, since we discretize the given interval $[0, T]$ into N sub-intervals, the whole computational costs for quadrature in the proposed scheme can be $\sum_{m=1}^{N} \frac{3mp}{2} = O(pN^2)$.

Theorem 3.3. *If $f(t) \in C^3[0, T]$, $y(t_k)$ and y_k, $k = 0, 1, \cdots, 2m$ and $T = t_{2m}$ be the solutions, then there exists a constant C_0 such that*

$$\left| \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} f(\tau) d\tau - \sum_{i=0}^{mp} \omega_{i,m+1} f(\hat{t}_i) \right| \le C_0 h^{3+\alpha}. \tag{3.16}$$

Proof.

$$\begin{aligned} & \left| \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} f(\tau) d\tau - \sum_{i=0}^{mp} \omega_{i,m+1} f(\hat{t}_i) \right| \\ = & \left| \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} f(\tau) d\tau - \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} P(\tau) \right| d\tau, \end{aligned}$$

where $P(\tau)$ is the piecewise quadratic polynomial defined in Eq (3.10). Therefore,

$$\begin{aligned} & \left| \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} f(\tau) d\tau - \sum_{i=0}^{mp} \omega_{i,m+1} f(\hat{t}_i) \right| \\ = & \left| \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} (f(\tau) - P(\tau)) d\tau \right| \\ = & \left| \sum_{k=0}^{j-1} \int_{2k}^{2k+2} (t_{m+1} - \tau)^{\alpha-1} (f(\tau) - P(\tau)) d\tau \right| \\ \le & \left| \sum_{k=0}^{j-1} \int_{2k}^{2k+2} (t_{m+1} - \tau)^{\alpha-1} \frac{f'''(\xi)}{3!} (\tau - t_{2k})(\tau - t_{2k+1})(\tau - t_{2k+2}) d\tau \right| \\ \le & \max_{0 \le \xi \le t_m} \left| \frac{f'''(\xi)}{3!} \right| h^3 \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} d\tau = Ch^3, \end{aligned} \tag{3.17}$$

where C is a constant depending on α. Since $\int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} d\tau = \frac{t^\alpha_{m+1}}{\alpha} - \frac{h^\alpha}{\alpha}$, Eq (3.17) is summarized as

$$\left| \int_0^{t_m} (t_{m+1} - \tau)^{\alpha-1} f(\tau) d\tau - \sum_{i=0}^{mp} \omega_{i,m+1} f(\hat{t}_i) \right| \le C_0 h^{3+\alpha}, \tag{3.18}$$

where C is a constant depending on α. □

4. Numerical results

In this section, we test several examples to examine the effectiveness of the proposed scheme. The numerical results are compared with exact solutions. For showing the superiority of the methods, the results are compared with those obtained by other methods [11, 21, 30, 31]. To investigate numerically the convergence order in each example, experimentally determined order of convergence (EOC) is calculated as follows:

$$EOC = \frac{log\|Error(h_2)/Error(h_1)\|}{log\|h_2/h_1\|},$$

where $Error(h)$ denotes the absolute error between the analytic solutions and the numerical solutions simulated with a step size h. In addition, if the right hand side of the FDEs is nonlinear, a nonlinear system is derived from Eq (3.7). For calculation of the nonlinear system, the matlab-builtin routine "fsolve" is used. Also, for the initial guess of the nonlinear solver, we use the fractional explicit Euler method, which is the most basic and economical method. Details of each problem will be explained in each subsection.

4.1. Linear case

As the first example, we consider a linear fractional differential equation described by

$$D^{\alpha}y(t) = t^2 + \frac{2}{\Gamma(3-\alpha)}t^{(2-\alpha)} - y(t), \tag{4.1}$$

with the initial condition $y(0) = 0$ and $y'(0) = 0$. The exact solutions is $y(t) = t^2$. For the experiment, it is marched from $t = 0.0$ to $t = 1.0$ with a step size $h = 0.1$ and 4 Chebyshev node points are used in each sub interval. To check the effectiveness of the proposed scheme, we plot the absolute errors between the proposed scheme and the analytic solution for different the value $\alpha = 0.25, 0.5,$ and 0.75 in Figure 1.

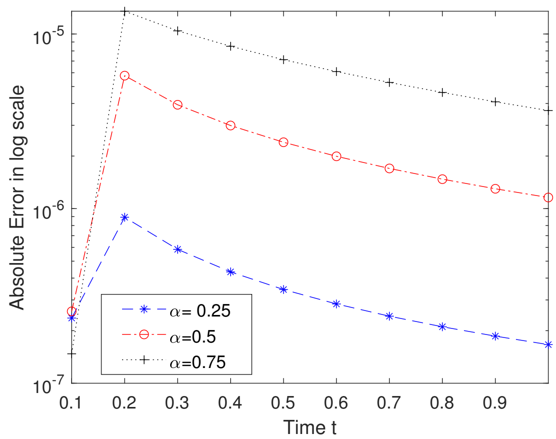

Figure 1. Comparison of absolute errors for different α.

It can be seen that the proposed scheme seems to work well for this problem.

To examine the numerical convergence order, we calculate numerical errors at $T = 1$ and the experimentally determined order of convergence (EOC) by varying the step size

$h = 1/10, 1/20, 1/40, 1/80, 1/160$ and $1/320$ for different $\alpha = 0.5$, 0.75 and 1.5 with 2 Chebyshev node points. All results are reported in Table 1 and Figure 2.

Table 1. Comparison of Absolute errors (Error) and the experimentally determined order of convergence (EOC) for $\alpha = 0.5, 0.75$ and $\alpha = 1.5$.

	$\alpha = 0.5$		$\alpha = 0.75$		$\alpha = 1.5$	
h	Error	EOC	Error	EOC	Error	EOC
1/10	8.5380e-07	-	5.2605e-06	-	0.0014	-
1/20	2.1429e-07	1.9943	1.1846e-06	2.1508	5.1105e-04	1.4539
1/40	4.3641e-08	2.2958	2.5665e-07	2.2065	1.8079e-04	1.4991
1/80	8.2507e-09	2.4031	5.4724e-08	2.2296	6.3940e-05	1.4995
1/160	1.5050e-09	2.4548	1.1623e-08	2.2352	2.2610e-05	1.4998
1/320	2.3670e-10	2.6686	2.4488e-09	2.2468	7.9944e-06	1.4999

Table 1 shows that for $\alpha = 0.5$, EOC converges to 2.5, for $\alpha = 0.75$ EOC to 2.25, and for $\alpha = 1.5$ EOC to 1.5. One can guess that for linear problems, the proposed algorithm has $3 - \alpha$ convergence order. For more details, we plot the numerical results and straight line $y = (3 - \alpha)x$ for $\alpha = 0.75$ in Figure 2. It can be seen that the two lines in the figure are parallel, so the convergence order of the numerical results for $\alpha = 0.75$ is 2.25. For $\alpha = 1.5$, one can convincingly conclude the convergence order, but for $\alpha = 0.5$ it is not so clear.

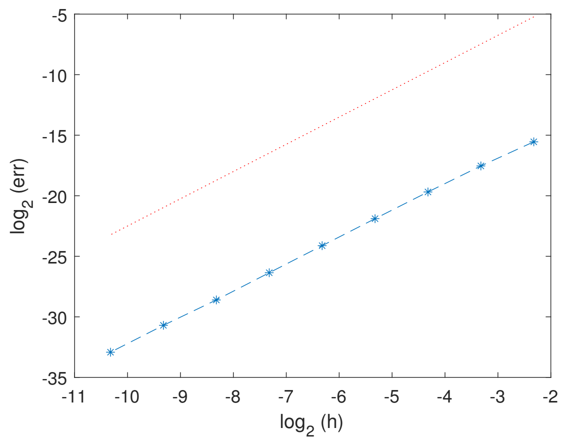

Figure 2. The experimentally determined order of convergence (EOC) for $\alpha = 0.75$; Blue line represents the absolute error with log-scale and red line is a linear line with a slope=2.25.

Note that for the linear problems of $1 < \alpha < 2$, this scheme seems not appropriate since the convergence order is quite low. Also, to be precise, we need to theoretically analyze the convergence orders.

4.2. Nonlinear case 1

In this subsection, the following nonlinear fractional differential equation is considered

$$D^{\alpha} y(t) = \frac{\Gamma(5+\alpha)}{24} t^4 + t^{8+2\alpha} - y^2(t), \tag{4.2}$$

with an initial condition $y(0) = 0$ and $y'(0) = 0$. The exact solution of this problem is given as

$$y(t) = t^{4+\alpha}.$$

For the experiment, we march from $t = 0.0$ to $t = 1.0$ with step size $h = 0.2$. We calculate the numerical errors between the proposed scheme and the analytic solution is calculated by varying the number of Chebyshev node points $n = 2, 4$, and 6 for the value $\alpha = 0.25, 0.5$, and 0.75 and all results are plotted in Figure 3. The results show that the proposed scheme has higher accuracy as the number of Chebyshev node points is increasing, which implies why the collocation methods are useful.

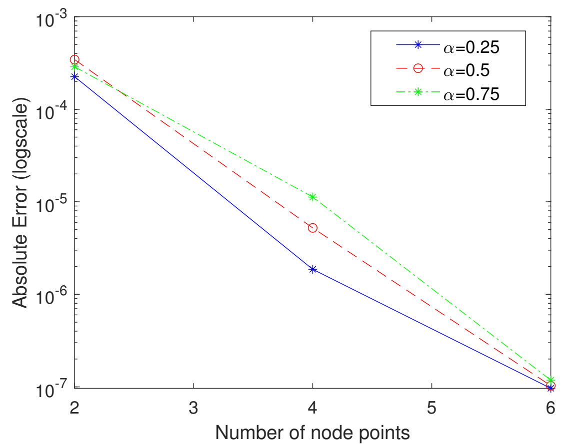

Figure 3. Comparison of absolute errors by varying the number of node points $n = 2, 4$, and 6 for different $\alpha = 0.25, 0.5$ and 0.75.

For investigating the numerical convergence order for the nonlinear problem, numerical errors and the experimentally determined order of convergence (EOC) are computed by varying the step size h for different $\alpha = 0.25, 0.5$ and 0.75 with 2 Chebyshev node points in time interval $[0, 1]$ and results are reported in Table 2.

Table 2. Comparison of Absolute errors (Error) and the experimentally determined order of convergence (EOC) for $\alpha = 0.25, 0.5$ and $\alpha = 0.75$.

	$\alpha = 0.25$		$\alpha = 0.5$		$\alpha = 0.75$	
h	Error	EOC	Error	EOC	Error	EOC
1/10	2.7529e-05	-	3.5555e-05	-	2.4196e-05	-
1/20	3.2219e-06	3.0950	3.4520e-06	3.3645	1.9254e-06	3.6515
1/40	3.6714e-07	3.1335	3.2360e-07	3.4151	1.4941e-07	3.6878
1/80	4.1133e-08	3.1580	2.9722e-08	3.4446	1.1444e-08	3.7066
1/160	4.5541e-09	3.1751	2.6903e-09	3.4657	8.6836e-10	3.7202
1/320	5.0643e-10	3.1687	2.2742e-10	3.5643	6.5728e-11	3.7237

In Table 2, we observe that for $\alpha = 0.25, 0.5$ and 0.75, the EOCs converge to 3.2, 3.5 and 3.75 as the step size is decreasing. Similar to the previous result, for $\alpha = 0.5$ and 0.75, one can convincingly conclude the convergence order, but for $\alpha = 0.25$ it is not so clear. It is almost $3 + \alpha$, unlike the case of the linear problem. It is almost identical to the theoretic convergence order described in Theorem 3.3.

Additionally, to investigate the efficiency of the proposed scheme, we compare the results from the proposed scheme with those from the existing higher order methods introduced in [7]. For the experiment, we march from $t = 0$ to $t = 1$ with various time steps $h = 1/10, 1/20, 1/40, 1/80, 1/160$ and $1/320$ to check the EOCs for both schemes at the same time. For the proposed scheme, 2 Chebyshev nodes are used. We calculate the absolute errors and the corresponding EOCs for both methods by difference between the results and analytic solutions. All results are reported in Table 3.

Table 3. Comparing Absolute errors (Error) and the experimentally determined order of convergence (EOC) obtained from the proposed scheme and the method in [7] for $\alpha = 0.5$.

	Proposed scheme		Method in [7]	
h	Error	EOC	Error	EOC
1/10	3.5555e-05	-	2.2974e-04	-
1/20	3.4520e-06	3.3645	2.2161e-05	3.3739
1/40	3.2360e-07	3.4151	2.0734e-06	3.4179
1/80	2.9722e-08	3.4446	1.9054e-07	3.4439
1/160	2.6903e-09	3.4657	1.7293e-08	3.4618
1/320	2.2742e-10	3.5643	1.5566e-09	3.4738

As mentioned, the method introduced in [7] is also higher order numerical scheme, so as seen in Table 3, the method in [7] have similar convergence order to the proposed scheme. However, it can be easily seen that the proposed scheme has more accurate results in this comparison. Therefore, we can conclude that the proposed scheme is quite efficient for this problem.

4.3. Nonlinear case 2

As the last example, we consider the following nonlinear fractional differential equation described by

$$D^{\alpha}y(t) = \frac{40320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(\frac{3}{2}t^{\alpha/2} - t^{4}\right)^{3} - y^{3/2}, \tag{4.3}$$

with an initial condition $y(0) = 0$ and $y'(0) = 0$. The exact solution of this problem is

$$y(t) = t^{8} - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}.$$

To investigate the accuracy of the proposed scheme, we compare the numerical results of the proposed scheme with those obtained from existing methods-one is a numerical method introduced in [21] and the other one is predictor-correction methods [11]. For the experiment, it is marched from $t = 0.0$ to $t = 1.0$ with step size $h = 0.025$ for $\alpha = 0.75$. The numerical results introduced in [21] have been already represented in [21], so the data are directly excerpted from the reference. The predictor-corrector methods [11] is implemented for the same setting and the proposed scheme uses 4 Chebyshev node points. Figure 4 represents numerical errors generated from three methods over the given time interval.

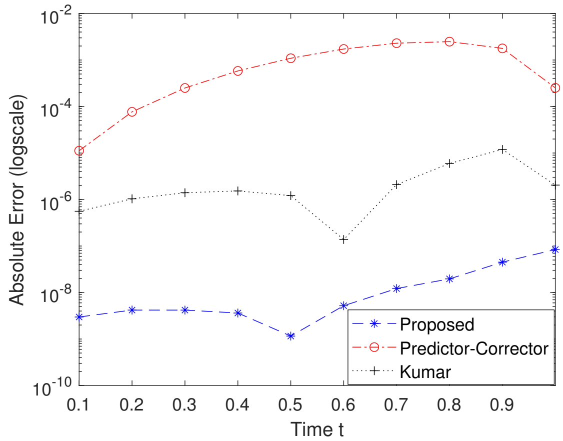

Figure 4. Comparison of absolute errors over time interval [0.0, 1.0] for proposed scheme (Proposed) and other methods (Kumar in [21] and Pred-Corr.) for case $\alpha = 0.75$.

The figure shows that the proposed scheme generates more accurate solution for the problem, compared with other existing methods. As well as the numerical methods in references [11, 21], there are several references to represent the numerical results of this example and the results can be found in [2, 29–31], etc. Even compared with the results in [31] where is most recently developed and has higher accuracy, the results of the proposed scheme is quite competitive.

To check the numerical convergence order, we investigate numerical errors at $T = 1$ and the experimentally determined order of convergence (EOC) by varying the step size h for different $\alpha = 0.25, 0.5$ and 0.75 with 2 Chebyshev node points in Table 4.

Table 4. Comparison of Absolute errors (Error) and the experimentally determined order of convergence (EOC) for $\alpha = 0.25, 0.5$ and $\alpha = 0.75$.

	$\alpha = 0.25$		$\alpha = 0.5$		$\alpha = 0.75$	
h	Error	EOC	Error	EOC	Error	EOC
1/20	4.2585e-05	-	3.9458e-05	-	2.0432e-05	-
1/40	4.9115e-06	3.1161	3.7647e-06	3.3897	1.6291e-06	3.6487
1/80	5.4822e-07	3.1633	3.4955e-07	3.4290	1.2757e-07	3.6747
1/160	6.4956e-09	3.2085	2.9121e-09	3.4545	7.5437e-10	3.7074
1/320	7.0151e-10	3.2109	2.4571e-10	3.5670	1.9482e-11	5.2751

The results show that for $\alpha = 0.25, 0.5$ and 0.75, the EOCs are $3.25, 3.5$ and 3.75, respectively. That is, as similar to the previous nonlinear example, a numerical convergence order for this problem is about $3 + \alpha$. For checking in detail, we plot the EOC for $\alpha = 0.5$ and a straight line with having a slope 3.5 in Figure 5. The figure shows the two lines are exactly parallel.

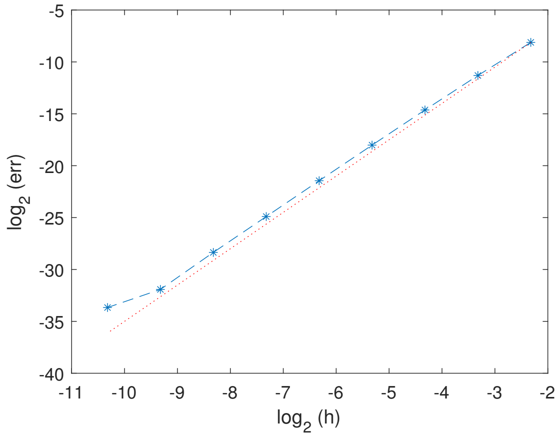

Figure 5. The experimentally determined order of convergence (EOC) for $\alpha = 0.5$: Blue line represents the absolute error with log-scale and red line is a linear line with a slope=3.5.

To examine the numerical convergence order for $1 < \alpha < 2$, Numerical errors at $T = 1$ and the experimentally determined order of convergence (EOC) are computed by varying the step size h for different $\alpha = 1.5$ and 1.75 with 2 Chebyshev node points and reported in Table 5.

Table 5. Comparison of Absolute errors (Error) and the experimentally determined order of convergence (EOC) for $\alpha = 1.5$ and $\alpha = 1.75$.

	$\alpha = 1.5$		$\alpha = 1.75$	
h	Error	EOC	Error	EOC
1/20	6.8725e-06	-	8.0313e-06	-
1/40	4.6601e-07	3.8824	5.1554e-07	3.9615
1/80	3.0672e-08	3.9254	3.2723e-08	3.9777
1/160	2.0802e-09	3.8821	2.0693e-09	3.9831
1/320	1.3313e-10	3.9658	1.3002e-10	3.9923
1/640	8.2365e-12	4.0147	7.7750e-12	4.0637
1/1280	4.2821e-13	4.2656	4.0962e-13	4.2465

The results show that for $\alpha = 1.5$ and 1.75, the EOC is over 4. It is quite competitive convergence order compared with the existing techniques.

5. Conclusions

In this paper, a mixed numerical technique is developed for solving fractional differential equations (FDEs) by splitting into two time intervals-the whole previous time interval and current time integration step. At a current time integration, we compute the solution by using Chebyshev collocation methods which can be rewritten as a Lagrangian interpolating form. By the Lagrangian interpolating form, we can remove a singularity which belongs originally to the FDEs. For calculating the fraction integral in the previous time interval, we use a piecewise quadratic quadrature technique to get higher accuracy by using all solutions including solutions at the collocation points. Several numerical examples are presented to show the efficiency of the proposed method and compare it with several existing methods. The numerical results present that the proposed techniques can get competitively better accuracy and and numerically convergence orders $3 - \alpha$ for linear cases. Also, the numerical convergence orders are $3 + \alpha$ and over 4 for nonlinear cases, when $0 < \alpha < 1$ and $1 < \alpha < 2$, respectively.

In order to fully explore the efficiency of the proposed scheme, several extended issues are currently being pursued. First, we apply the proposed scheme to other types of fractional differential equations such as several Bagley-Torvik equations, a popular FDE with $\alpha > 1$. Secondly, we are doing theoretically convergence analysis for linear which will be hopefully consistent with the numerical convergence orders. Lastly, an adaptive time stepping method should be considered by calculating the fractional integral for long time simulations. Preliminary results are quite promising. Results along these issues will be reported soon.

Acknowledgments

This work was supported by basic science research program through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (grant number: NRF-2019R1F1A1058378).

Conflict of interest

The author declares no conflict of interest in this paper.

References

1. K. E. Atkinson, *An introduction to numerical analysis*, John Wiley & Sons, 1989.
2. R. B. Albadarneh, M. Zerqat, I. M. Batiha, Numerical solutions for linear and non-linear fractional differential equations, *Int. J. Pure App. Math.*, **106** (2016): 859–871. doi: 10.12732/ijpam.v106i3.12.
3. M. Barton V. M. Calo, Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis, *Comput. Method. Appl. M.*, **305** (2016), 217–240. doi: 10.1016/j.cma.2016.02.034.
4. M. Barton, R. Ait-Haddou, V. M. Calo, Gaussian quadrature rules for C1 quintic splines with uniform knot vectors, *J. Comput. Appl. Math.*, **322** (2017), 57–70. doi:10.1016/j.cam.2017.02.022.
5. M. Barton, V. M. Calo, Gauss-Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis, *Comput. Aided Des.*, **82** (2017), 57–67. doi:10.1016/j.cad.2016.07.003.
6. S. Bu, W. Jung, P. Kim, An error embedded Runge-Kutta method for initial value problems, *Kyungpook Math. J.*, **56** (2016), 311–327. doi:10.5666/KMJ.2016.56.2.311.
7. J. Y. Cao, C. J. Xu, A high order schema for the numerical solution of the fractional ordinary differential equation, *J. Comput. Phys.*, **238** (2013), 154–168. doi:10.1016/j.jcp.2012.12.013.
8. K. Diethelm, Efficient solution of multi-term fractional differential equations using P(EC)mE methods, *Computing*, **71** (2003), 305-319. doi:10.1007/s00607-003-0033-3.

9. K. Diethelm, An investigation of some nonclassical methods for the numerical approximation of Caputo-type fractional derivatives, *Numer. Algor.*, **47** (2008), 361–390. doi:10.1007/s11075-008-9193-8.

10. K. Diethelm, The analysis of fractional differential equations: An application-oriented exposition using differential operators of Caputo type, Springer Science & Business Media, 2010.

11. K. Diethelm, N. J. Ford, A. D. Freed, A predictor-corrector approach for the numerical solution of fractional differential equations, *Nonlinear Dynam.*, **29** (2002), 3–22. doi:10.1023/A:1016592219341.

12. K. Diethelm, N. J. Ford, A. D. Freed, Detailed error analysis for a fractional Adams method, *Numer. Algor.*, **36** (2004), 31–52. doi:10.1023/B:NUMA.0000027736.85078.be.

13. W. Deng, Smoothness and stability of the solutions for nonlinear fractional differential equations, *Nonlinear Anal. Theor.*, **72** 2010, 1768–1777. doi:10.1016/j.na.2009.09.018.

14. W. Deng, S. Du, Y. Wu, High order finite difference WENO schemes for fractional differential equations, *Appl. Math. Lett.*, **26** (2012), 362–366. doi:10.1016/j.aml.2012.10.005.

15. N. Ford, M. Morgado, M. Rebelo, Nonpolynomial collocation approximation of solutions to fractional differential equations, *Frac. Calc. Appl. Anal.*, **16** (2013), 874–891. doi:10.2478/s13540-013-0054-3.

16. C. W. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, 1971.

17. R. Hiemstra, F. Calabro, D. Schillinger, T. J. R. Hughes, Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis, *Comput. Method. Appl. M.*, **316** (2017), 966–1004. doi:10.1016/j.cma.2016.10.049.

18. E. Hairer, S. P. Norsett, G. Wanner, *Solving ordinary differential equations. I nonstiff*, Springer, 1993.

19. T. J. R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, *Comput. Method. Appl. M.*, **199** (2010), 301–313. doi:10.1016/j.cma.2008.12.004.

20. Y. Jeon, S. Bak, S. Bu, Reinterpretation of multi-Stage methods for stiff systems: A comprehensive review on current perspectives and recommendations, *Mathematics*, **7** (2019), 1158. doi:10.3390/math7121158.

21. P. Kumar, O. P. Agrawal, An approximate method for numerical solution of fractional differential equations, *Signal Process.*, **86** (2006), 2602–2610. doi:10.1016/j.sigpro.2006.02.007.

22. S. Khatoon, I. Uddin, D. Baleanu, Approximation of fixed point and its application to fractional differential equation, *J. Appl. Math. Comput.*, **66** (2021), 507–525. doi:10.1007/s12190-020-01445-1.

23. C. Lv, M. Azaiez, C. Xu, Spectral deferred correction methods for fractional differential equations, *Numer. Math. Theor. Meth. Appl.*, **11** (2018), 729–751. doi: 10.4208/nmtma.2018.s03.

24. G. Nikolov, Gaussian quadrature formulae for splines, In: *ISNM International Series of Numerical Mathematics*, Basel: Birkhuser, 1993.

25. I. Podlubny, Numerical solution of ordinary fractional differential equations by the fractional difference method, In: *Proceedings of the Second International Conference on Difference Equations*, 1997.

26. I. Podlubny, *Fractional differential equations*, San Diego: Academic Press, 1999.

27. M. Rehman, A. Idrees, U. Saeed, A quadrature method for numerical solutions of fractional differential equations, *Appl. Math. Comput.*, **307** (2017), 38–49. doi:10.1016/j.amc.2017.02.053.

28. M. F. Simes Patrício, H. Ramos, M. Patrcio, Solving initial and boundary value problems of fractional ordinary differential equations by using collocation and fractional powers, *J. Comput. Appl. Math.*, **354** (2019), 348–359. doi:10.1016/j.cam.2018.07.034.

29. J. Xin, J. Huang, W. Zhao, J. Zhu, A spectral deferred correction method for fractional differential equations, *Abstr. Appl. Anal.*, **2013** (2013), 139530. doi:10.1155/2013/139530.

30. Y. Yan, K. Pal, N. Ford, Higher order numerical methods for solving fractional differential equations, *Bit Numer. Mathe.*, **54** (2014), 555–584. doi:10.1007/s10543-013-0443-3.

31. J. Zhao, Y. Li, Y. Xu, A kind of product integration scheme for solving fractional ordinary differential equations, *Appl. Numer. Math.*, **136** (2019), 279–292. doi:10.1016/j.apnum.2018.10.014.

32. Y. Zhong, X. B. Bao, L. B. Liu, Z. F. Liang, Analysis of a finite difference scheme for a nonlinear Caputo fractional differential equation on an adaptive grid, *AIMS Mathematics*, **6** (2021), 8611–8624. doi:10.3934/math.2021500.

AIMS Press