Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 206 (2007) 174–188
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

# Short memory principle and a predictor–corrector approach for fractional differential equations

Weihua Deng[a, b, *]

[a] *Schooll of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China*
[b] *Department of Mathematics, Shanghai University, Shanghai 200444, China*

Received 28 April 2006

**Abstract**

Fractional differential equations are increasingly used to model problems in acoustics and thermal systems, rheology and modelling of materials and mechanical systems, signal processing and systems identification, control and robotics, and other areas of application. This paper further analyses the underlying structure of fractional differential equations. From a new point of view, we apprehend the *short memory principle* of fractional calculus and farther apply a Adams-type predictor–corrector approach for the numerical solution of fractional differential equation. And the detailed error analysis is presented. Combining the *short memory principle* and the predictor–corrector approach, we gain a good numerical approximation of the true solution of fractional differential equation at reasonable computational cost. A numerical example is provided and compared with the exact analytical solution for illustrating the effectiveness of the *short memory principle*.


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

*Keywords:* Fractional differential equation; Caputo derivative; Short memory principle; Numerical solution; Predictor–corrector method

## 1. Introduction

Fractional calculus [4,22], which has almost the same history as classical calculus, did not attract enough attention for a long time. However, in recent decades fractional differential equations have been more and more applied to model acoustics and thermal systems, rheology and modelling of materials and mechanical systems, signal processing and systems identification, control and robotics, etc. [30,1,28,29,3]. Moreover, many systems modelled with the help of fractional calculus display rich fractional dynamical behavior, such as viscoelastic systems [23], colored noise [27], boundary layer effects in ducts [32], electromagnetic waves [19], fractional kinetics [24,26,34], and electrode–electrolyte polarization [21,33]. For linear fractional differential equations with constant coefficients, analytical solutions are available by applying Laplace–Fourier transform techniques (although sometimes they cannot be employed conveniently in engineering, because more often they are described by using Mittag-Leffler function)

* Corresponding author. School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China. Tel.: +86 21 8912483; fax: +86 21 8912481.
*E-mail address:* dengwh@lzu.edu.cn.

0377-0427/$ - see front matter 
doi:10.1016/j.cam.2006.06.008

[4,9,22,30]. However, plentifully utility problems are modelled by linear systems with variable coefficients or even nonlinear systems [25,2,7,8,10]. This paper discusses the underlying structure and numerical solution of the following initial value problem

$$D_*^\alpha x(t) = f(t, x(t)), \quad x^{(k)}(0) = x_0^{(k)}, \;\; k = 0, 1, \ldots, \lceil\alpha\rceil - 1, \tag{1}$$

where $\alpha \in (0, \infty)$, $x_0^{(k)}$ can be any real numbers and $D_*^\alpha$ denotes the fractional derivative in the Caputo sense [5], defined by

$$D_*^\alpha y(t) = J^{n-\alpha} D^n y(t).$$

Here $n := \lceil\alpha\rceil$ is the first integer not less than $\alpha$, $D^n$ is the classical $n$th-order derivative and for $\mu > 0$, $J^\mu$ is the $\mu$-order Riemann–Liouville integral operator expressed as follows:

$$J^\mu y(t) = \frac{1}{\Gamma(\mu)} \int_0^t (t-\tau)^{\mu-1} y(\tau)\,\mathrm{d}\tau.$$

Caputo derivative is widely used in engineering and numerical computation [28,29,9–18], although from pure mathematical viewpoint, Riemann–Liouville derivative is more welcome and many earlier research papers use it instead of Caputo derivative [4,22,30]. In general, we need to specify some additional conditions to make sure our discussed equations have a unique solution. These additional conditions, in many situations, describe some properties of the solution at the initial time [14], however the fractional derivative does not have convenient used physical meaning (there are already some progress in the geometric and physical interpretation of fractional calculus [31] and physical interpretation of the initial condition of fractional differential equations with Riemann–Liouville derivative [20]), so it is difficult to evaluate the initial value, some authors require homogeneous initial conditions when solving the fractional differential equations with Riemann–Liouville derivatives, we know Riemann–Liouville derivatives are equivalent to Caputo derivatives under homogeneous initial conditions [30]. However, when the Caputo derivative is chosen, it allows us to specify inhomogeneous initial conditions also if it is desired, because it just require the initial conditions are given in terms of integer derivatives of unknown functions which have clear physical meaning.

It is well known that the initial value problem (1) is equivalent to the Volterra integral equation [6,11,14,15]

$$x(t) = \sum_{k=0}^{\lceil\alpha\rceil-1} x_0^{(k)} \frac{t^k}{k!} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, x(\tau))\,\mathrm{d}\tau \tag{2}$$

in the sense that if a continuous function solves (2) if and only if it solves (1). Diethelm, Ford and their coauthors successfully presented the numerical approximation of (2) using Adams-type predictor–corrector approach and gave the corresponding detailed error analysis in [14] and [15], respectively, the convergent order of their approach was proved to be $\min(2, 1+\alpha)$. As being referred to in [14, Section 3.1], the arithmetic complexity of their algorithm with step size $h$ is $O(h^{-2})$, whereas a comparable algorithm for a classical initial value problem only give rise to $O(h^{-1})$. The difficulty of computational complexity is essentially because fractional derivatives are non-local operators. There are already two typical ways which are suggested to overcome this difficulty. One seems to be the fixed memory principle of Podlubny [30]. However, it is shown that the fixed memory principle is not suitable for Caputo derivative, because we cannot reduce the computational cost significantly for preserving the convergent order [14,18]. The other more hopeful idea seems to be the nested memory concept of Ford and Simpson [18] which can lead to $O(h^{-1}\log(h^{-1}))$ complexity, but still retain the order of convergence. This idea depends on the decaying of the integral kernel $(t-\tau)^{\alpha-1}$ of (2) as $t$ increases, so the available $\alpha$ must be limited to the interval $(0, 1)$. For more detailed analysis we refer to [18].

We apprehend the *short memory principle* (or fixed memory principle or logarithmic memory principle) from a new viewpoint and correspondingly extend the *short memory principle*'s effective range from $\alpha \in (0, 1)$ to $\alpha \in (0, 2)$, which is well in agreement with that the case $\alpha \geqslant 2$ does not seem to be of major practical interest [14]. When $\alpha \in (0, 1)$, the kernel's decaying is greatly speeded and it has the property of Podlubny's fixed memory principle. Especially the idea of Ford and Simpson's nested memory concept is also effective for numerical computation while $\alpha \in (1, 2)$. By applying the predictor–corrector approach which is different from [14], we obtain a good numerical approximation of the true solution of fractional differential equation with convergent order 2 as $\alpha \in (1, \infty)$. And in case $\alpha \in (1, 2)$, further combining the *short memory principle* and the predictor–corrector approach we minimize the computational complexity to $O(h^{-1} \log(h^{-1}))$ at preserving the order of accuracy.

## 2. The structure and short memory principle for fractional differential equations

As it is well known, the integer order (classical) differential operator is a local operator but fractional order differential operator is a non-local one. The so-called non-local property is to say the next state of one system not only depends on its current state but also its historical states starting from the initial time, which of course are more close to reality and also should be the main reason why fractional calculus become more and more popular. The local operator has the property that just present state to one system can determine its coming state. But the integer order differential operator is really irrelevant to its history? Let us see the following ODE:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(t, x(t)), \quad x(0) = x_0,$$

which is equivalent to

$$\begin{aligned} x(t_{n+1}) &= x_0 + \int_0^{t_{n+1}} f(\tau, x(\tau))\,\mathrm{d}\tau \\ &= x_0 + \int_0^{t_n} f(\tau, x(\tau))\,\mathrm{d}\tau + \int_{t_n}^{t_{n+1}} f(\tau, x(\tau))\,\mathrm{d}\tau \\ &= x(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, x(\tau))\,\mathrm{d}\tau, \end{aligned}$$

where, and in the following, $t_n = nh$, $t_{n+1} = (n+1)h$, $h$ is a small positive number, we also denote the step length by $h$ when performing error analysis.

From the above formula, we can see clearly $x(t_{n+1})$ relies on the values of $x$ in the whole interval $[0, t_{n+1}]$. But fortunately all the contributions of $x$ to $x(t_{n+1})$ in the interval $[0, t_n]$ can be represented by $x(t_n)$. A natural question is whether the fractional order operator has also the similar property? The answer of course is negative, because if it is true then fractional order operator becomes local. But we can further ask whether $x(t_n)$ can embrace almost all the contributions of $x$ to $x(t_{n+1})$ in the interval $[0, t_n]$ for the fractional order operator? In case $\alpha \in (0, 2)$, the answer is positive. In the following, we discuss it in detail.

For $\alpha \in (0, 1)$ and $\alpha \in (1, \infty)$, we can write (2) as, respectively,

$$\begin{aligned} x(t_{n+1}) = x(t_n) &+ \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\,\mathrm{d}\tau \\ &+ \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, x(\tau))\,\mathrm{d}\tau, \quad \alpha \in (0, 1), \end{aligned} \tag{3}$$

and

$$\begin{aligned} x(t_{n+1}) = \sum_{k=1}^{\lceil \alpha \rceil - 1} \frac{x_0^{(k)}}{k!} (t_{n+1}^k - t_n^k) + x(t_n) &+ \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\,\mathrm{d}\tau \\ &+ \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, x(\tau))\,\mathrm{d}\tau, \quad \alpha \in (1, \infty). \end{aligned} \tag{4}$$

By the observation of (3) and (4), we can see the non-local property of $D_*^\alpha$ induces the term

$$\frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, x(\tau))\,\mathrm{d}\tau. \tag{5}$$

In effect, if $\alpha \in (0, 2)$ the integration kernel of (5) fades quickly when the time history becomes longer,

$$
\begin{aligned}
&\frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1}) f(\tau, x(\tau))\, \mathrm{d}\tau \\
&= \frac{1}{\Gamma(\alpha)(\alpha-1)} \int_0^{t_n} \left( \int_{t_n-\tau}^{t_{n+1}-\tau} z^{\alpha-2}\, \mathrm{d}z \right) f(\tau, x(\tau))\, \mathrm{d}\tau \\
&= \frac{1}{\Gamma(\alpha)(\alpha-1)} \int_{t_{n-1}}^{t_n} \left( \int_{t_n-\tau}^{t_{n+1}-\tau} z^{\alpha-2}\, \mathrm{d}z \right) f(\tau, x(\tau))\, \mathrm{d}\tau \\
&\quad + \frac{1}{\Gamma(\alpha)(\alpha-1)} \int_{t_{n-2}}^{t_{n-1}} \left( \int_{t_n-\tau}^{t_{n+1}-\tau} z^{\alpha-2}\, \mathrm{d}z \right) f(\tau, x(\tau))\, \mathrm{d}\tau \\
&\quad + \cdots + \frac{1}{\Gamma(\alpha)(\alpha-1)} \int_{t_1}^{t_2} \left( \int_{t_n-\tau}^{t_{n+1}-\tau} z^{\alpha-2}\, \mathrm{d}z \right) f(\tau, x(\tau))\, \mathrm{d}\tau \\
&\quad + \frac{1}{\Gamma(\alpha)(\alpha-1)} \int_0^{t_1} \left( \int_{t_n-\tau}^{t_{n+1}-\tau} z^{\alpha-2}\, \mathrm{d}z \right) f(\tau, x(\tau))\, \mathrm{d}\tau \\
&= \frac{h}{\Gamma(\alpha)(\alpha-1)} \int_{t_{n-1}}^{t_n} (z_1^*(\tau))^{\alpha-2} f(\tau, x(\tau))\, \mathrm{d}\tau + \frac{h}{\Gamma(\alpha)(\alpha-1)} \int_{t_{n-2}}^{t_{n-1}} (z_2^*(\tau))^{\alpha-2} f(\tau, x(\tau))\, \mathrm{d}\tau \\
&\quad + \cdots + \frac{h}{\Gamma(\alpha)(\alpha-1)} \int_{t_1}^{t_2} (z_n^*(\tau))^{\alpha-2} f(\tau, x(\tau))\, \mathrm{d}\tau + \frac{h}{\Gamma(\alpha)(\alpha-1)} \int_0^{t_1} (z_n^*(\tau))^{\alpha-2} f(\tau, x(\tau))\, \mathrm{d}\tau,
\end{aligned}
\tag{6}
$$

where $z_1^*(\tau) \in (0, t_2),\ z_2^*(\tau) \in (t_1, t_3), \ldots, z_{n-1}^*(\tau) \in (t_{n-2}, t_n),\ z_n^*(\tau) \in (t_{n-1}, t_{n+1})$.

According to (6), we can note the integration (5)'s kernel $(t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1}$ decays (algebraically) by the order $2-\alpha$ when $\alpha \in (0, 2)$, but in [18] the integral kernel $(t_{n+1}-\tau)^{\alpha-1}$ decays only by the order $1-\alpha$ while $\alpha \in (0, 1)$. This is also the reason why we can extend the range of the *short memory principle* of fractional differential equations from $\alpha \in (0, 1)$ to $\alpha \in (0, 2)$. Because of the *short memory principle* of fractional differential equations, two possible ways to numerically approximate the integration (5) are discussed in the following two subsections.

### 2.1. Fixed integral length

For performing numerical computation, the simplest approach is to disregard the tail of the integration of (5) and to integrate only over a fixed period of recent history [30,18]. If we can do this, then the computational cost at each step is reduced to O(1). Based on this kind of idea, Podlubny [30] show that it is possible for Riemann–Liouville derivative and the use of a fixed integral length $T$ introduces an error $E$ (independent of the full interval of integration) satisfies $E < MT^{-\alpha}/\Gamma(1-\alpha)$. We can choose the value of $T$ such that it meets our desired accuracy. But for the Caputo derivative, Ford and Simpson [18] detailedly analysed the employ of a fixed integral length $T$ induces the truncation error $E < (t_{n+1}^{1-\alpha} - T^{1-\alpha})M/\Gamma(2-\alpha)$ and they drew the conclusion that unless the integral over which we are finding the solution is very large indeed, the fixed integral length with order preserved is unlikely to reduce significantly the computational effort compared with the full integral, even if $\alpha \in (0, 1)$. In the following, we demonstrate that for our understanding of the *short memory principle*, the use of fixed integral length instead of the full integral is possible and effective for Caputo derivative when $\alpha \in (0, 1)$.

For the computation of (5), we choose the fixed integral length $T$ ($t_n > T$) then the truncation error is

$$
\begin{aligned}
E &= \left| \frac{1}{\Gamma(\alpha)} \int_0^{t_n-T} ((t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1}) f(\tau, x(\tau))\, \mathrm{d}\tau \right| \\
&\leqslant \frac{M}{\Gamma(\alpha)} \left| \int_0^{t_n-T} ((t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1})\, \mathrm{d}\tau \right|
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{M}{\alpha\Gamma(\alpha)}|(h+T)^\alpha - t_{n+1}^\alpha - T^\alpha + t_n^\alpha| \\
&= \frac{M}{\Gamma(\alpha)}\left|\int_T^{T+h} z^{\alpha-1}\,\mathrm{d}z - \int_{t_n}^{t_{n+1}} z^{\alpha-1}\,\mathrm{d}z\right| \\
&= \frac{M}{\Gamma(\alpha)}|(z_1^*)^{\alpha-1}h - (z_2^*)^{\alpha-1}h| \\
&< \begin{cases} \dfrac{M}{\Gamma(\alpha)}T^{\alpha-1}h, & \alpha \in (0,1), \\ \dfrac{M}{\Gamma(\alpha)}(t_{n+1}^{\alpha-1} - T^{\alpha-1})h, & \alpha \in (1,\infty), \end{cases}
\end{aligned}
$$

where $z_1^* \in [T, T+h]$, $z_2^* \in [t_n, t_{n+1}]$, and $M = \max_{0 \leqslant \tau \leqslant t_n - T} f(\tau, x(\tau))$.

When doing the numerical computation of (2), for any given global error bound $E_{\text{global}}$ (with step length $h$) or local error bound $E_{\text{local}}$, we just need to choose $T$ such that

$$\frac{M}{\Gamma(\alpha)}T^{\alpha-1} < E_{\text{global}}, \quad \text{i.e., } T > \left(\frac{M}{\Gamma(\alpha)E_{\text{global}}}\right)^{1/(1-\alpha)}, \quad \alpha \in (0,1) \tag{7}$$

or

$$\frac{M}{\Gamma(\alpha)}T^{\alpha-1}h < E_{\text{local}}, \quad \text{i.e., } T > \left(\frac{Mh}{\Gamma(\alpha)E_{\text{local}}}\right)^{1/(1-\alpha)}, \quad \alpha \in (0,1) \tag{7$'$}$$

and

$$\frac{M}{\Gamma(\alpha)}(t_{n+1}^{\alpha-1} - T^{\alpha-1}) < E_{\text{global}}, \quad \text{i.e., } T^{\alpha-1} > t_{n+1}^{\alpha-1} - \frac{E_{\text{global}}\Gamma(\alpha)}{M}, \quad \alpha \in (1,\infty) \tag{8}$$

or

$$\frac{M}{\Gamma(\alpha)}(t_{n+1}^{\alpha-1} - T^{\alpha-1})h < E_{\text{local}}, \quad \text{i.e., } T^{\alpha-1} > t_{n+1}^{\alpha-1} - \frac{E_{\text{local}}\Gamma(\alpha)}{Mh}, \quad \alpha \in (1,\infty). \tag{8$'$}$$

In case $\alpha > 1$, in order to preserve the order of accuracy, we must choose $T$ satisfies (8) (or (8′)), it means that we will lose almost all of the computational benefits of the method of fixed integral length. But clearly from (7) (or (7′)), we know in case $\alpha \in (0, 1)$, the fixed integral length method is effective and the length $T$ is independent of the full interval of integration.

### 2.2. Nested meshes

The idea of nested memory concept introduced by Ford and Simpson in [18] can be well applied to numerically approximate (5) in case $\alpha \in (1, 2)$, thus the computational cost at each step is reduce to $\mathrm{O}(\log(h^{-1}))$ and the nested mesh scheme preserves the order of the underlying quadrature rule on which it is based [18, Theorem 1].

For (5), we decompose its integral interval in the following way:

$$[0, t_n] = [0, t_n - p^m T] \cup [t_n - p^m T, t_n - p^{m-1}T] \cup \cdots \cup [t_n - p^2 T, t_n - pT] \cup [t_n - pT, t_n], \tag{9}$$

where $T = \omega h$, $h \in \mathbb{R}^+$, $m, \omega, p \in \mathbb{N}$ and $p^m T \leqslant t_n < p^{m+1}T$.

If we denote $M_h = \{hn,\ n \in \mathbb{N}\}$ and $l_1, l_2 \in \mathbb{N}$, $l_1 > l_2$, then $M_{l_2 h} \supset M_{l_1 h}$ [18]. And the kernel of the integration (5) decays algebraically in case $\alpha \in (1, 2)$, so we can take the step length $h$ in the integral interval $[t_n - pT, t_n]$ and in the subsequent intervals $[t_n - p^2T, t_n - pT]$, $[t_n - p^3T, t_n - p^2T], \ldots, [t_n - p^mT, t_n - p^{m-1}T]$, $[0, t_n - p^mT]$, step lengths $ph, p^2h, \ldots, p^{m-1}h, p^mh$ are used, respectively. Here we note that very often $(t_n - p^mT) - 0$ cannot be divided by $p^m h$, so the integral in the interval $[0, l]$ ($l = (t_n - p^m T) - \lfloor (t_n - p^m T)/(p^m h) \rfloor \cdot (p^m h)$) is ignored, it does not destroy the computational accuracy in general.

## 3. The predictor–corrector algorithm

We carry over the idea of the predictor–corrector algorithm which is used to solve the numerical solution of (1) in [14], to the analytical formula (4) with some unavoidable modifications.

Firstly the product trapezoidal quadrature formula is applied to replace the integrals of (4), where nodes $t_j$ $(j=n, n+1)$ are taken with respect to the weight function $(t_{n+1} - \cdot)^{\alpha-1}$ for the first integral and nodes $t_j$ $(j = 0, 1, \ldots, n)$ are used with respect to the weight function $(t_{n+1} - \cdot)^{\alpha-1} - (t_n - \cdot)^{\alpha-1}$ for the second integral. That is, we employ the approximation

$$\int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau \approx \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} \widetilde{f}_{n+1}(\tau, x(\tau))\, \mathrm{d}\tau \tag{10}$$

and

$$\int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, x(\tau))\, \mathrm{d}\tau \approx \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) \widetilde{f}_n(\tau, x(\tau))\, \mathrm{d}\tau, \tag{11}$$

where $\widetilde{f}_{n+1}$ and $\widetilde{f}_n$ are the piecewise linear interpolations for $f$ with nodes and knots chosen at $t_j,\ j = n, n+1$ and $t_j,\ j = 0, 1, \ldots, n$, respectively. Using the standard technique of quadrature theory, it is found that we can write the integrals on the right hands of (10) and (11) as

$$\int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} \widetilde{f}_{n+1}(\tau, x(\tau))\, \mathrm{d}\tau = \frac{h^\alpha}{\alpha(\alpha+1)} (\alpha f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))) \tag{12}$$

and

$$\int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) \widetilde{f}_n(\tau, x(\tau))\, \mathrm{d}\tau = \frac{h^\alpha}{\alpha(\alpha+1)} \sum_{j=0}^{n} a_{j,n} f(t_j, x(t_j)),$$

where

$$a_{j,n} = \begin{cases} (n+1)^\alpha(\alpha - n) + n^\alpha(2n - \alpha - 1) - (n-1)^{\alpha+1}, & j = 0, \\ (n - j + 2)^{\alpha+1} + 3(n-j)^{\alpha+1} - 3(n - j + 1)^{\alpha+1} - (n - j - 1)^{\alpha+1}, & 1 \leqslant j \leqslant n-1, \\ 2^{\alpha+1} - \alpha - 3, & j = n. \end{cases} \tag{13}$$

So, in case $\alpha \in (1, 2)$ our corrector formula is given as

$$\begin{aligned} x_h(t_{n+1}) = x_0^{(1)} \cdot h + x_h(t_n) + \frac{h^\alpha}{\Gamma(\alpha+2)} (\alpha f(t_n, x_h(t_n)) \\ + f(t_{n+1}, x_h^{\mathrm{P}}(t_{n+1}))) + \frac{h^\alpha}{\Gamma(\alpha+2)} \sum_{j=0}^{n} a_{j,n} f(t_j, x_h(t_j)), \quad \alpha \in (1, 2), \end{aligned} \tag{14}$$

where we have used $\Gamma(\alpha)\alpha(\alpha+1) = \Gamma(\alpha+2)$, $x_h(t_j)$ $(\approx x(t_j),\ j = 1, 2, \ldots, n+1)$ are the approximate values we have already calculated (or will calculate) and $x_h^{\mathrm{P}}(t_{n+1})$ is the required preliminary approximation, the so-called predictor.

The staying problem is to determine the predictor formula, we need to calculate the value of $x_h^{\mathrm{P}}(t_{n+1})$, the idea is just not to use the unknown value $x_h(t_{n+1})$ when we compute the first integral of (4). For the first integral of (4), the product rectangle formula is used

$$\int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau \approx \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(t_n, x(t_n))\, \mathrm{d}\tau = \frac{h^\alpha}{\alpha} f(t_n, x(t_n)), \tag{15}$$

then the predictor formula is given as

$$x_h^{\mathrm{P}}(t_{n+1}) = x_0^{(1)} \cdot h + x_h(t_n) + \frac{h^\alpha}{\Gamma(\alpha+1)} f(t_n, x_h(t_n)) + \frac{h^\alpha}{\Gamma(\alpha+2)} \sum_{j=0}^{n} a_{j,n} f(t_j, x_h(t_j)), \quad \alpha \in (1,2). \tag{16}$$

Our predictor–corrector approach based on the analytical formula (4) is fully described by (14) and (16) with the weights $a_{j,n}$ being defined in (13). We notice that for the above predictor and corrector formulae they have the same term $\sum_{j=0}^{n} a_{j,n} f(t_j, x_h(t_j))$ which has the biggest computational burden $\mathrm{O}(h^{-1})$, so we minimize the computational cost in the sense that we just need to compute one times at each predictor–corrector iteration step.

**Remark 3.1.** If $\alpha \in (2, \infty)$, then the predictor and corrector formulae for solving (1) are described by, respectively,

$$x_h^{\mathrm{P}}(t_{n+1}) = \sum_{k=1}^{\lceil\alpha\rceil-1} \frac{x_0^{(k)}}{k!}(t_{n+1}^k - t_n^k) + x_h(t_n) + \frac{h^\alpha}{\Gamma(\alpha+1)} f(t_n, x_h(t_n)) + \frac{h^\alpha}{\Gamma(\alpha+2)} \sum_{j=0}^{n} a_{j,n} f(t_j, x_h(t_j)) \tag{17}$$

and

$$\begin{aligned} x_h(t_{n+1}) = {} & \sum_{k=1}^{\lceil\alpha\rceil-1} \frac{x_0^{(k)}}{k!}(t_{n+1}^k - t_n^k) + x_h(t_n) + \frac{h^\alpha}{\Gamma(\alpha+2)}(\alpha f(t_n, x_h(t_n)) \\ & + f(t_{n+1}, x_h^{\mathrm{P}}(t_{n+1}))) + \frac{h^\alpha}{\Gamma(\alpha+2)} \sum_{j=0}^{n} a_{j,n} f(t_j, x_h(t_j)), \end{aligned} \tag{18}$$

where $x_0^{(k)}$ $(k = 1, 2, \ldots, \lceil\alpha\rceil - 1)$ are initial values and the definitions of $a_{j,n}$ are given in (13).

Our following discussion focuses on reducing the computational effort of (5), that is, using nested meshes for the last sum term in our predictor and corrector formulae. For the integral (5), we decompose its integral interval as (9) and still use the product trapezoidal quadrature formula at each subinterval (the same as (11)) but with different step lengths. This idea's detailed discussion is presented in Section 2.2 of this paper

$$\begin{aligned} & \int_0^{t_n} ((t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1}) f(\tau, x(\tau))\, \mathrm{d}\tau \\ & = \left( \int_{t_n - p\omega h}^{t_n} + \sum_{i=1}^{m-1} \int_{t_n - p^{i+1}\omega h}^{t_n - p^i \omega h} + \int_0^{t_n - p^m \omega h} \right) ((t_{n+1}-\tau)^{\alpha-1} - (t_n-\tau)^{\alpha-1}) f(\tau, x(\tau))\, \mathrm{d}\tau \\ & \approx \frac{h^\alpha}{\alpha(\alpha+1)} \sum_{j=n-p\omega}^{n} b_{j,p^0,n} f(t_j, x(t_j)) \\ & \quad + \sum_{i=1}^{m-1} \frac{(p^i h)^\alpha}{\alpha(\alpha+1)} \left( \sum_{j=0}^{(p-1)\omega} b_{j,p^i,n} f(t_n - p^i(\omega+j)h, x(t_n - p^i(\omega+j)h)) \right) \\ & \quad + \frac{(p^m h)^\alpha}{\alpha(\alpha+1)} \sum_{j=0}^{\lceil n/p^m - \omega\rceil - 1} b_{j,p^m,n} f(t_n - p^m(\omega+j)h, x(t_n - p^m(\omega+j)h)), \end{aligned} \tag{19}$$

where $\lceil n/p^m - \omega\rceil$ stands for the first integer which is not less than $n/p^m - \omega$ and $p^m\omega h \leqslant t_n < p^{m+1}\omega h$ (i.e., $m = \lceil \ln((n-\omega p)/\omega + p)/\ln p\rceil - 1$) and

$$b_{j,p^0,n} = \begin{cases} (p\omega+1)^\alpha(\alpha - p\omega) + (p\omega)^\alpha(2p\omega - \alpha - 1) - (p\omega-1)^{\alpha+1}, & j = n - p\omega, \\ (n-j+2)^{\alpha+1} + 3(n-j)^{\alpha+1} - 3(n-j+1)^{\alpha+1} & \\ \quad -(n-j-1)^{\alpha+1}, & n - p\omega + 1 \leqslant j \leqslant n-1, \\ 2^{\alpha+1} - \alpha - 3, & j = n, \end{cases} \tag{20}$$

for $i = 1, 2, \ldots, m$,

$$b_{j,p^i,n} = \begin{cases} -(1/p^i + \omega)^{\alpha+1} + \omega^{\alpha+1} + (1/p^i + \omega + 1)^{\alpha+1} \\ \quad -(\omega + 1)^{\alpha+1} - ((1/p^i + \omega)^{\alpha} - \omega^{\alpha})(\alpha + 1), & j = 0, \\ (\omega + j - 1 + 1/p^i)^{\alpha+1} - (\omega + j - 1)^{\alpha+1} - 2(\omega + j + 1/p^i)^{\alpha+1} \\ \quad +2(\omega + j)^{\alpha+1} + (\omega + j + 1 + 1/p^i)^{\alpha+1} - (\omega + j + 1)^{\alpha+1}, & 1 \leqslant j \leqslant (p-1)\omega - 1, \\ (p^i\omega - 1 + 1/p^i)^{\alpha+1} - (p^i\omega - 1)^{\alpha+1} - (p^i\omega + 1/p^i)^{\alpha+1} \\ \quad +(p^i\omega)^{\alpha+1} + ((p^i\omega + 1/p^i)^{\alpha} - (p^i\omega)^{\alpha})(\alpha + 1), & j = (p-1)\omega. \end{cases} \tag{21}$$

By employing (19)–(21), we reduce the computational cost of (5) from $O(h^{-1})$ to $O(\log(h^{-1}))$ in case $\alpha \in (1, 2)$. It can be noted that we do not need to compute all the coefficients $b_{j,p^i,n}$ (because almost all of them have been computed in the previous iterations, at most there is one unknown coefficient which is necessary to compute) when performing one times predictor–corrector iteration. In general, after doing $p^m$ times predictor–corrector iterations, we require to compute one unknown coefficient.

As far as the stability properties are concerned, first the numerical computation of (5) is stable, because for

$$\int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\widetilde{f}_n(\tau, x(\tau))\,\mathrm{d}\tau = \frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{n} a_{j,n} f(t_j, x(t_j)),$$

all $a_{j,n}$ are negative and

$$\frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{n} a_{j,n} = -\frac{1}{\alpha}(h^\alpha + t_{n+1}^\alpha - t_n^\alpha),$$

if computing $f(t_j, x(t_j))$ induces an error $\varepsilon_j$, $(j = 0, 1, \ldots, n)$, then $\varepsilon_j$ arose the error

$$\begin{aligned} e_n &= \frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{n} a_{j,n} f(t_j, x(t_j)) - \frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{n} a_{j,n}(f(t_j, x(t_j)) + \varepsilon_j) \\ &= -\frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{n} a_{j,n}\varepsilon_j \leqslant -\frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{n} a_{j,n}\varepsilon = \frac{1}{\alpha}(h^\alpha + t_{n+1}^\alpha - t_n^\alpha)\varepsilon, \end{aligned}$$

where $\varepsilon = \max_{0 \leqslant j \leqslant n}|\varepsilon_j|$. If the formulae (19)–(21) are used, the stability property cannot be destroyed. Then the left stability analysis for (16), (14) and (17), (18) is same to that in classical Adams-Bashforth–Moulton scheme. One of the ways of improving the stability properties is to use the so-called $P(EC)^M E$ algorithm.

## 4. Error analysis of the predictor–corrector algorithm

Firstly, we propose several lemmas for giving the local error analysis of our predictor–corrector formulae. That is the errors which are induced by the approximations in (15), (10) and (11), respectively.

In the following error analysis, we always use the same $C$ to denote some fixed constants which may have dissimilar values at different formulae. For some different fixed constants at one formula, we employ $C_1$, $C_2$, ... to distinguish them.

The error of the product rectangle rule (15) is given as

**Lemma 4.1.**

$$\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1}(f(\tau, x(\tau)) - f(t_n, x(t_n)))\,\mathrm{d}\tau\right| \leqslant Ch^{\alpha+1},$$

*where $(\partial f(\tau, x(\tau))/\partial\tau) \in C[0, t)$ for some suitable $t$.*

**Proof.**

$$\begin{aligned}
&\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(f(\tau,x(\tau))-f(t_n,x(t_n)))\,\mathrm{d}\tau\right|\\
&\quad\leqslant\frac{\|\partial f(\tau,x(\tau))/\partial\tau\|_\infty}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(\tau-t_n)\,\mathrm{d}\tau\\
&\quad=\frac{\|\partial f(\tau,x(\tau))/\partial\tau\|_\infty}{\Gamma(\alpha)}\frac{1}{\alpha(\alpha+1)}h^{\alpha+1}\\
&\quad=\frac{\|\partial f(\tau,x(\tau))/\partial\tau\|_\infty}{\Gamma(\alpha+2)}h^{\alpha+1}\\
&\quad=Ch^{\alpha+1}\quad\text{where } C=\frac{\|\partial f(\tau,x(\tau))/\partial\tau\|_\infty}{\Gamma(\alpha+2)}.\qquad\square
\end{aligned}$$

The error in the approximation (16) is described by

**Lemma 4.2.** *If* $\partial^2 f(\tau,x(\tau))/\partial^2\tau\in C[0,t)$ *for some suitable* $t$, *then*

$$\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(f(\tau,x(\tau))-\widetilde{f}_{n+1}(\tau,x(\tau)))\,\mathrm{d}\tau\right|\leqslant Ch^{\alpha+2}.$$

**Proof.** According to the property of linear interpolation polynomials,

$$f(\tau,x(\tau))-\widetilde{f}_{n+1}(\tau,x(\tau))=f[\tau,t_n,t_{n+1}](\tau-t_n)(\tau-t_{n+1}),$$

where $f[\tau,t_n,t_{n+1}]$ is second divided differences. So,

$$\begin{aligned}
&\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(f(\tau,x(\tau))-\widetilde{f}_{n+1}(\tau,x(\tau)))\,\mathrm{d}\tau\right|\\
&\quad=\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}f[\tau,t_n,t_{n+1}](\tau-t_n)(\tau-t_{n+1})\,\mathrm{d}\tau\right|\\
&\quad=\left|\frac{1}{\Gamma(\alpha)}f[\xi,t_n,t_{n+1}]\right|\cdot\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(\tau-t_n)(\tau-t_{n+1})\,\mathrm{d}\tau,\\
&\quad=\left|\frac{1}{\Gamma(\alpha)}\frac{f''(\eta,x(\eta))}{2}\right|\cdot\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(\tau-t_n)(\tau-t_{n+1})\,\mathrm{d}\tau,\\
&\quad=\left|\frac{1}{\Gamma(\alpha)}\frac{f''(\eta,x(\eta))}{2}\right|\cdot\left(\frac{1}{\alpha}t_n^2h^\alpha+\frac{2}{\alpha(\alpha+1)}t_nh^{\alpha+1}+\frac{2}{\alpha(\alpha+1)(\alpha+2)}h^{\alpha+2}-\frac{1}{\alpha}(t_n+t_{n+1})t_nh^\alpha\right.\\
&\qquad\left.-\frac{1}{\alpha(\alpha+1)}(t_n+t_{n+1})h^{\alpha+1}+\frac{1}{\alpha}t_nt_{n+1}h^\alpha\right)\\
&\quad=\left|\frac{1}{\Gamma(\alpha)}\frac{f''(\eta,x(\eta))}{2}\right|\cdot\left(-\frac{1}{\alpha(\alpha+1)}h^{\alpha+2}+\frac{2}{\alpha(\alpha+1)(\alpha+2)}h^{\alpha+2}\right)\\
&\quad=\left|\frac{1}{\Gamma(\alpha)}\frac{f''(\eta,x(\eta))}{2}\right|\cdot\frac{1}{(\alpha+1)(\alpha+2)}h^{\alpha+2}\\
&\quad\leqslant\frac{\|f''(\eta,x(\eta))\|_\infty}{2\Gamma(\alpha)(\alpha+1)(\alpha+2)}h^{\alpha+2}\\
&\quad=Ch^{\alpha+2},
\end{aligned}$$

where $\xi,\ \eta \in [t_n, t_{n+1}]$, $C = \|f''(\eta, x(\eta))\|_\infty/(2\Gamma(\alpha)(\alpha+1)(\alpha+2))$ and in the above equalities the second integral mean value theorem and the properties of second divided differences are used. □

The error introduced by the approximation (11) is given as

**Lemma 4.3.**

$$\left|\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})\cdot(f(\tau,x(\tau))-\widetilde{f}_n(\tau,x(\tau)))\,\mathrm{d}\tau\right| \leqslant C\cdot h^{\min\{\alpha+2,3\}},$$

*where* $\partial^2 f(\tau, x(\tau))/\partial^2\tau \in C[0,t)$ *for some suitable* $t$.

**Proof.** The idea of this lemma's proof is similar to the above two lemmas, namely

$$\begin{aligned}
&\left|\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})\cdot(f(\tau,x(\tau))-\widetilde{f}_n(\tau,x(\tau)))\,\mathrm{d}\tau\right| \\
&\quad\leqslant \frac{\|f''(\eta,x(\eta))\|_\infty}{\Gamma(\alpha)}\cdot\left|\sum_{j=0}^{n-1}\int_{t_j}^{t_{j+1}}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})(\tau-t_j)(\tau-t_{j+1})\,\mathrm{d}\tau\right| \\
&\quad= \frac{\|f''(\eta,x(\eta))\|_\infty}{\Gamma(\alpha)}\cdot\left|\sum_{j=0}^{n-1}\int_{t_j}^{t_{j+1}}\tau^2\left(-\frac{1}{\alpha}\right)\cdot\mathrm{d}((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})\right. \\
&\quad\quad\left.+(t_j+t_{j+1})\int_{t_j}^{t_{j+1}}\tau\frac{1}{\alpha}\cdot\mathrm{d}((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})-\frac{1}{\alpha}t_jt_{j+1}((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})|_{\tau=t_j}^{\tau=t_{j+1}}\right| \\
&\quad= \frac{\|f''(\eta,x(\eta))\|_\infty}{\Gamma(\alpha)}\cdot\left|\sum_{j=0}^{n-1}\int_{t_j}^{t_{j+1}}\left(-\frac{1}{\alpha}\right)((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})\cdot 2\tau\,\mathrm{d}\tau\right. \\
&\quad\quad+\frac{t_j+t_{j+1}}{\alpha}\tau((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})|_{\tau=t_j}^{\tau=t_{j+1}}-\frac{t_j+t_{j+1}}{\alpha}\int_{t_j}^{t_{j+1}}((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})\,\mathrm{d}\tau \\
&\quad\quad\left.-\frac{1}{\alpha}t_jt_{j+1}((t_{n+1}-\tau)^{\alpha}-(t_n-\tau)^{\alpha})|_{\tau=t_j}^{\tau=t_{j+1}}\right| \\
&\quad= \frac{\|f''(\eta,x(\eta))\|_\infty}{\Gamma(\alpha)}\cdot\left|\sum_{j=0}^{n-1}\left\{-\frac{1}{\alpha}t_{j+1}^2(t_{n-j}^{\alpha}-t_{n-j-1}^{\alpha})+\frac{1}{\alpha}t_j^2(t_{n-j+1}^{\alpha}-t_{n-j}^{\alpha})\right.\right. \\
&\quad\quad-\frac{2}{\alpha(\alpha+1)}t_{j+1}(t_{n-j}^{\alpha+1}-t_{n-j-1}^{\alpha+1})+\frac{2}{\alpha(\alpha+1)}t_j(t_{n-j+1}^{\alpha+1}-t_{n-j}^{\alpha+1}) \\
&\quad\quad-\frac{2}{\alpha(\alpha+1)(\alpha+2)}(t_{n-j}^{\alpha+2}-t_{n-j-1}^{\alpha+2}-t_{n-j+1}^{\alpha+2}+t_{n-j}^{\alpha+2})+\frac{1}{\alpha}(t_j+t_{j+1})t_{j+1}(t_{n-j}^{\alpha}-t_{n-j-1}^{\alpha}) \\
&\quad\quad-\frac{1}{\alpha}(t_j+t_{j+1})t_j(t_{n-j+1}^{\alpha}-t_{n-j}^{\alpha})+\frac{1}{\alpha(\alpha+1)}(t_j+t_{j+1})(t_{n-j}^{\alpha+1}-t_{n-j-1}^{\alpha+1}-t_{n-j+1}^{\alpha+1}+t_{n-j}^{\alpha+1}) \\
&\quad\quad\left.\left.-\frac{t_jt_{j+1}}{\alpha}(t_{n-j}^{\alpha}-t_{n-j-1}^{\alpha}-t_{n-j+1}^{\alpha}+t_{n-j}^{\alpha})\right\}\right|
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \sum_{j=0}^{n-1} \left\{ -\frac{2}{\alpha(\alpha+1)(\alpha+2)}(t_{n-j}^{\alpha+2} - t_{n-j-1}^{\alpha+2} - t_{n-j+1}^{\alpha+2} + t_{n-j}^{\alpha+2}) \right.\right. \\
&\quad \left.\left. - \frac{h}{\alpha(\alpha+1)}(t_{n-j+1}^{\alpha+1} - t_{n-j-1}^{\alpha+1}) \right\} \right| \\
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \left\{ -\frac{2}{\alpha(\alpha+1)(\alpha+2)}(t_n^{\alpha+2} - t_{n+1}^{\alpha+2} + t_1^{\alpha+2}) - \frac{h}{\alpha(\alpha+1)}(t_{n+1}^{\alpha+1} + t_n^{\alpha+1} - t_1^{\alpha+1}) \right\} \right| \\
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \left\{ -\frac{2}{\alpha(\alpha+1)(\alpha+2)}(t_n^{\alpha+2} - t_{n+1}^{\alpha+2}) - \frac{h}{\alpha(\alpha+1)}(t_{n+1}^{\alpha+1} + t_n^{\alpha+1}) + \frac{h^{\alpha+2}}{(\alpha+1)(\alpha+2)} \right\} \right| \\
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \left\{ \frac{-h}{\alpha(\alpha+1)}(t_{n+1}^{\alpha+1} + t_n^{\alpha+1} - 2(z^*)^{\alpha+1}) + \frac{h^{\alpha+2}}{(\alpha+1)(\alpha+2)} \right\} \right| \\
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \left\{ \frac{-h}{\alpha(\alpha+1)}((t_{n+1}^{\alpha+1} - (z^*)^{\alpha+1}) - ((z^*)^{\alpha+1} - t_n^{\alpha+1})) + \frac{h^{\alpha+2}}{(\alpha+1)(\alpha+2)} \right\} \right| \\
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \left\{ -\frac{h^2}{\alpha}((z^{**})^\alpha - (\widetilde{z}^{**})^\alpha) + \frac{h^{\alpha+2}}{(\alpha+1)(\alpha+2)} \right\} \right| \\
&= \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| \left\{ -(z^{***})^{\alpha-1}h^3 + \frac{h^{\alpha+2}}{(\alpha+1)(\alpha+2)} \right\} \right| \\
&\leqslant C \cdot h^{\min\{\alpha+2,3\}},
\end{aligned}
$$

where $z^* \in [t_n, t_{n+1}]$, $z^{**} \in [z^*, t_{n+1}] \subset [t_n, t_{n+1}]$, $\widetilde{z}^{**} \in [t_n, z^*] \subset [t_n, t_{n+1}]$, $z^{***} \in [\widetilde{z}^{**}, z^{**}] \subset [t_n, t_{n+1}]$ and

$$
C = \frac{\|f''(\eta, x(\eta))\|_\infty}{\Gamma(\alpha)} \cdot \left| (z^{***})^{\alpha-1} - \frac{1}{(\alpha+1)(\alpha+2)} \right|. \qquad \square
$$

**Theorem 4.4.** *When* $\alpha > 1$, *if* $\partial^2 f(\tau, x(\tau))/\partial^2\tau \in C[0, t)$ *for some suitable* $t$, *then the local truncation error of our algorithm with the predictor and corrector formulae* (16), (14) ($\alpha \in (1, 2)$) *and* (17), (18) ($\alpha \in (2, \infty)$) *is* $O(h^3)$, *and the convergent order is* 2, *i.e.*, $\max_{j=0,1,\ldots,n+1} |x(t_j) - x_h(t_j)| = O(h^2)$.

**Proof.** This proof will be based on mathematical induction. In view of the given initial condition, the induction basis ($j = 0$) is presupposed, it has convergent order 2. Now, assume that the convergent order is 2 for $j = 0, 1, \ldots, k$, $k \leqslant n$, we have the local truncation error

$$
\begin{aligned}
&\left| x(t_{n+1}) - \left\{ \sum_{k=1}^{\lceil\alpha\rceil-1} \frac{x_0^{(k)}}{k!}(t_{n+1}^k - t_n^k) + x(t_n) \right.\right. \\
&\quad \left.\left. + \frac{h^\alpha}{\Gamma(\alpha+2)}(\alpha f(t_n, x(t_n)) + f(t_{n+1}, x_h^{\mathrm{P}}(t_{n+1}))) + \frac{h^\alpha}{\Gamma(\alpha+2)} \sum_{j=0}^{n} a_{j,n} f(t_j, x_h(t_j)) \right\} \right| \\
&= \left| \left\{ \sum_{k=1}^{\lceil\alpha\rceil-1} \frac{x_0^{(k)}}{k!}(t_{n+1}^k - t_n^k) + x(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, d\tau \right.\right.
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})f(\tau,x(\tau))\,\mathrm{d}\tau\Bigg\}-\Bigg\{\sum_{k=1}^{\lceil\alpha\rceil-1}\frac{x_0^{(k)}}{k!}(t_{n+1}^k-t_n^k)+x(t_n)\\
&+\frac{h^\alpha}{\Gamma(\alpha+2)}(\alpha f(t_n,x(t_n))+f(t_{n+1},x_h^{\mathrm{P}}(t_{n+1})))+\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x_h(t_j))\Bigg\}\Bigg|\\
&=\Bigg|\Bigg\{\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}f(\tau,x(\tau))\,\mathrm{d}\tau-\frac{h^\alpha}{\Gamma(\alpha+2)}(\alpha f(t_n,x(t_n))+f(t_{n+1},x_h^{\mathrm{P}}(t_{n+1})))\Bigg\}\\
&+\Bigg\{\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})f(\tau,x(\tau))\,\mathrm{d}\tau-\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x_h(t_j))\Bigg\}\Bigg|\\
&=\Bigg|\Bigg\{\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}f(\tau,x(\tau))\,\mathrm{d}\tau-\frac{h^\alpha}{\Gamma(\alpha+2)}(\alpha f(t_n,x(t_n))+f(t_{n+1},x(t_{n+1})))\Bigg\}\\
&+\frac{h^\alpha}{\Gamma(\alpha+2)}(f(t_{n+1},x(t_{n+1}))-f(t_{n+1},x_h^{\mathrm{P}}(t_{n+1})))\\
&+\Bigg\{\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})f(\tau,x(\tau))\,\mathrm{d}\tau-\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x(t_j))\Bigg\}\\
&+\Bigg\{\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x(t_j))-\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x_h(t_j))\Bigg\}\Bigg|\\
&\leqslant\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}(f(\tau,x(\tau))-\widetilde{f}_{n+1}(\tau,x(\tau)))\,\mathrm{d}\tau\right|+\left|\frac{h^\alpha}{\Gamma(\alpha+2)}\alpha\cdot L\cdot(x(t_{n+1})-x_h^{\mathrm{P}}(t_{n+1}))\right|\\
&+\left|\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})(f(\tau,x(\tau))-f_n(\tau,x(\tau)))\,\mathrm{d}\tau\right|\\
&+\left|\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}L\cdot(x(t_j)-x_h(t_j))\right|\\
&\leqslant C_1h^{\alpha+2}+\frac{\alpha L}{\Gamma(\alpha+2)}h^{\alpha+\min\{\alpha+1,3\}}+\frac{\alpha L}{\Gamma(\alpha+2)}h^{\min\{\alpha+2,3\}}+\left|\left(-\frac{1}{2}h^\alpha+(z^*)^{\alpha-1}h\right)\right|Lh^2\\
&<Ch^3,
\end{aligned}
$$

where $z^*\in(t_n,t_{n+1})$, Lemmas 2 and 3 in the above proof are used, and also we utilize the result $|x(t_{n+1})-x_h^{\mathrm{P}}(t_{n+1})|=\mathrm{O}(h^{\min\{\alpha+1,3\}})$ which can be proved by using Lemmas 1–3 and the similar idea to the above proof, its sketch proof is given as

$$
\begin{aligned}
&\left|x(t_{n+1})-\Bigg\{\sum_{k=1}^{\lceil\alpha\rceil-1}\frac{x_0^{(k)}}{k!}(t_{n+1}^k-t_n^k)+x(t_n)+\frac{h^\alpha}{\Gamma(\alpha+1)}f(t_n,x(t_n))+\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x_h(t_j))\Bigg\}\right|\\
&=\Bigg|\Bigg\{\sum_{k=1}^{\lceil\alpha\rceil-1}\frac{x_0^{(k)}}{k!}(t_{n+1}^k-t_n^k)+x(t_n)+\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}f(\tau,x(\tau))\,\mathrm{d}\tau
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})f(\tau,x(\tau))\,\mathrm{d}\tau\Bigg\}-\Bigg\{\sum_{k=1}^{\lceil\alpha\rceil-1}\frac{x_0^{(k)}}{k!}(t_{n+1}^k-t_n^k)+x(t_n)\\
&+\frac{h^\alpha}{\Gamma(\alpha+1)}f(t_n,x(t_n))+\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x_h(t_j))\Bigg\}\Bigg|\\
\leqslant{}&\left|\frac{1}{\Gamma(\alpha)}\int_{t_n}^{t_{n+1}}(t_{n+1}-\tau)^{\alpha-1}f(\tau,x(\tau))\,\mathrm{d}\tau-\frac{h^\alpha}{\Gamma(\alpha+1)}f(t_n,x(t_n))\right|\\
&+\left|\frac{1}{\Gamma(\alpha)}\int_0^{t_n}((t_{n+1}-\tau)^{\alpha-1}-(t_n-\tau)^{\alpha-1})f(\tau,x(\tau))\,\mathrm{d}\tau-\frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{j=0}^{n}a_{j,n}f(t_j,x_h(t_j))\right|\\
\leqslant{}&\cdots\leqslant Ch^{\min\{\alpha+1,3\}}.
\end{aligned}
$$

We have proved the local truncation error of our algorithm is $\mathrm{O}(h^3)$ when $\alpha>1$, so the convergent order is 2. □

**Lemma 4.5** (*Ford and Simpson [18, Theorem 1]*). *The nested mesh scheme preserves the order of the underlying quadrature rule on which it is based.*

Because of Theorem (4.4), Lemma (4.5) and the analysis in Section 2.2, we have

**Theorem 4.6.** *In case $\alpha\in(1,2)$, if $\partial^2 f(\tau,x(\tau))/\partial^2\tau\in C[0,t)$ for some suitable $t$, then the local truncation error of our algorithm with the predictor and corrector formulae* (22) *and* (23) *is* $\mathrm{O}(h^3)$,

$$
\begin{aligned}
x_h(t_{n+1})={}&x_0^{(1)}\cdot h+x_h(t_n)+\frac{h^\alpha}{\Gamma(\alpha+2)}(\alpha f(t_n,x_h(t_n))\\
&+f(t_{n+1},x_h^{\mathrm{P}}(t_{n+1})))+\frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=n-p\omega}^{n}b_{j,p^0,n}f(t_j,x(t_j))\\
&+\sum_{i=1}^{m-1}\frac{(p^ih)^\alpha}{\alpha(\alpha+1)}\left(\sum_{j=0}^{(p-1)\omega}b_{j,p^i,n}f(t_n-p^i(\omega+j)h,x(t_n-p^i(\omega+j)h))\right)\\
&+\frac{(p^mh)^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{\lceil n/p^m-\omega\rceil-1}b_{j,p^m,n}f(t_n-p^m(\omega+j)h,x(t_n-p^m(\omega+j)h))
\end{aligned}
\tag{22}
$$

*and*

$$
\begin{aligned}
x_h^{\mathrm{P}}(t_{n+1})={}&x_0^{(1)}\cdot h+x_h(t_n)+\frac{h^\alpha}{\Gamma(\alpha+1)}f(t_n,x_h(t_n))+\frac{h^\alpha}{\alpha(\alpha+1)}\sum_{j=n-p\omega}^{n}b_{j,p^0,n}f(t_j,x(t_j))\\
&+\sum_{i=1}^{m-1}\frac{(p^ih)^\alpha}{\alpha(\alpha+1)}\left(\sum_{j=0}^{(p-1)\omega}b_{j,p^i,n}f(t_n-p^i(\omega+j)h,x(t_n-p^i(\omega+j)h))\right)\\
&+\frac{(p^mh)^\alpha}{\alpha(\alpha+1)}\sum_{j=0}^{\lceil n/p^m-\omega\rceil-1}b_{j,p^m,n}f(t_n-p^m(\omega+j)h,x(t_n-p^m(\omega+j)h)),
\end{aligned}
\tag{23}
$$

*the convergent order is* 2, *i.e.*, $\max_{j=0,1,\ldots,n+1}|x(t_j)-x_h(t_j)|=\mathrm{O}(h^2)$, *where the meanings of* $p$, $\omega$ *and* $m$ *are same to those in* (9).

Table 1
Error behavior versus the variation of $p$ and $T$ (the definition of $p$ and $T$ are given in (9)) at time $t = 50$ with exact (analytical) value $x(50) = 2450$, fractional order $\alpha = 1.5$, step length $h = 1/80$

| $p$ | $T$ | Computing value | Absolute error | Relative error (%) |
|---|---|---|---|---|
| 1 | 1 | 2448.8 | 1.2 | 0.0489 |
| 2 | 4 | 2465.6 | 15.6 | 0.6367 |
| 2 | 3 | 2466.0 | 16.0 | 0.6530 |
| 2 | 2 | 2432.7 | 18.0 | 0.7347 |
| 3 | 4 | 2467.2 | 17.2 | 0.7020 |
| 3 | 3 | 2467.0 | 17.0 | 0.6939 |
| 3 | 2 | 2466.1 | 16.1 | 0.6571 |
| 4 | 4 | 2469.3 | 19.3 | 0.7878 |
| 4 | 3 | 2469.1 | 19.1 | 0.7796 |
| 4 | 2 | 2467.0 | 17.0 | 0.6939 |

**Remark 4.7.** When performing numerical computation, if $\alpha \in (0, 1)$ we can use the predictor–corrector approach mentioned in [14] and further uniting the nested mesh, so we have the convergent order $1 + \alpha$ and computational cost $O(h^{-1} \log h^{-1})$. If $\alpha \in (1, 2)$, combining the predictor–corrector approach in this paper and the nested mesh, i.e., with predictor and corrector formulae (22) and (23), we get the convergent order 2 and computational cost $O(h^{-1} \log h^{-1})$. Less occurring in practical application case $\alpha \in (2, \infty)$, both the predictor–corrector approaches in this paper and in [14] have the same convergent order 2 and computational cost $O(h^{-2})$.

## 5. A numerical example

The following fractional differential equation is considered [15]:

$$D_*^\alpha x(t) = \frac{2}{\Gamma(3-\alpha)} t^{2-\alpha} - x(t) + t^2 - t, \quad \alpha \in (1, 2), \tag{24}$$

with initial conditions

$$x(0) = 0, \quad x'(0) = -1.$$

Note that the exact solution to this problem is

$$x(t) = t^2 - t.$$

Table 1 shows the computing value, the absolute numerical error and the relative numerical error for different values of $p$ and $T$ which are defined in (9). The algorithm is implemented using the Matlab 6.5 on a Lenovo Pentium PC. According to the numerical results we can see computing errors are in general acceptable in engineering when the computational cost is greatly minimized, especially the computing error is not sensitive to the value of $p$. On the other hand, this numerical example also illuminates our algorithm is numerical stable.

## Acknowledgement

The author wishes to thank the anonymous referee for his/her invaluable comments. The author also wishes to thank Prof. Wu and Prof. Li for their constant encouragement.

## References

[1] O.P. Agrawal, J.A. Tenreiro Machado, Jocelyn Sabatier, Introduction, Nonlinear Dynamics 38 (2004) 1–2.
[2] E. Barkai, R. Metzler, J. Klafter, From continuous time random walks to the fractional Fokker–Planck equation, Phys. Rev. E 61 (2000) 132–138.
[3] C. Bonnet, J.R. Partington, Coprime factorizations and stability of fractional differential systems, Systems Control Lett. 41 (2000) 167–174.
[4] P.L. Butzer, U. Westphal, An Introduction to Fractional Calculus, World Scientific, Singapore, 2000.

[5] M. Caputo, Linear models of dissipation whose Q is almost frequency independent-II, Geophys. J. Roy. Astron. Soc. 13 (1967).
[6] V. Daftardar-Gejji, A. Babakhani, Analysis of a system of fractional differential equations, J. Math. Anal. Appl. 293 (2004) 511–522.
[7] W.H. Deng, C.P. Li, Synchronization of chaotic fractional Chen system, J. Phys. Soc. Jpn. 74 (2005) 1645–1648.
[8] W.H. Deng, C.P. Li, Chaos synchronization of the fractional Lü system, Physica A 353 (2005) 61–72.
[9] W.H. Deng, C.P. Li, J.H. Lü, Stability analysis of linear fractional differential system with multiple time-delays, Nonlinear Dynamics, accepted for publication.
[10] Z. Deng, V.P. Singh, L. Bengtsson, Numerical solution of fractional advection-dispersion equation, J. Hydraul. Eng. 130 (2004) 422–431.
[11] K. Diethelm, N.J. Ford, Analysis of fractional differential equations, J. Math. Anal. Appl. 265 (2002) 229–248.
[12] K. Diethelm, N.J. Ford, Numerical solution of the Bagley–Torvik equation, BIT 42 (2002) 490–507.
[13] K. Diethelm, N.J. Ford, Multi-order fractional differential equations and their numerical solution, Appl. Math. Comput. 154 (2004) 621–640.
[14] K. Diethelm, N.J. Ford, A.D. Freed, A predictor–corrector approach for the numerical solution of fractional differential equations, Nonlinear Dynamics 29 (2002) 3–22.
[15] K. Diethelm, N.J. Ford, A.D. Freed, Detailed error analysis for a fractional Adams method, Numer. Algorithms 36 (2004) 31–52.
[16] J.T. Edwards, N.J. Ford, A.C. Simpson, The numerical solution of linear multi-term fractional differential equations: systems of equations, J. Math. Anal. Appl. 148 (2002) 401–418.
[17] Z.F.A. El-Raheem, Modification of the application of a contraction mapping method on a class of fractional differential equation, Appl. Math. Comput. 137 (2003) 371–374.
[18] N.J. Ford, A.C. Simpson, The numerical solution of fractional differential equations: speed versus accuracy, Numer. Algorithms 26 (2001) 336–346.
[19] O. Heaviside, Electromagnetic Theory, Chelsea, New York, 1971.
[20] N. Heymans, I. Podlubny, Physical interpretation of initial conditions for fractional differential equations with Riemann–Liouville fractional derivatives, Rheol. Acta 37 (2005) 1–7.
[21] M. Ichise, Y. Nagayanagi, T. Kojima, An analog simulation of noninteger order transfer functions for analysis of electrode processes, J. Electroanal. Chem. 33 (1971) 253–265.
[22] S.M. Kenneth, R. Bertram, An Introduction to the Fractional Calculus and Fractional Differential Equations, Wiley-Interscience Publication, US, 1993.
[23] R.C. Koeller, Application of fractional calculus to the theory of viscoelasticity, J. Appl. Mech. (1984) 229–307.
[24] D. Kusnezov, A. Bulgac, G.D. Dang, Quantum levy processes and fractional kinetics, Phys. Rev. Lett. 82 (1999) 1136–1139.
[25] C.P. Li, W.H. Deng, D. Xu, Chaos synchronization of the Chua system with a fractional order, Physica A 360 (2006) 171–185.
[26] F. Mainardi, Fractional relaxation–oscillation and fractional diffusion-wave phenomena, Chaos Solitons Fractals 7 (1996) 1461–1477.
[27] B. Mandelbrot, Some noises with $1/f$ spectrum, a bridge between direct current and white noise, IEEE Trans. Inform. Theory 13 (1967) 289–298.
[28] D. Matignon, Stability results for fractional differential equations with applications to control processing, in: Computational Engineering in Systems and Application Multiconference, IMACS, IEEE-SMC, Lille, France, vol. 2, 1996, pp. 963–968.
[29] D. Matignon, Observer-based controllers for fractional differential equations, in: Conference on Decision and Control, SIAM, IEEE-CSS, San Diego, CA, 1997, pp. 4967–4972.
[30] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.
[31] I. Podlubny, Geometric and physical interpretation of fractional integration and fractional differentiation, Fract. Calculus Appl. Anal. 5 (2002) 367–386.
[32] N. Sugimoto, Burgers equation with a fractional derivative: hereditary effects on nonlinear acoustic waves, J. Fluid Mech. 225 (1991) 631–653.
[33] H.H. Sun, A.A. Abdelwahab, B. Onaral, Linear approximation of transfer function with a pole of fractional order, IEEE Trans. Automat. Control 29 (1984) 441–444.
[34] G.M. Zaslavsky, Chaos, fractional kinetics, and anomalous transport, Phys. Rep. 371 (2002) 461–580.