A 2ND-ORDER ONE-POINT NUMERICAL INTEGRATION SCHEME FOR FRACTIONAL ORDINARY DIFFERENTIAL EQUATIONS

WEN LI, SONG WANG AND VOLKER REHBOCK

Department of Mathematics & Statistics
Curtin University, GPO Box U1987, Perth WA 6845, Australia

(Communicated by Kok Lay Teo)

Abstract. In this paper we propose an efficient and easy-to-implement numerical method for an $\alpha$-th order Ordinary Differential Equation (ODE) when $\alpha \in (0, 1)$, based on a one-point quadrature rule. The quadrature point in each sub-interval of a given partition with mesh size $h$ is chosen judiciously so that the degree of accuracy of the quadrature rule is 2 in the presence of the singular integral kernel. The resulting time-stepping method can be regarded as the counterpart for fractional ODEs of the well-known mid-point method for 1st-order ODEs. We show that the global error in a numerical solution generated by this method is of the order $O(h^2)$, independently of $\alpha$. Numerical results are presented to demonstrate that the computed rates of convergence match the theoretical one very well and that our method is much more accurate than a well-known one-step method when $\alpha$ is small.

1. Introduction. Modelling and optimal control of many practical systems in engineering, science and economics traditionally involve Ordinary Differential Equation (ODE) systems of integer orders [2, 24, 25, 27, 28, 29, 30]. While integer order ODE systems are adequate for capturing the evolution of most standard phenomena, it has been shown over the last two decades that many complex systems in solid mechanics, viscoelastics, gas diffusion and heat conduction in porous media, signal and image processing, bio-engineering, biology, economics and financial engineering are better modelled by systems with fractional or non-integer-order differential equations (cf., for example, [3, 4, 5, 6, 7, 8, 22, 23, 26]). In particular, complex phenomena involving memory effects can be modelled more appropriately and accurately by fractional dynamical systems than by classical (integer) ones. As it is very rare that a system of practical significance can be solved analytically, one needs to be able to solve the system numerically. Clearly, an accurate and computationally efficient numerical scheme is crucial for solving fractional ODEs. This is particularly true when solving an optimal control problem involving such a system as an iterative computational procedure for computing the optimal control requires the solving the system repeatedly.

2010 Mathematics Subject Classification. Primary: 65L05, 65L20; Secondary: 49M25.

Key words and phrases. Fractional ordinary differential equation, optimal control, one-step time-stepping scheme, one-point quadrature rule, super-convergence, Caputo’s fractional derivative.

This work is supported by the AOARD Project # 15IOA095 from the US Air Force.

* Corresponding author: S. Wang.
We consider the following fractional initial value problem:

\[ 0 D^\alpha_t x(t) = f(t, x(t)), \quad t \in (0, T), \]  
\[ x(0) = x_0, \]  

where \( x_0 \) and \( T \) are positive constants. \( f \) is a known function and \( 0 D^\alpha_t x(t) \) denotes the following Caputo's \( \alpha \)th-order derivative of \( x(t) \) in \((0, T]\) with \( \alpha \in (0, 1)\):

\[ 0 D^\alpha_t x(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{x'(\tau)}{(t-\tau)^\alpha} \, d\tau. \]

In the above \( \Gamma(\cdot) \) denotes the Gamma function. Higher-order fractional initial value problems can be transformed into a system of fractional initial value problems of the form (1)-(2) and any efficient numerical method developed for (1)-(2) can be extended to a vector-valued initial value problem. There is also another representation of the \( \alpha \)th-derivative called the Riemann-Liouville fractional derivative. However, initial value problems involving the Riemann-Liouville fractional derivative can be readily transformed into (1)-(2) as demonstrated in [10, 14]. It has been proven [10, 14, 16] that solving (1)-(2) is equivalent to solving the following Volterra integral equation:

\[ x(t) = x_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, x(\tau)) \, d\tau, \quad \alpha \in (0, 1). \]

In the open literature, there are four main numerical methods for solving (3): Euler’s method [20], an Adams type predictor-corrector method [11, 12, 13, 19, 20], the \( p \)-th order method [20, 21] and the block by block method [1, 15, 17]. Euler’s method is simple, but the convergence order is only \( O(h^\alpha) \), where \( h \) denotes the mesh size of a uniform partition of \((0, T]\). The Adams type predictor-corrector method was first proposed by Diethelem et al. [11]. They showed that the convergence order of this method is \( O(h^{1+\alpha}) \). Based on the work of Diethelem et al., Deng and Li [9] have developed another improved predictor-corrector method. They proved that the order of convergence of the improved version is \( O(h^{\min(1+2\alpha, 2)}) \). Both of the schemes in [9] and [11] are single step methods. The \( p \)-th order and block-by-block methods have convergence rates of orders \( O(h^3) \) and \( O(h^{3+\alpha}) \), respectively. However, these methods are linear multiple step methods and thus computationally much more expensive than single step methods. Therefore, a question that arises is whether it is possible to design a single step method for (3) with an upper error bound better than \( O(h^{\min(1+2\alpha, 2)}) \) when \( \alpha < 0.5 \). In the integer case that \( \alpha = 1 \), the mid-point one step method has an upper error bound of order \( O(h^2) \). This method takes advantage of the property that the mid-point quadrature rule yields a ‘super-convergence’ point for numerical integration, i.e., the mid-point is the only one in an interval which gives the exact numerical integral when the integrand is a linear polynomial. Clearly, this super-convergence property of the mid-point quadrature rule does not hold true for integrals of the type (3), because the kernel becomes singular when \( \tau \) approaches \( t \). Thus, the question of what is the counterpart of the classical implicit mid-point numerical integration method for (3) arises.

In this work, we design a one-step numerical method for (3) which is easy to implement, computationally inexpensive, and results in a global error of order \( O(h^2) \) for any \( \alpha \in (0, 1) \). This method can be regarded as the counterpart for fractional ODEs of the implicit mid-point numerical integration method for first-order ODEs. In this method, we choose a numerical quadrature point in each of the sub-
intervals of a given partition judiciously so that the local approximation error is of a
higher order than that from the conventional one-point quadrature rule. This is the
counterpart of the mid-point quadrature rule in conventional numerical integration.
Based on this special numerical quadrature rule, we develop a one-step numerical
approximation of (3) based on a Taylor expansion. An error analysis of the approx-
imation is also presented. In Section 3, we propose an algorithm for implementing
the approximate equation and analyse its convergence. In Section 4 we apply our
method to several factional ODEs with known exact solutions to confirm the theo-
retical error estimate and demonstrate the superiority of our method over some of
the existing ones. Section 5 concludes the paper.

2. Approximation. For a given positive integer N, we first divide \((0,T]\) into \(N\)
sub-intervals with mesh points \(t_i = ih\) for \(i = 0, 1, \ldots, N\), where \(h = T/N\). Thus,
\((3)\) can be written as follows.

\[
x(t_i) = x_0 + \frac{1}{\Gamma(\alpha)} \int_0^{t_i} (t_i - \tau)^{\alpha - 1} f(\tau, x(\tau)) d\tau
\]

\[
= x_0 + \frac{1}{\Gamma(\alpha)} \int_0^{ih} (ih - \tau)^{\alpha - 1} f(\tau, x(\tau)) d\tau
\]

\[
= x_0 + \frac{1}{\Gamma(\alpha)} \sum_{j=1}^{i} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha - 1} f(\tau, x(\tau))d\tau.
\]

(4)

We now consider approximation of the integral on right hand side of \((3)\). Assume
that \(f(t, x(t))\) is twice continuously differentiable with respect to both \(t\) and \(x\). For
\(j = 1, 2, \ldots, i\), we apply Taylor’s theorem to \(f(\tau, x(\tau))\) at \(\tau_{ij}\) to get

\[
f(\tau, x(\tau)) = f(\tau_{ij}, x(\tau_{ij})) + K_{ij}(\tau - \tau_{ij}) + c_{ij}(\tau - \tau_{ij})^2,
\]

(5)

where \(c_{ij}\) a constant representing the 2nd derivative of \(f\) at a point between \(\tau_{ij}\) and
\(\tau\) and

\[
K_{ij} = f_\tau(\tau_{ij}, x(\tau_{ij})) + f_x(\tau_{ij}, x(\tau_{ij}))x_\tau'(\tau_{ij}).
\]

Therefore, replacing \(f(\tau, x(\tau))\) in the integrand of the last term in \((4)\) with the RHS
of \((5)\), we have, for any \(j = 1, \ldots, i,\)

\[
\frac{1}{\Gamma(\alpha)} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha - 1} f(\tau, x(\tau)) d\tau
\]

\[
= \frac{1}{\Gamma(\alpha)} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha - 1}[f(\tau_{ij}, x(\tau_{ij})) + K_{ij}(\tau - \tau_{ij})]d\tau + R_{ij}
\]

\[
= \frac{1}{\Gamma(\alpha)} f(\tau_{ij}, x(\tau_{ij})) \left[\frac{(ih - (j-1)h)^\alpha}{\alpha} - \frac{(ih - jh)^\alpha}{\alpha}\right]
\]

\[
+ K_{ij} \frac{1}{\Gamma(\alpha)} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha - 1}(\tau - \tau_{ij}) d\tau + R_{ij}
\]

\[
= \frac{h^\alpha}{\Gamma(\alpha + 1)} f(\tau_{ij}, x(\tau_{ij}))[(i - j + 1)^\alpha - (i - j)^\alpha]
\]
is given by
\begin{equation}
\int_{(j-1)h}^{jh} (ih - \tau)^{\alpha-1}(\tau - \tau_{ij})d\tau + R_{ij},
\end{equation}
where \( R_{ij} = \frac{1}{\Gamma(\alpha)} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha-1}c_{ij}(\tau - \tau_{ij})^2d\tau. \)

We now consider the choice of \( \tau_{ij} \). From (6) it is clear that \( \tau_{ij} \) should be chosen such that the second term becomes zero so that the truncation error in (6) is just \( R_{ij} \). The choice of \( \tau_{ij} \) is given in the following theorem.

**Theorem 2.1.** For any given \( i = 1, 2, \ldots, N \) and \( j = 1, 2, \ldots, i \), the unique solution to
\begin{equation}
\int_{(j-1)h}^{jh} (ih - \tau)^{\alpha-1}(\tau - \tau_{ij})d\tau = 0
\end{equation}
is given by
\begin{equation}
\tau_{ij} = h\left[(i-j+1)^{\alpha+1} - (i-j)^{\alpha+1}\right] + (\alpha + 1)\left[(i-j+1)^{\alpha}(j-1) - (i-j)^{\alpha}j\right]
\end{equation}
\begin{equation}
\frac{(\alpha + 1)(i-j+1)^{\alpha} - (i-j)^{\alpha}}{(\alpha + 1)(i-j+1)^{\alpha} - (i-j)^{\alpha}}.
\end{equation}

Furthermore, \((j-1)h < \tau_{ij} < jh\).

**Proof.** We first integrate the LHS of (7) using integration by parts as follows.
\begin{align*}
\int_{(j-1)h}^{jh} (ih - \tau)^{\alpha-1}(\tau - \tau_{ij})d\tau &= -\frac{(ih - \tau)^\alpha}{\alpha(\alpha+1)}\bigg|_{(j-1)h}^{jh} - \frac{(ih - \tau)^{\alpha+1}}{\alpha(\alpha+1)}\bigg|_{(j-1)h}^{jh} \\
&= -\frac{[(i-j)^{\alpha}j - (i-j+1)^{\alpha}(j-1)]h^{\alpha+1}}{\alpha(\alpha+1)\tau_{ij}} \\
&\quad \quad + \frac{(i-j+1)^{\alpha+1} - (i-j)^{\alpha+1}}{\alpha(\alpha+1)}h^{\alpha+1} \tag{8} \\
&= 0.
\end{align*}

From the above, we have
\begin{equation}
\frac{[(i-j+1)^{\alpha} - (i-j)^{\alpha}]\tau_{ij}}{\alpha+1} = \frac{(i-j+1)^{\alpha+1} - (i-j)^{\alpha+1}}{\alpha+1}h + [(i-j+1)^{\alpha}(j-1) - (i-j)^{\alpha}j]h.
\end{equation}
Solving this for \( \tau_{ij} \), we get (8).

We now show that \((j-1)h < \tau_{ij} < jh\). From (8),
\begin{align*}
\frac{\tau_{ij}}{h} &= \frac{[(i-j+1)^{\alpha+1} - (i-j)^{\alpha+1}] + (\alpha + 1)[(i-j+1)^{\alpha}(j-1) - (i-j)^{\alpha}j]}{(\alpha + 1)[(i-j+1)^{\alpha} - (i-j)^{\alpha}]} - j \\
&= \frac{[(i-j+1)^{\alpha+1} - (i-j)^{\alpha+1}] + (\alpha + 1)[(i-j+1)^{\alpha}(j-1) - (i-j)^{\alpha}j]}{(\alpha + 1)[(i-j+1)^{\alpha} - (i-j)^{\alpha}]} - j \\
&= \frac{(\alpha + 1)[(i-j+1)^{\alpha} - (i-j)^{\alpha}]}{(\alpha + 1)[(i-j+1)^{\alpha} - (i-j)^{\alpha}]}
\end{align*}
\begin{align*}
&= \frac{(\alpha + 1)[(i-j+1)^{\alpha} - (i-j)^{\alpha}]}{(\alpha + 1)[(i-j+1)^{\alpha} - (i-j)^{\alpha}]}
\end{align*}
\[
\frac{(i - j + 1)^{\alpha + 1} - (i - j)\alpha + 1 - (\alpha + 1)(i - j + 1)^\alpha}{(\alpha + 1)[(i - j + 1)^\alpha - (i - j)\alpha]}.
\]

To prove \(\tau_{ij} - jh < 0\), we only need to show that

\[
(i - j + 1)^{\alpha + 1} - (i - j)\alpha + 1 - (\alpha + 1)(i - j + 1)^\alpha < 0.
\]

Using the mean value theorem, we have

\[
(i - j + 1)^{\alpha + 1} - (i - j)\alpha + 1 - (\alpha + 1)(i - j + 1)^\alpha = (\alpha + 1)\xi^\alpha - (\alpha + 1)(i - j + 1)^\alpha < 0,
\]

since \(\xi \in (i - j, i - j + 1)\). Thus, we conclude that \(\tau_{ij} - jh < 0\). In a similar way, we can prove that \(\tau_{ij} - (j - 1)h > 0\).

Combining Theorem 2.1 and (4), we have the following representation for \(x(t_i)\).

\[x(t_i) = x_0 + \frac{h^\alpha}{\Gamma(\alpha + 1)} \sum_{j=1}^{i} f(\tau_{ij}, x(\tau_{ij}))[\alpha (i - j + 1)^\alpha - (i - j)^\alpha] + R_i \quad (9)\]

for \(i = 1, 2, \ldots, N\), where \(\tau_{ij}\) is given in (8) for \(j = 1, 2, \ldots, i\) and \(R_i = \sum_{j=1}^{i} R_{ij}\). Omitting the remainder \(R_i\) in (9), we have an equation approximating (4) which has the truncation error \(R_i\). An upper bound for \(R_i\) is given in the following theorem.

**Theorem 2.2.** Suppose that \(f(t, x(t))\) is twice continuously differentiable in \(t\) and \(x\). Then, for any \(i = 1, 2, \ldots, N\), the following estimate holds:

\[|R_i| \leq Ch^2, \quad (10)\]

where \(C\) denotes a positive constant independent of \(h\) and \(\alpha\).

**Proof.** For \(j = 1, 2, \ldots, i\), from the definition of \(R_i\) and Theorem 2.1 we have

\[
|R_{ij}| = \left| \frac{c_{ij}}{\Gamma(\alpha)} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha-1}(\tau - \tau_{ij})^2 d\tau \right| \\
\leq \left| \frac{c_{ij}h^2}{\Gamma(\alpha)} \int_{(j-1)h}^{jh} (ih - \tau)^{\alpha-1} d\tau \right| \\
= \left| \frac{c_{ij}h^2}{\Gamma(\alpha)} \left\{ \frac{[(i - j + 1)h]^\alpha}{\alpha} - \frac{[(i - j)h]^\alpha}{\alpha} \right\} \right|. 
\]

Since \(f\) is twice continuously differentiable, \(c_{ij}\) is bounded on \([0, T]\). Let \(c = \max_i(\max_{1 \leq j \leq i} |c_{ij}|)\). Then, from the definition of \(R_i\) and the above estimate, we have

\[
|R_i| \leq \sum_{j=1}^{i} |R_{ij}| \\
\leq \frac{ch^2}{\Gamma(\alpha)} \sum_{j=1}^{i} \left\{ \frac{[(i - j + 1)h]^\alpha}{\alpha} - \frac{[(i - j)h]^\alpha}{\alpha} \right\} \\
= \frac{ch^2}{\Gamma(\alpha)} \frac{(ih)^\alpha}{\alpha} \\
\leq \frac{ch^2}{\Gamma(\alpha)} \frac{T^\alpha}{\alpha} \\
= C_1 h^2,
\]
where \( C_1 = \frac{c}{\Gamma(\alpha)} T_1^\alpha = \frac{c}{(1+\alpha)} T_1^\alpha \). Since \( \alpha \in (0, 1) \), from [18] we have \( 2^{\alpha - 1} \leq \Gamma(1 + \alpha) \leq 1 \). Also, it is obvious that \( T_1^\alpha < \max\{1, T_1\} \) for \( \alpha \in (0, 1) \). Therefore, from the above estimates we see that [10] holds true for a \( C \) independent of both \( h \) and \( \alpha \). Thus, we have proved the theorem.

From [9] it is clear that to compute \( x(t_i) \), we need to calculate \( f(\tau_{ij}, x(\tau_{ij})) \), \( j = 1, 2, \ldots, i \). However, \( x(\tau_{ij}), j = 1, \ldots, i \), are not available directly from the scheme, although the \( \tau_{ij} \) are known. Thus, approximations for \( x(\tau_{ij}), j = 1, \ldots, i \), need to be determined. In the next section, we propose a single step numerical scheme for implementing [9] when the remainder \( R_i \) is omitted.

3. Algorithm and convergence. In this section, we propose an explicit time-stepping algorithm for approximating the solution of [9] when \( R_i \) is omitted, based on the linearization of the nonlinear term in \( x(t_i) \) in [9].

For any indices \( i \) and \( j \) satisfying \( 1 \leq j \leq i \leq N \), since \( \tau_{ij} \in (t_{j-1}, t_j) \) by Theorem 2.1, we use the following linear interpolation to approximate \( x(\tau_{ij}) \):

\[
x(\tau_{ij}) = x(t_{j-1}) + \rho_{ij}(x(t_j) - x(t_{j-1})) + \mathcal{O}(h^2),
\]

(11)

where

\[
\rho_{ij} := \frac{\tau_{ij} - t_{j-1}}{h} \in (0, 1).
\]

(12)

Then, we approximate \( f(\tau_{ij}, x(\tau_{ij})) \) as follows.

\[
f(\tau_{ij}, x(\tau_{ij})) = f(\tau_{ij}, x(t_{j-1}) + \rho_{ij}(x(t_j) - x(t_{j-1}))) + \mathcal{O}(h^2)
\]

(13)

for \( 1 \leq j \leq i \leq N \). Clearly, if we replace \( f(\tau_{ij}, x(\tau_{ij})) \) in [9] with the above expression and omit the truncation error terms of order \( \mathcal{O}(h^2) \), we define the following single step time-stepping scheme for [3]:

\[
x_i = x_0 + h_\alpha \sum_{j=1}^{i} f(\tau_{ij}, x_{j-1} + \rho_{ij}(x_j - x_{j-1})) [(i - j + 1)^\alpha - (i - j)^\alpha]
\]

(14)

\[
x_i = x_0 + h_\alpha g_i(x_0, x_1, \ldots, x_{i-1}) + h_\alpha f(\tau_{ii}, x_{i-1} + \rho_{ii}(x_i - x_{i-1}))
\]

(14)

for \( i = 1, 2, \ldots, N \), where \( \tau_{ij} \) is defined by [8] and

\[
h_\alpha = h^\alpha / \Gamma(1 + \alpha),
\]

(15)

\[
g_i(x_0, x_1, \ldots, x_{i-1}) = \begin{cases}
0, & i = 1, \\
\sum_{j=1}^{i-1} f(\tau_{ij}, x_{j-1} + \rho_{ij}(x_j - x_{j-1})) [(i - j + 1)^\alpha - (i - j)^\alpha], & i > 1.
\end{cases}
\]

(16)

The above scheme is implicit as the last term on the RHS of [14] contains a nonlinear function of \( x_i \). We can use an iterative method such as the conventional Newton’s method to solve [14] which is usually computationally more expensive than the predictor-corrector process. However, for this case, we may define an explicit single step scheme by further approximating the last term in [14] by the following truncated Taylor’s expansion:

\[
f(\tau_{ii}, x_{i-1} + \rho_{ii}(x_i - x_{i-1})) = f(\tau_{ii}, x_{i-1}) + f_x(\tau_{ii}, x_{i-1})\rho_{ii}(x_i - x_{i-1}) + \mathcal{O}(h^2).
\]

(17)

Combining [14] and [17], we propose the following explicit one-step algorithm for [3].
Algorithm A:

1. For a given positive integer \( N \), let \( t_i = ih \) for \( i = 0, 1, \ldots, N \), where \( h = T/N \).
2. For \( i = 1, 2, \ldots, N \), compute

\[
x_i = \frac{x_0 + h\alpha g_i(x_0, x_1, \ldots, x_{i-1}) + h\alpha (f(\tau_{ii}, x_{i-1}) - \rho_{ii} f_x(\tau_{ii}, x_{i-1}) x_{i-1})}{1 - h\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1})},
\]

(18)

where \( \tau_{ij}, \rho_{ij}, h_{ij} \) and \( g_i(x_0, \ldots, x_{i-1}) \) are defined in [8], [12], [13] and [16], respectively.

As can be seen later, Algorithm A provides an efficient and stable 2nd-order method for (3). Strictly speaking, Algorithm A is a multiple step method. This is because the fractional derivative is a global operator and thus all \( x_j, j < i \) are needed in order to evaluate \( x_i \) in Algorithm A. However, since the last two terms in the numerator on the RHS of (18) only involve \( x_{i-1} \), we still call it a one-step method.

Also, unlike the case of the explicit mid-point method for first-order ODEs, it is very hard to construct a one-step explicit or predictor-correct method based on (9). This is because construction of such a scheme usually requires a fractional Taylor expansion which typically has a truncation error of order \( O(h^n\alpha) \) for some positive integer \( n \). Clearly, \( n\alpha \to 0 \) as \( \alpha \) approaches 0.

Using linear interpolation theory and Taylor’s theorem, we are able to prove that, for any \( i = 1, 2, \ldots, N \), \( x_i \) generated by Algorithm A converges to \( x(t_i) \) at the rate \( O(h^2) \) when \( h \to 0^+ \). This is formalized in the following theorem.

**Theorem 3.1.** Let \( x(t) \) be the exact solution of (3)/[3]. If \( f(t, x) \) is twice continuously differentiable in \( t \) and \( x \), then there exists an \( h > 0 \) such that \( h_{ij} < h \) implies

\[
|x(t_i) - x_i| \leq Ch^2, \quad i = 1, 2, \ldots, N,
\]

(19)

where \( \{x_i\} \) is the sequence generated by Algorithm A, \( C \) is a positive constant, independent of \( h \) and \( \alpha \), and \( h_{ij} \) is defined in [13].

**Proof.** In what follows, we let \( C \) denote a generic positive constant, independent of \( h \). We now prove this theorem by mathematical induction. Firstly, we show that (19) holds for \( i = 1 \).

By (9), we have

\[
x(t_1) = x_0 + h\alpha f(\tau_{i1}, x(\tau_{i1})) + R_1.
\]

Furthermore, from (11) and (13), we have

\[
x(t_1) = x_0 + h\alpha \{f(\tau_{i1}, x_0) + f_x(\tau_{i1}, x_0)\rho_{i1}(x(t_1) - x_0) + O(h^2)\} + O(h^2) + R_1
\]

\[
= x_0 + h\alpha \{f(\tau_{i1}, x_0) - f(\tau_{i1}, x_0)\rho_{i1}x_0\} + h\alpha f_x(\tau_{i1}, x_0)\rho_{i1}x_1 + O(h^2)
\]

\[+
R_1,
\]

where \( \rho_{i1} \) is defined in (12). Solving this equation for \( x(t_1) \) and using (18) with \( i = 1 \) we have

\[
x(t_1) = \frac{x_0 + h\alpha \{f(\tau_{i1}, x_0) - f_x(\tau_{i1}, x_0)\rho_{i1}x_0\} + O(h^2) + R_1}{1 - h\alpha f_x(\tau_{i1}, x_0)\rho_{i1}}
\]

\[=

x_1 + \frac{O(h^2) + R_1}{1 - h\alpha f_x(\tau_{i1}, x_0)\rho_{i1}}.
\]
Therefore, using (10), the previous equation yields the inequality

\[ |x(t_1) - x_1| \leq \frac{Ch^2}{1 - h_\alpha f_x(\tau_{11}, x_0) \rho_{11}}. \] (20)

Note \( \rho_{11} \in (0, 1) \) by (12) and \( h_\alpha > 0 \). So, if \( f_x(\tau_{11}, x_0) \leq 0 \), from (20) we see that (19) is satisfied for \( i = 1 \). However, when \( f_x(\tau_{11}, x_0) > 0 \), we need to choose an upper bound for \( h_\alpha \) so that the denominator of (20) is bounded below by a positive constant. Clearly, for a given constant \( \sigma \in (0, 1) \), if we choose \( \bar{h}_1 := \frac{1}{\max(\rho_{11} f_x(\tau_{11}, x_0), 1)} \), then (19) is satisfied for \( i = 1 \), when \( h_\alpha \leq \bar{h}_1 \). To be more general, we choose

\[ \tilde{h} := \frac{1 - \sigma}{\max\{\max_{1 \leq i \leq N} \rho_{ii} f_x(\tau_{ii}, x_{i-1}), 1\}}. \] (21)

and when \( h_\alpha \leq \tilde{h} \), (19) is satisfied for \( i = 1 \).

We now consider the case of \( i \geq 2 \). Assume that

\[ \max_{1 \leq j \leq i-1} |x(t_j) - x_j| \leq Ch^2 \] (22)

when \( h_\alpha \leq \tilde{h} \). We now show that \( \max_{1 \leq j \leq i} |x(t_j) - x_j| \leq Ch^2 \), or equivalently, \( |x(t_i) - x_i| \leq Ch^2 \).

Using (13) and (17), we have, from (9) and (10)

\[ x(t_i) = x_0 + h_\alpha \sum_{j=1}^{i-1} \left\{ \left[ f(\tau_{ij}, x(t_{j-1}) + \rho_{ij} (x(t_j) - x(t_{j-1})) + \mathcal{O}(h^2) \right] 
\times (i-j+1)^\alpha - (i-j)^\alpha \right\} + R_i \]

\[ = x_0 \]

\[ + h_\alpha \sum_{j=1}^{i-1} f(\tau_{ij}, x(t_{j-1}) + \rho_{ij} (x(t_j) - x(t_{j-1})) + \mathcal{O}(h^2))[(i-j+1)^\alpha - (i-j)^\alpha] \]

\[ + h_\alpha \left[ f(\tau_{ii}, x(t_{i-1})) + f_x(\tau_{ii}, x(t_{i-1})) \rho_{ii} (x(t_i) - x(t_{i-1})) \right] + \mathcal{O}(h^2) + R_i \]

\[ = x_0 + A_{i-1} + B_i + \mathcal{O}(h^2). \] (23)

where

\[ A_{i-1} = h_\alpha \sum_{j=1}^{i-1} f(\tau_{ij}, x(t_{j-1}) + \rho_{ij} (x(t_j) - x(t_{j-1})) \]

\[ + \mathcal{O}(h^2))[(i-j+1)^\alpha - (i-j)^\alpha], \] (24)

\[ B_i = h_\alpha \left[ f(\tau_{ii}, x(t_{i-1})) + f_x(\tau_{ii}, x(t_{i-1})) \rho_{ii} (x(t_i) - x(t_{i-1})) \right]. \] (25)

Note that (18) can be re-written in the following form.

\[ x_i = x_0 + h_\alpha \sum_{j=1}^{i-1} f(\tau_{ij}, x_{j-1} + \rho_{ij} (x_j - x_{j-1}))[(i-j+1)^\alpha - (i-j)^\alpha] \]

\[ + h_\alpha \left[ f(\tau_{ii}, x_{i-1}) + f_x(\tau_{ii}, x_{i-1}) \rho_{ii} (x_i - x_{i-1}) \right] \]

\[ = x_0 + \tilde{A}_{i-1} + \tilde{B}_i, \] (26)
where
\[
\hat{A}_{i-1} = h_\alpha \sum_{j=1}^{i-1} f(\tau_{ij}, x_{j-1} + \rho_{ij}(x_j - x_{j-1}))[i-j+1]^\alpha - (i-j)^\alpha, \quad (27)
\]
\[
\hat{B}_i = h_\alpha [f(\tau_{ii}, x_{i-1}) + f_x(\tau_{ii}, x_{i-1})\rho_{ii}(x_i - x_{i-1})]. \quad (28)
\]
Subtracting (26) from (23) gives
\[
x(t_i) - x_i = (A_{i-1} - \hat{A}_{i-1}) + (B_i - \hat{B}_i) + O(h^2). \quad (29)
\]
Let us first estimate \(B_i - \hat{B}_i\). From (25) and (28), we have
\[
B_i - \hat{B}_i = [h_\alpha[f(\tau_{ii}, x_{i-1}) + f_x(\tau_{ii}, x_{i-1})\rho_{ii}(x_i - x_{i-1})]
- h_\alpha[f(\tau_{ii}, x_{i-1}) + f_x(\tau_{ii}, x_{i-1})\rho_{ii}(x_i - x_{i-1})]
+ h_\alpha\rho_{ii} f_x(\tau_{ii}, x_{i-1}) x_i]
- h_\alpha\rho_{ii} f_x(\tau_{ii}, x_{i-1}) x_i]
\]
\[
= h_\alpha\rho_{ii} f_x(\tau_{ii}, x_{i-1}) x_i - h_\alpha\rho_{ii} f_x(\tau_{ii}, x_{i-1}) x_i
\]
Note that \(f\) is twice continuously differentiable. Using a Taylor expansion, we get
\[
f_x(\tau_{ii}, x_{i-1}) = f_x(\tau_{ii}, x_{i-1}) + r_i, \quad (31)
\]
where
\[
r_i = f_{xx}(\tau_{ii}, \xi)(x_{i-1} - x_i)
\]
with \(\xi\) being a point between \(x(t_{i-1})\) and \(x_{i-1}\). Since \(|x(t_{i-1}) - x_{i-1}| \leq C h^2\) by Assumption (22), we have
\[
r_i = f_{xx}(\tau_{ii}, \xi) \cdot O(h^2) = O(h^2).
\]
Similarly, we have
\[
f(\tau_{ii}, x_{i-1}) - f(\tau_{ii}, x_{i-1}) = O(h^2).
\]
Using (31) and the above estimate we have, from (30),
\[
B_i - \hat{B}_i = h_\alpha O(h^2) + h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1}) (x_{i-1} - x_i) + x(t_i) r_i
- h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1}) (x_{i-1} - x_i) + x(t_i) r_i
= h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1}) (x_{i-1} - x_i) + h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1}) (x_{i-1} - x_{i-1})
+ O(h^{2+\alpha})
= h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1}) (x_{i-1} - x_i) + O(h^{2+\alpha}),
\]
since \(h_\alpha = h^\alpha / \Gamma(1 + \alpha)\) and \(|x_{i-1} - x(t_{i-1})| \leq C h^2\) from Assumption (22). Thus, from the above expression and (29), we get
\[
x(t_i) - x_i = (A_{i-1} - \hat{A}_{i-1}) + h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1}) (x(t_i) - x_i) + O(h^2).
\]
This implies
\[
x(t_i) - x_i = \frac{(A_{i-1} - \hat{A}_{i-1})}{1 - h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1})} + O(h^2)
\]
and so
\[
|x(t_i) - x_i| \leq \frac{|A_{i-1} - \hat{A}_{i-1}|}{1 - h_\alpha \rho_{ii} f_x(\tau_{ii}, x_{i-1})} + O(h^2)
\]
(32)
To estimate \(|x(t_i) - x_i|\), we need to estimate \(|A_{i-1} - \hat{A}_{i-1}|\). To simplify our notation, we let \(x_{ij} = x_{j-1} + \rho_{ij}(x_j - x_{j-1})\). We also use either the RHS or LHS
of (11) to represent the point \( x(\tau) \). From the definitions of \( A_{i-1} \) and \( \tilde{A}_{i-1} \) in (24) and (27), respectively, and using (11), we have

\[
|A_{i-1} - \tilde{A}_{i-1}| = h_{\alpha} \sum_{j=1}^{i-1} \left| f(\tau_{ij}, x(\tau_{ij})) - f(\tau_{ij}, x_{ij}) \right|[(i-j+1)^{\alpha} - (i-j)^{\alpha}]
\]

\[
\leq h_{\alpha} \sum_{j=1}^{i-1} \left| f(\tau_{ij}, x(\tau_{ij})) - f(\tau_{ij}, x_{ij}) \right|[(i-j+1)^{\alpha} - (i-j)^{\alpha}]
\]

\[
= h_{\alpha} \sum_{j=1}^{i-1} \left| f(\tau_{ij}, x(\tau_{ij})) - f(\tau_{ij}, x_{ij}) \right|[(i-j+1)^{\alpha} - (i-j)^{\alpha}],
\]

since \( z^\alpha \) is an increasing function of \( z \) for \( \alpha \in (0, 1) \). Because \( f \) is twice continuously differentiable, we have (recall that \( C \) is a generic positive constant, independent of \( h \))

\[
|f(\tau_{ij}, x(\tau_{ij})) - f(\tau_{ij}, x_{ij})| \leq C|x(\tau_{ij}) - x_{ij}|
\]

\[
= C\left[ |x(t_{j-1}) + \rho_{ij}(x(t_j) - x(t_{j-1})) + O(h^2)|
\right.
\]

\[
- |x_{j-1} + \rho_{ij}(x_j - x_{j-1})|
\]

\[
= C\left[ |x(t_{j-1}) - x_{j-1}| + \rho_{ij}|x(t_j) - x_j|
\right.
\]

\[
+ \rho_{ij}|x_{j-1} - x(t_{j-1})| + O(h^2)
\]

\[
\leq C \left( |x(t_{j-1}) - x_{j-1}| + |x(t_j) - x_j| + |x(t_{j-1}) - x_{j-1}| \right)
\]

\[
+ O(h^2),
\]

since \( \rho_{ij} \in (0, 1) \). In the above we have used (11). Thus, from Assumption (22), we have

\[
|f(\tau_{ij}, x(\tau_{ij})) - f(\tau_{ij}, x_{\tau_{ij}})| \leq Ch^2.
\]

Replacing \( |f(\tau_{ij}, x(\tau_{ij})) - f(\tau_{ij}, x_{\tau_{ij}})| \) in (33) with the above upper bound, we have

\[
|A_{i-1} - \tilde{A}_{i-1}| \leq h_{\alpha}Ch^2 \sum_{j=1}^{i-1} [(i-j+1)^{\alpha} - (i-j)^{\alpha}]
\]

\[
= h^\alpha \Gamma(\alpha + 1)Ch^2(i^\alpha - 1)
\]

\[
\leq h^\alpha \Gamma(\alpha + 1)Ch^2N^\alpha
\]

\[
= C \frac{h^\alpha \Gamma(\alpha + 1)}{\Gamma(\alpha + 1)} h^2(hN)^\alpha
\]

\[
= C \frac{h^2T^\alpha}{\Gamma(\alpha + 1)}
\]

\[
\leq Ch^2.
\]

Combining the above error bound with (32), we have

\[
|x(t_i) - x_i| \leq \frac{Ch^2}{\left| 1 - h_{\alpha} \rho_{ii} f_x(\tau_{ii}, x_{i-1}) \right|}.
\]
Therefore, when \(h_{\alpha} \leq \bar{h}\), where \(\bar{h}\) is defined in (21), we have

\[
|x(t_i) - x_i| \leq C h^2 \left| 1 - h_{\alpha} \rho_{ii} f_x(\tau_{ii}, x_{i-1}) \right| \leq \frac{C h^2}{\sigma}
\]

for a given \(\sigma > 0\).

Careful readers may have noticed the positive constant \(C\) used in the above proof is independent of \(h\), but a function of \(T^\alpha / \Gamma(1 + \alpha)\). However, in the proof of Theorem 2.2, we have shown that \(T^\alpha / \Gamma(1 + \alpha)\) is bounded above by a positive constant, independent of \(\alpha\). Thus, we have proved the theorem.

We comment that in the above theorem we have established an upper error bound of order \(O(h^2)\) for the numerical approximation to (1)–(2) generated by our proposed single-step Algorithm A, while all of the existing single-step methods proposed in \([9, 11, 12, 13, 19, 20]\) have the drawback that their rates of convergence approach \(O(h)\) as \(\alpha\) decreases.

We also comment that Algorithm A is a linearized form of our implicit method (14). This linearization does not affect the \(O(h^2)\)-order rate of convergence of (14). In other words, our explicit method represented in Algorithm A performs only one Newton iteration for (14) as performing more Newton iterations will not increase the accuracy of the numerical method due to the discretization errors.

4. Numerical Results. In this section, we solve two examples using our proposed method.

Example 1. Consider the following fractional differential equation

\[
\begin{align*}
o D_t^\alpha x(t) & = \frac{\Gamma(5)}{\Gamma(5 - \alpha)} t^{4 - \alpha} - x(t) + t^4, & t \in (0, 1], \\
x(0) & = 0.
\end{align*}
\]

The exact solution is

\[x(t) = t^4.\]

This test problem is taken from \([9]\) and it is solved by Algorithm A for various values of \(\alpha\) and mesh sizes \(h\). The computed errors \(E_{h_k} = \max_{1 \leq i \leq \tilde{h}_k} |x_i - x(t_i)|\) for \(\tilde{h}_k = 1/(2^k \times 10), k = 0, 1, ..., 6\) and the chosen values of \(\alpha\) are listed in Table 1. To estimate the rates of convergence, we calculate \(\log_2(E_{h_k}/E_{h_{k+1}})\) for \(k = 0, 1, ..., 5\) and the computed rates of convergence are also listed in Table 1. From the table we see that the computed rates of convergence are very close to the theoretical one in Theorem 3.1. In Table 1, we also compare our results with those obtained by the method in \([9]\). The latter method is a single-step predictor-corrector method with a theoretical rate of convergence of order \(O(h^{\min(1 + 2\alpha, 2)})\). The rates of convergence of our method are much higher than those of the method in \([9]\) for \(\alpha = 0.1\). From the table we also see that the absolute errors from our method are much smaller than those in \([9]\) unless \(\alpha\) is close to 1 in which case classic methods apply. Clearly, our proposed method is superior to that in \([9]\). In addition, one can expect that a predictor-corrector method should be computationally more expensive than our method.

From Table 1, we see that when \(\alpha\) is close to zero, the computed rate of convergence is slight worse than \(O(h^2)\). This may be because the constant \(C\) in (19) contains the term \(1/\Gamma(1 + \alpha)\), as noted in the previous section. However, when \(h\) decreases, the rate of convergence of our method increases.
Table 1. Maximum Errors and Convergence Rates for Example 1.

| $h$     | Our results | Results from [9] | Our results | Results from [9] |
|---------|-------------|------------------|-------------|------------------|
|         | $\alpha=0.1$ | Order   | $\alpha=0.1$ | Order   | $\alpha=0.3$ | Order |
| 1/10    | 4.06e-3     | -    | 0.364      | -    | 7.67e-3  | -    |
| 1/20    | 1.11e-3     | 1.86  | 0.170      | 1.10  | 2.00e-3  | 1.94  |
| 1/40    | 3.02e-4     | 1.89  | 7.13e-2    | 1.26  | 5.17e-4  | 1.95  |
| 1/80    | 8.08e-5     | 1.90  | 2.88e-2    | 1.31  | 1.32e-4  | 1.97  |
| 1/160   | 2.14e-5     | 1.92  | 1.15e-2    | 1.32  | 3.37e-5  | 1.97  |
| 1/320   | 5.65e-6     | 1.93  | 4.64e-3    | 1.31  | 8.55e-6  | 1.98  |
| 1/640   | 1.47e-6     | 1.93  | 1.88e-3    | 1.30  | 2.16e-6  | 1.98  |
| 1/1280  | 3.84e-7     | 1.94  | -          | -     | 5.46e-7  | 1.99  |

| $h$     | Our results | Results from [9] | Our results | Results from [9] |
|---------|-------------|------------------|-------------|------------------|
|         | $\alpha=0.5$ | Order   | $\alpha=0.5$ | Order   | $\alpha=0.9$ | Order |
| 1/10    | 8.49e-3     | -    | 0.0355     | -    | 7.88e-3  | -    |
| 1/20    | 2.16e-3     | 1.98  | 0.00879    | 2.01  | 1.97e-3  | 2.00  |
| 1/40    | 5.43e-4     | 1.99  | 2.16e-3    | 2.03  | 4.93e-4  | 2.00  |
| 1/80    | 1.37e-4     | 1.99  | 5.31e-4    | 2.02  | 1.23e-4  | 2.00  |
| 1/160   | 3.43e-5     | 2.00  | 1.31e-4    | 2.02  | 3.08e-5  | 2.00  |
| 1/320   | 8.58e-6     | 2.00  | 3.24e-5    | 2.02  | 7.70e-6  | 2.00  |
| 1/640   | 2.15e-6     | 2.00  | 8.03e-6    | 2.01  | 1.92e-6  | 2.00  |
| 1/1280  | 5.38e-7     | 2.00  | -          | -     | 4.81e-7  | 2.00  |

Example 2. Consider the following fractional differential equation

\[
_0D_t^\alpha x(t) = \frac{\Gamma(4 + \alpha)}{6} t^3 + t^{3+\alpha} - x(t), \quad t \in (0, 1],
\]

The exact solution is \( x(t) = t^{3+\alpha} \). This example is from [1] and it is solved using Algorithm A for various values of \( h \) and \( \alpha \). The computed errors and rates of convergence are listed in Table 2 from which we see that the computed rates of convergence are close to the theoretical one in Theorem 3.1.

Table 2. Maximum Errors and Convergence Rates for Example 2.

| $h$     | $\alpha=0.1$ | Order | $\alpha=0.3$ | Order | $\alpha=0.5$ | Order | $\alpha=0.9$ | Order |
|---------|-------------|-------|-------------|-------|-------------|-------|-------------|-------|
| 1/10    | 2.37e-3     | -     | 5.08e-3     | -     | 6.40e-3     | -     | 7.50e-3     | -     |
| 1/20    | 6.45e-4     | 1.88  | 1.31e-3     | 1.95  | 1.62e-3     | 1.98  | 1.88e-3     | 2.00  |
| 1/40    | 1.73e-4     | 1.90  | 3.39e-4     | 1.96  | 4.08e-4     | 1.99  | 4.69e-4     | 2.00  |
| 1/80    | 4.60e-5     | 1.91  | 8.65e-5     | 1.97  | 1.03e-4     | 1.99  | 1.17e-4     | 2.00  |
| 1/160   | 1.21e-5     | 1.92  | 2.20e-5     | 1.98  | 2.57e-5     | 2.00  | 2.93e-5     | 2.00  |
| 1/320   | 3.18e-6     | 1.93  | 5.58e-6     | 1.98  | 6.44e-6     | 2.00  | 7.32e-6     | 2.00  |
| 1/640   | 8.30e-7     | 1.94  | 1.41e-6     | 1.98  | 1.61e-6     | 2.00  | 1.83e-6     | 2.00  |
| 1/1280  | 2.15e-7     | 1.95  | 3.55e-7     | 1.99  | 4.04e-7     | 2.00  | 4.57e-7     | 2.00  |

Example 3. Consider the following fractional differential equation

\[
_0D_t^\alpha x(t) = \frac{\Gamma(4 + \alpha)}{6} t^3 + t^{4(3+\alpha)} - x^4(t), \quad t \in (0, 1],
\]
\[ x(0) = 0. \]

The exact solution is also \( x(t) = t^{3+\alpha} \). Note that the RHS of the above equation is nonlinear in both \( t \) and \( x \). This example is solved using Algorithm A for various values of \( h \) and \( \alpha \) and the computed errors and rates of convergence are listed in Table 3. From the table we see that the computed order of convergence is greater than 2 when \( \alpha \) is small.

**Table 3. Maximum Errors and Convergence Rates for Example 3.**

| \( h \)  | \( \alpha=0.1 \) | Order | \( \alpha=0.3 \) | Order | \( \alpha=0.5 \) | Order | \( \alpha=0.9 \) | Order |
|---------|-----------------|-------|-----------------|-------|-----------------|-------|-----------------|-------|
| 1/10    | 6.19e-2         | -     | 4.27e-2         | -     | 2.56e-2         | -     | 3.70e-3         | -     |
| 1/20    | 1.69e-2         | 1.87  | 1.03e-2         | 2.05  | 5.70e-3         | 2.17  | 6.90e-4         | 2.42  |
| 1/40    | 4.42e-3         | 1.93  | 2.368e-3        | 2.13  | 1.10e-3         | 2.37  | 1.88e-4         | 1.88  |
| 1/80    | 1.10e-3         | 2.00  | 5.055e-4        | 2.23  | 1.85e-4         | 2.57  | 5.92e-5         | 1.67  |
| 1/160   | 2.65e-4         | 2.05  | 1.025e-4        | 2.30  | 2.63e-5         | 2.82  | 2.11e-5         | 1.49  |
| 1/320   | 6.26e-5         | 2.08  | 2.00e-5         | 2.36  | 2.63e-6         | 3.32  | 6.19e-6         | 1.77  |
| 1/640   | 1.46e-5         | 2.10  | 3.77e-6         | 2.41  | 5.30e-7         | 2.31  | 1.68e-6         | 1.88  |
| 1/1280  | 3.41e-6         | 2.10  | 6.86e-7         | 2.46  | 1.55e-7         | 1.77  | 4.37e-7         | 1.94  |

To summarise, from Tables 1–3 we see that, although computed convergence rates fluctuate for different values of \( \alpha \) and \( h \), when \( h^\alpha \) is small enough, the convergence order is 2 confirming our theoretical analysis. This can also be seen from the last row of each of the tables corresponding to \( h = 1/1280 \). All the errors are of the magnitude \( h^2 \approx 6 \times 10^{-7} \). To check the robustness of our method in \( \alpha \), we have also solved Example 3 for \( \alpha = 10^{-i}, i = 1, \ldots, 8 \) and the computed results show that the orders of convergence are roughly 2, particularly when \( h \) is small. The robustness of our method may provide an effective way for solving problems with algebraic constraints, or differential algebraic equations. We will discuss this in a future paper.

5. **Conclusion.** In this paper, we proposed a new numerical method based on Taylor’s theorem and linear interpolation for solving fractional differential equations. The proposed method is simple and easy to use. We have proved that the convergence order of the method is 2. The numerical results confirm our theoretical analysis.

**Acknowledgement.** This work is supported by the AOARD Project #15IOA095 from the US Air Force.

**REFERENCES**

[1] J. Cao and C. Xu, A high order scheme for the numerical solution of the fractional ordinary differential equations, *Journal of Computational Physics*, **238** (2013), 154–168.

[2] S. Campbell and P. Kunkel, Solving higher index DAE optimal control problems, *Numerical Algebra, Control & Optimization*, **6** (2016), 447–472.

[3] A. Cartea and D. del-Castillo-Negrete, Fractional diffusion models of option prices in markets with jumps, *Physica A: Statistical Mechanics and its Applications*, **374** (2007), 749–763.

[4] W. Chen and S. Wang, A penalty method for a fractional order parabolic variational inequality governing American put option valuation, *Comp. Math. With Appl.*, **67** (2014), 77–90.
[5] W. Chen and S. Wang, A finite difference method for pricing European and American options under a geometric Levy process. *Journal of Industrial & Management Optimization*, 11 (2015), 241–264.

[6] W. Chen and S. Wang, 2nd-Order FDM for a 2D fractional Black-Scholes equation, in *Numerical Analysis and Its Applications. NAA 2016* (eds. Dimov I., Farag I., Vulkov L.), Lecture Notes in Computer Science, Springer, 10187 (2017), 46–57.

[7] W. Chen and S. Wang, A power penalty method for a 2D fractional partial differential linear complementarity problem governing two-asset American option pricing. *Applied Mathematics and Computation*, 305 (2017) 174–187.

[8] C. F. M. Coimbra, Mechanics with variable-order differential operators. *Ann. Phys. (Leipzig)*, 12 (2003), 692–703.

[9] W. Deng and C. Li, Numerical schemes for fractional ordinary differential equations. *Numerical Modeling*, Dr. Peep Miidla (Ed.), InTech, 2012.

[10] K. Diethelm and N. J. Ford, Analysis of fractional differential equations. *Journal of Mathematical Analysis and Applications*, 265 (2002), 229–248.

[11] K. Diethelm, N. J. Ford and A. D. Freed, A predictor corrector approach for the numerical solution of fractional differential equation. *Nonlinear Dynam.*, 29 (2002), 2–22.

[12] K. Diethelm, N. J. Ford and A. D. Freed, Detailed error analysis for a fractional Adams method. *Numer. Algorithms*, 36 (2004), 31–52.

[13] K. Diethelm, N. J. Ford, A. D. Freed and Yu. Luchko, Algorithms for the fractional calculus: a selection of numerical methods. *Comput. Method appl. Mech. Engrg.*, 194 (2005), 743–773.

[14] B. Guo, X. Pu and F. Huang, *Fractional Partial Differential Equations and Their Numerical Solutions* World Scientific, 2015.

[15] H. Huang, Y. Tang and L. Vazquez, Convergence analysis of a block-by block method for fractional differential equation. *Numer. Math. Theor. Methods Appl.*, 5 (2012), 229–241.

[16] A. A. Kilbas and S. A. Marzan, Cauchy problem for differential equation with Caputo derivative. *Fractional Calculus and Applied Analysis*, 7 (2014), 297–321.

[17] K. Kumar and O. F. Agrawal, An approximate method for numerical solution of fractional differential equations. *Signal Process.*, 86 (2006), 2602–2610.

[18] A. Laforgia and P. Natalini, Exponential, gamma and polygamma functions: Simple proofs of classical and new inequalities. *J. Math. Anal. Appl.*, 407 (2013), 459–504.

[19] C. Li and C. Tao, On the fractional Adams method. *Comput. Math. Appl.*, 58 (2009), 1573–1588.

[20] C. Li and F. Zeng, The finite difference methods for fractional ordinary differential equations. *Numerical Functional Analysis and Optimization*, 34 (2013), 149–179.

[21] R. Lin and F. Liu, Fractional high order methods for the nonlinear fractional ordinary differential equation. *Nonlinear Analysis*, 66 (2007), 856–869.

[22] R. Magin, *Fractional Calculus in Bioengineering*, Begell House Inc., Redding, CT, 2006.

[23] F. Mainardi, *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models* Imperial College Press, London, 2010.

[24] S. Mohd Mahali, S. Wang and X. Lou, Determination of effective diffusion coefficients of drug delivery devices by a state observer approach. *Discrete and Continuous Dynamical Systems Series B*, 17 (2011), 1119–1136.

[25] S. Mohd Mahali, S. Wang and X. Lou, Estimation of effective diffusion coefficients of drug delivery devices in a flow-through system. *Journal of Engineering Mathematics*, 87 (2014), 139–152.

[26] M. D. Ortigueria and J. A. T. Machodo, Special section: Fractional calculus applications in signals and systems. *Signal Processing*, 86 (2006), 2503–3094.

[27] B. Shen, X. Wang and C. Liu, Nonlinear state-dependent impulsive system in fed-batch culture and its optimal control. *Numerical Algebra, Control & Optimization*, 5 (2015), 369–380.
[28] S. Sorokin and M. Staritsyn, Feedback necessary optimality conditions for a class of terminal-constrained state-linear variational problems inspired by impulsive control, Numerical Algebra, Control & Optimization, 7 (2017), 201–210.

[29] M. Y. Tan, L. S. Jennings and S. Wang, Analysing human periodic walking at different speeds using parametrization enhancing transform in dynamic optimization, Pacific Journal of Optimization, 12 (2016), 557–586.

[30] Y. Wang, C. Yu and K. L. Teo, A new computational strategy for optimal control problem with a cost on changing control, Numerical Algebra, Control & Optimization, 6 (2016), 339–364.

Received January 2017; 1st revision June 2017; final revision July 2017.

E-mail address: wen.li@curtin.edu.au
E-mail address: song.wang@curtin.edu.au
E-mail address: V.Rehbock@curtin.edu.au