Fractional Order Runge-Kutta Methods

R. Ghaffari\(^1\), F. Ghoreishi\(^2\) and Nasser Saad\(^3\)**

\(^1\) Faculty of Mathematics, K. N. Toosi University of Technology, Tehran, Iran.; r.ghaffari85@gmail.com
\(^2\) Faculty of Mathematics, K. N. Toosi University of Technology, Tehran, Iran.; ghoreishif@kntu.ac.ir
\(^3\) School of Mathematical and Computational Sciences, University of Prince Edward Island, Charlottetown, Canada.; nsaad@upei.ca

\(^*\) These authors contributed equally to this work.

March 6, 2023

Abstract

This paper investigates a new class of fractional order Runge-Kutta (FORK) methods for numerical approximation to the solution of fractional differential equations (FDEs). By using the Caputo generalized Taylor formula for Caputo fractional derivative, we construct explicit and implicit FORK methods, as the well-known Runge-Kutta schemes for ordinary differential equations. In the proposed method, due to the dependence of fractional derivatives to a fixed base point \(t_0\), we had to modify the right-hand side of the given equation in all steps of the FORK methods. Some coefficients for explicit and implicit FORK schemes are presented. The convergence analysis of the proposed method is also discussed. Numerical experiments clarify the effectiveness and robustness of the method.

Keywords Fractional differential equations; Caputo fractional derivative; Convergence analysis; Consistency; Stability analysis.

MSC2010: 26A33; 41A25; 65L03.

1. Introduction

In recent years, the numerical approximation for the solutions of FDEs has attracted increasing attention in many fields of applied sciences and engineering \[1\]-\[3\]. It is common for FDEs to be used in formulating many problems in applied mathematics. Developing numerical methods for fractional differential problems is necessary and important because analytic solutions are usually challenging to obtain. Moreover, it is necessary to develop numerical methods that are highly accurate and easy to use.

It is well known that fractional derivatives have different definitions; the most common and important ones in applications are the Riemann–Liouville
and Caputo fractional derivatives. Models describing physical phenomena usually prefer the use of the Caputo derivative. One of the reasons is that the Riemann–Liouville derivative needs initial conditions containing the limit values of the Riemann–Liouville fractional derivative at the origin of time. In contrast, the initial conditions for Caputo derivatives are the same as for integer-order differential equations. Therefore, using the Caputo derivative, there is a clear physical interpretation of the prescribed data; see [1, 4, 5].

Numerous research papers have been published on numerical methods for FDEs. Many researchers considered the trapezoidal method, predictor-corrector method, extrapolation method, and spectral method [6–16]. Some of these methods discretize fractional derivatives directly. As an example, the L1 formula was created by a piecewise linear interpolation approximation for the integrand function on each small interval [17, 18]. In [19], the authors applied quadratic interpolation approximation using three points to approximate the Caputo fractional derivative, while in [20], a technique based on the block-by-block approach was presented. This technique became a common method for equations with integral operators. In [21], Caputo fractional differentiation was approximated by a weighted sum of the integer order derivatives of functions. In [22], several numerical algorithms were proposed to approximate the Caputo fractional derivatives by applying higher-order piecewise interpolation polynomials and the Simpson method to design a higher-order algorithm.

These methods are appropriate options if the resulting system of equations, generated from the numerical method, is linear and well-conditioned. However, they present a high computational cost when the problem we are solving is badly conditioned or nonlinear. In light of the above discussion and the analysis of other methods for FDEs, despite many papers on numerical methods for FDEs, there are still insufficient efficient numerical approaches for such equations. Therefore, further studies are still in demand. In this case, step-by-step methods such as the Runge–Kutta method are a good option. They are favored due to their simplicity in both calculation and analysis.

Several authors have used Runge–Kutta methods to solve ordinary, partial differential, and integral equations [23–30]. Lubich and others have done some fundamental works regarding Runge–Kutta methods for Volterra integral equations [28–30]. They used the order conditions to derive various Runge–Kutta methods.

One of the efficient implicit Runge–Kutta methods for the numerical approximation of some linear partial differential equations is the Rosenbrock procedure. It is a class of semi-implicit Runge–Kutta methods for the numerical solution of some stiff systems of ODEs. In Osterman and Roche’s papers [31, 32], the authors apply the Rosenbrock methods to solve linear partial differential equations, obtaining a sharp lower bound for the order of convergence. They show that the order of convergence is, in general, fractional. So, for the numerical solution of some fractional linear partial differential equations, we can construct fractional Rosenbrock-type methods, in which a special type of fractional semi-implicit Runge–Kutta method could be considered.

This paper introduces a new class of fractional order Runge–Kutta methods
for numerical approximation to the solution of FDEs. Using the Caputo generalized Taylor series formula for the Caputo fractional derivative, we construct explicit and implicit FORK methods comparable to the well-known Runge–Kutta schemes for ordinary differential equations.

The remainder of the paper is organized as follows. In Section 2, we review some definitions and properties of fractional calculus. We propose new explicit and implicit FORK methods for solving FDEs in Sections 3 and 4. In Section 5, the theoretical analysis of the convergency, stability, and consistency of the proposed methods is presented. Finally, in Section 6, some numerical examples demonstrate the effectiveness of the methods proposed. Also, in Appendix A, two Mathematica computer programming codes are given.

2. Preliminaries

In this section, we briefly state definitions of fractional integral and Caputo fractional derivative and some of them properties. For further information about fractional calculus and some other definitions of fractional derivatives, we refer the interested readers to the [1, 4, 33].

Definition 2.1. The Riemann Liouville fractional integral operator of order $\alpha > 0$ for a function $f(x) \in L_1[a, b]$ with $a \geq 0$ is defined as

$$J_0^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x-t)^{\alpha-1} f(t) \, dt, \quad x \in [a, b], \quad J_0^0 f(x) = f(x).$$

where $L_1[a, b] = \{f \mid f$ is a measurable function on $[a, b]$ and $\int_a^b |f(x)| \, dx < \infty\}$, $\Gamma$ is Gamma function and $\Gamma(\alpha + 1) = \alpha!$.

Definition 2.2. The Caputo fractional derivatives of order $\alpha > 0$ of a function $f(x) \in L_1[a, b]$ with $a \geq 0$ is defined as

$$c_a^\alpha D_t^n f(x) = J_0^{\alpha-n} D_t^n f(x) = \begin{cases} \frac{1}{\Gamma(n-\alpha)} \int_a^x (x-t)^{n-\alpha-1} D_t^n f(t) \, dt, & n-1 < \alpha < n, \quad n \in \mathbb{N}, \\ D_t^n f(x), & \alpha = n. \end{cases}$$

Theorem 2.3. (Generalized Taylor formula for Caputo fractional derivative [34]). Suppose that $(c_a^\alpha D_t^k)^{f(x)} \in C(a, b)$ for $k = 0, 1, \ldots, n+1$, where $0 < \alpha \leq 1$, then for $\forall x \in [a, b]$ there exist $\xi \in (a, x)$ such that

$$f(x) = \sum_{i=0}^{n} \frac{(x-a)^{i\alpha}}{\Gamma(i\alpha + 1)} (c_a^\alpha D_t^i)^{f(a)} + \frac{(x-a)^{(n+1)\alpha}}{\Gamma((n+1)\alpha + 1)} (c_a^\alpha D_t^{(n+1)})^{f}(\xi),$$

where $(c_a^\alpha D_t^\alpha)^{n} = c_a^\alpha D_t^\alpha \cdots c_a^\alpha D_t^\alpha$ (n times).

There are also two important functions in fractional calculus. They are direct generalization of the exponential series which play important roles in the solution of FDEs and stability analysis.
Definition 2.4. The Mittag-Leffler function is defined as
\[ E_\alpha(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha k + 1)}, \quad \Re(\alpha) > 0, x \in \mathbb{C}. \]
Also, the two-parameters Mittag-Leffler function is defined by
\[ E_{\alpha,\beta}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(\alpha k + \beta)}, \quad \Re(\alpha) > 0, \beta \in \mathbb{C}, x \in \mathbb{C}, \]
We note that \[ E_\alpha(x) = E_{\alpha,1}(x) \] and \[ E_1(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} = \exp(x). \]

3. Fractional order Runge-Kutta methods

In this section, a new class of FORK methods for numerical solution of FDE is investigated.

Consider the following FDE with \(0 < \alpha \leq 1\):
\[ C_0^\alpha D_t^\alpha y(t) = f(t, y(t)), \quad t \in [t_0, T], \]
\[ y(t_0) = y_0. \quad (3) \]
where \(y(t) \in C[t_0, T]\) and \(f(t, y(t)) \in C[t_0, T] \times \mathbb{R}\). \(t_0\) is called base point of fractional derivative.
We set \(t_n = t_0 + nh, \ n = 0, 1, \ldots, N^m\), where \(h = \frac{T - t_0}{N^m}\) is the step size, \(N\) is a positive integer and in section 5 we will prove that \(m \geq \frac{1}{\alpha}\).
For the existence and uniqueness of solution of the FDE (3), consider the following theorem from [4].

Theorem 3.1. Let \(\alpha > 0, y_0 \in \mathbb{R}, K > 0\) and \(T > 0\) and also let the function \(f : G \to \mathbb{R}\) be continuous and fulfill a Lipschitz condition with respect to the second variable, i.e.
\[ |f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \]
with some constant \(L > 0\) independent of \(t, y_1\) and \(y_2\). Define
\(G = \{(t, y) : t \in [0, T], |y - y_0| \leq K\}, \ M = \text{Sup}_{(t, z) \in G}|f(t, z)|\) and
\[ T^* = \begin{cases} T, & M = 0, \\
\min\{T, (K \Gamma(\alpha + 1)/M)^{\frac{1}{\alpha}}\}, & \text{else.} \end{cases} \]
Then, there exists a uniquely function \(y \in C[0, T^*]\) solving the initial-value problem (3).
In the sequel, we assume that \( f(t, y) \) has continuous partial derivatives with respect to \( t \) and \( y \) of as high an order as we desire.

Now, we introduce a \( s \)-stage explicit fractional order Runge-Kutta (EFORK) method for FDEs, which is discussed completely with 2 and 3 stages.

**Definition 3.2.** A family of \( s \)-stage EFORK methods is defined as

\[
\begin{align*}
K_1 &= h^\alpha f(t, y), \\
K_2 &= h^\alpha f(t + c_2 h, y + a_{21}K_1), \\
K_3 &= h^\alpha f(t + c_3 h, y + a_{31}K_1 + a_{32}K_2), \\
&\vdots \\
K_s &= h^\alpha f(t + c_s h, y + a_{s1}K_1 + a_{s2}K_2 + \cdots + a_{s,s-1}K_{s-1}),
\end{align*}
\]

with

\[ y_{n+1} = y_n + \sum_{i=1}^{s} w_i K_i, \quad (5) \]

where the unknown coefficients \( \{a_{ij}\}_{i=2,j=1}^{s,i-1} \) and the unknown weights \( \{c_i\}_{i=2}^{s} \) have to be determined.

To specify a particular method, one needs to provide \( \{a_{ij}\}_{i=2,j=1}^{s,i-1} \) and \( \{c_i\}_{i=2}^{s} \) accordingly. Following Butcher [23], a method of this type is designated by the following scheme.

We expand \( y_{n+1} \) in (5), in powers of \( h^\alpha \), such that it agrees with the Taylor series expansion of the solution of the FDE (3) in a specified number of terms, (see [35]). To do this, we need some changes on Caputo Taylor series expansion of \( y(t) \).

According to [21], generalized Taylor formula for \( \alpha \in (0, 1] \) with respect to Caputo fractional derivative of function \( y(t) \) is defined as follows

\[
y(t) = y(t_0) + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} \int_{t_0}^{t} \left( \int_{t_0}^{u} \frac{(u-t_0)^{2\alpha}}{\Gamma(2\alpha+1)} \left( \int_{t_0}^{v} \frac{(v-t_0)^{3\alpha}}{\Gamma(3\alpha+1)} y(v) \right) dv \right) du \right) dt_0 + \cdots
\]

\[ (6) \]
where
\[\begin{align*}
\left(\frac{c_0}{t_c}\right)^\alpha y(t) &= f(t, y), \\
\left(\frac{c_0}{t_c}\right)^\alpha y(t)^2 &= \left(\frac{c_0}{t_c}\right)^\alpha f(t, y), \\
\left(\frac{c_0}{t_c}\right)^\alpha y(t)^3 &= \left(\frac{c_0}{t_c}\right)^\alpha f(t, y), \\
\vdots
\end{align*}\]

For obtaining an explicit expression for (7), we propose total differential theorem for Caputo fractional derivative by using (1) and total differential theorem for derivatives of integer order.

Caputo fractional derivatives of composite function \( f(t, y(t)) \) can be computed by fractional Taylor series:
\[ f(t, y(t)) = f(t_0, y(t_0)) + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} f_1^\alpha(t_0, y(t_0)) + \frac{(y-y_0)^2}{2!} f_y(t_0, y(t_0)) + \frac{(t-t_0)^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t}^\alpha(t_0, y(t_0)) \]
\[ + \frac{(t-t_0)^\alpha(y-y_0)}{\Gamma(\alpha+1)} f_{t,y}^\alpha(t_0, y(t_0)) + \frac{(t-t_0)^{3\alpha}}{\Gamma(3\alpha+1)} f_{t,t,t}^\alpha(t_0, y(t_0)) + \cdots \] (8)
where \( f_1^\alpha \) is Caputo fractional derivative of \( f(t, y(t)) \) with respect to \( t \). After inserting \( y(t) - y(t_0) \) from (9) in (8) and by using fractional derivative of (5) for \( \alpha \in (0, 1] \), we have
\[ \frac{c_0}{t_c} D_\alpha^\alpha f(t, y(t)) = f_1^\alpha(t_0, y(t_0)) + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} f_{t,t}^\alpha(t_0, y(t_0)) \]
\[ + \frac{1}{2} \left( \frac{\Gamma(2\alpha+1)}{\Gamma(\alpha+1)^3} (t-t_0)^\alpha f_2(t_0, y(t_0)) + \cdots \right) f_y(t_0, y(t_0)) \]
\[ + \left( \frac{\Gamma(2\alpha+1)}{\Gamma(\alpha+1)^3} (t-t_0)^\alpha f(t_0, y(t_0)) + \cdots \right) f_{t,y}^\alpha(t_0, y(t_0)) \]
\[ + \frac{(t-t_0)^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t,t}^\alpha(t_0, y(t_0)) + \cdots , \]
and so
\[ \frac{c_0}{t_c} D_\alpha^\alpha f(t_0, y(t_0)) = f_1^\alpha(t_0, y(t_0)) + f(t_0, y(t_0)) f_y(t_0, y(t_0)). \] (9)

Also
\[ \frac{c_0}{t_c} D_\alpha^{2\alpha} f(t, y(t)) = \frac{c_0}{t_c} D_\alpha^\alpha f(t_0, y(t_0)) f_y(t_0, y(t_0)) + f_{t,y}^\alpha(t_0, y(t_0)) \]
\[ + \left( \frac{\Gamma(2\alpha+1)}{2\Gamma(\alpha+1)^2} f_2(t_0, y(t_0)) + \cdots \right) f_y(t_0, y(t_0)) \]
\[ + \left( \frac{\Gamma(2\alpha+1)}{\Gamma(\alpha+1)^3} f(t_0, y(t_0)) + \cdots \right) f_{t,y}^\alpha(t_0, y(t_0)) \]
\[ + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} f_{t,t,t}^\alpha(t_0, y(t_0)) + \cdots , \]
where \( f_{i,y}^{a,i} \), \( i = 1, 2, \ldots \) represents the \( i \)th integer derivative of the function \( f_{i,y}^{a} \) with respect to \( y \). As we can see from (9), in Caputo fractional derivatives \( (\int_{t}^{\cdot} D_{t}^{a} y(t))\), \( k = 0, 1, 2, \ldots \), argument \( t_{0} \) in \( y(t_{0}) \) and starting value in \( (\int_{t}^{\cdot} D_{t}^{a} y(t))^{k} \) are the same. To construct an efficient numerical scheme, we should obtain a similar series with the derivatives evaluated in any other point \( (t_{n} > t_{0}) \), such that the expansion can be constructed independently from the starting point \( t_{0} \). In other words, we need

\[
y(t_{n+1}) = y(t_{n}) + \frac{h^{\alpha}}{\Gamma(\alpha + 1)} D_{t}^{\alpha} y(t_{n}) + \frac{h^{2\alpha}}{\Gamma(2\alpha + 1)} (\int_{t}^{\cdot} D_{t}^{\alpha} y(t)) + \frac{h^{3\alpha}}{\Gamma(3\alpha + 1)} (\int_{t}^{\cdot} D_{t}^{\alpha} y(t)) + \cdots, \tag{12}
\]

and so \( (\int_{t}^{\cdot} D_{t}^{\alpha} y(t))^{i}, i = 1, 2, \ldots \). To do so, by using \( \int_{t_{0}}^{\cdot} D_{t}^{\alpha} y(t) \), we obtain \( \int_{t_{0}}^{\cdot} D_{t}^{\alpha} y(t) \) for \( n = 1, 2, \ldots, N_{m} - 1 \), as

\[
\int_{t_{0}}^{t_{n}} D_{t}^{\alpha} y(t) = \int_{t_{0}}^{t_{n}} D_{t}^{\alpha} y(t) - \frac{1}{\Gamma(1 - \alpha)} \int_{t_{0}}^{t_{n}} (t - s)^{-\alpha} D_{s} y(s) ds
\]

\[
= \int_{t_{0}}^{t_{n+1}} D_{t}^{\alpha} y(t) - \frac{1}{\Gamma(1 - \alpha)} \sum_{i=0}^{n-1} \int_{t_{i}}^{t_{i+1}} (t - s)^{-\alpha} D_{s} y(s) ds. \tag{13}
\]

By using Lagrange interpolation formula for \( y(s) \) in support points \( \{t_{i}, t_{i+1}\} \), we have

\[
y(s) \simeq \frac{(s - t_{i})}{(t_{i+1} - t_{i})} y_{i+1} - \frac{(s - t_{i+1})}{(t_{i} - t_{i+1})} y_{i}
\]

\[
= \frac{s - t_{i}}{h} y_{i+1} - \frac{(s - t_{i+1})}{h} y_{i}, \quad s \in [t_{i}, t_{i+1}], \quad i = 0, 1, \ldots, n - 1,
\]
where for sufficiently small \( h \) we have
\[
Dy(s) \simeq \frac{1}{h} (y_{i+1} - y_i), \quad s \in [t_i, t_{i+1}], \quad i = 0, 1, ..., n - 1,
\]
and
\[
\int_{t_i}^{t_{i+1}} (t-s)^{-\alpha} Dy(s) ds \simeq \frac{(y_{i+1} - y_i)}{h(1 - \alpha)} [(t - t_i)^{1-\alpha} - (t - t_{i+1})^{1-\alpha}], \quad i = 0, 1, ..., n - 1.
\]

From (13) and \( c t_0 D^\alpha y(t) = f(t, y) \), we have
\[
ct_n D^\alpha y(t) = f(t, y) - \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^{n-1} \frac{y_{i+1} - y_i}{h(1-\alpha)} [(t - t_i)^{1-\alpha} - (t - t_{i+1})^{1-\alpha}].
\]

So we may write
\[
\xi_n D^\alpha y(t) = F_n(t, y), \quad n = 0, 1, 2, ..., \quad (14)
\]
where \( F_0(t, y) = f(t, y) \) and for \( n = 1, 2, 3, ... \) we have
\[
F_n(t, y) = f(t, y) - \frac{1}{\Gamma(2-\alpha)} \sum_{i=0}^{n-1} \frac{y_{i+1} - y_i}{h} [(t - t_i)^{1-\alpha} - (t - t_{i+1})^{1-\alpha}].
\]

Clearly \( F_n(t, y) \), is continuous and Lipschitz condition with respect to the second variable, due to such properties of \( f(t, y) \). In what follows, for convenience of notation we rename \( F_n(t, y) \) as \( f(t, y) \), i.e., in any initial points \( t_n > t_0 \), \( n = 1, 2, ..., N^m - 1 \), we consider the right terms of (14) as \( f(t_n, y_n) \) instead \( F_n(t_n, y_n) \) in any stages.

Now, for constructing FORK methods, we can use the Taylor formula (6) and (11), where \( \xi_n D^\alpha y(t) \) is defined in (14).

### 3.1. EFORK method of order \( 2\alpha \)

Let us introduce following EFORK method with 2-stage:

\[
\begin{align*}
K_1 &= h^\alpha f(t_n, y_n), \\
K_2 &= h^\alpha f(t_n + c_2 h, y_n + a_{21} K_1), \\
y_{n+1} &= y_n + w_1 K_1 + w_2 K_2
\end{align*}
\]

where coefficients \( c_2, a_{21} \) and weights \( w_1, w_2 \) are chosen to make approximate value \( y_{n+1} \) as possible as closer to exact value \( y(t_{n+1}) \). We expand \( K_1 \) and \( K_2 \) about the point \( (t_n, y_n) \), where we use Caputo Taylor formula (12) about point
and standard integer order Taylor formula about $y_n$ as

$$K_1 = h^\alpha f(t_n, y_n),$$
$$K_2 = h^\alpha f(t_n + c_2 h, y_n + a_{21} K_1)$$

$$= h^\alpha f(t_n, y_n) + \frac{c_3^2 h^\alpha}{\Gamma(\alpha + 1)} f_t^\alpha + a_{21} h^\alpha f_n f_y + \frac{c_3^2 a_{21} h^\alpha}{\Gamma(2\alpha + 1)} f_{t,t}^\alpha + \frac{a_{21}^2 h^{2\alpha}}{2} f_{n,f,y}$$

$$+ \frac{c_3^2 a_{21} h^{2\alpha}}{\Gamma(\alpha + 1)} f_{n,f_{t,y}} + \cdots = h^\alpha f_n + h^{2\alpha} \left( \frac{c_3^2}{\Gamma(\alpha + 1)} f_t^\alpha + a_{21} f_n f_y \right)$$

$$+ h^{3\alpha} \left( \frac{c_3^2}{\Gamma(2\alpha + 1)} f_{t,t}^\alpha + \frac{a_{21}^2}{2} f_{n,f,y} + \frac{c_3^2 a_{21}}{\Gamma(\alpha + 1)} f_{n,f_{t,y}} \right) + \cdots$$

Substituting $K_1$ and $K_2$ in (15), we have

$$y_{n+1} = y_n + (w_1 + w_2) h^\alpha f_n + h^{2\alpha} w_2 \left( \frac{c_3^2}{\Gamma(\alpha + 1)} f_t^\alpha + a_{21} f_n f_y \right)$$

$$+ w_2 h^{3\alpha} \left( \frac{c_3^2}{\Gamma(2\alpha + 1)} f_{t,t}^\alpha + \frac{a_{21}^2}{2} f_{n,f,y} + \frac{c_3^2 a_{21}}{\Gamma(\alpha + 1)} f_{n,f_{t,y}} \right) + \cdots$$

(16)

Comparing (12) with (16) and matching coefficients of powers of $h^\alpha$, we obtain three equations

$$w_1 + w_2 = \frac{1}{\Gamma(\alpha + 1)},$$

$$w_2 \frac{c_3^2}{\Gamma(\alpha + 1)} = \frac{1}{\Gamma(2\alpha + 1)},$$

$$w_2 a_{21} = \frac{1}{\Gamma(2\alpha + 1)}.$$  (17)

From these equations, we see that, if $c_3^2$ is chosen arbitrarily (nonzero), then

$$a_{21} = \frac{c_3^2}{\Gamma(\alpha + 1)}, \quad w_2 = \frac{\Gamma(\alpha + 1)}{c_3^2 \Gamma(2\alpha + 1)}, \quad w_1 = \frac{1}{\Gamma(\alpha + 1)} - \frac{\Gamma(\alpha + 1)}{c_3^2 \Gamma(2\alpha + 1)}. \quad (18)$$

Inserting (17) and (18) in (16) we get

$$y_{n+1} = y_n + \frac{h^\alpha}{\Gamma(\alpha + 1)} f_n + \frac{h^{2\alpha}}{\Gamma(2\alpha + 1)} (f_t^\alpha + f_n f_y)$$

$$+ \frac{c_3^2 h^{3\alpha}}{\Gamma(2\alpha + 1)} \left[ \frac{\Gamma(\alpha + 1)}{\Gamma(2\alpha + 1)} f_{t,t}^\alpha + \frac{1}{2 \Gamma(\alpha + 1)} f_{n,f,y} + \frac{1}{\Gamma(\alpha + 1)} f_{n,f_{t,y}} \right] + \cdots$$

(19)

Subtracting (19) from (12), we obtain the local truncation error $T_n$
\[ T_n = y(t_{n+1}) - y_{n+1} = h^{3\alpha} \left( \frac{1}{\Gamma(3\alpha + 1)} - \frac{c_2 \Gamma(\alpha + 1)}{(2\alpha + 1)^2} \right) f^{\alpha,\alpha}_{t,t} \\
+ h^{3\alpha} \left( \frac{2}{\Gamma(3\alpha + 1)} - \frac{c_2}{2\Gamma(\alpha + 1)} \right) f^{2}_{n,y,y} \\
+ h^{3\alpha} \left( \frac{1}{\Gamma(3\alpha + 1)} - \frac{c_2}{\Gamma(\alpha + 1)} \right) f_{n,f^{\alpha,1}_{t,y}} + \cdots \quad (20) \]

We conclude that no choice of the parameter \( c_2 \) will make the leading term of \( T_n \) vanish for all functions \( f(t, y) \). Sometimes the free parameters are chosen to minimize the sum of the absolute values of the coefficients in \( T_n \). Such a choice is called optimal choice. Obviously the minimum of \( |T_n| \) occurs for

\[ c_2 = \frac{(\Gamma(2\alpha + 1))^2}{\Gamma(3\alpha + 1)\Gamma(\alpha + 1)}, \quad c_2 = \frac{4\Gamma(\alpha + 1)}{\Gamma(3\alpha + 1)} \text{ or } c_2 = \frac{\Gamma(\alpha + 1)}{\Gamma(3\alpha + 1)}. \]

From (20) we have \( \frac{T_n}{h^{3\alpha}} = (h^\alpha)^2 \). So, we deduce that the 2-stage EFORK method (15) is of order \( 2\alpha \).

Now, the 2-stage EFORK method by listing the coefficients is as follows:

| \( c_2 \) | \( a_{21} \) |
|---|---|
| \( w_1 \) | \( w_2 \) |

\[ \begin{array}{c|cc}
\frac{2\Gamma(\alpha + 1)^2}{\Gamma(2\alpha + 1)} & \frac{2\Gamma(\alpha + 1)}{\Gamma(2\alpha + 1)} \\
\frac{1}{2\Gamma(\alpha + 1)} & \frac{1}{2\Gamma(\alpha + 1)} \\
\end{array} \]

Also, the optimal cases of 2-stage EFORK method are:

\[ \begin{array}{c|ccc}
\left( \frac{(\Gamma(2\alpha + 1))^2}{\Gamma(3\alpha + 1)\Gamma(\alpha + 1)} \right)^\alpha & \left( \frac{(\Gamma(2\alpha + 1))^2}{\Gamma(3\alpha + 1)\Gamma(\alpha + 1)} \right)^2 & \left( \frac{(\Gamma(2\alpha + 1))^2}{\Gamma(3\alpha + 1)\Gamma(\alpha + 1)} \right)^3 \\
\frac{1}{\Gamma(\alpha + 1)} & \frac{1}{\Gamma(2\alpha + 1)} & \frac{1}{\Gamma(2\alpha + 1)} \\
\end{array} \]

\[ \begin{array}{c|ccc}
\left( \frac{4\Gamma(\alpha + 1)}{\Gamma(3\alpha + 1)} \right)^\alpha & \frac{4}{\Gamma(3\alpha + 1)} & \frac{4}{\Gamma(3\alpha + 1)} \\
\frac{1}{\Gamma(\alpha + 1)} & \frac{1}{4\Gamma(2\alpha + 1)} & \frac{1}{4\Gamma(2\alpha + 1)} \\
\end{array} \]
3.2. **EFORK method of order** $3\alpha$

Following (4)-(5), we define a 3-stage EFORK method as

$$
K_1 = h^\alpha f(t_n, y_n),
$$
$$
K_2 = h^\alpha f(t + c_2 h, y_n + a_{21} K_1),
$$
$$
K_3 = h^\alpha f(t + c_3 h, y_n + a_{31} K_1 + a_{32} K_2),
$$
$$
y_{n+1} = y_n + w_1 K_1 + w_2 K_2 + w_3 K_3.
$$

(21)

where unknown parameters $\{c_i\}_{i=2,3}$, $\{a_{ij}\}_{i=2,3,j=1}$, and $\{w_i\}_{i=1}^3$ have to be determined accordingly. By using the same procedure as we did for 2-stage EFORK method, expanding $K_1$, $K_2$ and $K_3$, comparing with (12) and matching coefficients of powers of $h^\alpha$, we obtain the following equations:

$$
w_1 + w_2 + w_3 = \frac{1}{\Gamma(\alpha + 1)}, \quad a_{21} = \frac{c_2^\alpha}{\Gamma(\alpha + 1)},
$$
$$
w_2 c_2^\alpha + w_3 c_3^\alpha = \frac{\Gamma(2\alpha + 1)}{\Gamma(3\alpha + 1)}, \quad a_{31} + a_{32} = \frac{c_3^\alpha}{\Gamma(\alpha + 1)},
$$
$$
w_2 c_2^\alpha + w_3 c_3^\alpha = \frac{\Gamma(\alpha + 1)}{\Gamma(2\alpha + 1)}, \quad w_3 a_{32} c_2^\alpha = \frac{\Gamma(\alpha + 1)}{\Gamma(3\alpha + 1)}.
$$

(22)

Now, we have six equations with eight unknown parameters. According to Butcher tableau for 3-stage EFORK method, we have:

$$
c_2 & a_{21} \\
c_3 & a_{31} & a_{32} \\
w_1 & w_2 & w_3
$$

If $c_2$ and $c_3$ are arbitrarily chosen, we calculate weights $\{w_i\}_{i=1}^3$ and coefficients $\{a_{ij}\}_{i=2,3,j=1}$ from (22) as:

$$
\left( \frac{1}{2\alpha!} \right)^\frac{1}{\alpha} - \frac{1}{2(\alpha!)^2} \quad \frac{1}{(\alpha!)^2(2\alpha!)!} + \frac{1}{\alpha!} - \frac{(2\alpha)!}{\Gamma(2\alpha)!(\alpha!)^2} - \frac{(2\alpha)!}{\Gamma(2\alpha)!(\alpha!)^2} - \frac{(2\alpha)!}{\Gamma(2\alpha)!(\alpha!)^2} - \frac{(2\alpha)!}{\Gamma(2\alpha)!(\alpha!)^2}
$$

As a result, we obtain $T_n = (h^\alpha)^3$. In a similar procedure with 2 and 3 stages EFORK methods we can construct s-stages EFORK methods for $s > 3$. 

11
As we can see, to obtain the higher fractional order Runge-Kutta methods, we must consider a method with additional stages. In next section, we express implicit fractional order Runge-Kutta (IFORK) methods with low stages and high orders.

4. IFORK methods

We define a s-stage IFORK method by the following equations:

\[ K_i = \frac{1}{s} h^\alpha \sum_{k=1}^{s} f(t_n + c_i h, y_n + \sum_{j=1}^{s} a_{ij} K_j), \quad i = 1, 2, ..., s, \]  
(23)

and

\[ y_{n+1} = y_n + \sum_{i=1}^{s} w_i K_i, \]  
(24)

where

\[ \frac{c_1^\alpha + c_2^\alpha + ... + c_s^\alpha}{\alpha!} = s(a_{i1} + a_{i2} + ... + a_{is}), \quad i = 1, 2, ..., s \]  
(25)

and the parameters \( \{a_{ij}\}_{i,j=1}^{s,s}, \{w_i\}_{i=1}^{s} \) are arbitrary. We state the IFORK method by listing the coefficients as follows:

\[
\begin{array}{cccc|ccccc}
   c_{11} & c_{12} & \cdots & c_{1s} & a_{11} & a_{12} & \cdots & a_{1s} \\
   c_{21} & c_{22} & \cdots & c_{2s} & a_{21} & a_{22} & \cdots & a_{2s} \\
   \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
   c_{s1} & c_{s2} & \cdots & c_{ss} & a_{s1} & a_{s2} & \cdots & a_{ss} \\
   \hline
   w_1 & w_2 & \cdots & w_s 
\end{array}
\]

Since the functions \( K_i \) are defined by a set of \( s \) implicit equations, the derivation of the implicit methods is complicated. Therefore, without loss of generality, only the case \( s = 2 \) is investigated.

Consider (23)-(25) with \( s = 2 \) as

\[ K_i = \frac{1}{2} h^\alpha [f(t_n + c_{i1} h, y_n + a_{i1} K_1 + a_{i2} K_2) + f(t_n + c_{i2} h, y_n + a_{i1} K_1 + a_{i2} K_2)], \quad i = 1, 2 \]  
(26)

\[ y_{n+1} = y_n + w_1 K_1 + w_2 K_2 \]  
(27)

where

\[ \frac{c_{i1}^\alpha + c_{i2}^\alpha}{\alpha!} = 2(a_{i1} + a_{i2}), \quad i = 1, 2 \]  
(28)

By using the similar procedure as we did for EFORK method, we expand \( K_i \) about the point \( (t_n, y_n) \), where we apply Caputo Taylor formula (12) about \( t_n \).
and standard integer order Taylor formula about $y_n$.

\[
K_i = \frac{1}{2} h^{\alpha}[2f_n + \frac{(c_{11}^{(\alpha)} + c_{12}^{(\alpha)})h^{\alpha}}{\alpha!}f_t^{\alpha} + 2(a_1K_1 + a_2K_2)f_y + \frac{(c_{11}^{(\alpha)} + c_{12}^{(\alpha)})h^{2\alpha}}{(2\alpha)!}f_{t,t}^{\alpha,\alpha}]
\]

\[
+ (a_1K_1 + a_2K_2)^2f_{y,y} + \frac{(c_{11}^{(\alpha)} + c_{12}^{(\alpha)})h^{\alpha}}{\alpha!}(a_1K_1 + a_2K_2)f_{t,y}^{\alpha,1}
\]

\[
+ \frac{(c_{11}^{(3\alpha)} + c_{12}^{(2\alpha)})h^{3\alpha}}{(3\alpha)!}f_{t,t,t}^{\alpha,\alpha,\alpha} + \frac{(c_{11}^{(2\alpha)} + c_{12}^{(2\alpha)})h^{2\alpha}}{(2\alpha)!}(a_1K_1 + a_2K_2)f_{t,y}^{\alpha,\alpha,1}
\]

\[
+ \frac{(c_{11}^{(\alpha)} + c_{12}^{(\alpha)})h^{\alpha}}{\alpha!}(a_1K_1 + a_2K_2)^2f_{t,y,y}^{\alpha,1,1} + \frac{(a_1K_1 + a_2K_2)^3}{3}f_{y,y,y}^{\alpha,1} + \cdots)
\]

(29)

where $i = 1, 2$.

Since equations (29) are implicit, we cannot obtain the explicit forms for $K_1$ and $K_2$. To determine the explicit form $K_i$, we consider

\[
K_i = h^{\alpha}A_i + h^{2\alpha}B_i + h^{3\alpha}C_i + \cdots, \quad i = 1, 2
\]

(30)

where $A_i$, $B_i$ and $C_i$ are unknowns. Substituting (30) into (29) and matching the coefficients of powers of $h^{\alpha}$, we get

\[
A_i = f_n,
\]

\[
B_i = \frac{c_{11}^{(\alpha)} + c_{12}^{(\alpha)}}{2(\alpha!)}f_t^{\alpha} + a_1f_yf_y + a_2f_yf_y = \frac{c_{11}^{(\alpha)} + c_{12}^{(\alpha)}}{2(\alpha!)}D^{\alpha}f,
\]

\[
C_i = \left( a_1 \frac{c_{11}^{(\alpha)} + c_{12}^{(\alpha)}}{2(\alpha!)} + a_2 c_{21}^{(\alpha)} + c_{22}^{(\alpha)} \right) f_yD^{\alpha}f + \frac{c_{11}^{(2\alpha)} + c_{12}^{(2\alpha)}}{2(2\alpha)!}f_{t,y}^{\alpha,\alpha,1}
\]

\[
+ \frac{1}{4} \left( \frac{c_{11}^{(\alpha)} + c_{12}^{(\alpha)}}{\alpha!} \right)^2 \left( \frac{1}{2} f^{2}f_{yy} + ff_{t,y} \right),
\]

(31)

Inserting (30) and (31) into (27), we have

\[
y_{n+1} = y_n + h^{\alpha}[w_1A_1 + w_2A_2] + h^{2\alpha}[w_1B_1 + w_2B_2] + h^{3\alpha}[w_1C_1 + w_2C_2] + \cdots
\]

(32)

Comparing (32) with (12) and equating the coefficient of powers of $h^{\alpha}$, we can get IFORK method of different orders.

4.1. **IFORK method of order $2\alpha$**

To obtain an IFORK method of order $2\alpha$, we equate the coefficients of $h^{\alpha}$ and $h^{2\alpha}$ in (12) and (32) correspondingly, to get

\[
w_1 + w_2 = \frac{1}{\alpha!},
\]

\[
w_1 \frac{c_{11}^{(\alpha)} + c_{12}^{(\alpha)}}{\alpha!} + w_2 \frac{c_{21}^{(\alpha)} + c_{22}^{(\alpha)}}{\alpha!} = \frac{2}{(2\alpha)!},
\]

\[
13
\]
where
\[ 2(a_{11} + a_{12}) = \frac{c_{11}^\alpha + c_{12}^\alpha}{\alpha!}, \quad 2(a_{21} + a_{22}) = \frac{c_{21}^\alpha + c_{22}^\alpha}{\alpha!}. \]

There are now six arbitrary parameters to be prescribed. If we neglect \( K_2 \), i.e., if we choose \( a_{21} = a_{22} = a_{12} = 0 \), \( w_2 = 0 \), from the above equations, we find
\[ w_1 = \frac{1}{\alpha!}, \quad c_{11}^\alpha + c_{12}^\alpha = \frac{(2\alpha)!}{(2\alpha)!}, \quad a_{11} = \frac{\alpha!}{(2\alpha)!}. \]

Therefore, a 1-stage IFORK method of order \( 2\alpha \) is obtained as follows:
\[ K_1 = \frac{1}{2} h^{\alpha} [f(t_n + c_{11}h, y_n + a_{11}K_1) + f(t_n + c_{12}h, y_n + a_{11}K_1)], \]
\[ y_{n+1} = y_n + w_1K_1. \] (33)

4.2. IFORK method of order \( 3\alpha \)

Also, we can get IFORK method of order \( 3\alpha \) with 2-stage \((26)-(28)\), when equating the coefficients of \( h^{\alpha} \), \( h^{2\alpha} \) and \( h^{3\alpha} \) in \((12)\) and \((32)\) accordingly. In such case, we obtain the following system of equations
\[ w_1 + w_2 = \frac{1}{\alpha!}, \quad w_1 \left( \frac{c_{11}^\alpha + c_{12}^\alpha}{\alpha!} + w_2 \frac{c_{21}^\alpha + c_{22}^\alpha}{\alpha!} \right) + w_2 \left( \frac{a_{11} c_{11}^\alpha + a_{12} c_{12}^\alpha}{\alpha!} + \frac{a_{21} c_{21}^\alpha + a_{22} c_{22}^\alpha}{\alpha!} \right) = \frac{2}{(3\alpha)!}, \]
\[ w_1 \left( \frac{c_{11}^\alpha + c_{12}^\alpha}{(2\alpha)!} + w_2 \frac{c_{21}^\alpha + c_{22}^\alpha}{(2\alpha)!} \right) = \frac{2}{(3\alpha)!}, \]
\[ w_1 \left( \frac{c_{11}^\alpha + c_{12}^\alpha}{\alpha!} \right)^2 + w_2 \left( \frac{c_{21}^\alpha + c_{22}^\alpha}{\alpha!} \right)^2 = \frac{8}{(3\alpha)!}, \]

where
\[ 2(a_{11} + a_{12}) = \frac{c_{11}^\alpha + c_{12}^\alpha}{\alpha!}, \quad 2(a_{21} + a_{22}) = \frac{c_{21}^\alpha + c_{22}^\alpha}{\alpha!}. \]

The three free parameters can be chosen in such a way that \( K_1 \) or \( K_2 \) be explicit. If we want \( K_1 \) to be explicit, we choose
\[ c_{11} = a_{11} = a_{12} = 0. \]

Thus, the IFORK method of order \( 3\alpha \) which is explicit in \( K_1 \) is given by
\[ K_1 = \frac{1}{2} h^{\alpha} [f(t_n, y_n) + f(t_n + c_{12}h, y_n)], \]
\[ K_2 = \frac{1}{2} h^{\alpha} [f(t_n + c_{21}h, y_n + a_{21}K_1 + a_{22}K_2) + f(t_n + c_{22}h, y_n + a_{21}K_1 + a_{22}K_2)], \]
\[ y_{n+1} = y_n + w_1K_1 + w_2K_2. \] (34)

Now, we can write the IFORK method of order \( 3\alpha \) as:
5. Theoretical analysis

The obtained difference approximation to the FDEs in FORK methods, does not guarantee that the solution of the difference equation can approximate the exact solution of the FDE correctly. Here the convergence analysis of the FORK methods which arises from some conditions for which the difference solutions converge to the exact solution is investigated.

In this section, firstly we consider a definition of consistency of the discussed methods in section 3 and 4.

5.1. Consistency

The EFORK and IFORK methods considered before belong to the class of methods which are characterized by the use of $y_n$ on the computation of $y_{n+1}$. These family of one-step methods admits the following representation

\[
y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, y_{n+1}, h), \quad n = 0, \ldots, N^m - 1, \quad (35)
y_0 = y(t_0),
\]

where $\Phi : [t_0, T] \times \mathbb{R}^2 \times [0, h_0] \to \mathbb{R}$ and for the particular case of the explicit methods we have the representation

\[
y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, h), \quad n = 0, \ldots, N^m - 1, \quad (36)
y_0 = y(t_0),
\]

with $\Phi : [t_0, T] \times \mathbb{R} \times [0, h_0] \to \mathbb{R}$.

We define the truncation error $\tau_n$ by

\[
\tau_n = \frac{y_{n+1} - y_n}{h^\alpha} - \Phi(t_n, y_n, y_{n+1}, h), \quad (37)
\]

The one-step method (35) and (36) is said consistent with the equation (3), if $\lim_{n \to \infty} \tau_n = 0$, $hN^m = T - t_0$. 

\[
\begin{array}{c|cc|c|cc|c|}
 & \begin{array}{c}
 \frac{(2\alpha)!}{(3\alpha)!} \sqrt{\frac{1}{2(2\alpha)! - 2(\alpha)!^2}} \end{array} & \begin{array}{c}
 \frac{(2\alpha)!}{(3\alpha)!} \sqrt{\frac{1}{2(2\alpha)! - 2(\alpha)!^2}} \end{array} \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\]
Therefore from (37) and (12) we may write
\[
\lim_{h \to 0} \tau_n = \lim_{h \to 0} \frac{y_{n+1} - y_n}{h^\alpha} - \lim_{h \to 0} \Phi(t_n, y_n, y_{n+1}, h),
\]
\[
= \frac{1}{\Gamma(\alpha + 1)} F_n(t_n, y(t_n)) - \lim_{h \to 0} \Phi(t_n, y(t_n), y(t_n+1), h),
\]
\[
= \frac{1}{\Gamma(\alpha + 1)} F_n(t_n, y(t_n)) - \lim_{h \to 0} \Phi(t_n, y(t_n), y(t_n+h), h).
\]

Now we conclude that the proposed one-step EFORK and IFORK methods are consistent if and only if
\[
\Phi(t, y, 0) = \frac{1}{\Gamma(\alpha + 1)} F_n(t, y),
\]
or briefly
\[
\Phi(t, y, 0) = \frac{1}{\Gamma(\alpha + 1)} f(t, y).
\]
Also in a similar manner, for explicit methods we have
\[
\Phi(t, y, 0) = \frac{1}{\Gamma(\alpha + 1)} f(t, y).
\]

As an example, consider the 2-stage EFORK method (15) with (17),
\[
y_{n+1} = y_n + h^\alpha \left[w_1 f(t_n, y_n) + w_2 f(t_n + c_2 h, y_n + a_{21} K_1)\right],
\]
where in comparison to (36), we have
\[
\Phi(t, y, h) = \left[w_1 f(t, y) + w_2 f(t + c_2 h, y + a_{21} K_1)\right],
\]
or when h tends to 0 yields
\[
\Phi(t, y, 0) = (w_1 + w_2) f(t, y),
\]
and by using (17), we may write
\[
\Phi(t, y, 0) = \frac{1}{\Gamma(\alpha + 1)} f(t, y).
\]

Therefore, 2-stage EFORK method (15) is consistent. Also, 3-stage EFORK method (15) is consistent, according to
\[
\Phi(t, y, h) = \left[w_1 f(t, y) + w_2 f(t + c_2 h, y + a_{21} K_1) + w_3 f(t + c_3 h, y + a_{31} K_1 + a_{32} K_2)\right],
\]
\[
\Phi(t, y, 0) = (w_1 + w_2 + w_3) f(t, y),
\]
and so from (22)
\[
\Phi(t, y, 0) = \frac{1}{\Gamma(\alpha + 1)} f(t, y).
\]

Similarly, we can show the consistency of all proposed FORK methods in sections 3 and 4.
5.2. Convergence analysis

In this section we investigate the convergence behavior of the proposed FORK methods (without loss of generality we consider only explicit FORK methods). To do so, we express a definition of regularity from [35].

Definition 5.1. A one-step method of the form (38)

\[ y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, h), \quad n = 0, 1, 2, ..., N^m - 1, \]

satisfies a Lipschitz condition in \( y \) continuously, thus EFORK methods are regular. To establish the convergence behavior, we need following Lemma from [35].

Lemma 5.2. Let \( \omega_0, \omega_1, \omega_2, ... \) be a sequence of real positive numbers which satisfy

\[ \omega_{n+1} \leq (1 + \zeta)\omega_n + \mu, \quad n = 0, 1, 2... \]

where \( \zeta, \mu \) are positive constants. Then

\[ \omega_n \leq e^{n\zeta}\omega_0 + \left( \frac{e^{n\zeta} - 1}{\zeta} \right) \mu, \quad n = 0, 1, 2,... \]

We now discuss on the behavior of the error \( e_n = y(t_n) - y_n \) in EFORK method for the initial-value problem (3).

Theorem 5.3. Consider the initial value problem (3) and let \( f(t, y(t)) \) be continuous and satisfy a Lipschitz condition with Lipschitz constant \( L \) and also \( (t_i^j, D^s_i) f(t) \) is continuous for \( t \in [t_0, T] \). Then the given EFORK method in section 3 is convergent for \( m\alpha \geq 1 \), if and only if it is consistent.
Proof. Let EFORK method be consistent and the method can be written in the form
\[ y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, h). \] (39)

The exact value \( y(t_n) \) will satisfy
\[ y(t_{n+1}) = y(t_n) + h^\alpha \Phi(t_n, y(t_n), h) + T_n, \] (40)

where \( T_n \) is the truncation error. By subtracting (39) from (40), we have
\[ |e_{n+1}| \leq |e_n| + h^\alpha |(\Phi(t_n, y(t_n), h) - \Phi(t_n, y_n, h))| + |T_n|. \]

Now from regularity of the EFORK method it follows that
\[ |e_{n+1}| \leq |e_n| + h^\alpha |y(t_n) - y_n| + |T_n| \leq (1 + h^\alpha L)|e_n| + |T_n|. \]

By using the Lemma 5.2 we have
\[ |e_n| \leq (1 + h^\alpha L)^n |e_0| + \left( \frac{e^{nh^\alpha L} - 1}{h^\alpha L} \right) |T_n|, \]

where we assumed that the local truncation error for sufficiently large \( n \) is constant, i.e. \( T = T_n, n = 0, 1, 2, \ldots \). Also, assume that \( e_0 = 0 \) and \( |T_n| = O(h^{p\alpha}), p \geq 3 \), therefore
\[ |e_n| \leq O(h^{p\alpha}) \left( \frac{e^{nh^\alpha L} - 1}{h^\alpha L} \right). \]

In section 3 we assumed \( N^m = \frac{T-t_0}{h} \), so we have
\[ |e_n| \leq O(h^{(p-1)\alpha}) \left( \frac{e^{(T-t_0)\frac{1}{m}h^\alpha} - 1}{L} \right). \]

Thus, the EFORK methods of subsections 3.1–3.2 are convergent if \( \alpha - \frac{1}{m} \geq 0 \), i.e. \( m\alpha \geq 1 \).

Now conversely, let EFORK method be convergent. It is sufficient we give a limit of (39) as \( h \) tends to 0. Now the proof of theorem is complete. \( \square \)

5.3. Stability analysis

For stability analysis of the proposed method in section 3.4, we consider the FDE
\[
\begin{align*}
\frac{c}{t_0} D^\alpha_t y(t) &= \lambda y(t), & \lambda \in \mathbb{C}, & 0 < \alpha \leq 1, \\
y(t_0) &= y_0.
\end{align*}
\] (41)
According to [1], the exact solution of (41) is
\[ y(t) = E_{\alpha}(\lambda(t - t_0)^{\alpha})y_0. \]
When \( \Re(\lambda) < 0 \), the solution of (41) asymptotically tends to 0 as \( t \to \infty \).

We apply the 2-stage EFORK method (15) to equation (41) and obtain
\[
K_1 = h^\alpha f(t_n, y_n) = \lambda h^\alpha y_n,
\]
\[
K_2 = h^\alpha f(t_n + c_2 h, y_n + a_{21} K_1) = \lambda h^\alpha (y_n + a_{21} \lambda h^\alpha y_n)
= \left[ \lambda h^\alpha + a_{21} (\lambda h^\alpha)^2 \right] y_n,
\]
\[
y_{n+1} = y_n + w_1 K_1 + w_2 K_2 = y_n + \frac{1}{2 \Gamma(\alpha + 1)} \left[ 2 \lambda h^\alpha + a_{21} (\lambda h^\alpha)^2 \right] y_n
= \left[ 1 + \frac{\lambda h^\alpha}{\Gamma(\alpha + 1)} + \frac{a_{21} (\lambda h^\alpha)^2}{2 \Gamma(\alpha + 1)} \right] y_n
= \left[ 1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{c_2 (\lambda h^\alpha)^2}{2 (\alpha!)^2} \right] y_n.
\]

Therefor, the growth factor for 2-stage EFORK method (15) is
\[ E(\lambda h^\alpha) = 1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{c_2 (\lambda h^\alpha)^2}{2 (\alpha!)^2}, \]

So, 2-stage method (15) is absolutely stable if
\[ |1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{c_2 (\lambda h^\alpha)^2}{2 (\alpha!)^2}| \leq 1. \]

If \( \lambda h^\alpha < 0 \), we can find the interval of absolute stability as follows:
\[ \frac{-2 \alpha!}{c_2^2} \leq \lambda h^\alpha < 0. \] (42)

According to (42), the interval of absolute stability for 2-stage EFORK method (15) depends on \( c_2^2 \).

For instance, if \( c_2^2 = \frac{2 (\alpha!)^2}{(2 \alpha)!} \), then and so the interval of absolute stability will be
\[ \frac{-2 \alpha!}{c_2^2} \leq \lambda h^\alpha < 0. \]

Also, for \( c_2^2 = \frac{(\Gamma(2\alpha + 1))^2}{\Gamma(3\alpha + 1)\Gamma(\alpha + 1)} \), we have
\[ \frac{-2 (\alpha!)^2 (3 \alpha)!}{((2 \alpha)!)^2} \leq \lambda h^\alpha < 0. \]

If we choose \( c_2^2 = \frac{4 \Gamma(\alpha + 1)}{\Gamma(3 \alpha + 1)} \), we get
\[ \frac{-3 \alpha!}{2} \leq \lambda h^\alpha < 0. \]
Fig. 1: The graph of $E(\lambda h^\alpha)$ for 2-stage EFORK method (15) with $c_2^\alpha = \frac{(\Gamma(2\alpha+1))^2}{(2\alpha)!}$ (left), and $c_2^\alpha = \frac{\Gamma(2\alpha+1)^2}{\Gamma(3\alpha+1)}$ (right).

Fig. 2: The graph of $E(\lambda h^\alpha)$ for 2-stage EFORK method (15) with $c_2^\alpha = \frac{\Gamma(\alpha+1)^2}{\Gamma(3\alpha+1)}$ (left), and $c_2^\alpha = \frac{\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}$ (right).

or, $c_2^\alpha = \frac{\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}$, we obtain

$$-2(3\alpha)! \leq \lambda h^\alpha < 0.$$

The graph of $E(\lambda h^\alpha)$ for different 2-stage EFORK methods are shown in Figures 1, 2. From these Figures for $(\lambda < 0)$, we can find the interval of absolute stability for various $\alpha$.

Also, we apply the 3-stage EFORK method (21) to equation (11) and get

$$y_{n+1} = \left[1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{(\lambda h^\alpha)^2}{(2\alpha)!} + \frac{(\lambda h^\alpha)^3}{(3\alpha)!}\right] y_n .$$

Thus, the growth factor for 3-stage EFORK method (21) is

$$E(\lambda h^\alpha) = 1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{(\lambda h^\alpha)^2}{(2\alpha)!} + \frac{(\lambda h^\alpha)^3}{(3\alpha)!}.$$
The 3-stage EFORK method is absolutely stable if
\[ |1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{(\lambda h^\alpha)^2}{(2\alpha)!} + \frac{(\lambda h^\alpha)^3}{(3\alpha)!}| \leq 1. \]

The graph of \(E(\lambda h^\alpha)\) for 3-stage EFORK method (21) is shown in Figure 3. In this Figure, we can see the interval of absolute stability for various \(\alpha\).

Next, we apply the IFORK method (33) to equation (41) and get
\[ E(\lambda h^\alpha) = 1 + \frac{1}{\alpha!} \frac{\lambda h^\alpha}{1 - \lambda h^\alpha} \left( \frac{\alpha!}{(2\alpha)!} \right), \]
with the interval of absolute stability \((-\infty, 0), \lambda < 0\). In a similar manner, we can obtain the interval of absolute stability for IFORK method (34). As we can see, in implicit fractional RK methods, interval of absolute stability is very large and they are A stable.

6. Numerical examples

In order to demonstrate the effectiveness and order of accuracy of the proposed methods in sections 3-4, two examples are considered.

**Example 1**: Let us consider fractional differential equation
\[
^{\alpha}D_t^\alpha y(t) = -y(t) + \frac{t^{4-\alpha}}{\Gamma(5-\alpha)}, \quad t \in [0, T],
\]
\[ y(0) = 0, \]
where, the exact solution of the equation is \(y(t) = t^4 E_{\alpha,5}(-t^\alpha)\).

For different values of \(h, \alpha\) and \(T\), the computed solutions are compared with
the exact solution. We have reported the absolute error in time $T$ as:

$$E(h, T) = |y(t_{Nm}) - y_{Nm}|.$$

Also, we calculated the computational orders of the presented method according to the following relation:

$$\log_2 \frac{E(h, T)}{E(h/2, T)}.$$

The computed solutions by 2-stage EFORK method (15), 3-stage EFORK method (21) and IFORK methods (33) and (34) are reported in Tables 1-6. From the Tables 1-6 we can conclude that the computed orders of truncation errors is in a good agreement with the obtained results of sections 3-4.

Table 1: 2-stage method (15) for $T = 1$, $m = 3$ in Example 1.

| $\alpha$ | $h$  | $E(h, T)$       | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ |
|----------|------|----------------|-------------------------------------|
| 1/3      | 1/40 | 1.09627e-2     | 1.1320                              |
| 1/80     | 1.251 | 4.97465e-3     | 1.0013                              |
| 1/160    | 1.252 | 2.48509e-3     | 0.9136                              |
| 1/320    | 1.252 | 1.31920e-3     | 0.8533                              |
| 1/640    | 1.252 | 7.30171e-4     | *                                  |

Table 2: 2-stage method (15) for $T = 1$, $m = 2$ in Example 1.

| $\alpha$ | $h$  | $E(h, T)$       | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ |
|----------|------|----------------|-------------------------------------|
| 1/2      | 1/40 | 2.05503e-3     | 1.2248                              |
| 1/80     | 1.251 | 8.79256e-4     | 1.1621                              |
| 1/160    | 1.252 | 3.92907e-4     | 1.1171                              |
| 1/320    | 1.252 | 1.81137e-4     | 1.0845                              |
| 1/640    | 1.252 | 8.54183e-5     | *                                  |

Table 3: 3-stage method (21) for $T = 1$, $m = 4$ in Example 1.

| $\alpha$ | $h$  | $E(h, T)$       | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ |
|----------|------|----------------|-------------------------------------|
| 1/4      | 1/40 | 9.94252e-4     | 0.8437                              |
| 1/80     | 1.251 | 5.54011e-4     | 0.8214                              |
| 1/160    | 1.252 | 3.13499e-4     | 0.8064                              |
| 1/320    | 1.252 | 1.79258e-4     | 0.7958                              |
| 1/640    | 1.252 | 1.03255e-4     | *                                  |
Table 4: 3-stage method [21] for $T = 1, m = 2$ in Example 1.

| $\alpha$ | $h$   | $E(h, T)$ | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ |
|----------|-------|-----------|-------------------------------------|
| $1/2$    | $1/40$| $7.45694e-5$ | $1.5942$                           |
|          | $1/80$| $2.46986e-5$ | $1.5789$                           |
|          | $1/160$| $8.26771e-6$ | $1.5625$                           |
|          | $1/320$| $2.79911e-6$ | $1.5478$                           |
|          | $1/640$| $9.57367e-7$ | *                                  |

Table 5: IFORK methods for $T = 1, m = 2$ in Example 1.

| $\alpha$ | $h$   | IFORK (33) | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ | IFORK (34) | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ |
|----------|-------|------------|-------------------------------------|------------|-------------------------------------|
| $1/2$    | $1/40$| $1.86448e-4$ | 1.0832                             | $3.56080e-4$| 1.7307                              |
|          | $1/80$| $8.79978e-5$ | 1.0492                             | $1.07291e-4$| 1.6666                              |
|          | $1/160$| $4.25221e-5$ | 1.0291                             | $3.37953e-5$| 1.6191                              |
|          | $1/320$| $2.08361e-5$ | 1.0174                             | $1.10016e-5$| 1.5846                              |
|          | $1/640$| $1.02928e-5$ | *                                  | $3.66806e-6$| *                                  |

Table 6: IFORK methods for $T = 1, m = 3$ in Example 1.

| $\alpha$ | $h$   | IFORK (33) | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ | IFORK (34) | $\log_2 \frac{E(h, T)}{E(h/2, T)}$ |
|----------|-------|------------|-------------------------------------|------------|-------------------------------------|
| $1/3$    | $1/40$| $3.04275e-4$ | 0.9083                             | $5.12337e-3$| 1.4661                             |
|          | $1/80$| $1.62123e-4$ | 0.8746                             | $1.85445e-3$| 1.3856                             |
|          | $1/160$| $8.84207e-5$ | 0.8482                             | $7.09751e-4$| 1.2940                             |
|          | $1/320$| $4.91160e-5$ | 0.8255                             | $2.89458e-4$| 1.2106                             |
|          | $1/640$| $2.77152e-5$ | *                                  | $1.25070e-4$| *                                  |

Fig 4 illustrates the error curves of the 2-stage EFORK method [15] and the 3-stage EFORK method [21] at $T = 1$, with $\alpha = 1/2, m = 2$ and different values of $N$.

Example 2: Let us consider the following fractional differential equation from [6]:

$$\frac{\nu}{\Gamma(3 - \alpha)} t^{2-\alpha} - \frac{1}{\Gamma(2 - \alpha)} t^{1-\alpha} - y(t) + t^2 - t, \quad t \in [0, T],$$

$$y(0) = 0,$$

where the exact solution of the problem is $y(t) = t^2 - t$.

23
Table 7: 2-stage method \((15)\) for \(T = 1, m = 3\) in Example 2.

| \(\alpha\) | \(h\)   | \(E(h, T)\)       | \(\log_2 \frac{E(h, T)}{E(h/2, T)}\) |
|-----------|--------|--------------------|--------------------------------------|
| 1/3       | 1/40   | \(1.00356e-1\)    | 1.1075                               |
|           | 1/80   | \(4.65748e-2\)    | 0.9928                               |
|           | 1/160  | \(2.34046e-2\)    | 0.9107                               |
|           | 1/320  | \(1.24493e-2\)    | 0.8523                               |
|           | 1/640  | \(6.89556e-3\)    | *                                    |

From these Tables, we conclude that the computational order for 2 and 3 stages EFORK methods are \(2\alpha\) and \(3\alpha\), respectively. As expected and seen in Tables and Figs, the 3-stage EFORK method in comparison with the 2-stage EFORK method, provides better results. In Table 13 the numerical results for different values of \(T\) are shown. Also, the numerical results for optimal case \(c_2^*=\frac{(2\alpha+1)^{\alpha}}{\Gamma(3\alpha+1)\Gamma(\alpha+1)}\), in 2-stage EFORK method are shown in Table 14 with \(\alpha = 1/2\) and different values of \(h\).
Fig. 5: The error curves of the 2-stage EFORK method \((15)\) (left), and the 3-stage EFORK method \((21)\) (right) in Example 2.

Fig. 6: The error curves of the IFORK method \((43)\) in Example 1 (left), and Example 2 (right).
Table 8: 2-stage method [15] for \( T = 1, m = 2 \) in Example 2.
\[
\begin{array}{ccc}
\alpha & h & E(h, T) \\
1/2 & 1/40 & 1.77152e-2 \\
 & 1/80 & 7.52581e-3 \\
 & 1/160 & 3.33574e-3 \\
 & 1/320 & 1.52680e-3 \\
 & 1/640 & 7.15859e-4 \\
\end{array}
\]

Table 9: 3-stage method [21] for \( T = 1, m = 4 \) in Example 2.
\[
\begin{array}{ccc}
\alpha & h & E(h, T) \\
1/4 & 1/40 & 9.90939e-3 \\
 & 1/80 & 5.54249e-3 \\
 & 1/160 & 3.14342e-3 \\
 & 1/320 & 1.79955e-3 \\
 & 1/640 & 1.03718e-3 \\
\end{array}
\]

Table 10: 3-stage method [21] for \( T = 1, m = 2 \) in Example 2.
\[
\begin{array}{ccc}
\alpha & h & E(h, T) \\
1/2 & 1/40 & 5.79341e-4 \\
 & 1/80 & 1.96590e-4 \\
 & 1/160 & 6.68302e-5 \\
 & 1/320 & 2.28624e-5 \\
 & 1/640 & 7.87606e-6 \\
\end{array}
\]

Table 11: IFORK methods for \( T = 1, m = 2 \) in Example 2.
\[
\begin{array}{cccc}
\alpha & h & \text{IFORK} [33] & \log_2 \frac{E(h, T)}{E(h/2, T)} \\
1/2 & 1/40 & 1.52888e-3 & 1.0777 \\
 & 1/80 & 7.24362e-4 & 1.0464 \\
 & 1/160 & 3.50728e-4 & 1.0279 \\
 & 1/320 & 1.72002e-4 & 1.0171 \\
 & 1/640 & 8.49858e-5 & * \\
\end{array}
\]

\[
\begin{array}{cccc}
\alpha & h & \text{IFORK} [34] & \log_2 \frac{E(h, T)}{E(h/2, T)} \\
1/2 & 1/40 & 2.99223e-3 & 1.6608 \\
 & 1/80 & 9.46328e-4 & 1.6244 \\
 & 1/160 & 3.06932e-4 & 1.5943 \\
 & 1/320 & 1.01649e-4 & 1.5703 \\
 & 1/640 & 3.42297e-5 & * \\
\end{array}
\]
Table 12: IFORK methods for $T = 1$, $m = 3$ in Example 2.

| $\alpha$ | $h$     | $E(h,T)$ | Log$_2$ $E(h,T)$ | $E(h,T)$ | Log$_2$ $E(h,T)$ |
|----------|---------|----------|------------------|----------|------------------|
| 1/3      | 1/40    | 2.8449e−3| 0.9005           | 4.7762e−2| 1.4860           |
|          | 1/80    | 1.5240e−3| 0.8707           | 1.7050e−2| 1.3835           |
|          | 1/160   | 8.3348e−4| 0.8463           | 6.5352e−3| 1.2859           |
|          | 1/320   | 4.6359e−4| 0.8246           | 2.6802e−3| 1.2023           |
|          | 1/640   | 2.6175e−4| *                | 1.1647e−3| *                |

Table 13: $E(h,T)$ for $\alpha = 1/2$, $m = 2$ and different values of $T$.

| $T$   | 2-stage-Exam.1 | 3-stage-Exam.1 | 2-stage-Exam.2 | 3-stage-Exam.2 |
|-------|----------------|----------------|----------------|----------------|
| 0.5   | 4.81351e−6     | 4.20218e−8     | 9.2906e−5      | 5.57177e−7     |
| 1.0   | 8.54183e−5     | 9.57367e−7     | 3.00335e−3     | 7.87606e−6     |
| 1.5   | 4.51426e−4     | 6.00243e−6     | 2.78127e−3     | 3.63915e−5     |
| 2     | 1.45778e−3     | 2.20455e−5     | 6.26033e−3     | 9.38735e−5     |
| 3     | 7.50603e−3     | 1.37070e−4     | 1.78449e−2     | 3.26003e−4     |

Table 14: Optimal 2-stage method [B3] for $T = 1$, $m = 2$.

| $\alpha$ | $h$     | $E(h,T)$, Example 1 | $E(h,T)$, Example 2 |
|----------|---------|---------------------|---------------------|
| 1/2      | 1/40    | 7.35533e−4          | 6.12299e−3          |
|          | 1/80    | 3.55401e−4          | 2.94885e−3          |
|          | 1/160   | 1.72778e−4          | 1.43060e−3          |
|          | 1/320   | 8.45336e−5          | 6.99024e−4          |
|          | 1/640   | 4.15855e−5          | 3.43589e−4          |

7. Conclusions

This paper introduces new efficient FORK methods for FDEs based on Caputo generalized Taylor formulas. The proposed methods were examined for consistency, convergence, and stability. The interval of absolute stability of FORK methods has been determined, and implicit fractional order RK methods were shown to be A stable. Some examples were provided to demonstrate the effectiveness of these numerical schemes. We can obtain these results for Riemann–Liouville and Gronwald–Letnikov fractional derivatives accordingly. Recently, a new concept of differentiation called fractal and fractional differentiation was suggested and numerically examined by many researchers [36, 37], where the differential operator has two orders: the first is fractional order and the second is the fractal dimension. These differential (integral) operators have not been studied intensively yet. In future work, we will extend the presented method for fractional differential equations with fractal–fractional derivatives.
References

[1] Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.

[2] Hilfer, R. *Applications of Fractional Calculus in Physics*; World Scientific: Singapore, 2000.

[3] Giona, M.; Roman, H.E. Fractional diffusion equation for transport phenomena in random media. *Physica A* 1992, 185, 87–97.

[4] Diethelm, K. *The Analysis of Fractional Differential Equations*; Lecture Notes in Mathematics (LNM, Volume 2004); Springer: Berlin/Heidelberg, Germany, 2004.

[5] Podlubny, I. Geometric and physical interpretation of fractional integration and fractional differentiation. *Fract. Calc. Appl. Anal.* 2002, 5, 367–386.

[6] Odibat, Z.M.; Momani, S. An algorithm for the numerical solution of differential equations of fractional order. *J. Appl. Math. Inform.* 2008, 26, 15–27.

[7] Gao, W.; Veeresha, P.; Prakasha, D.G.; Baskonus, H.M.; Ye, G. New numerical results for the Time-Fractional Phi-Four equation using a novel analytical approach. *Symmetry* 2020, 12, 478.

[8] Postavaru, O.; Toma, A. Numerical solution of two-dimensional fractional-order partial differential equations using hybrid functions. *Part. Diff. Eqs. Appl. Math.* 2021, 4, 100099.

[9] Aslan, E.; Kurkçu, O.K.; Sezer, M. A fast numerical method for fractional partial integro-differential equations with spatial-time delays. *Appl. Numer. Math.* 2021, 161, 525–539.

[10] Garrappa, R. Numerical Solution of Fractional Differential Equations: A Survey and a Software. *Mathematics* 2018, 6, 16.

[11] Sheng, C.; Cao, D.; Shen, J. Efficient spectral methods for PDEs with spectral fractional Laplacian. *J. Sci. Comput.* 2021, 88, 4.

[12] Zhang, T.; Li, Y. Exponential Euler scheme of multi-delay Caputo–Fabrizio fractional-order differential equations. *Appl. Math. Lett.* 2021, 124, 107709.

[13] Zhang, T.; Zhou, J.; Liao, Y. Exponentially stable periodic oscillation and Mittag–Leffler stabilization for fractional-order impulsive control neural networks with piecewise Caputo derivatives. *IEEE Trans. Cyber.* 2022, 52, 9670–9683.
[14] Diethelm, K.; Walz, G. Numerical solution of fractional order differential
equation by extrapolation. *Numer. Algorithms* 1997, *16*, 231–253.

[15] Diethelm, K.; Ford, N.J.; Freed, A.D. A Predictor-corrector approach for
the numerical solution of fractional differential equations. *Nonlinear Dyn.*
2002, *29*, 3–22.

[16] Mokhtary, P.; Ghoreishi F. Convergence analysis of spectral Tau method
for fractional Riccati differential equations. *Bull. Iran. Math. Soc.* 2014, *40*,
1275–1290.

[17] Langlands, T.; Henry, B. The accuracy and stability of an implicit solution
method for the fractional diffusion equation. *J. Comput. Phys.* 2005, *205*,
719–736.

[18] Sun, Z.Z.; Wu, X. A fully discrete difference scheme for a diffusion-wave
system. *Appl. Numer. Math.* 2006, *56*, 193–209.

[19] Gao, G.; Sun, Z.Z.; Zhang, H. A new fractional numerical differentiation
formula to approximate the Caputo fractional derivative and its applica-
tions. *J. Comput. Phys.* 2014, *259*, 33–50.

[20] Cao, J.; Xu, C. A high order scheme for the numerical solution of the
fractional ordinary differential equations. *J. Comput. Phys.* 2013, *238*, 154–
168.

[21] Odibat, Z. Approximations of fractional integrals and Caputo fractional
derivatives. *Appl. Math. Comput.* 2006, *178*, 527–533.

[22] Li, C.; Chen, A.; Ye, J. Numerical approaches to the fractional calculus
and fractional ordinary differential equation. *J. Comput. Phys.* 2011, *230*,
3352–3368.

[23] Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*;
John Wiley & Sons: Chichester, UK, 2008.

[24] Zingg, D.W.; Chisholm, T.T. Runge Kutta methods for linear ordinary
differential equations. *Appl. Numer. Math.* 1999, *31*, 227–238.

[25] Houwen, P. The development of Runge-Kutta methods for partial differential
equations. *Appl. Numer. Math.* 1996, *20*, 261–272.

[26] Butcher, J.C. Trees and numerical methods for ordinary differential equa-
tions. *Numer. Algorithms* 2010, *53*, 153–170.

[27] Butcher, J.C. Practical Runge Kutta methods for scientific computation.
*ANZIAM. J.* 2009, *50*, 333–342.

[28] Lubich, C. Runge-Kutta theory for Volterra and Abel integral equations of
the second kind. *Math. Comput.* 1983, *41*, 87–102.
[29] Brunner, H.; Hairer, E.; Norsett, S.P. Runge-Kutta theory for Volterra
integral equations of the second kind. *Math. Comput.* 1982, 39, 147–163.

[30] Izzo, G.; Russo, E.; Chiapparelli, C. Highly stable Runge–Kutta methods
for Volterra integral equations. *Appl. Numer. Math.* 2012, 62, 1002–1013.

[31] Ostermann, A.; Roche, M. Runge-Kutta Methods for Partial Differential
Equations and Fractional Orders of Convergence. *Math. Comp.* 1992, 59, 403–420.

[32] Ostermann, A.; Roche, M. Rosenbrock methods for partial differential
equations and fractional orders of convergence. *SIAM J. Numer. Anal.* 1993, 30, 1084–1098.

[33] Srivastava, H.M. Operators of Fractional Calculus and Their Applications.
*Mathematics* 2019, 6, 157.

[34] Odibat, Z.; Shawagfeh, N. Generalized Taylor’s formula. *Appl. Math. Com-
put.* 2007, 186, 286–293.

[35] Jain, M.K. *Numerical Solution of Differential Equations*, 2nd ed.; Wiley
Eastern Limited: Bridgewater, NJ, USA, 1991.

[36] Youssri, Y.H.; Atta, A.G. Spectral Collocation Approach via Normalized
Shifted Jacobi Polynomials for the Nonlinear Lane-Emden Equation with
Fractal-Fractional Derivative, *Fractal Fract.* 2023, 7, 133.

[37] Youssri, Y.H. Orthonormal ultraspherical operational matrix algorithm
for fractal–fractional Riccati equation with generalized Caputo derivative.
*Fractal Fract.* 2021, 5, 100.