fractal and fractional

Article

Fractional Order Runge–Kutta Methods

Farideh Ghoreishi [1], Rezvan Ghaffari [1] and Nasser Saad [2,*]

[1] Department of Mathematics, K. N. Toosi University of Technology, Tehran P.O. Box 16765-3381, Iran
[2] School of Mathematical and Computational Sciences, University of Prince Edward Island, Charlottetown, PE C1A 4P3, Canada
* Correspondence: nsaad@upei.ca

Abstract: This paper presents a new class of fractional order Runge–Kutta (FORK) methods for numerically approximating the solution of fractional differential equations (FDEs). We construct explicit and implicit FORK methods for FDEs by using the Caputo generalized Taylor series formula. Due to the dependence of fractional derivatives on a fixed base point, in the proposed method, 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 are presented to clarify the effectiveness and robustness of the method.

Keywords: fractional differential equations; Caputo fractional derivative; convergence analysis; consistency; stability analysis

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 Rieman–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.

Citation: Ghoreishi , F.; Ghaffari, R.; Saad, N. Fractional Order Runge–Kutta Methods. *Fractal Fract.* **2023**, *7*, 245. https://doi.org/10.3390/fractalfract7030245

Academic Editors: Libo Feng, Lin Liu and Yang Liu

Received: 1 February 2023
Revised: 21 February 2023
Accepted: 23 February 2023
Published: 8 March 2023

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 Rochet'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

This section briefly reviews the definitions of the fractional integral and Caputo fractional derivative and explores some of their properties. A more comprehensive introduction to the fractional derivatives can be found in [1,4,33].

Definition 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*

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

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

Definition 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*

$$\begin{aligned} {}_a^c D_x^\alpha f(x) &= J_a^{n-\alpha} D^n f(x) \\ &= \begin{cases} \dfrac{1}{\Gamma(n-\alpha)} \displaystyle\int_a^x (x-t)^{n-\alpha-1} D^n f(t)dt, \; n-1 < \alpha < n, \; n \in \mathbb{N}, \\ \\ D^n f(x), \qquad \alpha = n. \end{cases} \end{aligned} \tag{1}$$

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

$$f(x) = \sum_{i=0}^{n} \frac{(x-a)^{i\alpha}}{\Gamma(1+i\,\alpha)}(({}_{a}^{c}D_{t}^{\alpha})^{i} f)(a) + \frac{(x-a)^{(n+1)\alpha}}{\Gamma(1+(n+1)\,\alpha)}(({}_{a}^{c}D_{t}^{\alpha})^{(n+1)}) f)(\xi), \tag{2}$$

where $({}_{a}^{c}D_{t}^{\alpha})^{n} = \underbrace{{}_{a}^{c}D_{t}^{\alpha}\,{}_{a}^{c}D_{t}^{\alpha} \cdots {}_{a}^{c}D_{t}^{\alpha}}_{n\ times}$.

There are also two important functions in fractional calculus. They are a direct generalization of the exponential series, which play essential roles in solving the FDEs and in stability analysis.

Definition 3. *The Mittag–Leffler function is defined as*

$$E_{\alpha}(x) = \sum_{k=0}^{\infty} \frac{x^{k}}{\Gamma(1+\alpha\, k)}, \quad \Re(\alpha) > 0,\ x \in \mathbb{C}.$$

In addition, the two-parameter Mittag–Leffler function is defined by

$$E_{\alpha,\beta}(x) = \sum_{k=0}^{\infty} \frac{x^{k}}{\Gamma(\beta+\alpha\, k)}, \quad \Re(\alpha) > 0, \beta \in \mathbb{C},\ x \in \mathbb{C}.$$

We note that $E_{\alpha}(x) = E_{\alpha,1}(x)$ and

$$E_{1}(x) = \sum_{k=0}^{\infty} \frac{x^{k}}{\Gamma(1+k)} = \sum_{k=0}^{\infty} \frac{x^{k}}{k!} = \exp(x).$$

3. Fractional Order Runge–Kutta Methods

This section presents a new class of FORK methods for the numerical solutions of FDEs. Consider the following FDE with $0 < \alpha \leq 1$:

$$\begin{cases} {}_{t_0}^{c}D_{t}^{\alpha} y(t) = f(t, y(t)), & t \in [t_0, T], \\ y(t_0) = y_0. \end{cases} \tag{3}$$

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 the base point of fractional derivative.

We set $t_n = t_0 + n\,h$, $n = 0, 1, \cdots, N^m$, where $h = (T - t_0)/N^m$ is the step size, N is a positive integer (in Section 5, we prove that $m \geq 1/\alpha$). For the existence and uniqueness of the solution of the FDE (3), we consider the following theorem [4].

Theorem 2. *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*

$$G = \{(t, y) : t \in [0, T],\ |y - y_0| \leq K\}, \quad M = Sup_{(t,z)\in G}|f(t,z)|$$

and

$$T^{*} = \begin{cases} T, & M = 0, \\ \min\{T, (K\,\Gamma(\alpha+1)/M)^{1/\alpha}\}, & else. \end{cases}$$

Then, there exists a unique function $y \in C[0, T^*]$ solving the initial-value problem (3).

In the sequel, we assume $f(t, y)$ has continuous partial derivatives with respect to t and y to as high an order as we want.

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

Definition 4. *A family of s-stage EFORK methods is defined as*

$$\begin{aligned} K_1 &= h^\alpha f(t, y), \\ K_2 &= h^\alpha f(t + c_2 h, y + a_{21} K_1), \\ K_3 &= h^\alpha f(t + c_3 h, y + a_{31} K_1 + a_{32} K_2), \\ &\ \ \vdots \\ K_s &= h^\alpha f(t + c_s h, y + a_{s1} K_1 + a_{s2} K_2 + \cdots + a_{s,s-1} K_{s-1}), \end{aligned} \tag{4}$$

with

$$y_{n+1} = y_n + \sum_{i=1}^{s} w_i K_i, \tag{5}$$

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

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

$$\begin{array}{c|ccccc} c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & w_1 & w_2 & \cdots & w_{s-1} & w_s \end{array}$$

We expand y_{n+1} in (5), in powers of h^α, such that it agrees with the Taylor series expansion of the solution of the FDE (3) in a specified number of terms (see [35]). According to (2), the generalized Taylor formula for $\alpha \in (0, 1]$ with respect to the Caputo fractional derivative of the function $y(t)$ is defined as follows:

$$\begin{aligned} y(t) = y(t_0) &+ \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} {}_{t_0}^{c}D_t^\alpha y(t_0) + \frac{(t-t_0)^{2\alpha}}{\Gamma(2\alpha+1)} (({}_{t_0}^{c}D_t^\alpha)^2 y)(t_0) \\ &+ \frac{(t-t_0)^{3\alpha}}{\Gamma(3\alpha+1)} (({}_{t_0}^{c}D_t^\alpha)^3 y)(t_0) + \cdots, \end{aligned} \tag{6}$$

where using (3),

$${}_{t_0}^{c}D_t^\alpha\, y(t) = f(t, y),\ ({}_{t_0}^{c}D_t^\alpha)^2\, y(t) = {}_{t_0}^{c}D_t^\alpha f(t, y),\ ({}_{t_0}^{c}D_t^\alpha)^3\, y(t) = ({}_{t_0}^{c}D_t^\alpha)^2 f(t, y), \cdots. \tag{7}$$

Now, we obtain explicit expressions for (7).

Caputo fractional derivatives of composite function $f(t, y(t))$ can be computed by fractional Taylor series:

$$
\begin{aligned}
f(t,y(t)) = {} & f(t_0,y(t_0)) + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} f_t^\alpha(t_0,y(t_0)) + \frac{(y-y_0)}{1!} f_y(t_0,y(t_0)) \\
& + \frac{(t-t_0)^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t}^{\alpha,\alpha}(t_0,y(t_0)) + \frac{(y-y_0)^2}{2!} f_{y,y}(t_0,y(t_0)) \\
& + \frac{(t-t_0)^\alpha (y-y_0)}{\Gamma(\alpha+1)} f_{t,y}^{\alpha,1}(t_0,y(t_0)) + \frac{(t-t_0)^{3\alpha}}{\Gamma(3\alpha+1)} f_{t,t,t}^{\alpha,\alpha,\alpha}(t_0,y(t_0)) + \cdots, \qquad (8)
\end{aligned}
$$

where f_t^α is the Caputo fractional derivative of $f(t,y(t))$ with respect to t. After inserting $y(t)-y(t_0)$ from (6) in (8) and by using the fractional derivative of (8) for $\alpha \in (0,1]$, we have

$$
\begin{aligned}
{}_{t_0}^{c}D_t^\alpha f(t,y(t)) = {} & f_t^\alpha(t_0,y(t_0)) + \left(f(t_0,y(t_0)) + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} {}_{t_0}^{c}D_t^\alpha f(t_0,y(t_0)) + \ldots \right) f_y(t_0,y(t_0)) \\
& + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} f_{t,t}^{\alpha,\alpha}(t_0,y(t_0)) + \frac{1}{2}\left(\frac{\Gamma(2\alpha+1)}{\Gamma(\alpha+1)^3}(t-t_0)^\alpha f^2(t_0,y(t_0)) + \ldots \right) f_{y,y}(t_0,y(t_0)) \\
& + \left(\frac{\Gamma(2\alpha+1)}{\Gamma(\alpha+1)^3}(t-t_0)^\alpha f(t_0,y(t_0)) + \ldots \right) f_{t,y}^{\alpha,1}(t_0,y(t_0)) \\
& + \frac{(t-t_0)^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t,t}^{\alpha,\alpha,\alpha}(t_0,y(t_0)) + \cdots,
\end{aligned}
$$

and so

$$
{}_{t_0}^{c}D_t^\alpha f(t_0,y(t_0)) = f_t^\alpha(t_0,y(t_0)) + f(t_0,y(t_0)) f_y(t_0,y(t_0)). \qquad (9)
$$

In addition,

$$
\begin{aligned}
{}_{t_0}^{c}D_t^{2\alpha} f(t,y(t)) = {} & {}_{t_0}^{c}D_t^\alpha f(t_0,y(t_0)) f_y(t_0,y(t_0)) + f_{t,t}^{\alpha,\alpha}(t_0,y(t_0)) \\
& + \left(\frac{\Gamma(2\alpha+1)}{2\Gamma(\alpha+1)^2} f^2(t_0,y(t_0)) + \ldots \right) f_{y,y}(t_0,y(t_0)) \\
& + \left(\frac{\Gamma(2\alpha+1)}{\Gamma(\alpha+1)^2} f(t_0,y(t_0)) + \ldots \right) f_{t,y}^{\alpha,1}(t_0,y(t_0)) \\
& + \frac{(t-t_0)^\alpha}{\Gamma(\alpha+1)} f_{t,t,t}^{\alpha,\alpha,\alpha}(t_0,y(t_0)) + \cdots,
\end{aligned}
$$

which yields

$$
\begin{aligned}
{}_{t_0}^{c}D_t^{2\alpha} f(t_0,y(t_0)) = {} & f_t^\alpha(t_0,y(t_0)) f_y(t_0,y(t_0)) + f(t_0,y(t_0)) f_y^2(t_0,y(t_0)) + f_{t,t}^{\alpha,\alpha}(t_0,y(t_0)) \\
& + \frac{1}{2} f^2(t_0,y(t_0)) f_{y,y}(t_0,y(t_0)) + f(t_0,y(t_0)) f_{t,y}^{\alpha,1}(t_0,y(t_0)). \qquad (10)
\end{aligned}
$$

In a similar manner with (9) and (10), we can obtain the higher fractional derivatives of $f(t,y(t))$.

Now, by using (9) and (10), we have

$$
\begin{aligned}
{}_{t_0}^{c}D_t^\alpha y(t_0) &= f(t_0,y_0), \\
({}_{t_0}^{c}D_t^\alpha)^2 y(t_0) &= f_t^\alpha(t_0,y_0) + f(t_0,y_0) f_y(t_0,y_0), \\
({}_{t_0}^{c}D_t^\alpha)^3 y(t_0) &= f_t^\alpha(t_0,y(t_0)) f_y(t_0,y(t_0)) + f(t_0,y(t_0)) f_y^2(t_0,y(t_0)) + f_{t,t}^{\alpha,\alpha}(t_0,y(t_0) \\
& \quad + \frac{1}{2} f^2(t_0,y(t_0)) f_{y,y}(t_0,y(t_0)) + f(t_0,y(t_0)) f_{t,y}^{\alpha,1}(t_0,y(t_0)), \qquad (11)
\end{aligned}
$$

where $f_{t,y}^{\alpha,i}$, $i = 1,2,\cdots$, represents the ith integer derivative of the function f_t^α with respect to y. As we can see from (6), in Caputo fractional derivatives $(({}_{t_0}^{c}D_t^\alpha)^k y)(t_0)$, $k = 0,1,2,\cdots$, the argument t_0 in $y(t_0)$ and starting value in $({}_{t_0}^{c}D_t^\alpha)^k$ 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

$$y(t_{n+1}) = y(t_n) + \frac{h^\alpha}{\Gamma(\alpha+1)} {}_{t_n}^{c}D_t^\alpha y(t_n) + \frac{h^{2\alpha}}{\Gamma(2\alpha+1)}(({}_{t_n}^{c}D_t^\alpha)^2 y)(t_n) + \frac{h^{3\alpha}}{\Gamma(3\alpha+1)}(({}_{t_n}^{c}D_t^\alpha)^3 y)(t_n) + \cdots, \tag{12}$$

and $(({}_{t_n}^{c}D_t^\alpha)^i y)(t_n), i = 1, 2, \cdots$. To do so, by using ${}_{t_0}^{c}D_t^\alpha y(t)$, we obtain ${}_{t_n}^{c}D_t^\alpha y(t)$ for $n = 1, 2, \cdots, N^m - 1$, as

$$\begin{aligned} {}_{t_n}^{c}D_t^\alpha y(t) &= {}_{t_0}^{c}D_t^\alpha y(t) - \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t_n} (t-s)^{-\alpha} Dy(s) ds \\ &= {}_{t_0}^{c}D_t^\alpha y(t) - \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} (t-s)^{-\alpha} Dy(s) ds. \end{aligned} \tag{13}$$

By using the linear Lagrange interpolation formula for $y(s)$ in support abscissas $\{t_i, t_{i+1}\}$, we have

$$\begin{aligned} y(s) &\simeq \frac{(s-t_i)}{(t_{i+1}-t_i)} y_{i+1} - \frac{(s-t_{i+1})}{(t_{i+1}-t_i)} y_i \\ &= \frac{(s-t_i)}{h} y_{i+1} - \frac{(s-t_{i+1})}{h} y_i, \quad s \in [t_i, t_{i+1}], \quad i = 0, 1, \ldots, n-1, \end{aligned}$$

where, for a sufficiently small h, we have

$$Dy(s) \simeq \frac{1}{h}(y_{i+1} - y_i), \quad s \in [t_i, t_{i+1}], \quad i = 0, 1, \ldots, n-1,$$

and

$$\int_{t_i}^{t_{i+1}} (t-s)^{-\alpha} Dy(s) ds \simeq \frac{(y_{i+1}-y_i)}{h(1-\alpha)} \Big[(t-t_i)^{1-\alpha} - (t-t_{i+1})^{1-\alpha} \Big], \; i = 0, 1, \cdots, n-1.$$

From (13) and ${}_{t_0}^{c}D_t^\alpha y(t) = f(t, y)$, we have

$${}_{t_n}^{c}D_t^\alpha y(t) = f(t, y) - \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^{n-1} \frac{(y_{i+1}-y_i)}{h(1-\alpha)} \Big[(t-t_i)^{1-\alpha} - (t-t_{i+1})^{1-\alpha} \Big].$$

So we may write

$${}_{t_n}^{c}D_t^\alpha y(t) = F_n(t, y), \qquad n = 0, 1, 2, \ldots. \tag{14}$$

where $F_0(t, y) = f(t, y)$ and for $n = 1, 2, 3, \cdots$, we have

$$F_n(t, y) = f(t, y) - \frac{1}{\Gamma(2-\alpha)} \sum_{i=0}^{n-1} \frac{y_{i+1}-y_i}{h} \Big[(t-t_i)^{1-\alpha} - (t-t_{i+1})^{1-\alpha} \Big].$$

Clearly, $F_n(t, y)$ is continuous and satisfies the Lipschitz condition with respect to the second variable, due to the 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 $t_n > t_0$, $n = 1, 2, \ldots, N^m - 1$, we consider the right terms of (14) as $f(t_n, y_n)$ instead of $F_n(t_n, y_n)$ in any stages.

Now, to construct FORK methods, we can use the Taylor formula (6) and (11), where ${}_{t_n}^{c}D_t^\alpha y(t)$ is defined in (14).

3.1. EFORK Method of Order 2α

Let us introduce the following EFORK method with two stages:

$$\begin{aligned} K_1 &= h^\alpha f(t_n, y_n), \\ K_2 &= h^\alpha f(t_n + c_2 h, y_n + a_{21} K_1), \\ y_{n+1} &= y_n + w_1 K_1 + w_2 K_2, \end{aligned} \tag{15}$$

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

$$\begin{aligned} K_1 &= h^\alpha f(t_n, y_n), \\ K_2 &= h^\alpha f(t_n + c_2 h, y_n + a_{21} K_1) \\ &= h^\alpha \Bigg[f(t_n, y_n) + \frac{c_2^\alpha h^\alpha}{\Gamma(\alpha+1)} f_t^\alpha + a_{21} h^\alpha f_n f_y + \frac{c_2^{2\alpha} h^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t}^{\alpha,\alpha} + \frac{a_{21}^2 h^{2\alpha}}{2} f_n^2 f_{y,y} \\ &\quad + \frac{c_2^\alpha a_{21} h^{2\alpha}}{\Gamma(\alpha+1)} f_n f_{t,y}^{\alpha,1} + \cdots \Bigg] \\ &= h^\alpha f_n + h^{2\alpha} \left(\frac{c_2^\alpha}{\Gamma(\alpha+1)} f_t^\alpha + a_{21} f_n f_y \right) \\ &\quad + h^{3\alpha} \left(\frac{c_2^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t}^{\alpha,\alpha} + \frac{a_{21}^2}{2} f_n^2 f_{y,y} + \frac{c_2^\alpha a_{21}}{\Gamma(\alpha+1)} f_n f_{t,y}^{\alpha,1} \right) + \cdots . \end{aligned}$$

Substituting K_1 and K_2 in (15), we have

$$\begin{aligned} y_{n+1} &= y_n + (w_1 + w_2) h^\alpha f_n + h^{2\alpha} w_2 \left(\frac{c_2^\alpha}{\Gamma(\alpha+1)} f_t^\alpha + a_{21} f_n f_y \right) \\ &\quad + w_2 h^{3\alpha} \left(\frac{c_2^{2\alpha}}{\Gamma(2\alpha+1)} f_{t,t}^{\alpha,\alpha} + \frac{a_{21}^2}{2} f_n^2 f_{y,y} + \frac{c_2^\alpha a_{21}}{\Gamma(\alpha+1)} f_n f_{t,y}^{\alpha,1} \right) + \cdots \end{aligned} \tag{16}$$

Comparing (12) with (16) and matching coefficients of powers of h^α, we obtain three equations:

$$\begin{aligned} w_1 + w_2 &= \frac{1}{\Gamma(\alpha+1)}, \\ w_2 \frac{c_2^\alpha}{\Gamma(\alpha+1)} &= \frac{1}{\Gamma(2\alpha+1)}, \\ w_2 a_{21} &= \frac{1}{\Gamma(2\alpha+1)}. \end{aligned} \tag{17}$$

From these equations, we see that, if c_2^α is chosen arbitrarily (nonzero), then

$$a_{21} = \frac{c_2^\alpha}{\Gamma(\alpha+1)}, \quad w_2 = \frac{\Gamma(\alpha+1)}{c_2^\alpha \Gamma(2\alpha+1)}, \quad w_1 = \frac{1}{\Gamma(\alpha+1)} - \frac{\Gamma(\alpha+1)}{c_2^\alpha \Gamma(2\alpha+1)}. \tag{18}$$

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

$$\begin{aligned} y_{n+1} &= y_n + \frac{h^\alpha}{\Gamma(\alpha+1)} f_n + \frac{h^{2\alpha}}{\Gamma(2\alpha+1)} \left(f_t^\alpha + f_n f_y \right) \\ &\quad + \frac{c_2^\alpha h^{3\alpha}}{\Gamma(2\alpha+1)} \left[\frac{\Gamma(\alpha+1)}{\Gamma(2\alpha+1)} f_{t,t}^{\alpha,\alpha} + \frac{1}{2\Gamma(\alpha+1)} f_n^2 f_{y,y} + \frac{1}{\Gamma(\alpha+1)} f_n f_{t,y}^{\alpha,1} \right] + \cdots . \end{aligned} \tag{19}$$

Subtracting (19) from (12), we obtain the local truncation error T_n

$$\begin{aligned}T_n = y(t_{n+1}) - y_{n+1} &= h^{3\alpha}\left(\frac{1}{\Gamma(3\alpha+1)} - \frac{c_2^\alpha \Gamma(\alpha+1)}{(\Gamma(2\alpha+1))^2}\right) f_{t,t}^{\alpha,\alpha} \\ &+ h^{3\alpha}\left(\frac{2}{\Gamma(3\alpha+1)} - \frac{c_2^\alpha}{2\Gamma(\alpha+1)}\right) f_n^2 f_{y,y} \\ &+ h^{3\alpha}\left(\frac{1}{\Gamma(3\alpha+1)} - \frac{c_2^\alpha}{\Gamma(\alpha+1)}\right) f_n f_{t,y}^{\alpha,1} + \cdots . \end{aligned} \tag{20}$$

We conclude that no choice of the parameter c_2^α 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 the optimal choice.

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

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

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

$$\begin{array}{c|cc} c_2 & a_{21} & \\ \hline & w_1 & w_2 \end{array},$$

Choosing $w_1 = w_2$ yields

$$\begin{array}{c|cc} \left(\frac{2\Gamma(\alpha+1)^2}{\Gamma(2\alpha+1)}\right)^{\frac{1}{\alpha}} & \frac{2\Gamma(\alpha+1)}{\Gamma(2\alpha+1)} & \\ \hline & \frac{1}{2\Gamma(\alpha+1)} & \frac{1}{2\Gamma(\alpha+1)} \end{array}.$$

In addition, the optimal cases of the two-stage EFORK method are

$$\begin{array}{c|cc} \left(\frac{(\Gamma(2\alpha+1))^2}{\Gamma(3\alpha+1)\Gamma(\alpha+1)}\right)^{\frac{1}{\alpha}} & \frac{(\Gamma(2\alpha+1))^2}{\Gamma(3\alpha+1)\Gamma(\alpha+1)^2} & \\ \hline & \frac{1}{\Gamma(\alpha+1)} - \frac{\Gamma(3\alpha+1)\Gamma(\alpha+1)^2}{\Gamma(2\alpha+1)^3} & \frac{\Gamma(3\alpha+1)\Gamma(\alpha+1)^2}{\Gamma(2\alpha+1)^3} \end{array},$$

$$\begin{array}{c|cc} \left(\frac{4\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}\right)^{\frac{1}{\alpha}} & \frac{4}{\Gamma(3\alpha+1)} & \\ \hline & \frac{1}{\Gamma(\alpha+1)} - \frac{\Gamma(3\alpha+1)}{4\Gamma(2\alpha+1)} & \frac{\Gamma(3\alpha+1)}{4\Gamma(2\alpha+1)} \end{array},$$

$$\begin{array}{c|cc} \left(\frac{\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}\right)^{\frac{1}{\alpha}} & \frac{1}{\Gamma(3\alpha+1)} & \\ \hline & \frac{1}{\Gamma(\alpha+1)} - \frac{\Gamma(3\alpha+1)}{\Gamma(2\alpha+1)} & \frac{\Gamma(3\alpha+1)}{\Gamma(2\alpha+1)} \end{array}.$$

3.2. EFORK Method of Order 3α

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

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

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

$$\begin{aligned} &w_1 + w_2 + w_3 = \frac{1}{\Gamma(\alpha+1)}, \quad a_{21} = \frac{c_2^\alpha}{\Gamma(\alpha+1)}, \quad w_2\, c_2^{2\alpha} + w_3\, c_3^{2\alpha} = \frac{\Gamma(2\alpha+1)}{\Gamma(3\alpha+1)}, \\ &a_{31} + a_{32} = \frac{c_3^\alpha}{\Gamma(\alpha+1)}, \quad w_2\, c_2^\alpha + w_3\, c_3^\alpha = \frac{\Gamma(\alpha+1)}{\Gamma(2\alpha+1)}, \quad w_3\, a_{32}\, c_2^\alpha = \frac{\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}. \end{aligned} \tag{22}$$

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

$$\begin{array}{c|ccc} c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & w_1 & w_2 & w_3 \end{array}$$

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

$$\begin{array}{c|ccc} \left(\frac{1}{2\alpha!}\right)^{\frac{1}{\alpha}} & \frac{1}{2(\alpha!)^2} & & \\ \left(\frac{1}{4\alpha!}\right)^{\frac{1}{\alpha}} & \frac{(\alpha!)^2(2\alpha)!+2(2\alpha)!^2-(3\alpha)!}{4(\alpha!)^2(2(2\alpha)!^2-(3\alpha)!)} & -\frac{(2\alpha)!}{4(2(2\alpha)!)-(3\alpha)!} & \\ \hline & \frac{8(\alpha!)^2(2\alpha)!}{(3\alpha)!} - \frac{6(\alpha!)^2}{(2\alpha)!} + \frac{1}{\alpha!} & \frac{2(\alpha!)^2(4(2\alpha)!^2-(3\alpha)!)}{(2\alpha)!(3\alpha)!} & -\frac{8(\alpha!)^2(2(2\alpha)!^2-(3\alpha)!)}{(2\alpha)!(3\alpha)!} \end{array}$$

As a result, we obtain $\frac{T_n}{h^\alpha} = (h^\alpha)^3$. In a similar procedure to the two and three-stage EFORK methods, we can construct s-stage EFORK methods for $s > 3$.

As we can see, to obtain the higher fractional order Runge–Kutta methods, we must consider a method with additional stages. In the 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:

$$K_i = \frac{1}{s} h^\alpha \sum_{k=1}^{s} f(t_n + c_{ik} h, y_n + \sum_{j=1}^{s} a_{ij} K_j), \quad i = 1, 2, \ldots, s \tag{23}$$

and

$$y_{n+1} = y_n + \sum_{i=1}^{s} w_i K_i, \tag{24}$$

where

$$\frac{c_{i1}^\alpha + c_{i2}^\alpha + \ldots + c_{is}^\alpha}{\alpha!} = s(a_{i1} + a_{i2} + \cdots + a_{is}), \quad i = 1, 2, \cdots, s \tag{25}$$

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:

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

Since the functions K_i are defined by a set of s implicit equations, the derivation of the implicit methods is complicated. Therefore, only the case $s = 2$ is investigated.

Consider (23)–(25) with $s = 2$ as

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

$$y_{n+1} = y_n + w_1K_1 + w_2K_2 \tag{27}$$

where

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

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

$$\begin{aligned} K_i = &\frac{1}{2}h^\alpha[2f_n + \frac{(c_{i1}^\alpha + c_{i2}^\alpha)h^\alpha}{\alpha!}f_t^\alpha + 2(a_{i1}K_1 + a_{i2}K_2)f_y + \frac{(c_{i1}^\alpha + c_{i2}^\alpha)h^{2\alpha}}{(2\alpha)!}f_{t,t}^{\alpha,\alpha} \\ &+ (a_{i1}K_1 + a_{i2}K_2)^2 f_{y,y} + \frac{(c_{i1}^\alpha + c_{i2}^\alpha)h^\alpha}{\alpha!}(a_{i1}K_1 + a_{i2}K_2)f_{t,y}^{\alpha,1} \\ &+ \frac{(c_{i1}^{3\alpha} + c_{i2}^{3\alpha})h^{3\alpha}}{(3\alpha)!}f_{t,t,t}^{\alpha,\alpha,\alpha} + \frac{(c_{i1}^{2\alpha} + c_{i2}^{2\alpha})h^{2\alpha}}{(2\alpha)!}(a_{i1}K_1 + a_{i2}K_2)f_{t,t,y}^{\alpha,\alpha,1} \\ &+ \frac{(c_{i1}^\alpha + c_{i2}^\alpha)h^\alpha}{\alpha!}\frac{(a_{i1}K_1 + a_{i2}K_2)^2}{2}f_{t,y,y}^{\alpha,1,1} + \frac{(a_{i1}K_1 + a_{i2}K_2)^3}{3}f_{y,y,y} + \cdots], \end{aligned} \tag{29}$$

where $i = 1,2$.

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

$$K_i = h^\alpha A_i + h^{2\alpha}B_i + h^{3\alpha}C_i + \cdots, \quad i = 1,2 \tag{30}$$

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

$$\begin{aligned} A_i &= f_n, \\ B_i &= \frac{c_{i1}^\alpha + c_{i2}^\alpha}{2(\alpha!)}[f_t^\alpha + ff_y] = \frac{c_{i1}^\alpha + c_{i2}^\alpha}{2(\alpha!)}D^\alpha f, \\ C_i &= \left(a_{i1}\frac{c_{11}^\alpha + c_{12}^\alpha}{2(\alpha!)} + a_{i2}\frac{c_{21}^\alpha + c_{22}^\alpha}{2(\alpha!)}\right)f_y D^\alpha f + \frac{c_{i1}^{2\alpha} + c_{i2}^{2\alpha}}{2(2\alpha)!}f_{t,t}^{\alpha,\alpha} \\ &\quad + \frac{1}{4}\left(\frac{c_{i1}^\alpha + c_{i2}^\alpha}{\alpha!}\right)^2\left(\frac{1}{2}f^2 f_{yy} + ff_{t,y}^{\alpha,1}\right), \\ &\vdots \end{aligned} \tag{31}$$

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

$$y_{n+1} = y_n + h^\alpha[w_1A_1 + w_2A_2] + h^{2\alpha}[w_1B_1 + w_2B_2] + h^{3\alpha}[w_1C_1 + w_2C_2] + \cdots. \tag{32}$$

Comparing (32) with (12) and equating the coefficient of powers of h^α, we can get an IFORK method of different orders.

4.1. IFORK Method of Order 2α

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

$$
\begin{aligned}
&w_1 + w_2 = \frac{1}{\alpha!},\\
&w_1 \frac{c_{11}^{\alpha} + c_{12}^{\alpha}}{\alpha!} + w_2 \frac{c_{21}^{\alpha} + c_{22}^{\alpha}}{\alpha!} = \frac{2}{(2\alpha)!},
\end{aligned}
$$

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

$$
w_1 = \frac{1}{\alpha!}, \quad c_{11}^{\alpha} + c_{12}^{\alpha} = \frac{2(\alpha!)^2}{(2\alpha)!}, \quad a_{11} = \frac{\alpha!}{(2\alpha)!}.
$$

Therefore, a one-stage IFORK method of order 2α is obtained as follows:

$$
\begin{cases}
K_1 = \frac{1}{2} h^{\alpha} [f(t_n + c_{11}h, y_n + a_{11}K_1) + f(t_n + c_{12}h, y_n + a_{11}K_1)],\\
y_{n+1} = y_n + w_1 K_1.
\end{cases}
\tag{33}
$$

4.2. IFORK Method of Order 3α

In addition, we can get an IFORK method of order 3α with two stages (26)–(28) when equating the coefficients of h^{α}, $h^{2\alpha}$ and $h^{3\alpha}$ in (12) and (32) accordingly. In such case, we obtain the following system of equations:

$$
\begin{aligned}
&w_1 + w_2 = \frac{1}{\alpha!},\\
&w_1 \frac{c_{11}^{\alpha} + c_{12}^{\alpha}}{\alpha!} + w_2 \frac{c_{21}^{\alpha} + c_{22}^{\alpha}}{\alpha!} = \frac{2}{(2\alpha)!},\\
&w_1 \left(a_{11} \frac{c_{11}^{\alpha} + c_{12}^{\alpha}}{\alpha!} + a_{12} \frac{c_{21}^{\alpha} + c_{22}^{\alpha}}{\alpha!} \right) + w_2 \left(a_{21} \frac{c_{11}^{\alpha} + c_{12}^{\alpha}}{\alpha!} + a_{22} \frac{c_{21}^{\alpha} + c_{22}^{\alpha}}{\alpha!} \right) = \frac{2}{(3\alpha)!},\\
&w_1 \frac{c_{11}^{2\alpha} + c_{12}^{2\alpha}}{(2\alpha)!} + w_2 \frac{c_{21}^{2\alpha} + c_{22}^{2\alpha}}{(2\alpha)!} = \frac{2}{(3\alpha)!},\\
&w_1 \left(\frac{c_{11}^{\alpha} + c_{12}^{\alpha}}{\alpha!} \right)^2 + w_2 \left(\frac{c_{21}^{\alpha} + c_{22}^{\alpha}}{\alpha!} \right)^2 = \frac{8}{(3\alpha)!},
\end{aligned}
$$

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 so that K_1 or K_2 are explicit. If we want K_1 to be explicit, we choose

$$
c_{11} = a_{11} = a_{12} = 0.
$$

Thus, the IFORK method of order 3α, which is explicit in K_1 is given by

$$
\begin{cases}
K_1 = \frac{1}{2} h^{\alpha} [f(t_n, y_n) + f(t_n + c_{12}h, y_n)],\\
K_2 = \frac{1}{2} h^{\alpha} [f(t_n + c_{21}h, y_n + a_{21}K_1 + a_{22}K_2) + f(t_n + c_{22}h, y_n + a_{21}K_1 + a_{22}K_2)],\\
y_{n+1} = y_n + w_1 K_1 + w_2 K_2\,.
\end{cases}
\tag{34}
$$

where the coefficients are given by

$$
\begin{array}{cc|cc}
c_{11} & c_{12} & a_{11} & a_{12}\\
c_{21} & c_{22} & a_{21} & a_{22}\\
\hline
 & & w_1 & w_2
\end{array},
$$

$$
\begin{array}{cc|cc}
0 & 0 & 0 & 0 \\
\left(\frac{(2\alpha)!(2(\alpha!)-\sqrt{2}\sqrt{(2\alpha)!-2(\alpha!)^2})}{(3\alpha)!}\right)^{\frac{1}{\alpha}} & \left(\frac{(2\alpha)!(2(\alpha!)+\sqrt{2}\sqrt{(2\alpha)!-2(\alpha!)^2})}{(3\alpha)!}\right)^{\frac{1}{\alpha}} & \frac{(2\alpha)!}{(3\alpha)!} & \frac{(2\alpha)!}{(3\alpha)!} \\
\hline
 & & \frac{1}{\alpha!}-\frac{(3\alpha)!}{2(2\alpha)!^2} & \frac{(3\alpha)!}{2(2\alpha)!^2}
\end{array}
$$

5. Theoretical Analysis

To ensure that the obtained numerical solution of FORK algorithms approximates the exact solution of FDEs correctly, we first discuss in this section the consistency of the methods and second the convergence analysis of the FORK methods that may impose some additional conditions under which the approximate solution, discussed in Sections 3 and 4, converges to the exact solution of the problem.

5.1. Consistency

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

$$
\begin{aligned}
&y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, y_{n+1}, h), \quad n = 0, \dots, N^m - 1, \\
&y_0 = y(t_0).
\end{aligned} \tag{35}
$$

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

$$
\begin{aligned}
&y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, h), \quad n = 0, \dots, N^m - 1, \\
&y_0 = y(t_0).
\end{aligned} \tag{36}
$$

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

We define the truncation error τ_n by

$$
\tau_n = \frac{y_{n+1} - y_n}{h^\alpha} - \Phi(t_n, y_n, y_{n+1}, h). \tag{37}
$$

The one-step method (35) and (36) is said to be consistent with Equation (3) if

$$
\lim_{h \to 0} \tau_n = 0, \quad N^m = (T - t_0)/h.
$$

Using (12) and (37), we may write

$$
\begin{aligned}
\lim_{h \to 0} \tau_n &= \lim_{h \to 0} \frac{y_{n+1} - y_n}{h^\alpha} - \lim_{h \to 0} \Phi(t_n, y_n, y_{n+1}, h), \\
&= \frac{1}{\Gamma(\alpha+1)} {}_{t_n}^{c}D_t^\alpha y(t_n) - \lim_{h \to 0} \Phi(t_n, y(t_n), y(t_{n+1}), h), \\
&= \frac{1}{\Gamma(\alpha+1)} F_n(t_n, y(t_n)) - \lim_{h \to 0} \Phi(t_n, y(t_n), y(t_n + h), h).
\end{aligned}
$$

Hence, we may conclude that the proposed one-step IFORK methods are consistent if and only if

$$
\Phi(t, y, y, 0) = \frac{1}{\Gamma(\alpha+1)} F_n(t, y),
$$

or briefly

$$
\Phi(t, y, y, 0) = \frac{1}{\Gamma(\alpha+1)} f(t, y).
$$

Similarly, for explicit methods, we have

$$
\Phi(t, y, 0) = \frac{1}{\Gamma(\alpha+1)} f(t, y).
$$

As an example, consider the two-stage EFORK method (15) with (17),

$$y_{n+1} = y_n + h^\alpha [w_1 f(t_n, y_n) + w_2 f(t_n + c_2 h, y_n + a_{21} K_1)],$$

where, in comparison to (36), we have

$$\Phi(t, y, h) = [w_1 f(t, y) + w_2 f(t + c_2 h, y + a_{21} K_1)],$$

Hence, as h tends to 0, it 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, the two-stage EFORK method (15) is consistent. In addition, the three-stage EFORK method (21) is consistent for

$$\begin{aligned}
&\Phi(t, y, h) = [w_1 f(t, y) + w_2 f(t + c_2 h, y + a_{21} K_1) + w_3 f(t + c_3 h, y + a_{31} K_1 + a_{32} K_2)],\\
&\Phi(t, y, 0) = (w_1 + w_2 + w_3) f(t, y),
\end{aligned}$$

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

Here, 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. *A one-step method of the form (36)*

$$y_{n+1} = y_n + h^\alpha \Phi(t_n, y_n, h), \quad n = 0, 1, 2, \dots, N^m - 1, \tag{38}$$

is said to be regular if the function $\Phi(t, y, h)$ is defined and continuous in the domain $t \in [0, T]$, $y \in [0, T^*]$ and $h \in [0, h_0]$ (h_0 is a positive constant) and if there exists a constant L such that

$$|\Phi(t, y, h) - \Phi(t, z, h)| \le L|y - z|,$$

for every $t \in [0, T]$, $y, z \in [0, T^*]$ and $h \in [0, h_0]$.

To discuss the convergence of the EFORK methods, first, we prove that the given methods in Section 3 are regular. We know from Theorem 2 that $f(t, y)$ satisfies a Lipschitz condition with respect to the second variable. Thus,

$$\begin{aligned}
|\Phi(t_n, y_n, h) - \Phi(t_n, y_n^*, h)| &= h^{-\alpha} \left| \sum_{i=1}^{s} w_i K_i - \sum_{i=1}^{s} w_i K_i^* \right| \\
&\le h^{-\alpha} (w_1 |K_1 - K_1^*| + w_2 |K_2 - K_2^*| + \dots + w_s |K_s - K_s^*|) \\
&\le w_1 L |y_n - y_n^*| + w_2 L |y_n + a_{21} K_1 - y_n^* - a_{21} K_1^*| + \dots \\
&+ w_s L |y_n + a_{s1} K_1 + a_{s2} K_2 + \dots + a_{ss} K_s - y_n^* - a_{s1} K_1^* \\
&- a_{s2} K_2^* - \dots - a_{ss} K_s^*| \\
&\le \dots \le L^* |y_n - y_n^*|.
\end{aligned}$$

Therefore, the function Φ satisfies a Lipschitz condition in y and it is also continuous; thus, EFORK methods are regular. To establish the convergence behavior, we need the following Lemma from [35].

Lemma 1. *Let $\omega_0, \omega_1, \omega_2, \ldots$ be a sequence of real positive numbers that satisfy*

$$\omega_{n+1} \leq (1+\zeta)\omega_n + \mu, \qquad n = 0, 1, 2 \ldots$$

where ζ, μ are positive constants. Then,

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

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

Theorem 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 let $({}_{t_0}^{c}D_t^{\alpha})^{(s+1)}y(t)$ be 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 the EFORK method be consistent, and the method can be written in the form

$$y_{n+1} = y_n + h^{\alpha}\Phi(t_n, y_n, h). \tag{39}$$

The exact value $y(t_n)$ will satisfy

$$y(t_{n+1}) = y(t_n) + h^{\alpha}\Phi(t_n, y(t_n), h) + T_n, \tag{40}$$

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 the regularity of the EFORK method, it follows that

$$|e_{n+1}| \leq |e_n| + h^{\alpha}L|y(t_n) - y_n| + |T_n| \leq (1 + h^{\alpha}L)|e_n| + |T_n|.$$

By using the Lemma 1, 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 a sufficiently large n is constant, i.e., $\mathcal{T} = T_n, n = 0, 1, 2, \ldots$ In addition, 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 $N^m = (T - t_0)/h$, so we have

$$|e_n| \leq O(h^{(p-1)\alpha})\left(\frac{e^{(T-t_0)^{\frac{1}{m}}Lh^{\alpha-\frac{1}{m}}} - 1}{L}\right).$$

Thus, the EFORK methods of Sections 3.1 and 3.2 are convergent if $\alpha - \frac{1}{m} \geq 0$, i.e $m\alpha \geq 1$. Conversely, let the EFORK method be convergent. It is sufficient that we give a limit of (39) as h tends to 0. Now, the proof of the theorem is complete. □

5.3. Stability Analysis

For the stability analysis of the proposed methods in Sections 3 and 4, we consider the FDE:

$$\begin{aligned}&{}^{c}_{t_0}D^{\alpha}_{t}y(t) = \lambda y(t), \quad \lambda \in \mathbb{C}, \quad 0 < \alpha \leq 1,\\ &y(t_0) = y_0\,.\end{aligned} \tag{41}$$

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 two-stage EFORK method (15) to Equation (41) and obtain

$$\begin{aligned}K_1 &= h^{\alpha}f(t_n, y_n) = \lambda h^{\alpha}y_n,\\ K_2 &= h^{\alpha}f(t_n + c_2h, y_n + a_{21}K_1) = \lambda h^{\alpha}(y_n + a_{21}\lambda h^{\alpha}y_n)\\ &= \left[\lambda h^{\alpha} + a_{21}(\lambda h^{\alpha})^2\right]y_n,\\ y_{n+1} &= y_n + w_1K_1 + w_2K_2 = y_n + \frac{1}{2\Gamma(\alpha+1)}\left[2\lambda h^{\alpha} + a_{21}(\lambda h^{\alpha})^2\right]y_n\\ &= \left[1 + \frac{\lambda h^{\alpha}}{\Gamma(\alpha+1)} + \frac{a_{21}(\lambda h^{\alpha})^2}{2\Gamma(\alpha+1)}\right]y_n = \left[1 + \frac{\lambda h^{\alpha}}{\Gamma(\alpha+1)} + \frac{c_2^{\alpha}(\lambda h^{\alpha})^2}{2(\Gamma(\alpha+1))^2}\right]y_n\\ &= \left[1 + \frac{\lambda h^{\alpha}}{\alpha!} + \frac{c_2^{\alpha}(\lambda h^{\alpha})^2}{2(\alpha!)^2}\right]y_n.\end{aligned}$$

Therefore, the growth factor for the two-stage EFORK method (15) is [35]

$$E(\lambda h^{\alpha}) = 1 + \frac{\lambda h^{\alpha}}{\alpha!} + \frac{c_2^{\alpha}(\lambda h^{\alpha})^2}{2(\alpha!)^2},$$

Now, consider the following definition ([35]):

Definition 6. *A numerical method is called absolutely stable in the sense of Dahlquist if and only if $|E(\lambda h^{\alpha})| \leq 1$ when the method is applied with any positive step-size h to the test Equation (41).*

The interval of the absolute stability of a numerical method is defined as $[\mathrm{Re}(\lambda h^{\alpha}), 0)$ if and only if $|E(\lambda h^{\alpha})| \leq 1$. So, the two-stage method (15) is absolutely stable if

$$|1 + \frac{\lambda h^{\alpha}}{\alpha!} + \frac{c_2^{\alpha}(\lambda h^{\alpha})^2}{2(\alpha!)^2}| \leqslant 1\,.$$

If $\lambda\, h^{\alpha} < 0$, we can find the interval of absolute stability as follows:

$$\frac{-2\alpha!}{c_2^{\alpha}} \leqslant \lambda h^{\alpha} < 0. \tag{42}$$

According to (42), the interval of absolute stability for the two-stage EFORK method (15) depends on c_2^{α}. For instance, if $c_2^{\alpha} = 2(\alpha!)^2/(2\alpha)!$, then and so the interval of absolute stability will be

$$\frac{-(2\alpha)!}{\alpha!} \leqslant \lambda h^{\alpha} < 0.$$

In addition, for $c_2^{\alpha} = \frac{(\Gamma(2\alpha+1))^2}{\Gamma(3\alpha+1)\Gamma(\alpha+1)}$, we have

$$\frac{-2(\alpha!)^2(3\alpha)!}{((2\alpha)!)^2} \leqslant \lambda h^{\alpha} < 0.$$

If we choose $c_2^\alpha = \frac{4\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}$, we get

$$\frac{-(3\alpha)!}{2} \leqslant \lambda h^\alpha < 0.$$

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

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

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

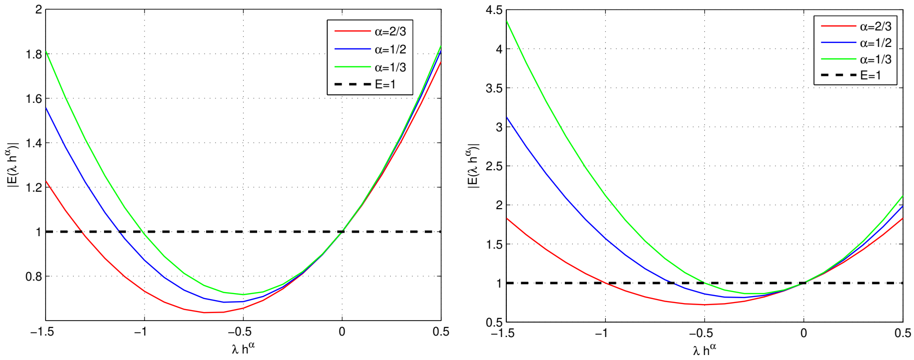

Figure 1. The graph of $E(\lambda h^\alpha)$ for the two-stage EFORK method (15) with $c_2^\alpha = \frac{2(\alpha!)^2}{(2\alpha)!}$ **(left)**, and $c_2^\alpha = \frac{(\Gamma(2\alpha+1))^2}{\Gamma(3\alpha+1)\Gamma(\alpha+1)}$ **(right)**.

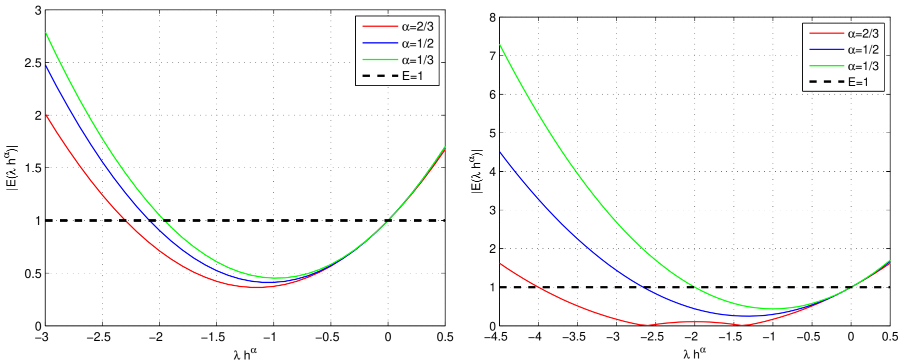

Figure 2. The graph of $E(\lambda h^\alpha)$ for two-stage EFORK method (15) with $c_2^\alpha = \frac{4\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}$ **(left)**, and $c_2^\alpha = \frac{\Gamma(\alpha+1)}{\Gamma(3\alpha+1)}$ **(right)**.

In addition, we apply the three-stage EFORK method (21) to Equation (41) 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 the three-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 three-stage EFORK method is absolutely stable if

$$|1 + \frac{\lambda h^\alpha}{\alpha!} + \frac{(\lambda h^\alpha)^2}{(2\alpha)!} + \frac{(\lambda h^\alpha)^3}{(3\alpha)!}| \leqslant 1.$$

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

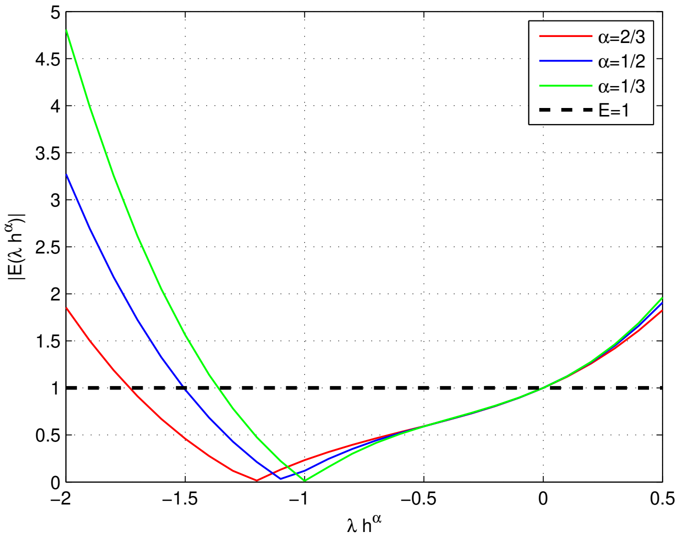

Figure 3. The graph of $E(\lambda h^\alpha)$ for three-stage EFORK method (21).

Finally, we apply the IFORK method (33) to Equation (41) and get

$$E(\lambda h^\alpha) = 1 + \frac{\frac{1}{\alpha!}\lambda h^\alpha}{1 - \lambda h^\alpha \frac{\alpha!}{(2\alpha)!}},$$

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, the interval of absolute stability is very large, and they are stable.

6. Numerical Examples

In order to demonstrate the effectiveness and order of accuracy of the proposed methods in Sections 3 and 4, two examples are considered. All computations have been carried out on a Core i7 PC with Mathematica 13.2 software.

Example 1. *Consider the fractional differential equation*

$$\begin{aligned} &{}_0^cD_t^\alpha y(t) = -y(t) + \frac{t^{4-\alpha}}{\Gamma(5-\alpha)}, \quad t \in [0, T], \\ &y(0) = 0, \end{aligned}$$

such that the exact solution is $y(t) = t^4 E_{\alpha,5}(-t^\alpha)$. *The approximate solutions by the two-stage EFORK method (15), three-stage EFORK method (21), and IFORK methods (33) and (34) are reported in Tables 1–6 (In Appendix A some Mathematica computer programming codes are presented). The computed solutions are compared with the exact solution for different values of* h, α, *and* T. *The absolute error in time* T *is given by*

$$E(h, T) = |y(t_{N^m}) - y_{N^m}|,$$

and the orders of the presented method are computed according to the following relation:

$$Log_2 \frac{E(h, T)}{E(h/2, T)}.$$

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

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/3	1/40	1.09027×10^{-2}	1.1320
	1/80	4.97465×10^{-3}	1.0013
	1/160	2.48509×10^{-3}	0.9136
	1/320	1.31920×10^{-3}	0.8533
	1/640	7.30171×10^{-4}	*

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

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/2	1/40	2.05503×10^{-3}	1.2248
	1/80	8.79256×10^{-4}	1.1621
	1/160	3.92907×10^{-4}	1.1171
	1/320	1.81137×10^{-4}	1.0845
	1/640	8.54183×10^{-5}	*

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

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/4	1/40	9.94252×10^{-4}	0.8437
	1/80	5.54011×10^{-4}	0.8214
	1/160	3.13499×10^{-4}	0.8064
	1/320	1.79258×10^{-4}	0.7958
	1/640	1.03255×10^{-4}	*

Table 4. Three-stage method (21) for $T = 1$, $m = 2$ in Example 1.

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/2	1/40	7.45694×10^{-5}	1.5942
	1/80	2.46986×10^{-5}	1.5789
	1/160	8.26771×10^{-6}	1.5625
	1/320	2.79911×10^{-6}	1.5478
	1/640	9.57367×10^{-7}	*

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

α	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	1.86448×10^{-4}	1.0832	3.56080×10^{-4}	1.7307
	1/80	8.79978×10^{-5}	1.0492	1.07291×10^{-4}	1.6666
	1/160	4.25221×10^{-5}	1.0291	3.37953×10^{-5}	1.6191
	1/320	2.08361×10^{-5}	1.0174	1.10016×10^{-5}	1.5846
	1/640	1.02928×10^{-5}	*	3.66806×10^{-6}	*

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

α	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	3.04275×10^{-4}	0.9083	5.12337×10^{-3}	1.4661
	1/80	1.62123×10^{-4}	0.8746	1.85445×10^{-3}	1.3856
	1/160	8.84207×10^{-5}	0.8482	7.09751×10^{-4}	1.2940
	1/320	4.91160×10^{-5}	0.8255	2.89458×10^{-4}	1.2106
	1/640	2.77152×10^{-5}	*	1.25070×10^{-4}	*

From the Tables 1–6, we can conclude that the computed orders of truncation errors are in good agreement with the obtained results of Sections 3 and 4. Figure 4, illustrates the error curves of the two-stage EFORK method (15) and the three-stage EFORK method (21) at $T = 1$, with $\alpha = 1/2$, $m = 2$ and different values of N.

Example 2. *Consider the following fractional differential equation from [6]:*

$$
\begin{aligned}
&{}_0^c D_t^{\alpha} y(t) = \frac{2}{\Gamma(3-\alpha)} t^{2-\alpha} - \frac{1}{\Gamma(2-\alpha)} t^{1-\alpha} - y(t) + t^2 - t, \quad t \in [0, T], \\
&y(0) = 0,
\end{aligned}
$$

with the exact solution $y(t) = t^2 - t$. Again, for different values of h, α, and T, we compared, in Tables 7–10, the obtained results by the two-stage EFORK method (15) and the three-stage EFORK method (21) with the exact solution. In Tables 11 and 12, we report the results obtained by the IFORK methods (33) and (34). From Tables 7–12, we can conclude that the computed orders are in good agreement with the given results of Sections 3 and 4.

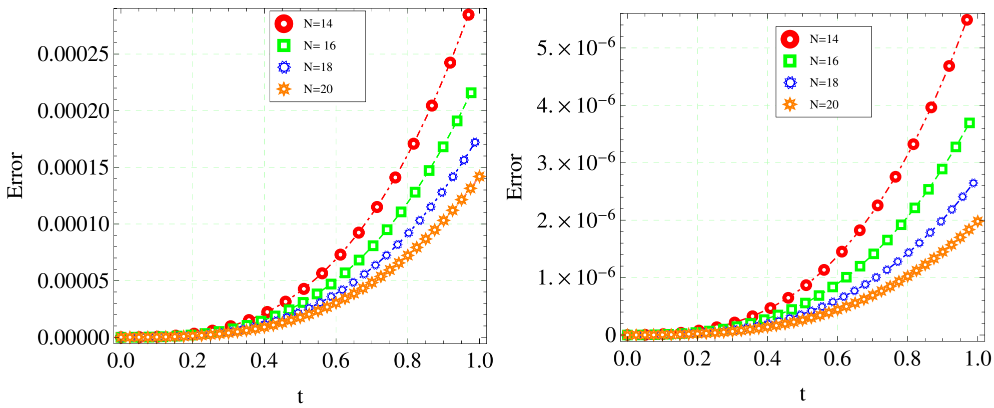

Figure 4. The error curves of the two-stage EFORK method (15) (**left**), and the error curves of the three-stage EFORK method (21) (**right**) for Example 1.

Table 7. Two-stage method (15) for $T = 1$, $m = 3$ in Example 2.

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/3	1/40	1.00356×10^{-1}	1.1075
	1/80	4.65748×10^{-2}	0.9928
	1/160	2.34046×10^{-2}	0.9107
	1/320	1.24493×10^{-2}	0.8523
	1/640	6.89556×10^{-3}	*

Table 8. Two-stage method (15) for $T = 1$, $m = 2$ in Example 2.

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/2	1/40	1.77152×10^{-2}	1.2351
	1/80	7.52581×10^{-3}	1.1738
	1/160	3.33574×10^{-3}	1.1275
	1/320	1.52680×10^{-3}	1.0928
	1/640	7.15859×10^{-4}	*

Table 9. Three-stage method (21) for $T = 1$, $m = 4$ in Example 2.

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/4	1/40	9.90939×10^{-3}	0.8383
	1/80	5.54249×10^{-3}	0.8182
	1/160	3.14342×10^{-3}	0.8047
	1/320	1.79955×10^{-3}	0.7950
	1/640	1.03718×10^{-3}	*

Table 10. Three-stage method (21) for $T = 1$, $m = 2$ in Example 2.

α	h	$E(h,T)$	$Log_2 \frac{E(h,T)}{E(h/2,T)}$
1/2	1/40	5.79341×10^{-4}	1.5592
	1/80	1.96590×10^{-4}	1.5566
	1/160	6.68302×10^{-5}	1.5475
	1/320	2.28624×10^{-5}	1.5374
	1/640	7.87606×10^{-6}	*

Table 11. IFORK methods for $T = 1$, $m = 2$ in Example 2.

α	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	1.52888×10^{-3}	1.0777	2.99223×10^{-3}	1.6608
	1/80	7.24362×10^{-4}	1.0464	9.46328×10^{-4}	1.6244
	1/160	3.50728×10^{-4}	1.0279	3.06932×10^{-4}	1.5943
	1/320	1.72002×10^{-4}	1.0171	1.01649×10^{-4}	1.5703
	1/640	8.49858×10^{-5}	*	3.42297×10^{-5}	*

Table 12. IFORK methods for $T = 1$, $m = 3$ in Example 2.

α	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	2.84493×10^{-3}	0.9005	4.77620×10^{-2}	1.4860
	1/80	1.52403×10^{-3}	0.8707	1.70509×10^{-2}	1.3835
	1/160	8.33483×10^{-4}	0.8463	6.53520×10^{-3}	1.2859
	1/320	4.63592×10^{-4}	0.8246	2.68020×10^{-3}	1.2023
	1/640	2.61754×10^{-4}	*	1.16478×10^{-3}	*

As we can see, the computational orders for two and three-stage EFORK methods approach 2α and 3α for $h > 0$, respectively. As seen in Tables and Figures, the three-stage EFORK method provides better results than the two-stage EFORK method. Table 13 shows the numerical results for different values of T.

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	4.81351×10^{-6}	4.20218×10^{-8}	9.29068×10^{-5}	5.57177×10^{-7}
1.0	8.54183×10^{-5}	9.57367×10^{-7}	3.00335×10^{-3}	7.87606×10^{-6}
1.5	4.51426×10^{-4}	6.00243×10^{-6}	2.78127×10^{-3}	3.63915×10^{-5}
2	1.45778×10^{-3}	2.20455×10^{-5}	6.26033×10^{-3}	9.38735×10^{-5}
3	7.50603×10^{-3}	1.37070×10^{-4}	1.78449×10^{-2}	3.26003×10^{-4}

In addition, Tables 5, 6, 11 and 12 show that the computational order from relations (33) and (34) of IFORK methods approach 2α and 3α for $h > 0$, respectively.

Figure 5, illustrates the numerical results of the two-stage EFORK method (15) and the three-stage EFORK method (21) at $T = 1$ for $\alpha = 1/2$, $m = 2$ and different values of N. Additionally, Figure 6 illustrates the numerical results of the IFORK method (33) for Example 1 and Example 2 at $T = 1$ for $\alpha = 1/2$, $m = 2$ and different values of N.

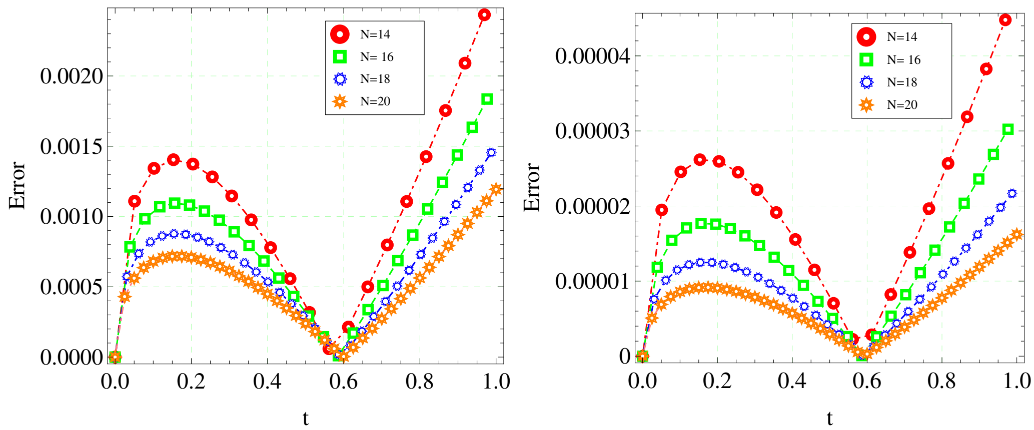

Figure 5. The error curves of the two-stage EFORK method (15) (**left**), and the three-stage EFORK method (21) (**right**) in Example 2.

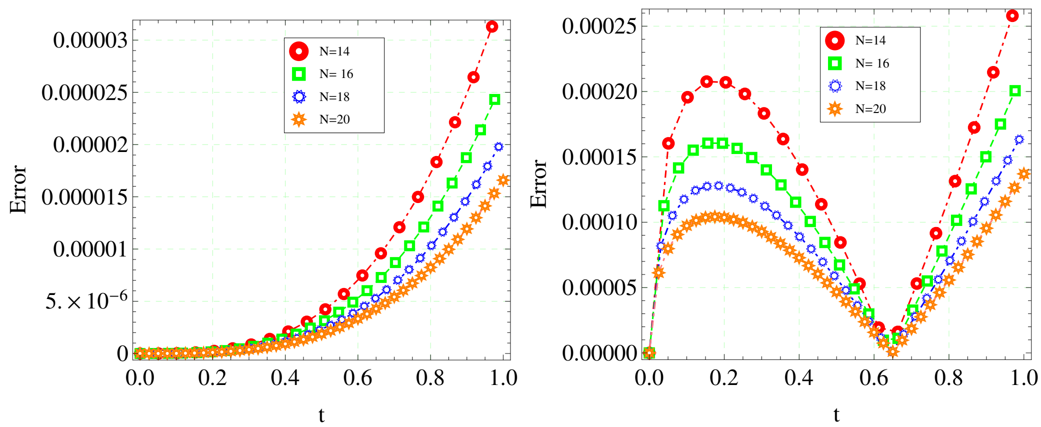

Figure 6. The error curves of the IFORK method (33) in Example 1 (**left**), and Example 2 (**right**).

In addition, the numerical results for the optimal case $c_2^{\alpha} = \frac{(\Gamma(2\alpha+1))^2}{\Gamma(3\alpha+1)\Gamma(\alpha+1)}$ in the two-stage EFORK method are shown in Table 14 with $\alpha = 1/2$ and different values of h.

Table 14. Optimal two-stage method (15) for $T = 1$, $m = 2$.

α	h	$E(h, T)$, Example 1	$E(h, T)$, Example 2
$1/2$	$1/40$	7.35533×10^{-4}	6.12299×10^{-3}
	$1/80$	3.55401×10^{-4}	2.94885×10^{-3}
	$1/160$	1.72778×10^{-4}	1.43060×10^{-3}
	$1/320$	8.45336×10^{-5}	6.99024×10^{-4}
	$1/640$	4.15855×10^{-5}	3.43589×10^{-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.

Author Contributions: Conceptualization, F.G. and R.G.; Methodology, F.G. and R.G.; Formal analysis, F.G. and R.G.; Investigation, F.G. and R.G.; Resources, N.S.; Writing—original draft, F.G. and R.G.; Writing—review and editing, N.S.; Funding acquisition, N.S. All authors have read and agreed to the published version of the manuscript.

Funding: Partial financial support of this work, under Grant No. GP249507 from the Natural Sciences and Engineering Research Council of Canada, is gratefully acknowledged by the third author.

Data Availability Statement: All data and methods used in this research are presented in sufficient detail in the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FORK	Fractional Order Runge–Kutta
EFORK	Explicit Fractional Order Runge–Kutta
IFORK	Implicit Fractional Order Runge–Kutta
FDEs	Fractional Differential Equations

Appendix A. Some Mathematica Codes of FORK Methods

3-stage EFORK method for Example 1.

$N_1 = \text{Input}[\text{“Please Enter } N^m\text{:”}]$;
$T = \text{Input}[\text{“Please Enter T:”}]$;
$\alpha = \text{Input}[\text{“Please Enter } \alpha\text{:”}]$;
$h = \frac{T}{N_1}$;
$\text{Do}[t_n = n * h, \{n, 0, N_1\}]$;
$y_0 = 0$;
$w_1 = \frac{8\Gamma[1+\alpha]^3\Gamma[1+2\alpha]^2 - 6\Gamma[1+\alpha]^3\Gamma[1+3\alpha] + \Gamma[1+2\alpha]\Gamma[1+3\alpha]}{\Gamma[1+\alpha]\Gamma[1+2\alpha]\Gamma[1+3\alpha]}$;
$w_2 = \frac{2\Gamma[1+\alpha]^2\left(4\Gamma[1+2\alpha]^2 - \Gamma[1+3\alpha]\right)}{\Gamma[1+2\alpha]\Gamma[1+3\alpha]}$;
$w_3 = -\frac{8\Gamma[1+\alpha]^2\left(2\Gamma[1+2\alpha]^2 - \Gamma[1+3\alpha]\right)}{\Gamma[1+2\alpha]\Gamma[1+3\alpha]}$;
$a_{11} = \frac{1}{2*\Gamma[\alpha+1]^2}$;

$a_{21} = \frac{\Gamma[1+\alpha]^2\Gamma[1+2\alpha]+2\Gamma[1+2\alpha]^2-\Gamma[1+3\alpha]}{4\Gamma[1+\alpha]^2(2\Gamma[1+2\alpha]^2-\Gamma[1+3\alpha])}$;
$a_{22} = -\frac{\Gamma[1+2\alpha]}{4(2\Gamma[1+2\alpha]^2-\Gamma[1+3\alpha])}$;
$c_2 = \left(\frac{1}{2\Gamma[1+\alpha]}\right)^{1/\alpha}$;
$c_3 = \left(\frac{1}{4\Gamma[1+\alpha]}\right)^{1/\alpha}$;
$f_0[\mathrm{t\_}, \mathrm{y\_}] = -y + \frac{t^{4-\alpha}}{\Gamma[5-\alpha]}$;
Do[
$K_1 = h^\alpha f_n[t_n, y_n]$;
$K_2 = h^\alpha f_n[t_n + c_2 * h, y_n + a_{11} * K_1]$;
$k_3 = h^\alpha f_n[t_n + c_3 * h, y_n + a_{22} * K_2 + a_{21} * K_1]$;
$y_{n+1} = y_n + w_1 * K_1 + w_2 * K_2 + w_3 * k_3$;
Print[“n=”, n, “: Explicit Error=”, Abs$[y_{n+1} - {t_{n+1}}^4 * N[\text{MittagLefflerE}[\alpha, 5, -{t_{n+1}}^\alpha]]]$];
$f_{n+1}[\mathrm{t\_}, \mathrm{y\_}] = f_n[t, y] - (y_{n+1} - y_n) * \frac{(t-t_n)^{(1-\alpha)} - (t-t_{n+1})^{(1-\alpha)}}{h*(1-\alpha)*\Gamma[1-\alpha]}$;
, $\{n, 0, N_1 - 1\}$]

2-stage IFORK method for Example 1.
$N_1 =$ Input[“Please Enter N^m:”];
$T =$ Input[“Please Enter T:”];
$\alpha =$ Input[“Please Enter α:”];
$h = \frac{T}{N_1}$;
Do[$t_n = n * h, \{n, 0, N_1\}$];
$y_0 = 0$;
$w_1 = \frac{1}{\Gamma[1+\alpha]} - \frac{\Gamma[1+3\alpha]}{2\Gamma[1+2\alpha]^2}$;
$w_2 = \frac{\Gamma[1+3\alpha]}{2\Gamma[1+2\alpha]^2}$;
$a_{22} = \frac{\Gamma[1+2\alpha]}{\Gamma[1+3\alpha]}$;
$c_{21} = (\frac{1}{\Gamma[1+3\alpha]^2}(2\Gamma[1+\alpha]\Gamma[1+2\alpha]\Gamma[1+3\alpha]$
$-\sqrt{2}\sqrt{\Gamma[1+2\alpha]^2(-2\Gamma[1+\alpha]^2 + \Gamma[1+2\alpha])\Gamma[1+3\alpha]^2}))^{1/\alpha}$;
$c_{22} = (\frac{1}{\Gamma[1+3\alpha]^2}(2\Gamma[1+\alpha]\Gamma[1+2\alpha]\Gamma[1+3\alpha]$
$+\sqrt{2}\sqrt{\Gamma[1+2\alpha]^2(-2\Gamma[1+\alpha]^2 + \Gamma[1+2\alpha])\Gamma[1+3\alpha]^2}))^{1/\alpha}$;
$a_{21} = \frac{\Gamma[1+2\alpha]}{\Gamma[1+3\alpha]}$;
$f_0[\mathrm{t\_}, \mathrm{y\_}] = -y + \frac{t^{4-\alpha}}{\Gamma[5-\alpha]}$;
Do[
$K_1 = h^\alpha f_n[t_n, y_n]$;
$K_2 = \frac{1}{2} * h^\alpha \frac{f_n[t_n + c_{21}*h, y_n + a_{21}*K_1] + f_n[t_n + c_{22}*h, y_n + a_{21}*K_1]}{1 + h^\alpha * a_{22}}$;
$y_{n+1} = y_n + w_1 * K_1 + w_2 * K_2$;
Print[“n=”, n, “: Implicit Error=”, Abs$[y_{n+1} - {t_{n+1}}^4 * N[\text{MittagLefflerE}[\alpha, 5, -{t_{n+1}}^\alpha]]]$];
$f_{n+1}[\mathrm{t\_}, \mathrm{y\_}] = f_n[t, y] - (y_{n+1} - y_n) * \frac{(t-t_n)^{(1-\alpha)} - (t-t_{n+1})^{(1-\alpha)}}{h*(1-\alpha)*\Gamma[1-\alpha]}$;
, $\{n, 0, N_1 - 1\}$]

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. [CrossRef]
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. [CrossRef]

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. [CrossRef]
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. [CrossRef]
10. Garrappa, R. Numerical Solution of Fractional Differential Equations: A Survey and a Software. *Mathematics* **2018**, *6*, 16. [CrossRef]
11. Sheng, C.; Cao, D.; Shen, J. Efficient spectral methods for PDEs with spectral fractional Laplacian. *J. Sci. Comput.* **2021**, *88*, 4. [CrossRef]
12. Zhang, T.; Li, Y. Exponential Euler scheme of multi-delay Caputo–Fabrizio fractional-order differential equations. *Appl. Math. Lett.* **2021**, *124*, 107709. [CrossRef]
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. [CrossRef]
14. Diethelm, K.; Walz, G. Numerical solution of fractional order differential equation by extrapolation. *Numer. Algorithms* **1997**, *16*, 231–253. [CrossRef]
15. Diethelm, K.; Ford, N.J.; Freed, A.D. A Predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dynam.* **2002**, *29*, 3–22. [CrossRef]
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. [CrossRef]
18. Sun, Z.Z.; Wu, X. A fully discrete difference scheme for a diffusion-wave system. *Appl. Numer. Math.* **2006**, *56*, 193–209. [CrossRef]
19. Gao, G.; Sun, Z.Z.; Zhang, H. A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications. *J. Comput. Phys.* **2014**, *259*, 33–50. [CrossRef]
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. [CrossRef]
21. Odibat, Z. Approximations of fractional integrals and Caputo fractional derivatives. *Appl. Math. Comput.* **2006**, *178*, 527–533. [CrossRef]
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. [CrossRef]
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. [CrossRef]
25. Houwen, P. The development of Runge-Kutta methods for partial differential equations. *Appl. Numer. Math.* **1996**, *20*, 261–272. [CrossRef]
26. Butcher, J.C. Trees and numerical methods for ordinary differential equations. *Numer. Algorithms* **2010**, *53*, 153–170. [CrossRef]
27. Butcher, J.C. Practical Runge Kutta methods for scientific computation. *ANZIAM J.* **2009**, *50*, 333–342. [CrossRef]
28. Lubich, C. Runge-Kutta theory for Volterra and Abel integral equations of the second kind. *Math. Comput.* **1983**, *41*, 87–102. [CrossRef]
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. [CrossRef]
30. Izzo, G.; Russo, E.; Chiapparelli, C. Highly stable Runge–Kutta methods for Volterra integral equations. *Appl. Numer. Math.* **2012**, *62*, 1002–1013. [CrossRef]
31. Ostermann, A.; Roche, M. Runge-Kutta Methods for Partial Differential Equations and Fractional Orders of Convergence. *Math. Comp.* **1992**, *59*, 403–420. [CrossRef]
32. Ostermann, A.; Roche, M. Rosenbrock methods for partial differential equations and fractional orders of convergence. *SIAM J. Numer. Anal.* **1993**, *30*, 1084–1098. [CrossRef]
33. Srivastava, H.M. Operators of Fractional Calculus and Their Applications. *Mathematics* **2019**, *6*, 157. [CrossRef]
34. Odibat, Z.; Shawagfeh, N. Generalized Taylor's formula. *Appl. Math. Comput.* **2007**, *186*, 286–293. [CrossRef]
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. [CrossRef]
37. Youssri, Y.H. Orthonormal ultraspherical operational matrix algorithm for fractal–fractional Riccati equation with generalized Caputo derivative. *Fractal Fract.* **2021**, *5*, 100. [CrossRef]