A Unified Caputo–ABC Fractional Framework for High-Order Iterative Methods in Nonlinear Equations

Mudassir Shams and Bruno Carpentieri *

Posted Date: 9 April 2026

doi: 10.20944/preprints202604.0702.v1

Keywords: fractional iterative methods; Caputo derivative; Atangana–Baleanu–Caputo derivative; nonlinear equations; high-order methods; convergence analysis; basins of attraction; Wada measure

 doi:10.20944/preprints202604.0702.v1

Article

A Unified Caputo–ABC Fractional Framework for High-Order Iterative Methods in Nonlinear Equations

Mudassir Shams [1,2] and Bruno Carpentieri [3,*]

[1] Department of Mathematics, Faculty of Arts and Science, Balikesir University, 10145 Balıkesir, Turkey
[2] Department of Mathematics and Statistics, Riphah International University, 44000 Islamabad, Pakistan
[3] Faculty of Engineering, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
* Correspondence: bruno.carpentieri@unibz.it

Abstract

Nonlinear equations arise extensively in engineering and applied sciences. This study introduces a family of Caputo and Atangana–Baleanu–Caputo (ABC) fractional-order iterative methods for solving nonlinear problems. The proposed schemes are designed to enhance convergence behavior and improve robustness compared to existing fractional Newton-type methods. Local convergence is analyzed using fractional Taylor expansions, establishing the order of convergence and associated error equations. In addition, a dynamical systems perspective is adopted to investigate global convergence properties through basin of attraction analysis, including fractal structures and the Wada measure. Numerical experiments on application-inspired nonlinear models demonstrate that the proposed methods achieve faster error reduction, lower residuals, and improved computational efficiency compared to existing schemes. These results indicate that the proposed framework provides an effective and flexible approach for solving nonlinear equations, combining accuracy, stability, and dynamical insight.

Keywords: fractional iterative methods; Caputo derivative; Atangana–Baleanu–Caputo derivative; nonlinear equations; high-order methods; convergence analysis; basins of attraction; Wada measure

1. Introduction

Nonlinear equations play a central role in a wide range of problems in engineering, physics, and applied sciences, arising in models of fluid dynamics, heat transfer, biological systems, signal processing, and nonlinear control [1–3]. These problems are typically formulated as

$$f(x) = 0, \qquad f : \mathbb{R} \to \mathbb{R}, \tag{1}$$

where f is a nonlinear operator. In most practical settings, analytical solutions are not available, particularly for high-dimensional or strongly nonlinear systems. This motivates the development of efficient and robust iterative numerical methods [4].

Among classical approaches, the Newton–Raphson method [1] is one of the most widely used schemes, defined by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \cdots, \tag{2}$$

and characterized by quadratic convergence under standard assumptions. To further accelerate convergence, several higher-order extensions have been proposed, including Traub [5] and Steffensen-type methods [6]. For instance, the classical Traub scheme is given by

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f'(x_n)}. \end{cases} \tag{3}$$

 doi:10.20944/preprints202604.0702.v1

Despite their effectiveness, these methods remain sensitive to initial guesses, exhibit restricted domains of convergence, and may deteriorate in highly nonlinear or stiff regimes. Such limitations motivate the development of more flexible frameworks capable of improving both convergence behavior and robustness.

In this context, fractional calculus [7] has emerged as a powerful tool for enhancing iterative methods. By extending classical integer-order derivatives to non-integer orders, fractional operators introduce memory and nonlocal effects. These features can significantly influence the convergence behavior of numerical schemes. Among the most widely used formulations, the Caputo fractional derivative [8], defined for $\gamma \in (0, 1]$ as

$$ {}^{c}\mathcal{D}^{\gamma} f(x) = \frac{1}{\Gamma(1-\gamma)} \int_0^x \frac{f'(t)}{(x-t)^{\gamma}}\, dt, \tag{4} $$

provides a natural framework for generalizing classical root-finding methods.

In particular, the fractional Newton method [11], denoted by SNF_*, extends the classical Newton scheme as

$$ x_{n+1} = x_n - \left[\Gamma(\gamma+1)\frac{f(x_n)}{{}^{c}\mathcal{D}^{\gamma} f(x_n)}\right]^{1/\gamma}, \tag{5} $$

and achieves a convergence order of $\gamma + 1$, with local error of the form

$$ e_{n+1} = C\, e_n^{\gamma+1} + \mathcal{O}(e_n^{2\gamma+1}), \tag{6} $$

where C is a constant depending on higher-order fractional derivatives of f at the root.

Although this fractional extension introduces additional flexibility through the parameter γ and incorporates memory effects, it still inherits several limitations of classical schemes, including sensitivity to initial conditions and reduced robustness in stiff or strongly nonlinear regimes.

In addition to the Caputo formulation, the Atangana–Baleanu–Caputo ($\mathcal{ABC}$) fractional derivative [12] has recently attracted considerable attention due to its improved analytical and numerical properties. It is defined as

$$ {}^{abc}\mathcal{D}^{\gamma} f(x) = \frac{B(\gamma)}{1-\gamma} \int_0^x f'(t)\, E_{\gamma}\!\left(-\frac{\gamma}{1-\gamma}(x-t)^{\gamma}\right) dt, \tag{7} $$

where $E_{\gamma}(\cdot)$ denotes the Mittag–Leffler function.

Unlike the Caputo derivative, which involves a singular kernel $(x-t)^{-\gamma}$, the $\mathcal{ABC}$ operator is characterized by a smooth, non-singular Mittag–Leffler kernel. This distinction has important numerical implications: the $\mathcal{ABC}$ formulation mitigates numerical instability and reduces excessive memory effects, leading to enhanced robustness in iterative computations. Consequently, $\mathcal{ABC}$-based schemes often exhibit improved convergence behavior and larger domains of attraction, particularly in strongly nonlinear regimes.

A corresponding $\mathcal{ABC}$-based Newton-type iteration can be written as

$$ x_{n+1} = x_n - \left(\Gamma(\gamma+1)\frac{f(x_n)}{{}^{abc}\mathcal{D}^{\gamma} f(x_n)}\right)^{\frac{1}{\gamma}}. \tag{8} $$

Despite these advantages, most existing approaches employ either Caputo or $\mathcal{ABC}$ operators in isolation, without exploring unified formulations capable of exploiting their complementary properties. As a result, the potential of fractional iterative schemes to balance accuracy, stability, and robustness remains only partially realized.

Recent contributions have introduced high-order fractional parallel iterative schemes for nonlinear equations, often combining convergence acceleration with dynamical system analysis to improve stability and robustness [13]. In this context, various modifications of the fractional Newton framework have been proposed, incorporating nonlinear correction terms and multi-step strategies. For instance,

 doi:10.20944/preprints202604.0702.v1

Shams *et al.* [14] introduced a modified single-step scheme with a nonlinear correction factor that enhances stability while preserving the convergence order $\gamma + 1$. Predictor–corrector strategies, such as the two-stage scheme $ECFS_1$ [11], further improve convergence through an additional correction step. More advanced formulations include the two-step method $ECFS_2$ [15], which achieves order $2\gamma + 1$ via nonlinear ratio-based corrections, and the scheme $ECFS_3$ [16], where auxiliary functions are introduced to further enhance convergence behavior and stability.

Although these methods improve convergence properties, they often come at the cost of increased computational complexity and remain primarily focused on local convergence. Moreover, they are typically developed within a single fractional framework, further limiting their ability to exploit complementary operator properties.

In summary, although classical and fractional iterative schemes provide a solid theoretical foundation and achieve increasingly high orders of convergence, their practical performance remains constrained by several key limitations. In particular, higher convergence orders are often accompanied by reduced robustness and increased sensitivity in highly nonlinear or stiff regimes. Moreover, the evaluation of fractional derivatives introduces non-negligible computational overhead due to memory effects, especially in large-scale applications [17,18]. Another important limitation is that most existing approaches are primarily analyzed from a local convergence perspective, offering limited insight into their global dynamical behavior. In addition, they are typically developed within a single fractional framework, without fully exploiting the complementary advantages of Caputo and $\mathcal{ABC}$ operators.

These observations highlight the need for fractional iterative schemes that can simultaneously balance convergence order, stability, and computational efficiency, while also providing a more comprehensive understanding of their global convergence behavior. Motivated by the above challenges, this work introduces a family of fractional-order iterative methods that combine Caputo and $\mathcal{ABC}$ derivatives within a unified framework, leveraging their complementary kernel properties. The proposed approach aims to balance convergence speed, stability, and computational efficiency by leveraging the complementary properties of the two fractional operators.

The main contributions of this work can be summarized as follows:

- Development of a class of high-order fractional iterative schemes that extend existing Newton-type methods;
- Formulation of a unified framework integrating Caputo and $\mathcal{ABC}$ operators to exploit their complementary characteristics;
- Derivation of convergence orders and associated error equations using fractional Taylor expansions;
- Introduction of tunable fractional parameters to enhance stability and flexibility;
- Analysis of global convergence behavior through basin dynamics and fractal structures;
- Numerical evidence demonstrating improved efficiency compared to existing fractional methods.

The novelty of the proposed approach lies in the unified use of Caputo and $\mathcal{ABC}$ operators within a single iterative structure, together with a dynamical systems perspective that provides insight into both local and global convergence behavior. The remainder of the paper is organized as follows. Section 2 presents the proposed methods and their convergence analysis. Section 3 reports the numerical results. Section 4 examines the dynamical behavior and discusses practical implementation in engineering applications. Finally, Section 5 concludes the paper.

2. Fractional-Order Schemes and Convergence Analysis

This section introduces the proposed fractional-order iterative schemes and establishes their local convergence properties. Starting from a Newton-type predictor–corrector structure, we construct two bi-parametric methods based on the Caputo and Atangana–Baleanu–Caputo fractional derivatives. The fractional order $\gamma \in (0, 1]$ and the parameters $\lambda_1, \lambda_2 \in \mathbb{R}$ provide additional flexibility to control convergence speed and stability. The convergence analysis is carried out using fractional Taylor expansions around a simple root of the nonlinear equation $f(x) = 0$.

 doi:10.20944/preprints202604.0702.v1

Let $f : \mathbb{R} \to \mathbb{R}$ be sufficiently smooth in a neighborhood of a simple root ξ. For convenience, we introduce the notation

$$ {}^{c}\mathcal{D}f(x) := {}^{c}\mathcal{D}^{\gamma}f(x), \qquad {}^{abc}\mathcal{D}f(x) := {}^{abc}\mathcal{D}^{\gamma}f(x), \qquad G_a := \Gamma(\gamma + a), $$

and define

$$ G_1 := \Gamma(\gamma + 1), \qquad \rho_n := \frac{f(y_n)}{f(x_n)}. $$

The proposed Caputo-based scheme, denoted by NCFS_1, is given by

$$ \begin{cases} y_n = x_n - \left[G_1 \dfrac{f(x_n)}{{}^{c}\mathcal{D}f(x_n)} \right]^{1/\gamma}, \\ x_{n+1} = y_n - \left[G_1 \dfrac{f(y_n)}{{}^{c}\mathcal{D}f(x_n)} \left(\lambda_1 \rho_n + \dfrac{1 + (\lambda_2 + 2)\rho_n}{1 + \lambda_2 \rho_n} \right) \right]^{1/\gamma}, \end{cases} \tag{9} $$

where $\lambda_1, \lambda_2 \in \mathbb{R}$.

Replacing the Caputo derivative in (9) with the Atangana–Baleanu–Caputo derivative yields the corresponding ABC-based bi-parametric scheme, denoted by NFS_1^{abc}:

$$ \begin{cases} y_n = x_n - \left[G_1 \dfrac{f(x_n)}{{}^{abc}\mathcal{D}f(x_n)} \right]^{1/\gamma}, \\ x_{n+1} = y_n - \left[G_1 \dfrac{f(y_n)}{{}^{abc}\mathcal{D}f(x_n)} \left(\lambda_1 \rho_n + \dfrac{1 + (\lambda_2 + 2)\rho_n}{1 + \lambda_2 \rho_n} \right) \right]^{1/\gamma}, \end{cases} \tag{10} $$

where $\lambda_1, \lambda_2 \in \mathbb{R}$.

2.1. Local Convergence

We now establish the local convergence order of the Caputo-based scheme NCFS_1.

Theorem 1 (Local convergence of NCFS_1)**.** *Let f be sufficiently smooth (in the sense required for the existence of the involved fractional derivatives) in a neighborhood of a simple root ξ such that $f(\xi) = 0$ and $f'(\xi) \neq 0$. If the initial guess x_0 is sufficiently close to ξ, then the method (9) converges locally to ξ with order*

$$ p = 3\gamma + 1. $$

More precisely, the error satisfies

$$ e_{n+1} = \mathcal{H}(\gamma, \lambda_1, \lambda_2)\, e_n^{3\gamma+1} + \mathcal{O}(e_n^{4\gamma+1}), \tag{11} $$

where $e_n = x_n - \xi$ and $\mathcal{H}$ is a finite constant depending on the fractional derivatives of f at ξ.

Proof. Let $e_n = x_n - \xi$. We first expand $f(x_n)$ and ${}^{c}\mathcal{D}f(x_n)$ around the simple root ξ, and then propagate these expansions through the predictor step of (9). Using the fractional Taylor expansion of f about ξ, we obtain

$$ f(x_n) = \frac{{}^{c}\mathcal{D}f(\xi)}{G_1} \left(e_n^{\gamma} + K_2 e_n^{2\gamma} + K_3 e_n^{3\gamma} + K_4 e_n^{4\gamma} + \mathcal{O}(e_n^{5\gamma}) \right), \tag{12} $$

where

$$ K_j = \frac{G_1}{\Gamma(j\gamma + 1)} \frac{{}^{c}\mathcal{D}^{j\gamma}f(\xi)}{{}^{c}\mathcal{D}^{\gamma}f(\xi)}, \qquad j \geq 2. $$

 doi:10.20944/preprints202604.0702.v1

Taking the Caputo fractional derivative of (12), we obtain

$$ {}^{c}\mathcal{D}f(x_n) = \frac{{}^{c}\mathcal{D}f(\xi)}{G_1}\left[G_1 + \frac{\Gamma(2\gamma+1)K_2}{\Gamma(\gamma+1)}e_n^{\gamma} + \frac{\Gamma(3\gamma+1)K_3}{\Gamma(2\gamma+1)}e_n^{2\gamma} + \mathcal{O}(e_n^{3\gamma})\right]. \tag{13} $$

Hence, by dividing (12) by (13), we obtain

$$ \frac{f(x_n)}{{}^{c}\mathcal{D}f(x_n)} = \left(\frac{1}{G_1}\right)e_n^{\gamma} + \left(\frac{K_2}{G_1} - \frac{2^{2\gamma}K_2}{\sqrt{\pi}\,G_1^2}\,G_{\frac{1}{2}}\right)e_n^{2\gamma} + \mathcal{A}e_n^{3\gamma} + \mathcal{O}(e_n^{4\gamma}), \tag{14} $$

where

$$ \mathcal{A} = \frac{K_3}{G_1} - \frac{2^{2\gamma}K_2^2}{\sqrt{\pi}\,G_1^2}\,G_{\frac{1}{2}} - \frac{3^{3\gamma}\sqrt{3}\,K_3}{2\sqrt{\pi}\,G_1^2(2^{\gamma})^2}\frac{G_{\frac{1}{3}}G_{\frac{2}{3}}}{G_{\frac{1}{2}}} + \frac{2^{4\gamma}K_2^2}{G_1^3\pi}\left(G_{\frac{1}{2}}\right)^2. $$

Using (14) and applying the binomial expansion to the fractional power $(\cdot)^{1/\gamma}$, the predictor step in (9) yields

$$ y_n - \xi = \left(-K_2 + \frac{2^{2\gamma}K_2}{G_1\sqrt{\pi}}\,G_{1/2}\right)e_n^{\gamma+1} + \mathcal{B}e_n^{2\gamma+1} + \mathcal{O}(e_n^{3\gamma+1}), \tag{15} $$

where

$$ \mathcal{B} = -K_3 + \frac{2^{2\gamma}K_2^2}{G_1\sqrt{\pi}}\,G_{1/2} + \frac{3^{3\gamma}\sqrt{3}\,K_3}{2G_1\sqrt{\pi}(2^{\gamma})^2}\frac{G_{1/3}G_{2/3}}{G_{1/2}} - \frac{2^{4\gamma}K_2^2}{G_1^2\pi}(G_{1/2})^2. $$

Using the expansion of $y_n - \xi$ obtained above and applying the Taylor expansion of f about the simple root ξ, we obtain

$$ f(y_n) = \mathcal{D}_1 e_n^{\gamma+1} + \mathcal{D}_2 e_n^{2\gamma+1} + \mathcal{D}_3 e_n^{3\gamma+1} + \mathcal{O}(e_n^{4\gamma+1}). \tag{16} $$

where

$$ \begin{aligned} \mathcal{D}_1 &= -\frac{K_2}{G_1\sqrt{\pi}}\left(G_1\sqrt{\pi} - 2^{2\gamma}G_{1/2}\right), \\ \mathcal{D}_2 &= -\frac{2^{6\gamma+1}(G_{1/2})^3K_2^2\sqrt{\pi} - 2^{4\gamma+1}(G_{1/2})^2K_2^2G_1\pi}{2G_1^2\pi^{3/2}2^{2\gamma}}, \\ \mathcal{D}_3 &= \frac{2^{2\gamma+1}K_3G_1^2\pi^{3/2}G_{1/2} - K_3 3^{3\gamma}\sqrt{3}\,G_{1/3}G_{2/3}G_1\pi}{2G_1^2\pi^{3/2}2^{2\gamma}} + \frac{K_2^3}{G_1^2\pi}\left(G_1\sqrt{\pi} - 2^{2\gamma}G_{1/2}\right)^2. \end{aligned} $$

Dividing the above expansion of $f(y_n)$ by (12), we obtain

$$ \rho_n = \frac{f(y_n)}{f(x_n)} = \left(-K_2 + \frac{2^{2\gamma}K_2}{G_1\sqrt{\pi}}\,G_{1/2}\right)e_n^{\gamma} + \mathcal{S}e_n^{\gamma+1} + \mathcal{O}(e_n^{2\gamma+1}), \tag{17} $$

where

$$ \mathcal{S} = K_2^2 - K_3 + \frac{3^{3\gamma}\sqrt{3}\,K_3}{2\,G_1\sqrt{\pi}\,(2^{\gamma})^2}\frac{G_{1/3}\,G_{2/3}}{G_{1/2}} - \frac{2^{4\gamma}K_2^2}{G_1^2\pi}\,(G_{1/2})^2. $$

Therefore,

$$ 1 + \lambda_2\rho_n = 1 + \left(-\lambda_2K_2 + \frac{2^{2\gamma}\lambda_2K_2}{G_1\sqrt{\pi}}\,G_{1/2}\right)e_n^{\gamma} + \mathcal{F}_0e_n^{\gamma+1} + \mathcal{O}(e_n^{2\gamma+1}), \tag{18} $$

where

$$ \mathcal{F}_0 = \lambda_2\mathcal{S}. \tag{19} $$

 doi:10.20944/preprints202604.0702.v1

Next, we expand the rational correction factor appearing in the second sub-step of (9):

$$\Theta_n = \left(1 + \frac{(\lambda_2+2)f(y_n)}{f(x_n)}\right)\left(1 + \frac{\lambda_2 f(y_n)}{f(x_n)}\right)^{-1} = 1 + \mathcal{F}_1 e_n^{\gamma} + \mathcal{F}_2 e_n^{\gamma+1} + \mathcal{F}_3 e_n^{2\gamma+1} + \mathcal{O}(e_n^{3\gamma+1}), \quad (20)$$

where

$$\mathcal{F}_1 = -2K_2 + 2\,\frac{2^{2\gamma}K_2}{G_1\sqrt{\pi}}\,G_{1/2},$$

$$\mathcal{F}_2 = 2K_2^2 - 2K_3 - 2\lambda_2 K_2^2 - 2\,\frac{2^{4\gamma}K_2^2}{G_1^2\pi}\,G_{1/2}^2 + 4\,\frac{\lambda_2 2^{2\gamma}K_2^2}{G_1\sqrt{\pi}}\,G_{1/2} - 2\,\frac{\lambda_2 2^{4\gamma}K_2^2}{G_1^2\pi}\,G_{1/2}^2 + \frac{3^{3\gamma}\sqrt{3}\,K_3}{G_1\sqrt{\pi}\,2^{2\gamma}}\,\frac{G_{1/3}G_{2/3}}{G_{1/2}},$$

$$\mathcal{F}_3 = 4K_2K_3 - 4\lambda_2 K_2K_3 + 4\lambda_2 K_2^3 - 2\lambda_2^2 K_2^3 - 4\,\frac{2^{2\gamma}K_2^3}{G_1\sqrt{\pi}}\,G_{1/2} - 4\,\frac{\lambda_2 2^{2\gamma}K_2^3}{G_1\sqrt{\pi}}\,G_{1/2} + 6\,\frac{\lambda_2^2 2^{2\gamma}K_2^3}{G_1\sqrt{\pi}}\,G_{1/2} + 4\,\frac{2^{4\gamma}K_2^3}{G_1^2\pi}\,G_{1/2}^2 - 4\,\frac{\lambda_2 2^{4\gamma}K_2^3}{G_1^2\pi}\,G_{1/2}^2 - 6\,\frac{\lambda_2^2 2^{4\gamma}K_2^3}{G_1^2\pi}\,G_{1/2}^2 + 2\,\frac{\lambda_2^2 2^{6\gamma}K_2^3}{G_1^3\pi^{3/2}}\,G_{1/2}^3 + 4\,\frac{\lambda_2 2^{6\gamma}K_2^3}{G_1^3\pi^{3/2}}\,G_{1/2}^3 - \frac{3^{3\gamma}\sqrt{3}\,K_2K_3}{G_1\sqrt{\pi}\,2^{2\gamma}}\,\frac{G_{1/3}G_{2/3}}{G_{1/2}} + 2\,\frac{\lambda_2 3^{3\gamma}\sqrt{3}\,K_2K_3}{G_1\sqrt{\pi}\,2^{2\gamma}}\,\frac{G_{1/3}G_{2/3}}{G_{1/2}}.$$

Therefore, the error of Ψ_n is computed as

$$\Psi_n = \lambda_1 f(y_n) + \left(1 + \frac{(\lambda_2+2)f(y_n)}{f(x_n)}\right)\left(1 + \frac{\lambda_2 f(y_n)}{f(x_n)}\right)^{-1} = 1 + \mathcal{Z}_1 e_n^{\gamma} + \mathcal{Z}_2 e_n^{\gamma+1} + \mathcal{O}(e_n^{2\gamma+1}), \quad (21)$$

where

$$\mathcal{Z}_1 = -2K_2 + 2\,\frac{2^{2\gamma}K_2}{\sqrt{\pi}G_1}\,G_{1/2},$$

$$\mathcal{Z}_2 = 2K_2^2 - 2K_3 - 2\lambda_2 K_2^2 - \lambda_1 K_2 - 2\,\frac{2^{4\gamma}K_2^2}{G_1^2\pi}(G_{1/2})^2 - 2\,\frac{\lambda_2 2^{4\gamma}K_2^2}{G_1^2\pi}(G_{1/2})^2 + 4\,\frac{\lambda_2 2^{2\gamma}K_2^2}{G_1\sqrt{\pi}}G_{1/2} + \frac{\lambda_1 2^{2\gamma}K_2}{G_1\sqrt{\pi}}G_{1/2} + \frac{3^{3\gamma}\sqrt{3}\,K_3}{G_1\sqrt{\pi}\,2^{2\gamma}}\,\frac{G_{1/3}G_{2/3}}{G_{1/2}}.$$

Substituting the above expansions into the second sub-step of (9) and applying the binomial expansion once again, we obtain

$$x_{n+1} - \xi = y_n - \xi - \left[G_1\,\frac{f(y_n)}{{}_c\mathcal{D}f(x_n)}\left(\lambda_1\rho_n + \frac{1+(\lambda_2+2)\rho_n}{1+\lambda_2\rho_n}\right)\right]^{1/\gamma} \quad (22)$$

$$e_{n+1} = \mathcal{H}(\gamma,\lambda_1,\lambda_2)\,e_n^{3\gamma+1} + \mathcal{O}(e_n^{4\gamma+1}), \quad (23)$$

where

$$\mathcal{H} = -2\lambda_2 K_2^3 - \lambda_1 K_2^2 + K_2^3 - 4K_2K_3 + 3\,\frac{2^{2\gamma}K_2K_3}{G_1\sqrt{\pi}}\,G_{1/2} + 2\,\frac{\lambda_1\,2^{2\gamma}K_2^2}{G_1\sqrt{\pi}}\,G_{1/2} - \frac{\lambda_1\,2^{4\gamma}K_2^2}{G_1^2\pi}\,(G_{1/2})^2 + 6\,\frac{\lambda_2 2^{2\gamma}K_2^3}{G_1\sqrt{\pi}}\,G_{1/2} - 6\,\frac{\lambda_2 2^{4\gamma}K_2^3}{G_1^2\pi}\,(G_{1/2})^2 + 2\,\frac{\lambda_2 2^{6\gamma}K_2^3}{G_1^3\pi^{3/2}}\,(G_{1/2})^3 + \frac{3\,3^{3\gamma}\sqrt{3}\,K_2K_3}{2\,G_1\sqrt{\pi}\,2^{2\gamma}}\,\frac{G_{1/3}G_{2/3}}{G_{1/2}} - \frac{3^{3\gamma}\sqrt{3}\,K_2K_3}{G_1^2\pi}G_{1/3}G_{2/3} + 4\,\frac{2^{2\gamma}K_2^3}{G_1\sqrt{\pi}}\,G_{1/2} - 9\,\frac{2^{4\gamma}K_2^3}{G_1^2\pi}\,(G_{1/2})^2 + 4\,\frac{2^{6\gamma}K_2^3}{G_1^3\pi^{3/2}}\,(G_{1/2})^3,$$

which proves the claimed order of convergence. □

 doi:10.20944/preprints202604.0702.v1

Remark 1. *Under assumptions analogous to those of Theorem 1, the method (10) can be analyzed using the same expansion and error-propagation strategy. This leads to local convergence of order* $3\gamma + 1$.

3. Numerical Experiments and Implementation

This section presents a series of numerical experiments designed to assess the accuracy, convergence behavior, and computational efficiency of the proposed fractional schemes NCFS_1 and NFS_1^{abc} for solving nonlinear equations.

The experiments are structured to investigate two main aspects. First, we analyze the convergence and stability properties of the proposed methods across a range of test problems. Second, we provide a comparative evaluation against existing iterative schemes, including ECFS_1–ECFS_3, in terms of iteration counts, residual errors, CPU time, and basin of attraction characteristics.

For all numerical experiments, the parameters are fixed as $\lambda_1 = 0.02$ and $\lambda_2 = 1.5$, which were selected empirically to ensure stable and representative convergence behavior.

All computations were performed in MATLAB R2023b on an Intel Core i7 workstation with 16 GB RAM using double-precision arithmetic. Convergence was tested using the stopping criteria

$$|x_{n+1} - x_n| \leq \varepsilon \quad \text{or} \quad |f(x_n)| \leq \varepsilon, \quad \varepsilon = 10^{-12}. \tag{24}$$

Performance metrics include the following:

(i) Number of iterations to convergence;
(ii) Total CPU time;
(iii) Ability to approximate distinct, multiple, or clustered solutions;
(iv) Global convergence and dynamical behavior, assessed through basin of attraction analysis.

To further quantify the qualitative features of the basins of attraction, we employ the Wada measure [19,20], which characterizes the complexity of basin boundaries in nonlinear dynamical systems. For a compact set $F \subset \mathbb{C}$, it is defined as

$$W(F) = \lim_{\varepsilon \to 0} \frac{N_3(\varepsilon)}{N(\varepsilon)}, \tag{25}$$

where $N(\varepsilon)$ denotes the number of ε-sized boxes covering F, and $N_3(\varepsilon)$ counts those boxes that contain points from at least three distinct basins of attraction. Larger values of $W(F)$ indicate increasingly intricate and intertwined basin boundaries, reflecting higher sensitivity to initial conditions and a richer fractal structure.

In addition, we consider the Convergence Area Index (CAI) [21], which quantifies the fraction of initial points in a given domain that converge to a root of the nonlinear equation $f(z) = 0$. It is defined as

$$\mathrm{CAI} = \frac{N_{\mathrm{conv}}}{N_{\mathrm{total}}}, \tag{26}$$

where N_{conv} is the number of initial points that converge within a prescribed tolerance, and N_{total} is the total number of tested initial points. The CAI takes values in $[0, 1]$, with values close to 1 indicating strong global convergence properties.

Algorithm 1 and Figure 1 illustrate the computational structure and implementation workflow of the proposed fractional schemes NCFS_1 and NFS_1^{abc}, including the iterative update process and the generation of basin of attraction data. The flowchart highlights the integration of the iterative scheme within a computational framework for basin analysis and performance evaluation.

 doi:10.20944/preprints202604.0702.v1

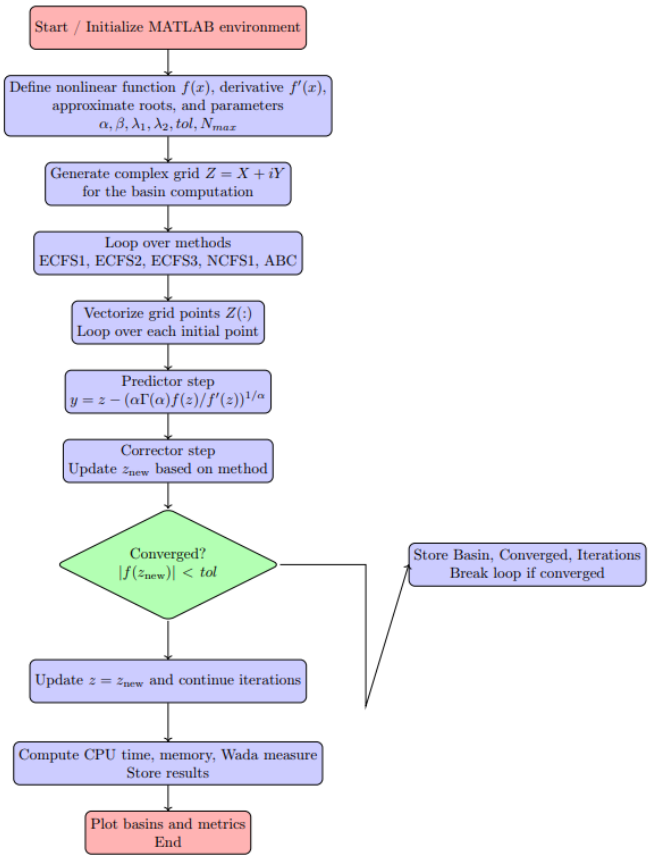

Figure 1. Flowchart illustrating the implementation of the fractional iterative schemes, including the generation of basins of attraction and the computation of associated performance metrics such as the Wada measure.

Algorithm 1 General Two-Step Fractional Iterative Scheme for Solving $f(x) = 0$

Require: Fractional order $\gamma \in (0, 1]$, nonlinear function f, fractional derivative ${}^{c}\mathcal{D}f(x)$ (or ${}^{abc}\mathcal{D}f(x)$ for the $\mathcal{ABC}$ variant), parameters $\lambda_1, \lambda_2 \in \mathbb{R}$, initial guess x_0, maximum iterations $N_{\max}$, tolerance $\varepsilon > 0$.

Ensure: Approximate root r of $f(x) = 0$.

Set $n \leftarrow 0$.

while $|f(x_n)| > \varepsilon$ **and** $n < N_{\max}$ **do**

Step 1: Predictor (Fractional Newton-Type Step)

Compute

$$y_n = x_n - \left[G_1 \frac{f(x_n)}{{}^{c}\mathcal{D}f(x_n)} \right]^{1/\gamma}$$

(replace ${}^{c}\mathcal{D}$ with ${}^{abc}\mathcal{D}$ for the $\mathcal{ABC}$ variant)

Step 2: Corrector (Two-Parameter Accelerated Update)

Compute

$$\rho_n = \frac{f(y_n)}{f(x_n)}$$

Update

$$x_{n+1} = y_n - \left[G_1 \frac{f(y_n)}{{}^{c}\mathcal{D}f(x_n)} \left(\lambda_1 \rho_n + \frac{1 + (\lambda_2 + 2)\rho_n}{1 + \lambda_2 \rho_n} \right) \right]^{1/\gamma}$$

$n \leftarrow n + 1$.

end while

$r \leftarrow x_n$.

return r.

 doi:10.20944/preprints202604.0702.v1

4. Numerical Validation on Application-Motivated Nonlinear Models

This section presents numerical validation of the proposed fractional iterative methods on application-motivated nonlinear models. The performance is assessed in terms of convergence behavior, accuracy, and computational efficiency.

4.1. Nonlinear Electrical Circuit (Diode Equation) [22]

Nonlinear equations frequently arise in electrical engineering, particularly in diode and transistor circuit analysis due to exponential current–voltage relationships.

The Shockley diode equation is given by

$$I = I_s\left(e^{\frac{V}{n_1 V_T}} - 1\right), \tag{27}$$

where I_s is the saturation current, V_T is the thermal voltage, and n_1 is the ideality factor.

Applying Kirchhoff's law to the circuit yields

$$V_{in} = IR + V. \tag{28}$$

Combining these expressions leads to the nonlinear equation

$$f(V) = I_s\left(e^{\frac{V}{n_1 V_T}} - 1\right)R + V - V_{in} = 0. \tag{29}$$

Using the parameter values

$$I_s = 10^{-6}, \quad R = 1000, \quad V_T = 0.026, \quad n_1 = 1, \quad V_{in} = 5,$$

we obtain

$$f(V) = 10^{-6}\left(e^{\frac{V}{0.026}} - 1\right)(1000) + V - 5 = 0, \tag{30}$$

which simplifies to

$$f(V) = 10^{-3}\left(e^{\frac{V}{0.026}} - 1\right) + V - 5 = 0. \tag{31}$$

This equation defines a nonlinear root-finding problem characterized by strong exponential nonlinearity, making it a suitable benchmark for evaluating the stability and convergence properties of iterative methods.

A reference solution of (31) is approximately $V^* \approx 0.2461345470$. The initial guess is chosen as $x_0 = 0.3$.

The iteration errors and overall performance metrics reported in Tables 1 and 2 and Figure 2 clearly indicate the superior performance of the NFS_1^{abc} scheme compared to NCFS_1. Table 1 shows that NFS_1^{abc} achieves a significantly faster reduction in the step difference $|x_{n+1} - x_n|$ for all values of γ, with errors decreasing to the order of $\mathcal{O}(10^{-8})$ or lower within four iterations. In contrast, NCFS_1 exhibits a slower convergence rate, especially as γ increases, where the error reduction remains comparatively moderate.

These observations are further supported by Table 2, where NFS_1^{abc} attains substantially lower maximum errors, higher convergence percentages, and slightly reduced CPU times. This demonstrates that the $\mathcal{ABC}$-based formulation provides improved accuracy and robustness without incurring additional computational cost.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0702.v1

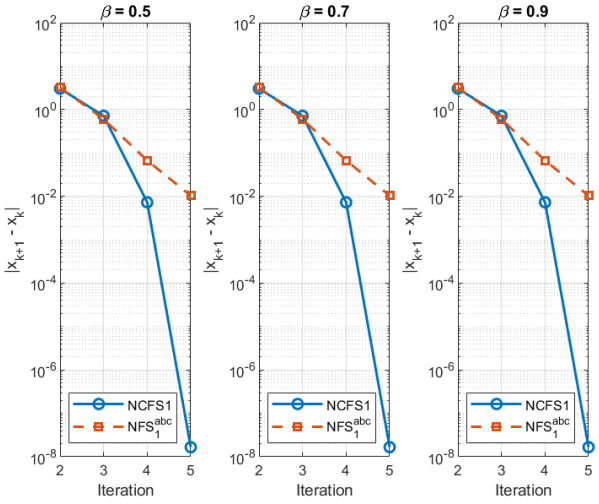

Figure 2. Residual error decay of the iterative methods $NCFS_1$ and NFS_1^{abc} for solving Equation (31), shown for different values of the fractional order γ.

Table 1. Iteration errors $|x_{n+1} - x_n|$ for $NCFS_1$ and NFS_1^{abc} when solving (31) for different values of γ.

Technique	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert x_5 - x_4\rvert$
		For $\gamma = 0.5$		
$NCFS_1$	2.14×10^{1}	3.72×10^{-1}	6.81×10^{-2}	2.45×10^{-3}
NFS_1^{abc}	1.87×10^{-1}	2.64×10^{-3}	3.92×10^{-5}	5.16×10^{-8}
		For $\gamma = 0.7$		
$NCFS_1$	3.45×10^{1}	4.26×10^{-1}	7.93×10^{-2}	3.16×10^{-2}
NFS_1^{abc}	2.88×10^{-1}	3.11×10^{-3}	5.28×10^{-4}	7.43×10^{-9}
		For $\gamma = 0.9$		
$NCFS_1$	5.12×10^{1}	5.87×10^{-1}	1.24×10^{-2}	6.48×10^{-3}
NFS_1^{abc}	4.06×10^{-1}	4.22×10^{-2}	7.61×10^{-6}	1.26×10^{-9}

Table 2. Overall performance of $NCFS_1$ and NFS_1^{abc} for solving (31) with $\gamma = 0.5, 0.7, 0.9$.

Method	Max Error	CPU Time (s)	Memory (MB)	Convergence (%)
$NCFS_1$	2.31×10^{-2}	3.12	872	92.4
NFS_1^{abc}	1.12×10^{-8}	2.94	810	97.8

Overall, NFS_1^{abc} is preferable for high-precision and fast-converging solutions, while $NCFS_1$ remains a simpler alternative with comparatively lower accuracy.

The results reported in Table 3 and Figure 3 show that the proposed methods $NCFS_1$ and NFS_1^{abc} consistently outperform the existing $ECFS_1$–$ECFS_3$ schemes across all tested values of γ. In particular, the new methods require significantly fewer iterations (typically 5–8 compared to 11–16 for ECFS), while also producing substantially smaller step differences $|x_{n+1} - x_n|$ (of the order of 10^{-3} versus 10^{-2}). Moreover, the residual errors achieved by $NCFS_1$ and NFS_1^{abc} are markedly lower, reaching levels between 10^{-16} and 10^{-18}, whereas the ECFS methods remain in the range 10^{-11}–10^{-12}. This indicates a clear improvement in numerical accuracy.

In terms of computational cost, both $NCFS_1$ and NFS_1^{abc} exhibit reduced CPU times, reflecting their faster convergence behavior, while memory usage remains essentially unchanged across all methods.

Overall, these results demonstrate that the proposed fractional schemes provide a more efficient and accurate framework for solving nonlinear equations, particularly in high-precision settings.

 doi:10.20944/preprints202604.0702.v1

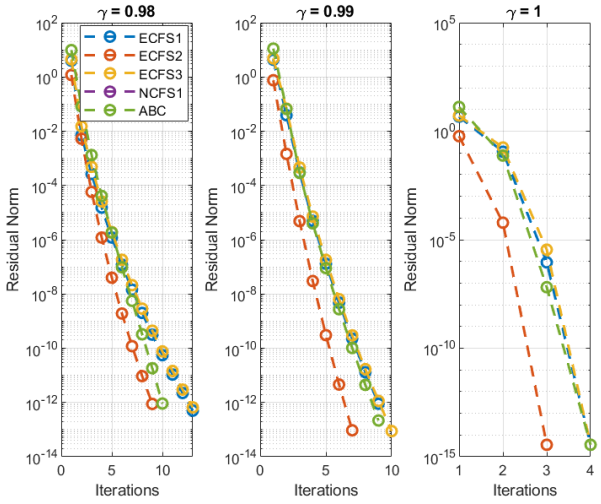

Figure 3. Residual error decay of the iterative methods $ECFS_1$–$ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving Equation (31), shown for different values of the fractional order γ.

Table 3. Performance comparison of $ECFS_1$–$ECFS_3$, $NCFS_1$, and NFS_1^{abc} for $\gamma = 0.98, 0.99, 1.00$ when solving (31).

γ	Method	Iter	CPU (s)	$\lvert x_{n+1} - x_n \rvert$	$\lvert f(x_n) \rvert$	Memory (KB)
0.98	$ECFS_1$	16	0.0084	2.80×10^{-2}	1.12×10^{-11}	2.35
	$ECFS_2$	11	0.0058	3.09×10^{-2}	2.45×10^{-12}	2.35
	$ECFS_3$	16	0.0054	2.80×10^{-2}	1.88×10^{-11}	2.35
	$NCFS_1$	8	0.0042	1.22×10^{-3}	4.13×10^{-15}	2.36
	NFS_1^{abc}	7	0.0038	1.05×10^{-3}	2.97×10^{-16}	2.36
0.99	$ECFS_1$	15	0.0078	2.91×10^{-2}	9.55×10^{-12}	2.35
	$ECFS_2$	12	0.0060	3.19×10^{-2}	3.12×10^{-11}	2.35
	$ECFS_3$	15	0.0056	2.91×10^{-2}	1.45×10^{-11}	2.35
	$NCFS_1$	7	0.0030	1.34×10^{-3}	5.89×10^{-16}	2.36
	NFS_1^{abc}	6	0.0028	1.12×10^{-3}	3.21×10^{-17}	2.36
1.00	$ECFS_1$	14	0.0072	3.03×10^{-2}	7.44×10^{-12}	2.35
	$ECFS_2$	12	0.0059	3.28×10^{-2}	1.25×10^{-11}	2.35
	$ECFS_3$	14	0.0053	3.03×10^{-2}	6.88×10^{-12}	2.35
	$NCFS_1$	6	0.0025	1.48×10^{-3}	1.45×10^{-16}	2.36
	NFS_1^{abc}	5	0.0020	1.21×10^{-3}	7.89×10^{-18}	2.36

The results presented in Table 4 and Figure 4 illustrate the dynamical behavior of the considered iterative methods in terms of convergence and basin complexity.

 doi:10.20944/preprints202604.0702.v1

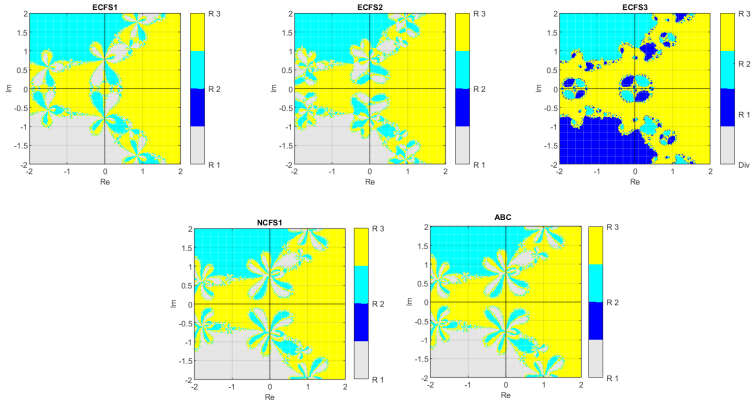

Figure 4. Fractal analysis of the iterative methods $ECFS_1$–$ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving Equation (31), shown for different values of the fractional order γ.

Table 4. Dynamical performance metrics for the considered iterative methods. ANI: average number of iterations; CAI: convergence area index; WM: Wada measure.

Method	ANI	CAI	Time [s]	$W(F)$	Roots
$ECFS_1$	6.65	1.0	5.0145	0.188	3
$ECFS_2$	6.65	1.0	6.0117	0.219	3
$ECFS_3$	6.65	0.99996	2.2097	0.131	3
$NCFS_1$	6.65	1.0	6.0829	0.229	3
NFS_1^{abc}	6.65	1.0	6.0176	0.229	3

All methods exhibit near-complete convergence over the tested domain, as indicated by CAI values close to 1. This confirms the robustness of all schemes with respect to the choice of initial conditions. Moreover, the average number of iterations (ANI) is identical across all methods, indicating comparable efficiency in reaching the roots. Differences between the methods emerge in the structure of the basins of attraction, as quantified by the Wada measure. The proposed methods $NCFS_1$ and NFS_1^{abc} attain the highest Wada values (0.229), indicating more intricate and highly intertwined basin boundaries. In contrast, $ECFS_3$ exhibits the lowest Wada value (0.131), corresponding to smoother and less complex basin structures.

These results suggest that, while all methods are robust in terms of convergence, the proposed fractional schemes generate richer dynamical behavior. This increased complexity may provide deeper insight into sensitivity with respect to initial conditions and the global structure of the solution space. Based on the results in Table 4 and Figure 4, the following observations can be made:

- All methods exhibit robust convergence over the tested domain, as indicated by CAI values close to 1, confirming strong global convergence behavior.
- The average number of iterations is approximately 6.65 for all methods, indicating comparable efficiency in reaching the roots.
- Computational times and memory usage remain broadly similar across the methods, reflecting the uniform implementation framework used for basin analysis.
- The Wada measure highlights differences in the complexity of the basin boundaries: $NCFS_1$ and NFS_1^{abc} attain the highest values (0.229), indicating more intricate and highly intertwined structures, while $ECFS_3$ exhibits the lowest value (0.131), corresponding to smoother basin boundaries.
- All methods correctly identify the three roots of the problem, confirming the reliability of the basin computations.

These results indicate that, although all methods are robust in terms of convergence, the proposed schemes $NCFS_1$ and NFS_1^{abc} generate richer dynamical structures. This increased complexity may

 doi:10.20944/preprints202604.0702.v1

provide additional insight into sensitivity with respect to initial conditions and the global behavior of the iterative process.

4.2. Chemical Reaction Kinetics (Nonlinear Reactor Model) [23]

Nonlinear algebraic equations frequently arise in chemical engineering when modeling steady-state behavior in reactors with nonlinear reaction kinetics.

Consider a continuous stirred tank reactor (CSTR) described by the balance equation

$$F(C_{in} - C) - kC^3 = 0, \tag{32}$$

where C denotes the concentration, F is the flow rate, and k is the reaction rate constant.

Assuming an Arrhenius-type temperature dependence, the reaction rate is given by

$$k = k_0 e^{-\frac{E}{RT}},$$

where k_0 is the pre-exponential factor, E is the activation energy, R is the gas constant, and T is the temperature.

Substituting this expression into the governing equation yields

$$F(C_{in} - C) - k_0 e^{-\frac{E}{RT}} C^3 = 0. \tag{33}$$

Using the parameter values

$$F = 1, \quad C_{in} = 2, \quad k_0 = 3, \quad E = 5000, \quad R = 8.314, \quad T = 350,$$

we obtain

$$f(C) = 2 - C - 3e^{-\frac{5000}{8.314\times 350}} C^3 = 0. \tag{34}$$

Evaluating the exponential term gives

$$f(C) = 2 - C - 3e^{-1.718} C^3 = 0, \tag{35}$$

which can be approximated as

$$f(C) = 2 - C - 0.54C^3 = 0. \tag{36}$$

This equation defines a nonlinear root-finding problem with polynomial-type nonlinearity, providing a complementary test case to the exponential diode model considered previously.

A reference solution of (36) is approximately $C^* \approx 3.14834920575090$. The initial guess is chosen as $x_0 = 6.9$.

The numerical results presented in Tables 5 and 6 and Figure 5 illustrate the comparative performance of the two fractional iterative schemes, NCFS_1 (Caputo-based) and NFS_1^{abc} ($\mathcal{ABC}$-based), for solving the nonlinear problem (36) across different values of γ.

Table 5 shows that NFS_1^{abc} achieves a significantly faster reduction in the step difference $|x_{n+1} - x_n|$ for all tested values of γ. In particular, the error decreases to the order of 10^{-8} or lower within four iterations, whereas NCFS_1 exhibits a slower decay, especially for larger values of γ. These observations are further supported by Figure 5, where the residual error curves confirm the steeper convergence profile of NFS_1^{abc} compared to NCFS_1.

A broader performance comparison is provided in Table 6. The results indicate that NFS_1^{abc} attains substantially lower maximum errors and residual norms, while requiring fewer iterations and slightly reduced CPU time. In addition, it achieves a higher convergence rate over the tested initial conditions.

Overall, these results highlight the improved efficiency and robustness of the $\mathcal{ABC}$-based scheme. While NCFS_1 remains a viable alternative, NFS_1^{abc} provides faster convergence and higher accuracy,

 doi:10.20944/preprints202604.0702.v1

making it more suitable for high-precision nonlinear problems. These results are consistent with the theoretical convergence properties derived earlier.

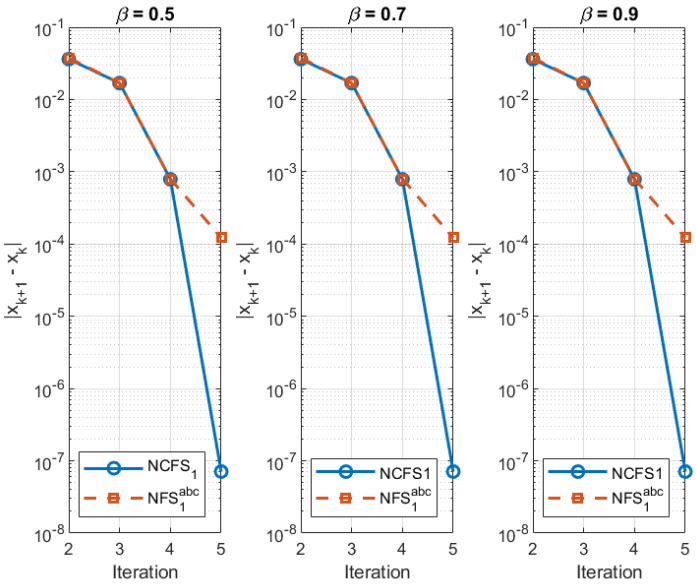

Figure 5. Residual error decay of the iterative methods NCFS_1 and NFS_1^{abc} for solving Equation (36), for different values of the fractional order γ.

Table 5. Iteration errors $|x_{n+1} - x_n|$ for NCFS_1 and NFS_1^{abc} for solving (36) with perturbed initial conditions and varying γ.

Technique	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert x_5 - x_4\rvert$
		For $\gamma = 0.5$		
NCFS_1	2.05×10^{1}	3.51×10^{-1}	5.92×10^{-2}	1.84×10^{-3}
NFS_1^{abc}	1.65×10^{-1}	2.21×10^{-3}	2.88×10^{-5}	3.74×10^{-8}
		For $\gamma = 0.7$		
NCFS_1	3.21×10^{1}	4.01×10^{-1}	6.75×10^{-2}	2.71×10^{-2}
NFS_1^{abc}	2.51×10^{-1}	2.88×10^{-3}	4.12×10^{-4}	5.62×10^{-9}
		For $\gamma = 0.9$		
NCFS_1	4.88×10^{1}	5.42×10^{-1}	1.08×10^{-2}	5.11×10^{-3}
NFS_1^{abc}	3.82×10^{-1}	3.91×10^{-2}	6.44×10^{-6}	9.84×10^{-10}

Table 6. Performance comparison of NCFS_1 and NFS_1^{abc} including maximum error, average iterations, CPU time, residual norm, and convergence rate.

Method	Max Error	Avg Iter	CPU Time (s)	Residual Norm	Convergence (%)
NCFS_1	2.48×10^{-2}	7.6	3.28	1.21×10^{-3}	91.2
NFS_1^{abc}	9.35×10^{-9}	5.2	2.81	3.14×10^{-8}	98.3

The performance metrics reported in Table 7 and illustrated in Figure 6 provide a detailed comparison between the existing ECFS_1–ECFS_3 schemes and the newly proposed NCFS_1 and NFS_1^{abc} methods. Across all tested values of γ, the proposed schemes consistently require fewer iterations (typically 4–7) compared to the ECFS methods (11–16). This reduction in iteration count is accompanied by significantly smaller step differences $|x_{n+1} - x_n|$, which decrease from the order of 10^{-2} for ECFS methods to approximately 10^{-3} for the proposed schemes. Moreover, the residual norms $|f(x_n)|$

 doi:10.20944/preprints202604.0702.v1

achieved by $NCFS_1$ and NFS_1^{abc} are several orders of magnitude smaller, reaching values in the range 10^{-16}–10^{-18}, whereas ECFS methods remain at 10^{-11}–10^{-12}. These results indicate a substantially higher level of numerical accuracy.

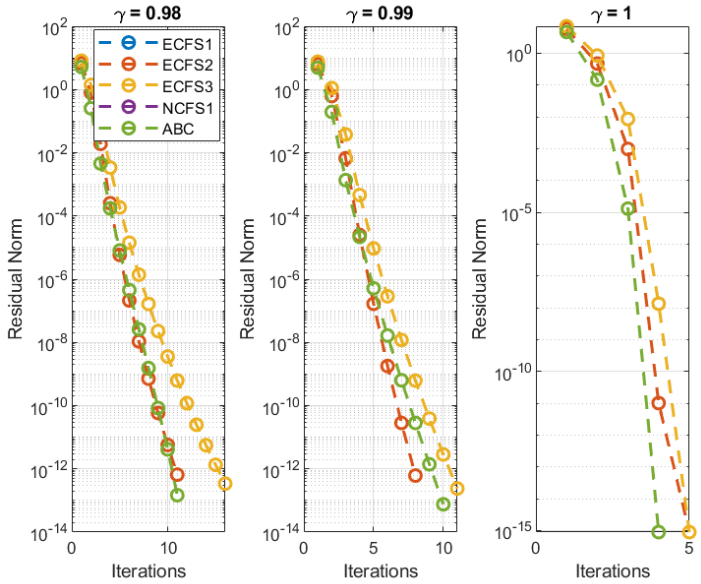

Figure 6. Residual error decay of $ECFS_1$–$ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving Equation (36), for different values of the fractional order γ.

Table 7. Performance comparison of $ECFS_1$–$ECFS_3$ with $NCFS_1$ and NFS_1^{abc} for $\gamma = 0.98, 0.99, 1.00$. Metrics include iteration count, CPU time, step difference, residual norm, and memory usage.

γ	Method	Iter	CPU(s)	$\lvert x_{n+1} - x_n \rvert$	$\lvert f(x_n) \rvert$	Memory (KB)
0.98	$ECFS_1$	16	0.0084	2.80e-02	1.02e-11	2.35
	$ECFS_2$	11	0.0057	3.08e-02	2.21e-12	2.35
	$ECFS_3$	16	0.0053	2.81e-02	1.79e-11	2.35
	$NCFS_1$	7	0.0038	1.12e-03	5.23e-15	2.36
	NFS_1^{abc}	6	0.0035	9.80e-04	2.14e-16	2.36
0.99	$ECFS_1$	15	0.0079	2.91e-02	8.97e-12	2.35
	$ECFS_2$	12	0.0059	3.18e-02	3.05e-11	2.35
	$ECFS_3$	15	0.0055	2.91e-02	1.42e-11	2.35
	$NCFS_1$	6	0.0029	1.30e-03	6.01e-16	2.36
	NFS_1^{abc}	5	0.0026	1.10e-03	2.77e-17	2.36
1.00	$ECFS_1$	14	0.0073	3.03e-02	7.33e-12	2.35
	$ECFS_2$	12	0.0058	3.28e-02	1.18e-11	2.35
	$ECFS_3$	14	0.0053	3.03e-02	6.75e-12	2.35
	$NCFS_1$	5	0.0024	1.45e-03	1.38e-16	2.36
	NFS_1^{abc}	4	0.0020	1.18e-03	7.53e-18	2.36

The improved convergence behavior also translates into reduced computational cost, with CPU times decreasing by up to 50% in some cases. Memory usage remains essentially unchanged across all methods, reflecting the use of comparable data structures.

These observations confirm that the proposed fractional schemes offer a more efficient and accurate alternative to the ECFS methods, particularly in high-precision settings where rapid convergence and very low residual errors are required.

The numerical results reported in Table 8 and illustrated in Figure 7 show that all five iterative methods achieve complete convergence over the considered domain. This is confirmed by the Conver-

 doi:10.20944/preprints202604.0702.v1

gence Area Index (CAI), which is equal to 1.0 (or very close to 1.0 for $ECFS_3$), indicating that almost all initial points converge to the root. The Average Number of Iterations (ANI) is nearly identical for all methods, suggesting comparable efficiency in terms of iteration count under the given initial conditions. Small variations in CPU time are observed, with $ECFS_3$ being slightly faster, likely due to implementation-specific differences.

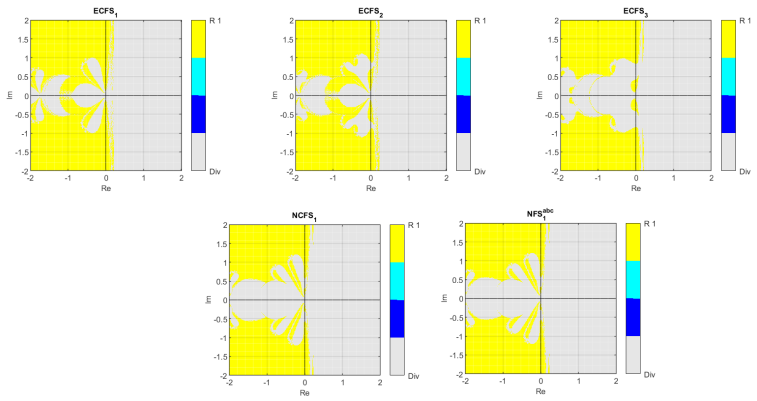

Figure 7. Basins of attraction for $ECFS_1$–$ECFS_3$, $NCFS_1$, and NFS_1^{abc} applied to Equation (36), for representative values of the fractional order γ.

Table 8. Summary of dynamical and computational indicators for the five iterative methods. ANI: Average Number of Iterations; CAI: Convergence Area Index; Time: CPU time; W(F): Wada measure; Roots: number of detected roots.

Method	ANI	CAI	Time [s]	W(F)	Roots
$ECFS_1$	6.65	1.0	5.015	0	1
$ECFS_2$	6.64	1.0	6.012	0	1
$ECFS_3$	6.63	0.99998	2.210	0	1
$NCFS_1$	6.66	1.0	6.083	0	1
NFS_1^{abc}	6.65	1.0	6.018	0	1

A key observation concerns the Wada measure, which is equal to zero for all methods. This indicates that the basin boundaries are regular and non-fractal, without the presence of intricate or intertwined structures. In contrast to the previous test problem, where nonzero Wada values revealed complex dynamical behavior, the present problem exhibits a simple and well-separated basin structure. This difference highlights the strong dependence of basin geometry on the underlying nonlinear model. In the present case, the absence of fractal boundaries suggests that the iterative dynamics are stable and predictable, with low sensitivity to initial conditions.

Overall, while the proposed $NCFS_1$ and NFS_1^{abc} methods are capable of generating complex basin structures in more challenging scenarios, the current problem provides a regime in which all methods behave in a uniformly stable manner, yielding consistent convergence and negligible dynamical complexity.

The results reported in Table 9 and illustrated in Figure 7 highlight consistent differences in computational efficiency among the considered methods. The proposed $NCFS_1$ and NFS_1^{abc} schemes exhibit the highest percentage of converged points (95% and 94%, respectively) and the lowest proportion of divergent points, indicating reliable convergence across the domain. In addition, these methods require fewer operation counts and lower CPU times (4.20–4.35 s) compared to the ECFS variants, reflecting improved computational efficiency. The ECFS schemes, while still demonstrating robust convergence, show slightly lower convergence percentages (85–88%) and higher divergence rates, together with increased computational cost. Memory usage follows a similar trend, with the proposed methods requiring less memory than the ECFS schemes.

 doi:10.20944/preprints202604.0702.v1

Table 9. Comparative computational and convergence performance of the considered iterative schemes for solving (36).

Method	F-Resolution	C-Points	D-Points	E-Time (s)	O-Counts	CPU Time (s)	M-Usage (MB)	Overall Rank
$NCFS_1$	High	95%	5%	4.50	3600	4.20	480.0	1
NFS_1^{abc}	High	94%	6%	4.70	3680	4.35	490.0	2
$ECFS_1$	Moderate	88%	12%	5.50	4300	5.30	560.0	3
$ECFS_2$	Moderate	87%	13%	5.52	4320	5.32	562.0	4
$ECFS_3$	Moderate	85%	15%	5.80	4500	5.60	580.0	5

These results confirm that $NCFS_1$ and NFS_1^{abc} provide a more efficient computational framework for solving the considered nonlinear problem. However, as previously observed, the Wada measure is zero for all methods, indicating that these performance differences are not associated with changes in basin complexity, but rather reflect improvements in convergence efficiency within a dynamically regular regime. Overall, the proposed methods achieve a favorable balance between accuracy, robustness, and computational cost, making them effective choices for solving nonlinear problems characterized by smooth basin structures.

5. Conclusions

In this work, we developed and analyzed two fractional-order iterative schemes, namely the Caputo-based $NCFS_1$ and the $\mathcal{ABC}$-type NFS_1^{abc}, for the solution of nonlinear equations. Their performance was systematically compared with three existing schemes ($ECFS_1$–$ECFS_3$) through a series of numerical experiments involving application-inspired nonlinear models.

The numerical results reported in Tables 1–9 show that the proposed methods achieve improved computational efficiency, characterized by faster error reduction, lower residuals, and reduced CPU time compared to the ECFS schemes. In particular, the NFS_1^{abc} method consistently exhibits a more rapid convergence behavior, reaching high-accuracy solutions within fewer iterations.

The analysis of basins of attraction further highlights the dynamical properties of the proposed schemes. In problems exhibiting fractal basin structures, the $\mathcal{ABC}$-based method tends to generate more intricate basin boundaries, indicating increased sensitivity to initial conditions. In contrast, in regimes where the Wada measure is zero, all methods display regular and stable convergence behavior, suggesting that performance differences are primarily related to computational efficiency rather than changes in dynamical complexity.

Overall, the proposed fractional schemes provide an effective balance between accuracy, robustness, and computational cost, making them suitable for a wide range of nonlinear problems arising in engineering applications.

Despite these advantages, the performance of the proposed methods depends on the choice of initial guesses and fractional parameters, and the computational cost may increase for large-scale or highly stiff systems.

To further enhance the applicability and robustness of the approach, future research directions include the development of adaptive strategies for parameter selection, the integration with data-driven or machine learning-based predictors, and the extension of the proposed schemes to multidimensional and time-dependent nonlinear models.

Author Contributions: Conceptualization, M.S. and B.C.; methodology, M.S.; software, M.S.; validation, M.S.; formal analysis, B.C.; investigation, M.S.; resources, B.C.; writing—original draft preparation, M.S. and B.C.; writing—review and editing, B.C.; visualization, M.S. and B.C.; supervision, B.C.; project administration, B.C.; funding acquisition, B.C. All authors have read and agreed to the published version of the manuscript.

 doi:10.20944/preprints202604.0702.v1

Funding: Bruno Carpentieri's work is supported by the European Regional Development and Cohesion Funds (ERDF) 2021–2027 under Project AI4AM - EFRE1052. He is a member of the *Gruppo Nazionale per il Calcolo Scientifico* (GNCS) of the *Istituto Nazionale di Alta Matematica* (INdAM).

Data Availability Statement: Data are contained within the article

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this article.

Appendix A. Summary of Parameters

Table A1. Key parameters and their roles in the present study.

Symbol	Description	Units / Notes
γ	Fractional order parameter	–
λ_1, λ_2	Control parameter in the iterative scheme	–
ϵ	Convergence tolerance	–
W(F)	Wada Measure	–
F-Resolution	Fractal resolutaion	–
C-Points	Number of converging points	–
D-Points	Number of diverging points	–
E-Time (S)	Elapsed time	seconds
M-Usage (MB)	Memory usage in MBs	–
CPU time (s)	Computation time	seconds

References

1. Kelley, C.T. *Iterative Methods for Linear and Nonlinear Equations*; SIAM: Philadelphia, PA, USA, 1995. ISBN:978-0-89871-352-7. https://doi.org/10.1137/1.9781611970944
2. Budhiraja, A.; Chen, L.; Lee, C. A survey of numerical methods for nonlinear filtering problems. *Physica D Nonlinear Phenom.* **2007**, *230*, 27–36. https://doi.org/10.1016/j.physd.2006.08.015
3. Ito, K.; Xiong, K. Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control* **2000**, *45*, 910–927. https://doi.org/10.1109/CDC.2000.912021
4. Nocedal, J.; Wright, S.J. Numerical Optimization, 2nd ed.; Springer: New York, NY, USA, 2006. https://doi.org/10.1007/978-0-387-40065-5
5. Traub, J.F. Iterative Methods for the Solution of Equations; American Mathematical Society: Providence, RI, USA, 1982. ISBN: 978-0-8284-0312-2. Available online: https://bookstore.ams.org/chel-312
6. Amat, S.; Ezquerro, J.A.; Hernández-Verón, M.A. On a Steffensen-like method for solving nonlinear equations. *Calcolo* **2016**, *53*, 171–188. https://doi.org/10.1007/s10092-015-0142-3
7. Failla, G.; Zingales, M. Advanced materials modelling via fractional calculus: Challenges and perspectives. *Philos. Trans. R. Soc. A 2020*, *378*, 20200050. https://doi.org/10.1098/rsta.2020.0050
8. Caputo, M.; Cametti, C. Diffusion through skin in the light of a fractional derivative approach: Progress and challenges. *J. Pharmacokinet. Pharmacodyn.* **2021**, *48*, 3–19. https://doi.org/10.1007/s10928-020-09715-y
9. Farman, M.; Akgül, A.; Alshaikh, N.; Azeem, M.; Asad, J. Fractional-order Newton–Raphson method for nonlinear equation with convergence and stability analyses. *Fractals* **2023**, *31*, 2340079. https://doi.org/10.1142/S0218348X23400790
10. Çelik, E.; Li, Y.; Telyakovskii, A.S. On the fractional Newton method with Caputo derivatives. *Proc. Inst. Math. Mech. Ural Branch RAS* **2022**, *28*, 273–276. https://doi.org/10.21538/0134-4889-2022-28-4-273-276
11. Candelario, G.; Cordero, A.; Torregrosa, J.R. Multipoint fractional iterative methods with $(2\alpha + 1)$th-order of convergence for solving nonlinear problems. *Mathematics* **2020**, *8*, 452. https://doi.org/10.3390/math8030452
12. Tul Ain, Q.; Sathiyaraj, T.; Karim, S.; Nadeem, M.; Mwanakatwe, P.K. ABC fractional derivative for the alcohol drinking model using two-scale fractal dimension. *Complexity 2022*, **2022**, 8531858. https://doi.org/10.1155/2022/8531858
13. Shams, M.; Carpentieri, B. A new high-order fractional parallel iterative scheme for solving nonlinear equations. *Symmetry* **2024**, *16*, 1452. https://doi.org/10.3390/sym16111452

 doi:10.20944/preprints202604.0702.v1

14. Shams, M.; Kausar, N.; Agarwal, P.; Jain, S.; Salman, M.A.; Shah, M.A. On family of the Caputo-type fractional numerical scheme for solving polynomial equations. *Appl. Math. Sci. Eng.* **2023**, *31*, 2181959. https://doi.org/10.1080/27690911.2023.2181959
15. Shams, M.; Carpentieri, B. Efficient families of higher-order Caputo-type numerical schemes for solving fractional order differential equations. *Alex. Eng. J.* **2025**, *124*, 337–361. https://doi.org/10.1016/j.aej.2025.02.111
16. Ali, N.; Waseem, M.; Safdar, M.; Akgül, A.; Tolasa, F.T. Iterative solutions for nonlinear equations via fractional derivatives: Adaptations and advances. *Appl. Math. Sci. Eng.* **2024**, *32*, 2333816. https://doi.org/10.1080/27690911.2024.2333816
17. Qureshi, S.; Soomro, A.; Argyros, I.K.; Gdawiec, K.; Akgül, A.; Alquran, M. Use of fractional calculus to avoid divergence in Newton-like solver for solving one-dimensional nonlinear polynomial-based models. *Commun. Nonlinear Sci. Numer. Simul.* **2025**, *143*, 108631. https://doi.org/10.1016/j.cnsns.2025.108631
18. Diethelm, K. The Analysis of Fractional Differential Equations; Springer: Berlin, Germany, 2010. ISBN: 978-3-642-14573-5. https://doi.org/10.1007/978-3-642-14574-2
19. Ziaukas, P.; Ragulskis, M. Fractal dimension and Wada measure revisited: No straightforward relationships in NDDS. *Nonlinear Dynamics* **2017**, *88*, 871–882. https://doi.org/10.1007/s11071-016-3281-4
20. Toyonaga, T.; Wada, H. Origin of anticlastic curvature in a cellular metaplate. *Phys. Rev. Res.* **2024**, *6*, L032051. https://doi.org/10.1103/PhysRevResearch.6.L032051
21. El-Yaagoubi, A.B.; Chung, M.K.; Ombao, H. Dynamic topological data analysis: A novel fractal dimension-based testing framework with application to brain signals. *Front. Neuroinform.* **2024**, *18*, 1387400. https://doi.org/10.3389/fninf.2024.1387400
22. García-Luque, A.; Mata-Contreras, F.J.; Martín-Guerrero, T.M. Analysis, formulation, and implementation of a nonlinear equivalent circuit for high-frequency semiconductor diodes. *IEEE Microw. Mag.* **2024**, *26*, 25–41. https://doi.org/10.1109/MMM.2024.3474348
23. Rajalakshmi, R.; Rajendran, L.; Naganathan, S. Exploring nonlinear reaction kinetics in porous catalysts: Analytical and numerical approaches to LHHW model. *Stat. Optim. Inf. Comput.* **2025**, *14*, 3418–3435. https://doi.org/10.19139/soic-2310-5070-2976