fractal and fractional

MDPI

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; mudassir.shams@balikesir.edu.tr
[2] Department of Mathematics and Statistics, Riphah International University, Islamabad 44000, 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, where efficient and reliable iterative solvers are required. This study introduces two fractional-order iterative schemes based on a common predictor–corrector structure: a Caputo-based method, $NCFS_1$, and an Atangana–Baleanu–Caputo (ABC)-based variant, NFS_1^{abc}. The proposed schemes incorporate a fractional order and two tunable parameters to improve flexibility in the iterative process. The local convergence behavior of the Caputo-based method is analyzed by means of fractional Taylor expansions, yielding an explicit error equation and convergence order, while analogous asymptotic considerations are discussed for the ABC-based variant. A dynamical-systems analysis is also performed through basins of attraction, the Convergence Area Index, and the Wada measure. Numerical experiments on application-motivated nonlinear models indicate that the proposed methods can provide faster error reduction, smaller residuals, and lower computational cost than selected existing fractional iterative schemes. These results suggest that the proposed framework is a flexible and effective approach for nonlinear root-finding problems, combining local convergence analysis with global dynamical assessment.

Keywords: fractional iterative methods; Caputo derivative; Atangana–Baleanu–Caputo derivative; nonlinear equations; Wada measure

MSC: 65H04; 65H05; 65B99; 65Y15; 65Y04

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, accurate, and robust iterative numerical methods [4].

Among classical approaches, the Newton–Raphson method [5] is one of the most widely used schemes, defined by

Check for updates

Academic Editor: Riccardo Caponetto

Received: 8 April 2026
Revised: 11 May 2026
Accepted: 15 May 2026
Published: 18 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

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

and characterized by quadratic convergence under standard assumptions. To further accelerate convergence, several higher-order extensions have been proposed, including Traub [6] and Steffensen-type methods [7]. For instance, the classical Traub scheme is given by

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

Despite their effectiveness, these methods exhibit well-known limitations. In particular, their convergence behavior may be highly sensitive to the choice of the initial guess, and their domains of attraction can be relatively restricted. Moreover, in the presence of strong nonlinearities or stiff behavior, their performance may deteriorate, leading to slow convergence or even divergence. These issues motivate the development of more flexible iterative frameworks capable of improving both robustness and convergence properties across a wider range of nonlinear problems.

In this context, fractional calculus [8] has emerged as a promising framework for enhancing iterative methods. By extending classical integer-order derivatives to non-integer orders, fractional operators introduce memory and nonlocal effects, which can influence the convergence behavior and stability of numerical schemes. Among the most widely used formulations, the Caputo fractional derivative [9], 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 convenient and widely adopted framework for generalizing classical root-finding methods.

In particular, the fractional Newton method [10–12], 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 evaluated at the root.

While this fractional extension introduces additional flexibility through the parameter γ and incorporates memory effects, its practical performance remains subject to limitations similar to those of classical schemes. In particular, convergence may still be sensitive to the choice of the initial guess, and robustness can deteriorate in the presence of strong nonlinearities or stiff behavior.

In addition to the Caputo formulation, the Atangana–Baleanu–Caputo ($\mathcal{ABC}$) fractional derivative [13] has recently attracted considerable attention due to its favorable 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 structural difference has important numerical implications. In particular, the $\mathcal{ABC}$ formulation can alleviate numerical instabilities associated with singular kernels and may reduce excessive memory effects, leading in practice to improved robustness of iterative schemes.

Remark on Fractional Operators

It is important to emphasize that the Caputo and $\mathcal{ABC}$ fractional derivatives are defined through fundamentally different kernel structures, corresponding to singular power-law and non-singular Mittag–Leffler kernels, respectively. For this reason, a direct analytical equivalence or a strict coupling of these operators is not assumed in this work.

Instead, the proposed approach adopts a common structural formulation for the iterative schemes, which is instantiated separately using each operator. This operator-agnostic construction enables a systematic comparison of their numerical behavior under the same algorithmic structure and comparable smoothness assumptions on the underlying problem. The term "unified" is therefore used in a computational sense, referring to the shared formulation of the schemes rather than to a mathematical unification of the underlying fractional operators.

Consequently, $\mathcal{ABC}$-based schemes often exhibit improved convergence behavior and, in some cases, 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 systematically exploring formulations that enable a direct comparison of their complementary properties within a common algorithmic structure. As a result, the potential of fractional iterative schemes to balance accuracy, stability, and robustness remains only partially realized.

Recent progress in fractional-order iterative methods has led to significant developments in both theoretical analysis and computational efficiency for nonlinear problems. In particular, several studies have focused on extending classical Newton-type schemes to the fractional setting, often through conformable or Caputo-based formulations. For instance, Candelario et al. [14] proposed generalized conformable fractional Newton-type methods for nonlinear systems, while Bayrak et al. [15] analyzed fractional Newton-type iterations with emphasis on convergence behavior and stability. Wang and Xu [16] further extended higher-order schemes by introducing a conformable vector Traub-type method for systems of nonlinear equations.

From a theoretical standpoint, significant effort has been devoted to improving convergence order and analyzing stability properties. Erfanifar et al. [17] constructed families of fractional iterative methods with applications to fractional differential equations, whereas Akram et al. [18] developed higher-order two-step schemes with enhanced numerical performance. In addition, Asghari et al. [19] investigated the dynamical behavior of conformable iterative methods, establishing connections between convergence properties and basin structures.

Further contributions have explored stability and applicability in practical contexts. Shams and Carpentieri [20] analyzed stability properties of fractional-order schemes in nonlinear engineering models, while Nayak and Parida [21] examined Caputo-based iterative methods with emphasis on global convergence and real-world applications. Moreover,

fractional iterative techniques have been increasingly considered in control theory [22] and dynamical systems [23], where memory effects play a fundamental role in stability and feedback mechanisms.

Overall, these studies demonstrate that fractional-order iterative schemes can enhance convergence behavior and extend modeling capabilities. However, they are typically developed within a single fractional framework and are primarily analyzed from a local or method-specific perspective, which limits a systematic understanding of how different fractional operators influence both convergence and global dynamical behavior.

Recent contributions have introduced high-order fractional iterative schemes for nonlinear equations, often combining convergence acceleration with dynamical system analysis to improve stability and robustness [24]. In this context, several modifications of the fractional Newton framework have been proposed, incorporating nonlinear correction terms and multi-step strategies. For instance, Shams et al. [25] 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 $ECFS_1$ [12], further improve convergence through an additional correction step. More advanced formulations include the two-step method $ECFS_2$ [26], which achieves order $2\gamma + 1$ via nonlinear ratio-based corrections, and the scheme $ECFS_3$ [27], where auxiliary functions are introduced to further enhance convergence behavior and stability.

While these approaches achieve higher convergence orders and improved local performance, they often come at the cost of increased computational complexity. Moreover, their analysis is predominantly focused on local convergence, and they are typically developed within a single fractional framework, which limits the ability to exploit complementary properties of different fractional operators.

In summary, although both classical and fractional iterative schemes provide a solid theoretical foundation and increasingly high convergence orders, their practical performance remains constrained by several key limitations. In particular, higher-order methods may exhibit reduced robustness and increased sensitivity in the presence of strong nonlinearities or stiff behavior. Furthermore, the evaluation of fractional derivatives introduces non-negligible computational overhead due to memory effects, especially in large-scale settings [28,29].

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 formulated within a single fractional framework, without fully exploiting the complementary characteristics 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 these challenges, this work introduces a family of fractional-order iterative methods based on a common computational structure that can be instantiated using both Caputo and $\mathcal{ABC}$ derivatives. The proposed approach aims to balance convergence speed, stability, and computational efficiency by leveraging the complementary characteristics 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 common computational framework for both Caputo and $\mathcal{ABC}$ operators, enabling a systematic and consistent comparison of their numerical behavior within the same algorithmic structure;

- 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 indicating improved efficiency with respect to selected existing fractional methods.

The novelty of the proposed approach lies in the construction of an operator-agnostic iterative structure that can be instantiated with different fractional derivatives, combined with a dynamical systems perspective that provides insight into both local convergence properties and global basin 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 for controlling convergence speed and stability. The convergence analysis is carried out using fractional Taylor expansions in a neighborhood of a simple root of the nonlinear equation $f(x) = 0$.

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

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

and define the ratio

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

The proposed Caputo-based scheme, denoted by NCFS_1, is defined as

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

where $\lambda_1, \lambda_2 \in \mathbb{R}$ are free parameters. Replacing the Caputo derivative in (9) with the Atangana–Baleanu–Caputo derivative yields the corresponding $\mathcal{ABC}$-based bi-parametric scheme, denoted by NFS_1^{abc}:

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

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

2.1. Justification of the Unified Fractional Framework

The proposed framework considers both Caputo and $\mathcal{ABC}$ derivatives, which differ in kernel structure (singular versus non-singular). These operators are not combined within a single iterative scheme; instead, they are used to instantiate two parallel methods, NCFS_1 and NFS_1^{abc}, sharing an identical algorithmic structure.

This common structure enables a systematic and consistent comparison of the resulting schemes within the same computational setting. In particular, both operators admit fractional Taylor expansions and exhibit comparable leading-order asymptotic behavior in a neighborhood of a simple root, under standard smoothness assumptions. This allows the convergence analysis to be developed within a unified computational framework.

The proposed approach is therefore complementary rather than algebraically unified, leveraging the distinct characteristics of the two operators—namely, the stronger memory effects of the Caputo derivative and the smoother kernel associated with the $\mathcal{ABC}$ formulation—without requiring a direct coupling between them.

2.2. Parameter Selection Guidelines

The performance of the proposed schemes depends on the appropriate choice of the fractional order γ and the parameters λ_1, λ_2. The following guidelines provide practical insight into their selection:

- Fractional order $\boldsymbol{\gamma}$:
  Values of γ close to 1 typically yield faster convergence and behavior closer to classical methods, while smaller values $\gamma \in (0.6, 0.9)$ may improve stability and robustness, particularly for highly nonlinear or stiff problems.
- Parameter $\boldsymbol{\lambda_1}$: This parameter controls the weight of the correction term. Moderate values (e.g., $\lambda_1 \in [0, 1]$) generally enhance convergence without introducing significant instability.
- Parameter $\boldsymbol{\lambda_2}$: This parameter influences the nonlinear rational correction. Small values (e.g., $\lambda_2 \approx 0$) tend to favor stability, while larger values may accelerate convergence at the expense of reduced robustness.
- Practical strategy: In numerical experiments, good performance is typically obtained by selecting γ close to 1 and tuning λ_1, λ_2 empirically based on residual reduction and convergence speed.

These guidelines are consistent with the numerical results reported in Section 3, where appropriate parameter choices lead to rapid convergence (often within a few iterations) and residual errors ranging from 10^{-3} to 10^{-18}.

2.3. Computational Complexity

At each iteration, the evaluation of the fractional derivatives ${}^{c}\mathcal{D}f(x_n)$ and ${}^{abc}\mathcal{D}f(x_n)$ involves the computation of history-dependent convolution terms. For n discretization points, a straightforward implementation leads to a cost of $\mathcal{O}(n^2)$ per iteration.

Consequently, the overall complexity of the proposed iterative schemes is $\mathcal{O}(kn^2)$, where k denotes the number of iterations. By employing fast convolution techniques or memory-efficient approximations, this cost can be reduced to $\mathcal{O}(kn \log n)$, thereby improving computational efficiency for larger-scale problems.

2.4. 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 $f'(\xi) \neq 0$. Assume that x_0 is sufficiently close to ξ and that*

$$\lambda_1 \neq -2, \qquad \mathcal{A}_1 \neq 0.$$

Then the method (9) converges locally to ξ with order

$$p = \gamma^2 + \gamma + 1, \qquad 0 < \gamma < 1.$$

More precisely, the leading error term satisfies

$$e_{n+1} = -\frac{(\lambda_1 + 2)\mathcal{A}_1^{\gamma+1}}{\gamma} e_n^{\gamma^2+\gamma+1} + \mathcal{O}\Big(e_n^{\gamma^2+\gamma+1+\eta}\Big),$$

where

$$\eta = \min\{\gamma^2, \gamma(1-\gamma)\}.$$

If the leading coefficient vanishes, for example when $\lambda_1 = -2$, then a higher asymptotic order may occur, determined by the first nonzero higher-order term.

Proof. Let $e_n = x_n - \xi$. We first expand $f(x_n)$ and ${}^c\mathcal{D}^\gamma f(x_n)$ 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

$$f(x_n) = \frac{{}^c\mathcal{D}^\gamma f(\xi)}{\Gamma(\gamma+1)}\Big(e_n^\gamma + K_2 e_n^{2\gamma} + K_3 e_n^{3\gamma} + \mathcal{O}(e_n^{4\gamma})\Big), \tag{11}$$

where

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

Taking the Caputo fractional derivative of (11), we obtain

$${}^c\mathcal{D}^\gamma f(x_n) = \frac{{}^c\mathcal{D}^\gamma f(\xi)}{\Gamma(\gamma+1)}\left[\Gamma(\gamma+1) + \frac{\Gamma(2\gamma+1)K_2}{\Gamma(\gamma+1)} e_n^\gamma + \frac{\Gamma(3\gamma+1)K_3}{\Gamma(2\gamma+1)} e_n^{2\gamma} + \mathcal{O}(e_n^{3\gamma})\right]. \tag{12}$$

Hence, by dividing (11) by (12), we obtain

$$\begin{aligned}\frac{f(x_n)}{{}^c\mathcal{D}^\gamma f(x_n)} = &\left(\frac{1}{\Gamma(\gamma+1)}\right) e_n^\gamma + \left(\frac{K_2}{\Gamma(\gamma+1)} - \frac{2^{2\gamma}K_2}{\sqrt{\pi}\,\Gamma(\gamma+1)^2}\,\Gamma\Big(\gamma+\frac{1}{2}\Big)\right) e_n^{2\gamma} \\ &+ \mathcal{A}_0 e_n^{3\gamma} + \mathcal{O}(e_n^{4\gamma}),\end{aligned} \tag{13}$$

where

$$\begin{aligned}\mathcal{A}_0 = &\frac{K_3}{\Gamma(\gamma+1) - \frac{2^{2\gamma}K_2^2}{\sqrt{\pi}\,\Gamma(\gamma+1)^2}\,\Gamma\Big(\gamma+\frac{1}{2}\Big) - \frac{3^{3\gamma}\sqrt{3}\,K_3}{2\sqrt{\pi}\,\Gamma(\gamma+1)^2 2^{2\gamma}} \frac{\Gamma\Big(\gamma+\frac{1}{3}\Big)}{\Gamma\Big(\gamma+\frac{2}{3}\Big)}} \cdot \frac{1}{\Gamma\Big(\gamma+\frac{1}{2}\Big)} \\ &+ \frac{2^{4\gamma}K_2^2}{\pi\,\Gamma(\gamma+1)^3}\Big[\Gamma\Big(\gamma+\tfrac{1}{2}\Big)\Big]^2.\end{aligned} \tag{14}$$

Using (13) and applying the binomial expansion to the fractional power $(\cdot)^{1/\gamma}$, the predictor step in (9) yields

$$y_n - \xi = \mathcal{A}_1 e_n^{\gamma+1} + \mathcal{A}_2 e_n^{2\gamma+1} + \mathcal{O}(e_n^{3\gamma+1}), \tag{15}$$

where

$$\mathcal{A}_1 = \left(\frac{-K_2}{\gamma} + \frac{2^{2\gamma}K_2}{\gamma\Gamma(\gamma+1)\sqrt{\pi}}\Gamma\left(\gamma+\frac{1}{2}\right)\right),$$

$$\mathcal{A}_2 = -\frac{1}{\gamma}\Bigg[-K_3 + \frac{2^{2\gamma}\Gamma\left(\gamma+\frac{1}{2}\right)K_2^2}{\Gamma(\gamma+1)\sqrt{\pi}} + \frac{3^{3\gamma}\sqrt{3}\,K_3}{2\,\Gamma(\gamma+1)\sqrt{\pi}\,(2^\gamma)^2}\frac{\Gamma\left(\gamma+\frac{1}{3}\right)\Gamma\left(\gamma+\frac{2}{3}\right)}{\Gamma\left(\gamma+\frac{1}{2}\right)}$$

$$-\frac{2^{4\gamma}\Gamma\left(\gamma+\frac{1}{2}\right)^2K_2^2}{\Gamma(\gamma+1)^2\pi}\Bigg]$$

$$-\frac{1}{2\gamma}\left(\frac{1}{\gamma}-1\right)\left(K_2 - \frac{2^{2\gamma}K_2}{\Gamma(\gamma+1)\sqrt{\pi}}\Gamma\left(\gamma+\frac{1}{2}\right)\right)^2.$$

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

$$f(y_n) = \frac{{}^c\mathcal{D}^\gamma f(\xi)}{\Gamma(\gamma+1)}\left[(y_n-\xi)^\gamma + K_2(y_n-\xi)^{2\gamma}\right] + \mathcal{O}\left(e_n^{3\gamma^2+3\gamma}\right). \tag{16}$$

By substituting (15) into (16), we get:

$$(y_n-\xi)^\gamma = \mathcal{A}_1^\gamma e_n^{\gamma^2+\gamma} + \gamma\,\mathcal{A}_1^{\gamma-1}\mathcal{A}_2\,e_n^{\gamma^2+2\gamma} + \mathcal{O}\left(e_n^{\gamma^2+3\gamma}\right). \tag{17}$$

and

$$(y_n-\xi)^{2\gamma} = \mathcal{A}_1^{2\gamma}e_n^{2\gamma^2+2\gamma} + \mathcal{O}\left(e_n^{2\gamma^2+3\gamma}\right). \tag{18}$$

The asymptotic ordering of the exponents plays a crucial role in establishing the convergence behavior of the proposed method. For $0 < \gamma < 1$, consider the exponents

$$\gamma^2+\gamma, \quad \gamma^2+2\gamma, \quad 2\gamma^2+2\gamma, \quad \gamma^2+3\gamma.$$

A direct comparison shows that

$$\gamma^2+\gamma < \gamma^2+2\gamma \quad \text{since } \gamma > 0,$$

$$\gamma^2+2\gamma < 2\gamma^2+2\gamma \quad \text{since } \gamma^2 > 0,$$

$$2\gamma^2+2\gamma < \gamma^2+3\gamma \quad \text{since } \gamma(1-\gamma) > 0.$$

Consequently, we obtain the strict ordering

$$\gamma^2+\gamma < \gamma^2+2\gamma < 2\gamma^2+2\gamma < \gamma^2+3\gamma, \qquad \forall\,\gamma \in (0,1).$$

This hierarchy implies that any term of order $e_n^{\gamma^2+\gamma}$ dominates all higher-order contributions in the asymptotic expansion, while terms involving $e_n^{\gamma^2+2\gamma}$, $e_n^{2\gamma^2+2\gamma}$, and $e_n^{\gamma^2+3\gamma}$ are strictly higher-order perturbations.

Therefore, truncation of the expansion at order $e_n^{\gamma^2+3\gamma}$ is rigorously justified and does not alter the leading-order asymptotic structure of the method. In particular, all neglected terms are asymptotically negligible with respect to the dominant term $e_n^{\gamma^2+\gamma}$ as $n \to \infty$.

$$f(y_n) = \frac{{}^{c}\mathcal{D}^{\gamma} f(\xi)}{\Gamma(\gamma+1)}\Bigg[\mathcal{A}_1^{\gamma} e_n^{\gamma^2+\gamma} + \gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2\, e_n^{\gamma^2+2\gamma} + K_2\mathcal{A}_1^{2\gamma}\, e_n^{2\gamma^2+2\gamma} + \frac{\gamma(\gamma-1)}{2}\mathcal{A}_1^{\gamma-2}\mathcal{A}_2^2\, e_n^{\gamma^2+3\gamma} + \mathcal{O}\Big(e_n^{2\gamma^2+3\gamma}\Big)\Bigg]. \tag{19}$$

Dividing the above expansion of $f(y_n)$ by (11), we obtain

$$\rho_n = \frac{f(y_n)}{f(x_n)} = \mathcal{A}_1^{\gamma} e_n^{\gamma^2} + \Big(\gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2 - K_2\mathcal{A}_1^{\gamma}\Big)e_n^{\gamma^2+\gamma} + K_2\mathcal{A}_1^{2\gamma}e_n^{2\gamma^2+\gamma} + \mathcal{A}_3 e_n^{\gamma^2+2\gamma} + \mathcal{O}\Big(e_n^{2\gamma^2+2\gamma}\Big). \tag{20}$$

where

$$\mathcal{A}_3 = \Bigg[\frac{\gamma(\gamma-1)}{2}\mathcal{A}_1^{\gamma-2}\mathcal{A}_2^2 - \gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2 K_2 + \Big(K_2^2 - K_3\Big)\mathcal{A}_1^{\gamma}\Bigg].$$

Therefore,

$$1 + \lambda_2\rho_n = 1 + \lambda_2\mathcal{A}_1^{\gamma} e_n^{\gamma^2} + \lambda_2\Big(\gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2 - K_2\mathcal{A}_1^{\gamma}\Big)e_n^{\gamma^2+\gamma} + \lambda_2 K_2\mathcal{A}_1^{2\gamma}e_n^{2\gamma^2+\gamma} + \lambda_2\mathcal{A}_3 e_n^{\gamma^2+2\gamma} + \mathcal{O}\Big(e_n^{2\gamma^2+2\gamma}\Big). \tag{21}$$

In the second sub-step of (9), we expand the rational correction factor as follows:

$$\Theta_n = \frac{1 + (\lambda_2+2)\rho_n}{1+\lambda_2\rho_n} = 1 + 2\mathcal{A}_1^{\gamma} e_n^{\gamma^2} - 2\lambda_2\mathcal{A}_1^{2\gamma}e_n^{2\gamma^2} + 2\Big(\gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2 - K_2\mathcal{A}_1^{\gamma}\Big)e_n^{\gamma^2+\gamma} + \mathcal{O}\Big(e_n^{2\gamma^2+\gamma}\Big). \tag{22}$$

Consequently, we derive the error expression for Ψ_n, which corresponds to the second sub-step of (9), as follows:

$$\Psi_n = \lambda_1\rho_n + \frac{1+(\lambda_2+2)\rho_n}{1+\lambda_2\rho_n} = 1 + (\lambda_1+2)\mathcal{A}_1^{\gamma} e_n^{\gamma^2} - 2\lambda_2\mathcal{A}_1^{2\gamma}e_n^{2\gamma^2} + (\lambda_1+2)\mathcal{B}_1 e_n^{\gamma^2+\gamma} + + \mathcal{O}\Big(e_n^{2\gamma^2+\gamma}\Big), \tag{23}$$

where

$$\mathcal{B}_1 = \gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2 - K_2\mathcal{A}_1^{\gamma}.$$

$$\frac{f(y_n)}{{}^{c}\mathcal{D}^{\gamma} f(x_n)} = \frac{\mathcal{A}_1^{\gamma}}{\Gamma(\gamma+1)}\, e_n^{\gamma^2+\gamma} + \left(\frac{\gamma\,\mathcal{A}_1^{\gamma-1}\mathcal{A}_2}{\Gamma(\gamma+1)} - \frac{2^{2\gamma}\Gamma\Big(\gamma+\frac{1}{2}\Big)}{\sqrt{\pi}\,\Gamma(\gamma+1)^2}K_2\mathcal{A}_1^{\gamma}\right)e_n^{\gamma^2+2\gamma} + \frac{K_2\mathcal{A}_1^{2\gamma}}{\Gamma(\gamma+1)}\, e_n^{2\gamma^2+2\gamma} + \mathcal{O}(e_n^{\gamma^2+2\gamma+1}). \tag{24}$$

Thus, multiplying (23) and (24), we obtain the result.

$$\begin{aligned}\Xi &= \Gamma(\gamma+1)\frac{f(y_n)}{{}_c\mathcal{D}^{\gamma}f(x_n)}\left(\lambda_1\rho_n + \frac{1+(\lambda_2+2)\rho_n}{1+\lambda_2\rho_n}\right)\\ &= \mathcal{A}_1^{\gamma}e_n^{\gamma^2+\gamma} + (\lambda_1+2)\mathcal{A}_1^{2\gamma}e_n^{2\gamma^2+\gamma} + \left(\gamma\mathcal{A}_1^{\gamma-1}\mathcal{A}_2 - \frac{2^{2\gamma}\Gamma\left(\gamma+\frac{1}{2}\right)}{\Gamma(\gamma)\gamma\sqrt{\pi}}K_2\mathcal{A}_1^{\gamma}\right)e_n^{\gamma^2+2\gamma}\\ &\quad + \left(K_2\mathcal{A}_1^{2\gamma} + (\lambda_1+2)\mathcal{A}_1^{\gamma}\mathcal{B}_1\right)e_n^{2\gamma^2+2\gamma} + \mathcal{O}\left(e_n^{\gamma^2+2\gamma+1}\right).\end{aligned} \tag{25}$$

Thus, factoring out the leading term and applying the generalized binomial expansion to the fractional power $(\cdot)^{1/\gamma}$, we obtain

$$\begin{aligned}&\left[\Gamma(\gamma+1)\frac{f(y_n)}{{}_c\mathcal{D}^{\gamma}f(x_n)}\left(\lambda_1\rho_n + \frac{1+(\lambda_2+2)\rho_n}{1+\lambda_2\rho_n}\right)\right]^{1/\gamma}\\ &\qquad = \mathcal{A}_1e_n^{\gamma+1} + \frac{(\lambda_1+2)\mathcal{A}_1^{\gamma+1}}{\gamma}e_n^{\gamma^2+\gamma+1} + \mathcal{O}\left(e_n^{\gamma^2+\gamma+1+\eta}\right),\end{aligned} \tag{26}$$

where

$$\eta = \min\{\gamma^2, \gamma(1-\gamma)\}.$$

The exponent $\gamma(1-\gamma)$ accounts for the contribution of order $e_n^{2\gamma+1}$ relative to the leading term $e_n^{\gamma^2+\gamma+1}$, while the term γ^2 accounts for the next possible contribution arising from the expansion of the rational correction factor. This gives a conservative leading-order remainder sufficient for determining the local convergence order.

Substituting this expansion and (15) into the second sub-step of (9), we obtain

$$\begin{aligned}e_{n+1} &= x_{n+1} - \xi\\ &= \mathcal{A}_1e_n^{\gamma+1} + \mathcal{A}_2e_n^{2\gamma+1} + \mathcal{O}(e_n^{3\gamma+1})\\ &\quad - \left[\mathcal{A}_1e_n^{\gamma+1} + \frac{(\lambda_1+2)\mathcal{A}_1^{\gamma+1}}{\gamma}e_n^{\gamma^2+\gamma+1} + \mathcal{O}\left(e_n^{\gamma^2+\gamma+1+\eta}\right)\right].\end{aligned} \tag{27}$$

After cancellation of the common predictor term, the leading error equation becomes

$$e_{n+1} = -\frac{(\lambda_1+2)\mathcal{A}_1^{\gamma+1}}{\gamma}e_n^{\gamma^2+\gamma+1} + \mathcal{O}\left(e_n^{\gamma^2+\gamma+1+\eta}\right). \tag{28}$$

Since $\lambda_1 \neq -2$ and $\mathcal{A}_1 \neq 0$, the leading coefficient is nonzero. Therefore, the local order of convergence is

$$p = \gamma^2 + \gamma + 1.$$

This proves the theorem. □

Remark 1 (Degenerate leading coefficient)**.** *The conclusion of Theorem 1 is stated under the nondegeneracy assumptions $\lambda_1 \neq -2$ and $\mathcal{A}_1 \neq 0$, which ensure that the leading coefficient in the local error equation does not vanish. If $\lambda_1 = -2$, the leading term of order $e_n^{\gamma^2+\gamma+1}$ cancels. In this degenerate case, the convergence order is determined by the first nonzero higher-order term in the error expansion. Therefore, a higher asymptotic order may occur, but its precise value depends on the structure of the remaining coefficients and must be determined from the next nonzero term.*

Remark 2 (Analogous convergence behavior of the $\mathcal{ABC}$-based scheme)**.** *Under assumptions analogous to those of Theorem 1, the method (10) can be analyzed using a similar asymptotic*

expansion and error-propagation strategy. This is motivated by the fact that both the Caputo and $\mathcal{ABC}$ fractional operators admit fractional Taylor-type expansions in a neighborhood of a simple root ξ, provided that f is sufficiently smooth and the corresponding fractional derivatives exist.

Although the two operators differ in their kernel structures, namely a singular power-law kernel in the Caputo case and a non-singular Mittag–Leffler kernel in the $\mathcal{ABC}$ case, their leading-order local asymptotic behavior near ξ exhibits analogous dominant terms. Consequently, the quantity

$$\frac{f(x_n)}{{}^{abc}\mathcal{D}^{\gamma} f(x_n)}$$

admits a formally similar expansion, allowing the same type of binomial and error-propagation arguments to be employed.

Therefore, the $\mathcal{ABC}$-based scheme (10) is expected to exhibit the same asymptotic convergence order

$$p = \gamma^2 + \gamma + 1,$$

possibly with a different error constant depending on the $\mathcal{ABC}$ fractional derivatives and kernel structure. This behavior is further investigated numerically in the next section.

3. Numerical Experiments and Implementation

This section describes the numerical framework used to assess the proposed fractional iterative schemes NCFS_1 and NFS_1^{abc} for solving nonlinear equations. The experiments are designed to evaluate both their computational performance and their global dynamical behavior.

The numerical assessment is organized around two complementary objectives. First, we examine the convergence behavior of the proposed methods on representative nonlinear test problems, using iteration counts, step differences, residual norms, CPU time, and memory usage as performance indicators. Second, we compare the proposed schemes with existing fractional iterative methods, including ECFS_1–ECFS_3, and analyze their basins of attraction in order to investigate the dependence of convergence on the initial guess.

For all experiments, the parameters are fixed as $\lambda_1 = 0.02$ and $\lambda_2 = 1.5$. These values were selected empirically to obtain stable convergence behavior across the considered test problems and are kept unchanged throughout the numerical study to ensure a consistent comparison among the methods. Although the theoretical analysis focuses on the fractional case $0 < \gamma < 1$, the value $\gamma = 1$ is also included in some experiments as the classical integer-order reference case. A detailed summary of the parameters used throughout the study is provided in Appendix A (Table A1).

All computations were carried out in MATLAB R2023b on an Intel Core i7 workstation with 16 GB RAM using double-precision arithmetic. The stopping criteria were

$$|x_{n+1} - x_n| \leq \varepsilon \quad \text{or} \quad |f(x_n)| \leq \varepsilon, \quad \varepsilon = 10^{-12}. \tag{29}$$

The numerical performance is evaluated using the following indicators:

(i) Number of iterations required to satisfy the stopping criteria;
(ii) Step differences and residual norms at convergence;
(iii) CPU time and memory usage;
(iv) Number of roots detected in the basin-of-attraction analysis;
(v) Global convergence behavior and basin complexity, assessed through the Convergence Area Index and the Wada measure.

To quantify the qualitative structure of the basins of attraction, we employ the Wada measure [30,31], which provides an indicator of the complexity of basin boundaries in nonlinear dynamical systems. For a compact set $F \subset \mathbb{C}$, it is defined as

$$W(F) = \lim_{\varepsilon \to 0} \frac{N_3(\varepsilon)}{N(\varepsilon)}, \tag{30}$$

where $N(\varepsilon)$ denotes the number of ε-sized boxes covering F, and $N_3(\varepsilon)$ denotes the number of such boxes containing points associated with at least three distinct basins of attraction. Larger values of $W(F)$ indicate more intricate and intertwined basin boundaries and, therefore, greater sensitivity of the basin geometry to the choice of the initial point. When fewer than three basins are detected, the Wada measure is correspondingly zero in the present computational setting.

In addition, we consider the Convergence Area Index (CAI) [32], which measures the fraction of initial points in a prescribed computational domain that converge to a root of the nonlinear equation $f(z) = 0$. It is defined as

$$\mathrm{CAI} = \frac{N_{\mathrm{conv}}}{N_{\mathrm{total}}}, \tag{31}$$

where N_{conv} is the number of initial points that satisfy the convergence criteria within the maximum number of iterations, and N_{total} is the total number of tested initial points. The CAI takes values in $[0, 1]$, with values close to 1 indicating that convergence occurs for most initial points in the selected domain.

Algorithm 1 summarizes the core two-step iteration used in the numerical experiments, while Figure 1 illustrates the computational workflow adopted for the basin-of-attraction analysis. For each initial point in the selected complex grid, the iterative scheme is applied until one of the prescribed stopping criteria is satisfied or the maximum number of iterations is reached. The resulting convergence information is then used to assign the initial point to a root or to a nonconvergent class and to compute the associated performance indicators, including the average number of iterations, the CAI, and the Wada measure.

Algorithm 1 Core two-step fractional iterative scheme and its $\mathcal{ABC}$-based variant for solving $f(x) = 0$

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

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

1: Set $n \leftarrow 0$.

2: **repeat**

Step 1: Predictor

3: Compute

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

For the $\mathcal{ABC}$ variant, replace ${}^{c}\mathcal{D}^{\gamma}$ with ${}^{abc}\mathcal{D}^{\gamma}$.

Step 2: Corrector

4: Compute

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

Algorithm 1 *Cont.*

5: Compute

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

For the $\mathcal{ABC}$ variant, replace ${}^{c}\mathcal{D}^{\gamma}$ with ${}^{abc}\mathcal{D}^{\gamma}$.

6: $n \leftarrow n+1$.

7: **until** $|x_n - x_{n-1}| \leq \varepsilon$ **or** $|f(x_n)| \leq \varepsilon$ **or** $n \geq N_{\max}$

8: $r \leftarrow x_n$.

9: **return** r.

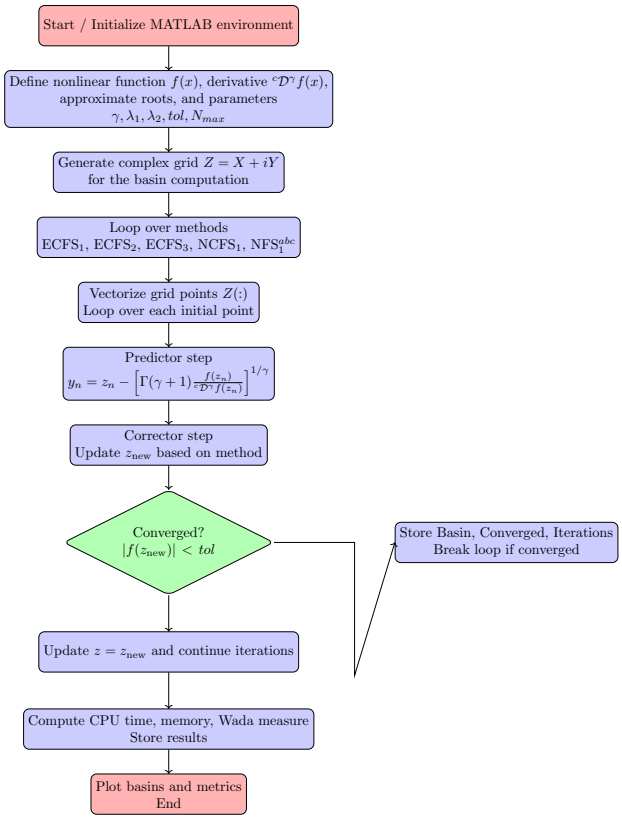


Figure 1. Computational workflow used for the basin-of-attraction analysis of the fractional iterative schemes. The core two-step iteration is applied to each initial point of the selected complex grid, and the resulting convergence data are used to construct convergence regions and compute performance indicators such as the Convergence Area Index and the Wada measure.

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) [33]

Nonlinear equations arise naturally in electrical engineering, particularly in the analysis of diode and transistor circuits, where exponential current–voltage relations lead to nonlinear root-finding problems.

We consider the Shockley diode model

$$I = I_s\left(e^{\frac{V}{n_1 V_T}} - 1\right), \tag{32}$$

where I_s is the saturation current, V is the diode voltage, V_T is the thermal voltage, and n_1 is the ideality factor. Applying Kirchhoff's voltage law to a simple diode-resistor circuit gives

$$V_{in} = IR + V, \tag{33}$$

where R is the resistance and V_{in} is the input voltage.

Substitution of the diode current into Kirchhoff's law leads to the nonlinear equation

$$f(V) = I_s R\left(e^{\frac{V}{n_1 V_T}} - 1\right) + V - V_{in} = 0. \tag{34}$$

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

$$f(V) = 10^{-6}(1000)\left(e^{\frac{V}{0.026}} - 1\right) + V - 5 = 0, \tag{35}$$

or equivalently

$$f(V) = 10^{-3}\left(e^{\frac{V}{0.026}} - 1\right) + V - 5 = 0. \tag{36}$$

The nonlinear Equation (36) is characterized by a strong exponential term and therefore provides a useful benchmark for testing the convergence behavior of the proposed fractional iterative schemes.

For error evaluation, a reference solution of (36) was computed numerically as

$$V^* = 0.220281002036106.$$

This value is used to evaluate the numerical accuracy of the approximations where applicable. At each iteration, the absolute error is evaluated relative to V^*, allowing a direct assessment of convergence behavior. The use of a reliable reference solution also facilitates a fair comparison among different iterative schemes. In particular, smaller deviations from V^* indicate higher numerical precision and faster stabilization of the iterative sequence. Throughout the computations, the initial approximation is selected as

$$x_0 = 0.3,$$

which provides the starting point for the iterative process.

The step differences and performance indicators reported in Tables 1 and 2 and Figure 2 show that NCFS_1 provides a faster reduction in the step difference than NFS_1^{abc} for this test problem. In particular, Table 1 indicates that, for the considered values of γ, NCFS_1 reduces $|x_{n+1} - x_n|$ to values of order 10^{-8} or smaller within the reported iterations. By contrast, NFS_1^{abc} exhibits a slower decay of the step difference, especially for larger values of γ.

These observations are consistent with the aggregate performance indicators in Table 2. In this experiment, NCFS_1 attains a smaller step-difference norm, a higher convergence percentage, and a slightly lower CPU time than NFS_1^{abc}. This suggests that the Caputo-based formulation can improve the convergence efficiency for the diode equation without increasing the CPU time.

The comparison with the existing ECFS_1–ECFS_3 schemes is reported in Table 3 and illustrated in Figure 3. The results show that, for $\gamma = 0.98, 0.99, 1.00$, the proposed methods NCFS_1 and NFS_1^{abc} require fewer iterations than the ECFS schemes. In the reported tests, NCFS_1 and NFS_1^{abc} converge in 5–8 iterations, whereas the ECFS methods require 11–16 iterations.

Table 1. Step differences $|x_{n+1} - x_n|$ obtained by $NCFS_1$ and NFS_1^{abc} for solving (36) 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 $\gamma = 0.5$		
NFS_1^{abc}	2.14×10^{1}	3.72×10^{-1}	6.81×10^{-2}	2.45×10^{-3}
$NCFS_1$	1.87×10^{-1}	2.64×10^{-3}	3.92×10^{-5}	5.16×10^{-8}
		For $\gamma = 0.7$		
NFS_1^{abc}	3.45×10^{1}	4.26×10^{-1}	7.93×10^{-2}	3.16×10^{-2}
$NCFS_1$	2.88×10^{-1}	3.11×10^{-3}	5.28×10^{-4}	7.43×10^{-9}
		For $\gamma = 0.9$		
NFS_1^{abc}	5.12×10^{1}	5.87×10^{-1}	1.24×10^{-2}	6.48×10^{-3}
$NCFS_1$	4.06×10^{-1}	4.22×10^{-2}	7.61×10^{-6}	1.26×10^{-9}

Table 2. Overall performance indicators of $NCFS_1$ and NFS_1^{abc} for solving (36) with $\gamma = 0.5, 0.7, 0.9$.

Method	Step-Difference Norm	CPU Time (s)	Memory (MB)	Convergence (%)
NFS_1^{abc}	2.31×10^{-2}	3.12	0.0812	92.4
$NCFS_1$	1.12×10^{-8}	2.94	0.0817	97.8

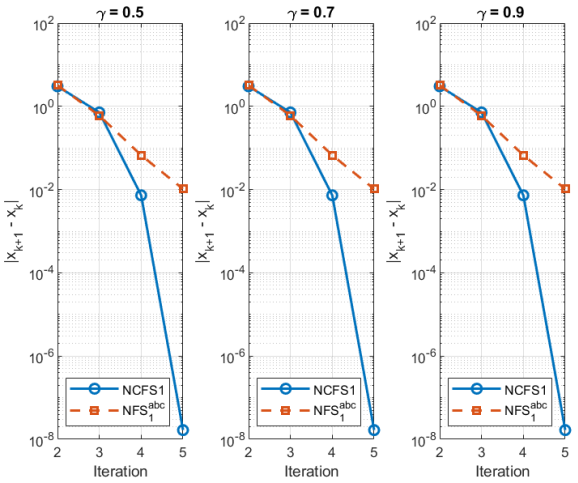


Figure 2. Step−difference decay of $NCFS_1$ and NFS_1^{abc} for solving Equation (36), for different values of the fractional order γ.

The proposed schemes also produce smaller final step differences and residual norms. In particular, the residuals obtained by $NCFS_1$ and NFS_1^{abc} reach values between 10^{-15} and 10^{-18}, while those obtained by the ECFS methods remain between 10^{-11} and 10^{-12}. This indicates higher numerical accuracy for the proposed schemes in this experiment.

In terms of computational cost, $NCFS_1$ and NFS_1^{abc} also show smaller CPU times, consistently with their lower iteration counts. The memory usage is of the same order for all methods, indicating that the improved performance is mainly due to the convergence behavior rather than to substantial differences in storage requirements. Overall, for the diode equation and for the near-classical values $\gamma = 0.98, 0.99, 1.00$, the proposed schemes provide a more efficient and accurate alternative to the ECFS methods, with NFS_1^{abc} giving the smallest residuals and requiring the fewest iterations among the tested methods.

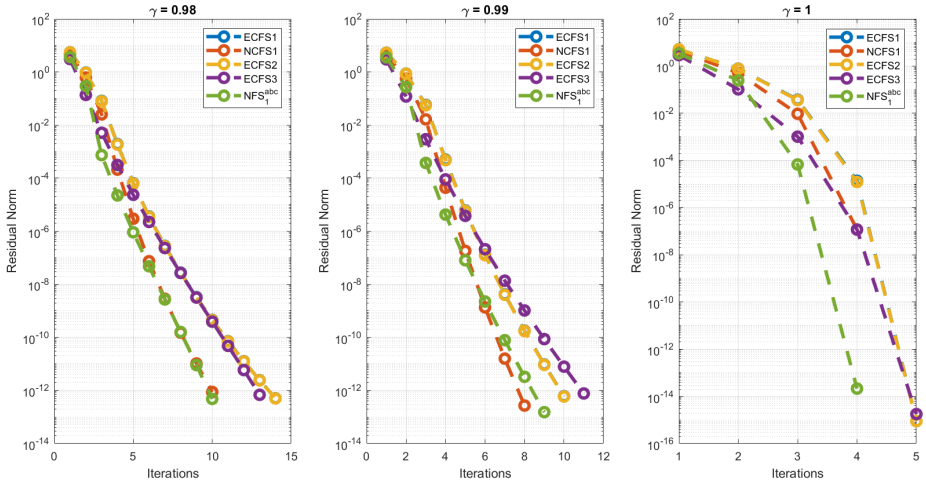


Figure 3. Residual error decay of $\mathrm{ECFS}_1-\mathrm{ECFS}_3$, NCFS_1, and NFS_1^{abc} for solving Equation (36), for $\gamma = 0.98, 0.99, 1.00$.

Table 3. Performance comparison of ECFS_1–ECFS_3, NCFS_1, and NFS_1^{abc} for solving (36) with $\gamma = 0.98, 0.99, 1.00$.

γ	Method	Iter	CPU (s)	$\lvert x_{n+1} - x_n \rvert$	$\lvert f(x_n) \rvert$	Memory (MB)
0.98	ECFS_1	16	0.0084	2.80×10^{-2}	1.12×10^{-11}	0.01243
	ECFS_2	11	0.0058	3.09×10^{-2}	2.45×10^{-12}	0.01345
	ECFS_3	16	0.0054	2.80×10^{-2}	1.88×10^{-11}	0.01543
	NCFS_1	8	0.0042	1.22×10^{-3}	4.13×10^{-15}	0.01254
	NFS_1^{abc}	7	0.0038	1.05×10^{-3}	2.97×10^{-16}	0.01154
0.99	ECFS_1	15	0.0078	2.91×10^{-2}	9.55×10^{-12}	0.03245
	ECFS_2	12	0.0060	3.19×10^{-2}	3.12×10^{-11}	0.03675
	ECFS_3	15	0.0056	2.91×10^{-2}	1.45×10^{-11}	0.03654
	NCFS_1	7	0.0030	1.34×10^{-3}	5.89×10^{-16}	0.03465
	NFS_1^{abc}	6	0.0028	1.12×10^{-3}	3.21×10^{-17}	0.03231
1.00	ECFS_1	14	0.0072	3.03×10^{-2}	7.44×10^{-12}	0.08546
	ECFS_2	12	0.0059	3.28×10^{-2}	1.25×10^{-11}	0.08435
	ECFS_3	14	0.0053	3.03×10^{-2}	6.88×10^{-12}	0.09763
	NCFS_1	6	0.0025	1.48×10^{-3}	1.45×10^{-16}	0.07324
	NFS_1^{abc}	5	0.0020	1.21×10^{-3}	7.89×10^{-18}	0.08563

The basin-of-attraction analysis provides complementary information on the global dynamical behavior of the methods. Table 4 reports the average number of iterations, the Convergence Area Index, the CPU time required for the basin computation, the Wada measure, and the number of detected roots.

Table 4. Dynamical performance metrics for the considered iterative methods. ANI: average number of iterations; CAI: Convergence Area Index; $W(F)$: 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

The CAI values are equal or very close to one for all methods, indicating that almost all initial points in the tested domain converge to one of the roots as presented in Figure 4. Moreover, all methods detect the three roots of the problem. The average number of iterations is also essentially the same, suggesting comparable global convergence efficiency within the basin-analysis setting.

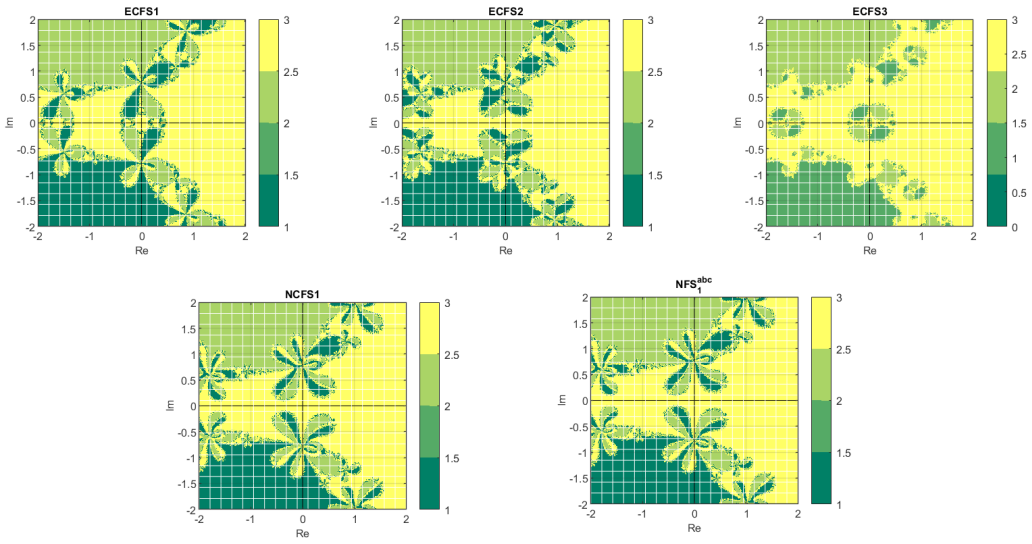


Figure 4. Basins of attraction associated with $ECFS_1-ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving Equation (36).

Differences between the methods are mainly reflected in the geometry of the basin boundaries, as quantified by the Wada measure. The proposed methods $NCFS_1$ and NFS_1^{abc} attain the largest Wada values, $W(F) = 0.229$, indicating more intricate and intertwined basin boundaries for this test problem. By contrast, $ECFS_3$ gives the smallest value, $W(F) = 0.131$, corresponding to a less complex boundary structure.

These results indicate that all methods are robust with respect to convergence over the tested domain, while the proposed schemes exhibit more complex basin geometries. This behavior suggests stronger sensitivity of the basin boundaries to the initial point, although it does not compromise the overall convergence coverage measured by the CAI. Overall, the basin analysis indicates that all methods exhibit reliable convergence over the tested domain, with CAI values equal or very close to one and with all three roots correctly detected. The average number of iterations is essentially identical across the methods, indicating comparable global convergence efficiency in this setting. The main difference concerns the geometry of the basin boundaries: $NCFS_1$ and NFS_1^{abc} produce the largest Wada values, suggesting more intricate boundary structures, whereas $ECFS_3$ yields the smallest value. These observations indicate that the proposed schemes maintain broad convergence coverage while generating more complex basin boundaries for this test problem.

4.2. Chemical Reaction Kinetics (Nonlinear Reactor Model) [34]

Nonlinear algebraic equations frequently arise in chemical engineering, particularly in the modeling of steady-state reactor behavior with nonlinear reaction kinetics.

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

$$F(C_{in} - C) - kC^3 = 0, \tag{37}$$

where C denotes the concentration, C_{in} is the inlet concentration, F is the flow rate, and k is the reaction rate constant.

Assuming an Arrhenius-type temperature dependence, the reaction rate constant 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 absolute temperature.

Substituting this expression into the governing equation gives

$$F(C_{in} - C) - k_0 e^{-\frac{E}{RT}} C^3 = 0. \tag{38}$$

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

$$f(C) = 2 - C - 3e^{-\frac{5000}{8.314 \times 350}} C^3 = 0. \tag{39}$$

Since

$$\frac{5000}{8.314 \times 350} \approx 1.718,$$

this equation can be written as

$$f(C) = 2 - C - 3e^{-1.718} C^3 = 0, \tag{40}$$

or, using the rounded coefficient $3e^{-1.718} \approx 0.54$, as

$$f(C) = 2 - C - 0.54C^3 = 0. \tag{41}$$

The nonlinear Equation (41) is characterized by a cubic reaction term and therefore provides a complementary benchmark to the exponential diode model considered previously. This test problem allows us to examine the behavior of the proposed methods for a polynomial-type nonlinearity, where the convergence properties are not influenced by the strong exponential growth present in (36). It also offers an additional setting for assessing accuracy, robustness, and sensitivity to the initial approximation.

For error evaluation, a reference solution of (41) was computed numerically as

$$C^* = 1.15909260912033.$$

This value is used to evaluate the numerical accuracy of the approximations where applicable. The initial approximation is selected as $x_0 = 6.9$ in order to test the convergence behavior of the methods from a relatively distant starting point.

The step differences and aggregate performance indicators reported in Tables 5 and 6 and Figure 5 compare the behavior of NCFS_1 and NFS_1^{abc} for solving the nonlinear problem (41) for different values of γ.

Table 5 shows that, for the considered values of γ, NCFS_1 produces a faster reduction in the step difference $|x_{n+1} - x_n|$ than NFS_1^{abc} within the reported iterations. In particular, the step differences for NCFS_1 decrease to values of order 10^{-8} or smaller, whereas NFS_1^{abc} exhibits a slower decay. Figure 5 provides a complementary graphical view of the same step-difference behavior.

The aggregate performance indicators in Table 6 lead to the same qualitative conclusion. In this experiment, NCFS_1 attains a smaller step-difference norm, a higher convergence percentage, and a slightly lower CPU time than NFS_1^{abc}.

Table 5. Step differences $|x_{n+1} - x_n|$ obtained by $NCFS_1$ and NFS_1^{abc} when solving (41) 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 $\gamma = 0.5$		
NFS_1^{abc}	2.05×10^{1}	3.51×10^{-1}	5.92×10^{-2}	1.84×10^{-3}
$NCFS_1$	1.65×10^{-1}	2.21×10^{-3}	2.88×10^{-5}	3.74×10^{-8}
		For $\gamma = 0.7$		
NFS_1^{abc}	3.21×10^{1}	4.01×10^{-1}	6.75×10^{-2}	2.71×10^{-2}
$NCFS_1$	2.51×10^{-1}	2.88×10^{-3}	4.12×10^{-4}	5.62×10^{-9}
		For $\gamma = 0.9$		
NFS_1^{abc}	4.88×10^{1}	5.42×10^{-1}	1.08×10^{-2}	5.11×10^{-3}
$NCFS_1$	3.82×10^{-1}	3.91×10^{-2}	6.44×10^{-6}	9.84×10^{-10}

Table 6. Overall performance of $NCFS_1$ and NFS_1^{abc} for solving (41), including step-difference norm, CPU time, memory usage, and convergence percentage.

Method	Step-Difference Norm	CPU Time (s)	Memory (MB)	Convergence (%)
NFS_1^{abc}	2.48×10^{-2}	3.28	0.0755	91.2
$NCFS_1$	9.35×10^{-9}	2.81	0.0689	98.3

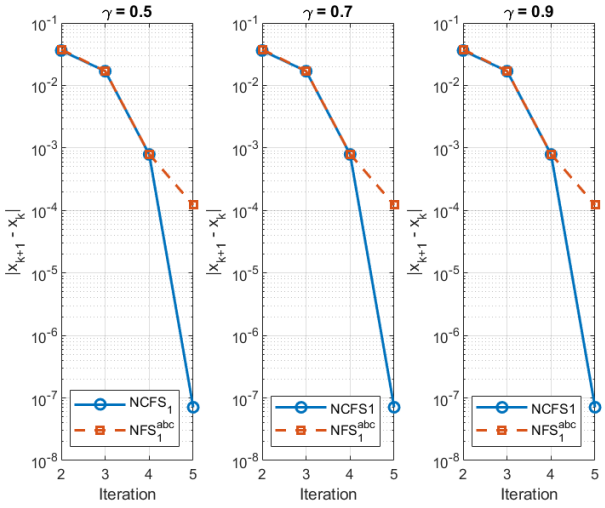


Figure 5. Step−difference decay of $NCFS_1$ and NFS_1^{abc} for solving Equation (41), for different values of the fractional order γ.

Overall, for the nonlinear reactor model and for the fractional orders $\gamma = 0.5, 0.7, 0.9$, the Caputo-based scheme shows better numerical performance than the $\mathcal{ABC}$-based scheme in terms of step-difference reduction, convergence percentage, and computational cost. NFS_1^{abc} remains a viable alternative, but $NCFS_1$ gives the smallest step-difference norm among the two proposed schemes in this test.

The performance metrics reported in Table 7 and illustrated in Figure 6 compare the existing $ECFS_1$–$ECFS_3$ schemes with the proposed $NCFS_1$ and NFS_1^{abc} methods for the reactor model (41). For all tested values of γ, the proposed schemes require fewer iterations than the ECFS methods. In the reported tests, $NCFS_1$ and NFS_1^{abc} converge in 4–7 iterations, whereas the ECFS methods require 11–16 iterations.

Table 7. Performance comparison of $ECFS_1$–$ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving (41) with $\gamma = 0.98, 0.99, 1.00$.

γ	Method	Iter	CPU (s)	$\lvert x_{n+1} - x_n \rvert$	$\lvert f(x_n) \rvert$	Memory (MB)
0.98	$ECFS_1$	16	0.0084	2.80×10^{-2}	1.02×10^{-11}	0.01543
	$ECFS_2$	11	0.0057	3.08×10^{-2}	2.21×10^{-12}	0.02435
	$ECFS_3$	16	0.0053	2.81×10^{-2}	1.79×10^{-11}	0.01593
	$NCFS_1$	7	0.0038	1.12×10^{-3}	5.23×10^{-15}	0.01143
	NFS_1^{abc}	6	0.0035	9.80×10^{-4}	2.14×10^{-16}	0.01546
0.99	$ECFS_1$	15	0.0079	2.91×10^{-2}	8.97×10^{-12}	0.03454
	$ECFS_2$	12	0.0059	3.18×10^{-2}	3.05×10^{-11}	0.04532
	$ECFS_3$	15	0.0055	2.91×10^{-2}	1.42×10^{-11}	0.04646
	$NCFS_1$	6	0.0029	1.30×10^{-3}	6.01×10^{-16}	0.03164
	NFS_1^{abc}	5	0.0032	1.10×10^{-3}	2.77×10^{-17}	0.03453
1.00	$ECFS_1$	14	0.0073	3.03×10^{-2}	7.33×10^{-12}	0.08765
	$ECFS_2$	12	0.0058	3.28×10^{-2}	1.18×10^{-11}	0.07964
	$ECFS_3$	14	0.0053	3.03×10^{-2}	6.75×10^{-12}	0.08656
	$NCFS_1$	5	0.0024	1.45×10^{-3}	1.38×10^{-16}	0.06571
	NFS_1^{abc}	4	0.0020	1.18×10^{-3}	7.53×10^{-18}	0.07861

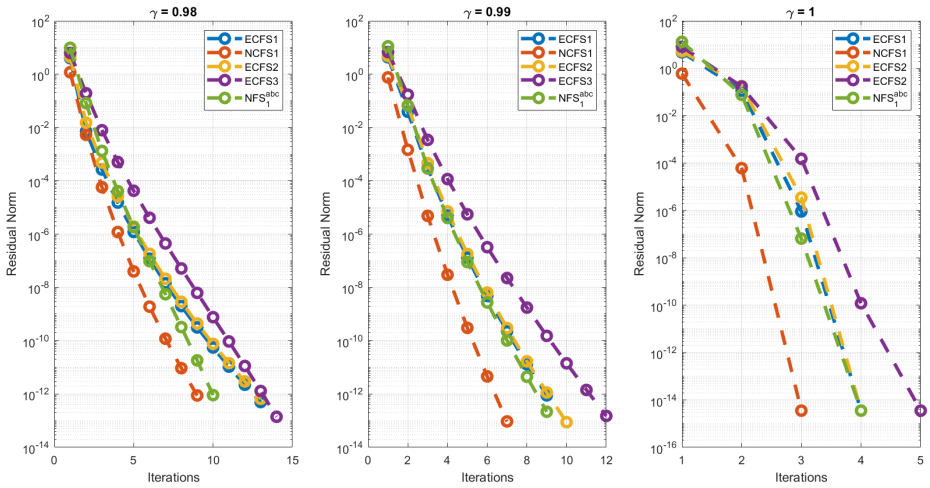


Figure 6. Residual error decay of $ECFS_1$−$ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving Equation (41), for $\gamma = 0.98, 0.99, 1.00$.

This reduction in iteration count is accompanied by smaller final step differences and residual norms. The step differences obtained by the proposed schemes are of order 10^{-3}, compared with values of order 10^{-2} for the ECFS methods. Moreover, the residual norms achieved by $NCFS_1$ and NFS_1^{abc} range from 10^{-15} to 10^{-18}, while those obtained by the ECFS methods remain between 10^{-11} and 10^{-12}. These results indicate higher numerical accuracy for the proposed schemes in this experiment.

In terms of computational cost, the proposed schemes generally require fewer iterations and competitive CPU times, consistently with their faster convergence behavior. The memory usage remains of the same order across all methods. Overall, for the reactor model and for the near-classical values $\gamma = 0.98, 0.99, 1.00$, the proposed schemes provide a more accurate alternative to the ECFS methods, with NFS_1^{abc} giving the lowest residuals and requiring the fewest iterations among the tested methods.

The numerical results reported in Table 8 and illustrated in Figure 7 show that all five iterative methods exhibit nearly complete convergence over the considered basin-computation domain. This is reflected by the Convergence Area Index, which is equal

to one for most methods and very close to one for $ECFS_3$. Moreover, all methods detect one root for the nonlinear problem (41). The average number of iterations is also nearly identical, indicating comparable convergence effort in this global basin-analysis setting. Small differences in CPU time are observed, with $ECFS_3$ requiring the lowest time in this particular computation, possibly due to implementation-dependent factors.

Table 8. Dynamical and computational indicators for the basin-of-attraction analysis of the five iterative methods. ANI: average number of iterations; CAI: Convergence Area Index; $W(F)$: 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

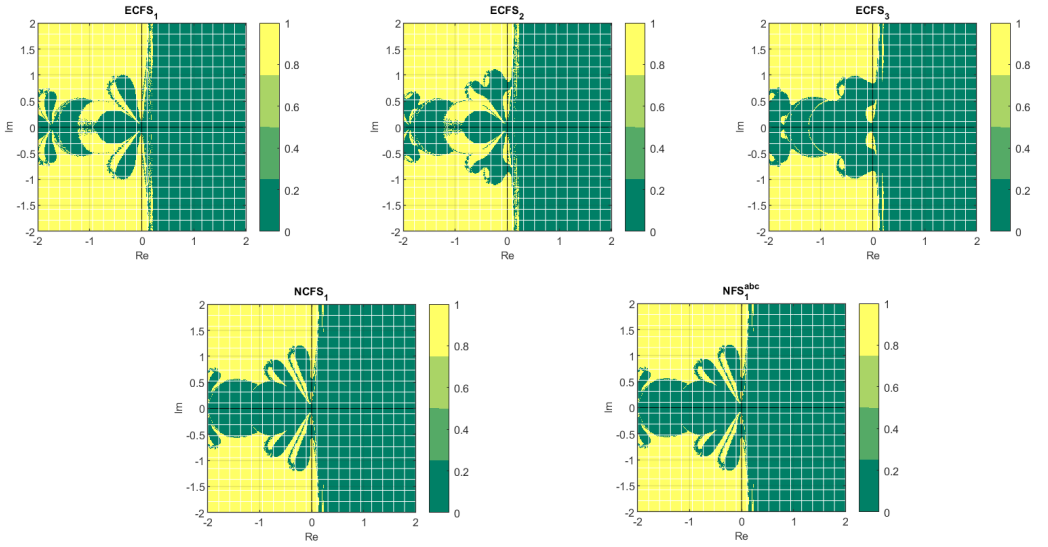


Figure 7. Basins of attraction associated with $ECFS_1-ECFS_3$, $NCFS_1$, and NFS_1^{abc} for solving Equation (41).

A notable feature of this test problem is that the Wada measure is equal to zero for all methods. This indicates that no Wada-type interlacing of three or more basins is detected in the tested domain. Since only one root is detected, the basin structure is correspondingly simpler than in the diode example, where several basins were present and nonzero Wada values were obtained. Thus, for the reactor model, the basin analysis suggests a regular global convergence pattern with low boundary complexity.

Overall, the basin-of-attraction results indicate that the considered methods exhibit regular convergence behavior for this test problem. In contrast with the previous diode example, the reactor model produces a dynamically simpler setting, where the main differences between the methods are not associated with basin complexity but rather with the local convergence indicators reported in Tables 5–7.

The results reported in Table 9 provide an additional comparative assessment of the computational cost and convergence behavior of the considered iterative schemes. In this setting, the proposed $NCFS_1$ and NFS_1^{abc} methods achieve a convergence rate of 100%, with no observed divergence. They also require lower operation counts and less memory than the ECFS variants, indicating a favorable computational profile in this basin-analysis experiment.

Table 9. Comparative computational and convergence indicators for the considered iterative schemes when solving (41).

Method	Resolution	Conv. Rate (%)	Div. Rate (%)	Exec. Time (s)	Op. Count	CPU Time (s)	Memory (MB)	Rank
$NCFS_1$	High	100	0	6.018	3600	5.72	480.0	1
NFS_1^{abc}	High	100	0	6.043	3680	5.86	488.0	2
$ECFS_1$	High	100	0	6.065	4300	5.89	560.0	3
$ECFS_2$	High	100	0	6.071	4320	5.93	562.0	4
$ECFS_3$	High	99.998	0.002	7.041	4500	5.97	578.0	5

In Table 9, *Resolution* denotes the basin grid density. *Conv. Rate* and *Div. Rate* denote the percentages of convergent and divergent initial points, respectively. *Exec. Time* is the total wall-clock time of the basin computation, *CPU Time* is the effective numerical processing time, and *Op. Count* represents the approximate number of floating-point operations. The overall rank summarizes the combined behavior in terms of convergence rate, operation count, CPU time, and memory usage.

The $ECFS_1$ and $ECFS_2$ methods also achieve full convergence, but with larger operation counts and higher memory usage. $ECFS_3$ shows a very small divergence rate of 0.002% and has the largest operation count and memory usage among the tested methods in this comparison. Thus, the comparison suggests that the proposed schemes provide a favorable balance between convergence reliability, operation count, CPU time, and memory usage for this basin-analysis experiment.

As observed in the basin analysis, the Wada measure is zero for all methods, indicating that the differences reported in Table 9 are not associated with increased basin-boundary complexity. Rather, they mainly reflect differences in computational and convergence indicators within a dynamically regular regime.

5. Conclusions and Future Work

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 solving nonlinear equations. The local convergence behavior of the Caputo-based scheme was investigated theoretically, while the numerical performance of both proposed methods was assessed through application-motivated nonlinear models and compared with three existing schemes, $ECFS_1$–$ECFS_3$.

The numerical results reported in Tables 1–9 and Figures 2–7 indicate that the proposed methods can provide improved computational performance in the considered test problems, as reflected by faster decay of the monitored convergence indicators, lower residuals, and reduced iteration counts compared with the ECFS schemes. The relative performance of $NCFS_1$ and NFS_1^{abc} depends on the fractional order and on the numerical setting. In the tests with $\gamma = 0.5, 0.7, 0.9$, $NCFS_1$ provides faster step-difference reduction than NFS_1^{abc}, whereas in the near-classical comparisons with $\gamma = 0.98, 0.99, 1.00$, NFS_1^{abc} gives the lowest residuals and requires the fewest iterations among the tested methods.

The basin-of-attraction analysis further illustrates the global dynamical behavior of the methods. For the diode equation, where nonzero Wada measures are observed, the proposed schemes generate more intricate basin boundaries, indicating stronger sensitivity of the basin geometry to the initial point. In contrast, for the reactor model, the Wada measure is zero for all methods, suggesting a more regular basin structure. In this case, the observed differences between the methods are mainly related to convergence and computational indicators rather than to basin-boundary complexity.

Overall, the proposed fractional schemes show a favorable balance between accuracy, convergence behavior, and computational cost for the nonlinear test problems considered

in this study. These results suggest that fractional iterative schemes can provide a useful framework for solving nonlinear equations arising in engineering applications.

Despite these advantages, the performance of the proposed methods depends on the choice of the initial approximation, the fractional order, and the parameters λ_1 and λ_2. Moreover, additional analysis is needed to clarify their behavior for large-scale problems, highly stiff nonlinear models, and multidimensional systems.

Open Problems and Future Work

Several directions remain open for further investigation:

- *Global convergence analysis:* although local convergence has been studied, deriving sufficient conditions for global convergence of fractional iterative schemes remains an important theoretical challenge.
- *Parameter selection:* theoretical and computational criteria for choosing the fractional order γ and the parameters λ_1, λ_2 should be further investigated.
- *Extension to systems of nonlinear equations:* generalizing the proposed methods to multidimensional problems requires an efficient treatment of Jacobian structures and fractional operators.
- *Efficient computation of fractional derivatives:* reducing the computational cost of fractional operators, especially in large-scale settings, remains an important practical issue.
- *Robustness under perturbations:* the stability of the proposed schemes in the presence of numerical noise, perturbed data, or inexact function evaluations deserves further analysis.

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.

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 resolution	–
C-Points	Number of converging points	–
D-Points	Number of diverging points	–
E-Time (S)	Elapsed time	seconds
Memory (MBs)	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. [CrossRef]
2. Budhiraja, A.; Chen, L.; Lee, C. A survey of numerical methods for nonlinear filtering problems. *Phys. D Nonlinear Phenom.* **2007**, *230*, 27–36. [CrossRef]
3. Ito, K.; Xiong, K. Gaussian filters for nonlinear filtering problems. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*; IEEE: NEW York, NY, USA, 2000; Volume 45, pp. 910–927. [CrossRef]
4. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: New York, NY, USA, 2006. [CrossRef]
5. Meza, J.C. Newton's method. *Wiley Interdiscip. Rev. Comput. Stat.* **2011**, *3*, 75–78. [CrossRef]
6. 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 (accessed on 14 May 2026).
7. Amat, S.; Ezquerro, J.A.; Hernández-Verón, M.A. On a Steffensen-like method for solving nonlinear equations. *Calcolo* **2016**, *53*, 171–188. [CrossRef]
8. Failla, G.; Zingales, M. Advanced materials modelling via fractional calculus: Challenges and perspectives. *Philos. Trans. R. Soc. A* **2020**, *378*, 20200050. [CrossRef]
9. 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. [CrossRef] [PubMed]
10. 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. [CrossRef]
11. 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. [CrossRef]
12. 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. [CrossRef]
13. 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. [CrossRef]
14. Candelario, G.; Cordero, A.; Torregrosa, J.R.; Vassileva, M.P. Generalized conformable fractional Newton-type method for solving nonlinear systems. *Numer. Algorithms* **2023**, *93*, 1171–1208. [CrossRef]
15. Bayrak, M.A.; Demir, A.; Ozbilge, E. On fractional Newton-type method for nonlinear problems. *J. Math.* **2022**, *2022*, 7070253. [CrossRef]
16. Wang, X.; Xu, J. Conformable vector Traub's method for solving nonlinear systems. *Numer. Algorithms* **2024**, *97*, 1563–1582. [CrossRef]
17. Erfanifar, R.; Hajarian, M.; Sayevand, K. A family of iterative methods to solve nonlinear problems with applications in fractional differential equations. *Math. Methods Appl. Sci.* **2024**, *47*, 2099–2119. https://onlinelibrary.wiley.com/doi/pdfdirect/10.1002/mma.9736. [CrossRef]
18. Akram, S.; Batool, R.; Akram, F. Development of two-step fractional iterative technique of convergence order $(2\mu+1)$ with applications. *Punjab Univ. J. Math.* **2025**, *57*, 504–533. [CrossRef]
19. Asghari, S.; Lotfi, T.; Moccari, M. Efficient conformable iterative approaches for solving nonlinear equations via comprehensive dynamical stability analysis. *J. Anal.* **2025**, *33*, 2061–2080. [CrossRef]
20. Shams, M.; Carpentieri, B. Stability Analysis of Fractional Order Schemes for Nonlinear Engineering Problems. In *Polynomials–Exploring Fundamental Mathematical Expressions*; IntechOpen: London, UK, 2025. [CrossRef]
21. Nayak, S.K.; Parida, P.K. Global convergence analysis of Caputo fractional Whittaker method with real world applications. *Cubo* **2024**, *26*, 167–190. [CrossRef]
22. Sweilam, N.H.; Al-Mekhlafi, S.M.; Assiri, T.; Atangana, A. Optimal control for cancer treatment mathematical model using Atangana–Baleanu–Caputo fractional derivative. *Adv. Differ. Equ.* **2020**, *2020*, 334. [CrossRef]
23. Syam, S.M.; Zain, M.; Alshateri, R.; Al-Dulaimi, A.; Syam, M.I. A modified operational matrix method for investigation of the nonlinear dynamics of fractional systems. *Fractals* **2025**, *33*, 2540155. [CrossRef]
24. Shams, M.; Carpentieri, B. A new high-order fractional parallel iterative scheme for solving nonlinear equations. *Symmetry* **2024**, *16*, 1452. [CrossRef]
25. 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. [CrossRef]
26. 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. [CrossRef]
27. 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. [CrossRef]

28. 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. [CrossRef]
29. Diethelm, K. *The Analysis of Fractional Differential Equations*; Springer: Berlin, Germany, 2010. [CrossRef]
30. Ziaukas, P.; Ragulskis, M. Fractal dimension and Wada measure revisited: No straightforward relationships in NDDS. *Nonlinear Dyn.* **2017**, *88*, 871–882. [CrossRef]
31. Toyonaga, T.; Wada, H. Origin of anticlastic curvature in a cellular metaplate. *Phys. Rev. Res.* **2024**, *6*, L032051. [CrossRef]
32. 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. [CrossRef]
33. 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. [CrossRef]
34. 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. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.