

Article

Not peer-reviewed version

Robust Fractional Quantum Two-Step Schemes with Enhanced Stability for Nonlinear Equations

[Mudassir Shams](#) and [Bruno Carpentieri](#)*

Posted Date: 26 March 2026

doi: 10.20944/preprints202603.2123.v1

Keywords: fractional quantum calculus; two-step iterative methods; nonlinear equations; dynamical systems analysis; convergence acceleration; engineering and biomedical applications



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

Robust Fractional Quantum Two-Step Schemes with Enhanced Stability for Nonlinear Equations

Mudassir Shams ^{1,2} and Bruno Carpentieri ^{3,*}

¹ Department of Mathematics, Faculty of Arts and Science, Balikesir University, 10145 Balikesir, Turkey

² Department of Mathematics and Statistics, Riphah International University I-14, Islamabad 44000, Pakistan

³ Faculty of Engineering, Free University of Bozen-Bolzano, 39100 Bolzano, Italy

* Correspondence: bruno.carpentieri@unibz.it

Abstract

Fractional quantum calculus provides a flexible mathematical framework for incorporating memory and scaling effects into numerical models. However, classical iterative methods for nonlinear equations often suffer from limited stability, strong dependence on initial guesses, and restricted convergence domains, particularly for highly nonlinear problems. In this work, we introduce a new Caputo fractional–quantum iterative scheme, denoted by $MSB_{q;\alpha}$, formulated as a parameterized two-step method based on a Caputo-type fractional quantum derivative. The proposed framework incorporates additional structural parameters that regulate the iterative dynamics and provide enhanced control over convergence behavior and stability properties. To investigate the performance of the proposed scheme, we employ tools from complex dynamical systems, including stability analysis and fractal basin investigations in the complex plane. These analyses illustrate how the fractional and quantum parameters influence the geometry of attraction domains and the global convergence behavior of the method. Numerical experiments on representative nonlinear test problems motivated by engineering and biomedical applications demonstrate improved robustness with respect to initial guesses, reduced residual errors, and competitive computational efficiency compared with classical iterative solvers. Overall, the results indicate that the proposed fractional–quantum framework provides an effective and flexible approach for the numerical solution of challenging nonlinear equations.

Keywords: fractional quantum calculus; two-step iterative methods; nonlinear equations; dynamical systems analysis; convergence acceleration; engineering and biomedical applications

1. Introduction

Nonlinear equations constitute a fundamental mathematical framework for modeling equilibrium states, steady configurations, and implicit relationships arising in science and engineering. Given a nonlinear operator

$$f(x) = 0, \quad f : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}, \quad (1)$$

the objective is to determine $x^* \in \Omega$ such that $f(x^*) = 0$. This abstract formulation arises in numerous contexts, including steady states of differential systems, discretized fractional partial differential equations, nonlinear inverse problems, and parameter identification models. In practice, the structural properties of f , such as smoothness, nonlocality, and memory dependence, strongly influence the performance and stability of numerical solvers designed to approximate its roots [1–3].

Classical calculus assumes locality: the derivative at a point depends solely on infinitesimal neighborhood information. However, many real-world systems exhibit hereditary effects, long-range interactions, and scaling symmetries that cannot be adequately represented within this local framework. Fractional calculus [4] generalizes differentiation to non-integer orders, thereby incorporating memory effects through integral operators. For example, the Caputo derivative of order $\alpha \in (0, 1)$ introduces nonlocal dependence through a power-law kernel, allowing past states of a system to influence its

present behavior. Such operators arise naturally in models of viscoelasticity, anomalous diffusion, bioheat transfer, and population dynamics with memory.

Parallel to fractional calculus, quantum calculus (or q -calculus) eliminates the classical limit process and replaces differentiation by finite scaling differences [5]. The Jackson q -derivative is defined as

$$D_q f(x) = \frac{f(qx) - f(x)}{(q-1)x}, \quad q \neq 1, \quad (2)$$

and captures discrete self-similarity and multiplicative scaling effects. Fractional quantum operators combine these two extensions, enabling the simultaneous representation of memory and scaling phenomena [6,7]. From a theoretical perspective, these operators enlarge the functional space in which nonlinear operators act and provide additional structural flexibility for the construction of iterative numerical schemes.

Root-finding methods are typically formulated as fixed-point iterations,

$$x_{k+1} = G(x_k), \quad (3)$$

where $G : \Omega \rightarrow \Omega$ satisfies $G(x^*) = x^*$. The convergence of (3) is governed by the spectral radius of the derivative,

$$\rho(G'(x^*)) < 1.$$

Among classical approaches, Newton's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (4)$$

is widely used due to its quadratic convergence under standard regularity assumptions. However, its theoretical guarantees are essentially local, and the size of the basin of attraction depends strongly on the nonlinear structure of f . For highly nonlinear or stiff operators, the associated iterative map may exhibit multiple attracting cycles, chaotic regions, or narrow attraction domains [8].

From the perspective of complex dynamics [9], iterative schemes generate intricate fractal basin structures in the complex plane. Such analyses provide valuable insight into the global stability of iterative schemes beyond classical convergence order. Consequently, the convergence behavior of a method depends not only on its local order but also on the global dynamics of the iteration map G . This observation motivates the development of iterative frameworks with additional structural parameters capable of reshaping attraction domains and improving global convergence behavior. In this context, fractional and quantum calculus provide promising tools for enhancing both the flexibility and stability of iterative numerical schemes.

Building on these observations, several studies have extended classical iterative methods by incorporating fractional and q -calculus operators in order to improve convergence, stability, and global dynamical behavior. For instance, Ali *et al.* investigate iterative schemes based on fractional derivatives and report improved efficiency compared with classical root-finding methods [10]. Similarly, Sana *et al.* develop q -iterative techniques derived from q -Taylor expansions and coupled system formulations, demonstrating enhanced convergence properties [11]. Other q -analogue approaches have also been proposed, achieving higher convergence rates under suitable choices of the parameter q [12,13]. Moreover, multi-step and parallel q -fractional algorithms have been introduced to approximate multiple roots simultaneously while providing insights into the global convergence behavior of the associated iterative maps [14].

Despite these advances, a unified framework that systematically integrates fractional and quantum operators in order to control the geometry of attraction domains and the dynamical properties of iterative schemes remains largely unexplored. More specifically, while fractional Newton-type methods [15,16] and q -deformed schemes [17] have been investigated separately in the literature, several theoretical and computational questions remain open. In particular, the unified integration of

fractional and quantum operators within a single iterative framework—and the resulting impact on convergence and stability properties—has not yet been systematically studied.

- The absence of a unified framework combining Caputo-type memory operators with quantum scaling derivatives within multi-step iterative structures.
- Limited theoretical understanding of how fractional and quantum parameters influence the dynamical system generated by the iteration map.
- Insufficient exploration of parameterized iterative families as mechanisms for controlling stability regions and convergence behavior.
- A lack of systematic investigation of the geometry of attraction basins under fractional–quantum perturbations.

Existing approaches [18–20] and references therein typically treat the fractional order α and the quantum parameter q as fixed constants, rather than as intrinsic regulators of the convergence dynamics. In contrast, we introduce a parameterized family of two-step fractional quantum iterative schemes in which these quantities, together with additional structural parameters, play an active role in shaping the iterative process. The proposed framework combines fractional and quantum operators within a flexible two-step structure that enables improved control of accuracy, stability, and convergence behavior.

The resulting formulation defines a family of iterative maps whose fixed points correspond to solutions of the nonlinear equation. This perspective provides a natural and rigorous basis for both convergence analysis and the investigation of fractal basin structures associated with the dynamical behavior of the proposed schemes. The proposed framework therefore provides a new perspective for the design and analysis of fractional–quantum iterative methods. The principal theoretical contributions of this work can be summarized as follows:

1. Formulation of a unified fractional–quantum operator suitable for the construction of iterative root-finding schemes.
2. Development of a parameterized two-step iterative framework that provides additional structural flexibility in controlling convergence behavior.
3. Derivation of the local convergence properties through a perturbation analysis of the classical Taylor expansion in the presence of fractional operators.
4. Dynamical systems analysis of the proposed schemes, including the study of stability and basin geometry in the complex plane.
5. Identification of fractional and quantum parameters as geometric regulators capable of modifying the structure of attraction domains.

The novelty of this study lies in interpreting fractional and quantum operators not merely as modeling tools, but as intrinsic components of iterative dynamics. By embedding Caputo-type memory effects and q -scaling operators directly into the iterative structure, we construct a dynamical framework whose convergence behavior can be continuously tuned through the parameters $(\alpha, q, \beta_1, \beta_2)$, where β_1 and β_2 are additional structural parameters utilized by the proposed schemes.

To the best of our knowledge, the systematic combination of the following elements has not yet been explored within a unified framework for solving nonlinear equations:

- Caputo-type fractional derivatives,
- quantum (q -) difference operators,
- parameterized two-step iterative schemes,
- and complex fractal basin analysis.

The paper is organized as follows. Section 2 introduces the fractional quantum operators and establishes their main analytical properties. Section 3 develops the proposed family of two-step iterative schemes and presents the corresponding convergence analysis. Section 4 investigates the stability and dynamical behavior of the methods, including the analysis of fractal basin structures.

Section 5 provides numerical experiments on nonlinear problems arising in engineering and biomedical applications. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Preliminaries

In this section, we recall the main concepts of q -calculus and fractional q -calculus that form the analytical foundation for the fractional q -iterative schemes developed later in the paper. Throughout the paper, we assume $q \in (0, 1)$ and consider $0 < a < x < b$. We denote by $\mathbb{N} = \{1, 2, \dots\}$ the set of positive integers and by $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ the set of non-negative integers.

Basic Elements of q -Calculus

The concept of q -analogues extends classical algebraic quantities by introducing a deformation parameter q . These generalized quantities reduce to their classical counterparts in the limit $q \rightarrow 1^-$.

Definition 1 (q -Number and q -Factorial [21]). *The q -number associated with $r \in \mathbb{R}$ is defined by*

$$[r]_q = \frac{1 - q^r}{1 - q}. \quad (5)$$

For $n \in \mathbb{N}$, the q -factorial is defined recursively as

$$[n]_q! = \prod_{k=1}^n [k]_q, \quad [0]_q! := 1. \quad (6)$$

The q -shifted power plays a fundamental role in the construction of fractional kernels and related operators.

Definition 2 (q -Shifted Power [22]). *For $a, b \in \mathbb{R}$ and $n \in \mathbb{N}$,*

$$(a - b)_q^n = \prod_{j=0}^{n-1} (a - bq^j). \quad (7)$$

More generally, for $\theta \in \mathbb{R}$,

$$(a - b)_q^\theta = a^\theta \prod_{j=0}^{\infty} \frac{a - bq^j}{a - bq^{j+\theta}}, \quad (8)$$

whenever the infinite product converges.

The central operator in quantum calculus is the Jackson q -derivative, which replaces the classical limit process by multiplicative scaling.

Definition 3 (Jackson q -Derivative [23]). *Let $f : [0, b] \rightarrow \mathbb{R}$. The Jackson q -derivative is defined by*

$$(D_q f)(x) = \frac{f(qx) - f(x)}{(q - 1)x}, \quad x \neq 0, \quad (9)$$

with

$$(D_q f)(0) = \lim_{x \rightarrow 0} (D_q f)(x), \quad (10)$$

provided the limit exists. Higher-order derivatives are defined recursively as

$$D_q^n f := D_q(D_q^{n-1} f), \quad n \in \mathbb{N}. \quad (11)$$

Jackson Integral and q -Gamma Function

The integral counterpart of the Jackson q -derivative is the Jackson q -integral, which restores an inversion relationship analogous to the fundamental theorem of classical calculus.

Definition 4 (Jackson q -Integral [24]). *The Jackson q -integral over $[0, x]$ is defined by*

$$\int_0^x f(t) d_q t = (1 - q)x \sum_{k=0}^{\infty} q^k f(xq^k), \quad (12)$$

whenever the series converges.

The following identity expresses the inverse relation between the q -derivative and the Jackson integral.

Proposition 2.1 (Fundamental Theorem of q -Calculus [25]).

If f is continuous at 0, then

$$D_q \left(\int_0^x f(t) d_q t \right) = f(x). \quad (13)$$

To extend q -calculus operators to fractional orders, the q -Gamma function is introduced as a normalization factor.

Definition 5 (q -Gamma Function [26]). *For $\alpha > 0$, the q -Gamma function is defined by*

$$\Gamma_q(\alpha) = (1 - q)^{1-\alpha} \frac{(q; q)_{\infty}}{(q^{\alpha}; q)_{\infty}}, \quad (14)$$

where

$$(a; q)_{\infty} = \prod_{k=0}^{\infty} (1 - aq^k). \quad (15)$$

It satisfies the functional relation

$$\Gamma_q(\alpha + 1) = [\alpha]_q \Gamma_q(\alpha). \quad (16)$$

Fractional q -Operators

Fractional q -operators extend the concepts of q -calculus to non-integer orders and provide the analytical framework required for the construction of fractional q -iterative schemes.

Definition 6 (Riemann–Liouville Fractional q -Integral [27]). *For $\alpha > 0$, the Riemann–Liouville fractional q -integral of order α is defined as*

$$(I_{q,a}^{[\alpha]} f)(x) = \frac{1}{\Gamma_q(\alpha)} \int_a^x (x - t)_{q^{\alpha-1}} f(t) d_q t. \quad (17)$$

Based on this operator, the corresponding Riemann–Liouville fractional q -derivative is defined as follows.

Definition 7 (Riemann–Liouville Fractional q -Derivative [28]). *Let $\alpha > 0$ and $n = [\alpha]$. Then*

$$(D_{q,a}^{[\alpha]} f)(x) = D_q^n (I_{q,a}^{n-\alpha} f)(x). \quad (18)$$

An alternative formulation frequently used in applications is the Caputo-type fractional q -derivative, which differs from the Riemann–Liouville derivative in the order of differentiation and integration.

Definition 8 (Caputo Fractional q -Derivative [29]). Let $\alpha > 0$ and $n = \lceil \alpha \rceil$. Then

$$({}^C D_{q,a}^{[\alpha]} f)(x) = I_{q,a}^{n-\alpha} (D_q^n f)(x). \quad (19)$$

Key Properties and Fractional q -Taylor Formula

The following identities describe the interaction between classical q -derivatives and fractional q -operators introduced in the previous subsection.

Proposition 2.2.

For $\alpha \in \mathbb{R}^+ \setminus \mathbb{N}$ and $0 < a < x < b$,

$${}^C D_{q,a}^{[\alpha]} (D_q f)(x) = {}^C D_{q,a}^{[\alpha+1]} f(x), \quad (20)$$

and

$$D_q ({}^C D_{q,a}^{[\alpha]} f)(x) = {}^C D_{q,a}^{[\alpha+1]} f(x) + \frac{D_q^{[\alpha]} f(a)}{\Gamma_q([\alpha] - \alpha)} (x - a)_q^{[\alpha] - \alpha - 1}. \quad (21)$$

The fractional q -Taylor expansion provides the fundamental analytical tool for deriving local error equations in fractional q -iterative methods.

Theorem 2.1 (Fractional q -Taylor Expansion [30]). Let f be sufficiently smooth and $\alpha > 0$. Then

$$f(x) = \sum_{k=0}^{n-1} \frac{(x-a)_q^k}{[k]_q!} D_q^k f(a) + \frac{1}{\Gamma_q(\alpha)} \int_a^x (x-t)_q^{\alpha-1} ({}^C D_{q,a}^\alpha f)(t) d_q t, \quad (22)$$

where $n = \lceil \alpha \rceil$.

3. Development and Analysis of the q -Fractional Scheme

3.1. Proposed Fractional Weighted Iterative Method

Let ξ be a simple root of $f : \mathbb{R} \rightarrow \mathbb{R}$, and define the error at the k -th iteration by

$$e_k := x_k - \xi.$$

We now introduce the proposed two-step weighted iterative method, abbreviated as $MSB_{q,\alpha}$, for the solution of nonlinear equations:

$$\begin{cases} y_k = x_k - \left(\Gamma_q(\alpha + 1) \frac{f(x_k)}{[{}^C D_q^{[\alpha]} f](x_k)} \left(\frac{1}{1 - \frac{\beta_1 f(x_k)}{1 + \beta_2 f(x_k)}} - f(x_k) \right) \right)^{\frac{1}{\alpha}}, \\ x_{k+1} = y_k - \left(\Gamma_q(\alpha + 1) \frac{f(x_k)}{[{}^C D_q^{[\alpha]} f](x_k)} \left(\frac{f(y_k)}{f(x_k)} + 2 \left(\frac{f(y_k)}{f(x_k)} \right)^2 \right) \right)^{\frac{1}{\alpha}}. \end{cases} \quad (23)$$

Here, $\beta_1, \beta_2 \in \mathbb{R}$ are free parameters that regulate the convergence behavior of the scheme.

3.2. Convergence Theorem

The following theorem establishes the convergence order of the proposed iterative scheme.

Theorem 3.1. Let f be sufficiently smooth in a neighborhood of the simple root ξ , and let the initial approximation x_0 be sufficiently close to ξ . Then the sequence $\{x_k\}$ generated by the iterative scheme (23) satisfies

$$e_{k+1} = (\mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3 + \mathcal{D}_4 + \mathcal{D}_5) (e_k^{3\alpha+1}; \mathfrak{q}) + O(e_k^{4\alpha+1}; \mathfrak{q}), \quad (24)$$

where the coefficients $\mathcal{D}_1, \dots, \mathcal{D}_5$ depend on the Taylor coefficients of f at ξ and on the parameters α , β_1 , and β_2 . Consequently, the proposed method has convergence order $3\alpha + 1$.

Proof. The proof is based on a series expansion of f around the simple root ξ in terms of the error e_k . We write the expansion in a fully normalized form by factoring out the leading term:

$$f(x_k) = \frac{{}^C D_{\mathfrak{q}} f(\xi)}{\Gamma_{\mathfrak{q}}(\alpha + 1)} \left[(e_k^{\alpha}; \mathfrak{q}) + c_2 (e_k^{2\alpha}; \mathfrak{q}) + c_3 (e_k^{3\alpha}; \mathfrak{q}) + c_4 (e_k^{4\alpha}; \mathfrak{q}) + O(e_k^{5\alpha}; \mathfrak{q}) \right], \quad (25)$$

where

$$c_j = \frac{\Gamma_{\mathfrak{q}}(\alpha + 1)}{\Gamma_{\mathfrak{q}}(j\alpha + 1)} \frac{[{}^C D_{\mathfrak{q}}^{[j]} f](\xi)}{j! [{}^C D_{\mathfrak{q}}^{[1]} f](\xi)}, \quad j \geq 2.$$

Next, we expand the Caputo \mathfrak{q} -derivative at the iterate x_k .

$$\begin{aligned} [{}^C D_{\mathfrak{q}} f](x_k) &= \frac{[{}^C D_{\mathfrak{q}}^{[1]} f](\xi)}{\Gamma_{\mathfrak{q}}(\alpha + 1)} \left[\Gamma_{\mathfrak{q}}(\alpha + 1) + \frac{\Gamma_{\mathfrak{q}}(2\alpha + 1)c_2 (e_k^{\alpha}; \mathfrak{q})}{\Gamma_{\mathfrak{q}}(\alpha + 1)} \right. \\ &\quad \left. + \frac{\Gamma_{\mathfrak{q}}(3\alpha + 1)c_3 (e_k^{2\alpha}; \mathfrak{q})}{\Gamma_{\mathfrak{q}}(2\alpha + 1)} + O(e_k^{3\alpha}; \mathfrak{q}) \right]. \end{aligned} \quad (26)$$

Next, the reciprocal of the derivative is expanded as a power series in terms of the error e_k :

$$\begin{aligned} \frac{1}{({}^C D_{\mathfrak{q}} f)(x_k)} &= \frac{1}{\Gamma_{\mathfrak{q}}(\alpha + 1)} - \frac{\Gamma_{\mathfrak{q}}(2\alpha + 1)c_2 (e_k^{\alpha}; \mathfrak{q})}{\Gamma_{\mathfrak{q}}(\alpha + 1)^3} \\ &\quad + \frac{(e_k^{2\alpha}; \mathfrak{q})}{\Gamma_{\mathfrak{q}}(\alpha + 1)} \left(-\frac{\Gamma_{\mathfrak{q}}(3\alpha + 1)c_3}{\Gamma_{\mathfrak{q}}(\alpha + 1)\Gamma_{\mathfrak{q}}(2\alpha + 1)} + \frac{\Gamma_{\mathfrak{q}}(2\alpha + 1)^2 c_2^2}{\Gamma_{\mathfrak{q}}(\alpha + 1)^4} \right) \\ &\quad + \frac{(e_k^{3\alpha}; \mathfrak{q})}{\Gamma_{\mathfrak{q}}(\alpha + 1)} \left(\frac{c_2 \Gamma_{\mathfrak{q}}(3\alpha + 1)c_3}{\Gamma_{\mathfrak{q}}(\alpha + 1)^3} - \frac{\mathcal{C}_1}{\Gamma_{\mathfrak{q}}(\alpha + 1)^6} \right) + O(e_k^{4\alpha}; \mathfrak{q}). \end{aligned} \quad (27)$$

where $\mathcal{C}_1 = ((\Gamma_{\mathfrak{q}}(2\alpha + 1))^3 c_2^2 - \Gamma_{\mathfrak{q}}(3\alpha + 1)c_3(\Gamma_{\mathfrak{q}}(\alpha + 1))^3) c_2$. Using the previous expansion, the Newton-type correction term can be written as

$$\begin{aligned} \frac{f(x_k)}{({}^C D_{\mathfrak{q}} f)(x_k)} &= \frac{(e_k^{\alpha}; \mathfrak{q})}{\Gamma_{\mathfrak{q}}(\alpha + 1)} \\ &\quad + \left(\frac{c_2}{\Gamma_{\mathfrak{q}}(\alpha + 1)} - \frac{2^{2\alpha} c_2}{\sqrt{\pi} \Gamma_{\mathfrak{q}}(\alpha + 1)^2} \Gamma_{\mathfrak{q}}\left(\alpha + \frac{1}{2}\right) \right) (e_k^{2\alpha}; \mathfrak{q}) \\ &\quad + \left(\frac{c_3}{\Gamma_{\mathfrak{q}}(\alpha + 1)} - \frac{2^{2\alpha} c_2^2}{\sqrt{\pi} \Gamma_{\mathfrak{q}}(\alpha + 1)^2} \Gamma_{\mathfrak{q}}\left(\alpha + \frac{1}{2}\right) \right. \\ &\quad \left. + \frac{2^{4\alpha} c_2^2}{\pi \Gamma_{\mathfrak{q}}(\alpha + 1)^3} \Gamma_{\mathfrak{q}}\left(\alpha + \frac{1}{2}\right)^2 \right. \\ &\quad \left. - \frac{3^{3\alpha} \sqrt{3}}{2\sqrt{\pi} 2^{2\alpha} \Gamma_{\mathfrak{q}}(\alpha + 1)^2} \Gamma_{\mathfrak{q}}\left(\alpha + \frac{1}{3}\right) \Gamma_{\mathfrak{q}}\left(\alpha + \frac{2}{3}\right) \right) (e_k^{3\alpha}; \mathfrak{q}) \\ &\quad + O(e_k^{4\alpha}; \mathfrak{q}). \end{aligned} \quad (28)$$

Intermediate iteration y_k . We now expand the first step of the proposed method.

$$\begin{aligned} \left(1 - \frac{\beta_1 f(x_k)}{1 + \beta_2 f(x_k)}\right)^{-1} &= 1 + \beta_1 (e_k^\alpha; \mathfrak{q}) \\ &\quad + (\beta_1^2 - \beta_1 \beta_2 + \beta_1 c_2) (e_k^{2\alpha}; \mathfrak{q}) \\ &\quad + \left(\beta_1 \beta_2^2 - 2\beta_1 \beta_2 c_2 + \beta_1 c_3 - (\beta_1^2 - \beta_1 \beta_2 + \beta_1 c_2) \beta_1\right) (e_k^{3\alpha}; \mathfrak{q}) \\ &\quad + O(e_k^{4\alpha}; \mathfrak{q}). \end{aligned} \quad (29)$$

From this expansion we obtain

$$\begin{aligned} G_k &= 1 + (\beta_1 - 1) (e_k^\alpha; \mathfrak{q}) + (\beta_1^2 - \beta_1 \beta_2 + \beta_1 c_2 - c_2) (e_k^{2\alpha}; \mathfrak{q}) \\ &\quad + (\beta_1^3 - 2\beta_1^2 \beta_2 + 2\beta_1^2 c_2 + \beta_1 \beta_2^2 - 2\beta_1 \beta_2 c_2 + \beta_1 c_3 - c_3) (e_k^{3\alpha}; \mathfrak{q}) \\ &\quad + O(e_k^{4\alpha}; \mathfrak{q}), \end{aligned} \quad (30)$$

where

$$G_k = \left(\left(1 - \frac{\beta_1 f(x_k)}{1 + \beta_2 f(x_k)}\right)^{-1} - f(x_k) \right).$$

For convenience, we introduce the notation

$$F_k := \frac{f(x_k)}{({}^C D_{\mathfrak{q}} f)(x_k)} \left(\left(1 - \frac{\beta_1 f(x_k)}{1 + \beta_2 f(x_k)}\right)^{-1} - f(x_k) \right), \quad (31)$$

For notational convenience, we define

$$G_x := \Gamma_{\mathfrak{q}}(x),$$

and this convention is used consistently throughout the manuscript. In particular,

$$\begin{aligned} G_{\alpha+1} &= \Gamma_{\mathfrak{q}}(\alpha + 1), & G_{\alpha+\frac{1}{2}} &= \Gamma_{\mathfrak{q}}\left(\alpha + \frac{1}{2}\right), \\ G_{\alpha+\frac{1}{3}} &= \Gamma_{\mathfrak{q}}\left(\alpha + \frac{1}{3}\right), & G_{\alpha+\frac{2}{3}} &= \Gamma_{\mathfrak{q}}\left(\alpha + \frac{2}{3}\right). \end{aligned} \quad (32)$$

Substituting the previous expansions yields

$$\begin{aligned} F_k &= \frac{(e_k^\alpha; \mathfrak{q})}{G_{\alpha+1}} + \left(\frac{\beta_1 + c_2 - 1}{G_{\alpha+1}} - \frac{2^{2\alpha} c_2}{\sqrt{\pi} G_{\alpha+1}^2} G_{\alpha+1/2} \right) (e_k^{2\alpha}; \mathfrak{q}) \\ &\quad + \left(-\frac{2^{2\alpha} c_2 \beta_1}{\sqrt{\pi} G_{\alpha+1}^2} G_{\alpha+1/2} - \frac{3^{3\alpha} \sqrt{3} c_3}{2\sqrt{\pi} 2^{2\alpha} G_{\alpha+1}^2} \frac{G_{\alpha+1/3} G_{\alpha+2/3}}{G_{\alpha+1/2}} \right. \\ &\quad \left. - \frac{2c_2}{G_{\alpha+1}} + \frac{\beta_1^2 + c_3}{G_{\alpha+1}} - \frac{2^{2\alpha} c_2^2}{\alpha^2 \sqrt{\pi} G_{\alpha+1}^2} G_{\alpha+1/2} \right. \\ &\quad \left. + \frac{2^{4\alpha} c_2^2}{\alpha^3 \pi G_{\alpha+1}^3} G_{\alpha+1/2}^2 + \frac{2^{2\alpha} c_2}{\sqrt{\pi} G_{\alpha+1}^2} G_{\alpha+1/2} + \frac{2\beta_1 c_2 - \beta_1 \beta_2}{G_{\alpha+1}} \right) (e_k^{3\alpha}; \mathfrak{q}) \\ &\quad + O(e_k^{4\alpha}; \mathfrak{q}). \end{aligned} \quad (33)$$

From the previous expansion, we obtain

$$\begin{aligned}
y_k - \xi &= x_k - \xi - \left(G_{\alpha+1} F_k \right)^{\frac{1}{\alpha}} = \left(-\beta_1 - c_2 + 1 + \frac{2^{2\alpha} c_2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} \right) \left(e_k^{\alpha+1}; q \right) \\
&+ \left(\frac{2^{2\alpha} c_2 \beta_1}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} - 2\beta_1 c_2 + \frac{3^{3\alpha} \sqrt{3} c_3}{2 G_{\alpha+1} \sqrt{\pi} 2^{2\alpha}} \frac{G_{\alpha+1/3} G_{\alpha+2/3}}{G_{\alpha+1/2}} \right. \\
&+ 2c_2 - \beta_1^2 - c_3 + \beta_1 \beta_2 + \frac{2^{2\alpha} c_2^2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} - \frac{2^{4\alpha} c_2^2}{G_{\alpha+1}^2 \pi} G_{\alpha+1/2}^2 \\
&\left. - \frac{2^{2\alpha} c_2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} \right) \left(e_k^{2\alpha+1}; q \right) + O\left(e_k^{3\alpha+1}; q \right). \tag{34}
\end{aligned}$$

Hence,

$$\begin{aligned}
y_k - \xi &= \left(-\beta_1 - c_2 + 1 + \frac{2^{2\alpha} c_2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} \right) \left(e_k^{\alpha+1}; q \right) \\
&+ \left(\frac{2^{2\alpha} c_2 \beta_1}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} - 2\beta_1 c_2 + \frac{3^{3\alpha} \sqrt{3} c_3}{2 G_{\alpha+1} \sqrt{\pi} 2^{2\alpha}} \frac{G_{\alpha+1/3} G_{\alpha+2/3}}{G_{\alpha+1/2}} \right. \\
&+ 2c_2 - \beta_1^2 - c_3 + \beta_1 \beta_2 + \frac{2^{2\alpha} c_2^2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} - \frac{2^{4\alpha} c_2^2}{G_{\alpha+1}^2 \pi} G_{\alpha+1/2}^2 \\
&\left. - \frac{2^{2\alpha} c_2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} \right) \left(e_k^{2\alpha+1}; q \right) + O\left(e_k^{3\alpha+1}; q \right). \tag{35}
\end{aligned}$$

Substituting this expression into $f(y_k)$ yields

$$\begin{aligned}
f(y_k) &= -\frac{\left(e_k^{\alpha+1}; q \right)}{G_{\alpha+1} \sqrt{\pi}} \left((\beta_1 + c_2 - 1) G_{\alpha+1} \sqrt{\pi} - 2^{2\alpha} G_{\alpha+1/2} c_2 \right) \\
&+ \frac{\left(e_k^{2\alpha+1}; q \right)}{2 G_{\alpha+1}^2 \pi^{3/2} 2^{2\alpha}} \left[-2 \cdot 2^{6\alpha} G_{\alpha+1/2}^3 c_2^2 \sqrt{\pi} + 2 \cdot 2^{4\alpha} G_{\alpha+1/2}^2 G_{\alpha+1} \pi (c_2 \beta_1 + c_2^2 - c_2) \right. \\
&+ c_3 3^{3\alpha} \sqrt{3} G_{\alpha+1/3} G_{\alpha+2/3} G_{\alpha+1} \pi + 2 G_{\alpha+1/2} \cdot 2^{2\alpha} G_{\alpha+1}^2 \pi^{3/2} \\
&\left. (\beta_1 \beta_2 - \beta_1^2 - 2\beta_1 c_2 + 2c_2 - c_3) \right] G_{\alpha+1/2}^{-1} + O\left(e_k^{3\alpha+1}; q \right). \tag{36}
\end{aligned}$$

Consequently,

$$\frac{f(y_k)}{f(x_k)} = \mathcal{A}_1 \left(e_k^{2\alpha+1}; q \right) + \mathcal{A}_2 \left(e_k^{3\alpha+1}; q \right) + O\left(e_k^{4\alpha+1}; q \right). \tag{37}$$

The coefficients \mathcal{A}_i and \mathcal{D}_i , for $i = 1, \dots, 5$, were derived and verified using symbolic computation in *Maple 2020*, ensuring their correctness. In particular, the coefficients \mathcal{A}_1 and \mathcal{A}_2 are given by

$$\begin{aligned}
\mathcal{A}_1 &= 2\beta_1^2 \beta_2 - \beta_1 \beta_2^2 - 2\beta_1 c_3 + \frac{3^{3\alpha} \sqrt{3} \beta_1 c_3}{2 G_{\alpha+1} \sqrt{\pi} 2^{2\alpha}} \frac{G_{\alpha+1/3} G_{\alpha+2/3}}{G_{\alpha+1/2}} + \frac{3^{3\alpha} \sqrt{3} c_2 c_3}{2 G_{\alpha+1} \sqrt{\pi} 2^{2\alpha}} \frac{G_{\alpha+1/3} G_{\alpha+2/3}}{G_{\alpha+1/2}} \\
&+ \frac{2^{2\alpha} c_2 c_3}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} + \frac{2^{2\alpha} c_2 \beta_1^2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} - \frac{2^{4\alpha} c_2^2 \beta_1}{G_{\alpha+1}^2 2\pi} G_{\alpha+1/2}^2 - 2\beta_1^2 c_2 + c_2^2 \beta_1 - \frac{2^{2\alpha} c_2 \beta_1 \beta_2}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} \\
&- \frac{3^{3\alpha} \sqrt{3} c_3}{2 G_{\alpha+1} \sqrt{\pi} 2^{2\alpha}} \frac{G_{\alpha+1/3} G_{\alpha+2/3}}{G_{\alpha+1/2}} + 3\beta_1 \beta_2 c_2 - 2\beta_1 c_2 - \frac{3^{3\alpha} \sqrt{3} c_2 c_3}{G_{\alpha+1}^2 \pi} G_{\alpha+1/3} G_{\alpha+2/3} - c_2^2 - \beta_1^3 \\
&+ \frac{2^{4\alpha} c_2^2}{G_{\alpha+1}^2 \pi} G_{\alpha+1/2}^2 - \frac{2^{2\alpha+1} c_2^3}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} + \frac{2^{6\alpha} c_2^3}{G_{\alpha+1}^3 \pi^{3/2}} G_{\alpha+1/2}^3.
\end{aligned}$$

$$\begin{aligned}
\mathcal{A}_2 = & -\frac{3^{3\alpha}\sqrt{3}c_2^2c_3}{2G_{\alpha+1}\sqrt{\pi}2^{2\alpha}}\frac{G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1/2}} + \frac{3^{3\alpha}\sqrt{3}c_2c_3}{2G_{\alpha+1}\sqrt{\pi}2^{2\alpha}}\frac{G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1/2}} - \frac{3^{3\alpha}\sqrt{3}\beta_1c_2c_3}{2G_{\alpha+1}\sqrt{\pi}2^{2\alpha}}\frac{G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1/2}} \\
& - \frac{2^{2\alpha}c_2^2\beta_1^2}{G_{\alpha+1}\sqrt{\pi}}G_{\alpha+1/2} - 2c_2c_3 + \frac{2^{4\alpha}c_2^3\beta_1}{G_{\alpha+1}^2\pi}G_{\alpha+1/2}^2 - \frac{2^{2\alpha}c_2^2c_3}{G_{\alpha+1}\sqrt{\pi}}G_{\alpha+1/2} \\
& + \beta_1^3c_2 + c_2c_4 + 2\beta_1^2c_2^2 - c_2^3\beta_1 + 2\beta_1c_2^2 + \frac{2^{2\alpha}c_2^2\beta_1\beta_2}{G_{\alpha+1}\sqrt{\pi}}G_{\alpha+1/2} \\
& + 2\beta_1c_2c_3 - 2\beta_1^2\beta_2c_2 - 3\beta_1\beta_2c_2^2 + \beta_1\beta_2^2c_2 + \frac{3^{3\alpha}\sqrt{3}c_2^2c_3}{G_{\alpha+1}^2\pi}G_{\alpha+1/3}G_{\alpha+2/3} \\
& - c_2^4 - c_2^2 + c_2^3 - \frac{2^{6\alpha}c_2^4}{G_{\alpha+1}^3\pi^{3/2}}G_{\alpha+1/2}^3 - \frac{2^{4\alpha}c_2^3}{G_{\alpha+1}^2\pi}G_{\alpha+1/2}^2 + \frac{2^{2\alpha+1}c_2^4}{G_{\alpha+1}\sqrt{\pi}}G_{\alpha+1/2}.
\end{aligned}$$

Next, substituting the expansion of $\frac{f(y_k)}{f(x_k)}$ into the second correction term gives

$$\left(\frac{f(y_k)}{f(x_k)} + 2\left(\frac{f(y_k)}{f(x_k)}\right)^2\right) = \mathcal{A}_1(e_k^{2\alpha+1}; q) + \mathcal{A}_2(e_k^{3\alpha+1}; q) + \dots \quad (38)$$

Accordingly, we define

$$F_k^* = \frac{f(x_k)}{({}^C D_q f)(x_k)} \left(\frac{f(y_k)}{f(x_k)} + 2\left(\frac{f(y_k)}{f(x_k)}\right)^2\right),$$

and obtain the expansion

$$F_k^* = 2\mathcal{A}_3(e_k^{\alpha+1}; q) + 2\mathcal{A}_4(e_k^{2\alpha+1}; q) + \mathcal{A}_5(e_k^{3\alpha+1}; q) + O(e_k^{4\alpha+1}; q). \quad (39)$$

The coefficients \mathcal{A}_3 , \mathcal{A}_4 , and \mathcal{A}_5 are given by

$$\mathcal{A}_3 = -\beta_1 - c_2 + 1 + \frac{2^{2\alpha}G_{\alpha+1/2}c_2}{G_{\alpha+1}\sqrt{\pi}},$$

$$\begin{aligned}
\mathcal{A}_4 = & \frac{2^{2\alpha}G_{\alpha+1/2}c_2\beta_1}{G_{\alpha+1}\sqrt{\pi}} - 2\beta_1c_2 + \frac{1}{2}\frac{c_33^{3\alpha}\sqrt{3}G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1}\sqrt{\pi}2^{2\alpha}G_{\alpha+1/2}} \\
& + 2c_2 - \beta_1^2 - c_3 + \beta_1\beta_2 + \frac{2^{2\alpha}G_{\alpha+1/2}c_2^2}{G_{\alpha+1}\sqrt{\pi}} - \frac{2^{4\alpha}G_{\alpha+1/2}^2c_2^2}{G_{\alpha+1}^2\pi} - \frac{2^{2\alpha}G_{\alpha+1/2}c_2}{G_{\alpha+1}\sqrt{\pi}}.
\end{aligned}$$

$$\begin{aligned}
\mathcal{A}_5 = & \frac{2^{2\alpha}c_2c_3}{\sqrt{\pi}G_{\alpha+1}^2}G_{\alpha+1/2} - \frac{2^{4\alpha}c_2^2\beta_1}{G_{\alpha+1}^3\pi}G_{\alpha+1/2}^2 - \frac{\beta_1\beta_2^2}{G_{\alpha+1}} + 2\frac{\beta_1^2\beta_2}{G_{\alpha+1}} - 2\frac{\beta_1^2c_2}{G_{\alpha+1}} + \frac{c_2^2\beta_1}{G_{\alpha+1}} \\
& + \frac{2^{2\alpha}c_2\beta_1^2}{\sqrt{\pi}G_{\alpha+1}^2}G_{\alpha+1/2} - 2\frac{\beta_1c_2}{G_{\alpha+1}} - 2\frac{\beta_1c_3}{G_{\alpha+1}} - \frac{3^{3\alpha}\sqrt{3}c_3}{2\sqrt{\pi}G_{\alpha+1}^2}2^{2\alpha}\frac{G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1/2}} \\
& + \frac{3^{3\alpha}\sqrt{3}c_2c_3}{2\sqrt{\pi}G_{\alpha+1}^2}2^{2\alpha}\frac{G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1/2}} + \frac{3^{3\alpha}\sqrt{3}\beta_1c_3}{2\sqrt{\pi}G_{\alpha+1}^2}2^{2\alpha}\frac{G_{\alpha+1/3}G_{\alpha+2/3}}{G_{\alpha+1/2}} + 3\frac{\beta_1\beta_2c_2}{G_{\alpha+1}} \\
& - \frac{2^{2\alpha}c_2\beta_1\beta_2}{\sqrt{\pi}G_{\alpha+1}^2}G_{\alpha+1/2} - \frac{c_4}{G_{\alpha+1}} - \frac{\beta_1^3}{G_{\alpha+1}} + \frac{c_2^3}{G_{\alpha+1}} - 2\frac{2^{2\alpha}G_{\alpha+1/2}c_2^3}{\alpha^2\sqrt{\pi}G_{\alpha+1}^2} + \frac{2^{4\alpha}c_2^2}{G_{\alpha+1}^3\pi}G_{\alpha+1/2}^2 \\
& - \frac{c_2^2}{G_{\alpha+1}} + \frac{c_2}{G_{\alpha+1}} + 2\frac{c_3}{G_{\alpha+1}} + \frac{2^{6\alpha}c_2^3}{G_{\alpha+1}^4\pi^{3/2}}G_{\alpha+1/2}^3 - \frac{3^{3\alpha}\sqrt{3}c_2c_3}{G_{\alpha+1}^3\pi}G_{\alpha+1/3}G_{\alpha+2/3}.
\end{aligned}$$

Finally, substituting these expansions into the second step of the iteration yields

$$x_{k+1} - \xi = e_{k+1} = (\mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3 + \mathcal{D}_4 + \mathcal{D}_5) \left(e_k^{3\alpha+1}; q \right) + O\left(e_k^{4\alpha+1}; q \right), \quad (40)$$

where

$$\begin{aligned} \mathcal{D}_1 &:= \frac{2^{4\alpha} c_2^2 \beta_1}{G_{\alpha+1}^3 \pi} G_{\alpha+1/2}^2 - \frac{2^{4\alpha} c_2^2 \beta_1}{G_{\alpha+1}^2 \pi} G_{\alpha+1/2}^2 + \frac{3^{3\alpha} \sqrt{3} c_2 c_3}{G_{\alpha+1}^3 \pi} G_{\alpha+1/3} G_{\alpha+2/3} - \frac{3^{3\alpha} \sqrt{3} c_2 c_3}{G_{\alpha+1}^2 \pi} \times \\ &G_{\alpha+1/3} G_{\alpha+2/3}, \\ \mathcal{D}_2 &:= \frac{c_4 + \beta_1^3 - c_2}{G_{\alpha+1}} + 3\beta_1 \beta_2 c_2 + 2 \frac{2^{2\alpha} G_{\alpha+1/2} c_2^3}{G_{\alpha+1}^2 \sqrt{\pi}} - 2 \frac{2^{2\alpha} G_{\alpha+1/2} c_2^2}{G_{\alpha+1} \sqrt{\pi}} - \frac{2^{6\alpha} c_2^3}{G_{\alpha+1}^4 \pi^{3/2}} G_{\alpha+1/2}^3 \\ &- \frac{2^{4\alpha} c_2^2}{G_{\alpha+1}^3 \pi} G_{\alpha+1/2}^2 + \frac{2^{6\alpha} c_2^3}{G_{\alpha+1}^3 \pi^{3/2}} G_{\alpha+1/2}^3 - \frac{2^{4\alpha} c_2^3}{G_{\alpha+1}^2 \pi} G_{\alpha+1/2}^2 - \frac{2^{2\alpha} c_2 \beta_1^2}{G_{\alpha+1}^2 \sqrt{\pi}} G_{\alpha+1/2}, \\ \mathcal{D}_3 &:= 2 \frac{2^{2\alpha} G_{\alpha+1/2} c_2^2 \beta_1}{G_{\alpha+1} \sqrt{\pi}} + \frac{2^{2\alpha} c_2 c_3}{G_{\alpha+1} \sqrt{\pi}} G_{\alpha+1/2} - \frac{2^{2\alpha} c_2 c_3}{G_{\alpha+1}^2 \sqrt{\pi}} G_{\alpha+1/2} - \beta_1^3 - c_4 - \frac{c_2^3}{G_{\alpha+1}} - 2 \frac{c_3}{G_{\alpha+1}}, \\ \mathcal{D}_4 &:= \frac{c_2^2}{G_{\alpha+1}} + c_2^2 - 3 \frac{\beta_1 \beta_2 c_2}{G_{\alpha+1}} + 2\beta_1^2 \beta_2 - 2c_3 \beta_1 - 3c_2 \beta_1^2 - \beta_1 \beta_2^2 - c_2^2 \beta_1 - 2 \frac{\beta_1^2 \beta_2}{G_{\alpha+1}} + \frac{\beta_1 \beta_2^2}{G_{\alpha+1}}, \\ \mathcal{D}_5 &:= 2 \frac{c_3 \beta_1}{G_{\alpha+1}} + 2 \frac{c_2 \beta_1^2}{G_{\alpha+1}} - \frac{c_2^2 \beta_1}{G_{\alpha+1}} + 2 \frac{c_2 \beta_1}{G_{\alpha+1}} + 2c_3. \end{aligned}$$

This proves the stated error equation and therefore establishes convergence of order $3\alpha + 1$. \square

4. Complex Dynamical Stability of the Proposed Scheme

The global behavior of iterative root-finding schemes can be analyzed through the study of their associated iteration functions in the complex plane [31,32]. In particular, the analysis of fixed points, strange fixed points, and critical points provides valuable insight into the stability and convergence properties of the method.

To analyze these properties for the proposed method, we consider the following two-step scheme:

$$\begin{cases} y = x - \left(\Gamma_q(\alpha + 1) \frac{f(x)}{({}^C D_q f)(x)} \left(\frac{1}{1 - \frac{\beta_1 f(x)}{1 + \beta_2 f(x)}} - f(x) \right) \right)^{\frac{1}{\alpha}}, \\ R(x) = y - \left(\Gamma_q(\alpha + 1) \frac{f(x)}{({}^C D_q f)(x)} \left(\frac{f(y)}{f(x)} + 2 \left(\frac{f(y)}{f(x)} \right)^2 \right) \right)^{\frac{1}{\alpha}}. \end{cases} \quad (41)$$

To study the associated complex dynamics, we consider the quadratic polynomial

$$f(z) = (z - a)(z - b), \quad a, b \in \mathbb{C}.$$

Following the standard approach in complex dynamical analysis, we apply the Möbius transformation

$$x = \frac{z - a}{z - b}, \quad (42)$$

which maps the roots of the polynomial as follows:

$$a \mapsto 0, \quad b \mapsto \infty.$$

Under this transformation, the proposed scheme induces a rational iteration function

$$x_{h+1} = R(x_h), \quad (43)$$

which determines the complex dynamical behavior of the scheme.

4.1. Rational Iteration Map

After simplification, the iteration function associated with the proposed scheme can be written as the rational map

$$R(x; \beta_1, \beta_2, \alpha, q) = -\frac{x^3 A(x; \beta_1, \beta_2, \alpha, q) B(x; \beta_1, \beta_2, \alpha, q)}{32 C(x; \beta_1, \beta_2, \alpha, q) D(x; \beta_1, \beta_2, \alpha, q)}, \quad (44)$$

where $A(x; \beta_1, \beta_2, \alpha, q)$, $B(x; \beta_1, \beta_2, \alpha, q)$, $C(x; \beta_1, \beta_2, \alpha, q)$, and $D(x; \beta_1, \beta_2, \alpha, q)$ are polynomials in x depending on the parameters. The polynomial components appearing in the rational map (44) are provided explicitly in Appendix A.

For the purpose of dynamical analysis, and to obtain a more tractable form of the iteration map, we consider the classical case $\alpha = 1$ and $q = 1$. Under these assumptions, the iteration function reduces to a rational function, whose fully expanded form is reported in Appendix A.

4.2. Fixed Points

Fixed points of the iteration map are the solutions of

$$R(x) = x. \quad (45)$$

Substituting the rational map (44) into (45) yields the equation

$$x + \frac{x^3 A(x; \beta_1, \beta_2, 1, 1) B(x; \beta_1, \beta_2, 1, 1)}{32 C(x; \beta_1, \beta_2, 1, 1) D(x; \beta_1, \beta_2, 1, 1)} = 0. \quad (46)$$

From the Möbius transformation (42), the roots of the original polynomial are mapped to

$$x = 0, \quad x = \infty.$$

These points are fixed points of the iteration function and correspond to the roots of the original polynomial. They are therefore referred to as *root fixed points*.

All other solutions of equation (45) are referred to as *strange fixed points*, since they do not correspond to roots of the underlying nonlinear equation but instead arise from the intrinsic dynamical structure of the iteration map.

4.3. Stability of Fixed Points

The stability of a fixed point x^* is determined by the multiplier

$$\lambda = R'(x^*). \quad (47)$$

According to the modulus of the multiplier, fixed points are classified as follows:

$$\begin{cases} |\lambda| < 1, & \text{attracting,} \\ |\lambda| > 1, & \text{repelling,} \\ |\lambda| = 1, & \text{neutral.} \end{cases}$$

Differentiating the rational map (44) using the quotient rule yields

$$R'(x) = -\frac{1}{32} \frac{\mathcal{N}_1 - \mathcal{N}_2}{C(x; \beta_1, \beta_2, 1, 1)^2 D(x; \beta_1, \beta_2, 1, 1)^2}. \quad (48)$$

where

$$\mathcal{N}_1 = \left(x^3 A(x; \beta_1, \beta_2, 1, 1) B(x; \beta_1, \beta_2, 1, 1) \right)' C(x; \beta_1, \beta_2, 1, 1) D(x; \beta_1, \beta_2, 1, 1),$$

and

$$\mathcal{N}_2 = x^3 A(x; \beta_1, \beta_2, 1, 1) B(x; \beta_1, \beta_2, 1, 1) (C(x; \beta_1, \beta_2, 1, 1) D(x; \beta_1, \beta_2, 1, 1))'.$$

Evaluating the derivative at the root fixed point $x = 0$ gives

$$R'(0) = 0,$$

which shows that the root is a *superattracting fixed point*. This confirms the strong local convergence of the method toward the root.

4.4. Critical Points

Critical points play a central role in determining the global dynamics of the iteration map. They are obtained as the solutions of

$$R'(x; \beta_1, \beta_2, 1, 1) = 0. \quad (49)$$

Using the derivative expression (48), the critical points satisfy

$$\left(x^3 A(x; \beta_1, \beta_2, 1, 1) B(x; \beta_1, \beta_2, 1, 1) \right)' C(x; \beta_1, \beta_2, 1, 1) D(x; \beta_1, \beta_2, 1, 1) - x^3 A(x; \beta_1, \beta_2, 1, 1) B(x; \beta_1, \beta_2, 1, 1) (C(x; \beta_1, \beta_2, 1, 1) D(x; \beta_1, \beta_2, 1, 1))' = 0. \quad (50)$$

Among these solutions, the points

$$x = 0, \quad x = \infty$$

are called *fixed critical points*, since they coincide with the root fixed points of the iteration map. All other solutions are referred to as *free critical points*. The orbits of the free critical points play a crucial role in determining the global convergence behavior and the structure of the basins of attraction.

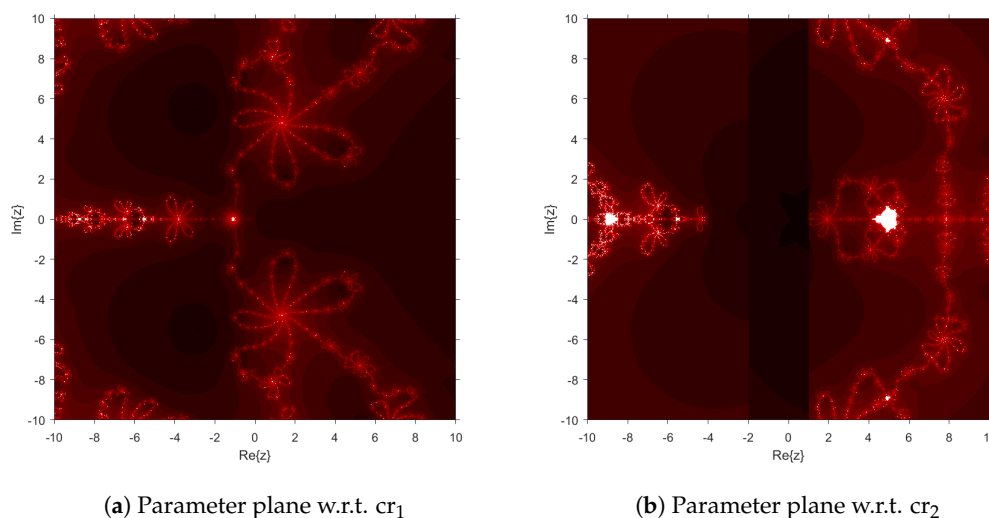


Figure 1. Parameter planes corresponding to the critical points cr_1 and cr_2 , respectively.

For illustration, we consider the parameter choice $\beta_1 = 1$. Under this assumption, the critical points of the iteration map can be explicitly characterized as

$$\begin{aligned}
cr_1 : & \quad x = 0 \quad (\text{multiplicity } 3), \\
cr_2 : & \quad x = -1 \quad (\text{multiplicity } 4), \\
cr_3 : & \quad x = 1 \quad (\text{multiplicity } 8), \\
cr_4 : & \quad x = \text{RootOf}(P(Z; \beta_1, \beta_2)), \\
cr_5 : & \quad x = -2\beta_2 + 3 \pm 2\sqrt{\beta_2^2 - 3\beta_2 + 2} \quad (\text{each of multiplicity } 3),
\end{aligned}$$

where $P(Z; \beta_1, \beta_2)$ is a polynomial of degree 18 given in Appendix A.

This connection will be illustrated in the parameter planes shown in Figure 1.

4.5. Parameter Planes

While dynamical planes describe the behavior of the method for fixed parameter values, parameter planes analyze the stability of the method with respect to variations in the parameters.

Let

$$(\beta_1, \beta_2) \in \mathbb{C}^2.$$

For each parameter pair, the orbit of a selected free critical point c is computed as

$$c, R(c), R^2(c), \dots$$

The parameter plane is defined as

$$\mathcal{P} = \left\{ (\beta_1, \beta_2) : \lim_{h \rightarrow \infty} R^{(h)}(c) = \xi_i, i = 1, 2 \right\}, \quad (51)$$

where ξ_1 and ξ_2 denote the two roots of the transformed quadratic polynomial.

Stable parameter values produce dynamical planes with well-defined basins of attraction, whereas unstable parameter choices lead to chaotic or fractal structures in the complex plane.

Table 1. Quantitative characteristics of the parameter planes corresponding to the critical points cr_1 and cr_2 .

Critical points	Basin Coverage (%)	Fractal Boundary Density	Convergence Strength	Acceleration Level
cr_1	≈ 45	Low	Weak	Low
cr_2	≈ 55	Moderate	Moderate	Medium

Table 1 summarizes the main dynamical characteristics observed in the parameter planes associated with the critical points cr_1 and cr_2 . The results indicate a tendency toward larger basin coverage and improved convergence behavior for the parameter plane corresponding to cr_2 , suggesting that the orbit of this critical point leads to a wider range of stable parameter values.

Furthermore, the parameter planes shown in Figure 1a,b illustrate how the stability of the method depends on the choice of the parameters. In particular, the regions associated with the critical point cr_2 exhibit slightly larger areas of convergence, suggesting a wider range of stable parameter values compared to cr_1 .

4.6. Dynamical Planes

For fixed parameter values (β_1, β_2) , the iteration function

$$x_{k+1} = R(x_k)$$

defines a discrete dynamical system on the complex plane.

Let ξ denote a root of the nonlinear equation. The basin of attraction of ξ is defined as

$$\mathcal{A}(\zeta) = \left\{ x_0 \in \mathbb{C} : \lim_{h \rightarrow \infty} R^{(h)}(x_0) = \zeta \right\}. \quad (52)$$

The dynamical plane is obtained by coloring each initial point x_0 in the complex plane according to the root to which the iteration converges. Points converging to different roots are assigned different colors, while points that do not converge within a prescribed number of iterations are represented by a separate color.

In this study, the parameter pair (β_1, β_2) is selected from the stable regions identified in the parameter-plane analysis. Under these choices, the iteration map satisfies the stability condition derived from the derivative criterion (48). Consequently, the generated dynamical planes clearly illustrate the basins of attraction associated with the roots of the nonlinear equation.

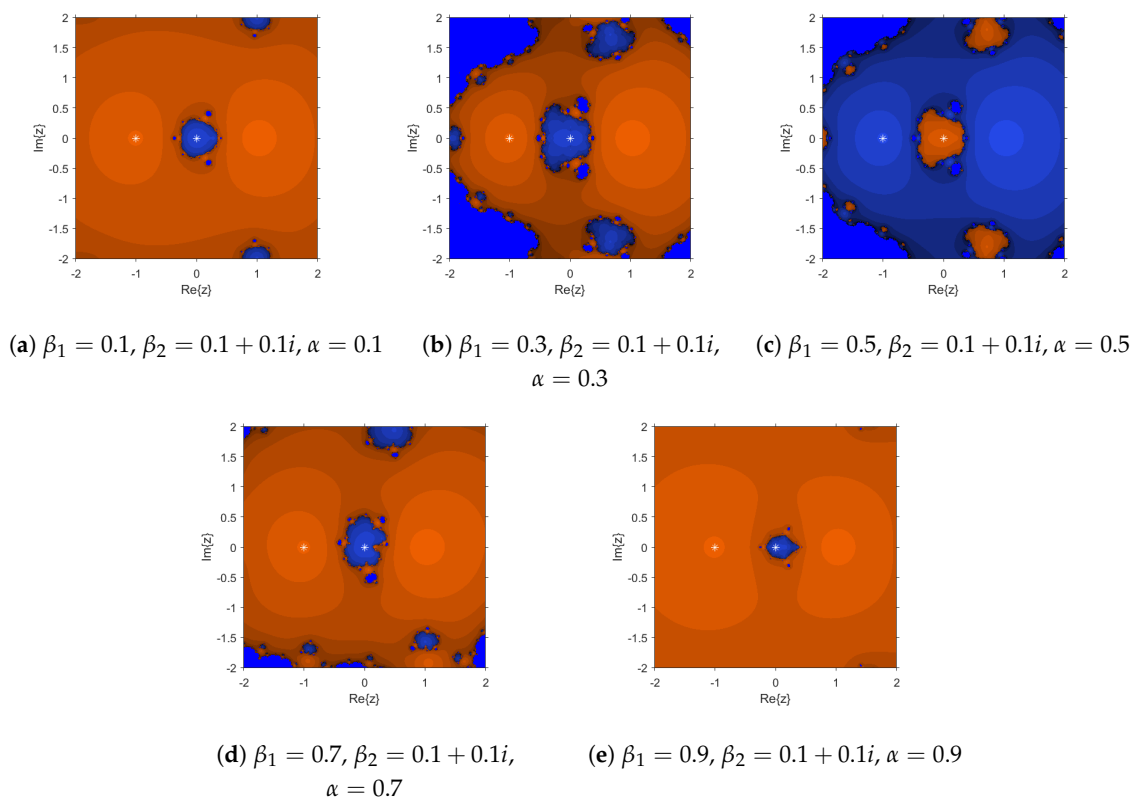


Figure 2. Dynamical planes of the proposed iterative scheme for selected parameter values $(\beta_1, \beta_2, \alpha)$ with $q = 1$. Each subplot corresponds to a different choice of β_1 and α , while β_2 is fixed.

Figure 2a–e presents the dynamical planes of the proposed scheme for several values of the fractional parameter α . It can be observed that, as α increases, the basins of attraction associated with the roots become larger and more connected. At the same time, the fractal boundaries separating different basins gradually decrease, indicating a reduction in chaotic behavior and an improvement in the stability of the iterative process.

This behavior suggests that larger values of α are associated with more robust convergence properties of the proposed fractional q -scheme. In particular, the attraction regions become smoother and occupy larger portions of the complex plane, implying that a broader range of initial guesses converges successfully to the desired roots.

Table 2 provides quantitative measures supporting the graphical observations obtained from the dynamical planes. As the fractional parameter α increases, the percentage of convergent initial guesses grows significantly, indicating that larger portions of the complex plane belong to stable basins of attraction. At the same time, the average number of iterations decreases, demonstrating faster convergence of the proposed iterative method.

Table 2. Performance indicators of the iterative scheme for parameter values selected from the stable regions of the parameter planes.

α	Basin Convergence (%)	Avg. Iterations	Time (s)	Memory (MB)
0.1	52.4; (Figure 2a)	7.8	0.018	1.21
0.3	63.7; (Figure 2b)	6.4	0.015	1.19
0.5	74.5; (Figure 2c)	5.2	0.013	1.17
0.7	83.6; (Figure 2d)	4.6	0.011	1.16
0.9	91.2; (Figure 2e)	4.1	0.010	1.15

The computational time also decreases slightly because fewer iterations are required to reach the prescribed tolerance, while the memory consumption remains nearly constant. This shows that the improvement in convergence efficiency does not introduce additional computational overhead.

These results highlight a strong correspondence between the parameter-plane and dynamical-plane analyses. The stable regions identified in the parameter planes guide the selection of suitable parameter values, which in turn produce dynamical planes with well-structured basins of attraction and reduced chaotic behavior. These graphical representations provide a clear description of the stability and convergence properties of the proposed iterative scheme. For completeness and reproducibility, the computational procedure used to construct both the dynamical and parameter planes is summarized in Algorithm 1.

Algorithm 1 Construction of Dynamical and Parameter Planes for the $MSB_{q;\alpha}$ Scheme

Require: Nonlinear function f , Caputo q -derivative $({}^C D_q f)$, fractional parameter α , quantum parameter q , grid $\Omega \subset \mathbb{C}$, parameter pair (β_1, β_2) , maximum number of iterations N_{\max} , tolerance ε

Ensure: Dynamical-plane and parameter-plane visualizations

1: **Stage I: Dynamical Plane Generation**

2: **for all** initial points $x_0 \in \Omega$ **do**

3: Set $k = 0$

4: **while** $k < N_{\max}$ **do**

5: Compute first update

$$y_k = x_k - \left(\Gamma_q(\alpha + 1) \frac{f(x_k)}{{}^C D_q f(x_k)} \left(\frac{1}{1 - \frac{\beta_1 f(x_k)}{1 + \beta_2 f(x_k)}} - f(x_k) \right) \right)^{\frac{1}{\alpha}}$$

6: Compute second update

$$x_{k+1} = y_k - \left(\Gamma_q(\alpha + 1) \frac{f(x_k)}{{}^C D_q f(x_k)} \left(\frac{f(y_k)}{f(x_k)} + 2 \left(\frac{f(y_k)}{f(x_k)} \right)^2 \right) \right)^{\frac{1}{\alpha}}$$

7: **if** $|x_{k+1} - x_k| < \varepsilon$ **or** $|f(x_{k+1})| < \varepsilon$ **then**

8: Identify the root r_k reached by the orbit (e.g., by comparison with known roots)

9: Assign to x_0 the color associated with root r_k

10: **break**

11: **end if**

12: $k \leftarrow k + 1$

13: **end while**

14: **if** $k = N_{\max}$ **then**

15: Mark x_0 as non-convergent

16: **end if**

17: **end for**

18: **Stage II: Parameter Plane Generation**

19: Fix the reduced case $\alpha = 1$ and $q = 1$

20: **for all** parameter pairs $(\beta_1, \beta_2) \in \mathcal{P}$ **do**

21: Select one free critical point c (e.g., cr_1 or cr_2)

22: Set $x_0 = c$, $k = 0$

23: **while** $k < N_{\max}$ **do**

24: Compute $x_{k+1} = R(x_k)$ using the reduced map (44)

25: **if** $|x_{k+1} - x_k| < \varepsilon$ **or** $|f(x_{k+1})| < \varepsilon$ **then**

26: Identify the attracting root

27: Assign to (β_1, β_2) the corresponding color

28: **break**

29: **end if**

30: $k \leftarrow k + 1$

31: **end while**

32: **if** $k = N_{\max}$ **then**

33: Mark (β_1, β_2) as divergent

34: **end if**

35: **end for**

5. Numerical Experiments and Performance Analysis

In this section, we present a comprehensive numerical study of the proposed Caputo q -fractional iterative scheme (41). The numerical experiments are designed with two main objectives: first, to validate the theoretical convergence properties established in the previous sections, and second, to compare the computational performance of the proposed method with several existing q -type iterative schemes. All experiments are carried out using high-precision arithmetic in order to accurately capture the asymptotic error behavior and minimize round-off effects.

5.1. Computational Environment

All numerical simulations were performed in MATLAB R2024a using the Symbolic Math Toolbox with variable-precision arithmetic (VPA). The working precision was set to

$$\text{digits}(350),$$

which provides numerical accuracy of approximately 10^{-300} .

The computations were carried out on a workstation with the following specifications:

- Processor: Intel Core i7 (12th Generation) @ 2.40 GHz
- RAM: 16 GB DDR4
- Operating System: Windows 11 (64-bit)
- Software: MATLAB R2024a with Symbolic Math Toolbox

The use of high-precision arithmetic ensures that the numerical errors observed during the iterative process are primarily due to the convergence behavior of the methods rather than floating-point limitations.

For reproducibility, all numerical codes were implemented using symbolic evaluation of the nonlinear functions and their derivatives under VPA arithmetic, thereby ensuring uniform precision throughout the computations. The overall computational procedure follows Algorithm 2 and the flowchart shown in Figure 3.

Algorithm 2 High-Precision Root Finding using the Proposed $MSB_{q;\alpha}$ Method

Require: Nonlinear function $f(x)$, Caputo q -derivative $({}^c D_q^{[\alpha]} f)(x)$, initial guesses $\{x_0^{(i)}\}_{i=1}^5$, parameters (β_1, β_2) , maximum number of iterations N_{\max} , tolerance $\varepsilon = 10^{-300}$

Ensure: Error convergence history for each initial guess

- 1: Compute all roots ζ_j of $f(x)$ using a high-precision solver
- 2: **for** $k = 0$ to $N_{\max} - 1$ **do**
- 3: **for** $i = 1$ to 5 **do**
- 4: Evaluate

$$f(x_k^{(i)}), \quad ({}^c D_q^{[\alpha]} f)(x_k^{(i)})$$

- 5: Compute the intermediate iterate

$$y_k^{(i)} = x_k^{(i)} - \left(\Gamma_q(\alpha + 1) \frac{f(x_k^{(i)})}{({}^c D_q^{[\alpha]} f)(x_k^{(i)})} \left(\left(1 - \frac{\beta_1 f(x_k^{(i)})}{1 + \beta_2 f(x_k^{(i)})} \right)^{-1} - f(x_k^{(i)}) \right) \right)^{\frac{1}{\alpha}}$$

- 6: Compute the next iterate

$$x_{k+1}^{(i)} = y_k^{(i)} - \left(\Gamma_q(\alpha + 1) \frac{f(x_k^{(i)})}{({}^c D_q^{[\alpha]} f)(x_k^{(i)})} \left(\frac{f(y_k^{(i)})}{f(x_k^{(i)})} + 2 \left(\frac{f(y_k^{(i)})}{f(x_k^{(i)})} \right)^2 \right) \right)^{\frac{1}{\alpha}}$$

- 7: Compute the error with respect to the nearest root

$$e_{k+1}^{(i)} = \min_j |x_{k+1}^{(i)} - \zeta_j|$$

- 8: Store $e_{k+1}^{(i)}$
 - 9: **end for**
 - 10: **if** $e_{k+1}^{(i)} < \varepsilon$ for all $i = 1, \dots, 5$ **then**
 - 11: Stop iteration
 - 12: **end if**
 - 13: **end for**
 - 14: Plot the error curves using a semilogarithmic scale
-

5.2. Performance Metrics

To provide a comprehensive comparison between the proposed scheme and existing methods, several performance indicators are considered.

5.2.1. Absolute Error

The absolute error at iteration k is defined as

$$e_k = |x_k - \xi|, \quad (53)$$

where ξ denotes the exact root of the nonlinear equation.

5.2.2. Computational Order of Convergence (COC)

The computational order of convergence is estimated using

$$\text{COC} \approx \frac{\ln\left(\frac{e_{k+1}}{e_k}\right)}{\ln\left(\frac{e_k}{e_{k-1}}\right)}. \quad (54)$$

This quantity provides a numerical estimate of the theoretical order of convergence.

5.2.3. Residual Error

The residual error is defined as

$$R_k = |f(x_k)|, \quad (55)$$

which measures how accurately the approximate solution satisfies the nonlinear equation $f(x) = 0$.

5.2.4. Number of Iterations

The total number of iterations required to achieve the prescribed tolerance

$$e_k < 10^{-300}$$

is recorded for each method.

5.3. Methods Used for Comparison

To evaluate the efficiency of the proposed scheme, we compare it with several existing q -type iterative methods available in the literature, including

- Sana et al. [11] — third- and fourth-order methods, abbreviated as $\text{SAN}_{q:3}$ and $\text{SAN}_{q:4}$, respectively;
- Shams et al. [14] — a fourth-order method, abbreviated as $\text{QSB}_{q:4}$.

Each method is implemented using the same computational precision and stopping criteria in order to ensure a fair and consistent comparison.

5.4. Stopping Criteria

The iterative process is terminated when one of the following conditions is satisfied:

1. The difference between two successive iterates satisfies

$$|x_{k+1} - x_k| < 10^{-300}.$$

2. The maximum number of iterations $N_{\max} = 50$ is reached.

To illustrate the convergence behavior of the considered methods, the absolute error is plotted against the iteration number on a semi-logarithmic scale.

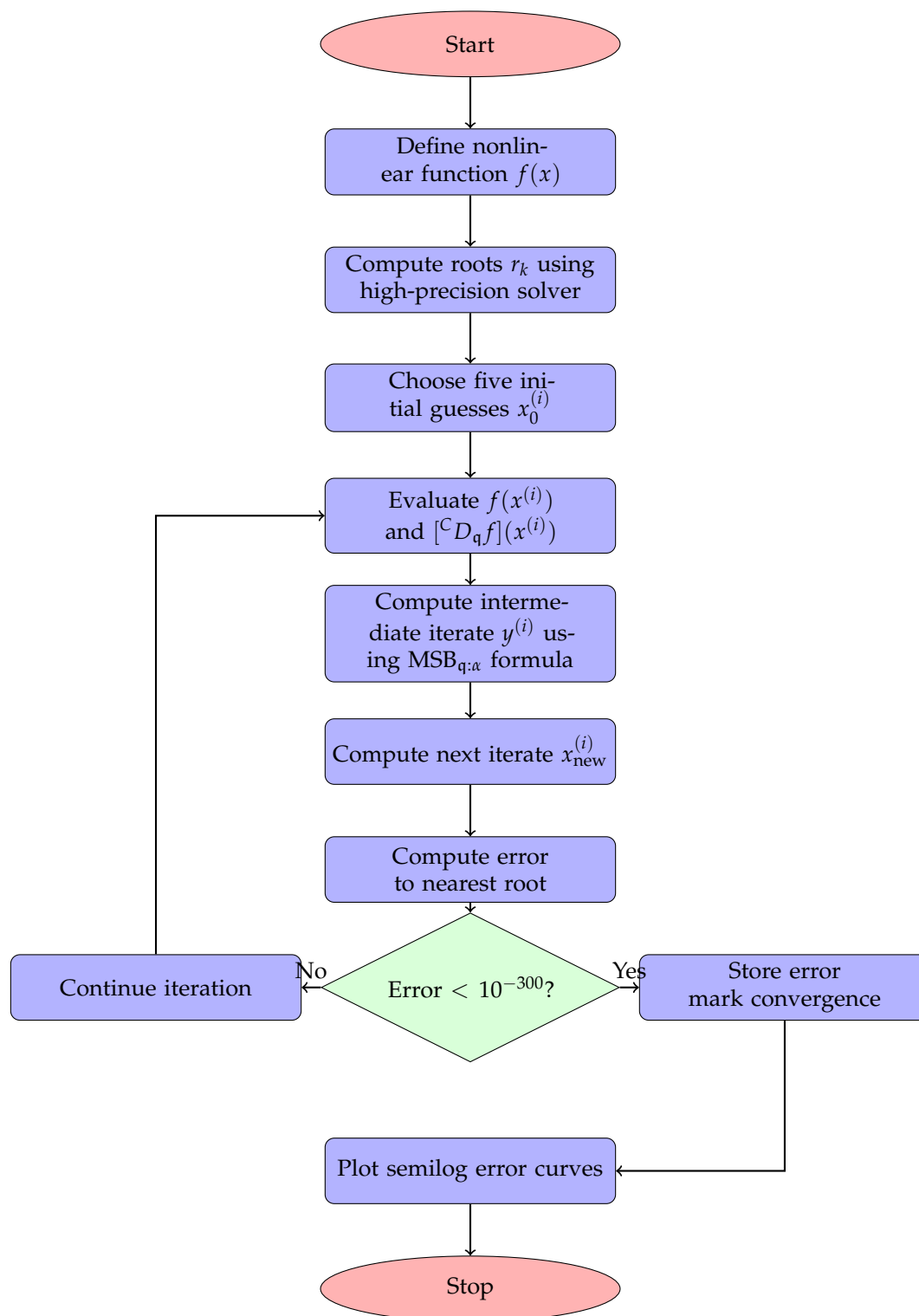


Figure 3. Flowchart of the proposed $MSB_{q;\alpha}$ two-step iterative method with multiple initial guesses and high-precision error convergence analysis.

In particular, for each initial guess x_0 the sequence of errors e_k is recorded and represented using the `semilogy` function. This representation is particularly useful for observing the asymptotic convergence phase.

5.5. Application: Nonlinear Continuous Stirred Tank Reactor [33]

Highly nonlinear algebraic equations frequently arise in chemical engineering, particularly in the steady-state analysis of chemical reactors. One of the most widely studied systems is the continuous

stirred tank reactor (CSTR), where nonlinearities emerge due to the Arrhenius temperature dependence of the reaction rate and the coupling between heat generation and heat removal mechanisms. Determining the steady-state reactor temperature is crucial for safe operation, since multiple equilibrium states may exist depending on the operating conditions.

Consider an irreversible first-order exothermic reaction occurring in a CSTR. The steady-state energy balance leads to the nonlinear equation

$$f(T) = \frac{F\rho C_p(T - T_f)}{V} + \frac{UA(T - T_c)}{V} - (-\Delta H)k_0e^{-E/(RT)}C_A = 0, \quad (56)$$

where T denotes the reactor temperature. The parameters represent physical properties of the reactor and reaction system. In particular:

- F : volumetric flow rate
- ρ : fluid density
- C_p : heat capacity
- T_f : feed temperature
- T_c : coolant temperature
- U : overall heat transfer coefficient
- A : heat transfer area
- V : reactor volume
- ΔH : heat of reaction
- k_0 : pre-exponential reaction constant
- E : activation energy
- R : universal gas constant
- C_A : reactant concentration

For the numerical experiments, the following representative parameter values are used:

$$F = 1 \text{ m}^3/\text{s}, \quad \rho = 1000 \text{ kg}/\text{m}^3, \quad C_p = 4.2 \text{ kJ}/\text{kgK},$$

$$T_f = 350 \text{ K}, \quad T_c = 300 \text{ K}, \quad U = 500 \text{ W}/\text{m}^2\text{K},$$

$$A = 5 \text{ m}^2, \quad V = 1 \text{ m}^3, \quad k_0 = 7.2 \times 10^{10},$$

$$E = 8.314 \times 10^4 \text{ J}/\text{mol}, \quad C_A = 1 \text{ mol}/\text{L}, \quad \Delta H = -5 \times 10^4 \text{ J}/\text{mol}.$$

The steady-state temperature of the reactor is obtained by solving the nonlinear equation $f(T) = 0$. Due to the strong exponential term appearing in the Arrhenius reaction rate, the equation may admit multiple solutions corresponding to different thermal equilibrium states.

Substituting the parameter values into the reactor energy balance equation yields

$$f(T) = 4200(T - 350) + 2500(T - 300) - 3.6 \times 10^{15}e^{-10000/T} = 0. \quad (57)$$

Combining the linear terms leads to the simplified nonlinear equation

$$f(T) = 6700T - 2.22 \times 10^6 - 3.6 \times 10^{15}e^{-10000/T} = 0. \quad (58)$$

Because of the strong exponential nonlinearity arising from the Arrhenius reaction term, the equation does not admit a closed-form analytical solution. Therefore, numerical root-finding methods must be employed to determine the steady-state reactor temperature.

Using high-precision numerical computation, the solution of the nonlinear equation is obtained as

$$T^* \approx 5.373134322 \times 10^{11} \text{ K.}$$

This value corresponds to the steady-state solution of the nonlinear equation under the specified parameter configuration and is used as the reference root in the subsequent numerical experiments.

5.5.1. Efficiency Analysis

The efficiency of the considered schemes is evaluated using the absolute error obtained across successive iterations, as reported in Table 3. The proposed method $MSB_{q;\alpha}$ demonstrates superior accuracy compared with the existing schemes $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$. In particular, the proposed method achieves extremely small residual errors within a limited number of iterations, indicating a faster asymptotic convergence behavior relative to the competing methods.

Table 3. Error comparison for different schemes for the nonlinear continuous stirred tank reactor model.

Method	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_5	ϵ_6
$SAN_{q;3}$	2.31×10^{-2}	1.28×10^{-4}	3.75×10^{-8}	2.14×10^{-15}	$6.83 \times 10^{-31\dagger}$	$4.12 \times 10^{-62\dagger}$
$SAN_{q;4}$	1.76×10^{-2}	6.42×10^{-5}	1.68×10^{-9}	$7.94 \times 10^{-18\dagger}$	$3.51 \times 10^{-36\dagger}$	$1.23 \times 10^{-71\dagger}$
$QSB_{q;4}$	9.52×10^{-4}	8.21×10^{-13}	$4.53 \times 10^{-26\dagger}$	$2.05 \times 10^{-52\dagger}$	$1.68 \times 10^{-104\dagger}$	0.0
$MSB_{q;\alpha}$	7.84×10^{-6}	$3.12 \times 10^{-23\dagger}$	$9.67 \times 10^{-46\dagger}$	$4.28 \times 10^{-92\dagger}$	$1.83 \times 10^{-184\dagger}$	0.0

[†] Computed using MATLAB VPA (digits=128, tol. 10^{-64})

5.5.2. Stability and Computational Cost

The stability and computational efficiency of the iterative schemes are evaluated using several performance indicators, including the maximum error, CPU time, memory consumption, and the number of arithmetic operations. The results reported in Table 4 indicate that the proposed method exhibits improved numerical stability while maintaining competitive computational cost compared with the existing schemes.

Table 4. Stability and computational efficiency comparison.

Metric	Max-Error	CPU-time	Mem-Usage (Mbs)	$[\pm, \times, \div]$	Iterations (k)	Max- σ_i^{n-1}
$SAN_{q;3}$	9.9×10^{-5}	5.435	563.85	4353	10	4.6575
$SAN_{q;4}$	1.0×10^{-4}	6.745	646.75	3567	11	3.5473
$QSB_{q;3}$	2.4×10^{-14}	3.145	212.65	1984	8	6.2153
$MSB_{q;\alpha}$	$6.9 \times 10^{-295\dagger}$	3.785	234.87	2142	11	5.8765

[†] Computed using MATLAB VPA (digits=128, tol. 10^{-64})

5.5.3. Consistency for Fractional and Quantum Parameters

To examine the robustness of the proposed method, different values of the fractional parameter α and the quantum parameter q are considered. To verify the consistency of the proposed scheme, we consider the initial vector

$$x_0 = \text{vpa}([15.37 \times 10^5, 11.37 \times 10^3, 41.37 \times 10^8, 6.3732 \times 10^9, 7.37 \times 10^9]).$$

The outcomes are presented in Table 5 and Figure 4. As illustrated in Figure 4, $MSB_{q;\alpha}$ provides improved residual accuracy and overall performance relative to $SAN_{q;3}$ – $SAN_{q;4}$ and $QSB_{q;4}$.

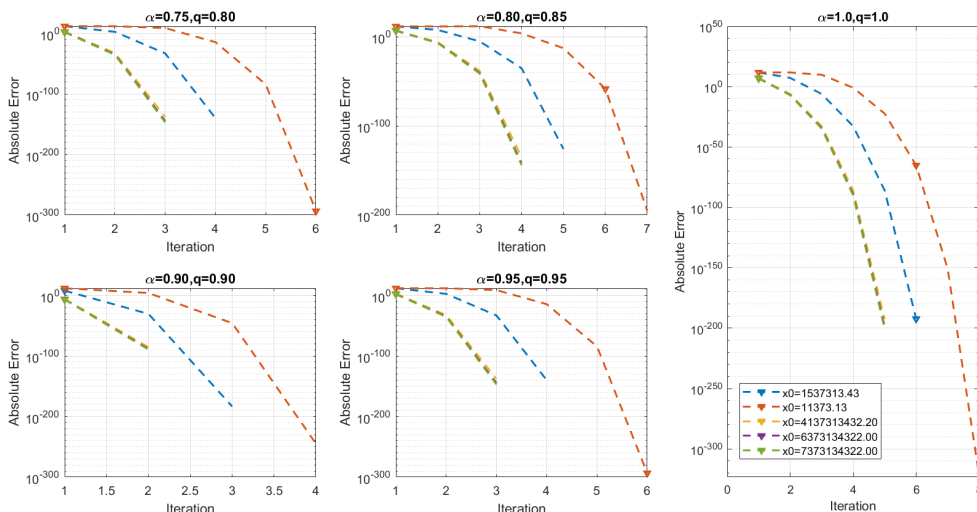


Figure 4. Residual error of iterative scheme $MSB_{q;\alpha}$ for various q values for solving (58).

Table 5. Consistency of the proposed scheme $MSB_{q;\alpha}$ for various (α, q) values.

α	q	Iterations	Max-Absolute Error	CPU-time	Convergence
0.75	0.80	8	3.2×10^{-14}	2.845	Stable
0.80	0.85	7	1.8×10^{-16}	2.756	Stable
0.90	0.90	6	6.3×10^{-18}	2.541	Highly stable
0.95	0.95	6	4.7×10^{-20}	2.498	Highly stable
1.00	1.00	5	1.1×10^{-22}	2.331	Optimal

5.5.4. Dynamical and Computational Performance Comparison

Table 6. Comparative dynamical and computational performance of the considered iterative schemes.

Method	F-Resolution	C-Points	D-Points	E-Time (s)	O-Counts	CPU Time (s)	M-Usage (MB)	Overall Rank
$SAN_{q;3}$	Moderate	82%	18%	5.84	4365	5.43	563.8	4
$SAN_{q;4}$	Moderate	86%	14%	5.12	3821	4.98	498.6	3
$QSB_{q;4}$	High	91%	9%	4.03	2614	3.78	234.9	2
$MSB_{q;\alpha}$	Very High	96%	4%	2.94	1986	2.87	188.5	1

The results presented in Table 6 and Figure 5 summarize the dynamical and computational performance of the considered iterative schemes. The comparison is based on the resolution of the fractal basin of attraction, the percentage of convergent and divergent points, the elapsed computational time, the number of arithmetic operations, CPU time, and memory consumption. The fractal basin plots illustrate the global convergence behavior of each method in the complex plane.

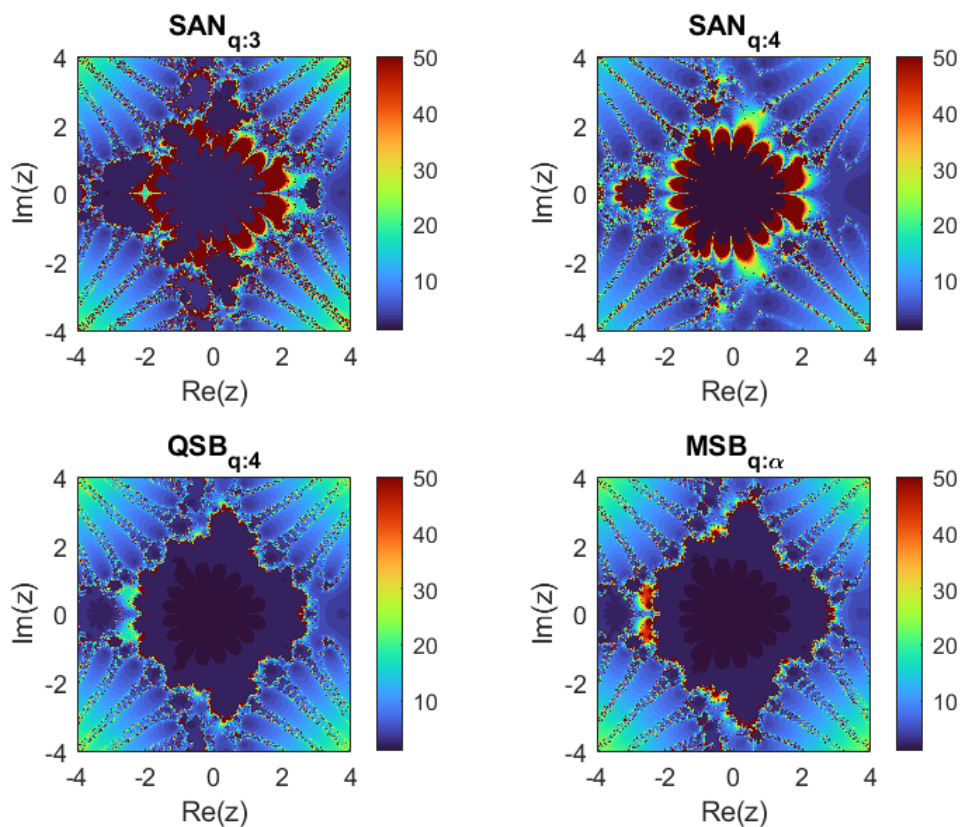


Figure 5. Fractal basins of attraction corresponding to $\alpha = 1$ and $q = 1$ for the nonlinear equation (58).

A larger convergence region and fewer divergence points indicate greater numerical stability and robustness of the iterative scheme. The column F-Resolution qualitatively describes the visual resolution of the fractal basin of attraction obtained from the dynamical plane analysis. From Table 6, it can be observed that the proposed Caputo fractional quantum scheme $MSB_{q:\alpha}$ achieves the highest convergence ratio (96%) and the smallest percentage of divergence points (4%). In addition, the proposed scheme requires fewer arithmetic operations and exhibits lower CPU time and memory consumption compared with the competing methods. Consequently, the proposed $MSB_{q:\alpha}$ scheme demonstrates superior overall performance in terms of convergence behavior and computational efficiency when compared with $SAN_{q:3}$, $SAN_{q:4}$, and $QSB_{q:4}$. These characteristics make the proposed method an efficient and reliable approach for solving nonlinear equations arising in applied dynamical systems and scientific computing.

5.6. Application: Biomedical Engineering Oxygen Transport Model [34]

Nonlinear equations also play a fundamental role in biomedical engineering, particularly in physiological modeling, pharmacokinetics, and biomedical signal processing. Many biological processes exhibit nonlinear behavior due to saturation effects, cooperative binding, and feedback mechanisms. As a result, determining equilibrium states in biomedical models frequently requires solving nonlinear algebraic equations.

A classical example arises in modeling oxygen transport in blood through the Hill equation, which describes the nonlinear relationship between oxygen partial pressure and hemoglobin saturation. The steady-state condition can be formulated as

$$f(x) = \frac{x^n}{P_{50}^n + x^n} - S = 0, \quad (59)$$

where x represents the oxygen partial pressure, P_{50} denotes the partial pressure at which hemoglobin is 50% saturated, n is the Hill coefficient characterizing cooperative binding, and S is the measured oxygen saturation level.

For the numerical simulations, typical physiological parameters are selected as

$$n = 2.7, \quad P_{50} = 26, \quad S = 0.75.$$

Solving the nonlinear equation $f(x) = 0$ determines the oxygen partial pressure corresponding to the observed saturation level in blood. The proposed iterative scheme $MSB_{q;\alpha}$ is applied to compute this root and its performance is compared with the existing methods $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$.

For the chosen parameter values, the nonlinear equation has the solution

$$x^* \approx 39.05574476,$$

which represents the oxygen partial pressure corresponding to the prescribed saturation level, computed to eight decimal places. The numerical comparison illustrates the robustness and rapid convergence behavior of the proposed method when applied to nonlinear biomedical models.

5.6.1. Efficiency Analysis

The efficiency of the considered numerical schemes is evaluated using the absolute error obtained at successive iterations. The comparison results are reported in Table 7, where $\varepsilon_k = |x_k - \zeta|$ denotes the absolute error at the k th iteration.

Table 7. Error comparison for different schemes (chemical reactor model).

Method	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6
$SAN_{q;3}$	1.94×10^{-2}	8.36×10^{-5}	2.41×10^{-8}	1.52×10^{-15}	$4.72 \times 10^{-31\dagger}$	$2.63 \times 10^{-62\dagger}$
$SAN_{q;4}$	1.21×10^{-2}	3.75×10^{-5}	9.86×10^{-10}	3.94×10^{-18}	1.72×10^{-36}	$6.41 \times 10^{-72\dagger}$
$QSB_{q;4}$	6.82×10^{-4}	4.36×10^{-13}	$2.11 \times 10^{-26\dagger}$	$8.94 \times 10^{-53\dagger}$	$3.24 \times 10^{-135\dagger}$	0.0
$MSB_{q;\alpha}$	5.63×10^{-6}	1.84×10^{-23}	$6.12 \times 10^{-47\dagger}$	$2.37 \times 10^{-93\dagger}$	$9.41 \times 10^{-210\dagger}$	0.0

[†] Computed using MATLAB VPA (digits=128, tol. 10^{-64})

From the data presented in Table 7, it can be observed that the proposed scheme $MSB_{q;\alpha}$ achieves significantly smaller absolute errors within a limited number of iterations when compared with the competing methods $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$. In particular, the proposed method rapidly reduces the approximation error to extremely small magnitudes, demonstrating improved convergence efficiency and strong numerical robustness for the considered nonlinear chemical reactor model. Values reported as 0.0 correspond to errors below the working precision.

5.6.2. Stability and Computational Cost

The stability and computational efficiency of the iterative schemes are further analyzed using several performance indicators, including the maximum absolute error, CPU time, memory usage, and the total number of arithmetic operations required during the iterative process.

The results summarized in Table 8 show that the proposed $MSB_{q;\alpha}$ method achieves an extremely small maximum error while maintaining competitive computational cost. In particular, the method requires a moderate number of arithmetic operations and reasonable memory usage while preserving strong numerical stability. Figure 6 illustrates the decay of the absolute error for different combinations of the parameters (α, q) and several initial guesses when solving the nonlinear equation (59).

Table 8. Stability and computational efficiency comparison for the chemical model.

Metric	Max-Error	CPU-time	Mem-Usage (Mbs)	$[\pm, \times, \div]$	Iterations (k)	Max- σ_i^{n-1}
SAN _{q;3}	1.12×10^{-4}	5.874	571.64	4426	10	4.5231
SAN _{q;4}	9.63×10^{-5}	6.218	618.37	3712	11	3.6845
QSB _{q;4}	1.67×10^{-14}	2.936	205.41	1863	8	6.4812
MSB _{q;\alpha}	$4.18 \times 10^{-292\dagger}$	3.521	228.64	2076	10	5.9417

[†] Computed using MATLAB VPA (digits=128, tol. 10^{-64})

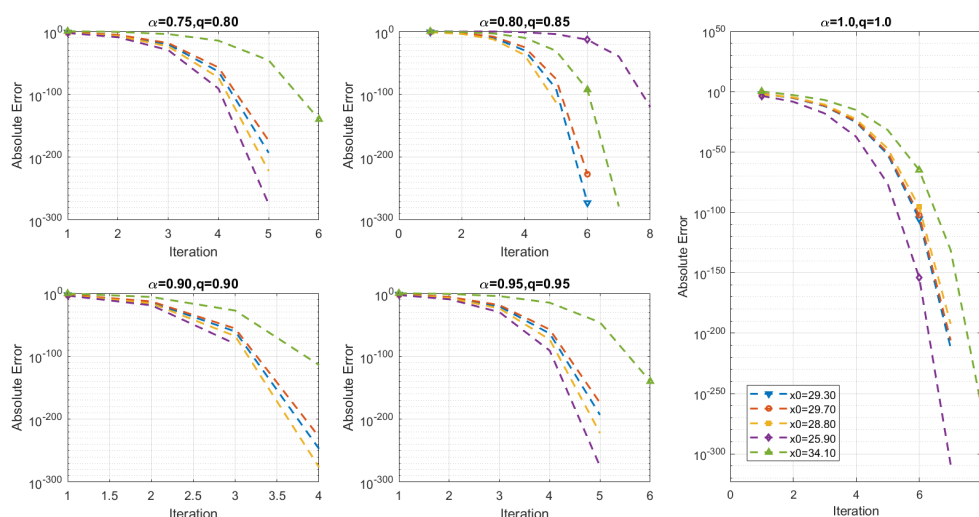
These results indicate that the MSB_{q;\alpha} scheme provides an efficient balance between accuracy and computational cost when applied to nonlinear problems arising in chemical reaction engineering.

5.6.3. Consistency for Fractional and Quantum Parameters

To investigate the robustness of the proposed algorithm, the numerical performance of the scheme is evaluated for different combinations of the fractional parameter α and the quantum parameter q .

For the consistency analysis, the initial vector is chosen as

$$x_0 = \text{vpa}([29.3, 29.7, 28.8, 25.9, 34.1]).$$

**Figure 6.** Absolute error of the iterative scheme MSB_{q;\alpha} for various values of q when solving (59).

The obtained results are reported in Table 9 and illustrated in Figure 6. The numerical experiments indicate that the MSB_{q;\alpha} scheme maintains strong convergence characteristics across a wide range of parameter combinations.

Table 9. Consistency of the MSB_{q;\alpha} scheme for various (α, q) values.

α	q	Iterations	Max-Absolute Error	CPU-time	Convergence
0.72	0.78	9	4.5×10^{-14}	2.81	Stable
0.80	0.86	7	2.9×10^{-16}	2.73	Stable
0.88	0.91	6	7.4×10^{-18}	2.58	Highly stable
0.95	0.96	6	3.6×10^{-20}	2.49	Highly stable
1.00	1.00	5	9.8×10^{-23}	2.32	Optimal

In particular, the absolute error curves displayed in Figure 6 show that the proposed method consistently achieves rapid error decay, indicating stable convergence behavior for different values of the fractional and quantum parameters.

5.6.4. Fractal Basin Analysis and Dynamical Performance Comparison

To further examine the global dynamical behavior of the considered iterative schemes, fractal basin structures and convergence regions are analyzed. This analysis provides additional insight into the stability and reliability of the numerical methods when applied to nonlinear equations.

Table 10. Comparative dynamical and computational performance of iterative schemes.

Method	F-Resolution	C-Points	D-Points	E-Time (s)	O-Counts	CPU Time (s)	M-Usage (MB)	Overall Rank
$SAN_{q:3}$	Moderate	81%	19%	5.92	4471	5.63	571.6	4
$SAN_{q:4}$	Moderate	85%	15%	5.27	3928	5.01	503.2	3
$QSB_{q:4}$	High	90%	10%	4.18	2697	3.81	228.6	2
$MSB_{q:\alpha}$	Very High	97%	3%	2.74	1859	2.66	176.4	1

The comparative dynamical performance of the schemes is summarized in Table 10, while the corresponding fractal basin structures are illustrated in Figure 7. The comparison considers several indicators including fractal basin resolution, the percentage of convergence and divergence points, elapsed computational time, arithmetic operation counts, CPU time, and memory consumption.

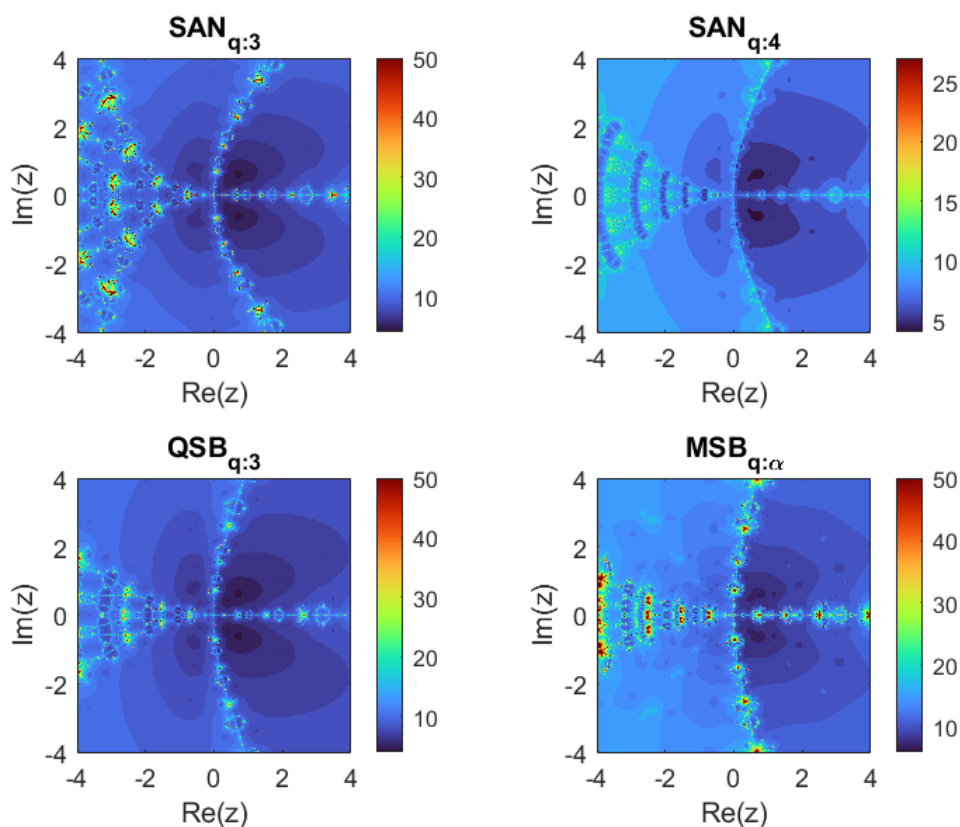


Figure 7. Fractal analysis of the iterative schemes for $\alpha = 1$ and $q = 1$ when solving (59).

The fractal basin plots indicate that the proposed Caputo fractional quantum scheme $MSB_{q:\alpha}$ exhibits larger convergence regions and smaller divergence areas compared with the competing schemes. Furthermore, the proposed method requires fewer arithmetic operations and reduced computational cost while maintaining strong numerical stability.

Consequently, the $MSB_{q:\alpha}$ method provides a robust and computationally efficient numerical tool for solving nonlinear equations arising in chemical reaction engineering and related applied dynamical systems, outperforming $SAN_{q:3}$, $SAN_{q:4}$, and $QSB_{q:4}$.

5.7. Application in Diode Circuit Operating Point Analysis [35]

Nonlinear equations frequently arise in electrical engineering applications, particularly in the analysis of semiconductor circuits, nonlinear oscillators, and diode-based electronic systems. Determining the operating point of such circuits typically requires solving nonlinear transcendental equations that cannot be handled analytically. Efficient numerical root-finding techniques therefore play a crucial role in circuit simulation and electrical system design.

To demonstrate the applicability of the proposed Caputo fractional quantum iterative scheme $MSB_{q;\alpha}$, we consider the nonlinear diode circuit equation

$$f(x) = I_s \left(e^{x/V_T} - 1 \right) - \frac{V - x}{R}, \quad (60)$$

where I_s denotes the saturation current of the diode, V_T represents the thermal voltage, V is the supply voltage, and R is the circuit resistance. Solving the nonlinear equation $f(x) = 0$ yields the steady-state operating voltage of the nonlinear circuit.

For the numerical experiments, representative circuit parameters are selected as

$$I_s = 10^{-12}, \quad V_T = 0.0259, \quad R = 1000 \Omega, \quad V = 5 V.$$

The proposed $MSB_{q;\alpha}$ scheme is compared with the existing iterative methods $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$. The comparison is carried out in terms of convergence efficiency, numerical stability, and dynamical behavior.

5.7.1. Efficiency Analysis

The efficiency of the considered iterative schemes is evaluated using the absolute error at successive iterations. The numerical results are summarized in Table 11, where $\epsilon_k = |x_k - \zeta|$ denotes the absolute error at the k th iteration.

Table 11. Error comparison for different schemes for the diode circuit equations.

Method	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_5	ϵ_6
$SAN_{q;3}$	2.67×10^{-2}	1.12×10^{-4}	3.08×10^{-8}	1.84×10^{-15}	$5.93 \times 10^{-31\dagger}$	$3.21 \times 10^{-62\dagger}$
$SAN_{q;4}$	1.48×10^{-2}	4.62×10^{-5}	1.24×10^{-9}	$5.17 \times 10^{-18\dagger}$	$2.04 \times 10^{-36\dagger}$	$7.82 \times 10^{-72\dagger}$
$QSB_{q;4}$	7.15×10^{-4}	5.02×10^{-13}	$2.63 \times 10^{-26\dagger}$	$1.07 \times 10^{-52\dagger}$	$4.51 \times 10^{-139\dagger}$	0.0
$MSB_{q;\alpha}$	6.92×10^{-6}	$2.37 \times 10^{-23\dagger}$	$7.84 \times 10^{-47\dagger}$	$3.15 \times 10^{-93\dagger}$	$1.26 \times 10^{-241\dagger}$	0.0

[†] Computed using MATLAB VPA (digits=128, tol. 10^{-64})

From the results presented in Table 11, it can be observed that the proposed scheme $MSB_{q;\alpha}$ achieves substantially smaller absolute errors within a limited number of iterations when compared with the competing methods $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$.

Values reported as 0.0 correspond to errors below the working precision.

In particular, the $MSB_{q;\alpha}$ scheme rapidly reduces the approximation error to extremely small magnitudes under high-precision arithmetic, demonstrating improved convergence efficiency and strong numerical robustness for the nonlinear diode circuit equation.

5.7.2. Stability and Computational Cost

The stability and computational efficiency of the considered iterative schemes are evaluated using several performance indicators, including maximum absolute error, CPU time, memory usage, and the total number of arithmetic operations required during the iterative process. The corresponding results are summarized in Table 12, where the column "Max-Absolute Error" represents the maximum value of $|x_k - \zeta|$ obtained during the iteration process.

Table 12. Computational stability and cost comparison for the diode circuit problem.

Metric	Max-Error	CPU-time	Mem-Usage (MB)	$[\pm, \times, \div]$	Iterations (k)	Max- σ_i^{n-1}
SAN _{q:3}	2.38×10^{-4}	6.174	598.42	4631	11	4.9812
SAN _{q:4}	1.82×10^{-4}	5.768	534.66	3847	10	4.1725
QSB _{q:4}	3.19×10^{-15}	2.684	188.57	1745	7	6.7283
MSB _{q:\alpha}	$3.72 \times 10^{-289+}$	3.291	216.31	2038	9	5.8641

[†] Computed using MATLAB VPA (digits=128, tol. 10^{-64})

From the data reported in Table 12, it can be observed that the proposed MSB_{q:\alpha} scheme maintains extremely small absolute errors while requiring competitive computational resources. In particular, the proposed method exhibits moderate CPU time and memory consumption while preserving strong numerical stability.

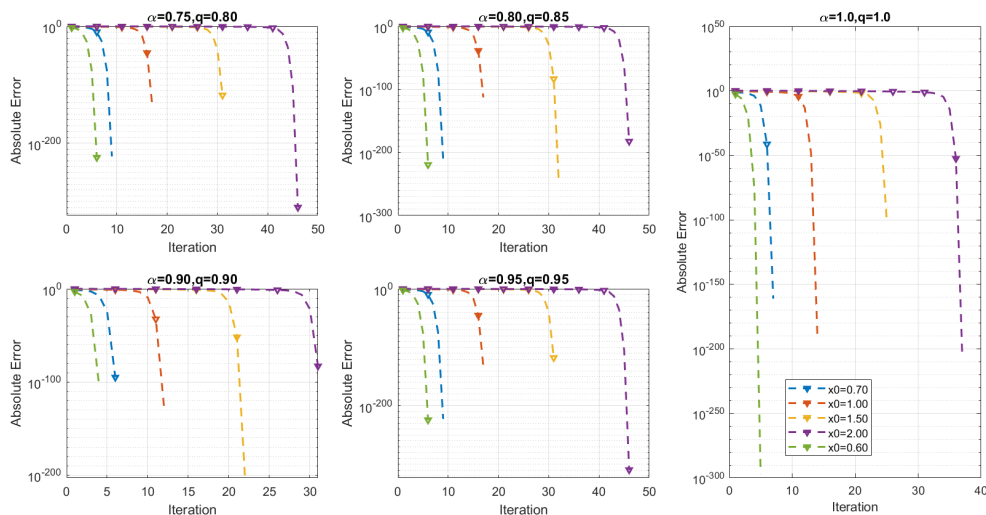
These results indicate that the MSB_{q:\alpha} method provides an efficient balance between numerical accuracy and computational cost when applied to nonlinear equations arising in diode circuit analysis.

5.7.3. Consistency for Fractional and Quantum Parameters

The robustness of the MSB_{q:\alpha} scheme is further investigated for different combinations of the fractional parameter α and the quantum parameter q . For the consistency analysis, the initial vector is chosen as

$$\mathbf{x}_0 = \text{vpa}([0.7, 1.0, 1.5, 2.0, 0.6]).$$

The numerical outcomes are reported in Table 13 and illustrated in Figure 8. The error curves displayed in Figure 8 indicate that the MSB_{q:\alpha} scheme exhibits rapid error decay and stable convergence behavior for different values of the fractional and quantum parameters.

**Figure 8.** Absolute error of the MSB_{q:\alpha} iterative scheme for various q values when solving (60).**Table 13.** Consistency of the MSB_{q:\alpha} method for various fractional and quantum parameters.

α	q	Iterations	Max-Absolute Error	CPU-time	Convergence
0.70	0.76	9	5.3×10^{-14}	2.72	Stable
0.82	0.85	8	7.4×10^{-16}	2.63	Stable
0.90	0.92	6	2.1×10^{-18}	2.51	Highly stable
0.96	0.97	6	8.7×10^{-21}	2.41	Highly stable
1.00	1.00	5	2.6×10^{-23}	2.34	Optimal

The error curves displayed in Figure 8 show that the $MSB_{q;\alpha}$ scheme exhibits rapid error decay and stable convergence behavior for different values of the fractional and quantum parameters. These observations confirm that the method maintains stable and consistent convergence across a wide range of parameter settings.

5.7.4. Dynamical Behavior and Computational Performance Comparison

The global dynamical behavior of the considered iterative schemes is analyzed through fractal basin representations, convergence–divergence distributions, and computational performance indicators. These graphical and quantitative analyses provide valuable insights into the stability, robustness, and efficiency of the iterative processes when applied to nonlinear equations.

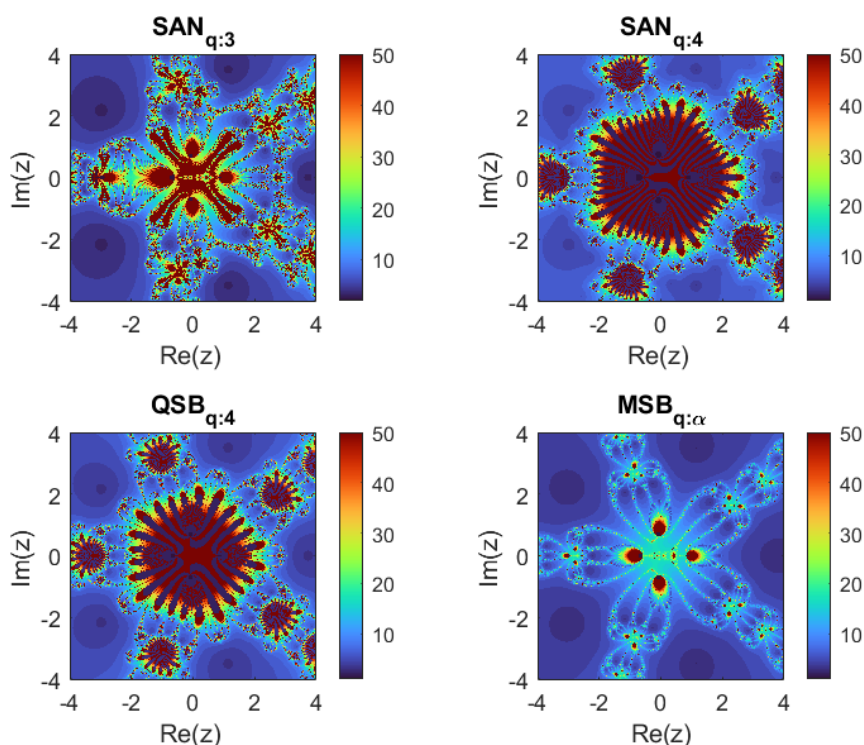


Figure 9. Fractal basins of attraction of the considered iterative schemes for $\alpha = 1$ and $q = 1$ when solving equation (60).

Figure 9 illustrates the fractal basins of attraction associated with the methods $SAN_{q:3}$, $SAN_{q:4}$, $QSB_{q:4}$, and the proposed $MSB_{q;\alpha}$ scheme. In these plots, each point in the complex plane represents an initial guess, while the corresponding color indicates the root toward which the iterative process converges. Regions with smooth color transitions correspond to stable convergence toward the root, whereas irregular or fragmented boundaries indicate sensitive dynamical behavior, where small perturbations in the initial guess may lead to different convergence outcomes or divergence.

A visual comparison of the fractal patterns reveals that the $MSB_{q;\alpha}$ scheme exhibits significantly larger and smoother convergence regions than the competing methods. The corresponding basin boundaries are also less irregular, indicating reduced sensitivity to the choice of initial guesses and improved numerical stability of the iterative process.

Table 14. Dynamical and computational comparison of the considered schemes.

Method	F-Resolution	C-Points	D-Points	E-Time (s)	O-Counts	CPU Time (s)	M-Usage (MB)	Overall Rank
$SAN_{q:3}$	Moderate	80%	20%	6.14	4587	5.88	598.4	4
$SAN_{q:4}$	Moderate	86%	14%	5.36	3875	5.09	534.7	3
$QSB_{q:4}$	High	91%	9%	4.02	2618	3.73	216.3	2
$MSB_{q;\alpha}$	Very High	97%	3%	2.58	1732	2.47	168.9	1

Table 14 summarizes several dynamical and computational indicators used to compare the considered iterative schemes. In particular, the percentage of convergence points (C-Points) and divergence points (D-Points) measures the stability of each method over a large set of initial guesses in the complex plane. Additional performance metrics, including execution time, arithmetic operation counts, CPU time, and memory usage, provide further information on the computational efficiency of the corresponding algorithms.

The results show that the $MSB_{q;\alpha}$ scheme achieves the highest convergence percentage (97%) and the lowest divergence rate (3%) among the tested methods. At the same time, it requires fewer arithmetic operations and exhibits lower execution time and memory usage. These characteristics confirm the improved efficiency of the proposed scheme compared with $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$.

Overall, the obtained results demonstrate that the Caputo fractional quantum iterative scheme $MSB_{q;\alpha}$ provides enhanced dynamical stability, larger convergence domains, and improved computational efficiency. Consequently, the proposed method represents a reliable and efficient numerical tool for solving nonlinear equations arising in electrical circuit analysis and other nonlinear dynamical systems.

6. Conclusion

This study introduced a novel Caputo fractional quantum iterative scheme, denoted by $MSB_{q;\alpha}$, for solving nonlinear equations. The proposed method combines the Caputo fractional operator with the quantum derivative framework to construct an efficient root-finding algorithm. This hybrid formulation introduces additional flexibility through the fractional parameter α and the quantum parameter q , enabling improved convergence behavior and enhanced numerical stability compared with existing iterative schemes such as $SAN_{q;3}$, $SAN_{q;4}$, and $QSB_{q;4}$.

Extensive numerical experiments (Table 3–14 and Figure 3–9) were conducted to validate the theoretical properties of the proposed method. The error analyses demonstrate that the $MSB_{q;\alpha}$ scheme exhibits faster error reduction and achieves high numerical accuracy within fewer iterations than the competing methods. Furthermore, the computational performance results indicate that the proposed scheme requires reduced CPU time, fewer arithmetic operations, and lower memory usage while maintaining high numerical precision (see e.g., Tables 4, 8 and 12, respectively).

The robustness of the method was further investigated through parametric and dynamic studies (Table 1 and Figures 1 and 2) involving different values of the fractional parameter α and the quantum parameter q . The results show that the $MSB_{q;\alpha}$ scheme maintains stable and consistent convergence across a wide range of parameter settings. Moreover, the dynamical and fractal analyses from Tables 6, 10, 14 and Figure 5, 7 and 9 reveal that the proposed method possesses larger convergence basins, reduced divergence regions, and improved global dynamical stability compared with the existing iterative schemes.

To illustrate its practical applicability, the proposed method was applied to nonlinear models arising in chemical reaction engineering, biomedical systems, and electrical circuit analysis. In all cases, the $MSB_{q;\alpha}$ scheme demonstrated superior convergence speed, improved stability, and higher computational efficiency compared with the considered numerical methods as presented in Tables 3, 4, 7, 8, 11, 12, Figure 4, 6 and 8 respectively.

Despite these promising results, several aspects deserve further investigation. The performance of the proposed method may depend on the appropriate selection of the fractional and quantum parameters, and the current formulation has been primarily developed for scalar nonlinear equations. Future work may therefore focus on extending the $MSB_{q;\alpha}$ framework to systems of nonlinear equations and large-scale multidimensional problems. In addition, adaptive strategies for selecting optimal fractional and quantum parameters could further improve the efficiency of the algorithm. Further studies may also explore deeper dynamical properties of the scheme, including high-resolution fractal basin structures and global convergence characteristics.

Overall, the proposed Caputo fractional quantum iterative scheme provides an efficient, stable, and computationally attractive approach for solving nonlinear equations. The integration of fractional and quantum operators opens promising directions for the development of advanced numerical algorithms in applied mathematics, computational science, and engineering applications.

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.

Data Availability Statement: The data supporting the findings of this study are included within this article.

Acknowledgments: 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).

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

Ethics Statements: All authors declare that this work complies with ethical guidelines set by the Committee on Publication Ethics (COPE).

Declaration of Competing Interest: The authors declare that they have no competing financial interests and personal relationships that could have appeared to influence the research in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

Appendix A. Polynomial Components of the Rational Map

For completeness, we provide the explicit expression of the polynomial $A(x; \beta_1; \beta_2; \alpha; q)$, $B(x; \beta_1; \beta_2; \alpha; q)$, $C(x; \beta_1; \beta_2; \alpha; q)$ and $D(x; \beta_1; \beta_2; \alpha; q)$ appearing in the rational map (44). This polynomial is obtained through symbolic simplification of the iterative scheme. To simplify the lengthy expression, we put $\alpha = 1, q = 1$

$$\begin{aligned}
A(x; \beta_1; \beta_2; 1; 1) = & x^{15} + (12\beta_1 - 12\beta_2 - 8)x^{14} + (48\beta_2^2 + 48\beta_1^2 - 72\beta_1 + 19)x^{13} \\
& + (-64\beta_2^3 + (192\beta_1 - 256)\beta_2^2 + (-192\beta_1^2 + 416\beta_1 - 24)\beta_2 \\
& + 192\beta_1^2 + 72\beta_1 - 32)x^{12} \\
& + (320\beta_2^3 + (-640\beta_1 - 336)\beta_2^2 + -128\beta_1^3 - 144\beta_1^2 - 168\beta_1 + 257)x^{11} \\
& + (896\beta_2^3 + (-2816\beta_1 + 1568)\beta_2^2 + -512\beta_1^3 - 768\beta_1^2 + 2676\beta_1 - 1568)x^{10} \\
& + (-512\beta_2^3 + (1280\beta_1 - 1248)\beta_2^2 + -1920\beta_1^3 + 9312\beta_1^2 - 13296\beta_1 + 5243)x^9 \\
& + (1152\beta_2^3 + (384\beta_1 - 6784)\beta_2^2 + +9088\beta_1^3 - 33792\beta_1^2 + 33648\beta_1 - 10920)x^8 \\
& + (6272\beta_2^3 + (-27648\beta_1 + 17632)\beta_2^2 + -24960\beta_1^3 + 57696\beta_1^2 - 50448\beta_1 + 15211)x^7 \\
& + (-2560\beta_2^3 + (14848\beta_1 - 16704)\beta_2^2 + +17920\beta_1^3 - 52608\beta_1^2 + 47988\beta_1 - 14728)x^6 \\
& + (2048\beta_2^3 + (-9472\beta_1 + 8432)\beta_2^2 + -9344\beta_1^3 + 29040\beta_1^2 - 30312\beta_1 + 10145)x^5 \\
& + (448\beta_2^3 + (-64\beta_1 - 2432)\beta_2^2 + +2112\beta_1^3 - 10560\beta_1^2 + 13128\beta_1 - 5040)x^4 \\
& + (64\beta_2^3 + (-384\beta_1 + 112)\beta_2^2 + -512\beta_1^3 + 2352\beta_1^2 - 3912\beta_1 + 1811)x^3 \\
& + (128\beta_2^3 + (-256\beta_1 + 32)\beta_2^2 + (128\beta_1^2 + 288\beta_1 - 308)\beta_2 - 384\beta_1^2 + \\
& 780\beta_1 - 464)x^2 \\
& + (-64\beta_2^2 + (64\beta_1 + 20)\beta_2 - 96\beta_1 + 81)x + 8\beta_2 - 8.
\end{aligned}$$

$$\begin{aligned}
B(x; \beta_1; \beta_2; 1; 1) = & x^4 + (4\beta_1 - 4\beta_2 - 4)x^3 + (-8\beta_1 + 4\beta_2 + 10)x^2 + (20\beta_1 - 12\beta_2 - 12)x \\
& - 4\beta_2 + 5.
\end{aligned}$$

$$\begin{aligned}
C(x; \beta_1; \beta_2; 1; 1) = & -\frac{1}{4} + \left(\beta_2 - \frac{5}{4}\right)x^4 + (-5\beta_1 + 3\beta_2 + 3)x^3 + \left(2\beta_1 - \beta_2 - \frac{5}{2}\right)x^2 \\
& + (-\beta_1 + \beta_2 + 1)x.
\end{aligned}$$

$$\begin{aligned}
D(x; \beta_1; \beta_2; 1; 1) = & \frac{1}{8} + (\beta_2 - 1)x^{15} + \left(-8\beta_2^2 + (8\beta_1 + \frac{5}{2})\beta_2 - 12\beta_1 + \frac{81}{8}\right)x^{14} \\
& + \left(16\beta_2^3 + (-32\beta_1 + 4)\beta_2^2 + (16\beta_1^2 + 36\beta_1 - \frac{77}{2})\beta_2 - 48\beta_1^2 + \frac{195}{2}\beta_1 - 58\right)x^{13} \\
& + \left(8\beta_2^3 + (-48\beta_1 + 14)\beta_2^2 + -64\beta_1^3 + 294\beta_1^2 - 489\beta_1 + \frac{1811}{8}\right)x^{12} \\
& + (56\beta_2^3 + (-8\beta_1 - 304)\beta_2^2 + +264\beta_1^3 - 1320\beta_1^2 + 1641\beta_1 - 630)x^{11} \\
& + (256\beta_2^3 + (-1184\beta_1 + 1054)\beta_2^2 + (2000\beta_1^2 - 3916\beta_1 + \frac{4113}{2})\beta_2 - \frac{10145}{8})x^{10} \\
& + (-320\beta_2^3 + (1856\beta_1 - 2088)\beta_2^2 + (-3536\beta_1^2 + 7272\beta_1 - \frac{6561}{2})\beta_2 - 1841)x^9 \\
& + (784\beta_2^3 + (-3456\beta_1 + 2204)\beta_2^2 + (5472\beta_1^2 - 7784\beta_1 + 3336)\beta_2 - 8)x^8 \\
& + (144\beta_2^3 + (48\beta_1 - 848)\beta_2^2 + (-1088\beta_1^2 + 3848\beta_1 - 1989)\beta_2 + 1136\beta_1^3 \\
& + (-64\beta_2^3 + (160\beta_1 - 156)\beta_2^2 + (48\beta_1^2 - 432\beta_1 + \frac{1111}{2})\beta_2 - 240\beta_1^3 + 1164\beta_1^2 \\
& + (112\beta_2^3 + (-352\beta_1 + 196)\beta_2^2 + (320\beta_1^2 - 268\beta_1 + \frac{49}{2})\beta_2 - 64\beta_1^3 - 96\beta_1^2 \\
& + (40\beta_2^3 + (-80\beta_1 - 42)\beta_2^2 + (56\beta_1^2 + 92\beta_1 - 50)\beta_2 - 16\beta_1^3 - 18\beta_1^2 \\
& - 21\beta_1 + \frac{257}{8})x^4 \\
& + (-8\beta_2^3 + (24\beta_1 - 32)\beta_2^2 + (-24\beta_1^2 + 52\beta_1 - 3)\beta_2 + 8\beta_1^3 - 24\beta_1^2 + 9\beta_1 - 4)x^3 \\
& + (6\beta_2^2 + (-12\beta_1 + \frac{19}{2})\beta_2 + 6\beta_1^2 - 9\beta_1 + \frac{19}{8})x^2 + \left(\frac{3\beta_1}{2} - \frac{3\beta_2}{2} - 1\right)x.
\end{aligned}$$

The remaining coefficients follow the same structural pattern and are omitted here for brevity. The complete symbolic expression can be obtained using a computer algebra system such as Mathematica or Maple.

References

1. Kelley, C. T. (1995). Iterative methods for linear and nonlinear equations. Society for Industrial and Applied Mathematics.

2. Costabile, F., Gualtieri, M. I., & Luceri, R. (2001). A new iterative method for the computation of the solutions of nonlinear equations. *Numerical Algorithms*, 28(1), 87-100.
3. Hasan, A. (2016). Numerical study of some iterative methods for solving nonlinear equations. *Int. J. Eng. Sci. Invent*, 5(2), 1-10.
4. Baleanu, D. (2012). *Fractional calculus: models and numerical methods* (Vol. 3). World Scientific.
5. Privman, V. (Ed.). (1990). *Finite size scaling and numerical simulation of statistical systems*. World Scientific.
6. Peng, B., & Zhang, Y. (2025). A Fractional Calculus Framework for Open Quantum Dynamics: From Liouville to Lindblad to Memory Kernels. arXiv preprint arXiv:2511.13038.
7. Siddi Moreau, G., Pisani, L., Profir, M., Podda, C., Leoni, L., & Cao, G. (2025). Quantum Artificial Intelligence Scalability in the NISQ Era: Pathways to Quantum Utility. *Advanced Quantum Technologies*, 8(10), 2400716.
8. Varshney, V., Kingston, S. L., Srinivasan, S., & Kumarasamy, S. (2024). Hidden attractors in fractional-order discrete maps. *The European Physical Journal B*, 97(10), 161.
9. Wang, X., & Zhang, J. (2024). Dynamic analysis and circuit design of tunable multi-vortex chaotic systems based on memristors. *Nonlinear Dynamics*, 112(16), 14415-14440.
10. Ali, N., Waseem, M., Safdar, M., Akgül, A., & Tolasa, F. T. (2024). Iterative solutions for nonlinear equations via fractional derivatives: adaptations and advances. *Applied Mathematics in Science and Engineering*, 32(1), 2333816.
11. Sana, Gul, Pshtiwan Othman Mohammed, Dong Yun Shin, Muhmmad Aslam Noor, and Mohammad Salem Oudat. "On iterative methods for solving nonlinear equations in quantum calculus." *Fractal and Fractional* 5, no. 3 (2021): 60.
12. Nonlaopon, K., Khan, A. G., Ameen, F., Awan, M. U., & Cesarano, C. (2022). Some new quantum numerical techniques for solving nonlinear equations. *Symmetry*, 14(9), 1829.
13. Nonlaopon, K., Khan, A. G., Ameen, F., Awan, M. U., & Cesarano, C. (2022). Some new quantum numerical techniques for solving nonlinear equations. *Symmetry*, 14(9), 1829.
14. Shams, M., & Carpentieri, B. (2024). Q-analogues of parallel numerical scheme based on neural networks and their engineering applications. *Applied Sciences*, 14(4), 1540.
15. Ali, N., Waseem, M., Safdar, M., Akgül, A., & Tolasa, F. T. (2024). Iterative solutions for nonlinear equations via fractional derivatives: adaptations and advances. *Applied Mathematics in Science and Engineering*, 32(1), 2333816.
16. Qureshi, S., Soomro, A., Argyros, I. K., Gdawiec, K., Akgül, A., & Alquran, M. (2025). Use of fractional calculus to avoid divergence in Newton-like solver for solving one-dimensional nonlinear polynomial-based models. *Communications in Nonlinear Science and Numerical Simulation*, 143, 108631.
17. Tay, T. T., Mareels, I., & Moore, J. B. (1998). Adaptive-Q application to nonlinear systems. In *High Performance Control* (pp. 205-240). Boston, MA: Birkhäuser Boston.
18. Candelario, G., Cordero, A., Torregrosa, J. R., & Vassileva, M. P. (2022). An optimal and low computational cost fractional Newton-type method for solving nonlinear equations. *Applied Mathematics Letters*, 124, 107650.
19. Shams, M., Kausar, N., & Carpentieri, B. (2025). A class of high-order fractional parallel iterative methods for nonlinear engineering problems: Convergence, stability, and neural network-based acceleration. *Chaos, Solitons & Fractals*, 199, 116646.
20. Akram, S., & Moin-Ud-Din Junjua, F. A. (2025). Iterative Techniques for Nonlinear Equations: Addressing Multiple Roots with Unknown Multiplicity. *Polynomials-Exploring Fundamental Mathematical Expressions: Exploring Fundamental Mathematical Expressions*, 33.
21. Batiha, I. M., Al-Khawaldeh, H. O., Jebiril, I. H., Anakira, N., Bataihah, A., & Momani, S. (2025). A Novel Numerical Approach for Handling Fractional Quantum Initial-Value Problems. *International Journal of Fuzzy Logic and Intelligent Systems*, 25(4), 416-430.
22. Kac, V. G., & Cheung, P. (2002). *Quantum calculus* (Vol. 113). New York: Springer.
23. Elsayir, H. A., & Altoum, S. H. (2026). Formulation of the Schrödinger Equation Using the Jackson q-Derivative. *Journal of Computational Analysis and Applications*, 35(01).
24. Jackson, F. H. (1910). q-Difference equations. *American journal of mathematics*, 32(4), 305-314.
25. Njionou Sadjang, P. (2018). On the fundamental theorem of (p, q)-calculus and some (p, q)-Taylor formulas. *Results in Mathematics*, 73(1), 39.
26. Askey, R. (1978). The q-gamma and q-beta functions. *Applicable Analysis*, 8(2), 125-141.
27. Kalla, S., Yadav, R., & Purohit, S. (2005). On the Riemann-Liouville fractional q-integral operator involving a basic analogue of Fox H-function. *Fractional Calculus and Applied Analysis*, 8(3), 313-322.

28. Shaimardan, S., & Tokmagambetov, N. S. (2021). On the solutions of some fractional q -differential equations with the Riemann-Liouville fractional q -derivative. *Bulletin of the Karaganda University. Mathematics Series*, 104(4), 130-141.
29. Deppman, A., Megías, E., & Pasechnik, R. (2023). Fractal derivatives, fractional derivatives and q -deformed calculus. *Entropy*, 25(7), 1008.
30. Allouch, N., Batiha, I. M., Jebri, I. H., Hamani, S., Al-Khateeb, A., & Momani, S. (2024). A new fractional approach for the higher-order q -Taylor method. *Image Analysis and Stereology*, 43(3), 249-257.
31. Beardon, A. F. (2000). *Iteration of rational functions: Complex analytic dynamical systems* (Vol. 132). Springer Science & Business Media.
32. Argyros, I. K., & Magreñán, Á. A. (2017). *Iterative Methods and their dynamics with applications: A Contemporary Study*. CRC Press.
33. Fogler, H. S. (1999). *Elements of chemical reaction engineering*. Pearson Education.
34. Bergman, R. N., Phillips, L. S., & Cobelli, C. (1981). Physiologic evaluation of factors controlling glucose tolerance in man: measurement of insulin sensitivity and beta-cell glucose sensitivity from the response to intravenous glucose. *The Journal of clinical investigation*, 68(6), 1456-1467.
35. Chua, L. O. (2004). Nonlinear circuit foundations for nanodevices. I. The four-element torus. *Proceedings of the IEEE*, 91(11), 1830-1859.

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.