Chaos, Solitons and Fractals 199 (2025) 116646

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

A class of high-order fractional parallel iterative methods for nonlinear engineering problems: Convergence, stability, and neural network-based acceleration

Mudassir Shams [a,*], Nasreen Kausar [a], Bruno Carpentieri [b]

[a] Department of Mathematics, Faculty of Arts and Science, Balikesir University, Balikesir 10145, Turkey
[b] Faculty of Engineering, Free University of Bozen-Bolzano, 39100, Bolzano, Italy

ARTICLE INFO

Keywords:
Fractional scheme
Fractal
Computational analysis
Convergence theorem
Engineering applications

ABSTRACT

Conventional analytical techniques often fail to yield efficient or closed-form solutions for nonlinear fractional problems due to their inherent nonlocality and complexity. This study introduces a new class of high-order parallel iterative methods for solving nonlinear equations, with a focus on fractional-order formulations. We first develop a sixth-order single-root finding scheme, which is then extended to a fractional-order method with convergence order $5\sigma + 1$, and further generalized into a parallel scheme achieving order $20\sigma + 8$. To improve computational performance, we propose a hybrid neural network-based parallel scheme, in which optimal parameter values are identified through dynamical systems analysis. The resulting methods exhibit strong stability, accuracy, and efficiency, and are robust with respect to both accurate and perturbed initial approximations. Comparative experiments on real-world engineering problems demonstrate that the proposed fractional parallel schemes consistently outperform existing methods in terms of residual error, convergence rate, and computational cost.

1. Introduction

Fractional-order differential equations (FDEs), grounded in fractional calculus, offer powerful tools for modeling complex systems that exhibit memory and hereditary effects—capabilities that often surpass those of classical integer-order models [1–3]. These equations have become fundamental in numerous scientific and engineering fields, including viscoelasticity, fluid dynamics, and electrical circuit analysis, where they provide more accurate descriptions of material behavior and dynamic processes. Furthermore, FDEs have greatly enhanced our understanding of nonlocal phenomena and anomalous diffusion in physical, chemical, and biological systems. Their ability to simultaneously capture local and global dynamics makes them particularly valuable for applications in system optimization and control, such as signal processing and control theory.

A general form of a fractional differential equation can be expressed as:

$$\begin{cases} {}_{\zeta}\mathfrak{D}_{\sigma_1}^{n\sigma} U(x) + \vartheta^{[*]} U(x) = Q(x,t); x_0 \le x \le x_n \\ U(x_0) = U_0^{[*]}, \\ U'(x_0) = U_1^{[*]}, \\ U''(x_0) = U_2^{[*]}, \\ \vdots \\ U^{(n-1)}(x_0) = U_{n-1}^{[*]}, \end{cases} \tag{1}$$

where $x \in [x_0, x_n]$. Solving nonlinear fractional differential equations of the general form

$$\wp(x) = 0, \tag{2}$$

* Corresponding author.
E-mail addresses: mudassir.shams@balikesir.edu.tr (M. Shams), nasreen.kausar@balikesir.edu.tr (N. Kausar), bruno.carpentieri@unibz.it (B. Carpentieri).

https://doi.org/10.1016/j.chaos.2025.116646
Received 7 April 2025; Received in revised form 16 May 2025; Accepted 24 May 2025
Available online 23 June 2025

is particularly challenging due to the nonlocal nature of fractional derivatives, which depend on the entire history of the function rather than solely on its local behavior. This intrinsic memory effect complicates the derivation of closed-form solutions and renders traditional analytical methods ineffective in most practical scenarios. Although some exact solutions [4] and analytical techniques [5,6] have been developed, they are typically limited to idealized cases and offer limited applicability to complex real-world systems.

To overcome these limitations, researchers have increasingly turned to numerical techniques. Direct numerical methods, including explicit single-step [7], multi-step [8], and hybrid block approaches [9], have been proposed to approximate solutions. However, these methods often suffer from high computational costs, numerical instability, and sensitivity to small errors, particularly in the presence of stiff or highly nonlinear problems. To improve accuracy and stability, iterative schemes have emerged as a promising alternative. Classical iterative techniques, such as those of Newton [10], Halley [11], Ostrowski [12], King [13], Jarratt [14], and their variants (see, e.g., [15–17] and references therein), have been widely employed to compute a single solution to nonlinear problems at a time. While these methods often exhibit a high local order of convergence, they typically require the exact evaluation of first- and second-order derivatives, which may become unstable when the function or its derivatives exhibit nonsmooth behavior near the solution. Moreover, these techniques are inherently local and sensitive to initial guesses, especially in the presence of multiple or clustered solutions, as discussed by Ortega and Rheinboldt [18] and Deuflhard [19]. To address these challenges, several approaches have extended classical root-finding techniques using fractional derivatives [20–23], achieving improved convergence properties and enhanced numerical stability. By leveraging the inherent advantages of fractional calculus, these iterative schemes provide a more robust framework for solving nonlinear fractional equations, making them particularly suitable for applications in engineering and the applied sciences. As a result, various researchers have developed fractional-order iterative methods based on Caputo and Riemann–Liouville derivatives. Notable examples include the work of Çelik et al. [24] (Caputo), Shams et al. [25] (single- and multi-step Caputo schemes), Hernández et al. [26] (Riemann–Liouville), Candelario et al. [27] (Caputo, conformable, and Riemann–Liouville-based schemes), and Gdawiec et al. [28] (fractional Newton with fixed-point iteration).

Most fractional derivatives — except for the Caputo derivative — do not satisfy the property:

$$\left[\eth^{\sigma}\right](C) = 0, \tag{3}$$

when σ is not a natural number and $C \in \mathbb{R}$. This property is fundamental, as it guarantees that the fractional derivative of a constant vanishes, thereby making the Caputo derivative particularly suitable for initial value problems. In contrast to other fractional derivatives, which typically require nonstandard formulations to enforce boundary conditions, the Caputo derivative preserves natural compatibility with classical initial conditions. This feature is especially advantageous in the context of iterative numerical methods, which often depend on well-posed initial conditions to ensure stability and convergence. Leveraging this key property, the present study proposes a fractional iterative method based on the Caputo derivative. This approach extends classical root-finding techniques while preserving consistency with standard boundary formulations. As a result, the method achieves improved accuracy and computational efficiency in solving nonlinear equations, particularly in engineering applications where reliability and robustness are essential.

Existing fractional-order methods typically exhibit local convergence behavior and often require highly accurate initial guesses to solve (2) successfully. To address this limitation, parallel iterative schemes have been explored. Algorithms such as the Weierstrass–Durand–Kerner method [29] and its various modifications — for instance, those proposed by Petković et al. [30], Lopes et al. [31], Chinesta et al. [32], Bhalla et al. [33], Shams et al. [34], and Cordero et al. [35] — are widely employed to compute multiple or distinct solutions of nonlinear problems by refining all estimates simultaneously and exhibiting global convergence properties. Recent developments have extended these methods to the fractional-order setting [36] and combined them with hybrid techniques [37–39]. However, despite their widespread use, parallel methods remain highly dependent on good initial approximations and may suffer from slow convergence, instability, or even divergence when applied to highly nonlinear problems with clustered solutions. Notably, little attention has been devoted to improving these initial approximations or ensuring global convergence under such conditions. To address this gap, and building on the aforementioned methods, we first introduce a novel fractional-order scheme designed to approximate one solution at a time. The method is then extended to a fractional-order parallel framework that alleviates memory sensitivity, improves robustness to initial conditions, and approximates all solutions simultaneously. Additionally, we propose a hybrid neural network-based fractional-order parallel scheme that improves convergence speed, efficiency, and stability in solving complex nonlinear engineering problems, while preserving global convergence behavior. This work makes several key contributions to the field of fractional numerical analysis:

- Development of high-order fractional schemes for solving nonlinear equations, yielding significant improvements in both accuracy and computational efficiency.
- Rigorous convergence analysis of the proposed methods, including the identification of optimal parameter values through complex dynamical analysis.
- Design and refinement of parallel root-finding algorithms, enabling their application to polynomials with multiple roots and improving stability, robustness to initial guesses, and convergence behavior.
- Integration of parallel processing techniques to accelerate convergence and support large-scale nonlinear problems in engineering and applied sciences.
- Comprehensive performance evaluation and benchmarking against existing fractional-order methods, with respect to computational efficiency, accuracy, and memory usage.
- Application of the proposed techniques to challenging nonlinear problems in science and engineering, highlighting their practical utility and broad applicability.

The remainder of this paper is organized as follows. Section 2 introduces fundamental concepts from fractional calculus, including the Gamma function, the Caputo fractional derivative, and the generalized Taylor expansion, which provide the theoretical foundation for the proposed methods. Section 3 describes the construction of the fractional and fractional-parallel schemes and presents their local convergence analysis. Section 4 provides a detailed dynamical analysis emphasizing the stability of the designed scheme and the hybrid neural network-based fractional-order parallel schemes. Section 5 applies the proposed methods to two engineering problems involving fractal analysis, demonstrating their effectiveness. Finally, Section 7 concludes the paper with a summary of the main results and final remarks.

2. Preliminaries on fractional calculus

Before presenting the proposed fractional iterative schemes, we provide essential background on fractional calculus, highlighting the definitions and properties relevant to the development of our numerical methods. This section introduces the Gamma function and the Caputo fractional derivative, both of which play a central role in the formulation of the method. In addition, we present the generalized Taylor expansion in the fractional setting, which forms the basis for the subsequent convergence analysis.

Definition: The Gamma function

The Gamma function, also known as the generalized factorial function [40], is defined as:

$$\Gamma(x) = \int_0^{+\infty} u^{x-1} e^{-u} du, \tag{4}$$

where $x > 0$. It satisfies the fundamental properties:

$$\Gamma(1) = 1, \quad \Gamma(n+1) = n! \quad \text{for} \quad n \in \mathbb{N}.$$

Definition: The Caputo fractional derivative

Let $\wp : \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function such that $\wp \in C^{+\infty}([\sigma, x])$, where $-\infty < \sigma < x < +\infty$, $\sigma \geq 0$, and $m = \lfloor \sigma \rfloor + 1$. The Caputo fractional derivative [41] of order σ is defined as:

$$\left[{}_C\partial_{\sigma_1}^{\sigma}\right] \wp(x) = \begin{cases} \frac{1}{\Gamma(m-\sigma)} \int_0^x \frac{d^m}{dt^m} \wp(t) \frac{1}{(x-t)^{\sigma-m+1}} dt, & \sigma \in N, \\ \frac{d^{m-1}}{dt^{m-1}} \wp(x), & \sigma = m-1 \in N \cup \{0\}, \end{cases} \tag{5}$$

where $\Gamma(x)$ is the Gamma function with $x > 0$.

Generalized Taylor expansion in the fractional setting

The generalized Taylor formula extends the classical Taylor series to fractional-order derivatives, offering a more flexible framework for function approximation in fractional calculus. This expansion plays a central role in the convergence analysis of the proposed iterative methods.

Theorem 1. *Suppose* $\left[{}_C\partial_{\sigma_1}^{\gamma\sigma}\right] \wp(x) \in \hbar\left([\sigma_1, \sigma_2]\right)$ *for* $\gamma = 1, \ldots, n+1$ *where* $\sigma \in (0, 1]$. *The Generalized Taylor Formula [42] is defined as*

$$\wp(x) = \sum_{i=0}^{n} \left[{}_C\partial_{\sigma_1}^{i\sigma}\right] \wp(\sigma_1) \frac{(x-\sigma_1)^{i\sigma}}{\Gamma(i\sigma+1)} + \left[{}_C\partial_{\sigma_1}^{(n+1)\sigma}\right] \wp(\xi) \frac{(x-\sigma_1)^{(n+1)\sigma}}{\Gamma((n+1)\sigma+1)}, \tag{6}$$

where

$$\sigma_1 \leq \xi \leq x, \ \forall x \in (\sigma_1, \sigma_2]. \tag{7}$$

Additionally, the following identity holds:

$$\left[{}_C\partial_{\sigma_1}^{n\sigma}\right] = \left[{}_C\partial_{\sigma_1}^{\sigma}\right] \cdot \left[{}_C\partial_{\sigma_1}^{\sigma}\right] \ldots \left[{}_C\partial_{\sigma_1}^{\sigma}\right] (n \text{ times}). \tag{8}$$

Caputo-type Taylor expansion.

We derive a Taylor expansion based on the Caputo fractional derivative to approximate functions in the fractional setting. This expansion provides a fundamental tool for the convergence and error analysis of the proposed methods. We consider the expansion of $\wp(x)$ around $\sigma_1 = \xi$ as follows:

$$\wp(x) = \frac{\left[{}_C\partial_{\xi}^{\sigma}\right] \wp(\xi)}{\Gamma(+1)} (x-\xi)^{\sigma} + \frac{\left[{}_C\partial_{\xi}^{2\sigma}\right] \wp(\xi)}{\Gamma(2\sigma+1)} (x-\xi)^{2\sigma} + O(x-\xi)^{3\sigma}. \tag{9}$$

By factoring out the leading term $\frac{\left[{}_C\partial_{\xi}^{1\sigma}\right]\wp(\xi)}{\Gamma(\sigma+1)}$, we rewrite the expansion as:

$$\wp(x) = \frac{\left[{}_C\partial_{\xi}^{\sigma}\right] \wp(\xi)}{\Gamma(\sigma+1)} \left[(x-\xi)^{\sigma} + \hbar_2 (x-\xi)^{2\sigma}\right] + O(x-\xi)^{3\sigma}, \tag{10}$$

where

$$\hbar_{\gamma} = \frac{\Gamma(\sigma+1)}{\Gamma(\gamma\sigma+1)} \frac{\left[{}_C\partial_{\xi}^{\gamma\sigma}\right] \wp(\xi)}{\left[{}_C\partial_{\xi}^{\sigma}\right] \wp(\xi)}, \gamma \geq 2. \tag{11}$$

The coefficient $\hbar_{\gamma}$ serves as a correction term that refines the higher-order contributions in the expansion. The corresponding approximation of the Caputo-type fractional derivative around $x = \xi$ is given by:

$$\left[{}_C\partial_{\xi}^{\sigma}\right] \wp(x) = \frac{\left[{}_C\partial_{\xi}^{\sigma}\right] \wp(\xi)}{\Gamma(\sigma+1)} \left[\Gamma(\sigma+1) + \frac{\Gamma(2\sigma+1)}{\Gamma(\sigma+1)} \hbar_2 (x-\xi)^{\sigma}\right] + O(x-\xi)^{2\sigma}. \tag{12}$$

These results serve as a foundation for analyzing the convergence properties of the proposed numerical methods.

3. Design and convergence analysis of a sixth-order Caputo-type iterative scheme

This section presents the formulation and theoretical analysis of numerical schemes for computing a single root of a nonlinear equation using the Caputo fractional derivative. The approach is then extended to simultaneous schemes for approximating all roots, and their theoretical convergence properties are analyzed.

3.1. Derivation of the sixth-order fractional iterative scheme

Iterative root-finding methods are essential for solving nonlinear equations, especially when analytical solutions are not available. Among classical approaches, the Newton–Raphson method [43] is widely studied for its efficiency and exhibits local quadratic convergence under appropriate conditions. Numerous modifications and extensions have been proposed to improve its accuracy, stability, and convergence behavior. The standard Newton–Raphson iteration is given by:

$$x^{[j+1]} = x^{[j]} - \frac{\wp(x^{[j]})}{\wp'(x^{[j]})}. \tag{13}$$

Method (13) approximates the roots of Eq. (2) through a single-step iteration exhibiting quadratic convergence. To improve its efficiency, various modifications have been proposed, including those by Kou and Li [44], Noor et al. [45], and Chicharro et al. [46]. These enhancements introduce weight functions, correction terms, and multi-step formulations aimed at accelerating convergence and improving stability.

Kou and Li [44] introduced an improved root-finding scheme that also exhibits quadratic convergence ($\mathcal{O}(2)$):

$$x^{[j+1]} = x^{[j]} - \left(1 + \frac{\lambda_1 \left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})}\right)}{1 + \left(\lambda_1 + \lambda_2\right)\left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})}\right)}\right) \frac{\left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})}\right)}{1 + \lambda_1 \left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})}\right)}, \tag{14}$$

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

Similarly, Noor et al. [45] proposed an alternative iterative method that also achieves quadratic convergence, i.e., of order $\mathcal{O}(2)$:

$$x^{[j+1]} = x^{[j]} - \frac{2\left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})}\right)}{1 + \sqrt{1 + 4\gamma^{[*]}\wp(x^{[j]})\left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})}\right)^2}}, \tag{15}$$

where $\gamma^{[*]} \in \mathbb{R}$.

Using a weight function technique, Chicharro et al. [46] later rederived both methods (14) and (15). Further extensions of method (13) have led to the development of multi-step iterative techniques. Notably, Chun et al. [47] proposed the following two-step method, which achieves fourth-order convergence, i.e., of order $\mathcal{O}(4)$:

$$x^{[j+1]} = x^{[j]} + \left(1 + \lambda_1\right)\left(\frac{\wp(x^{[j]}) + \wp(y^{[j]})}{\wp'(x^{[j]})}\right) - 2\left(\frac{\left(\wp(x^{[j]})\right)^2}{\wp'(x^{[j]})\left(\wp(x^{[j]}) - \wp(y^{[j]})\right)}\right) - \lambda_1 B^{[*]}, \tag{16}$$

where

$$y^{[j]} = x^{[j]} - \frac{\wp(x^{[j]})}{\wp'(x^{[j]})}, \quad B^{[*]} = \left(\frac{\wp(x^{[j]})}{\wp'(x^{[j]})} + \frac{\wp'(x^{[j]})\wp(y^{[j]})}{\left(\wp(x^{[j]})\right)^2 + \left(\wp'(x^{[j]})\right)^2}\right).$$

For $\lambda_1 = 0$, this method simplifies to the scheme proposed by Jisheng et al. [48]:

$$x^{[j+1]} = x^{[j]} - \left(\frac{\left(\wp(x^{[j]})\right)^2 + \left(\wp(y^{[j]})\right)^2}{\wp'(x^{[j]})\left(\wp(x^{[j]}) - \wp(y^{[j]})\right)}\right), \tag{17}$$

where

$$y^{[j]} = x^{[j]} - \frac{\wp(x^{[j]})}{\wp'(x^{[j]})}.$$

These techniques, along with several others (see, e.g., [49–51] and references therein), improve both the convergence order and computational efficiency of the classical Newton–Raphson method (13). It is important to note, however, that while these methods enhance root-finding performance for general nonlinear equations, they are formulated for integer-order problems and do not directly extend to fractional differential equations. The next section addresses this limitation by introducing a novel Caputo-type fractional iterative scheme that adapts Newton-like methods to the fractional setting.

Building on classical root-finding methods, we now introduce the iterative scheme (18), which is designed to achieve higher-order accuracy:

$$v^{[j]} = z^{[j]} - \frac{\wp(z^{[j]})}{\wp'(y^{[j]}) - \vartheta^{[*]}\wp(y^{[j]})}, \tag{18}$$

where $y^{[j]} = x^{[j]} - \frac{\wp(x^{[j]})}{\wp'(x^{[j]})}$, $z^{[j]} = y^{[j]} - \frac{\wp(y^{[j]})}{\wp'(y^{[j]})}$, and $\vartheta^{[*]} \in \mathbb{R}$. The sixth-order convergence of this method is formally established in the following theorem.

Theorem 2. *Let*

$$\wp : \supset \subseteq \mathbb{R} \to \mathbb{R} \tag{19}$$

be a continuous function, and let ξ be the exact root of $\wp(x)$, i.e., $\wp(\xi)=0$. Suppose that $\wp'(x^{[j]})$ is of order $\gamma\sigma$ for some $\gamma \geq 0$ and $\sigma \in (0,1]$. If $\{x^{[j]}\}$ is a sequence generated by the Caputo-type fractional iterative scheme (18) with an initial guess x_0 sufficiently close to ξ, then the sequence converges to ξ with at least order 6, and the associated error equation is given by:

$$\epsilon_*^{[j]} = \left(-\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4 + 2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5\right)\left[\epsilon^{[j]}\right]^6 + O\left(\left[\epsilon^{[j]}\right]^7\right). \tag{20}$$

Proof. Let ξ be a root of $\wp(x)$ and define the error term as $x^{[j]} = \xi + \epsilon^{[j]}$. By expanding $\wp(x^{[j]})$ and its derivative $\wp'(x^{[j]})$ in a Taylor series around $x=\xi$ and using the assumption $\wp(\xi)=0$, we obtain:

$$\wp(x^{[j]}) = \begin{bmatrix} \epsilon^{[j]} + \left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^2 + \left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^3 + \\ \left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^4 + \left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^5 + \left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^6 + \cdots \end{bmatrix}. \tag{21}$$

Differentiating (21), we obtain:

$$\wp'(x^{[j]}) = \begin{bmatrix} 1 + 2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\epsilon^{[j]} + 3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^2 + \\ 4\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^3 + 5\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^4 + 6\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^5 + \cdots \end{bmatrix}. \tag{22}$$

Expanding the inverse of (22) via Taylor expansion, we obtain:

$$\begin{aligned} \frac{1}{\wp'(x^{[j]})} &= 1 - 2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\epsilon^{[j]} + \left(4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 - 3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^2 + \\ &\quad \begin{bmatrix} 6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - 4\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) + \\ 2\left(-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 + 3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right) \end{bmatrix}\left[\epsilon^{[j]}\right]^3 + \\ &\quad \begin{bmatrix} 8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) - 5\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) + \\ 3\left(-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 + 3\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) + \\ 2\left(8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 - 12\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) + 4\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\right)\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right) \end{bmatrix}\left[\epsilon^{[j]}\right]^4 + \\ &\quad + \begin{bmatrix} 10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) - 6\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right) + \\ 4\left(-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 + 3\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) + \\ 3\begin{bmatrix} 8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 - \\ 12\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) + 4\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \end{bmatrix}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \\ +2\begin{bmatrix} -16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4 + \\ 36\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - 16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ -9\frac{\wp'''(\xi)^2}{3!\wp'(\xi)} + 5\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \end{bmatrix}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right) \end{bmatrix}\left[\epsilon^{[j]}\right]^5 + \cdots. \end{aligned} \tag{23}$$

By multiplying (22) and (23), we derive the expression:

$$\begin{aligned} \frac{\wp(x^{[j]})}{\wp'(x^{[j]})} &= \epsilon^{[j]} + \left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^2 + \left(2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 - 2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^3 - \\ &\quad \left(-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 + 7\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - 3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^4 + \\ &\quad \begin{bmatrix} 8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4 - 20\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) + \\ 10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) + 6\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2 - 4\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \end{bmatrix}\left[\epsilon^{[j]}\right]^5 + \\ &\quad \begin{bmatrix} -16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5 + 52\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \\ -28\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ -33\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2 + 13\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \\ +17\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) - 5\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right) \end{bmatrix}\left[\epsilon^{[j]}\right]^6 + \cdots. \end{aligned} \tag{24}$$

In the first step of (18), we use (24) to determine the error equation of (18) as

$$y^{[j]} = \xi + \left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^2 + \left(-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 + 2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^3 +$$

$$\left(4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3-7\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^4+\begin{pmatrix}-8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+20\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\\-10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\hbar_4\\-6\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2+4\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)\end{pmatrix}\left[\epsilon^{[j]}\right]^5+\begin{pmatrix}16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5-52\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\\+28\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\\+33\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2-13\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)\\-17\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+5\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right)\end{pmatrix}\left[\epsilon^{[j]}\right]^6+\cdots. \tag{25}$$

Expanding $\wp(y^{[j]})$ around ξ using a Taylor series yields:

$$\wp(y^{[j]})=\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^2+\left(-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2+2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^3+\begin{pmatrix}5\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^3-7\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\\+3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\end{pmatrix}\left[\epsilon^{[j]}\right]^4+\begin{pmatrix}-12\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+24\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)-\\10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)-6\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2+4\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)\end{pmatrix}\left[\epsilon^{[j]}\right]^5+\cdots. \tag{26}$$

Taking the derivative of (26), we have

$$\wp'(y^{[j]})=1+2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left[\epsilon^{[j]}\right]^2+2\left(-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2+2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^3+\begin{pmatrix}2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\begin{pmatrix}4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3-\\7\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\end{pmatrix}+\\3\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\end{pmatrix}\left[\epsilon^{[j]}\right]^4+\begin{pmatrix}2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\\\begin{pmatrix}-8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+20\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)-\\10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)-\\6\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2+4\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)\end{pmatrix}\\+6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\begin{pmatrix}-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\\+2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\end{pmatrix}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\end{pmatrix}\left[\epsilon^{[j]}\right]^5+\cdots. \tag{27}$$

Inverse of (27), is given as

$$\frac{1}{\wp'(y^{[j]})}=1-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left[\epsilon^{[j]}\right]^2-2\begin{pmatrix}-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2+\\2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\end{pmatrix}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^3+\begin{pmatrix}-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3-7\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\right)\\-3\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4\end{pmatrix}\left[\epsilon^{[j]}\right]^4+$$

$$\left(\begin{array}{c} -2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\begin{array}{c} -8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+20\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \\ -10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ -6\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2+4\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \end{array}\right)- \\ 6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\begin{array}{c} -2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 \\ +2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+ \\ 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2+2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right) \\ -8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2-\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right) \end{array}\right)\left[\epsilon^{[j]}\right]^5+\cdots . \tag{28}$$

The result of multiplying (27) and (28) is

$$\begin{aligned}\frac{\wp(y^{[j]})}{\wp'(y^{[j]})} &= +\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left[\epsilon^{[j]}\right]^2+\left(-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2+2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^3+ \\ &\left(3\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3-7\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^4+ \\ &\left(\begin{array}{c} -4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)- \\ 10\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)-6\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2+4\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^5+ \\ &\left(\begin{array}{c} -22\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5+41\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)- \\ 12\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)-8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^6+ \\ &\left(\begin{array}{c} 52\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^6-138\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+ \\ 64\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ +62\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2- \\ 16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)- \\ 24\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^7+\cdots \end{aligned} \tag{29}$$

Using (29) in the second-step of (18), to determine its corresponding error

$$\begin{aligned} z^{[j]} &= \xi+\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left[\epsilon^{[j]}\right]^4+\left(\begin{array}{c} -4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+ \\ 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^5+ \\ &\left(\begin{array}{c} 38\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5-93\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+40\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ +41\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)-13\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \\ -17\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+5\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^6+ \\ &\left(\begin{array}{c} -52\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^6+138\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \\ -64\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ -62\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2+16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \\ +24\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^7+O\left(\left[\epsilon^{[j]}\right]^8\right) \end{aligned} \tag{30}$$

Expanding $\wp(z^{[j]})$ around ξ using Taylor series yields:

$$\wp(z^{[j]})=\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left[\epsilon^{[j]}\right]^4-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2-\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\right)\left[\epsilon^{[j]}\right]^5$$

$$\left(\begin{array}{c} 38\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5-93\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+ \\ 40\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \\ +41\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2-13\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)- \\ 17\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+5\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^6+\cdots \tag{31}$$

Therefore

$$\begin{aligned}\wp'(y^{[j]})-\vartheta^{[*]}\wp(y^{[j]}) &= 1+\left(-\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)+2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\right)\left[\epsilon^{[j]}\right]^2+ \\ &\left(\begin{array}{c} 2\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3-2\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right) \\ +4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^3 \\ &\left(\begin{array}{c} -5\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3+8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+7\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)- \\ 11\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)-3\vartheta^{[*]}\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^4+ \\ &\left(\begin{array}{c} 12\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4-16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5- \\ 24\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \\ +28\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+ \\ 10\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+6\vartheta^{[*]}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2- \\ 20\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)- \\ 4\vartheta^{[*]}\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)+8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^5+\cdots \end{aligned} \tag{32}$$

Taking inverse of (32), we have

$$\begin{aligned}\frac{1}{\wp'(y^{[j]})-\vartheta^{[*]}\wp(y^{[j]})} &= 1+\left(\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)-2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\right)\left[\epsilon^{[j]}\right]^2+ \\ &\left(\begin{array}{c} -2\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2+4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3+ \\ 2\vartheta^{[*]}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)-4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^3+ \\ &\left(\begin{array}{c} 5\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3-8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4-7\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+ \\ 11\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+3\vartheta^{[*]}\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)- \\ 6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+\left(-\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)+2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\right)^2 \end{array}\right)\left[\epsilon^{[j]}\right]^4+ \\ &\left(\Lambda^{\{1\}}-\Lambda^{\{2\}}+\Lambda^{\{3\}}+\Lambda^{\{4\}}+\Lambda^{\{5\}}\right)\left[\epsilon^{[j]}\right]^5+ \\ &\left(\begin{array}{c} \left(-\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)+2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\right) \\ \Lambda^{\{6\}}+\Lambda^{\{7\}}+\Lambda^{\{8\}}+\Lambda^{\{9\}} \end{array}\right){e_n}^6+\cdots \end{aligned} \tag{33}$$

The following is the result of multiplying (32) with (33):

$$\begin{aligned}\frac{\wp(z^{[j]})}{\wp'(y^{[j]})-\vartheta^{[*]}\wp(y^{[j]})} &= \left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left[\epsilon^{[j]}\right]^4+\left(\begin{array}{c} -4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+ \\ 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^5+ \\ &\left(\begin{array}{c} \vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4+36\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5-93\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)+ \\ 40\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+41\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2- \\ 13\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right)-17\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right)+5\left(\frac{\wp^{vi}(\xi)}{6!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^6+ \\ &\left(\begin{array}{c} -6\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5+12\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^6 \\ +6\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \\ -12\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{array}\right)\left[\epsilon^{[j]}\right]^7+O\left(\left[\epsilon^{[j]}\right]^8\right) \end{aligned} \tag{34}$$

Using Eq. (34) in the final step of the single-root finding scheme (18), we obtain the following error equation:

$$\upsilon^{[j]} = \xi + \left(-\vartheta^{[*]} \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^4 + 2 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^5 \right) \left[\epsilon^{[j]}\right]^6 + \left(\begin{array}{c} 6\vartheta^{[*]} \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^5 - 64 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^6 - \\ 6\vartheta^{[*]} \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^3 \left(\frac{\wp'''(\xi)}{3!\wp'(\xi)} \right) \\ +150 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^4 \left(\frac{\wp'''(\xi)}{3!\wp'(\xi)} \right) - \\ 64 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^3 \left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)} \right) - \\ 62 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^2 \left(\frac{\wp'''(\xi)}{3!\wp'(\xi)} \right)^2 \\ +16 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^2 \left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)} \right) + \\ 24 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right) \left(\frac{\wp'''(\xi)}{3!\wp'(\xi)} \right) \left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)} \right) \end{array} \right) \left[\epsilon^{[j]}\right]^7 + O\left(\left[\epsilon^{[j]}\right]^8\right). \tag{35}$$

At each step of the iterative method, the error propagates according to the preceding approximations. By continuing the Taylor expansion and analyzing subsequent terms, we obtain:

$$\epsilon_*^{[j]} = \left(-\vartheta^{[*]} \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^4 + 2 \left(\frac{\wp''(\xi)}{2!\wp'(\xi)} \right)^5 \right) \left[\epsilon^{[j]}\right]^6 + O\left(\left[\epsilon^{[j]}\right]^7\right), \tag{36}$$

which confirms that the method exhibits at least sixth-order convergence. □

3.2. Fractional extension of the iterative scheme

While the iterative method in Section 3.1 achieves sixth-order convergence, it remains within the classical integer-order framework. However, many real-world problems exhibit memory effects and nonlocal behavior, requiring a more flexible approach. To extend its applicability to fractional settings, we introduce a generalized version that leverages Caputo-type fractional derivatives.

Using the Caputo fractional derivative, Candelario et al. [52] proposed the following extension of the classical Newton method for solving nonlinear equations:

$$x^{[j+1]} = x^{[j]} - \left(\Gamma(\sigma+1) \frac{\wp(x^{[j]})}{\left[{}_C\eth_{\sigma_1}^{\sigma}\right] \wp(x^{[j]})} \right)^{1/\sigma}, \tag{37}$$

where $\left[{}_C\eth_{\sigma_1}^{\sigma}\right] \wp(x^{[j]}) \approx \left[{}_C\eth_{\xi}^{\sigma}\right] \wp(\xi)$ for any $\sigma \in \mathbb{R}$. The fractional Newton method exhibits a convergence order of $\sigma + 1$ and satisfies the following error equation:

$$\epsilon^{[*]} = \frac{\Gamma(\sigma+1) - \Gamma(\sigma+1)}{\Gamma(\sigma+1)} \hbar_2 \left(\epsilon^{[j]}\right)^{\sigma+1} + O\left(\left(\epsilon^{[j]}\right)^{2\sigma+1}\right), \tag{38}$$

where $\epsilon^{[*]} = x^{[j]} - \xi$ and $\epsilon^{[j]} = x^{[j]} - \xi$, and $\hbar_\gamma = \frac{\Gamma(\sigma+1)}{\Gamma(\gamma\sigma+1)} \frac{\left[{}_C\eth_{\xi}^{\gamma\sigma}\right] \wp(\xi)}{\left[{}_C\eth_{\xi}^{\sigma}\right] \wp(\xi)}, \gamma \geq 2$.

Shams et al. [53] proposed the following single-step fractional iterative scheme:

$$x^{[j+1]} = x^{[j]} - \left(\Gamma(\sigma+1) \frac{\wp(x^{[j]})}{\left[{}_C\eth_{\sigma_1}^{\sigma}\right] \wp(x^{[j]})} \left[\frac{1}{1 - \sigma \frac{\wp(x^{[j]})}{1+\wp(x^{[j]})}} \right] \right)^{1/\sigma}. \tag{39}$$

This method also achieves a convergence order of $\sigma + 1$ and satisfies the following error equation:

$$\epsilon^{[*]} = \frac{\left(\sigma + \hbar_2\right) \Gamma^2(\sigma+1) - \hbar_2 \Gamma(2\sigma+1)}{\sigma\Gamma(\sigma+1)} \hbar_2 \left(\epsilon^{[j]}\right)^{\sigma+1} + O\left(\left(\epsilon^{[j]}\right)^{2\sigma+1}\right), \tag{40}$$

where $\epsilon^{[*]} = x^{[j]} - \xi$ and $\epsilon^{[j]} = x^{[j]} - \xi$ with $\hbar_\gamma = \frac{\Gamma(\sigma+1)}{\Gamma(\gamma\sigma+1)} \frac{\left[{}_C\eth_{\xi}^{\gamma\sigma}\right] \wp(\xi)}{\left[{}_C\eth_{\xi}^{\sigma}\right] \wp(\xi)}, \gamma \geq 2$.

A fractional version of the scheme ($S^{[\sigma]}$) is formulated as follows:

$$\upsilon^{[j]} = z^{[j]} - \left(\Gamma(\sigma+1) \frac{\wp(z^{[j]})}{\wp'(y^{[j]}) - \vartheta^{[*]} \wp(y^{[j]})} \right)^{\frac{1}{\sigma}}, \tag{41}$$

where

$$y^{[j]} = x^{[j]} - \left(\Gamma(\sigma+1) \frac{\wp(x^{[j]})}{\wp'(x^{[j]})} \right)^{\frac{1}{\sigma}},$$

$$z^{[j]} = y^{[j]} - \left(\Gamma(\sigma+1) \frac{\wp(y^{[j]})}{\wp'(y^{[j]})} \right)^{\frac{1}{\sigma}},$$

and $\vartheta^{[*]} \in \mathbb{R}$. To formally establish the convergence properties of the proposed fractional iterative method, we derive the following result, which guarantees a convergence order of at least $5\sigma + 1$.

Theorem 3. *Let*

$$\wp : \supset \subseteq \mathbb{R} \to \mathbb{R} \tag{42}$$

be a continuous function with $\left[{}_{C}\supset_{\sigma_1}^{\gamma\sigma}\right] \wp(x^{[j]})$ *of order* $\gamma\sigma$ *for some* $\gamma \geq 0$ *and* $\sigma \in (0, 1]$, *containing the exact root* ξ *of* $\wp(x)$. *Furthermore, for a sufficiently close initial value* $x^{[0]}$, *the convergence order of the Caputo-type fractional iterative scheme given by*

$$y^{[j]} = z^{[j]} - \left(\Gamma(\sigma+1)\frac{\wp(z^{[j]})}{\wp'(y^{[j]}) - \vartheta^{[*]}\wp(y^{[j]})}\right)^{\frac{1}{\sigma}}, \tag{43}$$

is at least $5\sigma + 1$, *and the error equation satisfies:*

$$\epsilon^{[*]} = \left(\frac{(2^{\sigma})^2\,\Gamma\left(\sigma+\frac{1}{2}\right)\hbar_2^2}{\sigma\Gamma(\sigma)\sqrt{\pi}} - \hbar_2^2\right)\left[\epsilon^{[j]}\right]^{5\sigma+1} + O\left(\left[\epsilon^{[j]}\right]^{6\sigma+1}\right), \tag{44}$$

where

$$\hbar\gamma = \frac{\Gamma(\sigma+1)}{\Gamma(\gamma\sigma+1)}\frac{\left[{}_{C}\supset_{\xi}^{\gamma\sigma}\right]\wp(\xi)}{\left[{}_{C}\supset_{\xi}^{\sigma}\right]\wp(\xi)}, \gamma \geq 2.$$

Proof. Let ξ be a root of $\wp$, and define $x_i = \xi + \epsilon^{[j]}$. By expanding $\wp(x_i)$ and $\left[{}_{C}\supset_{\sigma_1}^{\sigma}\right]\wp(x_i)$ using Taylor series around $x = \xi$ and setting $\wp(\xi) = 0$, we obtain:

$$\wp(x^{[j]}) = \frac{\left[{}_{C}\supset_{\xi}^{1\sigma}\right]\wp(\xi)}{\Gamma(\sigma+1)}\left[\left[\epsilon^{[j]}\right]^{\sigma} + \hbar_2\left[\epsilon^{[j]}\right]2^{\sigma} + \hbar_3\left[\epsilon^{[j]}\right]^{3\sigma}\right] + \cdots. \tag{45}$$

Applying the Caputo fractional derivative to (45) gives:

$$\left[{}_{C}\supset_{\sigma_1}^{\sigma}\right]\wp(x^{[j]}) = \frac{\left[{}_{C}\supset_{\xi}^{1\sigma}\right]\wp(\xi)}{\Gamma(\sigma+1)}\left[\Gamma(\sigma+1) + \frac{\Gamma(2\sigma+1)}{\Gamma(\sigma+1)}\hbar_2\left[\epsilon^{[j]}\right]^{\sigma} + \frac{\Gamma(3\sigma+1)}{\Gamma(2\sigma+1)}\hbar_3\left[\epsilon^{[j]}\right]^{2\sigma}\right] + \cdots. \tag{46}$$

Taking the inverse of (46), we obtain:

$$\begin{aligned}\frac{1}{\left[{}_{C}\supset_{\sigma_1}^{\sigma}\right]\wp(x^{[j]})} &= \frac{1}{\Gamma(\sigma+1)} - \frac{\Gamma(2\sigma+1)}{\Gamma(\sigma+1)}\hbar_2\left[\epsilon^{[j]}\right]^{\sigma} + \frac{\left(-\frac{\Gamma(3\sigma+1)}{\Gamma(\sigma+1)\Gamma(2\sigma+1)} + \frac{(\Gamma(2\sigma+1))^2}{(\Gamma(\sigma+1))^4}\right)}{\Gamma(\sigma+1)}\hbar_3\left[\epsilon^{[j]}\right]^{2\sigma} \\ &+ \left(\frac{\frac{\Gamma(3\sigma+1)}{\Gamma(\sigma+1)}\hbar_2\hbar_3 - \frac{(\Gamma(2\sigma+1))^3\hbar_2^2 - \Gamma(3\sigma+1)(\Gamma(\sigma+1))^3\hbar_2\hbar_3}{(\Gamma(\sigma+1))^6}}{\Gamma(\sigma+1)}\right)\left[\epsilon^{[j]}\right]^{3\sigma} + O\left(\left[\epsilon^{[j]}\right]^{4\sigma}\right).\end{aligned} \tag{47}$$

Multiplying (45) and (46) results in:

$$\begin{aligned}\frac{\wp(x^{[j]})}{\left[{}_{C}\supset_{\sigma_1}^{\sigma}\right]\wp(x^{[j]})} &= \frac{1}{\sigma\Gamma(\sigma)}\left[\epsilon^{[j]}\right]^{\sigma} + \left(\frac{(2^{\sigma})^2\,\Gamma\left(\sigma+\frac{1}{2}\right)\hbar_2}{\sigma^2\left(\Gamma((\sigma))\right)^2\sqrt{\pi}} + \frac{\hbar_2}{\sigma\Gamma((\sigma))}\right)\left[\epsilon^{[j]}\right]^{2\sigma} + \\ &\left(\begin{array}{c}\frac{(2^{\sigma})^4\left(\Gamma\left(\sigma+\frac{1}{2}\right)\right)^2\hbar_2^2}{\sigma^3(\Gamma(\sigma))^3\pi} - \frac{(2^{\sigma})^2\Gamma\left(\sigma+\frac{1}{2}\right)\hbar_2^3}{\sigma^2(\Gamma(\sigma))^2\sqrt{\pi}} - \\ \frac{1}{2}\frac{\hbar_3\left(3^{\sigma}\sqrt{3}\Gamma\left(\sigma+\frac{1}{3}\right)\Gamma\left(\sigma+\frac{2}{3}\right)\right)}{\sigma^2(\Gamma(\sigma))^2\sqrt{\pi}(2^{\sigma})^2\Gamma\left(\sigma+\frac{1}{2}\right)} + \frac{\hbar_3}{\sigma\Gamma(\sigma)}\end{array}\right)\left[\epsilon^{[j]}\right]^{3\sigma} + O(\left[\epsilon^{[j]}\right]^{4\sigma}),\end{aligned} \tag{48}$$

where $\Gamma\left(\frac{1}{\sigma}+1\right) = \frac{1}{\sigma}\Gamma\left(\frac{1}{\sigma}\right)$.

Using (48) in the final step of the root-finding scheme (41), we derive the final error equation:

$$y^{[j]} = w + \Lambda_1^{[*]}\left[\epsilon^{[j]}\right]^{\sigma+1} + \left(-\hbar_2 + \frac{(2^{\sigma})^2\,\Gamma(\sigma+1/2)\,\hbar_2}{\Gamma(\sigma)\,\sigma\sqrt{\pi}}\right)\left[\epsilon^{[j]}\right]^{2\sigma+1} + \cdots. \tag{49}$$

Expanding $\wp(y^{[j]})$ around ξ using Taylor series yields:

$$\begin{aligned}\wp(y^{[j]}) = &-\frac{\hbar_2\left(\begin{array}{c}\Gamma(\sigma)\,\sigma\sqrt{\pi} - \\ (2^{\sigma})^2\,\Gamma(\sigma+1/2)\end{array}\right)\left[\epsilon^{[j]}\right]^{\sigma+1}}{\Gamma(\sigma)\,\sigma\sqrt{\pi}} \\ &-\Lambda_2^{[*]}\left[\epsilon^{[j]}\right]^{2\sigma+1} + \Lambda_3^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1} + \Lambda_5^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1} + \cdots.\end{aligned} \tag{50}$$

The Caputo fractional derivative of (50) can be taken to get

$$\left[{}_{C}\supset_{\sigma_1}^{\sigma}\right]\wp(y^{[j]}) = 1 + 2\left(-\hbar_2 + \frac{(2^{\sigma})^2\,\Gamma(\sigma+1/2)\,\hbar_2}{\Gamma(\sigma)\,\sigma\sqrt{\pi}}\right)\hbar_2\left[\epsilon^{[j]}\right]^{\sigma+1} + \Lambda_4^{[*]}\left[\epsilon^{[j]}\right]^{2\sigma+1}\ldots$$

$$+3\left(-\hbar_2+\frac{(2^\sigma)^2\,\Gamma(\sigma+1/2)\,\hbar_2}{\Gamma(\sigma)\,\sigma\sqrt{\pi}}\right)^2\hbar_3\left[\epsilon^{[j]}\right]^{3\sigma+1}$$
$$+\Lambda_6^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1}+\Lambda_7^{[*]}\left[\epsilon^{[j]}\right]^{5\sigma+1}+\cdots. \tag{51}$$

The inverse of (51) is given as:

$$\frac{1}{\left[{}_C\mathfrak{D}^{\sigma}_{\sigma_1}\right]\wp(y^{[j]})}=1-2\left(-\hbar_2+\frac{(2^\sigma)^2\,\Gamma(\sigma+1/2)\,\hbar_2}{\Gamma(\sigma)\,\sigma\sqrt{\pi}}\right)\hbar_2\left[\epsilon^{[j]}\right]^{\sigma+1}+$$
$$\Lambda_{10}^{[*]}\left[\epsilon^{[j]}\right]^{2\sigma+1}+\Lambda_8^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1}+\Lambda_9^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1}+\cdots. \tag{52}$$

The result of multiplying (51) and (52) is

$$\frac{\wp\left(y^{[j]}\right)}{\left[{}_C\mathfrak{D}^{\sigma}_{\sigma_1}\right]\wp\left(y^{[j]}\right)}=\Lambda_{11}^{[*]}\left[\epsilon^{[j]}\right]^{\sigma+1}+\Lambda_{12}^{[*]}\left[\epsilon^{[j]}\right]^{2\sigma+1}+\Lambda_{13}^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1}$$
$$+\Lambda_{14}^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1}+\Lambda_{15}^{[*]}\left[\epsilon^{[j]}\right]^{5\sigma+1}+\cdots. \tag{53}$$

By applying equation (53) in the second step of method (41), we obtain the corresponding error expansion:

$$z^{[j]}=\xi+\Lambda_{16}^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1}+\Lambda_{17}^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1}+\Lambda_{18}^{[*]}\left[\epsilon^{[j]}\right]^{5\sigma+1}+\cdots. \tag{54}$$

Expanding $\wp(z^{[j]})$ around ξ using Taylor series yields:

$$\wp\left(z^{[j]}\right)=\begin{pmatrix}\frac{\Lambda_{20}^{[*]}\hbar_2{}^2e_n{}^5}{(\Gamma(\sigma))^3\sigma^3\pi^{5/2}(2^\sigma)^2\Gamma(\sigma+1/2)}+\\ \frac{e_n{}^4}{(\Gamma(\sigma))^2\sigma^2\pi^{3/2}}+\\ \frac{1/2\Lambda_{21}^{[*]}e_n{}^6}{(\Gamma(\sigma))^4\sigma^4\pi^{7/2}(2^\sigma)^4(\Gamma(\sigma+1/2))^2}\end{pmatrix}+\cdots \tag{55}$$

$$\left[{}_C\mathfrak{D}^{\sigma}_{\sigma_1}\right]\wp\left(y^{[j]}\right)-\vartheta^{[*]}\wp\left(y^{[j]}\right)=1+\Lambda_{22}^{[*]}\left[\epsilon^{[j]}\right]^{\sigma+1}+\Lambda_{23}^{[*]}\left[\epsilon^{[j]}\right]^{2\sigma+1}+$$
$$\Lambda_{24}^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1}+\Lambda_{25}^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1}+\cdots. \tag{56}$$

Taking the inverse of (56), we have

$$\frac{1}{\left[{}_C\mathfrak{D}^{\sigma}_{\sigma_1}\right]\wp\left(y^{[j]}\right)-\vartheta^{[*]}\wp\left(y^{[j]}\right)}=\begin{bmatrix}1+\Lambda_{26}^{[*]}\left[\epsilon^{[j]}\right]^{\sigma+1}+\\ \Lambda_{27}^{[*]}\left[\epsilon^{[j]}\right]^{2\sigma+1}+\Lambda_{28}^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1}+\cdots\end{bmatrix}, \tag{57}$$

where

$$\Lambda_{21}^{[*]}=\Delta^{[1]}+\Delta^{[2]}+\Delta^{[3]}+\Delta^{[4]}, \tag{58}$$

and the explicit forms of $\Delta^{[1]}$ through $\Delta^{[4]}$ are provided in Appendix A. The result of multiplying (56) and (57) is

$$\frac{\wp\left(z^{[j]}\right)}{\left[{}_C\mathfrak{D}^{\sigma}_{\sigma_1}\right]\wp\left(y^{[j]}\right)-\vartheta^{[*]}\wp\left(y^{[j]}\right)}=\Lambda_{29}^{[*]}\left[\epsilon^{[j]}\right]^{3\sigma+1}+\Lambda_{30}^{[*]}\left[\epsilon^{[j]}\right]^{4\sigma+1}+\Lambda_{31}^{[*]}\left[\epsilon^{[j]}\right]^{5\sigma+1}, \tag{59}$$

where the constants and $\Lambda^{\{1\}}-\Lambda^{\{7\}}$ and $\Lambda_1^{[*]}-\Lambda_{31}^{[*]}$ used in (44)–(59) are shown in Appendix A respectively. By substituting Eq. (59) into the final step of method (41), we obtain the corresponding error expression:

$$z^{[j]}=\xi+\begin{pmatrix}\Lambda_{32}^{[*]}+\Lambda_{33}^{[*]}+\\ -\Lambda_{34}^{[*]}\end{pmatrix}O\left(\left[\epsilon^{[j]}\right]^{5\sigma+1}\right)+O\left(\left[\epsilon^{[j]}\right]^{6\sigma+1}\right). \tag{60}$$

This completes the proof of the theorem. □

Although both fractional and classical schemes are effective for solving complex nonlinear problems, they also present significant challenges. Fractional schemes inherently rely on historical data, which increases memory requirements and computational cost. Their performance is highly sensitive to parameter selection, as the choice of fractional order directly influences both accuracy and convergence. Moreover, stability depends on careful selection of step sizes, discretization methods, and tuning parameters. Efficiency and convergence behavior can also vary across problem types, occasionally limiting their effectiveness for certain nonlinear systems.

To address these limitations, we propose a Caputo-type inverse parallel scheme that improves the stability and efficiency of fractional methods while retaining their key advantages. In contrast to the iterative techniques discussed earlier, which refine single-root approximations, the proposed parallel approach enables simultaneous computation of all roots, significantly accelerating convergence for nonlinear equations.

3.3. *Efficient Caputo-type inverse parallel method for nonlinear equations*

Parallel root-finding algorithms provide the broader computational foundation for our proposed approach. These methods play a critical role in efficiently solving nonlinear equations, particularly for polynomials with multiple roots. Unlike conventional single-root techniques, parallel schemes update approximations for all roots simultaneously, thereby reducing computational time and improving robustness. Their ability to exploit modern multi-core processors and distributed computing architectures makes them highly scalable for large-scale problems.

These methods are widely applied in control theory, signal processing, and engineering optimization, where complex root structures and high-dimensional systems demand efficient and reliable convergence. Compared to single-root and fractional-order schemes, parallel approaches offer several distinct advantages:

- They eliminate inefficiencies associated with computing one root at a time, thereby avoiding deflation techniques and repeated initial guesses.
- They refine all root estimates simultaneously through collective updates, reducing sensitivity to inaccurate initial approximations.
- They exploit parallelism to accelerate convergence and enhance numerical stability across a wide range of nonlinear problems.

Among various parallel numerical techniques, the Weierstrass–Durand–Kerner method [54] is one of the most widely studied. It is defined by:

$$y_i^{[j]} = x_i^{[j]} - w\left(x_i^{[j]}\right), \tag{61}$$

where

$$w\left(x_i^{[j]}\right) = \frac{\wp\left(x_i^{[j]}\right)}{\prod\limits_{\substack{t=1 \\ t\neq i}}^{n}\left(x_i^{[j]} - x_t^{[j]}\right)}, \quad (i,t = 1,\ldots,n), \tag{62}$$

is known as Weierstrass' correction. The method exhibits local quadratic convergence.

In 1977, Ehrlich [55] proposed a third-order simultaneous root-finding method, defined as:

$$y_i^{[j]} = x_i^{[j]} - \frac{1}{\frac{1}{N_i\left(x_i^{[j]}\right)} - \sum_{\substack{t=1 \\ t\neq i}}^{n}\left(\frac{1}{\left(x_i^{[j]} - x_t^{[j]}\right)}\right)}. \tag{63}$$

where $N_i(x)$ denotes a Newton-type correction term used to approximate the root update in simultaneous iterative schemes.

Later, Petković et al. [56] improved the convergence order from three to six by introducing a correction term $\epsilon_t^{[j]} = u_t^{[j]}$ into Eq. (63), resulting in the modified scheme:

$$y_i^{[j]} = x_i^{[j]} - \frac{1}{\frac{1}{N_i(x_i^{[j]})} - \sum_{\substack{t=1 \\ t\neq i}}^{n}\left(\frac{1}{\left(x_i^{[j]} - u_t^{[j]}\right)}\right)}. \tag{64}$$

Here,

$$u_t^{[j]} = x_t^{[j]} - \frac{\wp\left(y_t^{[j]}\right) - \wp\left(x_t^{[j]}\right)}{2 * \wp\left(y_t^{[j]}\right) - \wp\left(x_t^{[j]}\right)} \frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)},$$

$$y_t^{[j]} = x_t^{[j]} - \frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)},$$

and

$$N_i(x_i^{[j]}) = \frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)}.$$

Shams et al. in [57] proposed the following fractional parallel scheme:

$$x_i^{[j+1]} = x_i^{[j]} - \frac{w^{[**]}\left(x_i^{[j]}\right)}{1 + \frac{w^{[**]}\left(x_i^{[j]}\right)}{x_i^{[j]}}}, \tag{65}$$

where

$$w^{[**]}\left(x_i^{[j]}\right) = \frac{1}{2}\left[3 - \frac{w^{[*]}\left(x_i^{[j]}\right)\wp(s_i^{[j]})}{w\left(s_i^{[j]}\right)\wp(x_i^{[j]})}\right] w^{[*]}\left(x_i^{[j]}\right),$$

$$s_i^{[j]} = x_i^{[j]} - \frac{w^{[*]}\left(x_i^{[j]}\right)}{1 + \frac{w^{[*]}\left(x_i^{[j]}\right)}{x_i^{[j]}}},$$

$$w^{[*]}\left(x_i^{[j]}\right) = \frac{\wp(x_i^{[j]})}{\prod\limits_{\substack{t=1 \\ t\neq i}}^{m}(x_i^{[j]} - K_t^{[j]})},$$

$$K_t^{[j]} = \gamma_t^{[j]} - \left[\left(\frac{1}{2}\Gamma(\sigma+1)\left(3 - \frac{\left[{}^{C}\mathfrak{D}_{\sigma_1}^{\sigma}\right]\wp(s_t^{[j]})}{\left[{}^{C}\mathfrak{D}_{\sigma_1}^{\sigma}\right]\wp(x_t^{[j]})}\right)\right)\frac{\wp(x_t^{[j]})}{\left[{}^{C}\mathfrak{D}_{\sigma_1}^{\sigma}\right]\wp(x_t^{[j]})}\right)^{\frac{1}{\sigma}}\right].$$

Additionally, the scheme can be expressed as:

$$x_i^{[j+1]} = x_i^{[j]} - \frac{\phi_i}{\frac{\phi_i}{N_i(x_i^{[j]})} - \sum_{\substack{t=1 \\ t\neq i}}^{m}\left(\frac{\phi t}{\left(x_i^{[j]}-s_t^{[j]}\right)}\right) + \sum_{\substack{t=1 \\ t\neq i}}^{m}\left(\frac{(\phi t)^2 N_i\left(x_t^{[j]}\right)}{\left(x_i^{[j]}-s_t^{[j]}\right)^2}\right) + \sum_{\substack{t=1 \\ t\neq i}}^{m}\left(\frac{(\phi t)^3\left(N_i\left(x_t^{[j]}\right)\right)^2}{\left(x_i^{[j]}-s_t^{[j]}\right)^3}\right)}, \tag{66}$$

where

$$s_t^{[j]} = x_t^{[j]} - \left(\Gamma(\sigma+1)\frac{\wp(x_t^{[j]})}{\left[C\eth_{\sigma_1}^{\sigma}\right]\wp(x_t^{[j]})}\right)^{\frac{1}{\sigma}}.$$

While the previously discussed fractional parallel schemes offer notable improvements in efficiency and convergence, further refinements are necessary to enhance accuracy and computational robustness. Incorporating Weierstrass' correction — widely used in classical methods to improve root approximations — Cordero et al. [58] proposed a single-step method that underpins our enhanced fractional parallel scheme. The method is defined as:

$$v^{[j]} = z^{[j]} - \left(2 - \frac{\wp'(y^{[j]})}{\wp'(x^{[j]})}\right)\frac{\wp(z^{[j]})}{\wp'(x^{[j]})}, \tag{67}$$

where

$$z^{[j]} = y^{[j]} - \left(2 - \frac{\wp'(y^{[j]})}{\wp'(x^{[j]})}\right)\frac{\wp(y^{[j]})}{\wp'(x^{[j]})}, y^{[j]} = x^{[j]} - \frac{\wp(x^{[j]})}{\wp'(x^{[j]})}.$$

By applying Weierstrass' correction to (67), we develop the following fractional parallel scheme ($SS^{[\sigma]}$):

$$v_i^{[j]} = z_i^{[j]} - \frac{z_i^{[j]}\left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{y_i^{[j]}-y_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)\right)\prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{z_i^{[j]}-z_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)w\left(z_i^{[j]}\right)}{z_i^{[j]} + \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{y_i^{[j]}-y_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)\right)\prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{z_i^{[j]}-z_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)w\left(z_i^{[j]}\right)}, \tag{68}$$

where

$$z_i^{[j]} = y_i^{[j]} - \frac{z_i^{[j]}\left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{y_i^{[j]}-y_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)\right)\prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{y_i^{[j]}-y_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)w\left(z_i^{[j]}\right)}{z_i^{[j]} + \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{y_i^{[j]}-y_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)\right)\prod_{\substack{t=1 \\ i\neq t}}^{n}\left(\frac{y_i^{[j]}-y_t^{[j]}}{x_i^{[j]}-x_t^{[j]}}\right)w\left(z_i^{[j]}\right)},$$

$$y_i^{[j]} = \frac{x_i^{[j]}}{1 + \frac{w\left(x_i^{[j]}\right)}{x_i^{[j]}}}.$$

The following theorem establishes the convergence order of the method, demonstrating that $SS^{[\sigma]}$ achieves a high-order convergence rate of $20\sigma + 8$.

Theorem 4. *Let $\xi_1, \ldots, \xi_\sigma$ be simple zeros of a nonlinear equation. Then, for sufficiently close and distinct initial estimates $x_1^{[0]}, \ldots, x_n^{[0]}$ of the roots, the method $SS^{[\sigma]}$ exhibits convergence of order $20\sigma + 8$.*

Proof. Let $e_i = x_i^{[j]} - \xi_i, \epsilon_y = y_i^{[j]} - \xi_i$, $\epsilon_z = z_i^{[j]} - \xi_i$ and $\epsilon_v = v_i^{[j]} - \xi_i$ be the errors associated with $x_i^{[j]}$, $y_i^{[j]}$, $z_i^{[j]}$ and $v_i^{[j]}$ respectively.

From the first step of $SS^{[\sigma]}$, we obtain:

$$y_i^{[j]} - \zeta_i = x_i^{[j]} - \zeta_i - \frac{\left[\frac{\wp(x_j^{[j]})}{\prod_{\substack{t=1 \\ t\neq i}}^{n}(x_i^{[j]}-v_t^{[j]})}\right]}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}}, \tag{69}$$

which simplifies to

$$\epsilon_y = \epsilon_i - \frac{\epsilon_i\prod_{\substack{t=1 \\ t\neq i}}^{n}\left[\frac{x_i^{[j]}-\zeta_t}{x_i^{[j]}-v_t^{[j]}}\right]}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}}. \tag{70}$$

Rewriting (70), we obtain:

$$\epsilon_y = \epsilon_i \left(1 - \frac{\prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{x_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right]}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}} \right) = \epsilon_i \left(\frac{1 - \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{x_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right] + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}} \right). \tag{71}$$

Expanding further, the error term ϵ_y can be expressed as:

$$\epsilon_y = \epsilon_i \left(\frac{1 - \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{x_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right] + \frac{v_i^{[j]} - \zeta_j}{\epsilon_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{v_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right]}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}} \right). \tag{72}$$

Since

$$\prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{v_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right] - 1 = \sum_{k\neq i}^{n} \left(\frac{-\left[\epsilon_i\right]^{5\ \sigma+1}}{x_i^{[j]} - v_j^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{v_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right],$$

and using (26), we have $v_t^{[j]} - \zeta_t = O\left(\left| \left[\epsilon_t\right]^{5\sigma+1} \right| \right)$. Thus,

$$\epsilon_y = \epsilon_i \left(\frac{\sum_{k\neq i}^{n} \left(\frac{-\left[\epsilon_i\right]^{5\sigma+1}}{\epsilon_i^{[j]} - u_j^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{v_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right] + \frac{\left[\epsilon_i\right]^{5\sigma+1}}{x_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{v_i^{[j]} - \zeta_t}{x_i^{[j]} - v_t^{[j]}} \right]}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}} \right). \tag{73}$$

Assuming $\left| \left[\epsilon_k\right]^{5\sigma+1} \right| = \left| \left[\epsilon_t\right]^{5\sigma+1} \right|$, we obtain:

$$\epsilon_y = \epsilon_i \left(\left[\epsilon_k\right]^{5\sigma+1} \right) \left(\frac{\sum_{k\neq i}^{n} \left(\frac{-1}{\epsilon_i^{[j]} - u_t^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{u_i^{[j]} - \zeta_t}{\epsilon_i^{[j]} - u_t^{[j]}} \right] + \frac{1}{\epsilon_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{u_i^{[j]} - \zeta_t}{\epsilon_i^{[j]} - u_t^{[j]}} \right]}{1 + \frac{w^{[*]}(x_i^{[j]})}{x_i^{[j]}}} \right). \tag{74}$$

Thus, we conclude that

$$\epsilon_y = O\left(\left[\epsilon_k\right]^{5\sigma+2} \right). \tag{75}$$

For the second step, we have:

$$z_i^{[j]} - \xi = y_i^{[j]} - \xi - \frac{z_i^{[j]} \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(z_i^{[j]} \right)}{z_i^{[j]} + \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(z_i^{[j]} \right)}. \tag{76}$$

As

$$\prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) = 1,$$

we obtain:

$$\epsilon_z = \epsilon_y - \frac{y_i^{[j]} \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(y_i^{[j]} \right)}{y_i^{[j]} + \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(y_i^{[j]} \right)}. \tag{77}$$

Simplifying, we write:

$$\epsilon_z = \epsilon_y - \frac{\epsilon_y \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right]}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}}. \tag{78}$$

Rearranging further:

$$\epsilon_z = \epsilon_y \left(1 - \frac{\prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right]}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right) = \epsilon_y \left(\frac{1 - \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right] + \frac{w(y_i^{[j]})}{y_i^{[j]}}}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right). \tag{79}$$

Expanding further, the error ϵ_z can be written as:

$$\epsilon_z = \epsilon_y \left(\frac{1 - \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right] + \frac{y_i^{[j]} - \zeta i}{\epsilon_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right]}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right). \tag{80}$$

Since

$$\prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right] - 1 = \sum_{k\neq i}^{n} \left(\frac{-\epsilon_y}{y_i^{[j]} - y_j^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right],$$

we deduce:

$$\epsilon_z = \epsilon_y \left(\frac{\sum_{k\neq i}^{n} \left(\frac{-\epsilon_y}{y_i^{[j]} - y_t^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right] + \frac{\epsilon_y}{y_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right]}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right). \tag{81}$$

Thus, we obtain:

$$\epsilon_z = \left(\epsilon_y\right)^2 \left(\frac{\sum_{k\neq i}^{n} \left(\frac{-1}{y_i^{[j]} - y_t^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_j^{[j]}} \right] + \frac{1}{\epsilon_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right]}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right). \tag{82}$$

Finally, we establish that

$$\epsilon_z = O\left(\left[\epsilon_i\right]^{10\sigma+4} \right). \tag{83}$$

Considering the third step of the method, the error can be expressed as:

$$v_i^{[j]} - \xi = z_i^{[j]} - \xi - \frac{z_i^{[j]} \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{z_i^{[j]} - z_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(z_i^{[j]}\right)}{z_i^{[j]} + \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{z_i^{[j]} - z_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(z_i^{[j]}\right)}. \tag{84}$$

Since

$$\prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{z_i^{[j]} - z_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) = 1,$$

we obtain:

$$\epsilon_v = \epsilon_z - \frac{z_i^{[j]} \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{z_i^{[j]} - z_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(z_i^{[j]}\right)}{z_i^{[j]} + \left(2 - \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{y_i^{[j]} - y_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) \right) \prod_{\substack{t=1 \\ i\neq t}}^{n} \left(\frac{z_i^{[j]} - z_t^{[j]}}{x_i^{[j]} - x_t^{[j]}} \right) w\left(z_i^{[j]}\right)}. \tag{85}$$

Rewriting the expression in terms of the error ϵ_v, we obtain:

$$\epsilon_v = \epsilon_z - \frac{\epsilon_z \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right]}{1 + \frac{w(z_i^{[j]})}{z_i^{[j]}}}. \tag{86}$$

Similarly, from the second step, the error ϵ_z can be written as:

$$\epsilon_z = \epsilon_y \left(1 - \frac{\prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right]}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right) = \epsilon_y \left(\frac{1 - \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{y_i^{[j]} - \zeta_t}{y_i^{[j]} - y_t^{[j]}} \right] + \frac{w(y_i^{[j]})}{y_i^{[j]}}}{1 + \frac{w(y_i^{[j]})}{y_i^{[j]}}} \right). \tag{87}$$

For the third step, the error ϵ_v is given by:

$$\epsilon_v = \epsilon_z \left(\frac{1 - \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right] + \frac{z_i^{[j]} - \zeta i}{\epsilon_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right]}{1 + \frac{w(z_i^{[j]})}{z_i^{[j]}}} \right). \tag{88}$$

Since

$$\prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right] - 1 = \sum_{k\neq i}^{n} \left(\frac{-\epsilon_z}{z_i^{[j]} - z_j^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right],$$

we obtain:

$$\epsilon_v = \epsilon_z \left(\frac{\sum_{k\neq i}^{n} \left(\frac{-\epsilon_z}{z_i^{[j]} - z_t^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right] + \frac{\epsilon_z}{z_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right]}{1 + \frac{w(z_i^{[j]})}{z_i^{[j]}}} \right). \tag{89}$$

Thus, the error ϵ_v satisfies:

$$\epsilon_v = \left(\epsilon_z\right)^2 \left(\frac{\sum_{k\neq i}^{n} \left(\frac{-1}{z_i^{[j]} - z_t^{[j]}} \right) \prod_{t\neq i}^{k-1} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_j^{[j]}} \right] + \frac{1}{z_i^{[j]}} \prod_{\substack{t=1 \\ t\neq i}}^{n} \left[\frac{z_i^{[j]} - \zeta_t}{z_i^{[j]} - z_t^{[j]}} \right]}{1 + \frac{w(z_i^{[j]})}{z_i^{[j]}}} \right). \tag{90}$$

Finally, the error satisfies the bound:

$$\epsilon_z = O\left(\left[\epsilon_i\right]^{20\sigma+8} \right). \tag{91}$$

Hence, we complete the proof of this theorem. □

To further improve the convergence rate of the parallel scheme, we employ tools from complex dynamical systems to determine optimal parameter values, which are subsequently integrated into the parallel framework for solving nonlinear equations. In addition, by incorporating artificial neural networks (ANNs), we develop a hybrid parallel scheme that not only accelerates convergence but also extends global convergence properties. The following section examines this approach in greater detail.

4. Convergence enhancement techniques for fractional parallel schemes

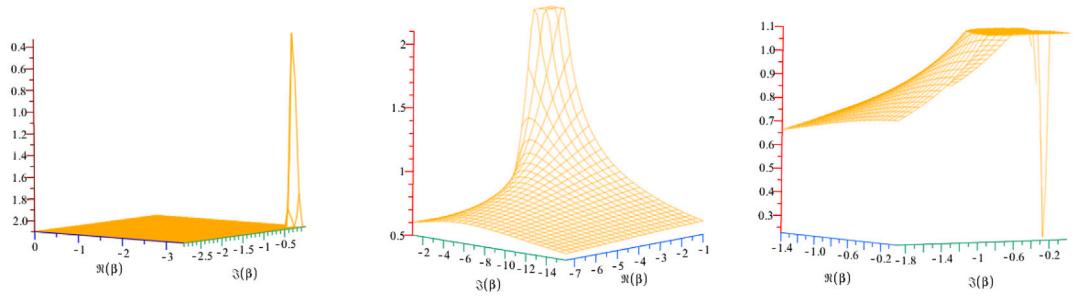
The Caputo-type family of parallel iterative algorithms is designed to compute all solutions of Eq. (2) simultaneously. This approach operates on a system of interdependent nonlinear equations, where each root of (2) is iteratively refined based on the current estimates of the others. While this method is computationally efficient, its convergence behavior is highly sensitive to the choice of initial approximations and algorithmic parameters, particularly when the roots are widely spaced or poorly separated.

To improve convergence rates, we employ complex dynamical systems analysis — specifically, stability analysis — to determine optimal parameter values for solving (2). In addition, we incorporate artificial neural networks (ANNs) to identify suitable regions for selecting initial approximations. By combining these techniques with the $SS^{[\sigma]}$ method, we achieve significantly enhanced stability, accelerated convergence, and reduced computational effort in computing all roots of (2).

4.1. Stability analysis of the $S^{[\sigma]}$ -scheme

Stability analysis plays a critical role in refining iterative schemes by guiding them toward regions of faster convergence and away from zones that may lead to divergence or slow convergence. In single root-finding methods for nonlinear equations, stability ensures that the iterative process remains reliable and robust. Specifically, stability refers to a method's ability to converge to the true root from a given initial approximation, even in the presence of small perturbations or computational errors.

Single root-finding methods typically exhibit local convergence, meaning they converge to the root when the initial estimate is sufficiently close to the true root. However, their stability depends on both the nature of the function and the choice of the initial approximation. In nonlinear settings, poor function behavior or an inadequate initial guess may cause divergence or convergence to extraneous fixed points unrelated to the desired root [59].



(a) Stability zone: $\vartheta^{[*]} = 4.03$. (b) Stability zone: $\vartheta^{[*]} = 3.3$. (c) Stability zone: $\vartheta^{[*]} = -1.4$.

Fig. 1. (a–c): Stability zones of the parallel schemes for different values of $\vartheta^{[*]}$.

The stability of such methods can be evaluated through convergence criteria, which measure the rate of convergence, and complex dynamical systems analysis, which assesses the method's sensitivity to variations in input values. Stability is typically achieved by carefully selecting the initial guess, tuning algorithmic parameters, and defining appropriate stopping criteria, while preserving the method's intrinsic convergence properties. This approach mitigates computational errors and enhances the overall reliability and consistency of the root-finding process [60].

To analyze stability, we derive the following rational map:

$$R(x) = x - \frac{2^{-\frac{1}{\sigma}} B^{\frac{1}{\sigma}} - 2^{-\frac{1}{\sigma}} A^{\frac{1}{\sigma}} - C_1^{[*]}}{\left(B^{\frac{2}{\sigma}} \vartheta^{[*]} 2^{\frac{1}{\sigma}} + 2B^{\frac{2}{\sigma}} \vartheta^{[*]} x 4^{\frac{1}{\sigma}} - C_2^{[*]}\right)^{\frac{1}{\sigma}}}, \tag{92}$$

where:

$$B = \frac{\Gamma(1+\sigma)(x^2-1)}{x}, \tag{93}$$

$$A = \Gamma(1+\sigma)\left(\frac{2^{\frac{1}{\sigma}} 4^{-\frac{1}{\sigma}} B^{\frac{2}{\sigma}} + 2^{\frac{1}{\sigma}} x^2 - 2B^{\frac{1}{\sigma}} x - 2^{\frac{1}{\sigma}}}{x 2^{\frac{1}{\sigma}} - B^{\frac{1}{\sigma}}}\right), \tag{94}$$

$$C_1^{[*]} = \Gamma(1+\sigma)\begin{bmatrix} 2^{\frac{1}{\sigma}} 4^{\frac{1}{\sigma}} x^2 + 2^{\frac{\sigma+1}{\sigma}} A^{\frac{1}{\sigma}} B^{\frac{1}{\sigma}} - 2\left(4^{\frac{1}{\sigma}}\right) A^{\frac{1}{\sigma}} x \\ -2\left(4^{\frac{1}{\sigma}}\right) B^{\frac{1}{\sigma}} x + \left(2^{\frac{1}{\sigma}}\right) A^{\frac{1}{\sigma}} - \left(4^{\frac{1}{\sigma}}\right) 2^{\frac{1}{\sigma}} \end{bmatrix}, \tag{95}$$

$$C_2^{[*]} = \begin{bmatrix} B^{\frac{2}{\sigma}} \vartheta^{[*]} 2^{\frac{1}{\sigma}} + 2B^{\frac{1}{\sigma}} \vartheta^{[*]} x 4^{\frac{1}{\sigma}} - \vartheta^{[*]} x 4^{\frac{1}{\sigma}} 2^{\frac{1}{\sigma}} \\ -2B^{\frac{1}{\sigma}} 4^{\frac{1}{\sigma}} + \vartheta^{[*]} 4^{\frac{1}{\sigma}} 2^{\frac{1}{\sigma}} + 2^{\frac{\sigma+1}{\sigma}} x 4^{\frac{1}{\sigma}} \end{bmatrix}. \tag{96}$$

For $\sigma \approx 1$, the rational function simplifies to:

$$R(x) = \frac{\begin{bmatrix} \vartheta^{[*]} x^{10} + 5\vartheta^{[*]} x^8 - 3x^9 - 6\vartheta^{[*]} x^6 - 36x^7 \\ -6\vartheta^{[*]} x^4 - 50x^5 + 5\vartheta^{[*]} x^2 - 36x^3 + \vartheta^{[*]} - 3x \end{bmatrix}}{x\left(x^2+1\right)^2\left(\vartheta^{[*]} x^4 - 2\vartheta^{[*]} x^2 - 4x^3 + \vartheta^{[*]} - 4x\right)}, \tag{97}$$

where $a, b \in \mathbb{C}$. Consequently, $R(x)$ depends on the parameters a and b. The conjugacy of $R(x)$ with a transformation operator is demonstrated via the Möbius transformation $M(x) = \frac{x-a}{x-b}$:

$$\mathrm{O}(x) = MoRoM^{-1}(x) = \frac{x^6\left(x^4 + \left(2\vartheta^{[*]} + 1\right)x^2 + 2\left(\vartheta^{[*]} - 1\right)\right)}{2x^4\left(\vartheta^{[*]} + 1\right) + \left(2\vartheta^{[*]} - 1\right)x^2 - 1}, \tag{98}$$

for $x \approx 1$, making it independent of a and b. Consequently, Eq. (97) aligns perfectly with:

$$R^{[*]}(x) = \frac{\sum_{i=0}^{n}\left(a_i x^i\right)}{\sum_{i=0}^{n}\left(a_{n-i} x^i\right)}, \quad \left\{a_i\right\}_{i=0}^{n} \in \mathbb{R}, \tag{99}$$

which exhibits remarkable properties [61].

The following proposition characterizes the fixed points of the rational map $\mathrm{O}(x)$, which play a central role in analyzing the behavior and convergence properties of the proposed schemes.

Proposition 5. *The fixed points of* $\mathrm{O}(x)$ *are classified as follows:*

- $x_0 = 0$ *and* $x_\infty = \infty$ *are* ***super-attracting*** *points.*
- $x_1 = 1$ *is a* ***repelling*** *point.*
- *The key points,* $x_0 = 0$ *and* $x_1 = 1$*, serve as a* ***super-attracting*** *and* ***repelling*** *point, respectively, for* $\sigma \approx 1$.

Proof. The fixed points of $\mathrm{O}(x)$ are determined by solving:

$$\mathrm{O}(x) = x \longleftrightarrow \frac{x^6\left(x^4 + \left(2\vartheta^{[*]} + 1\right)x^2 + 2\left(\vartheta^{[*]} - 1\right)\right)}{2x^4\left(\vartheta^{[*]} + 1\right) + \left(2\vartheta^{[*]} - 1\right)x^2 - 1} = x. \tag{100}$$

Rearranging, we obtain:

$$\frac{x^6\left(x^4+\left(2\vartheta^{[*]}+1\right)x^2+2\left(\vartheta^{[*]}-1\right)\right)}{2x^4\left(\vartheta^{[*]}+1\right)+\left(2\vartheta^{[*]}-1\right)x^2-1}-x=0. \tag{101}$$

Factoring out x, we get:

$$=0\longleftrightarrow x\left(\frac{x^5\left(x^4+\left(2\vartheta^{[*]}+1\right)x^2+2\left(\vartheta^{[*]}-1\right)\right)}{2x^4\left(\vartheta^{[*]}+1\right)+\left(2\vartheta^{[*]}-1\right)x^2-1}-1\right)=0. \tag{102}$$

Thus, one fixed point is $x_0=0$. To find additional fixed points, we solve:

$$\frac{x^5\left(x^4+\left(2\vartheta^{[*]}+1\right)x^2+2\left(\vartheta^{[*]}-1\right)\right)}{2x^4\left(\vartheta^{[*]}+1\right)+\left(2\vartheta^{[*]}-1\right)x^2-1}-1=0. \tag{103}$$

Furthermore, evaluating the limit as $x\to\infty$:

$$\lim_{x\to\infty}\frac{1}{\mathrm{O}\left(\frac{1}{x}\right)}=\frac{x^6\left(-x^4+\left(2\vartheta^{[*]}-1\right)x^2+2\left(\vartheta^{[*]}+1\right)\right)}{2x^4\left(\vartheta^{[*]}-1\right)+\left(2\vartheta^{[*]}+1\right)x^2-1}=0. \tag{104}$$

Thus, $x=\infty$ is also a fixed point.

To analyze the stability of these fixed points, we compute the derivative:

$$\mathrm{O}'(x)=4x^5\frac{\left[\begin{array}{c}3\vartheta^{[*]}x^8+4\vartheta^{[*]2}x^6+3x^8+103\vartheta^{[*]}x^6+\\ 8\vartheta^{[*]2}x^4+4\vartheta^{[*]2}x^2-6x^4-10\vartheta^{[*]}x^2-3\vartheta^{[*]}+3\end{array}\right]}{\left(2x^4\left(\vartheta^{[*]}+1\right)+\left(2\vartheta^{[*]}-1\right)x^2-1\right)^2}. \tag{105}$$

Setting $\mathrm{O}'(x)=0$ allows us to determine the critical points, given by:

$$Cr_1=\cdots=Cr_4=0,\quad Cr_{5,6}=\pm i, \tag{106}$$

$$Cr_{7,8}=\pm\frac{1}{3}\sqrt{3\sqrt{4\vartheta^{[*]2}+9}-6\vartheta^{[*]}}, \tag{107}$$

$$Cr_{9,10}=\pm\frac{1}{3}\sqrt{3\sqrt{4\vartheta^{[*]2}+9}-6\vartheta^{[*]}}, \tag{108}$$

$$Cr_{11,12}=\pm\frac{\sqrt{-\left(\vartheta^{[*]}+1\right)\left(\vartheta^{[*]}-1\right)}}{\left(\vartheta^{[*]}+1\right)}. \tag{109}$$

Evaluating the derivative of $\mathrm{O}(x)$ at these points yields the following classification:

- $x=0$ is a superattracting fixed point, since $\mathrm{O}'(0,\vartheta^{[*]})=0$.
- $x=-1$ is a repelling fixed point, as $\mathrm{O}'(-1)<-1$. □

4.2. Dynamical plane analysis for stability assessment

Dynamical planes offer a visual representation of the behavior and stability of iterative processes (Fig. 1(a–c)), making them valuable tools for analyzing the performance of nonlinear solvers. To improve the precision and efficiency of iterative methods, it is essential to examine convergence and divergence patterns within these planes, particularly by identifying fixed points, attractors, and chaotic regions.

In this study, we evaluate the stability of a single root-finding method by generating dynamical planes for different values of the fractional parameter within the interval $[0,1]$. The basins of attraction of $\mathrm{O}(x)$ — converging to zero — are shown in orange. Regions in blue indicate trajectories that tend toward infinity, while black regions denote divergence. In addition:

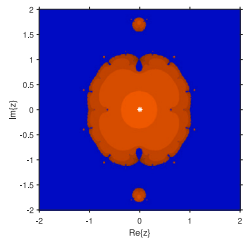
- Strange fixed points are marked with white circles.
- Free critical points are indicated by white squares.
- Fixed points are represented by white squares with a star.

The dynamical planes are constructed using initial values sampled from the domain $[-2,2]\times[-2,2]$. As shown in Fig. 2(a–e), the rational map predominantly converges to either zero or infinity, forming large basins of attraction. However, as the fractional parameter decreases from 1 to 0.5, the area of attraction progressively contracts, ultimately leading to divergence near zero, as illustrated in Figs. 3(a–c) and 4(a–c). This analysis indicates that the single-step method exhibits greater stability when the fractional parameter is close to 1, while its stability deteriorates as the parameter approaches zero. For visualization, we use the Jet colormap. As a stopping criterion, the iterative process is terminated when

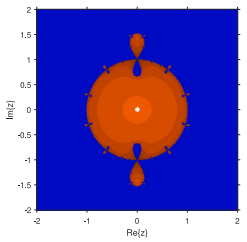
$$\left|f(x_i)\right|<0.0001$$

or when the number of iterations reaches a maximum of 20.

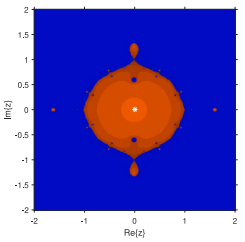
Table 1 reports the percentage of convergence (Per-C) and CPU time for different values of the parameter $\vartheta^{[*]}$. The results show that setting the fractional parameter to 1 yields higher convergence rates and shorter computation times. These findings are particularly relevant for optimizing hybrid parallel schemes and enhancing the efficiency of engineering applications.



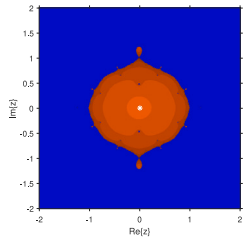
(a) Stable dynamic region: $\vartheta^{[*]} = 6.03$.



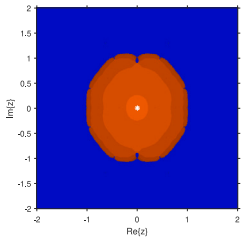
(b) Stable dynamic region: $\vartheta^{[*]} = 0.03$.



(c) Stable dynamic region: $\vartheta^{[*]} = 5.53$.

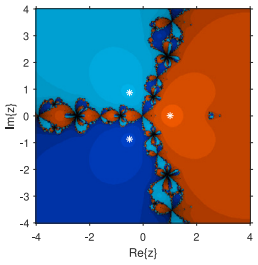


(d) Stable dynamic region: $\vartheta^{[*]} = 7.3$.

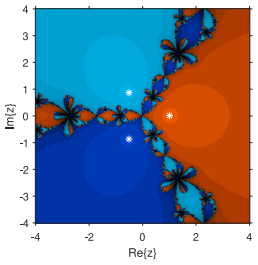


(e) Stable dynamic region: $\vartheta^{[*]} = -9.03$.

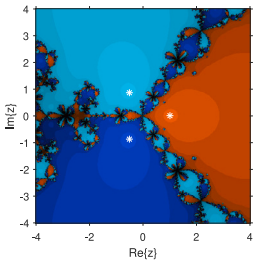
Fig. 2. (a–e): Stable dynamic behavior of iterative scheme for different values of $\vartheta^{[*]}$ and $\sigma = 1$.



(a) $\vartheta^{[*]} = 3.1$

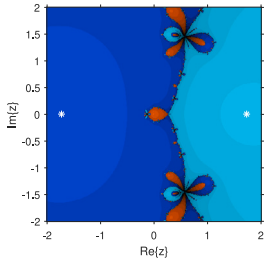


(b) $\vartheta^{[*]} = 1.1$

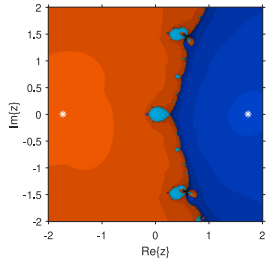


(c) The region of stability for $\vartheta^{[*]} = 7.3$

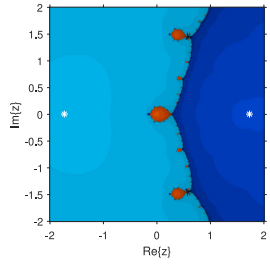
Fig. 3. (a–c): The iterative method's basins of attraction for solving $f(x) = x^3 - 1$ for various values of $\vartheta^{[*]}$.



(a) $\vartheta^{[*]} = 3.1$



(b) $\vartheta^{[*]} = 1.1$

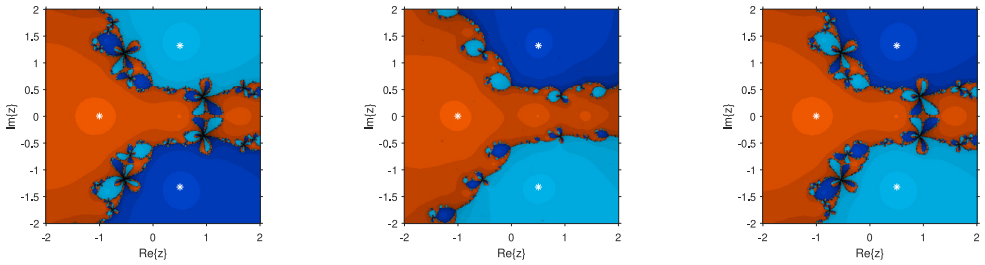


(c) $\vartheta^{[*]} = 7.3$

Fig. 4. (a–c): The iterative method's basins of attraction for solving $f(x) = x^2 + x - 3$ for various values of $\vartheta^{[*]}$.

5. Artificial neural network-based Caputo-type fractional parallel scheme

In the previous sections, we introduced fractional and parallel iterative schemes for solving nonlinear equations. While these methods improve convergence and robustness, their effectiveness is sensitive to the choice of initial values and algorithmic parameters. In parallel schemes in particular, poor initial estimates may result in slow convergence or even divergence. To address this issue, we incorporate artificial neural networks (ANNs), which have been widely applied in science and engineering for modeling complex systems, processing large datasets, and solving nonlinear problems. By leveraging their pattern recognition and adaptive learning capabilities, ANNs can significantly enhance the performance of fractional parallel root-finding methods.



(a) $\vartheta^{[*]} = 3.1$ (b) $\vartheta^{[*]} = 1.1$ (c) $\vartheta^{[*]} = 7.3$

Fig. 5. (a–c): The iterative method's basins of attraction for solving $f(x) = x^3 + x - 3$ for various values of $\vartheta^{[*]}$.

Table 1
Analysis of the schemes for different values of $\vartheta^{[*]}$.

Scheme	Fig. 1(a)	Fig. 1(b)	Fig. 1(c)
$\vartheta^{[*]}$	3.1	1.1	7.3
Per-C	98.4353%	99.76574%	99.87554%
CPU-time	2.34523	2.54646	3.007675
Scheme	Fig. 2(a)	Fig. 2(b)	Fig. 2(c)
$\vartheta^{[*]}$	3.1	1.1	7.3
Per-C	97.08934	99.39483	99.87364
CPU-time	1.12324	1.4252334	1.009243
Scheme	Fig. 3(a)	Fig. 3(b)	Fig. 3(c)
$\vartheta^{[*]}$	3.1	1.1	7.3
Per-C	99.00987	98.987632	98.087623
CPU-time	3.092832	4.49532	2.92842

The parallel inverse method for solving nonlinear equations can be further improved by integrating artificial neural networks (ANNs) into key components of the iterative process [62]. By training ANNs on representative polynomial structures, the following enhancements can be achieved:

- Prediction of high-quality initial estimates, reducing the number of iterations required for convergence.
- Adaptive control of the iteration process through dynamic adjustment of step sizes, improving both speed and stability.
- Robust handling of challenging cases, particularly those involving multiple or closely spaced roots where the Caputo-type fractional inverse parallel scheme may struggle.
- Reduction of numerical errors, leading to improved accuracy and computational stability.

By incorporating ANNs, we significantly increase the robustness and efficiency of the parallel root-finding scheme, making it applicable to a wider range of nonlinear problems.

To further accelerate convergence, we develop a hybrid neural network-based parallel approach. This method employs a three-layer artificial neural network (ANN) — consisting of an input layer, one or more hidden layers, and an output layer — trained using Particle Swarm Optimization (PSO) to identify all solutions of Eq. (2). The ANN's capacity for pattern recognition, dynamic adaptation, and error minimization significantly enhances the efficiency of the parallel root-finding technique. The key components of this approach are outlined below.

5.1. Dataset preparation for ANN training

To ensure faster convergence and high accuracy, an appropriately constructed dataset is essential for training the ANN. The dataset is generated as follows:

- The coefficients of the nonlinear equation (2) are randomly sampled from the interval $[0, 1]$.
- The dataset is divided into 70% for training and 30% for validation and testing.

These steps enable the ANN to generalize effectively across a wide range of nonlinear equations.

To compute high-quality initial approximations, we employ Cauchy's upper bound, which defines a disk of radius $r_b^{[i]}$ centered at the origin in the complex plane, containing all solutions of Eq. (2). The initial estimates are computed as:

$$x_i^{[0]} = \cos\left(2\pi \times r_b^{[i]}\right) + \sin\left(2\pi \times r_b^{[i]}\right) i \times r_b^{[i]} \tag{110}$$

where $r_b^{[i]} \in (0, 1)$ is given by:

$$r_b^{[i]} = 1 + \max\left\{\left|\frac{b_0}{b_n}\right|, \ldots, \left|\frac{b_{n-1}}{b_n}\right|\right\} \tag{111}$$

and $b_0 = b_1 = \cdots = b_n$ are the coefficients of the polynomial:

$$P_n(x) = b_1 x^n + \cdots + b_0. \tag{112}$$

5.2. Neural network model architecture

The ANN-based inverse parallel scheme is composed of the following components:

- An input layer that receives the coefficients of the nonlinear equation.
- One or two hidden layers equipped with nonlinear activation functions, which capture complex relationships between inputs and outputs.
- An output layer that generates approximations to the solutions of the nonlinear problem.

To minimize the error between predicted and true solutions, we adopt the mean squared error (MSE) as the loss function. The network's weights and biases are updated iteratively using the backpropagation algorithm to improve prediction accuracy.

5.3. Training and optimization of the ANN

The artificial neural network is trained using Particle Swarm Optimization (PSO), a robust metaheuristic algorithm known for its efficiency in identifying optimal solutions [63].

The training process involves the following steps:

- Iterative adjustment of the network's weights and biases to minimize prediction error.
- Adaptive learning from previous iterations to improve prediction accuracy.
- Optimization of step sizes to accelerate convergence.

The update rule for the ANN's weight vector is given by:

$$\Lambda^{[j+1]} = \Lambda^{[j]} - \frac{\left(\frac{\partial E_i^{[j]}}{\partial \Lambda i}\right)^T \times E_i^{[j]}}{\left(\left(\frac{\partial E_i^{[j]}}{\partial \Lambda i}\right)^T \left(\frac{\partial E_i^{[j]}}{\partial \Lambda i}\right) + \vartheta^{[*]} \times I\right)}, \tag{113}$$

where $E_i^{[j]}$ denotes the error vector at iteration j, and I is the identity matrix.

5.4. Performance evaluation using MSE

To evaluate the accuracy of the ANN predictions, we employ the mean squared error (MSE), defined as:

$$\mathrm{MSE} = \frac{1}{n}\sum_{j=1}^{n}\left(\xi_j - \mathrm{AN}^{[j]}\right)^2, \tag{114}$$

where ξ_j denotes the actual root values and $\mathrm{AN}^{[j]}$ denotes the ANN-predicted approximations.

- A lower MSE indicates that the ANN predictions are closer to the actual solutions.
- The loss function penalizes large deviations, promoting stable and accurate approximations of the roots.
- Minimizing MSE contributes to faster convergence and improved overall computational efficiency.

5.5. Advantages of the ANN-integrated parallel scheme

Integrating an artificial neural network (ANN) into the Caputo-type fractional inverse parallel scheme offers several advantages:

- Optimized Initial Values: The ANN predicts high-quality initial approximations, reducing the number of iterations required for convergence.
- Adaptive Learning: Step sizes and updates are dynamically adjusted to enhance numerical stability.
- Robustness in Complex Cases: The method remains effective even for closely spaced roots or ill-conditioned equations.
- Error Mitigation: ANN-guided optimization reduces numerical errors and accelerates convergence.

By combining artificial neural networks with Particle Swarm Optimization (PSO), the proposed hybrid approach enhances the efficiency, stability, and robustness of parallel fractional root-finding techniques. This integrated framework significantly improves convergence rates and provides a reliable tool for solving nonlinear equations in engineering and applied sciences.

6. Numerical experiments and performance evaluation

In this section, we present a series of numerical experiments to demonstrate the computational performance of the proposed methods in various engineering applications. The computer algorithms are implemented in `Maple 18`, and the termination criteria are defined as follows:

$$(i) \qquad e_i^{[j]} = \left|x_i^{[j+1]} - x_i^{[j]}\right| \le 10^{-6}, (ii) \qquad e_i^{[j]} = \left|f\left(x_i^{[j]}\right)\right| \le 10^{-6}, \tag{115}$$

where $e_i^{[j]}$ denotes the error at the j-th iteration. All simulations and numerical experiments were conducted using `MATLAB R2023a` on the following hardware configuration:

- Processor: Intel® Core™ i7-12700H, 2.30 GHz
- RAM: 32 GB DDR5

- GPU: NVIDIA RTX 3060 (6 GB)—used for neural network-based schemes
- Operating System: Windows 11 Pro (64-bit)

The Deep Learning Toolbox in MATLAB was employed to conduct neural network investigations. All computations were performed using double-precision arithmetic, which is reflected in the reported CPU times and performance indicators. In particular, this configuration ensures accuracy and reproducibility when assessing CPU time, memory usage, and neural network training efficiency. To enable a comprehensive evaluation of the proposed high-order fractional parallel iterative algorithms, we adopt the following performance metrics: accuracy, efficiency, stability, and computational cost.

- Number of Iterations: Indicates the number of iterations required to satisfy the stopping criteria defined in (115).
- Residual Error: The absolute value of the function evaluated at the most recent iterate, or the error between consecutive iterates, indicating the proximity to the true root.
- Computational Order of Convergence: Numerically approximated by

$$\rho_i^{[j-1]} = \frac{\ln\left|\frac{x_i^{[j+1]}-x_i^{[j]}}{x_i^{[j]}-x_i^{[j-1]}}\right|}{\ln\left|\frac{x_i^{[j]}-x_i^{[j-1]}}{x_i^{[j-1]}-x_i^{[j-2]}}\right|} \tag{116}$$

  which confirms the scheme's theoretical order of convergence.
- Computational Time: Measures the computational speed of the algorithm using MATLAB's `tic-toc` function to evaluate the time required for convergence.
- Function Evaluations: The variable `FEval` in MATLAB denotes the total number of function and derivative evaluations performed during the iterations.
- Arithmetic Operation Count: Estimates the number of basic arithmetic operations performed by the algorithm, providing insight into its computational complexity.
- Initial Guess Strategy: To evaluate the robustness and efficiency of the proposed methods, two types of initial guesses were considered.
  - Initial values close to the exact roots were used to evaluate the local convergence behavior and to validate the theoretical order of convergence of the fractional-order parallel schemes.
  - Random initial guesses were generated using MATLAB's `rand()` function to evaluate the global convergence behavior of the proposed methods.

  This dual strategy is particularly important when dealing with closely clustered roots, where convergence analysis becomes more challenging.
- Stability Under Noisy Inputs: To evaluate the robustness of the proposed technique, noisy inputs — i.e., initial guesses located far from the exact solution — are introduced. Despite this perturbation, the method still converges, demonstrating its global convergence behavior.
- Percentage Convergence (Percentage-C): The percentage convergence of the parallel schemes is computed as

$$\text{Percentage-C} = 100 \times \left(1 - \frac{\left|x_i^{[j+1]} - \xi\right|}{\left|x_i^{[j]} - \xi\right|}\right) \tag{117}$$

  which measures how rapidly the parallel sequence approaches the exact solution. We compare the proposed method with the parallel scheme $PM^{[1]}$ introduced in [64], defined as:

$$x_i^{[j+1]} = x_i^{[j]} - \frac{1}{\frac{1}{N_i(x_i^{[j]})} - \sum_{\substack{t=1 \\ t\neq i}}^{n}\left(\frac{1}{\left(x_i^{[j]} - Z_t^{[j]}\right)}\right)}, \tag{118}$$

where

$$Z_t^{[j]} = u_t^{[j]} - \frac{\left(y_t^{[j]} - \wp\left(u_t^{[j]}\right)\right)\left(\frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)}\right)}{\left(\wp\left(x_t^{[j]}\right) - \wp\left(u_t^{[j]}\right)\right)^2}\left[\wp\left(y_t^{[j]}\right) - \frac{\left(\wp\left(x_t^{[j]}\right)\right)^2}{\wp(y_t^{[j]}) - \wp\left(u_t^{[j]}\right)}\right], \tag{119}$$

$$u_t^{[j]} = y_t^{[j]} - \frac{\wp\left(x_t^{[j]}\right)\wp\left(y_t^{[j]}\right)\left(\frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)}\right)}{\left(\wp\left(x_t^{[j]}\right) - \wp\left(y_t^{[j]}\right)\right)^2}, y_t^{[j]} = x_t^{[j]} - \left(\frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)}\right). \tag{120}$$

Furthermore, we compare our scheme with the method proposed by Shams et al. ($SS^{[1]}$) in [65]:

$$x_i^{[j+1]} = u_i^{[j]} - \frac{\wp(x_j^{[j]})}{\prod_{\substack{t=1 \\ t\neq i}}^{n}(u_i^{[j]} - u_t^{[j]})}, \tag{121}$$

where

$$u_i^{[j]} = y_i^{[j]} - \frac{\wp(x_j^{[j]})}{\prod_{\substack{t=1 \\ t\neq i}}^{n}(y_i^{[j]} - y_t^{[j]})}, y_i^{[j]} = x_i^{[j]} - \frac{\wp(x_j^{[j]})}{\prod_{\substack{t=1 \\ t\neq i}}^{n}(x_i^{[j]} - s_t^{[j]})}, \tag{122}$$

$$s_t^{[j]} = x_t^{[j]} - \frac{\alpha\left(\wp\left(x_t^{[j]}\right)\right)^2}{\wp\left(x_t^{[j]} + \alpha\wp\left(x_t^{[j]}\right)\right) - \wp\left(x_t^{[j]}\right)}, \quad \alpha \in \mathbb{R}. \tag{123}$$

Additionally, we compare the results with the scheme SS[2] proposed in [66]:

$$x_i^{[j+1]} = y_i^{[j]} - \frac{\phi_i}{\frac{\phi i}{N_i(y_i^{[j]})} - \sum_{\substack{t=1 \\ t\neq i}}^{n}\left(\frac{\phi_t}{\left(y_i^{[j]} - y_t^{[j]}\right)}\right)}, \tag{124}$$

where

$$y_i^{[j]} = x_i^{[j]} - \frac{1}{\frac{1}{N_i(x_i^{[j]})} - \sum_{\substack{t=1 \\ t\neq i}}^{n}\left(\frac{1}{\left(x_i^{[j]} - u_t^{[j]}\right)}\right)}, \tag{125}$$

$$u_t^{[j]} = v_t^{[j]} - \phi_t\left(1 + \frac{\wp\left(v_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)}\right)^{\frac{2}{\phi_t}}\left(\frac{\wp\left(v_t^{[j]}\right)}{\wp'\left(v_t^{[j]}\right)}\right), \tag{126}$$

$$v_t^{[j]} = x_t^{[j]} - \left(\phi_t\frac{\wp\left(x_t^{[j]}\right)}{\wp'\left(x_t^{[j]}\right)}\right). \tag{127}$$

The following notations are used to present and interpret the numerical results:

Notations	
σ	Fractional parameters
PM[1], SS[1]–SS[2]	Existing Methods
SC[σ]	Newly developed Scheme
Per-convergence	Total convergence points
C-time	Computational time (in seconds)
ζ	Exact solution
A-SC[σ]	Newly developed ANN-based parallel scheme
Maximum-E	Maximum error using ANN
Local-C	Local computational order of convergence
Computational-T	Computational time of A-SC[σ]
$\epsilon^{[*]}$	Residual error
${}_C\mathfrak{D}^{\sigma}_{\sigma_1}$	Caputo fractional derivative

6.1. Engineering application: Bratu-type fractional boundary problem

The Bratu-type fractional boundary problem is a boundary value problem involving fractional differential equations. It generalizes the classical Bratu problem to a second-order fractional ordinary differential equation and is particularly relevant in engineering applications such as heat transfer, chemical reaction kinetics, and combustion processes [67,68]. The problem is formulated as:

$$\begin{cases} \left[{}_C\mathfrak{D}^{\sigma}_{\sigma_1}\right]U(x) - 6\left(U''(x)\right)^2 + (1+x)U(x) = x; 0 \le x \le 1, \\ U(0) = 0, \\ U'(1) = 0.25, \\ U''(1) = -0.25, \end{cases} \tag{128}$$

where $2 < \sigma \le 3$. Here, the order σ is not restricted to integer values but may assume any real number within the given range. The inclusion of fractional derivatives enables the modeling of complex physical phenomena such as memory effects and nonlocal interactions, which are essential in many engineering contexts.

To solve this problem, specialized techniques from fractional calculus — such as fractional integration and differentiation — combined with numerical methods tailored to fractional differential equations are required. Following the approach described in [69], we approximate the solution by expanding the polynomial up to two terms as:

$$U_0(x) = c_1 x + \frac{1}{2}c_2 x^2, \tag{129}$$

$$U_1(x) = -hx^{\tau+\sigma}\left[\frac{6c_2^2\Gamma(1+\sigma)}{\Gamma(1+\tau+\sigma)} + \frac{\left(1-c_1\right)\Gamma(2+\sigma)}{\Gamma(2+\tau+\sigma)}x - \frac{3c_1\Gamma(3+\sigma)}{2\Gamma(3+\tau+\ \sigma)}x^2 - \frac{c_1\Gamma(4+\sigma)}{2\Gamma(4+\tau+\sigma)}x^3\right], \tag{130}$$

$$U_2(x) = h^2\left[\frac{12c_2\Gamma(1+\sigma)\Gamma(2+\sigma)\Gamma(2\tau+\ \sigma)}{\Gamma(\sigma)\Gamma(2(\sigma+\tau))\Gamma(2+\tau+\sigma)}x^{2(\tau+\sigma)-1} + \cdots\right]. \tag{131}$$

Table 2
Results of $SC^{[\sigma]}$ for solving Eq. (132) using close initial guesses.

σ	$e_1^{[3]}$	$e_2^{[3]}$	$e_3^{[3]}$	$e_4^{[3]}$	$e_5^{[3]}$	$e_6^{[3]}$	$\rho_i^{[j-1]}$
0.1	3.43	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	5.57×10^{-5}	6.6434
0.3	0.03	4.45×10^{-32}	5.17×10^{-35}	1.08×10^{-17}	5.68×10^{-29}	5.57×10^{-25}	8.7654
0.5	0.75	0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	5.57×10^{-35}	12.7664
0.7	0.75	0.0	6.00×10^{-89}	3.02×10^{-84}	5.57×10^{-95}	0.0	18.2324
0.9	0.77	0.0	0.0	0.0	0.0	0.0	20.9087

Table 3
Results of parallel schemes for solving Eq. (132) using close initial guesses.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$e_5^{[j]}$	$e_6^{[j]}$	$\rho_i^{[j-1]}$
$SC^{[\sigma]}$	1.19×10^{-232}	0.0	0.0	0.0	0.0	0.0	21.30
$PM^{[1]}$	1.19×10^{-32}	1.19×10^{-32}	4.19×10^{-45}	1.08×10^{-17}	5.68×10^{-29}	1.57×10^{-55}	8.64
$SS^{[1]}$	0.0	0.0	6.50×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	3.43×10^{-65}	9.87
$SS^{[2]}$	1.19×10^{-52}	0.0	0.0	0.0	5.57×10^{-55}	42	10.99

Table 4
Stability results of the numerical schemes for solving (132).

Scheme	Iterations	$[+,-,\times,\div]$	$[\wp,\wp']$	Percentage-C	Computational-T
$SC^{[\sigma]}$	3	52	15	99.5458772	1.76×10^{-2}
$PM^{[1]}$	5	125	25	85.7766565	1.08×10^{-2}
$SS^{[1]}$	5	85	18	86.7646676	3.02×10^{-1}
$SS^{[2]}$	5	87	16	79.8756563	3.02542

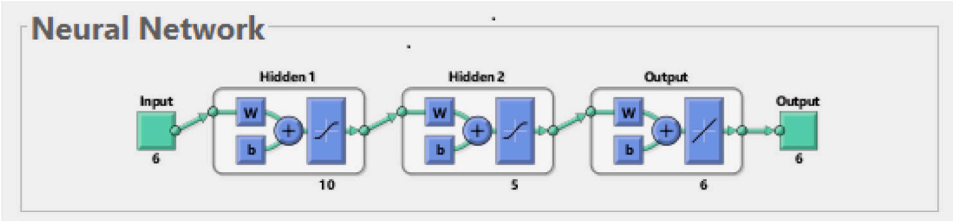


Fig. 6. Architecture of artificial neural networks for solving engineering Application 1 employing a parallel scheme.

For the specific parameter values $h=-1$, $\sigma=2$, $\tau=0$, $c_1=-0.584094$, and $c_2=-0.154811$, the approximate solution is given by:

$$U(x)=\begin{bmatrix} -0.00135856666x^6-0.0081514x^5+0.028081x^4+0.02396644572x^3 \\ -0.0774055x^2+0.326056x \end{bmatrix}, \tag{132}$$

The exact roots, accurate to four decimal places, are as follows:

$$\begin{bmatrix} \xi_1=0,\xi_2=3.115448818,\xi_3=-2.575426379, \\ \xi_4=-8.116897033,\xi_{5,6}=0.7884372957\pm1.750281412i, \end{bmatrix}. \tag{133}$$

We approximate all solutions of (141) using the following initial approximations:

$$\left[x_1^{[0]}=0.1,x_2^{[0]}=3.1,x_3^{[0]}=-2.5,x_4^{[0]}=-8.1,x_{5,6}^{[0]}=0.7\pm1.7i,\right]. \tag{134}$$

The numerical performance of various schemes for solving this equation is summarized in Tables 2–3.

Table 2 presents the results of the parallel scheme $SC^{[\sigma]}$ for solving equation (132) with varying values of σ. The residual error and the computational order of convergence confirm the theoretical convergence behavior of $SC^{[\sigma]}$. When the initial values are chosen close to the exact solution, the accuracy of the scheme improves, and the number of required iterations decreases as σ increases from 0.1 to 1.0.

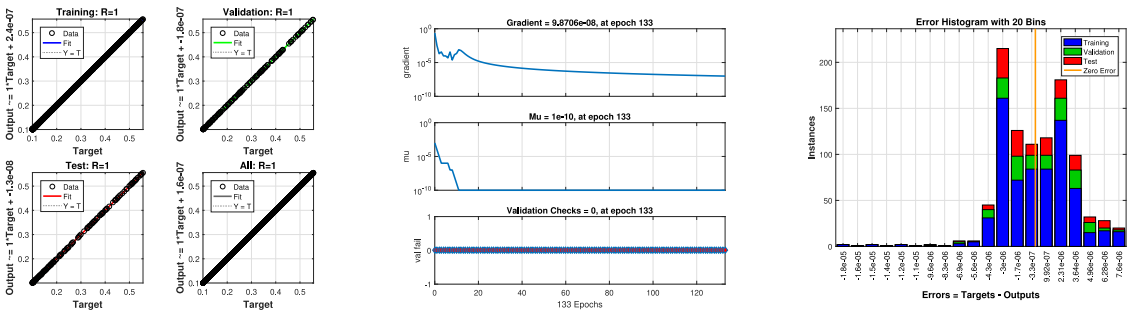
Tables 3–4 compare the numerical results of the proposed method with those of established approaches in the literature, namely $PM^{[1]}$, $SS^{[1]}$, and $SS^{[2]}$. As shown in Table 2, the proposed technique achieves significantly lower residual errors while requiring fewer iterations (j) than the competing methods. Furthermore, the observed computational order of convergence closely matches the theoretical expectation.

The stability analysis includes metrics such as the total number of arithmetic operations $(+,-,\times,\div)$ per iteration, function and fractional derivative evaluations $(\wp,\wp')$, percentage convergence (Percentage-C), and computational time in seconds (Computational-T). These indicators collectively demonstrate that the proposed scheme $SC^{[\sigma]}$ outperforms $PM^{[1]}$, $SS^{[1]}$, and $SS^{[2]}$ in both accuracy and efficiency.

To further enhance convergence, the parameter values in the fractional schemes were selected based on the stability profiles observed in the dynamical planes (Figs. 1–5). The results of the hybrid ANN-based schemes are presented in Tables 5–8. The neural network was trained using the dataset described in Appendix C, and its architecture for solving equation (132) is depicted in Fig. 6.

Fig. 7(a–c) provide a detailed visualization of the neural network implementation and performance. Fig. 7(a) displays the fitness curve, which reflects both the accuracy and stability of the training process. Fig. 7(b) presents the transition statistics curve, indicating an efficient convergence rate. The error histogram in Fig. 7(c) shows the distribution of prediction errors, further confirming the consistency of the ANN-based approach.

Together, these visualizations and the numerical results in Table 4 validate the reliability and robustness of the proposed neural network-enhanced framework for solving this class of engineering problems.



(a) Regression curve (b) Transition statistics (c) Error histograms

Fig. 7. Regression analysis of the artificial neural network for solving Engineering Application 1: (a) regression curve, (b) transition statistics, and (c) error histogram.

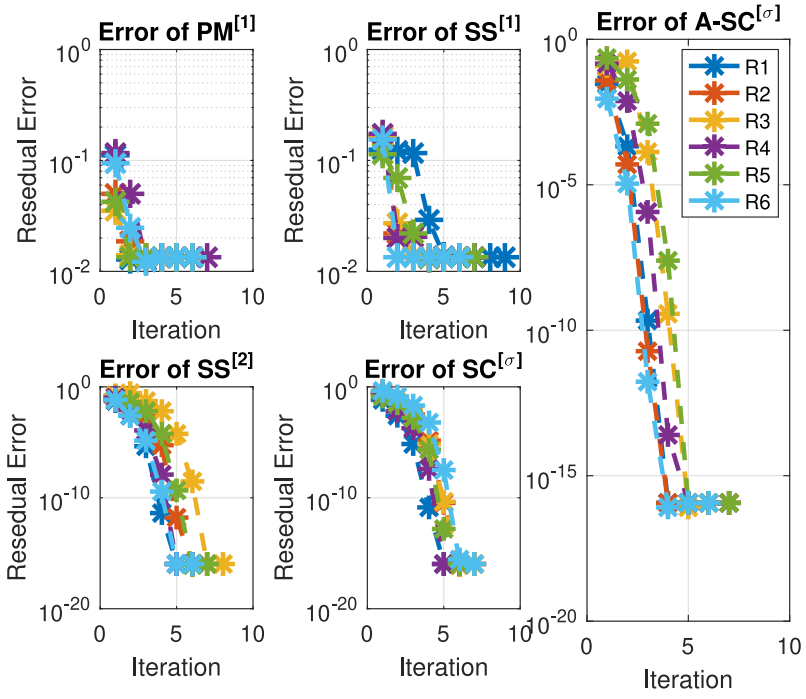


Fig. 8. Error plot of the parallel scheme for solving Engineering Application 1 using randomly generated test vectors.

Table 5
Results of parallel and hybrid parallel schemes using random initial guesses.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$e_5^{[j]}$	$e_6^{[j]}$	$\rho_i^{[j-1]}$
SC$^{[\sigma]}$	3.43	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	18	6.30
PM$^{[1]}$	0.03	1.19×10^{-32}	4.19×10^{-35}	1.08×10^{-17}	5.68×10^{-29}	17	2.64
SS$^{[1]}$	0.75	0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	35	2.87
SS$^{[2]}$	0.77	0.0	0.0	0.0	0.0	42	9.99
A-SC$^{[\sigma]}$	0.0	0.0	0.0	0.0	0.0	0.0	.0

To illustrate the effectiveness of fractional-order parallel schemes under challenging conditions — such as the presence of clustered or multiple roots — randomly generated initial guesses are employed, as these scenarios often hinder rapid convergence. The results for A-SC$^{[\sigma]}$ with random initial values (see Appendix B), along with comparisons to the parallel schemes SC$^{[\sigma]}$, PM$^{[1]}$, SS$^{[1]}$, and SS$^{[2]}$, are summarized in Table 5.

As shown in Table 5, A-SC$^{[\sigma]}$ significantly outperforms SC$^{[\sigma]}$, PM$^{[1]}$, SS$^{[1]}$, and SS$^{[2]}$ when applied to randomly initialized inputs. In particular, A-SC$^{[\sigma]}$ exhibits markedly faster convergence across all test cases.

Tables 6–7 present the efficiency and stability analysis of the parallel schemes based on the hybrid ANN-enhanced approach A-SC$^{[\sigma]}$, applied with randomly generated initial values. Informed by insights from dynamical analysis, the numerical results demonstrate that A-SC$^{[\sigma]}$ yields substantial improvements over existing methods in terms of local convergence, number of iterations, maximum error, residual error (as shown in Fig. 8), and CPU time.

The stability results presented in Table 8 indicate that the proposed scheme A-SC$^{[\sigma]}$ outperforms SC$^{[\sigma]}$, PM$^{[1]}$, SS$^{[1]}$, and SS$^{[2]}$ in terms of the total number of arithmetic operations per iteration, function and fractional derivative evaluations, percentage convergence, and computational time (in seconds).

Table 6
Comparison of the consistency between ANN-integrated and standard parallel schemes using random initial values. The table reports the selected parameter $\vartheta^{[*]}$, maximum error (Maximum-E), and the computational time (Computational-T).

Scheme	$\vartheta^{[*]}$	Maximum-E	Computational-T
$SC^{[\sigma]}$	0.1	5.15×10^{-25}	2.1034573454
$PM^{[1]}$	1.2	5.15×10^{-15}	1.1034564554
$SS^{[1]}$	$5.5i$	1.19×10^{-12}	4.1964364534
$SS^{[2]}$	0.4	1.19×10^{-12}	4.1975633453
A-$SC^{[\sigma]}$	$0.9i$	1.19×10^{-34}	0.0435234345

Table 7
Efficiency analysis of ANN-integrated and classical parallel schemes using random initial values. The table reports the parameter $\vartheta^{[*]}$, local convergence indicator (Local-C), and number of iterations.

Scheme	$\vartheta^{[*]}$	Local-C	Iterations
$SC^{[\sigma]}$	0.1	7.15367453	11
$PM^{[1]}$	1.2	5.152345456	18
$SS^{[1]}$	$5.5i$	9.19456865	18
$SS^{[2]}$	0.4	7.19456356	16
A-$SC^{[\sigma]}$	$0.9i$	15.07643457	5

Table 8
Stability analysis of A-$SC^{[\sigma]}$ and classical parallel schemes. The table reports arithmetic operations per step, function evaluations, and percentage convergence using random initial values.

Scheme	$\vartheta^{[*]}$	$[+, -, \times, \div]$	$[\wp, \wp']$	Percentage-C
$SC^{[\sigma]}$	0.1	52	15	91.874543454
$PM^{[1]}$	1.2	121	25	85.125006756
$SS^{[1]}$	$5.5i$	89	18	86.764453645
$SS^{[2]}$	0.4	88	16	69.877756536
A-$SC^{[\sigma]}$	$0.9i$	25	16	99.124427755

Table 9
Results of parallel and hybrid parallel schemes using random initial guesses.

Scheme	$e_1^{[7]}$	$e_2^{[7]}$	$e_3^{[7]}$	$e_4^{[7]}$	$e_5^{[7]}$	$e_6^{[7]}$	C-Time
$SC^{[\sigma]}$	1.3	5.15×10^{-15}	8.0×10^{-14}	7.6×10^{-13}	1.3×10^{-10}	0.0	6.30
$PM^{[1]}$	2.3	1.19×10^{-32}	4.9×10^{-35}	9.8×10^{-17}	5.8×10^{-29}	0.3×10^{-10}	2.64
$SS^{[1]}$	5.5	0.0	6.0×10^{-62}	8.2×10^{-44}	5.7×10^{-55}	0.0	2.87
$SS^{[2]}$	1.7	0.0	0.0	0.0	0.0	0.0	9.99
A-$SC^{[\sigma]}$	0.0	0.0	0.0	0.0	0.0	0.0	10.5

Selecting initial guess values in a high-noise environment—i.e., placing them far from the exact roots, such as

$$\left[x_1^{[0]} = 4.0, x_2^{[0]} = 7.0, x_3^{[0]} = -3, x_4^{[0]} = -10, x_{5,6}^{[0]} = 2 \pm 3i, \right] \tag{135}$$

introduces significant variability in the convergence behavior of iterative methods, as illustrated in Table 9.

The results clearly indicate that, under highly perturbed initial conditions, the rate of convergence decreases and the time required to estimate the roots increases—particularly when the fractional parameter approaches 1.0. Despite these challenges, the proposed scheme $SC^{[\sigma]}$ consistently outperforms existing methods in terms of residual error and computational time. It also demonstrates significantly improved convergence behavior in noisy environments, highlighting its robustness, long-term reliability, and global convergence capability.

6.1.1. *Bratu-type fractional boundary problem: Physical interpretation*

The solution to the Bratu-type fractional boundary value problem provides a physical simulation of the spatial behavior of nonlocal, nonlinear quantities such as temperature or concentration.

- Temperature or Concentration Profile: In heat transfer and chemical reaction problems, the function $U(x)$ typically represents the temperature distribution, concentration profile, or potential field within a given domain. The solution to the fractional Bratu-type equation captures how these quantities evolve spatially, particularly in systems characterized by nonlinearity and nonlocality.
- Effect of Fractional Order: The fractional parameter σ governs the extent of memory and hereditary effects in the system. Smaller values of σ correspond to stronger nonlocal behavior and memory effects, resulting in slower diffusion and smoother, more spread-out profiles. In contrast, larger values of σ resemble classical differential models, leading to faster propagation and sharper gradients.
- Implications for Numerical Modeling: Varying the fractional order σ effectively simulates systems with different degrees of long-range interaction and memory. Numerical solutions for various values of σ confirm that the proposed scheme accurately captures complex dynamic behavior. This highlights the method's robustness and effectiveness in modeling real-world engineering processes involving nonlocal effects.

6.2. *Civil engineering application: Bratu-type fractional boundary problem-engineering application*

The Bratu-type fractional boundary problem is a boundary value problem governed by fractional differential equations. It generalizes the classical Bratu problem by incorporating a fractional formulation of a second-order ordinary differential equation subject to boundary conditions. This model

Table 10
Results of $SC^{[\sigma]}$ for solving Eq. (141) using close initial guesses.

σ	$e_1^{[3]}$	$e_2^{[3]}$	$e_3^{[3]}$	$e_4^{[3]}$	$e_5^{[3]}$	$e_6^{[3]}$	$e_7^{[3]}$
0.1	3.43	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	1.19×10^{-12}	1.19×10^{-12}
0.3	0.03	1.19×10^{-32}	4.19×10^{-35}	1.08×10^{-17}	5.68×10^{-29}	0.0	0.0
0.5	0.75	0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	1.19×10^{-32}	1.19×10^{-32}
0.7		0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	1.19×10^{-32}	0.0
0.9	0.77	0.0	0.0	0.0	0.0	42	0.0

$e_8^{[3]}$	$e_9^{[3]}$	$e_{10}^{[3]}$	$e_{11}^{[3]}$	$e_{12}^{[3]}$	$e_{13}^{[3]}$	$\rho_i^{[j-1]}$
1.19×10^{-32}	5.87×10^{-15}	8.37×10^{-14}	1.64×10^{-13}	1.33×10^{-10}	18	9.3087
1.19×10^{-32}	1.65×10^{-32}	4.19×10^{-35}	1.08×10^{-37}	5.68×10^{-39}	17	7.6435
1.19×10^{-32}	0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	35	8.8787
0.0	0.0	0.0	0.0	0.0	0.0	8.8787
0.0	0.0	0.0	0.0	0.0	0.0	21.9999

has significant applications in engineering domains such as heat transfer, chemical reaction kinetics, and combustion processes [70]. The problem is formulated as follows:

$$\begin{cases} \left[{}_{C}\mathfrak{D}_{\sigma_1}^{\sigma}\right] U(x) - (U(x))^2 = G(x); 0 \le x \le 1, \\ U(0) = 0, \\ U'(0) = 0, \\ U(1) = 0, \\ U'(1) = 0, \end{cases} \tag{136}$$

where the function $G(x)$ is given by:

$$G(x) = -10x^{10} + 9x^9 - 4x^8 - 4x^7 + 8x^6 - 4x^4, 3 < \sigma \le 4. \tag{137}$$

The order σ satisfies $3 < \sigma \le 4$. In this context, σ is not restricted to integer values, enabling a more flexible mathematical model that captures complex phenomena such as memory effects and long-range interactions through fractional derivatives.

Solving such problems typically requires advanced techniques from fractional calculus, including fractional integration and differentiation, along with numerical methods specifically designed for fractional differential equations. Following the approach proposed in [71,72], we approximate the solution using a polynomial expansion truncated after two terms, as follows:

$$U(x) = \left[3.379805x^2 - 0.5478633333x^3 - 2x^{\sigma}\left[\ \frac{\Gamma(5)}{\Gamma(\sigma+1)} - \varphi_1 x + \varphi_2 x^2\ \right]\right], \tag{138}$$

where

$$\varphi_1 = \left(\frac{\Gamma(6)}{2\Gamma(\sigma+2)}\right) + \frac{3\left(16 - 4.99255^2\right)}{2\Gamma(\sigma+2)}x^3 - \left(\frac{-0.5047705542}{\Gamma(\sigma+6)}\right)x^4 - \left(\frac{965.7110970}{\Gamma(\sigma+7)}\right)x^5 \tag{139}$$

$$\varphi_2 = \begin{matrix} -\frac{2\Gamma(8)}{\Gamma(\sigma+8)}x^5 + \frac{2\Gamma(9)}{\Gamma(\sigma+9)}x^6 \\ -\frac{2\Gamma(10)}{\Gamma(\sigma+10)}x^7 + \frac{\Gamma(11)}{\Gamma(\sigma+11)}x^8. \end{matrix} \tag{140}$$

For $\sigma = 3.75$, we obtain:

$$U(x) = \begin{bmatrix} 3.379805000x^2 - 0.5478633333x^3 - 2.893970956x^{3.75} \\ +1.523142609x^{4.75} - 0.3758819712\times10^{-2}x^{7.75} \\ -0.2143152472\times10^{-2}x^{8.75} + 0.8427807740\times10^{-3}x^{9.75} \\ +0.9275885860\times10^{-3}x^{10.75} - 0.6315496756\times10^{-3}x^{11.75} \\ +0.4457997712\times10^{-3}x^{12.75} - 0.1621090076\times10^{-3}x^{13.75} \end{bmatrix}. \tag{141}$$

The exact roots, accurate to four decimal places, are:

$$\begin{bmatrix} \xi_1 = 0, \xi_2 = 2.8508, \xi_{3,4} = 1.3990 \pm 6506i, \\ \xi_{5,6} = 2.0920 \pm 1.7155i, \xi_{7,8} = 0.5112 \pm 2.8802i, \\ \xi_{9,10} = -1.2206 \pm 2.4702i, \xi_{11,12} = -2.4389 \pm 0.9873i. \end{bmatrix}. \tag{142}$$

We approximate all solutions to (141) using the following initial estimates:

$$\begin{bmatrix} x_1^{[0]} = 0.01, x_2^{[0]} = 2.8, x_{3,4}^{[0]} = 1.3 \pm 6i, x_{5,6}^{[0]} = 2.0 \pm 1.7i, x_{7,8}^{[0]} = 0.5 \pm 2.8i, \\ x_{9,10}^{[0]} = -1.2 \pm 2.4i, x_{11,12}^{[0]} = -2.4 \pm 0.9i. \end{bmatrix} \tag{143}$$

The numerical analysis of the schemes used to solve this equation is presented in Tables 10 and 11.

Table 10 presents the results of the parallel schemes $SC^{[\sigma]}$ for solving equation (132) with varying values of σ. The residual error and the computational order of convergence confirm the theoretical convergence behavior of $SC^{[\sigma]}$. When the initial values are chosen close to the exact solutions, the accuracy of the scheme improves, and the number of required iterations decreases as σ increases from 0.1 to 1.0.

Tables 11 and 12 compare the numerical results of the proposed method with existing approaches from the literature, namely $PM^{[1]}$, $SS^{[1]}$, and $SS^{[2]}$. As shown in Table 11, the proposed numerical schemes achieve significantly lower residual errors and require fewer iterations (j) than the competing methods. The computational order of convergence confirms close alignment with the theoretical convergence rate. The stability analysis, presented in Table 12, evaluates key performance metrics, including the total number of arithmetic operations per iteration ($+, -, \times, \div$), the number of function and fractional derivative evaluations ($\wp, \wp'$), the percentage of successful convergence (Percentage-C), and the computational time in

Table 11
Results of parallel approaches for solving Eq. (141) using close initial guesses.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$e_5^{[j]}$	$e_6^{[j]}$	$e_7^{[j]}$
$SC^{[\sigma]}$	3.43	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	4.12×10^{-55}	0.0
$PM^{[1]}$	0.03	1.19×10^{-32}	4.19×10^{-35}	1.08×10^{-17}	5.68×10^{-29}	1.08×10^{-55}	0.0
$SS^{[1]}$	0.75	0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	6.55×10^{-55}	1.53×10^{-55}
$SS^{[2]}$	0.77	0.0	0.0	0.0	0.0	0.0	0.0

$e_8^{[3]}$	$e_9^{[3]}$	$e_{10}^{[3]}$	$e_{11}^{[3]}$	$e_{12}^{[3]}$	$e_{13}^{[3]}$	$\rho_i^{[j-1]}$
3.43	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	5.08×10^{-19}	6.3087
0.03	1.19×10^{-32}	4.19×10^{-35}	1.08×10^{-17}	5.68×10^{-29}	9.66×10^{-25}	2.6435
0.75	0.0	6.00×10^{-62}	3.02×10^{-44}	5.57×10^{-55}	0.0	2.8787
0.77	0.0	0.0	0.0	0.0	0.0	9.9999

Table 12
Stability results of the numerical schemes for solving Eq. (141).

Scheme	Iterations	$[+,-,\times,\div]$	$[\wp,\wp']$	Percentage-C	Computational-T
$SC^{[\sigma]}$	3	52	15	99.5458772	1.76×10^{-2}
$PM^{[1]}$	5	125	25	85.7766565	1.08×10^{-2}
$SS^{[1]}$	5	85	18	86.7646676	3.02×10^{-1}
$SS^{[2]}$	5	87	16	79.8756563	3.02542

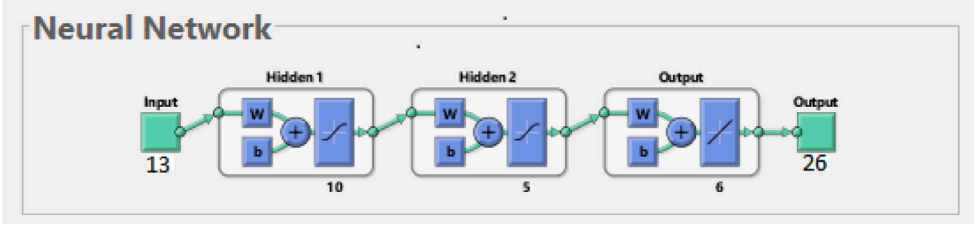
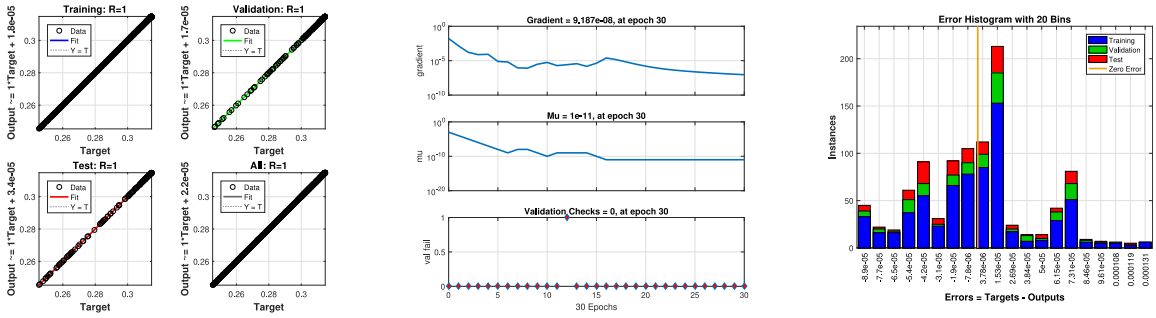


Fig. 9. Architecture of artificial neural networks for solving engineering Application 2 employing a parallel scheme.



(a) Regression curve (b) Transition statistics (c) Error histogram

Fig. 10. Regression analysis of the artificial neural network for solving Engineering Application 2: (a) regression curve, (b) transition statistics, and (c) error histogram.

seconds (Computational-T). These results clearly demonstrate that the proposed method $SC^{[\sigma]}$ outperforms $PM^{[1]}$, $SS^{[1]}$, and $SS^{[2]}$ in terms of both stability and efficiency. To further improve convergence, the parameter values used in the fractional schemes are optimized using dynamical plane analysis. The performance results of the hybrid ANN-based schemes are presented in Tables 13–16.

Fig. 10(a–c) provide a comprehensive overview of the neural network implementation and performance evaluation. The ANN was trained using the dataset described in Appendix C (see Table C.1), and its architecture for solving equation (141) is illustrated in Fig. 9.

- Fig. 10(a) presents the regression plots for training, validation, and test sets, confirming the accuracy and generalization capability of the neural network (with $R = 1$ across all phases).
- Fig. 10(b) displays the training progress curves, including the gradient, Levenberg–Marquardt parameter (μ), and validation checks over 30 epochs, highlighting the stability and convergence of the optimization process.
- Fig. 10(c) shows the error histogram, illustrating the distribution of prediction errors across all subsets and reflecting the overall consistency of the trained network.

These visualizations, together with the numerical results reported in Table 12, confirm the reliability and robustness of the ANN-based computations for this engineering application.

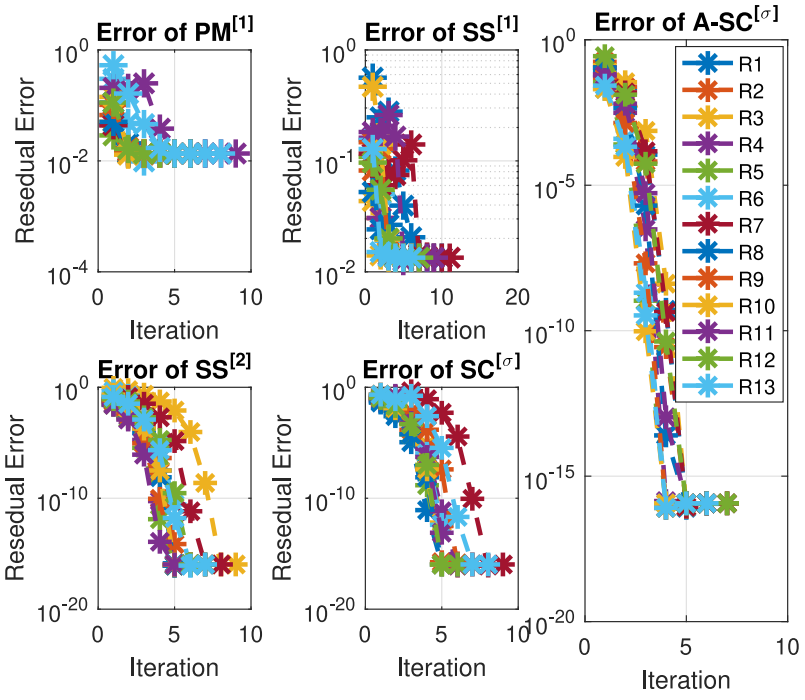


Fig. 11. Error plot of the parallel scheme for solving Engineering Application 2 using randomly generated test vectors.

Table 13
Results of parallel and hybrid parallel schemes using random initial guesses.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$e_5^{[j]}$	$e_6^{[j]}$	$e_7^{[j]}$
SC$^{[\sigma]}$	0.0	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	3.52×10^{-24}	0.0
PM$^{[1]}$	3.42×10^{-24}	1.54×10^{-12}	4.54×10^{-25}	1.08×10^{-17}	1.22×10^{-19}	5.42×10^{-24}	0.0
SS$^{[1]}$	2.04×10^{-24}	0.0	6.00×10^{-12}	3.02×10^{-24}	1.46×10^{-15}	6.22×10^{-14}	0.0
SS$^{[2]}$	1.12×10^{-24}	0.0	0.0	0.0	0.0	3.11×10^{-14}	0.0
A-SC$^{[\sigma]}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0

$e_8^{[3]}$	$e_9^{[3]}$	$e_{10}^{[3]}$	$e_{11}^{[3]}$	$e_{12}^{[3]}$	$e_{13}^{[3]}$	$\rho_i^{[j-1]}$
9.99×10^{-15}	5.15×10^{-15}	8.10×10^{-14}	1.76×10^{-13}	1.33×10^{-10}	0.0	6.3087
6.67×10^{-12}	6.49×10^{-12}	4.19×10^{-15}	1.08×10^{-17}	1.68×10^{-19}	0.0	2.6435
0.0	0.0	6.00×10^{-12}	3.02×10^{-14}	5.27×10^{-15}	0.0	2.8787
0.0	0.0	6.00×10^{-12}	3.02×10^{-14}	3.47×10^{-15}	0.0	2.8787
0.0	0.0	0.0	0.0	0.0	0.0	9.9999

Table 14
Consistency analysis of ANN and parallel scheme results using random initial values.

Scheme	$\vartheta^{[*]}$	Maximum-E	Computational-T
SC$^{[\sigma]}$	0.1	5.15×10^{-25}	4.1009978758
PM$^{[1]}$	1.2	5.15×10^{-15}	5.1039860065
SS$^{[1]}$	$5.5i$	1.19×10^{-12}	6.1000787567
SS$^{[2]}$	0.4	1.19×10^{-12}	5.1699995662
A-SC$^{[\sigma]}$	$0.9i$	1.19×10^{-34}	0.0445452245

Table 15
Efficiency analysis of ANN and parallel scheme results using random initial values.

Scheme	$\vartheta^{[*]}$	Local-C	Iterations
SC$^{[\sigma]}$	0.1	7.985008653	21
PM$^{[1]}$	1.2	5.70045688	38
SS$^{[1]}$	$5.5i$	9.194454565	28
SS$^{[2]}$	0.4	7.190086768	26
A-SC$^{[\sigma]}$	$0.9i$	11.07000866	7

The results of the A-SC$^{[\sigma]}$ scheme with randomly generated initial guesses (see Appendices B and C), along with those of the parallel schemes SC$^{[\sigma]}$, PM$^{[1]}$, and SS$^{[1]}$–SS$^{[2]}$, are presented in Table 13. As shown in the table, A-SC$^{[\sigma]}$ consistently outperforms the other methods in terms of both accuracy and convergence speed. This improvement is further illustrated in Fig. 11.

Tables 14 and 15 present the efficiency and stability analysis of parallel schemes incorporating the hybrid ANN-based approach A-SC$^{[\sigma]}$ with randomly generated initial values. Informed by insights from dynamical analysis, the numerical results demonstrate that A-SC$^{[\sigma]}$ offers significant

Table 16
Stability Analysis: A-SC$^{[\sigma]}$ and parallel systems' results in terms of operations and percentage convergence using random initial values.

Scheme	$\vartheta^{[*]}$	$[+,-,\times,\div]$	$[\wp,\wp']$	Percentage-C
SC$^{[\sigma]}$	0.1	65	19	81.854644262
PM$^{[1]}$	1.2	157	31	75.124526724
SS$^{[1]}$	$5.5i$	102	28	88.764005757
SS$^{[2]}$	0.4	109	26	59.870000576
A-SC$^{[\sigma]}$	$0.9i$	45	27	97.199868868

Table 17
Results of parallel and hybrid parallel algorithms with highly perturbed initial guesses.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$e_5^{[j]}$	$e_6^{[j]}$	$e_7^{[j]}$
SC$^{[\sigma]}$	0.0	0.18×10^{-15}	9.95×10^{-34}	0.0	0.0	4.65×10^{-14}	0.0
PM$^{[1]}$	0.24×10^{-13}	5.55×10^{-11}	7.67×10^{-21}	1.08×10^{-17}	0.0	7.86×10^{-24}	0.0
SS$^{[1]}$	5.06×10^{-19}	0.0	6.00×10^{-11}	0.0	1.46×10^{-15}	4.82×10^{-14}	0.0
SS$^{[2]}$	4.46×10^{-21}	0.0	0.0	0.0	0.0	6.81×10^{-14}	0.0
A-SC$^{[\sigma]}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0

$e_8^{[3]}$	$e_9^{[3]}$	$e_{10}^{[3]}$	$e_{11}^{[3]}$	$e_{12}^{[3]}$	$e_{13}^{[3]}$	C-Time
3.30×10^{-35}	5.10×10^{-35}	0.10×10^{-24}	1.76×10^{-23}	0.0	0.0	1.3087
1.20×10^{-9}	3.09×10^{-11}	4.34×10^{-15}	1.08×10^{-17}	1.69×10^{-19}	0.0	3.7655
0.0	0.0	5.98×10^{-12}	3.02×10^{-14}	5.27×10^{-15}	0.0	4.3367
0.0	0.0	6.56×10^{-12}	3.02×10^{-14}	3.47×10^{-15}	0.0	2.57687
0.0	0.0	0.0	0.0	0.0	0.0	0.9999

improvements over existing methods. In particular, the scheme achieves enhanced local convergence, requires fewer iterations, reduces both maximum and residual errors (as shown in Fig. 3), and lowers CPU time.

The stability results in Table 16 demonstrate that the proposed scheme A-SC$^{[\sigma]}$ outperforms SC$^{[\sigma]}$, PM$^{[1]}$, and SS$^{[1]}$–SS$^{[2]}$ in terms of total arithmetic operations per iteration, function and fractional derivative evaluations, percentage convergence, and computational time (in seconds).

The visualizations in Fig. 10(a–c), together with the results in Tables 14–16, confirm the reliability and stability of the neural network-based computations for this engineering application. Selecting initial guess values in a high-noise environment—i.e., values placed far from the exact roots, such as

$$\left[\begin{array}{c} x_1^{[0]}=3.7, x_2^{[0]}=6.5, x_{3,4}^{[0]}=4\pm4i, x_{5,6}^{[0]}=3.0\pm5i, x_{7,8}^{[0]}=5\pm5i, \\ x_{9,10}^{[0]}=-1\pm3i, x_{11,12}^{[0]}=-1\pm2i. \end{array}\right] \tag{144}$$

introduces significant variability in the convergence behavior of iterative methods, as demonstrated in Table 17.

The results clearly indicate that, under highly perturbed initial conditions, the convergence rate decreases and the time required to approximate the roots increases—particularly when the fractional parameter approaches 1.0. Despite these challenges, the proposed scheme SC$^{[\sigma]}$ consistently outperforms existing methods in terms of residual error and computational time. Notably, it exhibits markedly better convergence behavior in noisy environments, demonstrating both long-term reliability and robust global convergence capabilities.

6.2.1. Physical interpretation of the problem

The solution to the Bratu-type fractional boundary value problem provides a physical representation of the spatial distribution of nonlocal, nonlinear quantities such as temperature or concentration.

- The solution to the Bratu-type fractional boundary problem describes the spatial distribution of physical quantities such as temperature or concentration in nonlinear, nonlocal systems.
- The fractional parameter σ governs the memory and hereditary effects within the system. Smaller values of σ correspond to stronger nonlocal behavior and memory effects, resulting in broader profiles and reduced diffusion.
- Interpretation in Relation to Numerical Schemes: The fractional order σ encapsulates memory and long-range interaction effects—where lower values of σ correspond to more pronounced nonlocal behavior. A comparison of numerical solutions for varying σ values confirms the scheme's ability to accurately capture complex system dynamics, thereby demonstrating its effectiveness in modeling real-world engineering processes.

6.3. Time-fractional Riccati equation: An engineering problem

The time-fractional Riccati equation is a fractional differential equation in time, commonly formulated as [73]:

$$\begin{cases} \left[{}_C\mathfrak{D}_{\sigma_1}^{\sigma}\right] U(x)-(U(x))^2=x; 0\le x\le 1, \\ U(0)=0, \\ U(1)=0, \end{cases} \tag{145}$$

where the fractional order σ satisfies $0<\sigma\le 2$. This equation models systems with memory effects and anomalous diffusion, and finds applications in control theory, physics, and biology. The fractional order σ introduces nonlocal temporal behavior, allowing the current state to depend on the entire history of the system.

Table 18
Results of $SC^{[\sigma]}$ for solving Eq. (149) using close initial guesses.

σ	$e_1^{[3]}$	$e_2^{[3]}$	$e_3^{[3]}$	$e_4^{[3]}$	$\rho_i^{[j-1]}$
0.1	9.103×10^{-3}	9.180×10^{-15}	1.187×10^{-14}	2.762×10^{-13}	9.45865
0.3	0.102×10^{-2}	4.549×10^{-32}	0.0	0.0	7.62246
0.5	3.894×10^{-3}	0.0	7.650×10^{-62}	0.0	8.57675
0.7	5.459×10^{-1}	0.0	6.335×10^{-62}	7.424×10^{-44}	8.65466
0.9	7.175×10^{-2}	1.096×10^{-32}	0.0	0.0	20.0032

Table 19
Results of parallel approaches for solving Eq. (149) using close initial guesses.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$\rho_i^{[j-1]}$
$SC^{[\sigma]}$	3.43	9.1564×10^{-75}	0.0	0.0	9.305687
$PM^{[1]}$	0.03	1.3419×10^{-42}	9.1954×10^{-35}	9.0843×10^{-37}	5.006754
$SS^{[1]}$	0.75	1.3419×10^{-52}	8.0350×10^{-62}	0.0	7.887667
$SS^{[2]}$	0.77	1.3419×10^{-42}	0.0	0.0	6.8754679

Table 20
Stability results of the numerical schemes for solving Eq. (149).

Scheme	Iterations	$[+,-,\times,\div]$	$[\wp,\wp']$	Percentage-C	Computational-T
$SC^{[\sigma]}$	2	52	11	99.5458772	0.076×10^{-2}
$PM^{[1]}$	4	125	20	85.7766565	1.078×10^{-2}
$SS^{[1]}$	5	85	17	86.7646676	2.062×10^{-1}
$SS^{[2]}$	6	87	13	79.8756563	3.652542

To address such problems, specialized techniques from fractional calculus — such as fractional integration and differentiation — along with numerical methods tailored for fractional differential equations, are often required. Following the approach presented in [74–76], we approximate the solution using a polynomial expansion truncated after two terms, as follows:

$$U(x)=\left[\frac{x^{\sigma}}{\Gamma(\sigma+1)}+\frac{2x^{2\sigma}B_1}{\Gamma^3(\sigma)\sigma^2}+\frac{2B_2}{\Gamma^7(\sigma)\sigma^4}\right], \tag{146}$$

where

$$B_1=\int_0^t (x-t)^{\sigma-1}\,dt;\ B_2=\int_0^t (x-t)^{\sigma-1}\,B^{[*]}dt, \tag{147}$$

and

$$B^{[*]}=x^{2\sigma}\sigma^2\Gamma^4(\sigma)+2x^{3\sigma}B_1\sigma\Gamma^2(\sigma)+x^{4\sigma}\left(B_1\right)^2. \tag{148}$$

For $\sigma=\frac{1}{2}$, we obtain:

$$U(x)=\left[\frac{2\sqrt{x}}{\sqrt{\pi}}+\frac{16x^{\frac{3}{2}}}{\pi^{\frac{3}{2}}}+\frac{16x^{\frac{3}{2}}\left(\pi^2+8\pi x+16x^2\right)}{\pi^{\frac{3}{2}}}\right]. \tag{149}$$

The exact roots, accurate to four decimal places, are:

$$\left[\ \xi_1=0,\xi_2=-0.3337,\xi_{3,4}=-0.3043\pm0.4446i\ \right]. \tag{150}$$

We approximate all solutions to Eq. (149) using the following initial estimates:

$$\left[x_1^{[0]}=0.01,\quad x_2^{[0]}=-0.3,\quad x_{3,4}^{[0]}=-0.3\pm0.4i\right].$$

The numerical analysis of the schemes used to solve this equation is presented in Tables 18 and 19.

Table 18 presents the results of the parallel schemes $SC^{[\sigma]}$ for solving equation (149) with different values of σ. The residual error and the computational order of convergence validate the theoretical performance of $SC^{[\sigma]}$. When the initial values are close to the exact solution, the accuracy of the scheme improves and the number of iterations decreases as σ increases from 0.1 to 1.0.

Tables 19 and 20 compare the numerical performance of the proposed scheme with existing methods from the literature, namely $PM^{[1]}$, $SS^{[1]}$, and $SS^{[2]}$. As shown in Table 19, the proposed fractional scheme $SC^{[\sigma]}$ achieves significantly lower residual errors and requires fewer iterations (j) compared to the reference methods. The computed order of convergence closely aligns with the theoretical predictions, confirming the reliability of the scheme. The stability analysis, summarized in Table 20, evaluates key performance metrics, including the total number of arithmetic operations per iteration ($+,-,\times,\div$), function and fractional derivative evaluations ($\wp,\wp'$), percentage of successful convergence (Percentage-C), and computational time (in seconds). The results clearly demonstrate that $SC^{[\sigma]}$ outperforms $PM^{[1]}$, $SS^{[1]}$, and $SS^{[2]}$ in both stability and computational efficiency. To further enhance convergence, the fractional parameter values were optimized using insights from dynamical plane analysis. The performance of the hybrid ANN-based schemes, also evaluated in Tables 19 and 20, further highlights the advantages of this approach.

Fig. 13(a–c) provide a comprehensive overview of the neural network's implementation and performance evaluation. The ANN was trained using the dataset described in Appendix C, and its architecture for solving equation (141) is shown in Fig. 12.

- Fig. 13(a) presents the regression plots for training, validation, and testing datasets, as well as the overall regression. The results indicate a high degree of accuracy and generalization, with $R=1$ in all cases.

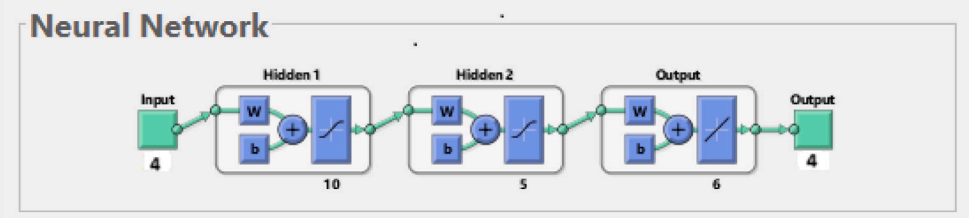
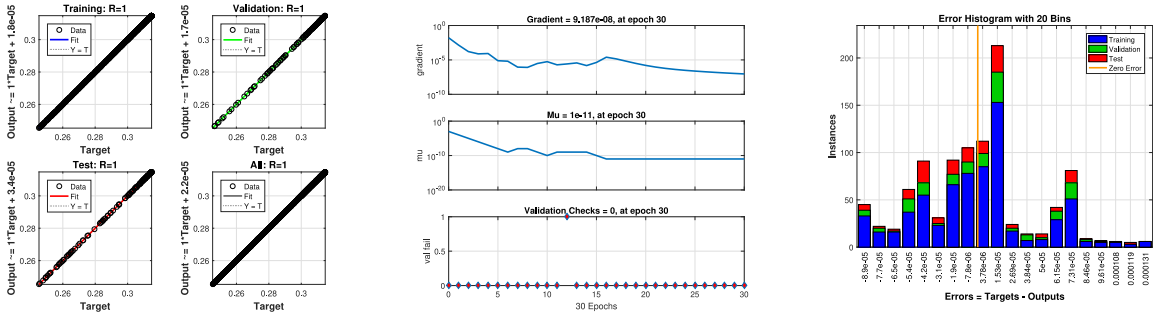


Fig. 12. Architecture of artificial neural networks for solving engineering Application 3 employing a parallel scheme.



(a) Regression curve (b) Transition statistics (c) Error histogram

Fig. 13. Regression analysis of the artificial neural network for solving Engineering Application 3: (a) regression curve, (b) transition statistics, and (c) error histogram.

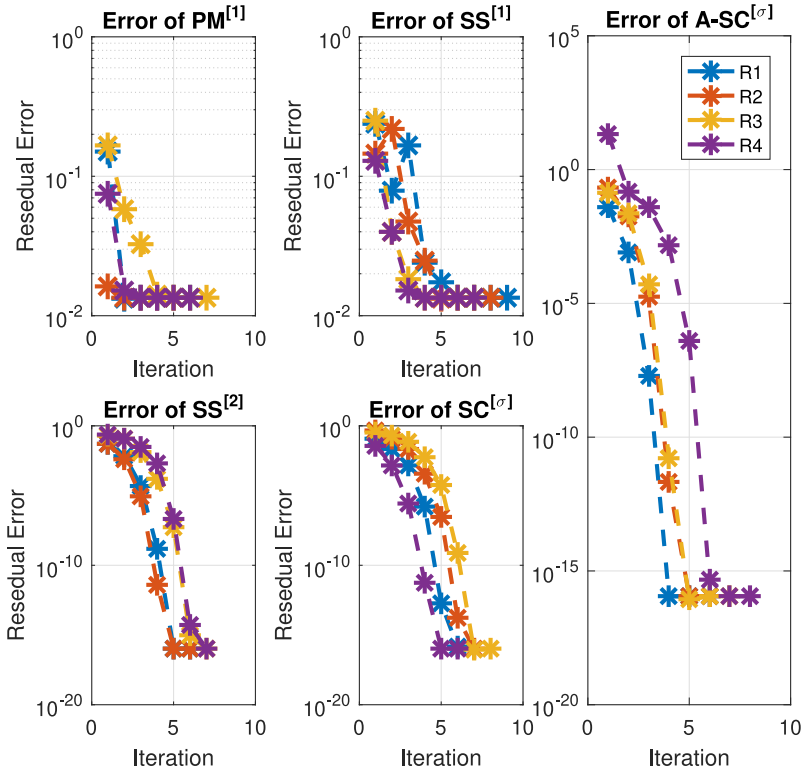


Fig. 14. Error plot of the parallel scheme for solving Engineering Application 3 using randomly generated test vectors.

- Fig. 13(b) shows the training progress through three subplots: the gradient curve, the Levenberg–Marquardt parameter μ, and the number of validation checks. These confirm smooth convergence with no early stopping.
- Fig. 13(c) displays the error histogram, which illustrates the distribution of prediction errors across the training, validation, and test sets. The narrow spread around zero indicates stable and consistent model performance.

Together with the numerical results in Table 20, these visualizations confirm the reliability and robustness of the ANN-based computations for this engineering application.

Table 21
Results of parallel and hybrid parallel methods utilizing random initial values.

Scheme	$e_1^{[j]}$	$e_2^{[j]}$	$e_3^{[j]}$	$e_4^{[j]}$	$\rho_i^{[j-1]}$
SC$^{[\sigma]}$	0.0	1.19×10^{-25}	0.0	0.0	8.87782
PM$^{[1]}$	9.42×10^{-21}	6.54×10^{-12}	9.94×10^{-15}	8.621×10^{-11}	6.74605
SS$^{[1]}$	8.04×10^{-19}	0.0	5.01×10^{-12}	8.026×10^{-21}	6.80877
SS$^{[2]}$	3.096×10^{-17}	0.0	0.0	0.0	7.00873
A-SC$^{[\sigma]}$	0.0	0.0	0.0	0.0	9.99996

Table 22
Consistency analysis of ANN and parallel scheme results using random initial values.

Scheme	$\vartheta^{[*]}$	Maximum-E	Computational-T
SC$^{[\sigma]}$	0.1	5.15×10^{-25}	4.1009978758
PM$^{[1]}$	1.2	5.15×10^{-15}	5.1039860065
SS$^{[1]}$	$5.5i$	1.19×10^{-12}	6.1000787567
SS$^{[2]}$	0.4	1.19×10^{-12}	5.1699995662
A-SC$^{[\sigma]}$	$0.9i$	1.19×10^{-34}	0.0445452245

Table 23
Efficiency analysis of ANN and parallel scheme results using random initial values.

Scheme	$\vartheta^{[*]}$	Local-C	Iterations
SC$^{[\sigma]}$	0.1	7.985008653	21
PM$^{[1]}$	1.2	5.70045688	38
SS$^{[1]}$	$5.5i$	9.194454565	28
SS$^{[2]}$	0.4	7.190086768	26
A-SC$^{[\sigma]}$	$0.9i$	11.07000866	7

Table 24
Stability Analysis: A-SC$^{[\sigma]}$ and parallel systems' results in terms of operations and percentage convergence on random initial values.

Scheme	$\vartheta^{[*]}$	$[+,-,\times,\div]$	$[\wp,\wp']$	Percentage-C
SC$^{[\sigma]}$	0.1	65	19	81.854644262
PM$^{[1]}$	1.2	157	31	75.124526724
SS$^{[1]}$	$5.5i$	102	28	88.764005757
SS$^{[2]}$	0.4	109	26	59.870000576
A-SC$^{[\sigma]}$	$0.9i$	45	27	97.199868868

Table 25
Results of parallel and hybrid parallel algorithms using highly perturbed initial guesses.

Scheme	$e_1^{[5]}$	$e_2^{[5]}$	$e_3^{[5]}$	$e_4^{[5]}$	C-Time
SC$^{[\sigma]}$	0.0	0.0	8.0×10^{-34}	0.0	0.30127
PM$^{[1]}$	5.15×10^{-14}	1.19×10^{-12}	4.9×10^{-25}	9.8×10^{-17}	3.66574
SS$^{[1]}$	5.15×10^{-11}	0.0	6.0×10^{-10}	0.0	3.89673
SS$^{[2]}$	0.0	0.0	0.0	0.0	1.99653
A-SC$^{[\sigma]}$	0.0	0.0	0.0	0.0	0.05355

The results of the A-SC$^{[\sigma]}$ scheme applied to random initial guess values (see Appendices B and C), along with the outcomes of the parallel schemes SC$^{[\sigma]}$, PM$^{[1]}$, and SS$^{[1]}$–SS$^{[2]}$, are summarized in Table 21. As shown in the table, A-SC$^{[\sigma]}$ consistently outperforms the other methods in terms of accuracy and convergence speed when initialized with random estimates. This improvement is further visualized in Fig. 14.

Tables 22 and 23 present the efficiency and stability analysis of the parallel schemes incorporating the hybrid ANN-based approach A-SC$^{[\sigma]}$, evaluated using random initial values. By leveraging insights from dynamical analysis, the numerical results clearly demonstrate that A-SC$^{[\sigma]}$ significantly outperforms existing methods. In particular, the scheme achieves improved local convergence, requires fewer iterations, yields lower maximum and residual errors (as shown in Fig. 14), and reduces overall computational time.

The stability results in Table 24 demonstrate that the proposed scheme A-SC$^{[\sigma]}$ outperforms SC$^{[\sigma]}$, PM$^{[1]}$, and SS$^{[1]}$–SS$^{[2]}$ across multiple performance metrics. These include the total number of arithmetic operations per iteration step, the number of function and corresponding fractional derivative evaluations, percentage of successful convergence, and computational time (in seconds).

These visualizations, together with the results in Tables 23–24, confirm the reliability and stability of the neural network-based computations for this engineering application. To further evaluate robustness, we consider a scenario involving large random perturbations in the initial guess values—that is, values set far from the exact roots. For instance:

$$\left[x_1^{[0]}=5.1,\quad x_2^{[0]}=-3.7,\quad x_{3,4}^{[0]}=-4\pm4i\right]$$

This configuration introduces significant variability in the convergence behavior of iterative approaches, as demonstrated by the results in Table 25.

Using highly perturbed initial estimates delays convergence and increases the computational time required to approximate the roots, particularly when the fractional parameter approaches 1.0. Despite these challenges, the proposed scheme SC$^{[\sigma]}$ outperforms existing methods in terms of

residual error and computational time. In fact, it demonstrates significantly better convergence behavior under noisy conditions, highlighting its long-term reliability and strong global convergence properties.

6.3.1. Physical interpretation of the time fractional Riccati equation

The solution to the time-fractional Riccati equation offers valuable physical insight into the temporal evolution of systems governed by nonlinear feedback and memory effects. In particular:

- It characterizes how the state variable — such as displacement, voltage, or concentration — evolves under the influence of nonlinear self-interaction, represented by the term $(U(x))^2$, and an external forcing term, exemplified by x.
- The fractional-order derivative captures history-dependent behavior, meaning that, unlike in classical models, the solution reflects how past states continuously influence the present state of the system.
- In the case $\sigma < 1$, the solution typically exhibits subdiffusive or slower dynamics, indicating that the system's response diminishes progressively over time—a characteristic behavior of materials with thermal, electrical, or mechanical memory.
- In control and physical systems, the evolution and shape of the solution offer predictive insights into long-term behavior, support stability analysis, and enable the assessment of overall system performance.
- Interpretation in Relation to Numerical Schemes: The fractional order σ captures memory effects and long-range interactions—smaller values of σ correspond to stronger nonlocal influences. The comparison of numerical solutions across different σ values demonstrates the proposed scheme's ability to accurately model complex dynamic behavior, confirming its effectiveness in simulating real-world engineering processes.

7. Conclusion and future work

This study introduced a new class of high-order Caputo-type fractional iterative methods for solving nonlinear equations, achieving sixth-order convergence for single-root problems and order $5\sigma + 1$ for the fractional extension. To enhance computational performance, a parallel correction scheme was developed, accelerating the convergence rate to $20\sigma + 8$. Key parameter values were optimized using dynamical systems analysis and incorporated into a hybrid parallel framework, resulting in significant improvements in solution accuracy, stability, and computational efficiency. Theoretical convergence analysis confirmed the effectiveness of the proposed methods, and extensive numerical experiments validated their superior performance across a broad range of fractional parameters σ. The fractional schemes $SC^{[\sigma]}$ and A-$SC^{[\sigma]}$ consistently outperformed existing approaches, including $PM^{[1]}$ and $SS^{[1]}$–$SS^{[2]}$. Moreover, fractal behavior analysis (Table 1 and Figs. 1–5) demonstrated the robustness and stability of the proposed methods. When benchmarked against classical techniques, the hybrid ANN-based scheme A-$SC^{[\sigma]}$ achieved lower residual errors, faster convergence, and reduced computational cost (Tables 5–25; Figs. 6–14).

The key contributions of this work are as follows:

- Development of a novel class of high-order fractional iterative schemes that extend classical root-finding methods.
- Design of an efficient hybrid parallel framework that accelerates convergence and improves computational efficiency.
- Optimization of fractional-order parameters through dynamical systems analysis, enhancing stability and reliability.
- Integration of artificial neural networks (ANNs) to improve initial approximations and further increase robustness and accuracy.

Future research directions include:

- Exploring alternative definitions of fractional derivatives to improve accuracy and stability.
- Extending the proposed methods to multidimensional nonlinear systems and fractional differential equations.
- Enhancing adaptability through ANN-driven frameworks that dynamically tune fractional parameters during the iteration process.
- Applying the hybrid parallel approach to large-scale engineering, physics, and data science problems by leveraging high-performance computing architectures.

These research directions have the potential to substantially expand the applicability of fractional iterative schemes in addressing complex nonlinear problems across a wide range of scientific and engineering disciplines.

CRediT authorship contribution statement

Mudassir Shams: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Nasreen Kausar:** Writing – review & editing, Writing – original draft, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization. **Bruno Carpentieri:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

In accordance with the ethical standards of the journal, Author would like to declare that the authors of this manuscripts have no conflicts of interest that could have influenced the research or it outcomes, Specifically:

1. **Financial Interests**: None of the authors have any financial interests, such as employment, stock ownerships, honoraria, paid expert testimony, or patent applications registrations, that may perceive as affecting the objectivity or impartially of the research.
2. **Personal Relationships**: None of the authors have personal relationships or affiliations with organizations or individuals that could potentially bias the work.
3. **Academic and Intellectual Conflicts**: There are no academic or intellectual conflicts that could have influenced the content or conclusion of the manuscripts. Professional or Institutional Affiliations: The affiliations of the authors, as listed on the manuscript, have not led to any conflicts of interest related to this research.
4. **Funding Sources**: The research was conducted without any funding from organizations that could be viewed as having a vested interest in the outcome of the study.

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), and this work was partially supported by INdAM-GNCS under the *Progetti di Ricerca 2024* program.

Appendix A

The coefficients of $\epsilon^{[j]}$ were used in the proof of Theorem 2.

$$\Lambda^{\{1\}} = \begin{pmatrix} -12\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4 + 16\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^5 \\ +24\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{pmatrix},$$

$$\Lambda^{\{2\}} = \begin{pmatrix} 28\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - \\ 10\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) - 6\vartheta^{[*]}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right)^2 \end{pmatrix},$$

$$\Lambda^{\{3\}} = \begin{pmatrix} 20\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp^{iv}(\xi)}{4!\wp'(\xi)}\right) + 4\vartheta^{[*]}\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \\ -8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp^{v}(\xi)}{5!\wp'(\xi)}\right) \end{pmatrix},$$

$$\Lambda^{\{4\}} = 2\left(-\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right) + 2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\right),$$

$$\Lambda^{\{5\}} = \begin{pmatrix} 2\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 - 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 - \\ 2\vartheta^{[*]}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) + 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{pmatrix},$$

$$\Lambda^{\{6\}} = \begin{pmatrix} -5\,\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 + 8\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4 \\ +7\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - \\ 11\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - 3\,\vartheta^{[*]}\hbar_4 + 6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\hbar_4 \end{pmatrix},$$

$$\Lambda^{\{7\}} = \begin{pmatrix} 2\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 - 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 - \\ 2\vartheta^{[*]}\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) + 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) \end{pmatrix}^2,$$

$$\Lambda^{\{8\}} = \begin{pmatrix} -\vartheta^{[*]2}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2 - \vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^3 + \\ 4\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^4 + 7\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - \\ 11\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\left(\frac{\wp'''(\xi)}{3!\wp'(\xi)}\right) - 3\vartheta^{[*]}\hbar_4 + 6\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)\hbar_4 \end{pmatrix},$$

$$\Lambda^{\{9\}} = \left(-\vartheta^{[*]}\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right) + 2\left(\frac{\wp''(\xi)}{2!\wp'(\xi)}\right)^2\right)$$

$$\Lambda_1^{[*]} = \begin{pmatrix} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} \\ -\frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{pmatrix},$$

$$\Lambda_2^{[*]} = 1/2\,\frac{\left(\Lambda_2^{[**]} + \Lambda_1^{[**]}\right)}{(\Gamma(\sigma))^2\,\sigma^2\pi^{3/2}\,(2^\sigma)^2\,\Gamma(\sigma+1/2)},$$

$$\Lambda_1^{[**]} = \begin{pmatrix} +2\,\hbar_3\,(\Gamma(\sigma))^2\,\sigma^2\pi^{3/2}\,(2^\sigma)^2\,\Gamma(\sigma+1/2) \\ -\hbar_3\,(3^\sigma)^3\,\sqrt{3}\Gamma(\sigma+1/3)\,\Gamma(\sigma+2/3)\,\Gamma(\sigma)\,\sigma\,\pi \end{pmatrix},$$

$$\Lambda_2^{[**]} = \begin{pmatrix} 2\,(2^\sigma)^6\,(\Gamma(\sigma+1/2))^3\,\hbar_2{}^2\sqrt{\pi} - \\ 2\,(2^\sigma)^4\,(\Gamma(\sigma+1/2))^2\,\hbar_2{}^2\Gamma(\sigma)\,\sigma\,\pi \end{pmatrix},$$

$$\Lambda_3^{[*]} = \left(\frac{\hbar_2{}^3\left(\Gamma(\sigma)\,\sigma\,\sqrt{\pi} - (2^\sigma)^2\,\Gamma(\sigma+1/2)\right)^2}{(\Gamma(\sigma))^2\,\sigma^2\pi}\right),$$

$$\Lambda_4^{[*]} = 2\left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + \\ 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} - \frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right)\hbar_2,$$

$$\Lambda_5^{[*]} = \left(\frac{\hbar_2{}^2\left(\Gamma(\sigma)\sigma\sqrt{\pi} - (2^\sigma)^2\,\Gamma(\sigma+1/2)\right)\Lambda_3^{[**]}}{(\Gamma(\sigma))^3\,\sigma^3\pi^2\,(2^\sigma)^2\,\Gamma(\sigma+1/2)}\right),$$

$$\Lambda_3^{[**]} = \left(\begin{array}{c} 2\,(2^\sigma)^6\,(\Gamma(\sigma+1/2))^3\,\hbar_2{}^2\sqrt{\pi} - 2\,(2^\sigma)^4 \\ (\Gamma(\sigma+1/2))^2\,\hbar_2{}^2\Gamma(\sigma)\,\sigma\,\pi + \\ 2\,\hbar_3\,(\Gamma(\sigma))^2\,\sigma^2\pi^{3/2}\,(2^\sigma)^2\,\Gamma(\sigma+1/2) \\ -\hbar_3\,(3^\sigma)^3\,\sqrt{3}\Gamma(\sigma+1/3)\,\Gamma(\sigma+2/3)\,\Gamma(\sigma)\,\sigma\,\pi \end{array}\right),$$

$$\Lambda_6^{[*]} = 6\left(-\hbar_2 + \frac{(2^\sigma)^2\,\Gamma(\sigma+1/2)\,\hbar_2}{\Gamma(\sigma)\,\sigma\sqrt{\pi}}\right)$$
$$\left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + \\ 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} - \frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right)\hbar_3,$$

$$\Lambda_7^{[*]} = 3\left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + \\ 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} - \frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right)^2\hbar_3,$$

$$\Lambda_8^{[*]} = \left(\begin{array}{c} -3\left(-\hbar_2 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2}{\Gamma(\sigma)\sigma\sqrt{\pi}}\right)^2\hbar_3 - \\ 4\,\frac{\hbar_2{}^3\left(\Gamma(\sigma)\sigma\sqrt{\pi} - (2^\sigma)^2\Gamma(\sigma+1/2)\right)}{\Gamma(\sigma)\sigma\sqrt{\pi}}\left(-\hbar_2 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2}{\Gamma(\sigma)\sigma\sqrt{\pi}}\right) \end{array}\right),$$

$$\Lambda_9^{[*]} = \left(\begin{array}{c} -6\left(-\hbar_2 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2}{\Gamma(\sigma)\sigma\sqrt{\pi}}\right) \\ \Lambda_4^{[**]}\hbar_3 \\ -4\,\frac{\hbar_2{}^3\left(\Gamma(\sigma)\sigma\sqrt{\pi} - (2^\sigma)^2\Gamma(\sigma+1/2)\right)}{\Gamma(\sigma)\sigma\sqrt{\pi}}\Lambda_5^{[**]} \\ -2\,\frac{\Lambda_6^{[**]}\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi^{3/2}(2^\sigma)^2\Gamma(\sigma+1/2)} \\ \left(-\hbar_2 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2}{\Gamma(\sigma)\sigma\sqrt{\pi}}\right) \end{array}\right),$$

$$\Lambda_4^{[**]} = \left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + \\ 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} - \frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right),$$

$$\Lambda_5^{[**]} = \left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + \\ 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} \\ -\frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right),$$

$$\Lambda_6^{[**]} = \left(\begin{array}{c} 2\,(2^\sigma)^6\,(\Gamma(\sigma+1/2))^3\,\hbar_2{}^2\sqrt{\pi} - \\ 2\,(2^\sigma)^4\,(\Gamma(\sigma+1/2))^2\,\hbar_2{}^2\Gamma(\sigma)\,\sigma\,\pi \\ +2\,\hbar_3\,(\Gamma(\sigma))^2\,\sigma^2\pi^{3/2}\,(2^\sigma)^2 \\ \Gamma(\sigma+1/2) - \hbar_3\,(3^\sigma)^3 \\ \sqrt{3}\Gamma(\sigma+1/3)\,\Gamma(\sigma+2/3)\,\Gamma(\sigma)\,\sigma\,\pi \end{array}\right),$$

$$\Lambda_{10}^{[*]} = -2\left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} \\ -\frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right)\hbar_2.$$

$$\Lambda_{11}^{[*]} = \left(-\hbar_2 + \frac{(2^\sigma)^2\,\Gamma(\sigma+1/2)\,\hbar_2}{\Gamma(\sigma)\,\sigma\sqrt{\pi}}\right),$$

$$\Lambda_{12}^{[*]} = \left(\begin{array}{c} -\hbar_3 + \frac{(2^\sigma)^2\Gamma(\sigma+1/2)\hbar_2{}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}} + 1/2\,\frac{\hbar_3(3^\sigma)^3\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2\Gamma(\sigma+1/2)} \\ -\frac{(2^\sigma)^4(\Gamma(\sigma+1/2))^2\hbar_2{}^2}{(\Gamma(\sigma))^2\sigma^2\pi} \end{array}\right),$$

$$\Lambda_{13}^{[*]} = \left(-{\hbar_2}^3 - \frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^3}{(\Gamma(\sigma))^2 \sigma^2 \pi} + 2 \frac{{\hbar_2}^3 (2^\sigma)^2 \Gamma(\sigma+1/2)}{\Gamma(\sigma)\sigma\sqrt{\pi}} \right),$$

$$\Lambda_{14}^{[*]} = \begin{pmatrix} -4 \frac{{\hbar_2}^4 (2^\sigma)^4 (\Gamma(\sigma+1/2))^2}{(\Gamma(\sigma))^2 \sigma^2 \pi} + 2 \frac{{\hbar_2}^4 (2^\sigma)^2 \Gamma(\sigma+1/2)}{\Gamma(\sigma)\sigma\sqrt{\pi}} \\ +2 \frac{{\hbar_2}^4 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ -\frac{{\hbar_2}^2 \hbar_3 (3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{(\Gamma(\sigma))^2 \sigma^2 \pi} - \\ 2 {\hbar_2}^2 \hbar_3 + 2 \frac{(2^\sigma)^2 \Gamma(\sigma+1/2) {\hbar_2}^2 \hbar_3}{\Gamma(\sigma)\sigma\sqrt{\pi}} \\ + \frac{{\hbar_2}^2 \hbar_3 (3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2 \Gamma(\sigma+1/2)} \end{pmatrix},$$

$$\Lambda_{15}^{[*]} = \begin{pmatrix} 5 \frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^2 \sigma^2 \pi} + 3 {\hbar_2}^3 \hbar_3 - \\ 5 \frac{{\hbar_2}^3 (2^\sigma)^2 \Gamma(\sigma+1/2) \hbar_3}{\Gamma(\sigma)\sigma\sqrt{\pi}} - 2 {\hbar_2} {\hbar_3}^2 + 6 \frac{(2^\sigma)^6 (\Gamma(\sigma+1/2))^3 {\hbar_2}^5}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ -2 \frac{(3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3) {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^2 \sigma^2 \pi} - \\ 3/2 \frac{(3^\sigma)^6 (\Gamma(\sigma+1/3))^2 (\Gamma(\sigma+2/3))^2 {\hbar_2} {\hbar_3}^2}{(\Gamma(\sigma))^2 \sigma^2 \pi (2^\sigma)^4 (\Gamma(\sigma+1/2))^2} - 2 \frac{(2^\sigma)^8 (\Gamma(\sigma+1/2))^4 {\hbar_2}^5}{(\Gamma(\sigma))^4 \sigma^4 \pi^2} - \\ 8 \frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^5}{(\Gamma(\sigma))^2 \sigma^2 \pi} - 2 {\hbar_2}^5 + 6 \frac{(2^\sigma)^2 \Gamma(\sigma+1/2) {\hbar_2}^5}{\Gamma(\sigma)\sigma\sqrt{\pi}} \\ +2 \frac{(2^\sigma)^2 \Gamma(\sigma+1/2)(3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3) {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ +2 \frac{(3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3) {\hbar_2} {\hbar_3}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2 \Gamma(\sigma+1/2)} - 3 \frac{{\hbar_2}^3 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3 \hbar_3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \end{pmatrix},$$

$$\Lambda_{16}^{[*]} = \left({\hbar_2}^3 + \frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^3}{(\Gamma(\sigma))^2 \sigma^2 \pi} - 2 \frac{{\hbar_2}^3 (2^\sigma)^2 \Gamma(\sigma+1/2)}{\Gamma(\sigma)\sigma\sqrt{\pi}} \right),$$

$$\Lambda_{17}^{[*]} = \begin{pmatrix} 4 \frac{{\hbar_2}^4 (2^\sigma)^4 (\Gamma(\sigma+1/2))^2}{(\Gamma(\sigma))^2 \sigma^2 \pi} - 2 \frac{{\hbar_2}^4 (2^\sigma)^2 \Gamma(\sigma+1/2)}{\Gamma(\sigma)\sigma\sqrt{\pi}} - \\ 2 \frac{{\hbar_2}^4 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} + \frac{{\hbar_2}^2 \hbar_3 (3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{(\Gamma(\sigma))^2 \sigma^2 \pi} + \\ 2 {\hbar_2}^2 \hbar_3 - 2 \frac{(2^\sigma)^2 \Gamma(\sigma+1/2) {\hbar_2}^2 \hbar_3}{\Gamma(\sigma)\sigma\sqrt{\pi}} - \frac{{\hbar_2}^2 \hbar_3 (3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2 \Gamma(\sigma+1/2)} \end{pmatrix},$$

$$\Lambda_{18}^{[*]} = \begin{pmatrix} -5 \frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^2 \sigma^2 \pi} - 3 {\hbar_2}^3 \hbar_3 + \\ 5 \frac{{\hbar_2}^3 (2^\sigma)^2 \Gamma(\sigma+1/2) \hbar_3}{\Gamma(\sigma)\sigma\sqrt{\pi}} + 2 {\hbar_2} {\hbar_3}^2 - \\ 6 \frac{(2^\sigma)^6 (\Gamma(\sigma+1/2))^3 {\hbar_2}^5}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} + \\ 2 \frac{(3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3) {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^2 \sigma^2 \pi} \\ +3/2 \frac{(3^\sigma)^6 (\Gamma(\sigma+1/3))^2 (\Gamma(\sigma+2/3))^2 {\hbar_2} {\hbar_3}^2}{(\Gamma(\sigma))^2 \sigma^2 \pi (2^\sigma)^4 (\Gamma(\sigma+1/2))^2} + \\ 2 \frac{(2^\sigma)^8 (\Gamma(\sigma+1/2))^4 {\hbar_2}^5}{(\Gamma(\sigma))^4 \sigma^4 \pi^2} + 8 \frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^5}{(\Gamma(\sigma))^2 \sigma^2 \pi} \\ +2 {\hbar_2}^5 - 6 \frac{(2^\sigma)^2 \Gamma(\sigma+1/2) {\hbar_2}^5}{\Gamma(\sigma)\sigma\sqrt{\pi}} - \\ 2 \frac{(2^\sigma)^2 \Gamma(\sigma+1/2)(3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3) {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ -2 \frac{(3^\sigma)^3 \sqrt{3} \Gamma(\sigma+1/3)\Gamma(\sigma+2/3) {\hbar_2} {\hbar_3}^2}{\Gamma(\sigma)\sigma\sqrt{\pi}(2^\sigma)^2 \Gamma(\sigma+1/2)} + 3 \frac{{\hbar_2}^3 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3 \hbar_3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \end{pmatrix},$$

$$\Lambda_{19}^{[*]} = {\hbar_2}^3 \left(\sqrt{\pi} (2^\sigma)^4 (\Gamma(\sigma+1/2))^2 + (\Gamma(\sigma))^2 \sigma^2 \pi^{3/2} - 2 \Gamma(\sigma) (2^\sigma)^2 \Gamma(\sigma+1/2) \pi \sigma \right),$$

$$\Lambda_{20}^{[*]} = \begin{pmatrix} 4 \Gamma(\sigma) \pi^{3/2} (2^\sigma)^6 (\Gamma(\sigma+1/2))^3 \sigma {\hbar_2}^2 \\ -2 (\Gamma(\sigma))^2 \pi^2 (2^\sigma)^4 \\ (\Gamma(\sigma+1/2))^2 \sigma^2 {\hbar_2}^2 + \\ \sqrt{3} \Gamma(\sigma) \pi^{3/2} (2^\sigma)^2 \Gamma(\sigma+1/2) \\ (3^\sigma)^3 \Gamma(\sigma+1/3) \Gamma(\sigma+2/3) \sigma \hbar_3 - 2 \\ (2^\sigma)^8 (\Gamma(\sigma+1/2))^4 \pi {\hbar_2}^2 \\ +2 (\Gamma(\sigma))^3 \pi^{5/2} (2^\sigma)^2 \Gamma(\sigma+1/2) \sigma^3 \hbar_3 \\ -2 (\Gamma(\sigma))^2 \pi^2 (2^\sigma)^4 (\Gamma(\sigma+1/2))^2 \sigma^2 \hbar_3 \\ -\sqrt{3} (\Gamma(\sigma))^2 \pi^2 (3^\sigma)^3 \Gamma(\sigma+1/3) \Gamma(\sigma+2/3) \sigma^2 \hbar_3 \end{pmatrix},$$

$$\Delta^{[1]} = \begin{pmatrix} 4 (2^\sigma)^{12} (\Gamma(\sigma+1/2))^6 {\hbar_2}^4 \pi^{3/2} + \\ 16 (2^\sigma)^8 (\Gamma(\sigma+1/2))^4 {\hbar_2}^4 (\Gamma(\sigma))^2 \sigma^2 \pi^{5/2} \\ +4 {\hbar_2}^4 (\Gamma(\sigma))^4 \sigma^4 \pi^{7/2} (2^\sigma)^4 (\Gamma(\sigma+1/2))^2 - \\ 12 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3 {\hbar_2}^4 (\Gamma(\sigma))^3 \sigma^3 \pi^3 \\ -10 (2^\sigma)^8 (\Gamma(\sigma+1/2))^4 {\hbar_2}^2 \hbar_3 (\Gamma(\sigma))^2 \sigma^2 \pi^{5/2} + \end{pmatrix},$$

$$\Delta^{[2]} = \begin{pmatrix} 4\,(3^{\sigma})^{3}\,\sqrt{3}\Gamma\,(\sigma+1/3)\,\Gamma\,(\sigma+2/3)\,\hbar_2{}^{2}\hbar_3 \\ (\Gamma\,(\sigma))^{2}\,\sigma^{2}\pi^{5/2}\,(2^{\sigma})^{4}\,(\Gamma\,(\sigma+1/2))^{2} \\ -12\,(2^{\sigma})^{10}\,(\Gamma\,(\sigma+1/2))^{5}\,\hbar_2{}^{4}\Gamma\,(\sigma)\,\sigma\,\pi^{2} \\ -6\,\hbar_2{}^{2}\hbar_3\,(\Gamma\,(\sigma))^{4}\,\sigma^{4}\pi^{7/2}\,(2^{\sigma})^{4}\,(\Gamma\,(\sigma+1/2))^{2} \\ +10\,(2^{\sigma})^{6}\,(\Gamma\,(\sigma+1/2))^{3}\,\hbar_2{}^{2}\hbar_3\,(\Gamma\,(\sigma))^{3}\,\sigma^{3}\pi^{3} \end{pmatrix},$$

$$\Delta^{[3]} = \begin{pmatrix} +6\,(2^{\sigma})^{10}\,(\Gamma\,(\sigma+1/2))^{5}\,\hbar_2{}^{2}\hbar_3\Gamma\,(\sigma)\,\sigma\,\pi^{2} \\ -4\,(2^{\sigma})^{6}\,(\Gamma\,(\sigma+1/2))^{3}\,(3^{\sigma})^{3}\,\sqrt{3}\Gamma\,(\sigma+1/3) \\ \Gamma\,(\sigma+2/3)\,\hbar_2{}^{2}\hbar_3\Gamma\,(\sigma)\,\sigma\,\pi^{2}+ \\ 4\,\hbar_3{}^{2}\,(\Gamma\,(\sigma))^{4}\,\sigma^{4}\pi^{7/2}\,(2^{\sigma})^{4}\,(\Gamma\,(\sigma+1/2))^{2}- \end{pmatrix},$$

$$\Delta^{[4]} = \begin{pmatrix} \hbar_2 \begin{pmatrix} 4\,(3^{\sigma})^{3}\,\sqrt{3}\Gamma\,(\sigma+1/3)\,\Gamma\,(\sigma+2/3) \\ \hbar_3{}^{2}\,(\Gamma\,(\sigma))^{3}\,\sigma^{3}\pi^{3}\,(2^{\sigma})^{2}\,\Gamma\,(\sigma+1/2) \\ +3\,(3^{\sigma})^{6}\,(\Gamma\,(\sigma+1/3))^{2}\,(\Gamma\,(\sigma+2/3))^{2} \\ \hbar_3{}^{2}\,(\Gamma\,(\sigma))^{2}\,\sigma^{2}\pi^{5/2} \end{pmatrix} \end{pmatrix},$$

$$\Lambda_{22}^{[*]} = \begin{pmatrix} \sigma\,\hbar_2 + 2\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} \\ -2\,\hbar_2{}^{2} - \frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2}{\Gamma(\sigma)\sqrt{\pi}} \end{pmatrix},$$

$$\Lambda_{23}^{[*]} = \begin{pmatrix} \sigma\,\hbar_3 + 2\,\frac{\hbar_2{}^{3}(2^{\sigma})^{2}\Gamma(\sigma+1/2)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} - 2\,\frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{3}}{(\Gamma(\sigma))^{2}\sigma^{2}\pi} \\ -1/2\,\frac{\hbar_3(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sqrt{\pi}(2^{\sigma})^{2}\Gamma(\sigma+1/2)}+ \\ \frac{\hbar_3(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)\hbar_2}{\Gamma(\sigma)\sigma\,\sqrt{\pi}(2^{\sigma})^{2}\Gamma(\sigma+1/2)}- \\ 2\,\hbar_2\hbar_3 - \frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}}{\Gamma(\sigma)\sqrt{\pi}} + \frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{2}}{(\Gamma(\sigma))^{2}\sigma\,\pi} \end{pmatrix},$$

$$\Lambda_{24}^{[*]} = \begin{pmatrix} -\sigma\,\hbar_2{}^{3} - 6\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}\hbar_3}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} + 3\,\frac{\hbar_3(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{2}}{(\Gamma(\sigma))^{2}\sigma^{2}\pi}+ \\ 3\,\hbar_2{}^{2}\hbar_3 + 2\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{3}}{\Gamma(\sigma)\sqrt{\pi}} - \frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{3}}{(\Gamma(\sigma))^{2}\sigma\,\pi} \end{pmatrix},$$

$$\Lambda_{25}^{[*]} = \begin{pmatrix} -2\,\sigma\,\hbar_2{}^{2}\hbar_3 + 3\,\frac{\hbar_2\hbar_3{}^{2}(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{(\Gamma(\sigma))^{2}\sigma^{2}\pi}- \\ \frac{\hbar_2{}^{2}\hbar_3(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{(\Gamma(\sigma))^{2}\sigma\,\pi}+ \\ \frac{\hbar_2{}^{2}\hbar_3(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sqrt{\pi}(2^{\sigma})^{2}\Gamma(\sigma+1/2)} \\ -6\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{3}\hbar_3}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} + 12\,\frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{3}\hbar_3}{(\Gamma(\sigma))^{2}\sigma^{2}\pi} \\ -6\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2\hbar_3{}^{2}}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} - 6\,\frac{(2^{\sigma})^{6}(\Gamma(\sigma+1/2))^{3}\hbar_2{}^{3}\hbar_3}{(\Gamma(\sigma))^{3}\sigma^{3}\pi^{3/2}} \\ -3\,\frac{\hbar_2\hbar_3{}^{2}(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}(2^{\sigma})^{2}\Gamma(\sigma+1/2)}+ \\ 6\,\hbar_2\hbar_3{}^{2} + 2\,\frac{\hbar_2{}^{4}(2^{\sigma})^{2}\Gamma(\sigma+1/2)}{\Gamma(\sigma)\sqrt{\pi}} \\ -4\,\frac{\hbar_2{}^{4}(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}}{(\Gamma(\sigma))^{2}\sigma\,\pi}+ \\ 2\,\frac{\hbar_2{}^{4}(2^{\sigma})^{6}(\Gamma(\sigma+1/2))^{3}}{(\Gamma(\sigma))^{3}\sigma^{2}\pi^{3/2}} + 2\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}\hbar_3}{\Gamma(\sigma)\sqrt{\pi}} \end{pmatrix},$$

$$\Lambda_{26}^{[*]} = \left(-\sigma\,\hbar_2 - 2\,\frac{(2^{\sigma})^{2}\,\Gamma\,(\sigma+1/2)\,\hbar_2{}^{2}}{\Gamma\,(\sigma)\,\sigma\,\sqrt{\pi}} + 2\,\hbar_2{}^{2} + \frac{(2^{\sigma})^{2}\,\Gamma\,(\sigma+1/2)\,\hbar_2}{\Gamma\,(\sigma)\,\sqrt{\pi}}\right),$$

$$\Lambda_{27}^{[*]} = \begin{pmatrix} -\sigma\,\hbar_3 - 2\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{3}}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} + 2\,\frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{3}}{(\Gamma(\sigma))^{2}\sigma^{2}\pi}+ \\ 1/2\,\frac{\hbar_3(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sqrt{\pi}(2^{\sigma})^{2}\Gamma(\sigma+1/2)}- \\ \frac{\hbar_2\hbar_3(3^{\sigma})^{3}\sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}(2^{\sigma})^{2}\Gamma(\sigma+1/2)} + 2\,\hbar_2\hbar_3 \\ +\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}}{\Gamma(\sigma)\sqrt{\pi}} - \frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{2}}{(\Gamma(\sigma))^{2}\sigma\,\pi} \end{pmatrix},$$

$$\Lambda_{28}^{[*]} = \begin{pmatrix} \sigma\,\hbar_2{}^{3} + 6\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}\hbar_3}{\Gamma(\sigma)\sigma\,\sqrt{\pi}}- \\ 3\,\frac{\hbar_3(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{2}}{(\Gamma(\sigma))^{2}\sigma^{2}\pi} - 3\,\hbar_2{}^{2}\hbar_3 - 2\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{3}}{\Gamma(\sigma)\sqrt{\pi}}+ \\ \frac{(2^{\sigma})^{4}(\Gamma(\sigma+1/2))^{2}\hbar_2{}^{3}}{(\Gamma(\sigma))^{2}\sigma\,\pi}+ \\ \frac{\hbar_2\left(\Gamma(\sigma)\sqrt{\pi}\sigma^{2}-2\,\Gamma(\sigma)\sqrt{\pi}\sigma\,\hbar_2-(2^{\sigma})^{2}\Gamma(\sigma+1/2)\sigma+2\,(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2\right)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} \\ \left(\sigma\,\hbar_2 + 2\,\frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2{}^{2}}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} - 2\,\hbar_2{}^{2} - \frac{(2^{\sigma})^{2}\Gamma(\sigma+1/2)\hbar_2}{\Gamma(\sigma)\sqrt{\pi}}\right) \end{pmatrix},$$

Table B.1
Randomly generated initial guesses.

$\mathbf{x}^{[0]}$	$[x_1^{[0]}$	$x_2^{[0]}$	$x_3^{[0]}$	$x_4^{[0]}$	$x_5^{[0]}$	$x_6^{[0]}]$
$\mathbf{x}^{[0]}$	[0.13	0.65	0.19	0.82	0.54	0.64]
$\mathbf{x}^{[0]}$	[0.44	0.09	0.26	0.87	0.76	0.26]
$\mathbf{x}^{[0]}$	[0.93	0.45	0.27	0.19	0.86	0.98]

Table B.2
Randomly generated initial guesses.

$\mathbf{x}^{[0]}$	$[x_1^{[0]}$	$x_2^{[0]}$	$x_3^{[0]}$	$x_4^{[0]}$	$x_5^{[0]}$	$\cdots$	$x_{13}^{[0]}]$
$\mathbf{x}^{[0]}$	[0.97	0.75	0.18	0.43	0.54	$\cdots$	0.02]
$\mathbf{x}^{[0]}$	[0.12	0.09	0.40	0.09	0.76	$\cdots$	0.77]
$\mathbf{x}^{[0]}$	[0.98	0.05	0.11	0.69	0.86	$\cdots$	0.93]

Table C.1
The head of the output data set used to learn the real and imaginary roots of Applications 1–2.

$\zeta^{[0]}$	ζ_0	ζ_1	ζ_2	ζ_3	$[\cdots,\ldots]$	ζ_{13}
	[Re, Im]	[Re, Im]	[Re, Im]	[Re, Im]	$[\cdots,\ldots]$	[Re, Im]
$\zeta_1^{[0]}$	[0.95, 0.56]	[0.12, 0.23]	[0.42, 0.63]	[0.52, 0.36]	$[\cdots,\ldots]$	[0.02, 0.45]
$\zeta_2^{[0]}$	[0.13, 0.93]	[0.54, 0.74]	[0.37, 0.85]	[0.37, 0.65]	$[\cdots,\ldots]$	[0.02, 0.43]
$\zeta_3^{[0]}$	[0.82, 0.52]	[0.13, 0.79]	[0.66, 0.48]	[0.36, 0.35]	$[\cdots,\ldots]$	[0.02, 0.57]
$\vdots$	$[\vdots,\vdots]$	$[\vdots,\vdots]$	$[\vdots,\vdots]$	$[\vdots,\vdots]$	$[\cdots,\ldots]$	$[\vdots,\vdots]$

$$\Lambda_{29}^{[*]} = \left({\hbar_2}^3 + \frac{(2^\sigma)^4 \left(\Gamma\left(\sigma+1/2\right)\right)^2 {\hbar_2}^3}{\left(\Gamma\left(\sigma\right)\right)^2 \sigma^2 \pi} - 2\,\frac{(2^\sigma)^2\,\Gamma\left(\sigma+1/2\right){\hbar_2}^3}{\Gamma\left(\sigma\right)\sigma\,\sqrt{\pi}} \right),$$

$$\Lambda_{30}^{[*]} = \begin{pmatrix} 4\,\frac{{\hbar_2}^4 (2^\sigma)^4 (\Gamma(\sigma+1/2))^2}{(\Gamma(\sigma))^2 \sigma^2 \pi} - 2\,\frac{{\hbar_2}^4 (2^\sigma)^2 \Gamma(\sigma+1/2)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} \\ -2\,\frac{{\hbar_2}^4 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ +\frac{{\hbar_2}^2 \hbar_3 (3^\sigma)^3 \sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{(\Gamma(\sigma))^2 \sigma^2 \pi} + 2\,{\hbar_2}^2 \hbar_3 \\ -2\,\frac{(2^\sigma)^2 \Gamma(\sigma+1/2){\hbar_2}^2 \hbar_3}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} - \frac{{\hbar_2}^2 \hbar_3 (3^\sigma)^3 \sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}(2^\sigma)^2 \Gamma(\sigma+1/2)} \end{pmatrix},$$

$$\Lambda_{31}^{[*]} = \begin{pmatrix} -5\,\frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^2 \sigma^2 \pi} - 3\,{\hbar_2}^3 \hbar_3 \\ +5\,\frac{(2^\sigma)^2 \Gamma(\sigma+1/2){\hbar_2}^3 \hbar_3}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} \\ +2\hbar_2 {\hbar_3}^2 - 8\,\frac{(2^\sigma)^6 (\Gamma(\sigma+1/2))^3 {\hbar_2}^5}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ +2\,\frac{(3^\sigma)^3 \sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3){\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^2 \sigma^2 \pi} + \\ 3/2\,\frac{(3^\sigma)^6 (\Gamma(\sigma+1/3))^2 (\Gamma(\sigma+2/3))^2 {\hbar_2}{\hbar_3}^2}{(\Gamma(\sigma))^2 \sigma^2 \pi\,(2^\sigma)^4 (\Gamma(\sigma+1/2))^2} \\ +2\,\frac{(2^\sigma)^8 (\Gamma(\sigma+1/2))^4 {\hbar_2}^5}{(\Gamma(\sigma))^4 \sigma^4 \pi^2} + 14\,\frac{(2^\sigma)^4 (\Gamma(\sigma+1/2))^2 {\hbar_2}^5}{(\Gamma(\sigma))^2 \sigma^2 \pi} + \\ 4{\hbar_2}^5 - 12\,\frac{(2^\sigma)^2 \Gamma(\sigma+1/2){\hbar_2}^5}{\Gamma(\sigma)\sigma\,\sqrt{\pi}} - \\ 2\,\frac{(2^\sigma)^2 \Gamma(\sigma+1/2)(3^\sigma)^3 \sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3){\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \\ -2\,\frac{{\hbar_2}{\hbar_3}^2 (3^\sigma)^3 \sqrt{3}\Gamma(\sigma+1/3)\Gamma(\sigma+2/3)}{\Gamma(\sigma)\sigma\,\sqrt{\pi}(2^\sigma)^2 \Gamma(\sigma+1/2)} + \frac{{\hbar_2}^4 (2^\sigma)^6 (\Gamma(\sigma+1/2))^3}{(\Gamma(\sigma))^3 \sigma^2 \pi^{3/2}} \\ -3\,\frac{{\hbar_2}^4 (2^\sigma)^4 (\Gamma(\sigma+1/2))^2}{(\Gamma(\sigma))^2 \sigma\,\pi} - \sigma {\hbar_2}^4 + \\ 3\,\frac{{\hbar_2}^4 (2^\sigma)^2 \Gamma(\sigma+1/2)}{\Gamma(\sigma)\sqrt{\pi}} + 3\,\frac{(2^\sigma)^6 (\Gamma(\sigma+1/2))^3 {\hbar_2}^3 \hbar_3}{(\Gamma(\sigma))^3 \sigma^3 \pi^{3/2}} \end{pmatrix},$$

Appendix B

Table B.1 lists the randomly generated initial guesses used in the numerical experiments for Nonlinear Engineering Application 1.
Table B.2 lists the randomly generated initial guesses used in the numerical experiments for Nonlinear Engineering Application 2.

Appendix C

To determine the real and complex roots of the polynomial equations in Engineering Applications 1 and 2, the artificial neural network (ANN) is trained on the output dataset using the CAS-MATLAB® 2012b environment (see Table C.1).

Data availability

No data was used for the research described in the article.

References

[1] Udegbe FC, Nwankwo EI, Igwama GT, Olaboye JA. Real-time data integration in diagnostic devices for predictive modeling of infectious disease outbreaks. Comput Sci IT Res J 2023;4(3).
[2] Ali MA, Elsayed A, Elkabani I, Akrami M, Youssef ME, Hassan GE. Optimizing artificial neural networks for the accurate prediction of global solar radiation: A performance comparison with conventional methods. Energies 2023;16(17):6165.
[3] Udristioiu MT, Mghouchi YE, Yildizhan H. Prediction, modelling, and forecasting of PM and AQI using hybrid machine learning. J Clean Prod 2023;421:138496.
[4] Orlov V, Gasanov M. The maximum domain for an analytical approximate solution to a nonlinear differential equation in the neighborhood of a moving singular point. Axioms 2023;12(9):844.
[5] Ismail GM, Kamel A, Alsarrani A. Approximate analytical solutions to nonlinear oscillations via semi-analytical method. Alex Eng J 2024;98:97–102.
[6] Hussain S, Haq F, Shah A, Abduvalieva D, Shokri A. Comparison of approximate analytical and numerical solutions of the Allen Cahn equation. Int J Differ Equ 2024;2024(1):8835138.
[7] Bakre OF, Wusu AS, Akanbi MA. An explicit single-step nonlinear numerical method for first order initial value problems (IVPs). J Appl Math Phys 2020;8(09):1729.
[8] Ait-Ameur K, Maday Y. Multi-step variant of the parareal algorithm: convergence analysis and numerics. ESAIM Math Model Numer Anal 2024;58(2):673–94.
[9] Temirbekov N, Imangaliyev Y, Baigereyev D, Temirbekova L, Nurmangaliyeva M. Numerical simulation of inverse geochemistry problems by regularizing algorithms. Cogent Eng 2022;9(1):2003522.
[10] Kelley CT. Solving Nonlinear Equations with Newton's Method. Society for Industrial and Applied Mathematics.; 2003.
[11] Barrada M, Benkhouya R, Chana I. A new Halley's family of third-order methods for solving nonlinear equations. IAENG Int J of Appl Math 2020;50(1).
[12] Argyros IK, George S, Argyros CI. On the Ostrowski method for solving equations. Eur J Math Anal 2022;2. 3-3.
[13] Akram S, Khalid M, Junjua MUD, Altaf S, Kumar S. Extension of King's iterative scheme by means of memory for nonlinear equations. Symmetry 2023;15(5):1116.
[14] Ogbereyivwe O, Atajeromavwo EJ, Umar SS. Jarratt and Jarratt-variant families of iterative schemes for scalar and system of nonlinear equations. Iran J Numer Anal Optim 2024;14(2):391–416.
[15] Wang X, Liu L. Two new families of sixth-order methods for solving non-linear equations. Appl Math Comput 2009;213(1):73–8.
[16] Parhi SK, Gupta DK. A sixth order method for nonlinear equations. Appl Math Comput 2008;203(1):50–5.
[17] Sivakumar P, Madhu K, Jayaraman J. Optimal fourth order methods with its multi-step version for nonlinear equation and their basins of attraction. SeMA J 2019;76:559–79.
[18] Ortega JM, Rheinboldt WC. Iterative Solution of Nonlinear Equations in Several Variables. Society for Industrial and Applied Mathematics; 2000.
[19] Deuflhard P. A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. Numer Math 1974;22(4):289–315.
[20] Torres-Hernandez A F, Brambila-Paz A F, Ursula Iturrarán-Viveros F, Reyna Caballero-Cruz. Fractional Newton–Raphson method accelerated with Aitken's method. Axioms 10(2):2021.
[21] Akgül A, Cordero A, Torregrosa JR. A fractional Newton method with 2σth-order of convergence and its stability. Appl Math Lett 98(2019):344–51.
[22] Cajori F. Historical note on the Newton–Raphson method of approximation. Am Math Mon 1911;18(2):29–32.
[23] Kumar P, Agrawal OP. An approximate method for numerical solution of fractional differential equations. Signal Process 2006;86(10):2602–10.
[24] Çelik E, Li Y, Telyakovskii AS. On the fractional Newton method with Caputo derivatives. ТРУД ИНСТИТУТА МАТЕМАТИКИ И МЕХАНИКИ УРО РАН 2022;28(4):273–6.
[25] Shams M, Carpentieri B. Efficient families of higher-order Caputo-type numerical schemes for solving fractional order differential equations. Alex Eng J 2025;124:337–61.
[26] Torres-Hernandez A, Brambila-Paz F, Iturrarán-Viveros U, Caballero-Cruz R. Fractional Newton–Raphson method accelerated with Aitken's method. Axioms 2021;10(2):47.
[27] Candelario G, Cordero A, Torregrosa JR, Vassileva MP. Generalized conformable fractional Newton-type method for solving nonlinear systems. Numer Algorithms 2023;93(3):1171–208.
[28] Gdawiec K, Kotarski W, Lisowska A. Newton's method with fractional derivatives and various iteration processes via visual analysis. Numer Algorithms 2021;86(3):953–1010.
[29] Reinke B, Schleicher D, Stoll M. The Weierstrass–Durand–Kerner root finder is not generally convergent. Math Comp 2023;92(340):839–66.
[30] Petković I, Herceg Đ. Computers in mathematical research: the study of three-point root-finding methods. Numer Algorithms 2020;84:1179–98.
[31] Lopes LG, Machado RN. A new ehrlich-type sixth-order simultaneous method for polynomial complex zeros. VETOR- Rev Ciências Exatas Eng 2023;33(2):52–9.
[32] Chinesta F, Cordero A, Garrido N, Torregrosa JR, Triguero-Navarro P. Simultaneous roots for vectorial problems. Comput Appl Math 2023;42(5):227.
[33] Bhalla S, Panwar M, Behl R, Chun C. Simultaneous root approximation: A high-convergence iterative approach. Math Methods Appl Sci 2025.
[34] Shams M, Kausar N, Samaniego C, Agarwal P, Ahmed SF, Momani S. On efficient fractional Caputo-type simultaneous scheme for finding all roots of polynomial equations with biomedical engineering applications. Fractals 2023;31(04):2340075.
[35] Cordero A, Torregrosa JR, Triguero-Navarro P. Jacobian-free vectorial iterative scheme to find simple several solutions simultaneously. Math Methods Appl Sci 2025.
[36] Shams M, Carpentieri B. An efficient and stable Caputo-type inverse fractional parallel scheme for solving nonlinear equations. Axioms 2024;13(10):671.
[37] Proinov PD, Petkova MD. On the convergence of a new family of multi-point Ehrlich-type iterative methods for polynomial zeros. Mathematics 2021;9(14):1640.
[38] Ivanov SI. Families of high-order simultaneous methods with several corrections. Numer Algorithms 2024;97(2):945–58.
[39] Nayak SK, Parida PK. Global convergence analysis of Caputo fractional whittaker method with real world applications. Cubo (Temuco) 2024;26(1):167–90.
[40] Mathai AM, Haubold HJ. Fractional and multivariable calculus. In: Model Building and Optimization Problems. Springer Optimization and Its Applications, vol. 122, Berlin, Germany: Springer; 2017.
[41] Atanackovic TM, Pilipovic S, Stankovic B, Zorica D. Fractional calculus with applications in mechanics: Wave propagation. In: Impact and Variational Principles. London, UK: Wiley; 2014.
[42] Odibat ZM, Shawagfeh NT. Generalized Taylor's formula. Appl Math Comput 2007;186:286–93.
[43] Ā-zban AY. Some new variants of Newton's method. Appl Math Lett 2004;17(6):677–82.
[44] Kou J, Li Y. A family of new Newton-like methods. Appl Math Comput 2007;192(1):162–7.
[45] Noor MA, Noor KI, Khan WA, Ahmad F. On iterative methods for nonlinear equations. Appl Math Comput 2006;183(1):128–33.
[46] Chicharro FI, Cordero A, Garrido N, Torregrosa JR. Generating root-finder iterative methods of second order: convergence and stability. Axioms 2019;8(2):55.
[47] Chun C. A family of composite fourth-order iterative methods for solving nonlinear equations. Appl Math Comput 2007;187(2):951–6.
[48] Kou J, Li Y, Wang X. A composite fourth-order iterative method for solving non-linear equations. Appl Math Comput 2007;184(2):471–5.
[49] Amat S, Busquier S, editors. In: Advances in iterative methods for nonlinear equations, Vol. 10, Cham, Switzerland; 2016.
[50] Basto M, Semiao V, Calheiros FL. A new iterative method to compute nonlinear equations. Appl Math Comput 2006;173(1):468–83.
[51] Sherman AH. On Newton-iterative methods for the solution of systems of nonlinear equations. SIAM J Numer Anal 1978;15(4):755–71.
[52] Candelario G, Cordero A, Torregrosa JR. Multipoint fractional iterative methods with $(2\alpha + 1)$ th-order of convergence for solving nonlinear problems. Mathematics 2020;8(3):452.
[53] Shams M, Kausar N, Agarwal P, Jain S, Salman MA, Shah MA. On family of the Caputo-type fractional numerical scheme for solving polynomial equations. Appl Math Sci Eng 2023;31(1):2181959.
[54] Reinke B, Schleicher D, Stoll M. The Weierstrass–Durand–Kerner root finder is not generally convergent. Math Comp 2023;92(340):839–66.
[55] Gemignani L, Noferini V. The Ehrlich–Aberth method for palindromic matrix polynomials represented in the Dickson basis. Linear Algebra Appl 2013;438(4):1645–66.
[56] Petković MS, Petković LD, Džunić J. On an efficient method for the simultaneous approximation of polynomial multiple roots. Appl Anal Discrete Math 2014;7:3–94.
[57] Shams M, Carpentieri B. An efficient and stable Caputo-type inverse fractional parallel scheme for solving nonlinear equations. Axioms 2024;13(10):671.
[58] Shams M, Carpentieri B. On highly efficient fractional numerical method for solving nonlinear engineering models. Mathematics 2023;11(24):4914.
[59] Cordero Barbero A, Ledesma A, Maimó JG, Torregrosa Sánchez JR. Design and dynamical behavior of a fourth order family of iterative methods for solving nonlinear equations. AIMS Math 2024;9(4):8564–93.
[60] Candelario G, Cordero A, Torregrosa JR, Vassileva MP. Solving nonlinear transcendental equations by iterative methods with conformable derivatives: A general approach. Mathematics 2023;11(11):2568.
[61] Cordero Alicia, Villalba Eva G, Torregrosa Juan R, Paula Triguero-Navarro. Convergence and stability of a parametric class of iterative schemes for solving nonlinear systems. Mathematics 2021;9(1):86.

[62] Shen C, Wang L, Li Q. Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method. J Mater Process Technol 2007;183(2–3):412–8.
[63] Petković MS, Petković LD, Džunić J. On an efficient simultaneous method for finding polynomial zeros. Appl Math Lett 2014;28:60–5.
[64] Shams M, Rafiq N, Kausar N, Agarwal P, Park C, Mir NA. On highly efficient derivative-free family of numerical methods for solving polynomial equation simultaneously. Adv Difference Equ 2021;2021:1–10.
[65] Shams M, Rafiq N, Kausar N, Agarwal P, Park C, Momani S. Efficient iterative methods for finding simultaneously all the multiple roots of polynomial equation. Adv Difference Equ 2021;2021:1–11.
[66] Rafiq N, Akram S, Mir NA, Shams M. Study of dynamical behavior and stability of iterative methods for non-linear equations with application in engineering. Math Probl Eng 2020:20.
[67] El-Ajou A, Arqub OA, Momani S. Solving fractional two-point boundary value problems using continuous analytic method. Ain Shams Eng J 2013;4(3):539–47.
[68] Steven C, Chapra. Applied Numerical Methods with MATLAB for Engineers and Scientists. Sixth ed.. New York, NY, USA: McGraw-Hill; 2010.
[69] Sirisubtawee S, Kaewta S. New modified adomian decomposition recursion schemes for solving certain types of nonlinear fractional two-point boundary value problems. Int J Math Math Sci 2017;2017.
[70] Jassim MH, Al-Hayani WM. Combining Laplace transform and variational iteration method for solving singular IVPs and BVPs of lane–Emden type equation. Wasit J Comput Math Sci 2024;3(2):8–18.
[71] Ervin VJ, Roop JP. Variational solution of fractional advection dispersion equations on bounded domains in d, Numer. Methods Partial Differ Equ 2007;23(2):256–81.
[72] Chu Y, Rafiq N, Shams M, Akram S, Mir NA, Kalsoom H. Computer methodologies for the comparison of some efficient derivative free simultaneous iterative methods for finding roots of nonlinear equations. CMC- Comput Mater Contin 2020;66(1):275–90.
[73] Hu Y, Luo Y, Lu Z. Analytical solution of the linear fractional differential equation by Adomian decomposition method. J Comput Appl Math 2008;215(1):220–9.
[74] Mohammed PO, Machado JAT, Guirao JL, Agarwal RP. Adomian decomposition and fractional power series solution of a class of nonlinear fractional differential equations. Mathematics 2021;9(9):1070.
[75] Cordero A, Garrido N, Torregrosa JR, Triguero-Navarro P. An iterative scheme to obtain multiple solutions simultaneously. Appl Math Lett 2023;145:108738.
[76] Bo P, Mai X, Meng W, Zhang C. Improving geometric iterative approximation methods using local approximations. Comput Graph 2023.