mathematics

MDPI

Article

Chaos in Inverse Parallel Schemes for Solving Nonlinear Engineering Models

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

[1] Faculty of Engineering, Free University of Bozen-Bolzano, 39100 Bolzano, Italy; mudassir4shams@gmail.com or mudassir.shams@unibz.it
[2] Department of Mathematics, Balıkesir University, 10145 Balıkesir, Turkey
[3] Department of Mathematics and Statistics, Riphah International University I-14, Islamabad 44000, Pakistan
* Correspondence: bruno.carpentieri@unibz.it

Abstract: Nonlinear equations are essential in research and engineering because they simulate complicated processes such as fluid dynamics, chemical reactions, and population growth. The development of advanced methods to address them becomes essential for scientific and applied research enhancements, as their resolution influences innovations by aiding in the proper prediction or optimization of the system. In this research, we develop a novel biparametric family of inverse parallel techniques designed to improve stability and accelerate convergence in parallel iterative algorithm. Bifurcation and chaos theory were used to find the best parameter regions that increase the parallel method's effectiveness and stability. Our newly developed biparametric family of parallel techniques is more computationally efficient than current approaches, as evidenced by significant reductions in the number of iterations and basic operations each iterations step for solving nonlinear equations. Engineering applications examined with rough beginning data demonstrate high accuracy and superior convergence compared to existing classical parallel schemes. Analysis of global convergence further shows that the proposed methods outperform current methods in terms of error control, computational time, percentage convergence, number of basic operations per iteration, and computational order. These findings indicate broad usage potential in engineering and scientific computation.

Keywords: parallel schemes; bifurcation; chaos; global convergence

MSC: 65H04; 65H05; 65H10; 65H17

check for updates

Academic Editor: Ricardo Lopez-Ruiz

Received: 3 November 2024
Revised: 14 December 2024
Accepted: 25 December 2024
Published: 27 December 2024

Citation: Shams, M.; Carpentieri, B. Chaos in Inverse Parallel Schemes for Solving Nonlinear Engineering Models. *Mathematics* **2025**, *13*, 67. https://doi.org/10.3390/math13010067

1. Introduction

Nonlinear equations have been essential throughout history, especially in the prehistoric understanding of astronomy, architecture, and natural occurrences [1,2]. Understanding proportional curves and stability requires basic nonlinear problem-solving skills, which were implicitly used by early civilizations such as the Egyptians and Babylonians [3] while building pyramids and ziggurat. Greek geometry advancements [4] led to the employment of nonlinear concepts in architecture and mechanics, while ancient astronomers [5] utilized them to study celestial bodies in order to anticipate eclipses and planetary movements, which is a nonlinear undertaking due to complicated orbital patterns and interactions.

Nowadays, scientific and engineering domains rely heavily on nonlinear equations

$$f(x) = 0, \tag{1}$$

which are used in population biology, fluid dynamics, electronics, and climate modeling [6,7]. They play a crucial role in the modeling of electrical circuits with nonlinear components such as diodes and transistors [8], the design of robust structures under varied loads [9], and the comprehension of chaotic weather patterns [10]. The analysis of complex systems also relies heavily on nonlinear dynamics, such as the interactions in ecological or economic models where minor adjustments can have disproportionately big effects, a phenomenon which is frequently seen in chaotic systems.

Analytical and exact approaches [11–13] for solving nonlinear equations are frequently inadequate and ineffective when used to tackle challenging real-world issues. Although these methods are effective for linear or specifically structured equations, nonlinear equations usually have high degrees of complexity, chaotic behavior, or interconnected variables that make explicit solutions all but impossible. In order to solve nonlinear equations analytically, for example, certain forms and simplifications must be used, which may not accurately reflect the system's dynamics and lead to less accurate approximations. Moreover, it might be tricky to obtain a single general solution utilizing exact methods, as nonlinear equations often produce multiple solutions or show sensitivity to initial conditions [14]. This rigidity significantly restricts the use of conventional analytical methods in dynamic domains such as quantum mechanics, structural engineering, and meteorology.

Niels Henrik Abel made a significant contribution in 1824 with the Abel Impossibility Theorem [15], which demonstrated that conventional algebraic approaches will never be sufficient for certain nonlinear equations. In consequence, research on the development of numerical schemes to approximate solutions was initiated, bringing forth new methods for solving nonlinear equations. As a result, Isaac Newton and Joseph Raphson devised Newton's numerical technique [16], which was widely used in the nineteenth and twentieth centuries and became a cornerstone for approximating the roots of nonlinear functions. Many studies were then conducted to develop single-step, two-step, and multi-step algorithms with and without parameters for solving nonlinear equations; see, e.g., [17–20] and references therein.

Although these single root-finding methods iteratively approached the exact solutions using tangent lines, they had a number of significant drawbacks:

- These methods have local convergence behavior, and require close values for the initial guess.
- Single root-finding methods are very sensitive to complex roots or equations.
- Equations with closely spaced roots can easily diverge or settle on an incorrect root.
- Single root-finding algorithms frequently diverge when the derivative of the functions approaches zero.
- These methods require computing the first or higher derivative at each iteration, which can be computationally costly, increasing overall time complexity.
- In certain circumstances, particularly when starting values are poorly chosen or functions have complex shapes, the approach might enter cycles (bifurcation or chaos [21]) in which it oscillates between points without reaching a solution, resulting in waste of both computational resources and time.

To overcome these concerns, more complex strategies such as parallel techniques [22] have been developed, which use several processors on a computer to find all the roots of (1). These parallel algorithms provide more stability and are less susceptible to initial guesses [23], making them especially valuable for complex nonlinear problems.

The Weierstrass–Durand–Kerner [24] algorithm is an iterative approach for solving nonlinear equations by simultaneously improving all roots. Given enough initial guesses, the technique becomes efficient for nonlinear equations with distinct roots due to its simplicity and quadratic convergence:

$$y_i^{[u]} = x_i^{[u]} - \wp\left(x_i^{[u]}\right) \tag{2}$$

where $\wp\left(x_i^{[u]}\right)$ is Weierstrass' correction, provided as follows:

$$\wp\left(x_i^{[u]}\right) = \frac{f\left(x_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]} - x_j^{[u]}\right)}, \quad (i, j = 1, \ldots, n). \tag{3}$$

The parallel scheme in (2) offers local quadratic convergence for sequentially finding all roots of (1) at a time. The inverse Weierstrass method, which is a modified version of (1), was developed by Nedzibov et al. [25], and is written as follows:

$$y_i^{[u]} = \frac{\left(x_i^{[u]}\right)^2 \prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]} - x_j^{[u]}\right)}{x_i^{[u]} \prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]} - x_j^{[u]}\right) + f\left(x_i^{[u]}\right)}. \tag{4}$$

This is also known as the inverse Weierstrass method, which has quadratic convergence.

$$\overline{y}_i^{[u]} = x_i^{[u]} - \frac{f\left(x_i^{[u]}\right)\left(1 + f\left(x_i^{[u]}\right)\right)}{\left(1 + \left(1 - \alpha f\left(x_i^{[u]}\right)\right)\right) + \frac{f\left(x_i^{[u]}\right)}{x_i^{[u]}}\left(1 + f\left(x_i^{[u]}\right)\right)}. \tag{5}$$

The inverse parallel scheme shown above (5) has local quadratic convergence.

In 1977, Ehrlich [26] introduced a convergent simultaneous method of the third order:

$$y_i^{[u]} = x_i^{[u]} - \frac{1}{\frac{1}{N_i\left(x_i^{[u]}\right)} - \sum\limits_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{1}{\left(x_i^{[u]} - x_j^{[u]}\right)}\right)} \tag{6}$$

where $N_i(x_i^{[u]}) = \frac{g\left(x_j^{[u]}\right)}{g'\left(x_j^{[u]}\right)}$.

Shams et al. [27] presented the fractional parallel approach with a convergence order of $\psi + 1$ for solving (1):

$$x_i^{[u+1]} = \frac{\left(x_i^{[u]}\right)^2 \prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]} - v_j^{[u]}\right)}{\left(x_i^{[u]}\right) \prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]} - v_j^{[u]}\right) + f\left(x_i^{[u]}\right)} \tag{7}$$

where $v_j^{[u]} = x_j^{[u]} - \left(\Gamma(\psi + 1)\frac{f(x_j^{[u]})}{\left[{}^{\check{c}}\ni^{\psi}_{\psi_1}\right] f(x_j^{[u]})}\right)^{1/\psi}$. Further, we compare our newly developed method with the Nourien method [28] (NSM$^{[4]}$), which has a convergence order of four:

$$x_i^{[u+1]} = x_i^{[u]} - \frac{\wp\left(x_i^{[u]}\right)}{1 + \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{\wp\left(x_j^{[u]}\right)}{\left(x_i^{[u]} - \wp\left(x_j^{[u]}\right) - x_j^{[u]}\right)} \right)}. \tag{8}$$

Motivated by the aforementioned method, the main aim of this study, which relies on the strengths of the earlier approaches, is to develop an efficient parallel scheme for simultaneously obtaining all roots of (1). Using insights from bifurcation and chaos theory, we can carefully avoid parameter regions within the scheme that are inclined toward bifurcation and chaotic behavior.

The main contribution of this study are:

- We develop new efficient parallel scheme families for finding all solutions to (1) via parallel processing.
- We investigate the local convergence of the proposed biparametric families of an inverse parallel scheme.
- We use the notion of bifurcation and chaos to avoid the trap of regions having periodic cycling and reduce the chance of divergence or slow convergence.
- We analyze the percentage computing efficiency of parallel schemes in comparison to classical methods.
- We assess the global convergence behavior with random initial values.

To the best of our knowledge, this contribution is novel. A thorough literature survey revealed that little work exists on construction, analysis, and chaos in inverse parallel schemes that are designed to find all roots of Equation (1). The rest of this study is structured as follows. First, in Section 2, we construct and analyze biparametric families of parallel schemes and determine their local convergence. In order to prevent the zone of instability and periodic doubling, tripling, and other issues, Section 3 discusses choices and bifurcations. The percentage computational efficiency is developed in Section 4. Section 5 deals with numerical results from some engineering applications. The study is concluded in the last section.

2. Development and Analysis of Inverse Parallel Scheme

Inverse parallel algorithms have made major contributions to solving nonlinear equations by allowing for simultaneous root-finding. In contrast to sequential approaches, these schemes update each estimate continuously and iteratively on a parallel processor, tackling all roots as interdependent. The above procedure reduces the sensitivity to initial guesses while simultaneously increasing convergence speed, which is especially valuable for nonlinear equations that have multiple roots. These techniques are perfect for sophisticated and high-degree complex nonlinear problems, as they minimize computing time by utilizing parallel processing. Their parallel nature also improves robustness, enabling the method to have a high percentage of computational efficiency and preserve stability in systems that are prone to chaotic or cyclic behavior. Considering the following single root-finding method [29]:

$$z^{[u]} = y^{[u]} - \frac{4f'\left(y^{[u]}\right)}{f'\left(x^{[u]}\right) - 3f'\left(y^{[u]}\right)} \frac{f\left(y^{[u]}\right)}{f'\left(y^{[u]}\right) + f'\left(x^{[u]}\right)}, \tag{9}$$

where $y^{[u]} = x^{[u]} - \frac{f\left(x^{[u]}\right)}{f'\left(x^{[u]}\right)}$, replacing $x_j^{[u]}$ by $V_j^{[u]}$ in (3), we have

$$\wp_i^{[*]}\left(x_i^{[u]}\right) = \frac{f\left(x_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]} - V_j^{[u]}\right)}, \tag{10}$$

where $V_j^{[u]} = x_j^{[u]} - \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)}\left[\frac{1}{1+\vartheta_1^{[*]}\left(\frac{f\left(x_j^{[u]}\right)}{1+\vartheta_2^{[*]}\left(f\left(x_j^{[u]}\right)\right)^2}\right)}\right]$ is a single-step biparametric ($\vartheta_1^{[*]}, \vartheta_1^{[*]} \in \mathbb{R}$) family of iterative methods ($\mathrm{M}^{[*]}$) of convergence order 2 proposed by Shams et al. [30]. Using (10) and Weierstrass' correction in (9), we develop a new biparametric family of inverse parallel schemes ($\mathrm{BM}^{[*]}$) to locate all roots of (1) simultaneously, as follows:

$$z_i^{[u]} = y_i^{[u]} - \frac{\left(\frac{\left(4\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(\frac{f\left(y_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(y_i^{[u]}-y_j^{[u]}\right)}\right)}{\left(1-3\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(1+\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)}\right)}{1+\frac{\left(\frac{\left(4\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(\frac{f\left(y_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(y_i^{[u]}-y_j^{[u]}\right)}\right)}{\left(1-3\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(1+\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)}\right)}{y_i^{[u]}}} \tag{11}$$

where $y_i^{[u]} = x_i^{[u]} - \frac{\frac{f\left(x_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]}-V_j^{[u]}\right)}}{1+\frac{\frac{f\left(x_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\ j\neq i}}^{n}\left(x_i^{[u]}-V_j^{[u]}\right)}}{x_i^{[u]}}}$ and

$V_j^{[u]} = x_j^{[u]} - \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)}\left[\frac{1}{1+\vartheta_1^{[*]}\left(\frac{f\left(x_j^{[u]}\right)}{1+\vartheta_2^{[*]}\left(f\left(x_j^{[u]}\right)\right)^2}\right)}\right]$, $\vartheta_1^{[*]}, \vartheta_2^{[*]} \in \mathbb{R}$. The methods should also be written as follows:

$$z_i^{[u]} = y_i^{[u]} - \frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{1+\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}, \tag{12}$$

$$\text{where } \wp_i^{[**]}\left(y_i^{[u]}\right) = \left(\frac{\left(4\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(\frac{f\left(y_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(y_i^{[u]}-y_j^{[u]}\right)}\right)}{\left(1-3\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(1+\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)}\right),$$

$$\wp_i\left(y_i^{[u]}\right) = \left(\frac{f\left(y_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(y_i^{[u]}-y_j^{[u]}\right)}\right), y_i^{[u]} = x_i^{[u]} - \frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}},$$

$$\wp_i^{[*]}\left(x_i^{[u]}\right) = \left(\frac{f\left(x_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(x_i^{[u]}-V_j^{[u]}\right)}\right), \text{ and}$$

$$V_j^{[u]} = x_j^{[u]} - \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)}\left[\frac{1}{1+\vartheta_1^{[*]}\left(\frac{f\left(x_j^{[u]}\right)}{1+\vartheta_2^{[*]}\left(f\left(x_j^{[u]}\right)\right)^2}\right)}\right].$$

The local order of convergence of the biparametric families of inverse parallel schemes is determined by the subsequent theorem.

Theorem 1. *Let $\zeta_1, ..., \zeta_\sigma$ be simple roots of a nonlinear equation, and assume that the initial distinct estimates $x_1^{[0]}, ..., x_n^{[0]}$ are sufficiently close to the true roots; then, the BM$^{[*]}$ method achieves a convergence order of six.*

Proof. Let $\epsilon_i = x_i^{[u]} - \zeta_i, \epsilon_i' = y_i^{[u]} - \zeta_i$, and $\epsilon_i^{[*]} = z_i^{[u]} - \zeta_i$ represent the errors in $x_i^{[u]}$, $y_i^{[u]}$, and $u_i^{[u]}$, respectively. From the first step of BM$^{[*]}$, we have

$$y_i^{[u]} - \zeta_i = x_i^{[u]} - \zeta_i - \frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}}, \tag{13}$$

$$\epsilon_i' = \epsilon_i - \frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}} = \epsilon_i - \frac{\epsilon_i\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{x_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}}, \tag{14}$$

$$\epsilon_i' = \epsilon_i\left[1-\frac{\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{x_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}}\right] = \epsilon_i\left[\frac{1-\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{x_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}+\frac{\epsilon_i}{x_i^{[u]}}\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{x_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}}\right]. \tag{15}$$

Utilizing $\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{x_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}-1=\sum\limits_{k\neq j}^{n}\frac{-(\epsilon_i)^2}{x_i^{[u]}-V_j^{[u]}}\prod\limits_{j\neq i}^{k-n}\frac{V_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}$, we have

$$\epsilon_i' = \epsilon_i\left[\frac{\sum\limits_{k\neq j}^{n}\frac{-\left(\epsilon_j\right)^2}{x_i^{[u]}-V_j^{[u]}}\prod\limits_{j\neq i}^{k-n}\left(\frac{V_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}\right)+\frac{(\epsilon_i)^2}{x_i^{[u]}}\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{V_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}\right)}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}}\right]. \tag{16}$$

If $\left|\epsilon_j\right| = |\epsilon_i| = |\epsilon|$, then

$$\epsilon_i' = \epsilon_i.(\epsilon_i)^2\left[\frac{\sum\limits_{k\neq j}^{n}\frac{-1}{x_i^{[u]}-V_j^{[u]}}\prod\limits_{j\neq i}^{k-n}\left(\frac{V_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}\right)+\frac{1}{x_i^{[u]}}\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{V_i^{[u]}-\zeta_j^{[u]}}{x_i^{[u]}-V_j^{[u]}}\right)}{1+\frac{\wp_i^{[*]}\left(x_i^{[u]}\right)}{x_i^{[u]}}}\right] = O\left|\epsilon^3\right|. \tag{17}$$

In the second step, we obtain

$$z_i^{[u]}-\zeta_i = y_i^{[u]}-\zeta_i-\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{1+\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}, \tag{18}$$

where $\wp_i^{[**]}\left(y_i^{[u]}\right) = \left(\frac{\left(4\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(\frac{f\left(y_i^{[u]}\right)}{\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(y_i^{[u]}-y_j^{[u]}\right)}\right)}{\left(1-3\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\right)\left(1+\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)}\right)$. Now, considering $\prod\limits_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-y_j^{[u]}}{x_i^{[u]}-x_j^{[u]}}\right)\approx 1$, we have

$$\epsilon_i' = \epsilon_i'-\frac{\epsilon_i'\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}}{1+\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}, \tag{19}$$

$$z_i^{[u]}-\zeta_i = y_i^{[u]}-\zeta_i-\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{1+\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}, \tag{20}$$

$$\epsilon_i^{[*]} = \epsilon_i'\left[1-\frac{\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}}{1+\frac{\wp_i^{[*]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}\right] = \epsilon_i'\left[\frac{1-\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}--\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}+\frac{\epsilon_i}{y_i^{[u]}}\prod\limits_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}}{1+\frac{\wp_i^{[**]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}\right]. \tag{21}$$

Utilizing $\prod_{\substack{j=1\\j\neq i}}^{n}\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}-1=\sum_{k\neq j}^{n}\frac{-\left(\epsilon_j'\right)}{y_i^{[u]}-y_j^{[u]}}\prod_{j\neq i}^{k-n}\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}$, we have

$$\epsilon_i^{[*]}=\epsilon_i'\left[\frac{\sum_{k\neq j}^{n}\frac{-\left(\epsilon_j'\right)}{y_i^{[u]}-y_j^{[u]}}\prod_{j\neq i}^{k-n}\left(\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}\right)+\frac{\left(\epsilon_j'\right)}{y_i^{[u]}}\prod_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}\right)}{1+\frac{\wp_i^{[*]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}\right]. \tag{22}$$

If $\left|\epsilon_j'\right|=\left|\epsilon_i'\right|=\left|\epsilon'\right|$, then

$$\epsilon_i^{[*]}=\left(\epsilon_j'\right).\left(\epsilon_j'\right)\left[\frac{\sum_{k\neq j}^{n}\frac{-1}{y_i^{[u]}-y_j^{[u]}}\prod_{j\neq i}^{k-n}\left(\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}\right)+\frac{1}{\left(\epsilon_i'\right)}\prod_{\substack{j=1\\j\neq i}}^{n}\left(\frac{y_i^{[u]}-\zeta_j^{[u]}}{y_i^{[u]}-y_j^{[u]}}\right)}{1+\frac{\wp_i^{[*]}\left(y_i^{[u]}\right)}{y_i^{[u]}}}\right]=O\left|\left(\epsilon'\right)^2\right|, \tag{23}$$

$$\epsilon_i^{[*]}=O\left|\left(\epsilon'\right)^2\right|=O\left|\epsilon^6\right|. \tag{24}$$

This completes the proof. □

3. Bifurcation Analysis of the Parallel Scheme

Bifurcation and chaos are essential in science and engineering, as they demonstrate how even minor adjustments to parameters may significantly modify the behavior of a system, thereby affecting its predictability and stability. Knowing about bifurcations [31] aids in system design, helping to prevent undesirable transitions, while an understanding of chaos [32] is necessary for managing unexpected dynamics in domains such as fluid dynamics, electrical circuits, and weather forecasting. Both ideas have stimulated innovation in control strategies, allowing for improved models in scientific study and more robust engineering designs [33,34]. Bifurcation and chaos are important considerations in the selection of parameters for numerical iterative schemes for solving nonlinear equations, as they have a direct impact on the algorithm's stability, speed, and convergence behavior. The iterative technique $\mathrm{M}^{[*]}$ has two free parameters $\vartheta_1^{[*]}, \vartheta_2^{[*]}$ that affect its convergence. In this context, we can dissect the contributions of bifurcation and chaos to see how they affect the convergence of schemes that simultaneously find all of the solutions at once.

Convergence Stability and Stable Fixed Point: In the parallel scheme, stable fixed points are values of the parameter $\vartheta_1^{[*]}, \vartheta_2^{[*]}$ which ensure that the iterative scheme converges to an exact solution of (1) without oscillations or divergence. A stable fixed point (see Table 1, column 2) is critical for parallel algorithms, as it guarantees that the method will converge to the root as desired. Whether or not the fixed points are stable via parallel iteration using multiple processors to simulate all solutions of (1) at once depends on the choice of $\vartheta_1^{[*]}, \vartheta_2^{[*]}$. If $\vartheta_1^{[*]}, \vartheta_2^{[*]}$ is chosen in such a way that the fixed point becomes unstable, then the parallel iterative algorithm may fail to converge or may converge slowly, requiring additional CPU time and increasing the computational cost. Provided that the fixed point remains stable, the parallel algorithms can show stable fast convergence by carefully choosing the values of $\vartheta_1^{[*]}$, and $\vartheta_2^{[*]}$. This stability can improve the convergence rate by reducing the number of iterations required to achieve the desire accuracy with less iteration steps, enabling parallel schemes that are reliable, consistent, and robust.

Table 1. Analysis of the bifurcation of the iterative scheme M1 containing the region of stability and instability.

Figure	$\left[\vartheta_1^{[*]}, \vartheta_2^{[*]}, \vartheta\right]$	$\mathbf{S}_1^{[*]}$	$\mathbf{B}_1^{[*]}$	$\mathbf{B}_2^{[*]}$	$\mathbf{B}_{4-8}^{[*]}$	$\mathbf{B}_n^{[*]}$	$\mathbf{CSB}_2^{[*]}$
Figure 1a	[1.5, 1.5, 1]	[1, 1.5]	(1.5, 2.6]	No	[3.4, 3.6]	[2.7, 3.4]	[3.6, 5]
Figure 1b	[1.5, 1.5, 1.9]	[1, 1.9]	(1.9, 2.0]	(2, 2.1]	(2.4, 2.9)	[2.12, 2.4]	[3, 5]
Figure 1c	[1.5, 1.5, 1.5]	(1, 1.5]	(1.5, 2)	(2.1, 2.2)	(2.2, 2.41)	[2, 2.4] (2.4, 2.6) (2.6, 3.4)	(3.4, 5]
Figure 1d	[3.5, 3.5, 1]	[1, 2.2]	No	[3.9, 4]	[3.6, 4]	[2.21, 3.6]	[4, 5]
Figure 1e	[3.5, 3.5, 0.5]	[1, 2.4]	No	No	No	[2.4, 2.7]	[2.7, 5]
Figure 2a	[3, 3, 2.9]	[1, 2.1]	No	No	[2.2, 2.4]	[2.4, 3.5]	[3.1, 5]
Figure 2b	[3, 3, 1.1]	[1, 2.1]	No	No	[3.5, 3.9]	[2.1, 3.5]	[3.9, 5]
Figure 2c	[3, 3, 2.1]	[1, 2.2]	No	No	[2.2, 2.4]	[2.4, 3.1]	[3.1, 5]
Figure 2d	[2, 2, 1]	[1, 1.5] [1.6, 2]	[2.0, 2.6]	No	No	[1.6, 1.7] [2.0, 2.1] [2.6, 3.6]	(3.7, 5]
Figure 2e	[2, 2, 2.1]	(1, 1.9)	(2, 2.2)	No	No	(1.9, 2.0) (2.2, 2.9)	(2.9, 5]
Figure 2f	[2, 2, 3.1]	(1.5, 1.9)	(1.9, 2)	(2, 2.2)	No	No	[2.2, 2.5]
Figure 2g	[1, 1, 0.3]	[1, 1.6]	(1.6, 1.7)	(2.6, 2.8) (4.5, 4.51)	(2.8, 2.9)	(1.6, 1.7) (2.1, 2.6) (4.4, 4.5)	(4.51, 5]
Figure 2h	[1, 1, 0.4]	(1, 1.6) (3, 3.6)	(1.6, 1.61)	(2.5, 3)	(4.2, 4.5)	[1.6, 1.7] (2.0, 2.5) (4.0, 4.2)	(4.5, 5]
Figure 2i	[1, 1, 0.7]	(1, 1.6)	(1.6, 1.61) (2.4, 3.4)	(1.9, 2.4) [3.5, 4.0]	(1.61, 1.9)	No	(4.1, 5]

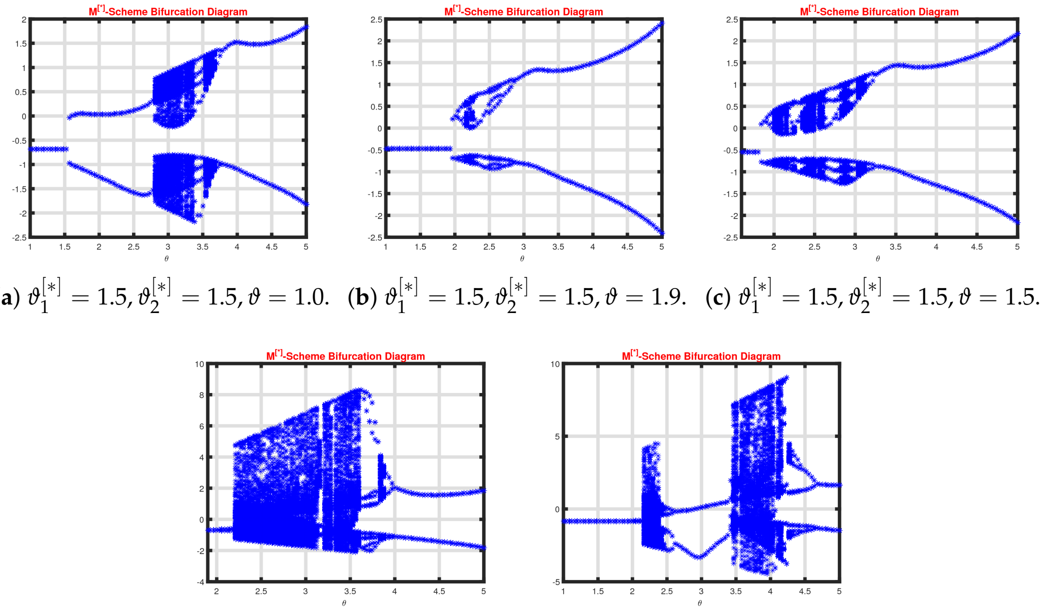
Parallel Schemes: Bifurcation and Transition to Periodic Behavior: When the parameters of parallel schemes change, the system may undergo bifurcation (see Table 1, column 3), with the previously stable fixed point separating into periodic cycles. This indicates that the parallel iteration values alternate between a predetermined number of points rather than converging to the exact solutions of (1). In parallel iterative schemes, periodic behavior shows that the algorithm is cycling between points rather than approaching the exact solution of (1). In order to avoid parameter ranges where bifurcations occur, it is possible to carefully select the parameter values so as to preserve a predictable and monotonic convergence towards the exact solution of (1). By identifying critical values of the parameter used in the parallel scheme where the system transitions from stable convergence to periodic cycles (see Table 1, column 3–4), bifurcation analysis can help to make for a more informed choice of parameters for sure convergence to the exact solutions that are required for accuracy.

Parallel Schemes' Chaos and Sensitivity to Initial Conditions: The iterative sequence will become very sensitive to initial conditions and exhibit unpredictable, non-converging behavior if the parallel computer algorithm enters a chaotic phase for certain parameter values. Chaos in parallel iterative algorithms is problematic because it implies that even a minor change to the initial guess can result in an entirely different outcome, making the process unpredictable and unreliable. Chaotic behavior affects the rate of convergence because it prevents the method from settling near to the root and instead produces an apparently random series of points. To keep the method in a stable or periodic setting which supports convergence, parameter selection should avoid values that produce chaos (see Table 1, column 6).

To examine bifurcation and chaos in the parallel iterative sequence, we can apply the following nonlinear function:

$$f(x) = x^3 + \vartheta x + 1. \tag{25}$$

In the parallel iterative scheme with damping parameter θ [35], we see distinct types of chaotic behavior for various parameter values $\vartheta_1^{[*]}, \vartheta_2^{[*]}$, as shown in Table 1 and Figures 1a–e and 2a–i . Figures 1a–e and 2a–i depicts how the iterative scheme progresses from stability to chaos as the parameters are changed. Initially, the system behaves consistently, with predictable and reliable convergence; however, if the parameters deviate, period-doubling bifurcations occur, causing the system to move towards more complex periodic orbits such as two- and four-period cycles. Eventually, the system enters a chaotic regime characterized by scattered and irregular behavior with no clear periodicity, highlighting its sensitivity to initial conditions and parameter changes. This chaotic behavior highlights the scheme's fluctuation in many places, where slight modifications might result in significantly distinct results. Despite this, transition zones demonstrate that the system can recover from chaos and restore stability, returning to predictable periodic orbits or fixed-point convergence when the parameters are carefully adjusted.



(**a**) $\vartheta_1^{[*]} = 1.5, \vartheta_2^{[*]} = 1.5, \vartheta = 1.0$. (**b**) $\vartheta_1^{[*]} = 1.5, \vartheta_2^{[*]} = 1.5, \vartheta = 1.9$. (**c**) $\vartheta_1^{[*]} = 1.5, \vartheta_2^{[*]} = 1.5, \vartheta = 1.5$.

(**d**) $\vartheta_1^{[*]} = 3.5, \vartheta_2^{[*]} = 3.5, \vartheta = 1.0$. (**e**) $\vartheta_1^{[*]} = 3.5, \vartheta_2^{[*]} = 3.5, \vartheta = 05$.

Figure 1. (**a**–**e**) Bifurcation and chaos in biparametric inverse parallel scheme for (25).

In Table 1, $\mathrm{S}_1^{[*]}$,$\mathrm{B}_1^{[*]} - \mathrm{B}_n^{[*]}$ and $\mathrm{CSB}_2^{[*]}$ represents the following:

$\mathrm{S}_1^{[*]}$: The system converges to a fixed point at these parameter values, represented by a single straight line. These values show that the approach is stable.

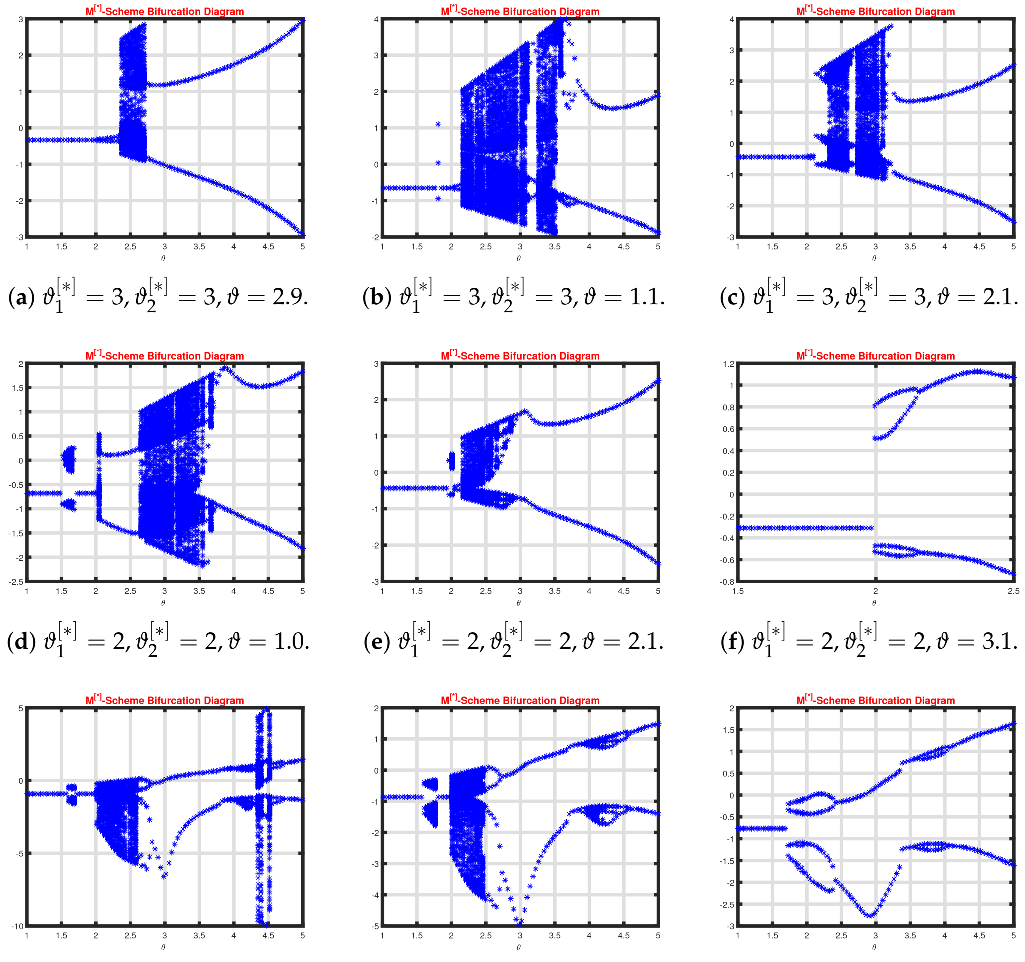
$\mathrm{B}_1^{[*]}$: The interval where a single line separates into many branches, representing the first bifurcation. This first split results in a two-period orbit in which the system oscillates between two values rather than resting on one.

$\mathrm{B}_2^{[*]}$: Intervals that suggest period-doubling bifurcations.

$\mathrm{B}_{4-8}^{[*]}$: The intervals during which the system transitions from two-period to four-period, eight-period, and so on, resulting in progressively more complex periodic orbits that lead to a dense region where chaos occurs.

$\mathrm{B}_n^{[*]}$: The dense cluster of points in this region with values that appear to be scattered and lack obvious periodicity implies chaotic behavior. This chaotic zone indicates that inverse families of parallel schemes are extremely sensitive, with even minor changes in parameters or initial starting values resulting in dramatically different results as the method diverges.

$\mathrm{CSB}_2^{[*]}$: These intervals indicate that the iterative sequence is transitioning from the chaotic regime to more predictable periodic orbits, and that it may eventually converge to stable behavior for the interval containing the parameter values (see Table 1, last column). The bifurcation analysis illustrates how sensitive the iterative technique is to parameter choice, exhibiting transitions from stable convergence to chaos. Stable regions $\mathrm{S}_1^{[*]}$ relate to fixed-point convergence, which has predictable and reliable behavior. As deviation in parameters takes place, a period-doubling bifurcation occurs ($\mathrm{B}_2^{[*]}$); then, higher orders of periodic orbits occur ($\mathrm{B}_{4-8}^{[*]}$) and chaos regimes take place ($\mathrm{B}_n^{[*]}$), which leads to irregular sensitivity to initial conditions. However, the $\mathrm{CSB}_2^{[*]}$ transition intervals demonstrate that the system can readily return from chaos into a predictable orbit if the parameters are controlled appropriately. These findings demonstrate the importance of tuning parameters for stability, preventing chaos, and ensuring efficient iterative convergence of the scheme.



(**a**) $\vartheta_1^{[*]} = 3, \vartheta_2^{[*]} = 3, \vartheta = 2.9$.
(**b**) $\vartheta_1^{[*]} = 3, \vartheta_2^{[*]} = 3, \vartheta = 1.1$.
(**c**) $\vartheta_1^{[*]} = 3, \vartheta_2^{[*]} = 3, \vartheta = 2.1$.

(**d**) $\vartheta_1^{[*]} = 2, \vartheta_2^{[*]} = 2, \vartheta = 1.0$.
(**e**) $\vartheta_1^{[*]} = 2, \vartheta_2^{[*]} = 2, \vartheta = 2.1$.
(**f**) $\vartheta_1^{[*]} = 2, \vartheta_2^{[*]} = 2, \vartheta = 3.1$.

(**g**) $\vartheta_1^{[*]} = 1, \vartheta_2^{[*]} = 1, \vartheta = 0.3$.
(**h**) $\vartheta_1^{[*]} = 1, \vartheta_2^{[*]} = 1, \vartheta = 0.4$.
(**i**) $\vartheta_1^{[*]} = 1, \vartheta_2^{[*]} = 1, \vartheta = 0.7$.

Figure 2. (**a**–**i**) Bifurcation and chaos in biparametric inverse parallel scheme for (25).

The choice of the parameters that the scheme utilizes influences how fast the error decreases with each iteration. Along with the stability of the fixed point, both of these factors affect the convergence rate of parallel iterative schemes. Delayed convergence can occur when choosing parameter values that are either too large or too small, as the correction term may not properly bring the current estimate closer to the exact solution of (1). The convergence rate can be increased by carefully adjusting the parameters to keep the parallel scheme within a stable range. This allows the correction term to effectively minimize the error in each iterative step, improving the stability, consistency, and robustness of the scheme. Numerical schemes that prevent bifurcation and chaos tend to have faster convergence, as the iterations are well-directed toward the exact solution of (1).

4. Computational Efficiency and Numerical Results

Percentage computational efficiency is essential in iterative approaches, as it measures how well computer resources are used to achieve convergence. This measure shows how quickly and effectively the algorithm produces accurate results by comparing the method's speed and resource consumption to the ideal or maximum possible performance. High computational efficiency indicates that the approach uses less memory, time, and computing power, all of which are critical for complicated or large-scale systems where these resources are scarce.

Through computational efficiency, we can find out more about:

- *Convergence Performance:* Efficiency provides information about how fast a method converges to a practical solution, which helps in choosing the right technique based on the desired convergence speed.
- *Optimization of Resources:* High efficiency indicates that the technique requires less memory and CPU, which is vital for large-scale or real-time engineering applications.
- *Flexibility:* Effective techniques are better suited to parallel computing, which reveals how well they tackle higher-degree complex problems.
- *Error Minimization:* This emphasizes how effectively the technique reduces error every repetitions, which is crucial in applications requiring accuracy.

Computational efficiency provides a comprehensive evaluation of the method's overall performance, guiding iterative approach selection based on resource requirements, speed, and accuracy. The efficiency index for the parallel iterative scheme (PS) $\pounds(PS, n)$ is provided as follows:

$$\pounds(PS, n) = \frac{\log[r]}{\Game} \tag{26}$$

where r is the convergence order of the parallel scheme and $\Game$ is the computational cost [36]. To evaluate the computing cost $\Game$, we utilize arithmetic operation per iteration with a specific weight based on the execution time of the operation. The weights used for division, multiplication, and addition plus subtraction are ϕ_{as}, ϕ_m, and ϕ_d, respectively. The notation AS_n, M_n, and D_n represents the number of divisions, multiplications, additions, and subtractions for each root for a given polynomial of degree n. The computation cost can be estimated as

$$\Game = \phi_{as} AS_n + \phi_m M_n + D_n \phi_d. \tag{27}$$

Thus,

$$(SB^{[*]}, \varphi) = \left(\frac{\pounds\left(\mathrm{SB}^{[*]}, n\right)}{\pounds(\varphi, n)} - 1 \right) \times 100. \tag{28}$$

Here, we compare the computational efficiency of our newly designed biparametric family to the previous technique.

Zhang et al. proposed the following method [37] (ZM$^{[*]}$):

$$x_i^{[u+1]} = x_i^{[u]} - \frac{w\left(x_i^{[u]}\right)}{1 + \Delta^{[*]}\left(x_i^{[u]}\right) + \sqrt{\begin{array}{c}\left(1 + \Delta^{[*]}\left(x_i^{[u]}\right)\right)^2 + \\ 4\wp\left(x_i^{[u]}\right) \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{\wp\left(x_j^{[u]}\right)}{\left(x_i^{[u]} - x_j^{[u]}\right)\left(x_i^{[u]} - \wp(x_i^{[u]}) - x_j^{[u]}\right)} \right)\end{array}}}, \tag{29}$$

where $\Delta^{[*]}\left(x_i^{[u]}\right) = \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{\wp\left(x_j^{[u]}\right)}{x_i^{[u]} - x_j^{[u]}} \right)$. Using $x_j^{[u]} = v_j^{[u]}$ as a correction in (4), Petkovic et al. [38] accelerated the convergence order from three to six (abbreviated as PM$^{[*]}$):

$$x_i^{[u+1]} = x_i^{[u]} - \frac{1}{\frac{1}{N_i(x_i^{[u]})} - \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{1}{\left(x_i^{[u]} - v_j^{[u]}\right)} \right)} \tag{30}$$

where $v_j^{[u]} = x_j^{[u]} - \frac{f\left(s_j^{[u]}\right) - f\left(x_j^{[u]}\right)}{2f\left(s_j^{[u]}\right) - f\left(x_j^{[u]}\right)} \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)}$, and $s_j^{[u]} = x_j^{[u]} - \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)}$. Shams et al. presented the following parallel schemes (SM$^{[*]}$) of sixth-order convergence [39]:

$$x_i^{[u+1]} = x_i^{[u]} - \frac{1}{\frac{1}{N_i(x_i^{[u]})} - \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{1}{\left(x_i^{[u]} - Z_j^{[u]}\right)} \right)} \tag{31}$$

where $Z_j^{[u]} = y_j^{[u]} - \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)} \left[\frac{\left(\frac{f\left(y_j^{[u]}\right)}{f\left(x_j^{[u]}\right)} \right)}{1 - 2\left(\frac{f\left(y_j^{[u]}\right)}{f\left(x_j^{[u]}\right)} \right) + \vartheta_1^{[*]} \left(\frac{f\left(y_j^{[u]}\right)}{f\left(x_j^{[u]}\right)} \right)^2} \right]$,

$y_j^{[u]} = x_j^{[u]} - \frac{f\left(x_j^{[u]}\right)}{f'\left(x_j^{[u]}\right)}$, and $\vartheta_1^{[*]} \in R$. Table 2 displays the number of basic arithmetic operations used by parallel algorithms in each iterative iteration to approximate all roots of (1) and (Table 3), illustrates the percentage computational efficiency.

Table 2. Operations utilized by parallel schemes.

Scheme	ZM$^{[*]}$	PM$^{[*]}$	SM$^{[*]}$	BM$^{[*]}$
$+, -$	$20\varnothing^{[**]}$	$29\varnothing^{[**]}$	$17\varnothing^{[**]}$	$15\varnothing^{[**]}$
$\times$	$18\varnothing^{[**]}$	$26\varnothing^{[**]}$	$21\varnothing^{[**]}$	$20\varnothing^{[**]}$

Here, $\varnothing^{[**]} = n^2 + O(n)$, where the number of division for all methods are the same, i.e., $2\varnothing^{[**]}$.

Table 3. Percentage computational efficiency.

Parallel Schemes	$\varphi = \text{ZM}^{[*]}$	$\varphi = \text{PM}^{[*]}$	$\varphi = \text{SM}^{[*]}$
(SB$^{[*]}$, φ)	55.65%	29.23%	20.76%

To examine the effectiveness and stability of our proposed method, several engineering applications are analyzed in this section. In our experiments, we used the following termination criteria:

$$(i) \qquad \epsilon_i^{[u]} = \left\| x_i^{[u+1]} - x_i^{[u]} \right\| < 10^{-32}, \tag{32}$$

where $\epsilon_i^{[u]}$ represents the norm-2 residual error.

4.1. Example 1: Beam Design Model—Mechanical Engineering Application

In mechanical engineering, the beam designing concept is essential to building structures that can effectively support loads and are failure-proof. Through the prediction of stress distribution, deflection, and probable failure spots under varied forces, this model aids engineers in the design of beams. Beams are crucial load-bearing components in machinery, building frameworks, automobile chassis, and bridge construction, among other applications for the model. The beam design model helps to minimize material usage and optimize design by mimicking real-world conditions and guaranteeing structural integrity and safety. Such models are crucial for increasing the overall performance of mechanical systems, reducing expenses, and boosting durability. These models also aid in adhering to industry norms and rules, resulting in engineering solutions that are dependable and robust. The model [40] uses the following nonlinear equations:

$$f(x) = x^4 + 4.0x^3 - 24.0x^2 + 16(x+1) \tag{33}$$

or

$$f(x) = (x-2)^2\left(x^2 + 8.0x + +4.0\right). \tag{34}$$

Using MatLab's built-in command, we obtained the following exact radical roots:

$$\zeta_{1,2} = 2, 2, \zeta_{3,4} = -4 \pm 2\sqrt{3}i. \tag{35}$$

Using the information in Table 1, we selected the optimal parameter values and obtained numerical results for extremely rough initial guess values of

$$[x_1^{[0]}, x_2^{[0]}, x_3^{[0]}, x_4^{[0]}] = [5, 4, -6+3i, -6-3i], \tag{36}$$

as shown in Table 4.

Table 4. Numerical results of parallel schemes for (31).

Method	$e_1^{[6]}$	$e_2^{[6]}$	$e_3^{[6]}$	$e_4^{[6]}$	CPU-time
ZM$^{[*]}$	4.1×10^{-25}	0.1×10^{-14}	8.1×10^{-33}	2.7×10^{-44}	3.2234534
PM$^{[*]}$	6.0×10^{-32}	0.5×10^{-32}	8.5×10^{-35}	6.0×10^{-27}	2.9283224
SM$^{[*]}$	3.4×10^{-34}	0.0	1.8×10^{-22}	3.5×10^{-24}	2.3453532
BM$^{[*]}$	2.1×10^{-54}	2.0×10^{-55}	2.4×10^{-51}	0.0	1.2345353

Table 4 clearly shows that our newly created methods are more efficient in terms of residual error and computing time in seconds compared to existing approaches. To test the global convergence of the parallel scheme, we can consider random initial starting values such as

$$
\begin{aligned}
x^{[0]} &= [x_1^{[0]}, x_2^{[0]}, x_3^{[0]}, x_4^{[0]}], \\
x_1^{[0]} &= [0.23, 0.32, 0.53, 0.54], \\
x_2^{[0]} &= [0.13, 0.21, 0.03, 0.34], \\
x_3^{[0]} &= [0.23, 0.32, 0.03, 0.34].
\end{aligned} \tag{37}
$$

The numerical outcomes using initial approximation (37) are presented in Table 5.

Table 5. Consistency analysis of the parallel schemes for (31).

Scheme	Max-Err	Max-It	$[\pm, \times, \div]$	P-Con	Local-COC
PM$^{[*]}$	0.1×10^{-08}	7	59.0	31.453%	4.4003224
NN$^{[*]}$	6.6×10^{-10}	5	46.0	55.657%	5.67008645
NF$^{[*]}$	8.9×10^{-11}	5	55.0	76.765%	5.45556455
SB$^{[*]}$	1.4×10^{-19}	4	24.0	98.766%	6.1546746

Table 5 demonstrates the better performance of our newly developed inverse family of parallel schemes SB$^{[*]}$ for random initial values in terms of maximum error (Max-Err), number of average iterations needed for convergence (Max-It), total number of basic arithmetic operations per iteration ($[\pm, \times, \div]$), percentage convergence (P-Con), and local computational order of convergence (Local-COC) compared to PM$^{[*]}$, NN$^{[*]}$, NF$^{[*]}$. The results show that our inverse parallel scheme SB$^{[*]}$ is more feasible, consistent, and efficient than the existing PM$^{[*]}$, NN$^{[*]}$, and NF$^{[*]}$ methods. For random iterations, the greatest residual error is shown in Figure 3.

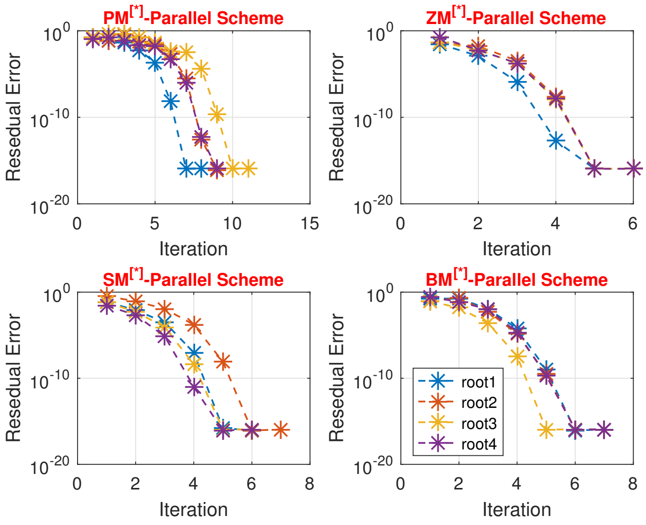


Figure 3. Error graph of the parallel scheme for solving (34) using random test vectors.

4.2. Example 2: Thermodynamics Mechanical Engineering Application [41]

Numerous businesses rely on thermodynamics, a mechanical engineering application, to design systems that transform energy between different forms. The devices used in these applications include compressors, engines, turbines, and refrigerators, all of which are critical components of manufacturing, HVAC systems, and power generation. In order to minimize energy losses and maximize performance, engineers can lower operating costs by assessing energy transformations and efficiencies. Understanding the characteristics of working fluids, heat transmission, and work exchanges is made possible by thermodynamic concepts, which also enable efficient system design. These insights are used by mechanical engineers to create technology that is more dependable, efficient, and ecologically friendly. The future of energy-efficient technologies will ultimately be shaped by these applications, which are propelling developments in industrial automation, transportation, and renewable energy. The following nonlinear equations are frequently modeled by thermodynamic systems:

$$\rho = 1.9520.0 \times 10^{-14}x^4 - 9.5838 \times 10^{-11}x^3 + 9.7215 \times 10^{-8}x^2 + 1.671 \times 10^{-4}x + 0.99403, \tag{38}$$

where ρ is the specific hear and For $\rho = 1.20$. This simplifies to

$$\begin{aligned} f(x) &= 1.9520.0 \times 10^{-14}x^4 - 9.5838 \times 10^{-11}x^3 + 9.7215 \times 10^{-8}x^2 \\ &\quad +1.671 \times 10^{-4}x + 0.99403.00. \end{aligned} \tag{39}$$

The nonlinear Equation (39) has the following exact roots accurate up to two decimal places:

$$\zeta_1 = 1126.009, \zeta_{2,3} = 2536.83 \pm 910.50i, \zeta_4 = -1289.95. \tag{40}$$

Using the information in Table 1, we selected the optimal parameter values and obtained the numerical results for extremely rough initial guess values, as shown in Table 6:

$$[x_1^{[0]}, x_{2,3}^{[0]}, x_4^{[0]}] = [1500, 2700 \pm 11i, -1500]. \tag{41}$$

The numerical outcomes using the initial approximation (39) are presented in Table 6.

Table 6. Numerical results of parallel schemes for (39).

Method	$e_1^{[6]}$	$e_2^{[6]}$	$e_3^{[6]}$	$e_4^{[6]}$	CPU-time
ZM[*]	5.7×10^{-15}	1.0×10^{-19}	9.1×10^{-23}	6.7×10^{-44}	4.546435
PM[*]	4.5×10^{-22}	0.1×10^{-32}	4.1×10^{-25}	9.9×10^{-27}	3.453646
SM[*]	1.1×10^{-24}	0.7×10^{-21}	6.0×10^{-22}	1.0×10^{-24}	2.342564
BM[*]	0.1×10^{-54}	0.0	0.0	0.1×10^{-64}	1.976578

Table 6 clearly shows that our newly created methods are more efficient in terms of residual error and computation time in seconds compared to the existing PM[*], NN[*], and NF[*] methods. To test the global convergence of the parallel approach, we can consider random initial starting values such as

$$\begin{aligned} x^{[0]} &= [x_1^{[0]}, x_2^{[0]}, x_3^{[0]}, x_4^{[0]}], \\ x_1^{[0]} &= [0.00, 0.32, 0.73, 0.84], \\ x_2^{[0]} &= [0.05, 0.11, 0.06, 0.38], \\ x_3^{[0]} &= [0.11, 0.42, 0.53, 0.09]. \end{aligned} \tag{42}$$

The numerical outcomes using initial approximation (42) are presented in Table 7.

Table 7. Consistency analysis of the parallel schemes for (39).

Scheme	Max-Err	Max-It	$[\pm, \times, \div]$	P-Con	Local-COC
PM[*]	6.1×10^{-10}	9	45	48.876%	4.4533444
NN[*]	8.7×10^{-12}	7	35	61.879%	5.67545645
NF[*]	0.1×10^{-11}	6	27	95.980%	5.87864545
SB[*]	1.2×10^{-17}	4	19	99.879%	6.04353546

Our newly developed inverse family of parallel schemes $\mathrm{BM}^{[*]}$ outperforms with respect to the maximum error, number of iterations needed for convergence, total number of basic arithmetic operations per iteration, percentage divergence, and local computational order of convergence for random initial values, as Table 7 demonstrates. Consequently, our parallel technique $\mathrm{BM}^{[*]}$ is more realistic, consistent, and efficient than the current $\mathrm{PM}^{[*]}$, $\mathrm{NN}^{[*]}$, and $\mathrm{NF}^{[*]}$ methods. Figure 4, illustrates the highest residual error on random iterations.

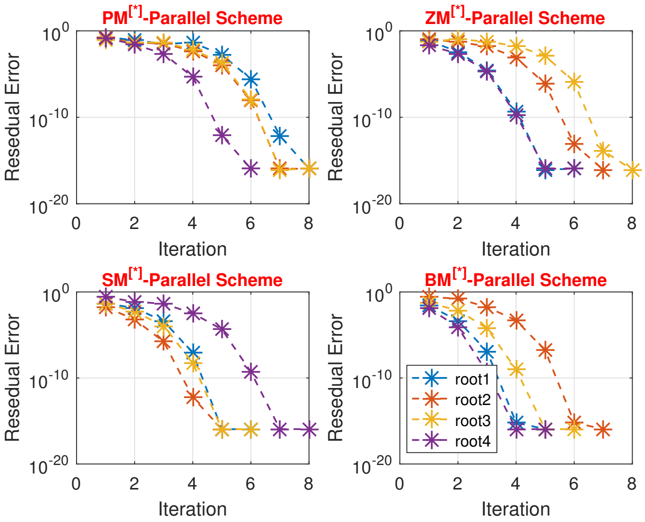


Figure 4. Error graph of the parallel scheme for solving (38) using random test vectors.

4.3. Example 3: Chinese Women's Osteoporosis

Chinese women are more likely to develop osteoporosis due to genetic, dietary, and lifestyle factors. When bone density decreases with age, Chinese women are more prone to fractures, which can lead to disability and a reduced quality of life. In this population, talking about osteoporosis is essential for raising awareness and promoting preventative measures, as early treatments such as diet, exercise, and medication can dramatically reduce risk. Addressing this issue adds to public health measures aimed at minimizing the burden of osteoporosis among China's aging population. The design of such osteoporosis measures can be mathematically modeled [42], leading to the following nonlinear equation [38]:

$$\Lambda^{[*]} = 0.0039x^3 - 0.78x^2 + 39.9x + 3383, \tag{43}$$

where $\Lambda^{[*]} = 3850$ provides the nonlinear equations

$$h(x) = 0.0039x^3 - 0.78x^2 + 39.9x - 467. \tag{44}$$

The exact roots up to one decimal place are obtained using the built-in "Solve[.]" command in Matlab, as follows:

$$\zeta_1 = 16.702300, \zeta_2 = 56.574000, \zeta_3 = 126.723500. \tag{45}$$

Using the information provided in Table 1, we selected the optimal parameter values and achieved the numerical results displayed in Table 8 for very rough initial guess values:

$$[x_1^{[0]}, x_2^{[0]}, x_3^{[0]}] = [20, 60, 130]. \tag{46}$$

The numerical outcomes using initial approximation (44) are presented in Table 8.

Table 8. Numerical results of parallel schemes for (44).

Method	$e_1^{[6]}$	$e_2^{[6]}$	$e_3^{[6]}$	CPU-time
ZM[*]	4.1×10^{-5}	5.1×10^{-14}	8.1×10^{-13}	2.343535
PM[*]	6.0×10^{-32}	1.1×10^{-22}	0.0	2.135034
SM[*]	0.0	0.0	6.0×10^{-42}	1.438593
BM[*]	0.0	0.0	2.0×10^{-51}	0.002133

Table 8 clearly shows that our newly created methods are more efficient in terms of residual error and computation time in seconds than the existing PM[*], NN[*], and NF[*] methods. To test the global convergence, we considered random initial starting values such as

$$\begin{aligned} x^{[0]} &= [x_1^{[0]}, x_2^{[0]}, x_3^{[0]}], \qquad (47) \\ x_1^{[0]} &= [0.17, 0.11, 0.53], \\ x_2^{[0]} &= [0.03, 0.01, 0.00], \\ x_3^{[0]} &= [0.20, 0.03, 0.14]. \end{aligned}$$

The numerical outcomes using initial approximation (47) are presented in Table 5.

Table 9 shows that for random initial values, our newly developed inverse family of parallel schemes BM[*] outperforms the existing PM[*], NN[*], and NF[*] methods in terms of maximum error, number of iterations needed for convergence, total number of basic arithmetic operations per iteration, percentage divergence, and local computational order of convergence. Consequently, our parallel computation algorithm BM[*] is more flexible, consistent, and efficient than the current PM[*], NN[*], and NF[*] methods. Figure 5 shows the maximum residual error on random iterations.

Table 9. Consistency analysis of the parallel schemes for (31).

Scheme	Max-Err	Max-It	$[\pm, \times, \div]$	P-Con	Local-COC
PM[*]	7.1×10^{-03}	10	67	33.879%	4.0653403
NN[*]	8.8×10^{-07}	8	41	78.876%	5.67545645
NF[*]	0.9×10^{-11}	7	34	87.875%	5.87000945
SB[*]	2.2×10^{-18}	5	29	98.765%	6.11343546

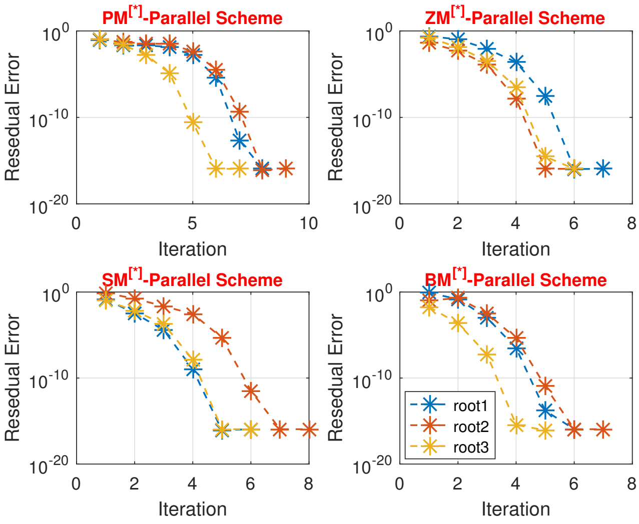


Figure 5. Error graph of the parallel scheme for solving (44) using random test vectors.

4.4. Example 4: Fractional Relaxation Oscillation Equations: Chemical Engineering Application

An important tool in chemical engineering, fractional relaxation oscillation equations shed light on processes such as memory effects and anomalous diffusion that frequently occur in complicated fluids and porous media. In situations where conventional integer-order models might not be adequate, these equations use fractional derivatives to simulate oscillatory and relaxation behaviors. Viscoelastic materials and catalytic processes are examples of systems with time-dependent features whose dynamics can be precisely captured by engineers by varying the fractional order. They are very helpful in chemical engineering for modeling processes such as transport phenomena, adsorption, and reaction kinetics. This fractional technique improves prediction and control when developing effective separation systems and reactors. It also helps to optimize parameters in order to attain the necessary level of process stability and product quality. The fractional relaxation can be mathematically modeled [43], leading to the following nonlinear differential equations:

$$\begin{cases} \frac{d^n g(x)}{dx^n} + \phi^{[*]} g(x) = g^n(x) + 1; \sigma_0 \leq x \leq \sigma_0^{[*]} \\ \frac{d^{n-1} g(\sigma_0)}{dx^{n-1}} = \sigma_{n-1}^{[*]}, \\ \frac{d^{n-2} g(\sigma_0)}{dx^{n-2}} = \sigma_{n-2}^{[*]}, \\ \vdots \\ g(\sigma_0) = \sigma_0^{[*]} \end{cases} \tag{48}$$

where $n = 2$, $\sigma_{n-1}^{[*]} = 1, \sigma_{n-2}^{[*]} =, \ldots, = \sigma_{*}^{[*]} = 0$. Using the method defined in [44], we simulated (48) by the following nonlinear function:

$$g(x) = \frac{x}{\Gamma(2)} + \frac{x^2}{\Gamma(3)} + \frac{2x^4}{\Gamma(5)} + \frac{6x^5}{\Gamma(6)} + \frac{6x^6}{\Gamma(7)} \tag{49}$$

where $\Gamma(.)$ represents the gamma function. The exact roots up to one decimal place were obtained using the built-in "Solve[.]" commend in MatLab:

$$\zeta_{1,2} = 1.2401 \pm 1.7446i, \zeta_{3,4} = -3.3561 \pm 1.7321i, \zeta_5 = 0.0, \zeta_6 = -1.608. \tag{50}$$

For higher accuracy, we utilized the information regarding parameter selection in Table 1 to implement the parallel scheme to simultaneously find all solutions, as shown in Table 10, by choosing the following rough initial approximations:

$$[x_1^{[0]}, x_2^{[0]}, x_3^{[0]}, x_4^{[0]}, x_5^{[0]}, x_6^{[0]}] = \begin{bmatrix} 3+2i, 3-2i, -6+3i, \\ -6-3i, 3.0, -3.0 \end{bmatrix}. \tag{51}$$

Table 10. Consistency analysis of the parallel schemes for solving (48).

Scheme	Max-Error	Max-It	$[\pm, \times, \div]$	Per-Divergence	Local-COC
ZM$^{[*]}$	6.1×10^{-05}	8	45	18.657656%	4.40075148
PM$^{[*]}$	1.5×10^{-12}	8	35	38.87576%	5.11345645
SM$^{[*]}$	3.9×10^{-14}	6	27	51.65865%	5.77000175
BM$^{[*]}$	1.0×10^{-19}	5	19	88.76878%	6.08009465

Using the information in Table 1, we selected the optimal parameter values and achieved the numerical results displayed in Table 11 for extremely rough initial estimations.

Table 11. Numerical results of parallel schemes for solving (48).

Method	$e_1^{[6]}$	$e_2^{[6]}$	$e_3^{[6]}$	$e_4^{[6]}$	$e_5^{[6]}$	$e_6^{[6]}$	CPU-time
ZM$^{[*]}$	0.1×10^{-15}	5.1×10^{-14}	0.0	1.7×10^{-44}	1.3×10^{-21}	4.1×10^{-35}	4.324242
PM$^{[*]}$	1.0×10^{-32}	1.1×10^{-32}	0.0	1.0×10^{-27}	5.6×10^{-29}	6.0×10^{-42}	3.535335
SM$^{[*]}$	9.9×10^{-54}	9.1×10^{-31}	0.0	3.0×10^{-24}	5.5×10^{-25}	0.0	2.356464
BM$^{[*]}$	8.7×10^{-64}	0.0	0.0	0.1×10^{-74}	0.0	0.0	1.543563

Table 4 clearly shows that our novel approaches are more efficient in terms of residual error and computing time in seconds when compared to the existing PM$^{[*]}$, NN$^{[*]}$, and NF$^{[*]}$ methods. Taking into account a random initial starting value, we can check the method's global convergence as follows:

$$\begin{aligned} x^{[0]} &= [x_1^{[0]}, x_2^{[0]}, x_3^{[0]}, x_4^{[0]}, x_5^{[0]}, x_6^{[0]}], \\ x_1^{[0]} &= [0.23, 0.92, 0.53, 0.54, 24, 06], \\ x_2^{[0]} &= [0.09, 0.21, 0.01, 0.34, 65, 70], \\ x_3^{[0]} &= [0.00, 0.32, 0.03, 0.34, 30, 13]. \end{aligned} \tag{52}$$

The numerical outcomes using initial approximation (52) are presented in Table 10.

Table 10 shows that for random initial values, our newly developed inverse family of parallel schemes BM$^{[*]}$ performs better then the classical parallel methods PM$^{[*]}$, NN$^{[*]}$, and NF$^{[*]}$ in terms of the maximum error, number of iterations required for convergence, total number of basic arithmetic operations per iteration, percentage divergence, and local computational order of convergence. Therefore, in comparison to the existing methods PM$^{[*]}$, NN$^{[*]}$, and NF$^{[*]}$, our parallel approach BM$^{[*]}$ is more reliable, consistent, and effective. In Figure 6, the maximum residual error on random iterations is depicted.

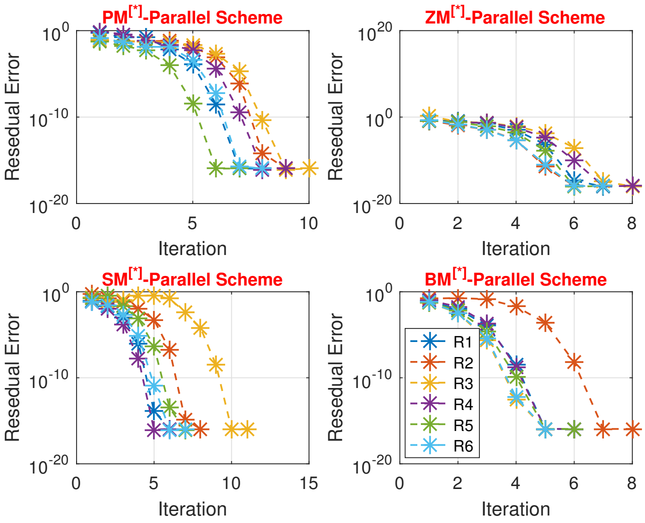


Figure 6. Error graph of the parallel scheme for solving (48) using random test vectors.

Results and Discussion

An inverse biparametric family of parallel algorithms BM$^{[*]}$ was developed in this study to solve nonlinear equations more efficiently. Bifurcation diagrams displaying chaotic behavior in the parameter space were used to determine the ideal parameter values that result in solution stability and enhanced performance.

Efficiency Analysis: The suggested approach was tested on three different applications and compared to other methods (PM$^{[*]}$, NN$^{[*]}$, NF$^{[*]}$). The key performance indicators we examined included the error, CPU time, percentage convergence–divergence, and percentage computational efficiency. Our suggested approach performed better than the current approaches in these tests in the following ways:

- Our $BM^{[*]}$ technique outperformed the $PM^{[*]}$, $NN^{[*]}$, and $NF^{[*]}$ parallel algorithms in terms of overall computation efficiency when accuracy and speed were combined (see Tables 2 and 3). This high percentage efficiency demonstrates its practical applicability.
- The suggested method's error was substantially lower than that of $PM^{[*]}$, $NN^{[*]}$, and $NF^{[*]}$. This further illustrates its accuracy and dependability in root-finding procedures (see Tables 4, 6, 8 and 11).
- The proposed method had lower computational costs as assessed by CPU time. These advantages can be ascribed to carefully selected parameter values and an optimized parallel structure (see Tables 5, 7, 9 and 10 and Figures 3–6).
- Our algorithm exhibited a higher proportion of convergence to the correct roots and fewer divergent scenarios (see Table 1 and Figures 1 and 2). This indicates its robust stability across a wide range of starting estimations and problem environments.

The validity of the inverse biparametric family of parallel schemes is confirmed by the results. The proposed technique using bifurcation analysis for parameter optimization represents a more reliable, faster, and accurate alternative to classical schemes such as $PM^{[*]}$, $NN^{[*]}$, and $NF^{[*]}$. The proposed approach provides a significant advancement in solving nonlinear equations, especially in cases when computing efficiency and precision are crucial.

The proposed parallel techniques improved computing performance in engineering models, but have limitations when applied to nonlinear equations or higher-dimensional problems due to their increased complexity. Using parallel techniques for root-finding of dimension higher than one requires significant reconfiguration.

In the future, the proposed approach could be used to solve systems of nonlinear equations or generic higher-dimensional problems. To achieve this, research on novel parallel schemes tailored specifically to multidimensional spaces, as well as to convergence and efficiency, may be required. Hybrid strategies that combine the proposed method with global optimization techniques such as swarm intelligence or evolutionary algorithms will be more reliable in reducing the risk of initial estimates by investigating globalization strategies.

5. Conclusions

In this study, we construct a new family of inverse biparametric parallel techniques $BM^{[*]}$ in order to improve the stability, convergence order, and efficiency of parallel iterative schemes. Utilizing the concepts of bifurcation and chaos theory, we computed the ideal parameter values in which the biparametric family of inverse parallel techniques shows the best stability and fastest convergence (see, e.g., Figures 1–3). According to a comparison with the current $PM^{[*]}$, $NN^{[*]}$, and $NF^{[*]}$ techniques, the suggested strategies not only produce faster convergence rates but also enhance computing efficiency. This was assessed using percentage computational efficiency, revealing a considerable reduction in computational cost and a more efficient iteration process (see Tables 1 and 2).

The usefulness of the proposed biparametric family of inverse parallel schemes $BM^{[*]}$ was further tested through engineering applications. Even with rough initial values, the proposed schemes show strong convergence behavior and good accuracy (see, e.g., Tables 4, 6, 8 and 11). We also tested the $BM^{[*]}$,$PM^{[*]}$,$NN^{[*]}$, and $NF^{[*]}$ algorithms on random initial values to investigate their global convergence qualities. The findings demonstrated that our $BM^{[*]}$ method consistently beats the previous $PM^{[*]}$, $NN^{[*]}$, and $NF^{[*]}$ approaches in terms of maximum error, number of iterations, basic arithmetic operations per iteration, and local computational order of convergence (see Tables 5, 7, 9 and 10 and Figures 4–6).

Future work might concentrate on applying these biparametric families of inverse parallel methods to more complex scientific and engineering applications where reliable and quick convergence is essential. Furthermore, investigating adaptive parameter tuning processes may improve the stability and effectiveness of the proposed parallel approach in a variety of contexts. Higher computational efficiency could be achieved by optimizing these techniques for high-performance computing settings [45,46] through more research into parallel computation implementations.

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

Funding: The work was supported by the Free University of Bozen-Bolzano (IN200Z SmartPrint) and by Provincia Autonoma di Bolzano/Alto Adige-Ripartizione Innovazione, Ricerca, Università e Musei (CUP codex I53C22002100003 PREDICT). Bruno Carpentieri 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 Progetti di Ricerca 2024.

Data Availability Statement: Data are contains within the article.

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

Abbreviations

In this article, the following abbreviations are used:

BM$^{[*]}$	Newly developed schemes
n	Iterations
CPU time	Computational time in seconds
e−	$10^{-()}$
$Local-COC$	Computational local convergence order

References

1. Sulaiman, T.A.; Bulut, H.; Baskonus, H.M. On the exact solutions to some system of complex nonlinear models. *Appl. Math. Nonlinear Sci.* **2021**, *6*, 29–42. [CrossRef]
2. Magli, G. *Archaeoastronomy: Introduction to the Science of Stars and Stones*; Springer Nature: Berlin/Heidelberg, Germany, 2020.
3. Wengrow, D. Landscapes of knowledge, idioms of power: The African foundations of ancient Egyptian civilization reconsidered. In *Ancient Egypt in Africa*; Routledge: London, UK, 2016; pp. 121–135.
4. Yehia, H.M.A.A.S. Sound Techniques in Ancient Civilizations: An Analytical Study of the Geometric Shapes of Places of Worship. *Am. J. Civ. Eng. Archit.* **2024**, *12*, 8–13.
5. Sen, S.; Agarwal, S.; Chakraborty, P.; Singh, K.P. Astronomical big data processing using machine learning: A comprehensive review. *Exp. Astron.* **2022**, *53*, 1–43. [CrossRef]
6. Van Quang, T.; Doan, D.T.; Yun, G.Y. Recent advances and effectiveness of machine learning models for fluid dynamics in the built environment. *Int. J. Model. Simul.* **2024**, 1–27. [CrossRef]
7. Smith, R.C. *Uncertainty Quantification: Theory, Implementation, and Applications*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2024.
8. Zhang, A.; Gao, J. Comprehensive analysis of linear and nonlinear equivalent circuit model for GaAs–PIN diode. *IEEE Trans. Ind. Electron.* **2021**, *69*, 11541–11548. [CrossRef]
9. Keshtegar, B.; Nehdi, M.L.; Trung, N.T.; Kolahchi, R. Predicting load capacity of shear walls using SVR—-RSM model. *Appl. Soft Comput.* **2021**, *112*, 107739. [CrossRef]

10. Mashuri, A.; Adenan, N.H.; Abd Karim, N.S.; Tho, S.W.; Zeng, Z. Application of Chaos Theory in Different Fields–A Literature Review. *J. Sci. Math. Lett.* **2024**, *12*, 92–101. [CrossRef]
11. Ali, A.; Jamshad, A.; Sara, J. Exact soliton solutions and stability analysis to (3+1)—Dimensional nonlinear Schrö dinger model. *Alex. Eng. J.* **2023**, *76*, 747–756. [CrossRef]
12. Ahmad, J.; Mustafa, Z. Dynamics of exact solutions of nonlinear resonant Schrödinger equation utilizing conformable derivatives and stability analysis. *Eur. Phys. J. D* **2023**, *77*, 123. [CrossRef]
13. Liaqat, M.I.; Etemad, S.; Rezapour, S.; Park, C. A novel analytical Aboodh residual power series method for solving linear and nonlinear time–fractional partial differential equations with variable coefficients. *Aims. Math.* **2022**, *7*, 16917–16948. [CrossRef]
14. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics—Informed neural networks: Where we are and what's next. *J. Sci. Comput.* **2022**, *92*, 88. [CrossRef]
15. Pouzet, M. A projection property and Arrow's impossibility theorem. *Discret. Math.* **1998**, *192*, 293–308. [CrossRef]
16. Ypma, T.J. Historical development of the Newton—-Raphson method. *SIAM Rev.* **1995**, *37*, 531–551. [CrossRef]
17. Erfanifar, R.; Hajarian, M. A new multi-step method for solving nonlinear systems with high efficiency indices. *Numer. Algorithms* **2024**, *97*, 1–26. [CrossRef]
18. Halilu, A.S.; Waziri, M.Y.; Abdullahi, H.; Majumder, A. On the hybridization of the double step length method for solving system of nonlinear equations. *Malays. J. Math. Sci.* **2022**, *16*, 329–349. [CrossRef]
19. Moscoso–Martínez, M.; Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. Performance of a new sixth–order class of iterative schemes for solving non–linear systems of Equations. *Mathematics* **2023**, *11*, 1374. [CrossRef]
20. Liu, C.S.; Chang, C.W.; Kuo, C.L. Memory–Accelerating Methods for One–Step Iterative Schemes with Lie Symmetry Method Solving Nonlinear Boundary–Value Problem. *Symmetry* **2024**, *16*, 120. [CrossRef]
21. Chen, Y.; Leung, A.Y. *Bifurcation and Chaos in Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
22. Cholakov, S.I.; Vasileva, M.T. A convergence analysis of a fourth–order method for computing all zeros of a polynomial simultaneously. *J. Comput. Appl. Math.* **2017**, *321*, 270–283. [CrossRef]
23. Shams, M.; Kausar, N.; Araci, S.; Kong, L.; Carpentieri, B. Highly efficient family of two-step simultaneous method for all polynomial roots. *AIMS Math.* **2024**, *9*, 1755–1771. [CrossRef]
24. Marcheva, P.I.; Ivanov, S.I. Convergence analysis of a modified Weierstrass method for the simultaneous determination of polynomial zeros. *Symmetry* **2020**, *12*, 1408. [CrossRef]
25. Nedzhibov, G.H. Inverse Weierstrass–Durand–Kerner Iterative Method. *Int. J. Appl. Math.* **2013**, *28*, 1258–1264.
26. Proinov, P.D.; Vasileva, M.T. A new family of high–order ehrlich–type iterative methods. *Mathematics* **2021**, *9*, 1855. [CrossRef]
27. Shams, M.; Carpentieri, B. Efficient inverse fractional neural network-based simultaneous schemes for nonlinear engineering applications. *Fractal Fract.* **2023**, *7*, 849. [CrossRef]
28. Anourein, A.W.M. An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial. *Inter. J. Comput. Math.* **1977**, *6*, 241–252. [CrossRef]
29. Zein, A. A general family of fifth-order iterative methods for solving nonlinear equations. *Eur. J. Pure Appl. Math.* **2023**, *16*, 2323–2347. [CrossRef]
30. Shams, M.; Kausar, N.; Araci, S.; Kong, L. On the stability analysis of numerical schemes for solving non-linear polynomials arises in engineering problems. *AIMS Math.* **2024**, *9*, 8885–8903. [CrossRef]
31. Zhang, E.; Shateyi, S. Exploring limit cycle bifurcations in the presence of a generalized heteroclinic loop. *Mathematics* **2023**, *11*, 3944. [CrossRef]
32. Fernández–Díaz, A. Overview and perspectives of chaos theory and its applications in economics. *Mathematics* **2023**, *12*, 92. [CrossRef]
33. Cross, N. *Engineering Design Methods: Strategies for Product Design*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
34. Aithal, P.S.; Aithal, S. Application of ChatGPT in higher education and research—A futuristic analysis. *Int. J. Appl. Eng. Manag. Lett. (IJAEML)* **2023**, *7*, 168–194. [CrossRef]
35. Salinger, A.G.; Burroughs, E.A.; Pawlowski, R.P.; Phipps, E. T.; Romero, L.A. Bifurcation tracking algorithms and software for large scale applications. *Int. J. Bifurc. Chaos* **2005**, *15*, 1015–1032. [CrossRef]
36. Shams, M.; Carpentieri, B. An Efficient and Stable Caputo–Type Inverse Fractional Parallel Scheme for Solving Nonlinear Equations. *Axioms* **2024**, *13*, 671. [CrossRef]
37. Zhang, X.; Peng, H.; Hu, G. A high order iteration formula for the simultaneous inclusion of polynomial zeros. *Appl. Math. Comput.* **2006**, *179*, 545–552. [CrossRef]
38. Petkovic, M. *Iterative Methods for Simultaneous Inclusion of Polynomial Zeros (Vol. 1387)*; Springer: Berlin/Heidelberg, Germany, 2006.
39. Rafiq, N.; Akram, S.; Mir, N.A.; Shams, M. Study of dynamical behavior and stability of iterative methods for nonlinear equation with applications in engineering. *Math. Probl. Eng.* **2020**, *2020*, 3524324. [CrossRef]

40. Teodorescu, P.; Stanescu, N.D.; Pandrea, N. *Numerical Analysis with Applications in Mechanics and Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
41. Hamming, R.W. *Introduction to Applied Numerical Analysis*; Courier Corporation: North Chelmsford, MA, USA, 2012.
42. Phillips, G.M.; Taylor, P.J. (Eds.) *Theory and Applications of Numerical Analysis*; Elsevier: Amsterdam, The Netherlands, 1996.
43. Gerald, C.F. *Applied Numerical Analysis*; Pearson Education India: Delhi, India, 2004
44. Odibat; Momani, S. Application of variational iteration method to nonlinear differential equations of fractional order. *Int. J. Nonlinear Sci. Numer.* **2006**, *7*, 27–34. [CrossRef]
45. Moiz, A.A.; Pal, P.; Probst, D.; Pei, Y.; Zhang, Y.; Som, S.; Kodavasal, J. A machine learning–genetic algorithm (ML–GA) approach for rapid optimization using high-performance computing. *SAE Int. J. Commer. Veh.* **2018**, *11*, 291–306. [CrossRef]
46. Mohammed, A.; Eleliemy, A.; Ciorba, F.M.; Kasielke, F.; Banicescu, I. An approach for realistically simulating the performance of scientific applications on high performance computing systems. *Future Gener. Comput. Syst.* **2020**, *111*, 617–633. [CrossRef]

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