

Article

# On a Stable Multiplicative Calculus-Based Hybrid Parallel Scheme for Nonlinear Equations

Mudassir Shams <sup>1,2</sup>

<sup>1</sup> Faculty of Engineering, Free University of Bozen-Bolzano (BZ), 39100 Bolzano, Italy; mudassir4shams@gmail.com or mudassir.shams@unibz.it

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

**Abstract:** Fractional-order nonlinear equation-solving methods are crucial in engineering, where complex system modeling requires great precision and accuracy. Engineers may design more reliable mechanisms, enhance performance, and develop more accurate predictions regarding outcomes across a range of applications where these problems are effectively addressed. This research introduces a novel hybrid multiplicative calculus-based parallel method for solving complex nonlinear models in engineering. To speed up the method's rate of convergence, we utilize a second-order multiplicative root-finding approach as a corrector in the parallel framework. Using rigorous theoretical analysis, we illustrate how the hybrid parallel technique based on multiplicative calculus achieves a remarkable convergence order of 12, indicating its effectiveness and efficiency in solving complex nonlinear equations. The intrinsic stability and consistency of the approach—when applied to nonlinear situations—are clearly indicated by the symmetry seen in the dynamical planes for various parameter values. The method's symmetrical behavior indicates that it produces accurate findings under a range of scenarios. Using a dynamical system procedure, the ideal parameter values are systematically analyzed in order to further improve the method's performance. Implementing the aforementioned parameter values using the parallel approach yields very reliable and consistent outcomes. The method's effectiveness, reliability, and consistency are evaluated through the analysis of numerous nonlinear engineering problems. The analysis provides a detailed comparison with current techniques, emphasizing the benefits and potential improvements of the novel approach.



**Citation:** Shams, M. On a Stable Multiplicative Calculus-Based Hybrid Parallel Scheme for Nonlinear Equations. *Mathematics* **2024**, *12*, 3501. <https://doi.org/10.3390/math12223501>

Academic Editor: Takashi Suzuki

Received: 20 September 2024

Revised: 21 October 2024

Accepted: 7 November 2024

Published: 9 November 2024



**Copyright:** © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** nonlinear equations; multiplicative calculus; multiplicative scheme; convergence planes; computational time

**MSC:** 65H04; 65H05; 65H10; 65H20; 65H17

## 1. Introduction

Nonlinear equations have a long history in science and engineering, dating back to the 18th and 19th centuries, when they were used to research celestial mechanics and fluid dynamics [1–3]. These formulas explain complex phenomena such as chaos [4], bifurcations [5], and solitons in systems with non-linear outputs [6]. In areas such as quantum physics, thermodynamics, and material science, nonlinear equations, e.g.,

$$f(x) = 0 \quad (1)$$

are necessary for describing real-world phenomena.

Many physical processes with memory effects, anomalous diffusion [7], or hereditary features can be better represented using fractional differential equations [8], which are fundamentally nonlinear. When traditional integer-order models fail, fractional-order models enhance the flexibility and precision of system modeling. Both nonlinear equations and fractional differential equations are important in describing the complexities of dynamic

systems across a range of scientific and engineering challenges, often leading to the following differential equations [9,10]:

$$\left\{ \begin{array}{l} \frac{d^n f(x)}{dx^n} + \left( \frac{d^{\frac{n-1}{2}} f(x)}{dx^{\frac{n-1}{2}}} \right) + (f(x))^{n-1} = x^{n+5}, \alpha_{01}^{[*]} \leq x \leq \alpha_1^{[*]}, \\ f(\alpha_{01}^{[*]}) = \left( \frac{d^{n-3} f(\alpha_{01}^{[*]})}{dx^{n-3}} \right) = \left( \frac{d^{n-2} f(\alpha_{01}^{[*]})}{dx^{n-2}} \right) = \alpha_{01}^{[*]}, \left( \frac{d^{n-1} f(\alpha_{01}^{[*]})}{dx^{n-1}} \right) = \alpha_{02}^{[*]}. \end{array} \right. \quad (2)$$

In Equation (2),  $\alpha_{01}^{[*]}$  represents the starting initial conditions and  $x \in [\alpha_{01}^{[*]}, \alpha_1^{[*]}]$ . Exact solutions to fractional-order equations are highly desirable because of their precision and ability to avoid approximations. However, finding precise solutions is frequently impossible due to the intrinsic complexity of fractional-order systems, which are frequently characterized by non-local features and memory effects. Analytical techniques like integral transformations, series expansions, and special functions are needed to move beyond this obstacle. By using these analytical or semi-analytical iterative techniques, closed or partially closed solutions can be constructed, which provide important details about the underlying processes and behaviors of complex systems. Even in situations where perfect solutions are not possible, analysts can gain a deeper understanding of fractional systems by utilizing these methodologies to capture both the qualitative and quantitative aspects of the system. Furthermore, these results from the analysis serve as standards for evaluating and improving numerical techniques. They contribute to the reliability and precision of numerical methods by offering standards by which numerical results may be evaluated, particularly in cases where exact solutions are difficult or impossible to achieve.

Multiplicative calculus [11], developed in the twentieth century, builds on classical calculus by focusing on growth rates in multiplicative terms rather than differences. It gained popularity in the 1970s with contributions from researchers such as Grossman and Katz, who formalized its fundamental notions by establishing a new type of derivative and integral, shifting the roles of subtraction and addition to division and multiplication [12]. The theoretical foundation of multiplicative calculus was established by Bashirov et al. [13,14], who also used it to develop multiplicative differential equations [15] and their engineering applications. It works especially well to explain scaling phenomena [16], geometric progressions [17], and exponential growth—all of which are less intuitive concepts in traditional calculus. Its fundamental advantage is that it provides a natural framework for many evolutionary processes, such as population growth, financial returns, and fractal systems. To represent some real-world phenomena more accurately and simply—where ratios and proportionalities are more significant than differences—this calculus makes use of multiplicative derivatives and integrals. In recent years, multiplicative calculus has been utilized in various fields; for example, Ozbay [18] used multiplicative calculus in the backpropagation processes of artificial neural networks (ANNs), Karthikeyan et al. [19] used it to explore the properties of a class of analytic functions, Othman et al. [20] developed a multiplicative calculus-based digital image interpolation technique, and Eyilmaz et al. [21] utilized it to investigate inverse nodal problems. It has also been used in many other works; see [22–24] and the references therein. Furthermore, Calogero et al. [25], Sana et al. [26], Mateen et al. [27], Bilgehan et al. [28], and Boruah et al. [29] used multiplicative calculus in their numerical analyses to develop numerical schemes for solving nonlinear engineering problems.

Motivated by the aforementioned work, the primary goal of this research is to develop a family of multiplicative calculus-based simple root-finding schemes and then extend these to parallel numerical schemes for approximating all solutions to (2) simultaneously, with global convergence behavior. This approach uses complex dynamical system methodologies to determine optimal parameter values, which are then utilized in parallel schemes to accelerate the rate of convergence.

Some fundamental multiplicative calculus concepts are reviewed, which were used in the design of our numerical techniques to find all solutions to (2) simultaneously.

**Definition 1.** A function  $\zeta : D \subset \mathbb{R} \rightarrow \mathbb{R}$  is said to be multiplicative differential at  $x$  or on  $D$  if  $\zeta > 0$ , and differentiable, respectively, at  $x$  or on  $\Omega$ , and satisfies the following limits:

$$\zeta^{[*]}(x) = \frac{d^{[*]}\zeta}{dx} = \lim_{h \rightarrow 0} \left( \frac{\zeta(x+h)}{\zeta(x)} \right)^{\frac{1}{h}}, \tag{3}$$

$$\begin{aligned} \zeta^{[*]}(x) &= \lim_{h \rightarrow 0} \left( \frac{\zeta(x+h)}{\zeta(x)} \right)^{\frac{1}{h}} = e^{\frac{\zeta'(x)}{\zeta(x)}}, \\ \zeta^{[*]}(x) &= e^{(\ln \circ \zeta)'(x)}, \end{aligned} \tag{4}$$

where  $\ln \circ \zeta = \ln(\zeta(x))$ . Similarly, the second multiplicative derivative is defined as follows [30]:

$$\zeta^{[**]}(x) = e^{(\ln \circ \zeta)''(x)}. \tag{5}$$

The higher-order multiplicative derivative is defined similarly, as follows:

$$\zeta^{[***]}(x) = e^{(\ln \circ \zeta)'''(x)} \tag{6}$$

and in a more general form, is as follows:

$$\zeta^{[*](n)}(x) = e^{(\ln \circ \zeta)^{(n)}(x)}, n = 0, 1, \dots \tag{7}$$

where  $n = 0$ , no multiplicative derivative exists, and it represents the original function  $\zeta(x) = 1$ .

**Definition 2.** Suppose  $\zeta : D \subset \mathbb{R} \rightarrow \mathbb{R}^+$  is a positive nonlinear function, then the multiplicative nonlinear equation [31] is defined as follows:

$$\zeta(x) = 1. \tag{8}$$

In multiplicative calculus, the multiplicative function is quite similar to the ordinary function in classical calculus. In classical calculus, changes are assessed additively, whereas in multiplicative calculus, changes are measured relatively or proportionately. Some important results of multiplicative differentiation are defined for multiplicative differentiable functions  $\zeta$  and  $h$  as follows:

- $(C)^{[*]} = 1,$
- $(C\zeta)^{[*]}(x) = \zeta^{[*]}(x),$
- $(\zeta \circ h)^{[*]}(x) = \zeta^{[*]}(x)h^{[*]}(x),$
- $\left(\frac{\zeta}{h}\right)^{[*]}(x) = \frac{\zeta^{[*]}(x)}{h^{[*]}(x)},$
- $(\zeta^\psi)^{[*]}(x) = \zeta^{[*]}(x)\psi(x)\zeta(x)\psi'(x),$
- $(\zeta \circ \psi)^{[*]}(x) = \zeta^{[*]}(x)\psi(x)\psi'(x).$

The multiplicative Taylor theorem [32] is defined in the following theorem, which is used in the construction of the new numerical scheme for solving nonlinear problem (2).

**Theorem 1.** Let  $\zeta : D \rightarrow \mathbb{R}$  be a  $(n + 1)$ -times multiplicative differential in an open interval  $D$ , then for any  $x, x + a \in D, \exists$  a number  $\eta \in (0, 1)$  such that we have the following:

$$\zeta(x+a) = \prod_{j=0}^n \left( \zeta^{[*](j)}(x) \right)^{\frac{a^j}{j!}} \left( \zeta^{[*](n+1)}(x + \eta a) \right)^{\frac{a^{n+1}}{(n+1)!}}. \tag{9}$$

The layout of this paper is as follows: We examine the creation, convergence characteristics, and intricate dynamical analysis of simple and parallel multiplicative-order systems intended to solve Equation (1) in Section 2. The mathematical and computational aspects of the suggested method are extensively clarified in this section. In Section 3, I

developed a new multiplicative parallel technique for solving (1). Following that, Section 4 includes an in-depth evaluation of the method’s stability and efficacy using a variety of numerical examples, enhanced by comparisons with other approaches to emphasize its performance advantages. Finally, Section 5 summarizes the findings and discusses prospective areas for future research.

## 2. Development and Dynamic Analysis of Numerical Methods

The Newton–Raphson iterative method [33,34] is a classical iterative methodology that refines an initial estimate to solve nonlinear problems through a succession of linear approximations. Using derivatives, the method approximates the behavior of a nonlinear function close to a specified point, producing a series of values that converge to the true solution. Since its conceptual creation in the late 17th century, this method—which is renowned for its usually rapid convergence and is frequently quadratic—has served as a fundamental tool in numerical analysis. It is one of the methods used for solving nonlinear equations that is most frequently used in a variety of scientific and technical domains due to its effectiveness and reliability. Using multiplicative analysis, the multiplicative Newton theorem [35] is written as follows:

$$\zeta(x) = \zeta\left(x^{(l)}\right) \int_{x^{(l)}}^x \zeta^{[*]}(s) ds = \zeta\left(x^{(l)}\right) e^{x^{(l)} \int_{x^{(l)}}^x (\ln \zeta(s))' ds} \tag{10}$$

Using the Newton–Cotes quadrature [36] of 0-degree order for (10), we have the following:

$$\int_{x^{(l)}}^x \zeta^{[*]}(s) ds = e^{x^{(l)} \int_{x^{(l)}}^x (\ln \zeta(s))' ds} = e^{(x-x^{(l)}) (\ln(\zeta(x^{(l)}))')} = \left(\zeta^{[*]}(x^{(l)})\right)^{x-x^{(l)}} \tag{11}$$

Using  $\zeta(x) = 1$ , the multiplicative classical Newton–Raphson technique—an extension of the classical Newton iterative algorithm—is produced as follows:

$$x^{(l+1)} = x^{(l)} - \frac{\ln\left(\zeta\left(x^{(l)}\right)\right)}{\ln\left(\zeta^{[*]}(x^{(l)})\right)} \tag{12}$$

The multiplicative Newton technique retains the same convergence order as the classical method.

In multiplicative calculus, numerical schemes make it easier to treat scale-invariant processes, whereas classical calculus sometimes struggles with complicated transformations. Furthermore, multiplicative techniques avoid concerns such as sum divergence, which improves performance in circumstances with large-scale variations. They also improve the precision of error estimation in systems based on ratios rather than absolute values. Multiplicative relationships are especially important in fields such as biology, economics, fractal analysis, and differential equations. Furthermore, Singh et al. [37] proposed the multiplicative version of the Schröder method as follows:

$$u^{(l)} = x^{(l)} - \frac{\ln\left(\zeta\left(x^{(l)}\right)\right) \ln\left(\zeta^{[*]}(x^{(l)})\right)}{\left(\ln\left(\zeta^{[*]}(x^{(l)})\right)\right)^2 - \ln\left(\zeta^{[*]}(x^{(l)})\right) \ln\left(\zeta(x^{(l)})\right)} \tag{13}$$

This method has a convergence order of 2. Similarly, Waseem et al. [38] presented the following iterative approach with quadratic convergence, using multiplicative calculus:

$$u^{(l)} = x^{(l)} - \frac{\ln\left(\zeta\left(x^{(l)}\right)\right)}{\ln\left(\zeta^{[*]}(x^{(l)})\right) - \alpha \ln\left(\zeta(x^{(l)})\right)}$$

We suggest the following multiplicative technique to obtain simple roots of (1); we have the following:

$$v^{[l]} = x^{[l]} - \frac{\ln(\zeta(x^{[l]}))}{\ln \zeta^{[*]}(x^{[l]})} \left[ \frac{1}{1 + \alpha \left( \frac{\ln \zeta(x^{[l]})}{1 + \alpha \ln \zeta(x^{[l]})} + \beta (\ln \zeta(x^{[l]}))^2 \right)} \right]. \tag{14}$$

We abbreviate this method as CM<sup>[\*]</sup>.

2.1. Theoretical Convergence of the Multiplicative Calculus-Based Method

The theoretical convergence of the multiplicative calculus-based simple root-finding approach is essential since it demonstrates the technique’s dependability and effectiveness in accurately identifying roots. This convergence analysis assures that the approach moves gradually toward solutions, providing vital insights into its stability and performance over a wide range of applications. Verifying the method’s application in real-world scenarios—especially scenarios where traditional calculus-based approaches might not be as successful—requires an understanding of this convergence. In order to ascertain the order of convergence of the iterative scheme provided by (14), I establish the subsequent theorem:

**Theorem 2.** Let

$$\zeta : \mathcal{D} \subseteq \mathbb{R} \rightarrow \mathbb{R} \tag{15}$$

be a function that is continuous, with  $\ln \zeta^{m[*]}(x^{[l]})$  of order  $m$  containing the exact root  $\xi$  of  $\zeta(x^{[l]})$ . In addition, if the initial estimate  $x^{[0]}$  is sufficiently close, the scheme, i.e.,

$$v^{[l]} = x^{[l]} - \frac{\ln(\zeta(x^{[l]}))}{\ln \zeta^{[*]}(x^{[l]})} \left[ \frac{1}{1 + \alpha \left( \frac{\ln \zeta(x^{[l]})}{1 + \alpha \ln \zeta(x^{[l]})} + \beta (\ln \zeta(x^{[l]}))^2 \right)} \right] \tag{16}$$

has a convergence order of at least 2, utilizing the error formula described via the following:

$$\epsilon_v^{[l]} = (-\alpha - t_2) (\epsilon^{[l]})^2 + O\left((\epsilon^{[l]})^3\right). \tag{17}$$

**Proof.** Let  $\xi$  be a simple exact solution of  $\zeta(x) = 1$  and assume  $x^{[l]} = \xi + \epsilon^{[l]}$ . Using the multiplicative Taylor formula of  $\zeta(x)$  around  $x = \xi$ , we obtain the following:

$$\zeta(x^{[l]}) = \zeta(\xi + \epsilon^{[l]}) = \left( \begin{array}{l} \zeta(\xi) (\zeta^{[*]}(\xi))^{\epsilon^{[l]}} (\zeta^{[**]}(\xi))^{\frac{(\epsilon^{[l]})^2}{2!}} \\ (\zeta^{[***]}(\xi))^{\frac{(\epsilon^{[l]})^3}{3!}} + O\left((\epsilon^{[l]})^4\right) \end{array} \right). \tag{18}$$

Using the natural logs on both sides of (18), we obtain the following:

$$\ln \zeta(x^{[l]}) = \left( \begin{array}{l} \ln \zeta(\xi) + \ln \zeta^{[*]}(\xi) \epsilon^{[l]} + \ln \zeta^{[**]}(\xi) \frac{(\epsilon^{[l]})^2}{2!} + \\ \ln \zeta^{[***]}(\xi) \frac{(\epsilon^{[l]})^3}{3!} + O\left((\epsilon^{[l]})^4\right) \end{array} \right), \tag{19}$$

$$\ln \zeta(x^{(l)}) = \ln \zeta^{[*]}(\zeta) \left[ \begin{array}{c} \epsilon^{[l]} + \frac{1}{2!} \frac{\ln \zeta^{[**]}(\zeta)}{\ln \zeta^{[*]}(\zeta)} (\epsilon^{[l]})^2 + \\ \frac{1}{3!} \frac{\ln \zeta^{[***]}(\zeta)}{\ln \zeta^{[*]}(\zeta)} (\epsilon^{[l]})^3 + O\left((\epsilon^{[l]})^4\right) \end{array} \right], \tag{20}$$

$$\ln \zeta(x^{(l)}) = \ln \zeta^{[*]}(\zeta) \left[ \epsilon^{[l]} + t_2 (\epsilon^{[l]})^2 + t_3 (\epsilon^{[l]})^3 + O\left((\epsilon^{[l]})^4\right) \right], \tag{21}$$

where  $t_2 = \frac{1}{j!} \frac{\ln \zeta^{[j*]}(\zeta)}{\ln \zeta^{[*]}(\zeta)}$ ;  $j \geq 2$ . Taking the derivative of (21), we have the following:

$$\ln \zeta^{[*]}(x^{(l)}) = \left( \begin{array}{c} \ln \zeta^{[*]}(\zeta) + \ln \zeta^{2[**]}(\zeta) (\epsilon^{[l]}) + \\ \ln \zeta^{3[***]}(\zeta) (\epsilon^{[l]})^2 + O\left((\epsilon^{[l]})^3\right) \end{array} \right), \tag{22}$$

$$\ln \zeta^{[*]}(x^{(l)}) = \ln \zeta^{[*]}(\zeta) \left[ \begin{array}{c} 1 + \frac{1}{2!} \frac{\ln \zeta^{[**]}(\zeta)}{\ln \zeta^{[*]}(\zeta)} \epsilon^{[l]} + \frac{1}{3!} \frac{\ln \zeta^{[***]}(\zeta)}{\ln \zeta^{[*]}(\zeta)} (\epsilon^{[l]})^2 \\ + O\left((\epsilon^{[l]})^3\right) \end{array} \right], \tag{23}$$

$$\ln \zeta^{[*]}(x^{(l)}) = \ln \zeta^{[*]}(\zeta) \left[ 1 + 2t_2^{[l]} \epsilon^{[l]} + 3t_3 (\epsilon^{[l]})^2 + O\left((\epsilon^{[l]})^3\right) \right]. \tag{24}$$

Dividing  $\ln \zeta(x^{(l)})$  by  $\ln \zeta^{[*]}(x^{(l)})$ , we have the following:

$$\frac{\ln(\zeta(x^{(l)}))}{\ln \zeta^{[*]}(x^{(l)})} = \epsilon^{[l]} + t_2 (\epsilon^{[l]})^2 + 2(t_3 - t_2^2) (\epsilon^{[l]})^3 + O\left((\epsilon^{[l]})^4\right). \tag{25}$$

Thus, we have the following:

$$1 + \alpha \ln \zeta(x^{(l)}) = 1 + \alpha \epsilon^{[l]} + \alpha t_2 (\epsilon^{[l]})^2 + \alpha t_3 (\epsilon^{[l]})^3 + \dots \tag{26}$$

Taking the inverse of (26), we have the following:

$$\frac{1}{1 + \alpha \ln \zeta(x^{(l)})} = 1 - \alpha \epsilon^{[l]} + (\alpha^2 - \alpha t_2) (\epsilon^{[l]})^3 + \dots \tag{27}$$

$$\begin{aligned} & 1 + \alpha \left( \frac{\ln \zeta(x^{(l)})}{1 + \alpha \ln \zeta(x^{(l)})} + \beta (\ln \zeta(x^{(l)}))^2 \right) \\ &= \left[ \begin{array}{c} 1 + \alpha \epsilon^{[l]} + \left( \begin{array}{c} -\alpha^2 + \alpha \beta \\ + \alpha t_2 \end{array} \right) (\epsilon^{[l]})^2 + \\ \left( \begin{array}{c} \alpha^3 - 2\alpha^2 t_2 \\ + 2\alpha \beta t_2 \\ + \alpha t_3 \end{array} \right) (\epsilon^{[l]})^3 + \dots \end{array} \right]. \end{aligned} \tag{28}$$

Using (28) in inverse form, we have the following:

$$\frac{1}{1 + \alpha \left( \frac{\ln \zeta(x^{(l)})}{1 + \alpha \ln \zeta(x^{(l)})} + \beta (\ln \zeta(x^{(l)}))^2 \right)} = 1 - \alpha \epsilon^{[l]} + \left( \begin{array}{c} 2\alpha^2 \\ -\alpha \beta \\ -\alpha t_2 \end{array} \right) (\epsilon^{[l]})^2 + \dots \tag{29}$$

Multiplying (25) and (29) yields the following:

$$\frac{\ln(\zeta(x^{[l]}))}{\ln \zeta^{[*]}(x^{[l]})} \left[ \frac{1}{1 + \alpha \left( \frac{\ln \zeta(x^{[l]})}{1 + \alpha \ln \zeta(x^{[l]})} + \beta (\ln \zeta(x^{[l]}))^2 \right)} \right] \tag{30}$$

$$= \epsilon^{[l]} + \begin{pmatrix} -\alpha \\ -t_2 \end{pmatrix} (\epsilon^{[l]})^2 + \begin{pmatrix} 2\alpha^2 - \alpha\beta \\ +2t_2^2 \\ -\alpha t_2 \end{pmatrix} (\epsilon^{[l]})^3 + \dots \tag{31}$$

Using (31) in (14), we obtain the following:

$$v^{[l]} - \zeta = x^{[l]} - \zeta - \frac{\ln \zeta(x^{[l]})}{\ln \zeta^{[*]}(x^{[l]})} \left[ \frac{1}{1 + \alpha \left( \frac{\ln \zeta(x^{[l]})}{1 + \alpha \ln \zeta(x^{[l]})} + \beta (\ln \zeta(x^{[l]}))^2 \right)} \right], \tag{32}$$

$$\epsilon_v^{[l]} = (-\alpha - t_2) (\epsilon^{[l]})^2 + O\left((\epsilon^{[l]})^3\right). \tag{33}$$

Thus, the theorem is proven. □

### 2.2. Dynamical Analysis of the Multiplicative Scheme CM<sup>[\*]</sup>

An in-depth analysis of the dynamic behavior inherent in solution approaches is necessary to ensure the reliability and resilience of nonlinear equation-solving methods [39,40]. Dynamic analysis seeks to determine how well a multiplicative calculus-based method can converge to the exact solution given an initial approximation, even in the presence of slight errors or instabilities during computation. In general, root-finding algorithms are effective when the initial guess is somewhat close to the real root (this is a phenomenon known as local convergence). The function must be performed effectively for these strategies to be the most effective, and their starting point must be carefully determined. However, the accuracy of the initial estimate and the characteristics of the function being solved have a significant impact on the stability and effectiveness of such algorithms. In circumstances where the function exhibits irregular or complex behavior, or when the initial starting values are very far from the exact solutions, classical parallel iterative techniques may fail to converge or produce a solution that is far from the exact solutions. As a consequence, it is essential to consider these characteristics when choosing or developing root-finding approaches in order to ensure reliable and exact results across a wide range of nonlinear problems.

Complex dynamical systems are used to assess the stability of a root-finding algorithm, which is crucial for its performance. These systems help examine the sensitivity of the solution to small changes in input values. Furthermore, the convergence criterion plays an important role in determining how quickly and consistently the algorithm approaches the true root. Stability often requires a careful balance between the intrinsic convergence features of the approach and the careful selection of important parameters, including the initial guess, the particular attributes of the function being solved, and the conditions under which the algorithm stops. Enhancing these components can improve the method's resilience and dependability and guarantee that the result will remain consistent even in a fluctuating computing environment. This method is especially crucial in applications

where accuracy and efficiency are critical, as minor errors in inputs or estimates can result in considerable differences in the final result.

The rational multiplicative calculus-based iterative map  $f(x) = (x - a)(x - b)$  is obtained as follows:

$$R(x) = - \frac{\left[ \begin{aligned} &\ln(A_1^{[*]}) \ln(B_1^{[*]})^3 \beta \alpha^2 x + \ln(A_1^{[*]}) (B_1^{[*]})^2 \beta \alpha x + \\ &2 \ln(A_1^{[*]}) (B_1^{[*]}) \alpha x - \alpha \ln(B_1^{[*]})^2 + \\ &x \ln(A_1^{[*]}) - \ln(B_1^{[*]}) \end{aligned} \right]}{\ln(A_1^{[*]}) \ln(B_1^{[*]})^3 \beta \alpha^2 + \beta \alpha \ln(B_1^{[*]})^2 + 2 \alpha \ln(B_1^{[*]}) + 1}, \tag{34}$$

where  $A_1^{[*]} = e^{\frac{2x}{x^2-1}}$ ,  $B_1^{[*]} = x^2 - 1$  and  $a, b$  belong to  $\mathbb{C}$ . Although algorithms may transform complex planes with preserved angles and arcs, Möbius transformations (MST) are crucial for conformal mappings in complex analysis and geometry. Many disciplines, including physics, engineering, and computer graphics, use their adaptability to simplify and understand intricate patterns and behaviors. Thus, iterative map (34) depends on  $a, b$ , and  $x$ . The MST [41]  $M(x) = \frac{x-a}{x-b}$ , is used to demonstrate that the multiplicative calculus-based rational map  $R(x)$  is conjugate with the operator as follows:

$$O(x) = \frac{\left[ \begin{aligned} &(A_2^{[*]}) \beta x + 4 \ln(A_2^{[*]}) \ln(B_2^{[*]}) \alpha x + 4 \ln(2)^2 \alpha x + \\ &4 \ln(B_2^{[*]}) \ln(2) \alpha x + \ln(B_2^{[*]})^2 \alpha x - 4 \ln(2)^2 \alpha - \\ &4 \ln(2) \ln(B_2^{[*]}) \alpha - \ln(B_2^{[*]})^2 \alpha - \\ &2 \ln(A_2^{[*]}) x + 2 \ln(A_2) x + \\ &\ln(B_2^{[*]}) x - 2 \ln(2) - \ln(B_2^{[*]}) \end{aligned} \right]}{\left[ \begin{aligned} &(A_2^{[*]}) \beta x + 4 \ln(2)^2 \alpha x + 4 \ln(2) \ln(B_2^{[*]}) \alpha x \\ &+ \ln(B_2^{[*]})^2 \alpha x + 8 \ln(A_2^{[*]}) \ln(2) \alpha + \\ &4 \ln(A_2^{[*]}) \ln(2) \alpha - \ln(2)^2 \alpha - 4 \ln(2) \ln(B_2^{[*]}) \alpha - \\ &\ln(B_2^{[*]})^2 \alpha + 2 \ln(2) x + \ln(B_2^{[*]}) x + \\ &2 \ln(A_2^{[*]}) - 2 \ln(A_2) - \ln(B_2^{[*]}) \end{aligned} \right]}, \tag{35}$$

where  $A^{[*]} = \left[ \begin{aligned} &16 \ln(A_2) \ln(2)^3 + 24 \ln(A_2) \ln(2)^2 \ln(B_2) \alpha^2 + \\ &12 \ln(A_2) \ln(2) \ln(B_2)^2 \alpha^2 \\ &+ 2 \ln(A_2) \ln(B_2)^3 \alpha^2 + \\ &8 \ln(A_2) \ln(2)^2 \alpha + 8 \ln(A_2) \ln(2) \ln(B_2) \alpha \\ &+ 2 \ln(A_2) \ln(B_2)^2 \alpha \end{aligned} \right]$ ,  $A_2^{[*]} = e^{\frac{1}{2} \frac{(x^2-1)}{x}}$ ,  $B_2^{[*]} = \frac{x}{(x-1)^2}$ , since it is independent of  $a$  and  $b$ , and maps with 0 or infinity ( $\infty$ ). The iterative map  $R(x)$ , which is based on multiplicative calculus, aligns with the following:

$$R^{[*]}(x) = \frac{\sum_{i=0}^n (a_i x^i)}{\sum_{i=0}^n (a_{n-i} x^i)}; \{a_i\}_{i=0}^n \in \mathbb{R}, \tag{36}$$

possessing intriguing characteristics [42].

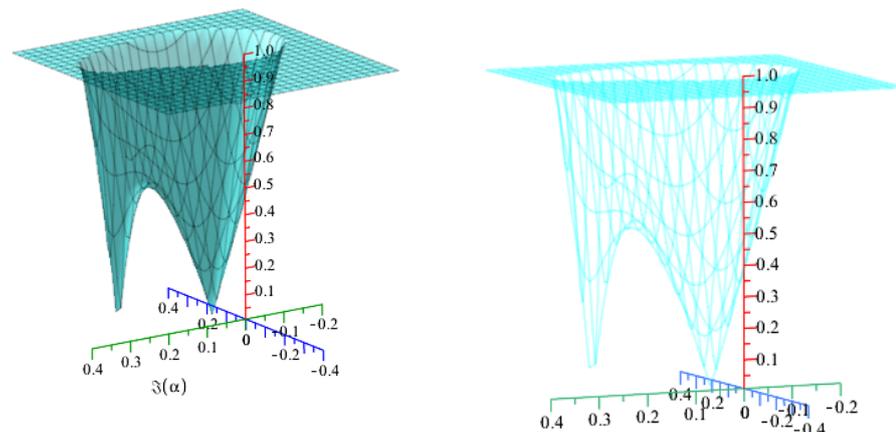
**Proposition 1.** *The iterative map's (35) fixed points are given as follows:*

- $x_0 = 0.00; x_\infty = \infty$  are super-attracting fixed points in the multiplicative-iterative process.
- $x_1 = 1.00$ , is a repulsive point of the multiplicative map.
- The super-attracting and repelling points are 0.0 and 1.0, respectively.

While assessing the behavior of the iterative process, this stability function is vital, especially when examining how resilient it is to errors and interruptions in the initial guesses. The robustness and effectiveness of the multiplicative-calculus-based approach when convergent toward a root can be determined by analyzing the stability function. The stability function for multiplicative single root-finding techniques, denoted as  $CM^{[*]}$ , can be expressed as follows for  $\beta$  values close to 1.00:

$$\left( R^{[*]}(1, \alpha; \beta) \right)' = \left| \frac{\begin{pmatrix} -4 - 5.0 \times 10^{-9}\alpha - 25.13274124i\alpha + \\ 1.6 \times 10^{-8}\alpha^2\beta + 124.0251068i\alpha^2\beta + \\ 39.47841762\alpha\beta - 2.0 \times 10^{-8}i\alpha\beta \end{pmatrix}}{-9.69604405i \times 10^8\alpha + 0.5i\alpha + 3.141592654 \times 10^8i} \right|. \quad (37)$$

The stability areas are analyzed for different values of the parameter  $\beta \in [-10, 10]$ , as seen in Figures 1a,b and 2a–d.



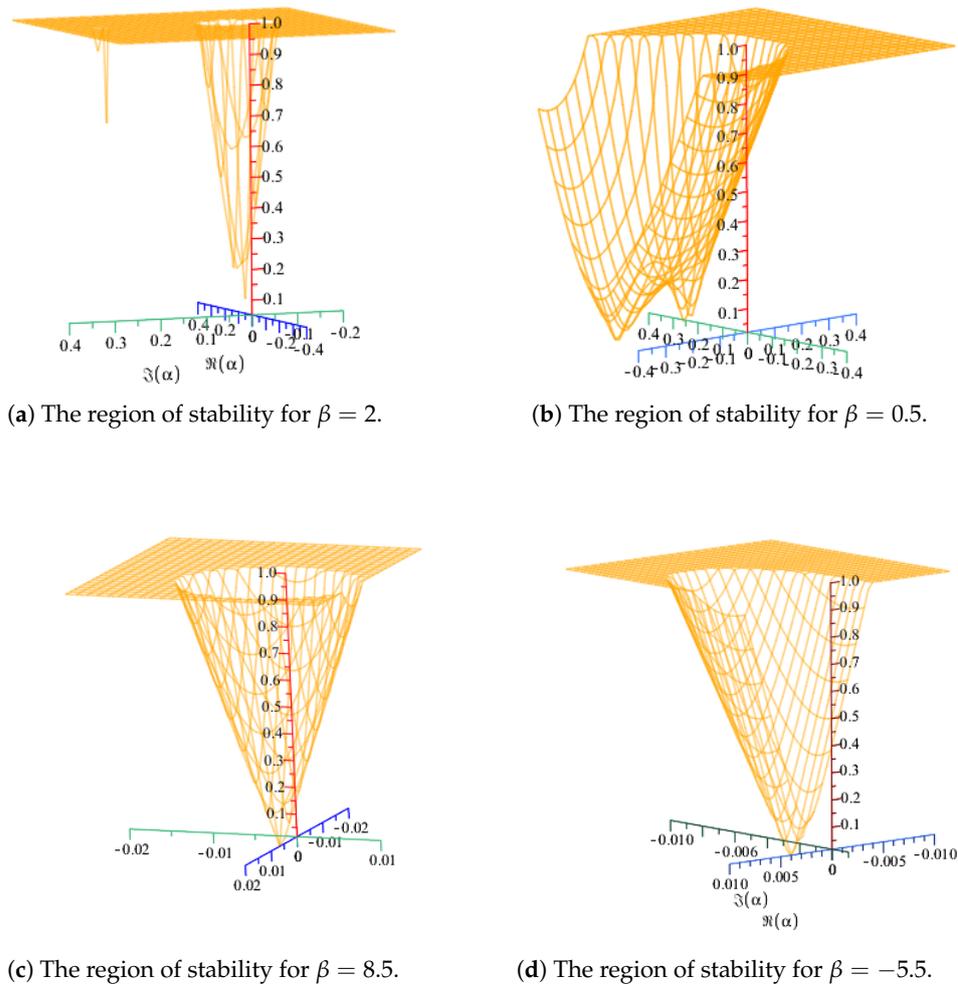
(a) The region of stability for  $\beta = 1.00$ .

(b) The region of stability for  $\beta = 0.995$ .

**Figure 1.** (a,b): The stability zones of the multiplicative simple root-finding method,  $CM^{[*]}$ , for various values of  $\beta$  and  $\alpha$ .

In Figure 1a,b, the stability zone—which displays the starting estimations or parameters for which the multiplicative calculus-based iterative technique converges to the exact solution of (2)—is displayed by the multiplicative-iterative method,  $CM^{[*]}$ , to obtain the root of (1). The multiplicative-iterative technique’s resilience and efficacy in solving (1) are illustrated in Figure 2a–d for a variety of  $\beta$  parameter values. The figure emphasizes the significance of the  $b$  and  $\alpha$  parameters in determining the stability and efficacy of the method. This strategy is most stable when  $\beta = 1$ , and it progressively loses stability as  $\beta$  moves closer to zero. The most significant aspects of the iterative mapping are summarized in Figure 3, which visually depicts the various behaviors of the method’s iterations over dynamic planes, including regions of convergence, deviation, and chaotic patterns. Stable and unstable behaviors for various combinations of  $\beta$  and  $\alpha$  values are shown in Figures 4a–e, 5a–e, and 6a–e. The visual representations illustrate several kinds of crucial points in the multiplicative calculus-based iterative process using different markers. In order to locate fixed points—places where the values stay constant throughout iteration—small circle and white asterisk are used. Super-attracting fixed points, indicated by squares with embedded stars, emphasize regions where convergence speeds up because of the effectively appealing features of the points. The behavior and stability of the iterative process are greatly influenced by essential or critical points, which are marked with plain squares. Analyzing the convergence properties of iterative techniques for solving nonlinear

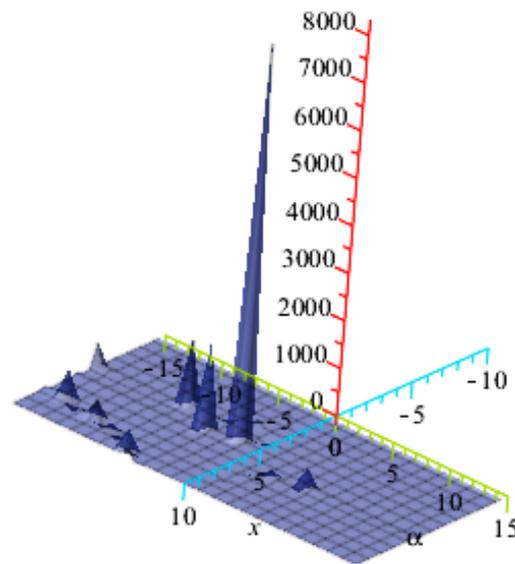
equations requires the use of dynamic planes. These planes assist in visualizing stability and efficiency, aiding in determining the ideal values of the  $\beta$  and  $\alpha$  parameters that improve accuracy and speed of convergence. These planes also display divergence zones, enabling the avoidance of factors that can cause instability or failure. Dynamical plane analysis enables one to determine the optimal parameters for reducing iterations and enhancing solution stability. The multiplicative scheme  $CM^{[*]}$  exhibits consistent behavior for a range of parameter values  $\beta, \alpha$ , as illustrated in Figures 4a–e, 5a–e, and 6a–e, respectively.



**Figure 2.** (a–d): The stability zones of the multiplicative simple root-finding method,  $CM^{[*]}$ , for various values of  $\beta$  and  $\alpha$ .

The statistics also show how sensitive the method is to parameter alterations, helping to determine if the methodology remains resilient or requires precise parameter tweaks for maximum performance. They also disclose the convergence characteristics—whether linear, superlinear, or quadratic—which indicate how quickly the multiplicative calculus-based iterative method approaches the exact solution of (1). It is crucial to comprehend these rates of convergence because they aid in evaluating the method’s effectiveness and speed in producing exact findings. This knowledge is crucial for selecting and adjusting parameters to ensure that the method performs well across a variety of problem settings, thereby enhancing its effectiveness and reliability.

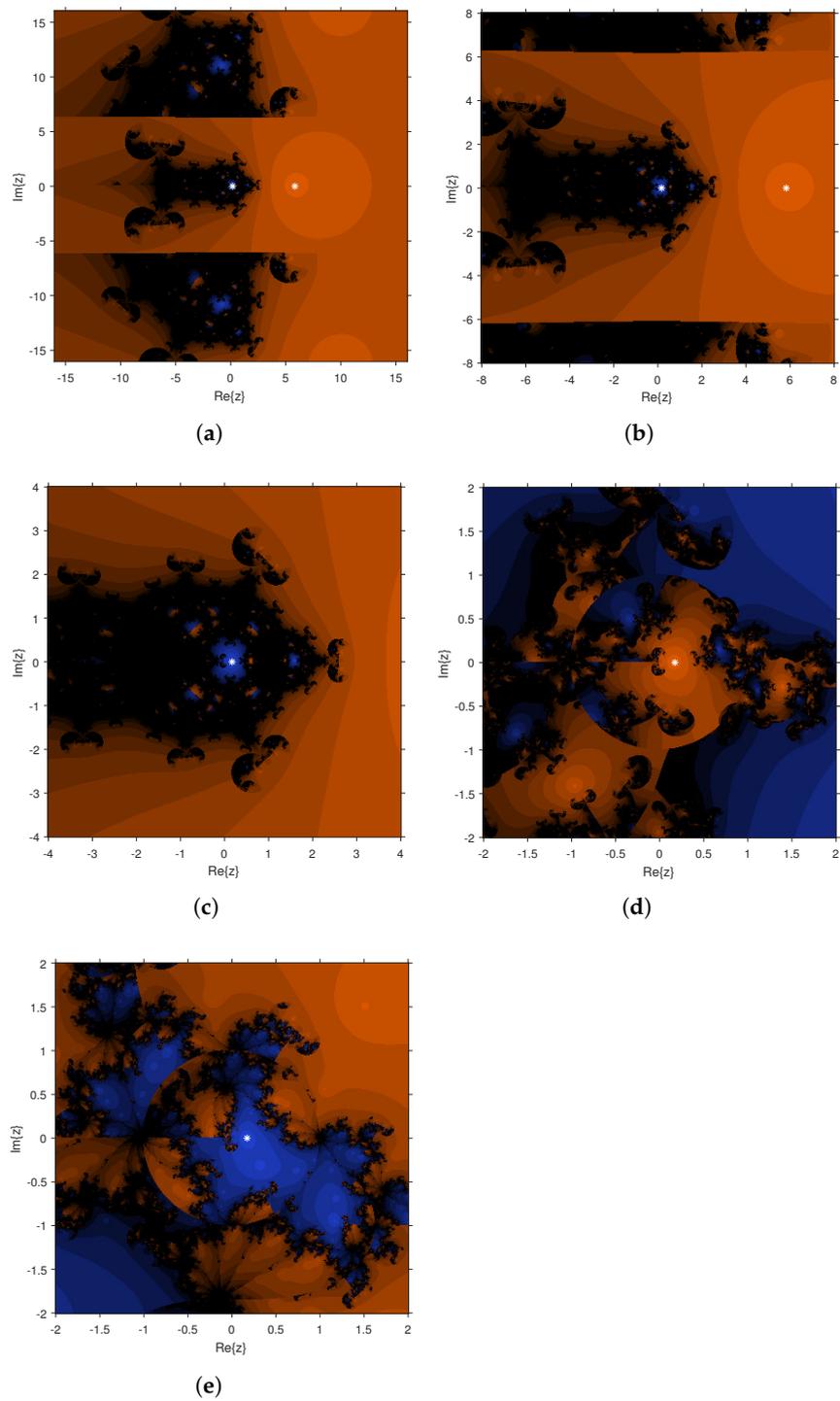
To efficiently solve nonlinear equations like Equation (2), iterative approaches require a thorough grasp of their dynamics. Dynamic analysis can be used to evaluate the impacts of initial estimations on the iterative approach's convergence behavior, rate of solution, and overall stability. In addition to shedding light on the method's reliability, this study enhances its efficiency and lowers the possibility of problems like sluggish convergence or possible divergence. Since dynamic analysis allows practitioners to adapt their methods to the unique characteristics of the problem, it is a fundamental component of applied professions such as engineering and the sciences that use iterative processes to achieve accurate solutions. By carefully analyzing these dynamics, it is possible to enhance convergence rates, modify techniques to fit the properties of the function, and guarantee stable solutions (even in complex or uncertain situations).



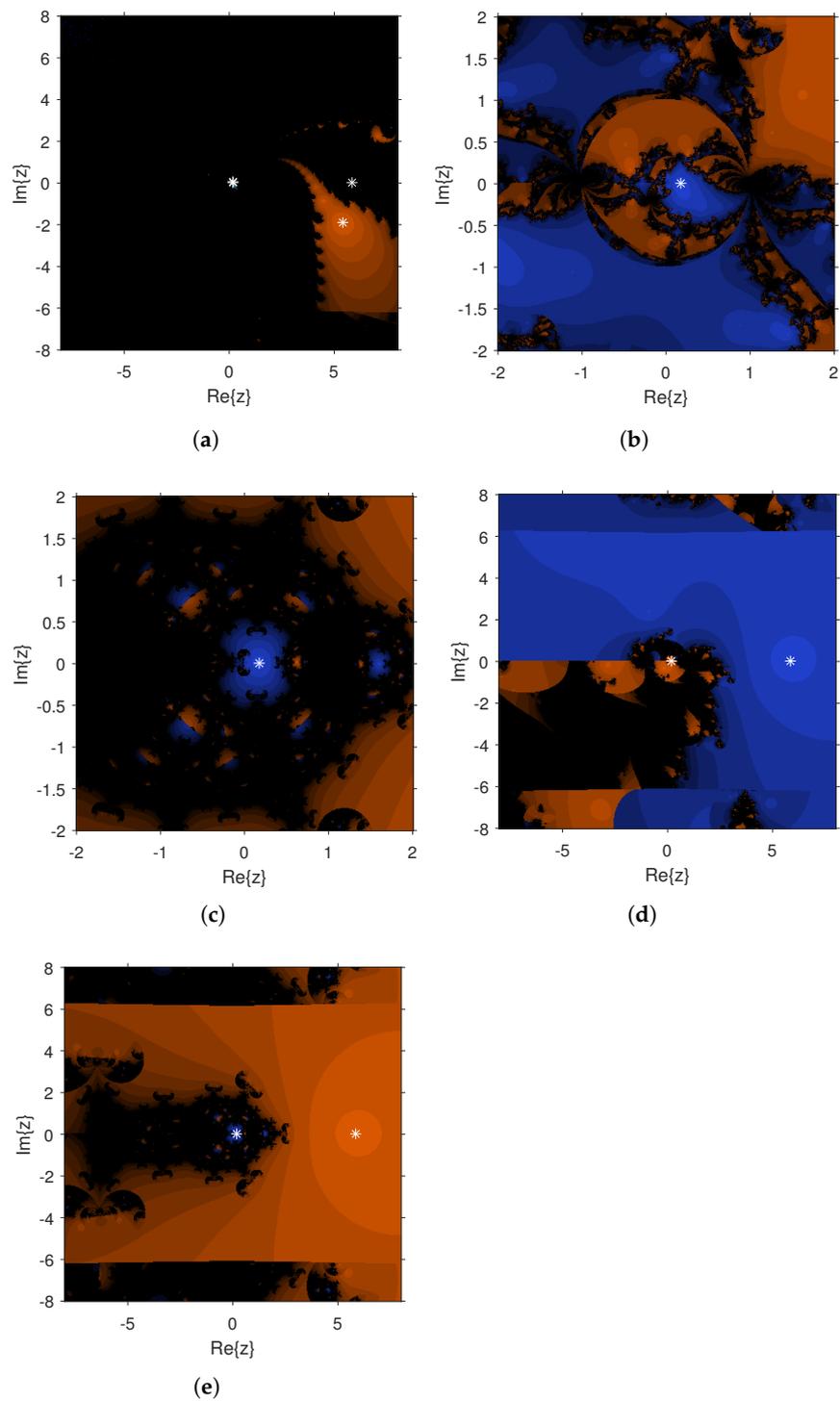
**Figure 3.** The region of the  $CM^{[*]}$ -generated iterative map contains the critical points.

Iterative sequences converge to solutions in regions of attraction, whereas they diverge in regions of repulsion, as depicted by the dynamical planes. These planes also show chaotic behaviors, periodic cycles, and fixed spots, which shed light on how sensitive iterative techniques are to initial conditions. Using iterative algorithms to solve nonlinear equations more robustly and reliably, this kind of study is essential because it provides a deeper knowledge of how specific factors can affect convergence behavior. Unstable results might result from choosing values of  $\beta$  within the divergence zone, namely  $\beta \in (-\infty, -5] \cup [5, \infty)$ , as shown in Figure 7a,b. Understanding these zones makes it possible to choose parameters more effectively, which encourages reliable convergence in real-world applications.

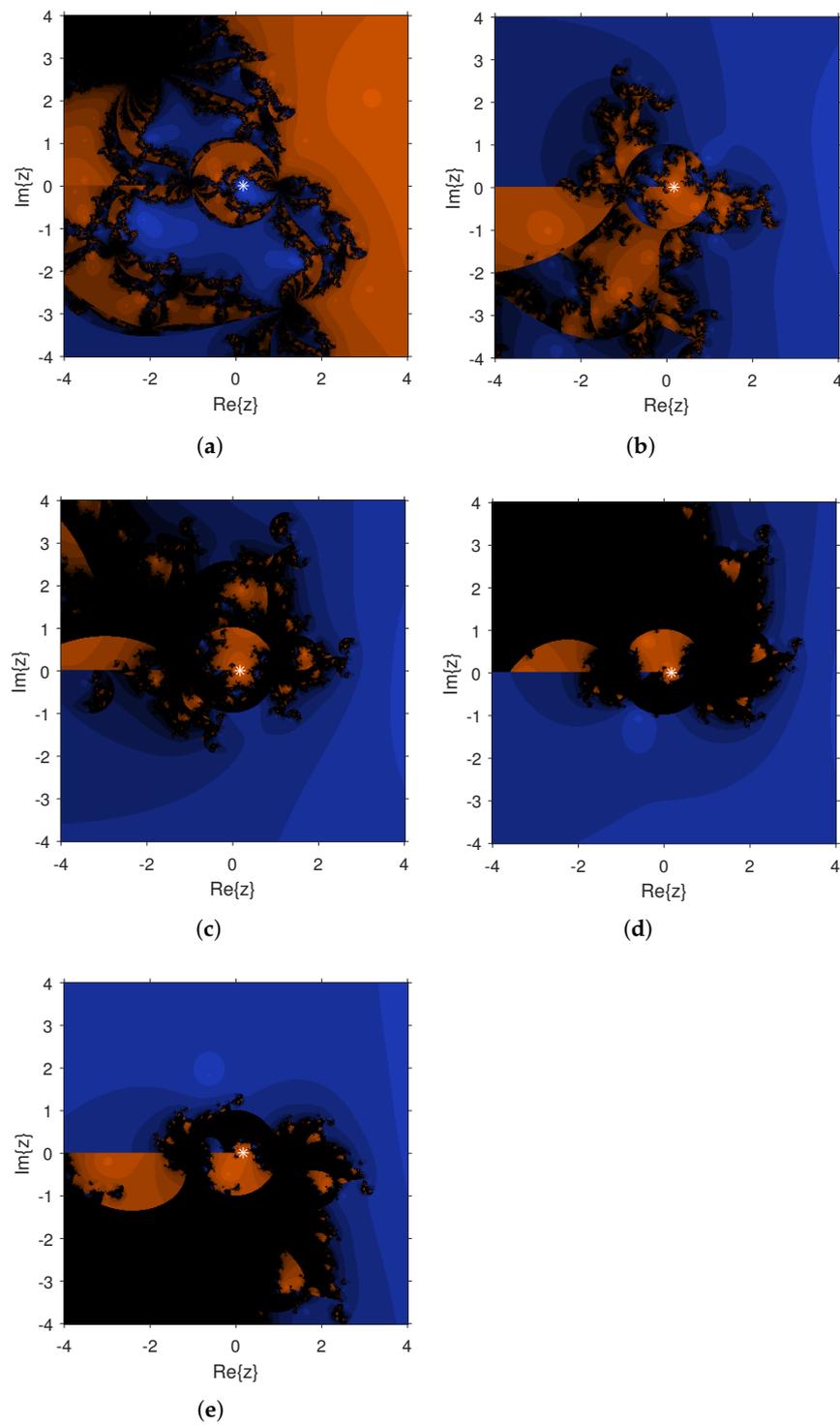
Utilizing multiplicative calculus, the  $CM^{[*]}$  method is a productive way to find simple roots. Using dynamical and stability plane representations, this approach finds the ideal parameter that speeds up convergence. A strong and dependable method for root-finding is demonstrated by  $CM^{[*]}$ , which increases convergence rates. We propose a hybrid parallel approach based on multiplicative calculus that uses  $CM^{[*]}$  as a correction factor. This hybrid approach is intended to analyze Equation (2) with enhanced accuracy, stability, and computational efficiency. It is described in the section that follows. In particular, in complex or large-scale computational scenarios, this novel multiplicative calculus-based parallel technique provides a promising framework for simultaneously finding all solutions to nonlinear engineering models.



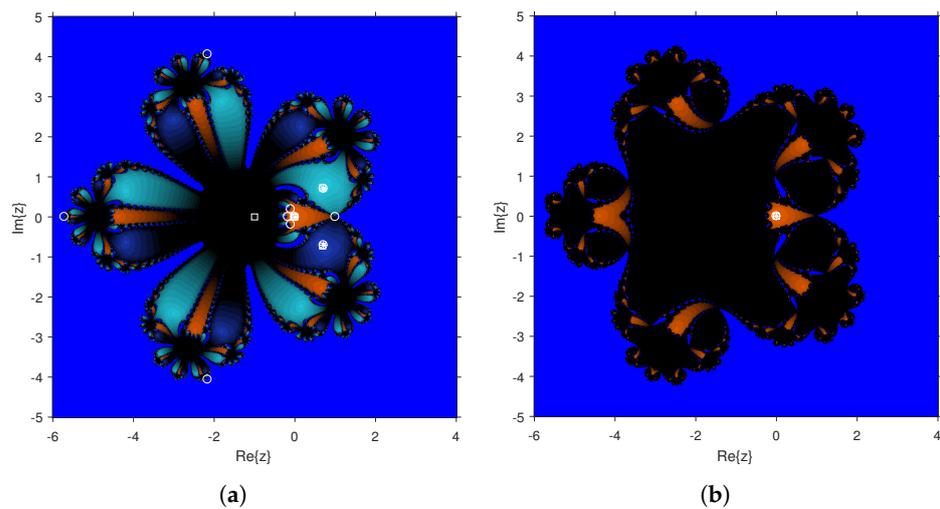
**Figure 4.** (a–e):  $CM^{[*]}$ -generated rational map for solving (68) with dynamical planes for different parameter  $\alpha, \beta$  values: Stable behavior. (a) The dynamic regions for  $\alpha \approx 0.001 - 0.001i$  and  $\beta = 1$ . (b) The dynamic regions for  $\alpha \approx 0.1 + 0.00i$  and  $\beta = 1$ . (c) The dynamic regions for  $\alpha \approx -0.1i$  and  $\beta = 1$ . (d) The dynamic regions for  $\alpha \approx 2.1 - 2.1i$  and  $\beta = 1$ . (e) The dynamic regions for  $\alpha \approx 3.001 + 3.001i$  and  $\beta = 1$ .



**Figure 5.** (a–e):  $CM^{[*]}$ -generated rational map for solving (68) with dynamical planes for different parameter  $\alpha, \beta$  values: Stable behavior. (a) The dynamic regions for  $\alpha \approx 2.1i$  and  $\beta = 2$ . (b) The dynamic regions for  $\alpha \approx 0.1 + 0.1i$  and  $\beta = 2$ . (c) The dynamic regions for  $\alpha \approx 0.001$  and  $\beta = 2$ . (d) The dynamic regions for  $\alpha \approx 0.5i$  and  $\beta = 2$ . (e) The dynamic regions for  $\alpha \approx 0.7$  and  $\beta = 2$ .



**Figure 6.** (a–e):  $CM^{[*]}$ -generated rational map for solving (68) with dynamical planes for different parameter  $\alpha, \beta$  values: Stable behavior. (a) The dynamic regions for  $\alpha \approx 0.1 + 0.1i$  and  $\beta = -0.5$ . (b) The dynamic regions for  $\alpha \approx 0.03 + 0.03i$  and  $\beta = 0.3$ . (c) The dynamic regions for  $\alpha \approx -0.04 - 0.07i$  and  $\beta = 9.5$ . (d) The dynamic regions for  $\alpha \approx -0.01 + 0.09i$  and  $\beta = -3.3i$ . (e) The dynamic regions for  $\alpha \approx 0.001$  and  $\beta = -0.5$ .



**Figure 7.** (a,b):CM<sup>[\*]</sup>-generated rational map for solving (68) with dynamical planes for different parameter  $\alpha, \beta$  values: Unpredictable behavior. (a) The unpredictable behavior for  $\alpha \approx 0.5i$  and  $\beta = 20.9$ . (b) The unpredictable behavior for  $\alpha \approx 10.9$  and  $\beta = -30.5$ .

### 3. Construction and Analysis of the Multiplicative Parallel Method

Parallel computing techniques are robust computational algorithms used to simultaneously locate all feasible solutions to nonlinear equations. Their simplicity is what makes them unique: they utilize computer multiprocessing to compute all solutions in parallel, iteratively improving guesses regardless of their proximity to the precise solutions. This approach is especially useful for high-degree nonlinear equations with complex solutions that require rapid convergence. The multi-threading capability of these techniques makes them an effective tool in computational mathematics for solving nonlinear equations. The Weierstrass–Durand–Kerner method [43], also called the Weierstrass–Dochive method, is one of the best derivative-free parallel methodologies. It is defined as follows:

$$Z_i^{[l+1]} = x_i^{[l]} - \vartheta(x_i^{[l]}), \tag{38}$$

where

$$\vartheta(x_i^{[l]}) = \frac{f(x_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (x_i^{[l]} - x_j^{[l]})}, \quad (i, j = 1, \dots, n), \tag{39}$$

is referred to as Weierstrass’ modification. Local quadratic convergence is illustrated using this parallel approach.

In 1977, Ehrlich [44] suggested a third-order accurate simultaneous iteration approach with the express purpose of effectively locating multiple roots of nonlinear equations. This novel method represented a major breakthrough in the field, as it enabled the refinement of several root estimates simultaneously within a single iterative framework, as follows:

$$Z_i^{[l+1]} = x_i^{[l]} - \frac{1}{\frac{1}{N_i(x_i^{[l]})} - \sum_{\substack{j=1 \\ j \neq i}}^n \left( \frac{1}{(x_i^{[l]} - x_j^{[l]})} \right)}, \tag{40}$$

where  $x_j^{[l]} = u_j^{[l]}$  is used as a correction in (5). In numerical analysis, this approach has since been used as a paradigm to construct other high-order iterative schemes. For example,

Petkovic et al. [45] enhanced the convergence order of the Ehrlich parallel approach from 3 to 6, as follows:

$$Z_i^{[l+1]} = x_i^{[l]} - \frac{1}{\frac{1}{N_i(x_i^{[l]})} - \sum_{\substack{j=1 \\ j \neq i}}^n \left( \frac{1}{(x_i^{[l]} - u_j^{[l]})} \right)}, \tag{41}$$

with the following corrective factor:  $u_j^{[l]} = x_j^{[l]} - \frac{f(s_j^{[l]}) - f(x_j^{[l]})}{2f(s_j^{[l]}) - f(x_j^{[l]})} \frac{f(x_j^{[l]})}{f'(x_j^{[l]})}$  where  $s_j^{[l]} = x_j^{[l]} - \frac{f(x_j^{[l]})}{f'(x_j^{[l]})}$ .

The method’s convergence order was further accelerated from 3 to 10 by Petkovic et al. [46], resulting in the following method (PM<sup>[\*]</sup>):

$$Z_i^{[l+1]} = x_i^{[l]} - \frac{1}{\frac{1}{N_i(x_i^{[l]})} - \sum_{\substack{j=1 \\ j \neq i}}^n \left( \frac{1}{(x_i^{[l]} - T_j^{[l]})} \right)}, \tag{42}$$

where  $T_j^{[l]} = \gamma_j^{[l]} - \frac{(\sigma_j^{[l]} - \gamma_j^{[l]})f(\gamma_j^{[l]}) \left( \frac{f(x_j^{[l]})}{f'(x_j^{[l]})} \right)}{(f(x_j^{[l]}) - f(\gamma_j^{[l]}))^2} \left[ f(\sigma_j^{[l]}) - \frac{f(x_j^{[l]})}{f(\sigma_j^{[l]}) - f(\gamma_j^{[l]})} \right]$ ,  $\gamma_j^{[l]} = \sigma_j^{[l]} - \frac{f(x_j^{[l]})f(\sigma_j^{[l]}) \left( \frac{f(x_j^{[l]})}{f'(x_j^{[l]})} \right)}{(f(x_j^{[l]}) - f(\sigma_j^{[l]}))^2}$ ,  $\sigma_j^{[l]} = x_j^{[l]} - \frac{f(x_j^{[l]})}{f'(x_j^{[l]})}$ .

Consider the well-known single-root-finding scheme of seventh-order convergence presented by Kou et al. [47], given as follows:

$$v^{[l]} = \gamma^{[l]} - \frac{f(\gamma^{[l]})}{f'(x^{[l]})} \left[ \left( \frac{f(x^{[l]}) - f(\sigma^{[l]})}{f(x^{[l]}) - 2f(\sigma^{[l]})} \right)^2 + \frac{f(\gamma^{[l]})}{f(\sigma^{[l]}) - \alpha f(\gamma^{[l]})} \right], \tag{43}$$

where  $\gamma^{[l]} = \sigma^{[l]} - \frac{f(x^{[l]})}{(f(x^{[l]}) - 2f(\sigma^{[l]}))} \frac{f(\sigma^{[l]})}{f'(x^{[l]})}$ ,  $\sigma^{[l]} = x^{[l]} - \frac{f(x^{[l]})}{f'(x^{[l]})}$ . Using (30) in (6), to find all the solutions to nonlinear Equation (1), we constructed a new parallel multiplicative calculus-based iterative scheme (CM<sub>1</sub><sup>[\*]</sup>) as follows:

$$x_i^{[l+1]} = \gamma_i^{[l]} - \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})} \left[ \left( \frac{1 - \varnothing_2^{[*]}}{1 - 2\varnothing_2^{[*]}} \right)^2 + \frac{\varnothing_3^{[*]}}{1 - \alpha \varnothing_3^{[*]}} \right] \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \gamma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right), \tag{44}$$

where  $\gamma_i^{[l]} = \sigma_i^{[l]} - \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})} \frac{1}{(1-2\varnothing_1^{[*]})} \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \sigma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right)$ ,  $\sigma_i^{[l]} = x_i^{[l]} - \frac{f(x_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (x_i^{[l]} - v_j^{[l]})}$ , and

$$v_j^{[l]} = x_j^{[l]} - \frac{\ln(\zeta(x_j^{(l)}))}{\ln(\zeta^{[*]}(x_j^{(l)}))} \left[ \frac{1}{1+\alpha \left( \frac{\ln(\zeta(x_j^{(l)}))}{1+\alpha \ln(\zeta(x_j^{(l)}))} + \beta (\ln(\zeta(x_j^{(l)})))^2 \right)} \right], \varnothing_1^{[*]} = \left( \frac{f(\sigma_i^{[l]})}{f(x_i^{[l]})} \right), \varnothing_2^{[*]} = \left( \frac{f(\gamma_i^{[l]})}{f(x_i^{[l]})} \right),$$

$$\varnothing_3^{[*]} = \left( \frac{f(\gamma_i^{[l]})}{f(\sigma_i^{[l]})} \right).$$

To comprehend the efficiency of parallel numerical iterative algorithms in resolving nonlinear equations, a theoretical convergence analysis is necessary. Scientists can measure the technique’s efficiency and reliability across different function types by examining convergence properties such as iteration stability and order. Faster and more reliable root-finding techniques in practical applications are the outcome of this research’s contribution to algorithm design. Thus, the theoretical convergence analysis of hybrid parallel schemes based on multiplicative calculus will be discussed in the following analysis:

**Theorem 3.** Consider  $\zeta_1, \dots, \zeta_\sigma$  to be single roots of (1). Assume that the values  $x_1^{[0]}, \dots, x_n^{[0]}$  are close enough to the exact solutions. Then, a twelfth-convergence order is attained using the  $CM_1^{[*]}$  parallel approach.

**Proof.** Let  $\epsilon_i = x_i^{[l]} - \zeta_i, \epsilon_\sigma = \sigma_i^{[l]} - \zeta_i, \epsilon_\gamma = \gamma_i^{[l]} - \zeta_i$  and  $\epsilon_i^{[*]} = x_i^{[l+1]} - \zeta_i$  represent the errors in  $x_i^{[l]}, \sigma_i^{[l]}, \gamma_i^{[l]}$  and  $x_i^{[l+1]}$ , respectively. From the first step of  $CM_1^{[*]}$ , we have the following:

$$\sigma_i^{[l]} - \zeta_i = x_i^{[l]} - \zeta_i - \frac{f(x_j^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (x_i^{[l]} - v_j^{[l]})}, \tag{45}$$

$$\epsilon_\sigma = \epsilon_i - \epsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n \left[ \frac{x_i^{[l]} - \zeta_j}{x_i^{[l]} - v_j^{[l]}} \right], \tag{46}$$

$$\epsilon_\sigma = \epsilon_i \left( 1 - \prod_{\substack{j=1 \\ j \neq i}}^n \left[ \frac{x_i^{[l]} - \zeta_j}{x_i^{[l]} - v_j^{[l]}} \right] \right), \tag{47}$$

where  $\prod_{\substack{j=1 \\ j \neq i}}^n \left[ \frac{x_i^{[l]} - \zeta_j}{x_i^{[l]} - v_j^{[l]}} \right] = 1 + \frac{v_j^{[l]} - \zeta_j}{x_i^{[l]} - v_j^{[l]}} = \left( 1 + O([\epsilon_i]^2) \right)^{l+1}$  and, from (23),  $v_j^{[l]} - \zeta_j = O([\epsilon_i]^2)$ . Thus, we have the following:

$$\epsilon_\sigma = \epsilon_i \left( 1 + (l-1)O([\epsilon_i]^2) \right), \tag{48}$$

Assuming  $|\epsilon_i| = |\epsilon_j|$ , we have the following:

$$\epsilon_\sigma = O([\epsilon_i]^3). \tag{49}$$

In the second step, we obtain the following:

$$\gamma_i^{[l]} - \zeta_i = \sigma_i^{[l]} - \zeta_i - \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})} \frac{1}{(1 - 2\varnothing_1^{[*]})} \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \sigma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right), \tag{50}$$

leading to the following:

$$\epsilon_\gamma = \epsilon_\sigma - \frac{\epsilon_\sigma}{\epsilon_\sigma} \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})} \frac{1}{(1 - 2\varnothing_1^{[*]})} \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \sigma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right), \tag{51}$$

where  $\prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \sigma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right) = 1$ , and  $\varnothing_1^{[*]} = \frac{\epsilon_\sigma}{\epsilon_i} \left( 1 + \epsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n \frac{1}{x_i^{[l]} - x_j^{[l]}} \right)$ , thus, we have the following:

$$\epsilon_\gamma = \epsilon_\sigma \left( 1 - \frac{1}{\epsilon_\sigma} \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})} \frac{1}{1 - 2 \left( \frac{\epsilon_\sigma}{\epsilon_i} \left( 1 + \epsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n \frac{1}{x_i^{[l]} - x_j^{[l]}} \right) \right)} \right), \tag{52}$$

$$\epsilon_\gamma = \epsilon_\sigma \left( 1 - \frac{1}{\epsilon_\sigma} \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})} \left( 1 - 2 \left( \frac{\epsilon_\sigma}{\epsilon_i} (1 + \Lambda_1^{[*]}) \right) \right)^{-1} \right), \tag{53}$$

where  $\Lambda_1^{[*]} = \epsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n (x_i^{[l]} - x_j^{[l]})^{-1}$

$$\epsilon_\gamma = \epsilon_\sigma \left( 1 - (1 + O([\epsilon_\sigma])) \left( 1 + 2 \left( \frac{\epsilon_\sigma}{\epsilon_i} (1 + \Lambda_1^{[*]}) \right) + \dots \right) \right), \tag{54}$$

$$\epsilon_\gamma = \epsilon_\sigma \left( (-O([\epsilon_\sigma])) \left( 1 + 2 \left( \frac{\epsilon_\sigma}{\epsilon_i} (1 + \Lambda_1^{[*]}) \right) + \dots \right) \right), \tag{55}$$

$$\epsilon_\gamma = \epsilon_\sigma (O([\epsilon_\sigma])) = O([\epsilon_\sigma]^2) = O([\epsilon_i^3]^2), \tag{56}$$

$$\epsilon_\gamma = O([\epsilon_i]^6). \tag{57}$$

Consider the third step of the multiplicative parallel scheme CM, i.e.,  $\mathbb{1}^{[*]}$

$$x_i^{[l+1]} - \zeta_i = \gamma_i^{[l]} - \zeta_i - \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})} \left[ (\Lambda_2^{[*]})^2 + \Lambda_3^{[*]} \right] \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \gamma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right), \tag{58}$$

where  $\Lambda_2^{[*]} = \left( \frac{1-\varrho_2^{[*]}}{1-2\varrho_2^{[*]}} \right), \Lambda_3^{[*]} = \frac{\varrho_3^{[*]}}{1-\alpha\varrho_3^{[*]}}$ .

$$\epsilon_i^{[*]} = \epsilon_\gamma - \frac{\epsilon_\gamma}{\epsilon_\gamma} \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})} \left[ (\Lambda_2^{[*]})^2 + \Lambda_3^{[*]} \right] \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \gamma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right), \tag{59}$$

where  $\prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \gamma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right) \approx 1, \frac{1-\varrho_2^{[*]}}{1-2\varrho_2^{[*]}} = \frac{\epsilon_i - \epsilon_\sigma \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \zeta_j}{x_i^{[l]} - \zeta_j} \right)}{\epsilon_i - 2\epsilon_\sigma \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \zeta_j}{x_i^{[l]} - \zeta_j} \right)} = \frac{\epsilon_i - \epsilon_\sigma (\sigma_1^{[*]})}{\epsilon_i - 2\epsilon_\sigma (\sigma_1^{[*]})}$  and  $\sigma_1^{[*]} = \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \zeta_j}{x_i^{[l]} - \zeta_j} \right)$ .

Similarly,  $\frac{\varrho_3^{[*]}}{1-\alpha\varrho_3^{[*]}} = \frac{\epsilon_\gamma \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \zeta_j}{\sigma_i^{[l]} - \zeta_j} \right)}{\epsilon_\sigma - \alpha\epsilon_\gamma \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \zeta_j}{\sigma_i^{[l]} - \zeta_j} \right)} = \frac{\epsilon_\gamma (\sigma_2^{[*]})}{\epsilon_\sigma - \alpha\epsilon_\gamma (\sigma_2^{[*]})}$  and  $\sigma_2^{[*]} = \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \zeta_j}{\sigma_i^{[l]} - \zeta_j} \right)$ . Using

these values in (59), we have the following:

$$\epsilon_i^{[*]} = \epsilon_\gamma - \frac{\epsilon_\gamma}{\epsilon_\gamma} \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})} \left[ \left( \frac{\epsilon_i - \epsilon_\alpha (\sigma_1^{[*]})}{\epsilon_i - 2\epsilon_\alpha (\sigma_1^{[*]})} \right)^2 + \frac{\epsilon_\gamma (\sigma_2^{[*]})}{\epsilon_\sigma - \alpha\epsilon_\gamma (\sigma_2^{[*]})} \right], \tag{60}$$

$$\epsilon_i^{[*]} = \epsilon_\gamma \left( 1 - \frac{1}{\epsilon_\gamma} \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})} \left( 1 + 2(\Lambda_4^{[*]}) + \dots - \frac{1}{\alpha} + \frac{\epsilon_\sigma}{\epsilon_\sigma - \alpha\epsilon_\gamma (\sigma_2^{[*]})} \right) \right), \tag{61}$$

where  $\Lambda_4^{[*]} = \frac{\epsilon_i - \epsilon_\alpha (\sigma_1^{[*]})}{\epsilon_i - 2\epsilon_\alpha (\sigma_1^{[*]})}$ ,

$$\epsilon_i^{[*]} = \epsilon_\gamma \left( 1 - (1 - O([\epsilon_\gamma])) \left( \frac{1 + 2(\Lambda_4^{[*]}) + \dots - \frac{\epsilon_\sigma}{\alpha}}{\epsilon_\sigma - \alpha\epsilon_\gamma (\sigma_2^{[*]})} \right) \right), \tag{62}$$

$$\epsilon_i^{[*]} = \epsilon_\gamma \left( O([\epsilon_\gamma]) \left( \frac{1 + 2(\Lambda_4^{[*]}) + \dots}{-\frac{\epsilon_\sigma}{\alpha} + \frac{\epsilon_\sigma}{\alpha} (\epsilon_\sigma + \epsilon_\gamma (\sigma_2^{[*]}) + \dots)} \right) \right), \tag{63}$$

$$\epsilon_i^{[*]} = \epsilon_\gamma O([\epsilon_\gamma]) = O([\epsilon_\gamma])^2 = O([\epsilon_i]^6)^2, \tag{64}$$

$$\epsilon_i^{[*]} = O([\epsilon_i]^{12}). \tag{65}$$

This completes the proof. □

#### 4. Numerical Results

Computational results show the effectiveness of the suggested methods and their ability to solve (2) for the hybrid multiplicative calculus-based parallel scheme. This validates the accuracy of the algorithm, compares the findings to existing methods, and offers quantitative evidence to support theoretical assumptions. Convergence rates, data processing efficiency, and error analysis are among the numerical results of parallel schemes that are crucial for assessing the overall performance and impact of the study (2). This

section uses various types of engineering problems to demonstrate the efficiency and stability of the suggested solution. Its performance in the Maple 18 environment is assessed according to particular computer termination criteria. These requirements, which provide accurate findings and allow for trustworthy comparisons of convergence behavior across applications, illustrate the method’s stability and suitability for complicated engineering computations. Using the following criteria, these engineering examples further support the parallel multiplicative calculus-based method’s ability to solve nonlinear equations, highlighting its potential as a reliable tool for engineering analysis:

$$(i) \quad e_i^{[l]} = \left\| x_i^{[l+1]} - x_i^{[l]} \right\| < \epsilon = 10^{-18}, \tag{66}$$

where  $e_i^{[l]}$  is the absolute error norm-2. Using the parallel numerical scheme (SM<sub>1</sub><sup>[\*]</sup>) described in [48], we compare our newly developed multiplicative calculus-based algorithm with the PM<sup>[\*]</sup> method and the parallel iterative computer algorithm presented in [49].

$$x_i^{[l+1]} = \gamma_i^{[l]} - \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})},$$

where  $\gamma_i^{[l]} = \sigma_i^{[l]} - \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})}$  and  $\sigma_i^{[l]} = x_i^{[l]} - \frac{f(x_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (x_i^{[l]} - v_j^{[l]})}$ ,  $v_j^{[l]} = x_j^{[l]} - \frac{f(x_j^{[l]})}{f(x_j^{[l]} + f(x_j^{[l]})) - f(x_j^{[l]})}$ ,

and with [50]

$$x_i^{[l+1]} = \sigma_i^{[l]} - \frac{\phi_i}{\phi_i \frac{f'(\sigma_i^{[l]})}{f(\sigma_i^{[l]})} - \sum_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\phi_j}{(x_i^{[l]} - x_j^{[l]})} \right)},$$

where  $x_i^{[l+1]} = x_i^{[l]} - \frac{\phi_i}{\phi_i \frac{f'(x_i^{[l]})}{f(x_i^{[l]})} - \sum_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\phi_j}{(x_i^{[l]} - z_j^{[l]})} \right)}$ ,  $z_j^{[l]} = \sigma_j^{[l]} - \phi_j \frac{f(\sigma_j^{[l]})}{f'(\sigma_j^{[l]})}$ ,  $\sigma_j^{[l]} = x_j^{[l]} - \sqrt{\phi_j} \frac{f(x_j^{[l]})}{f'(x_j^{[l]})}$

(abbreviated as SM<sub>2</sub><sup>[\*]</sup>). All solutions to the nonlinear Equation (1) are simultaneously computed using Algorithm 1.

**Algorithm 1** For the fractional numerical scheme  $CM_1^{[*]}$ .

Step 1: Given initial values  $x_i^{[0]} (i = 1, \dots, N)$ ,  
 tolerance  $\epsilon > 0$  and set  $l = 0$  for iterations  $qq$

Step 2: Compute the inner and outer loop

Compute: 
$$v_j^{[l]} = x_j^{[l]} - \frac{\ln(\zeta(x_j^{[l]}))}{\ln(\zeta^{[*]}(x_j^{[l]}))} \left[ \frac{1}{1 + \alpha \left( \frac{\ln(\zeta(x_j^{[l]})}{1 + \alpha \ln(\zeta(x_j^{[l]})} + \beta (\ln(\zeta(x_j^{[l]})))^2 \right)} \right].$$

For  $j$  from 0 to  $n$  do  
 for  $i \neq j$

Update 
$$\sigma_i^{[l]} = x_i^{[l]} - \frac{f(x_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (x_i^{[l]} - v_j^{[l]})}$$

$$\gamma_i^{[l]} = \sigma_i^{[l]} - \left( \frac{f(\sigma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\sigma_i^{[l]} - \sigma_j^{[l]})} \right) \left( \frac{1}{(1 - 2\varnothing_1^{[*]})} \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\sigma_i^{[l]} - \sigma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right) \right)$$

$$x_i^{[l+1]} = \gamma_i^{[l]} - \frac{f(\gamma_i^{[l]})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\gamma_i^{[l]} - \gamma_j^{[l]})} \left( \left[ \left( \frac{1 - \varnothing_2^{[*]}}{1 - 2\varnothing_2^{[*]}} \right)^2 + \frac{\varnothing_3^{[*]}}{1 - \alpha \varnothing_3^{[*]}} \right] \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\gamma_i^{[l]} - \gamma_j^{[l]}}{x_i^{[l]} - x_j^{[l]}} \right) \right],$$

End do.

Step 3:  
 $x_i^{[1+1]} = x_i^{[l]} (i = 1, \dots, n).$

if  $e_i^{[l]} = \|x_i^{[1+1]} - x_i^{[1]}\| < 10^{-32} < \epsilon = 10^{-30}$  or  $\sigma > pp$ , then stop.

Step 4:  
 Set  $l = l + 1$  and then move to step 2.  
 End do.  
 End do.

4.1. Example 1: [51]

Modeling complicated engineering systems requires the use of fractional differential equations, particularly in situations like the suspension design of an automobile, where memory consequences and dampening significantly impact system performance. In contrast to conventional integer-order differential equations, fractional-order models can represent the subtleties of hereditary traits, which are prevalent in systems and materials that entail energy dissipation or display viscoelasticity, such as suspension mechanisms in automobiles. In these applications, fractional differential equations enable the creation of suspension models that improve both performance and stability, providing a more precise framework for addressing difficulties in automotive engineering and control systems. Consequently, these equations are crucial instruments for creating complex models that take into consideration the time-dependent, non-local dynamics inherent in these systems,

resulting in formulations that can faithfully capture behaviors found in the actual world. The fractional differential equation that results from this modeling technique is as follows:

$$\begin{cases} \frac{d^4 f(x)}{dx^4} + \frac{d^{\frac{n-1}{2}} f(x)}{dx^{\frac{n-1}{2}}} + (f(x))^{n-1} = x^{n+5}, \alpha_{01}^{[*]} \leq x \leq \alpha_1^{[*]} \\ f(\alpha_{01}^{[*]}) = \frac{d^{n-3} f(\alpha_{01}^{[*]})}{dx^{n-3}} = \frac{d^{n-2} f(\alpha_{01}^{[*]})}{dx^{n-2}} = 0.00, \frac{d^{n-1} f(\alpha_{01}^{[*]})}{dx^{n-1}} = 6.00, \end{cases} \tag{67}$$

where  $\alpha_{01}^{[*]} = 0, \alpha_1^{[*]} = 1$  and  $n = 4$ . Using the analytical approach from [52], we simulate Equation (67) utilizing the following non-linear multiplicative estimations as follows:

$$\zeta_5(x) \approx -1 + x - 2x^2 + \frac{14x^3}{3} - \frac{35x^4}{3} + \frac{91}{3}x^5. \tag{68}$$

Equation (52) has five solutions:

$$\begin{aligned} \zeta_1 &= 0.0, \\ \zeta_{2,3} &= -0.152018637 \pm 0.03931364228i, \\ \zeta_{4,5} &= 0.314432633 \pm 0.25883527180i. \end{aligned}$$

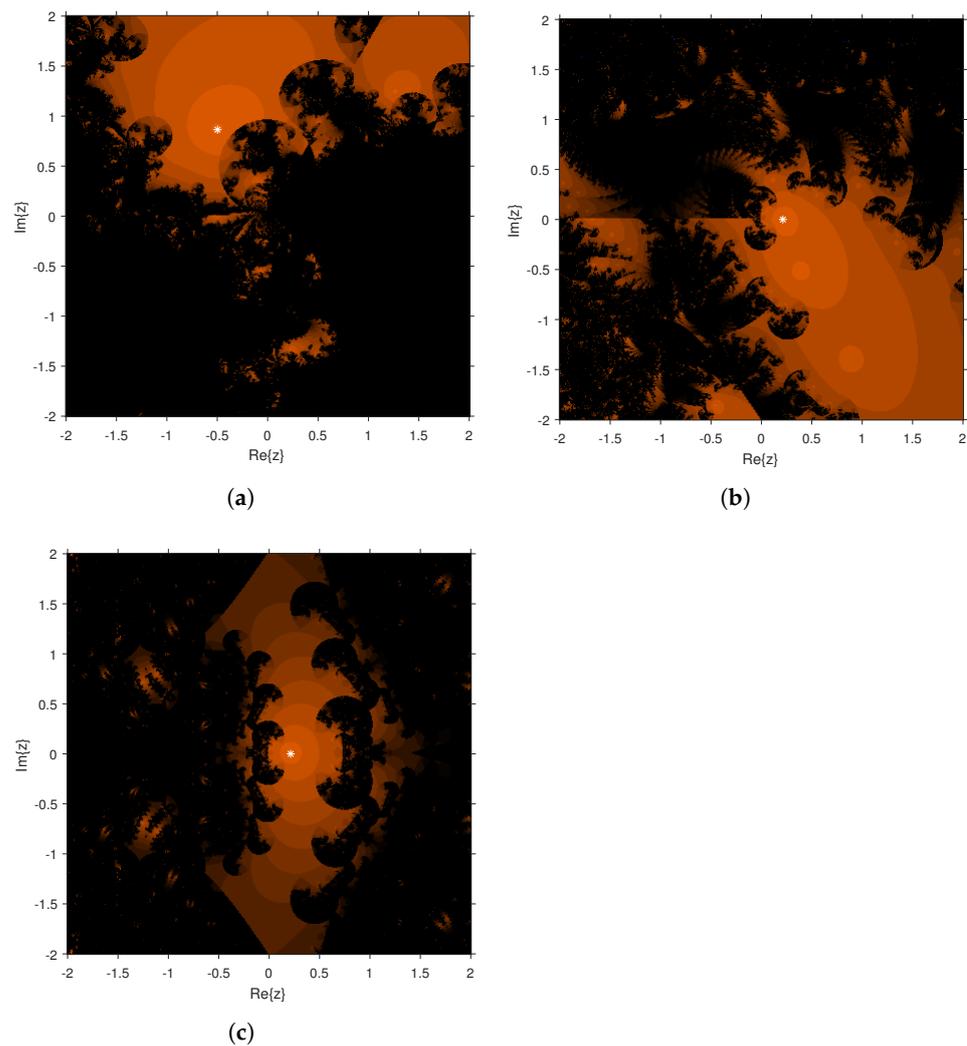
Equation (54) yields an exact solution of zero. We performed a detailed dynamical analysis to investigate the efficiency and behavior of the simple solution-finding multiplicative calculus-based technique used to solve (68). This investigation recorded numerous performance measures, such as elapsed computing time, convergence rate (represented as a percentage), total convergent points assessed, and the number of iterations necessary for convergence across varying values of the  $\alpha$  and  $\beta$  parameters. This evaluation’s findings are compiled in Table 1, and Figure 8a–c provides graphical representations that demonstrate how these factors affect the algorithm’s convergence properties.

The optimal parameter values found by dynamical analysis and the data shown in Table 1 are used in parallel computational approaches to solve nonlinear equations. The convergence rate of the simultaneous techniques is improved by initializing estimates that are close to the exact solution (within an error range of  $\epsilon = 10^{-2}$ ). This method accelerates convergence to exact roots and minimizes the number of iterations required, making the method more efficient and effective for solving nonlinear equations.

**Table 1.** The dynamic outputs of CM<sup>[\*]</sup> for solving (68).

$\alpha$	$\beta$	Iterations	Total Points	Converging Points	Elapsed Points
0.1 + 0.1i	1.0	21	640,000	54.9854%	2.14584165
0.02 + 0.02i	1.5	22	640,000	59.3254%	3.65483535
−0.05	0.05	27	640,000	58.5412%	5.36528456

The rates of convergence, convergence order, and residual errors for the different approaches are clearly compared in Table 2. With all of these metrics, it is evident that the proposed strategy outperforms existing approaches when the parameters are set at  $\beta = 0.05$  and  $\alpha \approx 0.01$ . Specifically, the rate of convergence and the convergence order are greatly improved, and the residual errors for every root are greatly reduced. Since the method provides a more reliable and consistent solution mechanism, this demonstrates its potential to solve nonlinear equations more accurately and with less computing time.



**Figure 8.** (a–c):  $CM^{[*]}$ -generated rational map for solving (68) with dynamical planes for different parameter  $\alpha, \beta$  values. (a) The dynamic regions for  $\alpha \approx 1.9i$  and  $\beta = 0.5$ . (b) The dynamic regions for  $\alpha \approx 0.08 - 1.0i$  and  $\beta = 1.5$ . (c) The dynamic regions for  $\alpha \approx 5.1$  and  $\beta = 1.1$ .

**Table 2.** The numerical results of parallel computer techniques for solving (68).

Err	$PM^{[*]}$	$SM_1^{[*]}$	$SM_2^{[*]}$	$CM_1^{[*]}$
$e_1^{[5]}$	$9.31 \times 10^{-45}$	$1.11 \times 10^{-99}$	$1.1 \times 10^{-104}$	0.0
$e_2^{[5]}$	$2.09 \times 10^{-55}$	$3.02 \times 10^{-76}$	0.0	0.0
$e_3^{[5]}$	$0.17 \times 10^{-76}$	$0.12 \times 10^{-85}$	0.0	0.0
$e_4^{[5]}$	$0.07 \times 10^{-39}$	$6.27 \times 10^{-101}$	$3.95 \times 10^{-165}$	$2.0 \times 10^{-285}$
$e_4^{[5]}$	$1.07 \times 10^{-39}$	$6.27 \times 10^{-101}$	$6.93 \times 10^{-165}$	0.0
CPU	0.0045	0.0063	0.0023	0.0011

During the simulation, P-convergence represents the percent convergence, P-divergence represents the percentage divergence, A-iteration represents the average number of iterations, T-function represents the total number of functions and derivatives per iteration, Operations-it indicates how many mathematical operations are performed per iteration, Max-Error represents the maximum error per iteration, and Local-COC indicates local convergence. Table 3 clearly illustrates that, in terms of consistency and stability, our method outperforms existing methods in the literature, i.e.,  $PM^{[*]}, SM_1^{[*]} - SM_2^{[*]}$ , respectively.

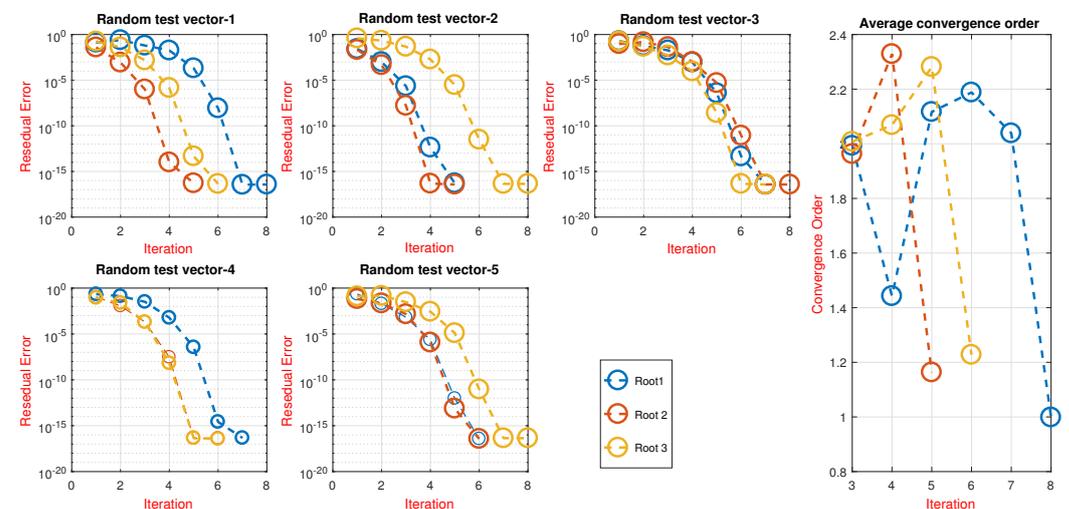
**Table 3.** Parallel scheme analysis for closed starting values leading to an exact solution.

Err	PM <sup>[*]</sup>	SM <sub>1</sub> <sup>[*]</sup>	SM <sub>2</sub> <sup>[*]</sup>	CM <sub>1</sub> <sup>[*]</sup>
P-convergence	93%	97	99	100
P-divergence	7.0	3.0	1.0	0.0
A-Iterations	07	5	5	3
T-functions	31	21	20	16
Operations-it	78	63	60	48
Max-Error	$0.07 \times 10^{-39}$	$6.27 \times 10^{-101}$	$4.9 \times 10^{-165}$	0.0
Local-COC	8.0045	10.0063	11.0023	12.0011

Table 4 clearly illustrates that by using the parameter values  $\alpha, \beta$  obtained by formalizing the concept of the dynamic system and selecting random initial values (Table A1), the newly developed schemes CM<sub>1</sub><sup>[\*]</sup> show consistent behavior and are more stable. The scheme’s global convergence property, CM<sub>1</sub><sup>[\*]</sup>, outperforms other schemes (SM<sub>1</sub><sup>[\*]</sup> – SM<sub>2</sub><sup>[\*]</sup> and PM<sup>[\*]</sup>) in terms of efficiency, computation time, iterations, and residual error (Figure 9).

**Table 4.** Outcomes of the multiplicative scheme CM<sub>1</sub><sup>[\*]</sup> for different initial test vectors for solving (68).

Ran-Test	Iteration	$\ \tau_{i+1} - \tau_i\ $	CPU
1	36	$0.2543 \times 10^{-15}$	0.097
2	24	$1.4433 \times 10^{-17}$	0.096
3	22	$5.7343 \times 10^{-26}$	0.076
4	12	$0.5433 \times 10^{-14}$	0.066
5	06	$2.6453 \times 10^{-16}$	0.001



**Figure 9.** The error graph of the multiplicative calculus-based parallel scheme CM<sub>1</sub><sup>[\*]</sup> for solving (68) using random test vectors.

4.2. Example 2: [53]

Fractional calculus was established to improve the modeling capabilities of classical differential equations and depict complicated physical systems more accurately. This method captures the complex dynamics seen in systems such as fluid mechanics, electrical circuits, and elastic substances by extending traditional calculus and adding derivatives and integrals of arbitrary, non-integer order. A more flexible framework for explaining phenomena like irregular wave-spreading and dissemination—which are frequently found in disciplines like microbiology and substance science—is provided through fractional calculus. Fractional calculus enhances prediction accuracy in these domains by allowing

for these improved descriptions, resulting in the formulation of the fractional initial value problem shown below [40]:

$$\begin{cases} \frac{d^{\frac{5}{2}}f(x)}{dx^{\frac{5}{2}}} + x \frac{d^{\frac{19}{10}}f(x)}{dx^{\frac{19}{10}}} + (f(x))^3 = h^*(x), 0 \leq x \leq 1, \\ f(0) = \frac{df(0)}{dx} = \frac{d^2f(0)}{dx^2} = 0. \end{cases} \tag{69}$$

where  $h^*(x) = 6.770275002x^{0.5} + 5.733474578x^{2.1} + x^3 + x$ . The following non-linear simulation is used to simulate (69) using the technique from [54]:

$$\zeta_3(x) \approx x^3 + 3x^2 + 4x + 0.00000105 - 1. \tag{70}$$

Equation (70) has the following solutions:

$$\begin{aligned} \zeta_1 &= -0.000002625005168, \\ \zeta_{2,3} &= -1.49998687 \pm 1.322874167i. \end{aligned}$$

The effectiveness of this root-finding technique is assessed through a thorough dynamical analysis that takes into account variables like the number of iterations, total number of calculated points, convergence rate, and elapsed time over a range of values of the  $\beta$  and  $\alpha$  parameters. Table 5 presents these indicators, which provide insight into how various parameter choices impact the algorithm’s effectiveness and efficiency in reaching the root. To understand the method’s robustness and computational behavior under different conditions, this assessment is vital.

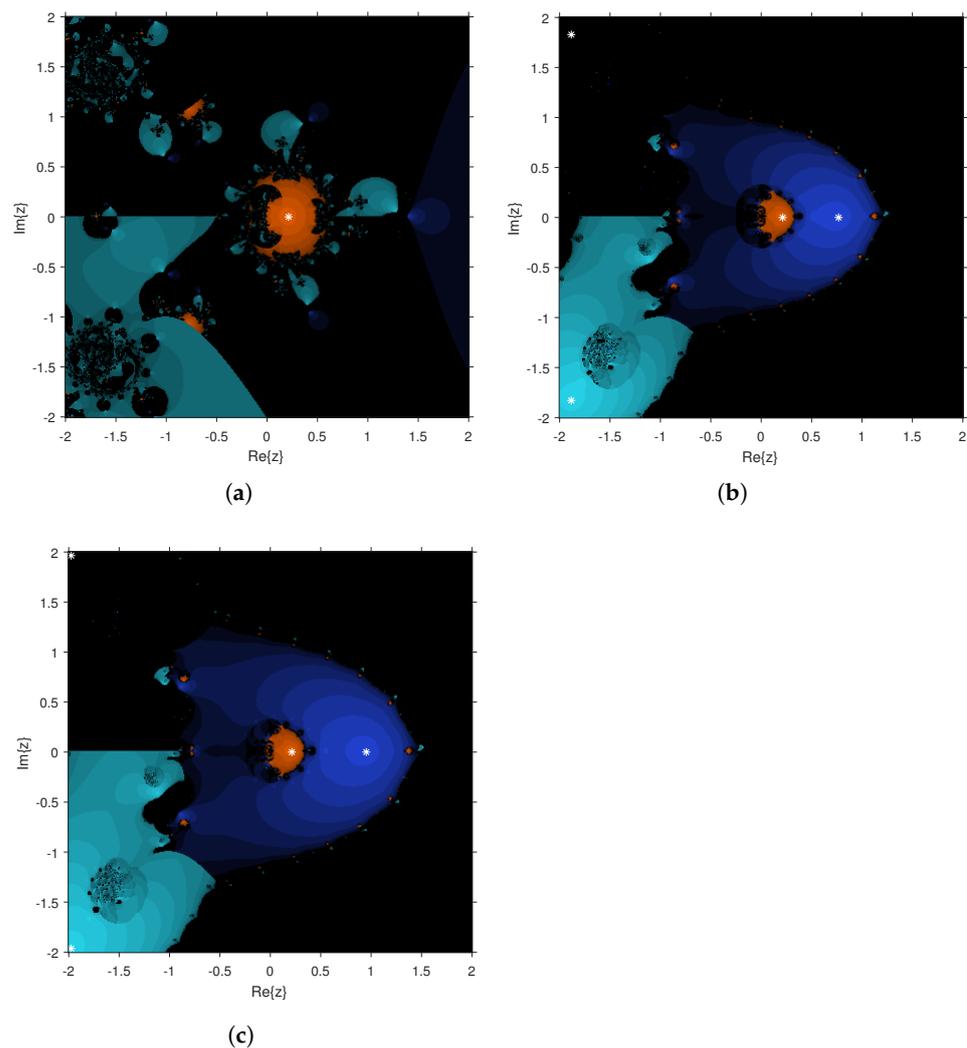
**Table 5.** The dynamic outputs of CM[\*] for solving (70).

$\alpha$	$\beta$	Iterations	Total Points	Converging Points	Elapsed Points
0.1 + 0.1i	1.0	21	640,000	54.545%	1.33464564
0.02 + 0.02i	1.5	18	640,000	59.5412%	4.67454366
−0.05	1.00	20	640,000	58.2541%	5.233554675

The convergence rate of simultaneous iterative algorithms is greatly increased by using starting guesses that are sufficiently close to the right solution (within a tolerance of  $\epsilon = 10^{-2}$ ). This proximity allows the techniques to arrive at accurate solutions more quickly, usually in fewer iterations. Figure 10a–c and Table 6 present the outcomes of using randomly selected initial estimations, highlighting the method’s adaptability and efficiency at various starting points. These results validate the versatility of the method by showing flexibility and excellent accuracy, irrespective of initial conditions.

**Table 6.** The numerical results of parallel computer techniques for solving (70).

Err	PM[*]	SM <sub>1</sub> [*]	SM <sub>2</sub> [*]	CM <sub>1</sub> [*]
$e_1^{[3]}$	$0.23 \times 10^{-53}$	$1.1 \times 10^{-51}$	0.0	0.0
$e_2^{[3]}$	$1.44 \times 10^{-44}$	0.0	0.0	0.0
$e_3^{[3]}$	$7.43 \times 10^{-45}$	$5.01 \times 10^{-57}$	$5.03 \times 10^{-65}$	$5.34 \times 10^{-65}$
$e_4^{[3]}$	$2.87 \times 10^{-45}$	$1.02 \times 10^{-57}$	$1.02 \times 10^{-65}$	$5.06 \times 10^{-65}$
$e_5^{[3]}$	$0.76 \times 10^{-45}$	$5.02 \times 10^{-57}$	$3.02 \times 10^{-65}$	$5.97 \times 10^{-65}$
$\sigma_i^{[n-1]}$	0.0036	0.0023	0.0011	0.0011



**Figure 10.** (a–c):  $CM^{[*]}$ -generated rational map for solving (70) with dynamical planes for different parameter  $\alpha, \beta$  values. (a) The dynamic regions for  $\gamma \approx 3.4 - 3.4i$  and  $\beta = 1.1i$ . (b) The dynamic regions for  $\gamma \approx 6.9$  and  $\beta = 2.5$ . (c) The dynamic regions for  $\alpha \approx 4.9$  and  $\beta = 6.3$ .

The information shows that each root’s residual error, convergence order, and rate of convergence significantly surpasses conventional parallel methods for the range of  $\alpha$  and  $\beta$  values (Table 5). This improved performance illustrates the long-term reliability as well as efficiency of the proposed parallel methodology based on multiplicative calculus. Statistics show that the newly created computer approach has reduced residual errors and faster convergence rates across a wide range of parameter values, indicating that it can estimate solutions more accurately and quickly. These results indicate that the parallel iterative method is a good alternative to existing methods and shows promise for a wide range of nonlinear equation-solving applications where stability and fast convergence are essential.

During the simulation, P-convergence represents the percentage convergence, P-divergence represents the percentage divergence, A-iteration represents the average number of iterations, T-function represents the total number of functions and derivatives per iteration, Operations-it indicates how many mathematical operations are performed per iteration, Max-error represents the maximum error per iteration, and Local-COC indicates local convergence. Table 7 clearly illustrates that, in terms of consistency and stability, our newly developed parallel multiplicative calculus-based technique outperforms existing methods,  $PM^{[*]}$ ,  $SM_1^{[*]} - SM_2^{[*]}$ .

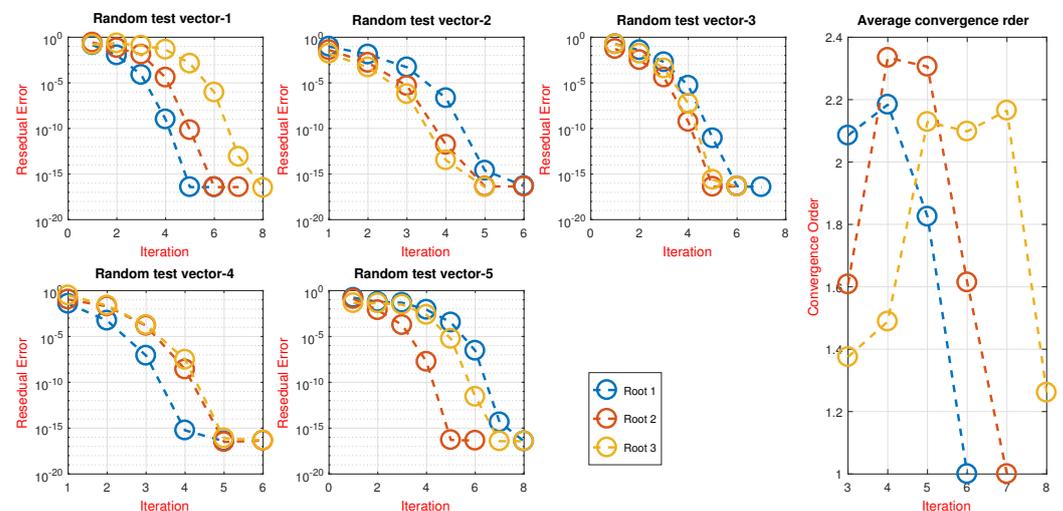
**Table 7.** Parallel scheme analysis for closed starting values yielding an exact solution.

Err	PM <sup>[*]</sup>	SM <sub>1</sub> <sup>[*]</sup>	SM <sub>2</sub> <sup>[*]</sup>	CM <sub>1</sub> <sup>[*]</sup>
P-convergence	93%	97%	99%	100%
P-divergence	7.0%	3.0%	1.0%	0.0%
A-Iterations	07	5	5	3
T-functions	31	21	20	16
Operations-it	78	63	60	48
Max-Error	$0.07 \times 10^{-39}$	$6.27 \times 10^{-101}$	$4.9 \times 10^{-165}$	0.0
Local-COC	8.0045	10.0063	11.0023	12.0011

Table 8, clearly illustrates that by using the parameter values  $\alpha, \beta$  obtained by formalizing the concept of the dynamic system and selecting random initial values (Table A2), the newly developed schemes CM<sub>1</sub><sup>[\*]</sup> reveal consistent behavior and are more stable. This illustrates the scheme’s CM<sub>1</sub><sup>[\*]</sup> global convergence property in terms of effectiveness, processing time, number of repetitions, and absolute residual error when compared to other existing schemes SM<sub>1</sub><sup>[\*]</sup> – SM<sub>2</sub><sup>[\*]</sup> and PM<sup>[\*]</sup> (Figure 11).

**Table 8.** Results of the multiplicative scheme CM<sub>1</sub><sup>[\*]</sup> for different initial test vectors for (70).

Ran-Test	<i>n</i>	$\ \tau_{i+1} - \tau_i\ $	CPU
1	05	$0.37567 \times 10^{-4}$	0.09732
2	05	$1.54567 \times 10^{-10}$	0.08634
3	06	$0.75635 \times 10^{-11}$	0.08823
4	06	$7.46535 \times 10^{-12}$	0.07643
5	05	$3.76543 \times 10^{-27}$	0.00123



**Figure 11.** The error graph of the multiplicative calculus-based parallel scheme CM<sub>1</sub><sup>[\*]</sup> for solving (70) using random test vectors.

### 5. Conclusions

We developed an innovative family of iterative techniques based on a multiplicative approach with a local convergence order of 2 for efficiently finding simple solutions to Equation (2). A hybrid strategy utilizing multiplicative calculus was further developed from this method, resulting in a parallel numerical methodology capable of simultaneously solving all solutions at one time for nonlinear fractional problems. Our convergence analysis shows significant performance benefits with a significantly higher convergence order of 12 for these parallel systems based on multiplicative calculus. Dynamical planes, as shown in Figures 1–11, were used for optimal starting value selection in order to further

increase the convergence rate of the  $CM1^{[*]}$  method. This reduced the number of iterations required to find exact answers. Extensive testing on various nonlinear problems confirmed the robustness, stability, and reliability of  $CM1^{[*]}$ , outperforming comparable approaches such as  $SM1^{[*]}$ ,  $SM2^{[*]}$ , and  $PM^{[*]}$ . The  $CM1^{[*]}$  families of parallel schemes outperform both  $SM1^{[*]} - SM2^{[*]}$  and  $PM^{[*]}$  methods in terms of absolute residual error (Tables 2–4 and 6–8), the central processing unit time, and the log of residual error graphs (Figures 9 and 11) for varying  $\alpha, \beta$  values, as demonstrated by the numerical results in Tables 1–8.

In the future, new efficient hybrid multiplicative calculus-based inverse parallel schemes with global convergence behavior [55,56] will be developed to address complex epidemic and vectorial problems.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The author declares that there are no conflicts of interest regarding the publication of this article.

**Abbreviations**

In this article, the following abbreviations are used:

$CM1^{[*]}$	parallel scheme
$SM1^{[*]} - SM2^{[*]}$	existing parallel scheme
$PM^{[*]}$	existing parallel scheme
CPU time	computational time
e-	$10^{-()}$
$\rho_{\zeta}^{[k-1]}$	local computational order
n	iteration

**Appendix A**

The random initial starting vector is employed in Example 1 to show the global convergence of the numerical schemes, as presented in Table A1.

**Table A1.** Random initial vectors for Example 1.

Ran-Test	$x_1^{[0]}$ ,	$x_2^{[0]}$ ,	$x_3^{[0]}$	$x_4^{[0]}$	$x_5^{[0]}$
1	[0.732,	0.091,	0.007	0.181	0.097]
2	[0.822,	0.048,	0.131	0.104	0.634]
3	[0.083,	0.243,	0.224	0.590	0.823]
4	[0.076,	0.432,	0.046	0.913	0.643]
5	[0.011,	0.283,	0.057	0.703	0.253]

The random initial starting vector is employed in Example 2 to show the global convergence of the numerical schemes, as presented in Table A2.

**Table A2.** Random initial vectors for Example 2.

Ran-Test	$x_1^{[0]}$ ,	$x_2^{[0]}$ ,	$x_3^{[0]}$
1	[0.052,	0.101,	0.097]
2	[0.071,	0.048,	0.674]
3	[0.078,	0.003,	0.173]
4	[0.106,	0.102,	0.243]
5	[0.001,	0.203,	0.003]

## References

1. Malkus, D.S.; Nohel, J.A.; Plohr, B.J. Dynamics of shear flow of a non-Newtonian fluid. *J. Comput. Phys.* **1990**, *87*, 464–487. [[CrossRef](#)]
2. Feng, J.; Hu, H.H.; Joseph, D.D. Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid Part 1. Sedimentation. *J. Fluid Mech.* **1994**, *261*, 5–134. [[CrossRef](#)]
3. Blair, P.M.; Weinaug, C.F. Solution of two-phase flow problems using implicit difference equations. *Soc. Pet. Eng. J.* **1969**, *9*, 417–424. [[CrossRef](#)]
4. Levy, D. Chaos theory and strategy: Theory, application, and managerial implications. *Strateg. Manag. J.* **1994**, *15*, 167–178. [[CrossRef](#)]
5. Chen, G.; Moiola, J.L.; Wang, H.O. Bifurcation control: Theories, methods, and applications. *Int. J. Bifurc. Chaos* **2000**, *10*, 511–548. [[CrossRef](#)]
6. Pritchett, L.; Woolcock, M. Solutions when the solution is the problem: Arraying the disarray in development. *World Dev.* **2004**, *32*, 191–212. [[CrossRef](#)]
7. Metzler, R.; Klafter, J. The random walk's guide to anomalous diffusion: A fractional dynamics approach. *Phys. Rep.* **2000**, *339*, 1–77. [[CrossRef](#)]
8. Diethelm, K.; Ford, N.J. Analysis of fractional differential equations. *J. Math. Anal. Appl.* **2002**, *265*, 229–248. [[CrossRef](#)]
9. Daftardar-Gejji, V.; Babakhani, A. Analysis of a system of fractional differential equations. *J. Math. Anal. Appl.* **2004**, *293*, 511–522. [[CrossRef](#)]
10. Lakshmikantham, V.; Vatsala, A.S. Basic theory of fractional differential equations. *Nonlinear Anal. Theory Methods Appl.* **2008**, *69*, 2677–2682. [[CrossRef](#)]
11. Stanley, D. A multiplicative calculus. *Probl. Resour. Issues Math. Undergrad. Stud.* **1999**, *9*, 310–326. [[CrossRef](#)]
12. Grossman, M.; Katz, R. Non-Newtonian Calculus: A Self-contained, Elementary Exposition of the Authors' Investigations. *Non-Newton. Calc.* **1972**, *1*, 1–1090.
13. Bashirov, A.E.; Kurpinar, E.M.; Özyapıcı, A. Multiplicative calculus and its applications. *J. Math. Anal. Appl.* **2008**, *337*, 36–48. [[CrossRef](#)]
14. Bashirov, A.E.; Riza, M. On complex multiplicative differentiation. *TWMS J. Appl. Eng. Math.* **2011**, *1*, 75–85.
15. Bashirov, A.E.; Misirli, E.; Tandoğdu, Y.; Özyapıcı, A. On modeling with multiplicative differential equations. *Appl. Math.-J. Chin. Univ.* **2011**, *26*, 425–438. [[CrossRef](#)]
16. Willinger, W.; Govindan, R.; Jamin, S.; Paxson, V.; Shenker, S. Scaling phenomena in the Internet: Critically examining criticality. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 2573–2580. [[CrossRef](#)]
17. Harima, Y.; Sakamoto, Y.; Tanaka, S.L.; Kawai, M. Validity of the geometric-progression formula in approximating gamma-ray buildup factors. *Nucl. Sci. Eng.* **1986**, *94*, 24–35. [[CrossRef](#)]
18. Ozbay, S. Modified Backpropagation Algorithm with Multiplicative Calculus in Neural Networks. *Elektronika ir Elektrotechnika* **2023**, *29*, 55–61. [[CrossRef](#)]
19. Karthikeyan, K.R.; Murugusundaramoorthy, G. Properties of a Class of Analytic Functions Influenced by Multiplicative Calculus. *Fractal Fract.* **2024**, *8*, 131. [[CrossRef](#)]
20. Othman, G.M.; Yurtkan, K.; Özyapıcı, A. Improved digital image interpolation technique based on multiplicative calculus and Lagrange interpolation. *Signal Image Video Process.* **2023**, *17*, 3953–3961. [[CrossRef](#)]
21. Eyilmaz, E.; Gunes, E. Inverse nodal problem for the Sturm-Liouville equation in multiplicative case. *Ann. Math. Comput. Sci.* **2023**, *13*, 42–52.
22. Rasham, T.; Nazam, M.; Agarwal, P.; Hussain, A.; Al Sulmi, H.H. Existence results for the families of multi-mappings with applications to integral and functional equations. *J. Inequalities Appl.* **2023**, *2023*, 82. [[CrossRef](#)]
23. Goktas, S. A New Type of Sturm-Liouville Equation in the Non-Newtonian Calculus. *J. Funct. Spaces* **2021**, *1*, 5203939. [[CrossRef](#)]
24. Yalcın, N.; Dedeturk, M. Solutions of multiplicative linear differential equations via the multiplicative power series method. *Sigma* **2023**, *41*, 837–847.
25. Calogero, F.; Yi, G. Can the general solution of the second-order ODE characterizing Jacobi polynomials be polynomial? *J. Phy. A Math. Theor.* **2012**, *45*, 095206. [[CrossRef](#)]
26. Sana, G.; Mohammed, P.O.; Shin, D.Y.; Noor, M.A.; Oudat, M.S. On iterative methods for solving nonlinear equations in quantum calculus. *Fractal Fract.* **2021**, *5*, 60. [[CrossRef](#)]
27. Mateen, A.; Zhang, Z.; Ali, M.A.; Feckan, M. Generalization of Some Integral Inequalities in Multiplicative Calculus with Their Computational Analysis. *Preprint* **2024**, 1–509. [[CrossRef](#)]
28. Bilgehan, B.; Özyapıcı, A.; Hammouch, Z.; Gurefe, Y. Predicting the spread of COVID-19 with a machine learning technique and multiplicative calculus. *Soft Comput.* **2022**, *26*, 8017–8024. [[CrossRef](#)]
29. Boruah, K.; Hazarika, B. Some basic properties of bigeometric calculus and its applications in numerical analysis. *Afr. Mat.* **2021**, *32*, 211–227. [[CrossRef](#)]
30. Goktas, S.; Yilmaz, E.; Yar, A.C. Multiplicative derivative and its basic properties on time scales. *Math. Meth. Appl. Sci.* **2022**, *45*, 2097–2109. [[CrossRef](#)]
31. Özyapıcı, A.; Sensoy, Z.B.; Karanfiller, T. Effective Root-Finding Methods for Nonlinear Equations Based on Multiplicative Calculi. *J. Math.* **2016**, *2016*, 8174610. [[CrossRef](#)]

32. Du, T.; Peng, Y. Hermite–Hadamard type inequalities for multiplicative Riemann–Liouville fractional integrals. *J. Comput. Appl. Math.* **2024**, *440*, 115582. [[CrossRef](#)]
33. Chun, C. Some fourth-order iterative methods for solving nonlinear equations. *Appl. Math. Comput.* **2008**, *195*, 454–459. [[CrossRef](#)]
34. Shams, M.; Kausar, N.; Samaniego, C.; Agarwal, P.; Ahmed, S.F.; Momani, S. On efficient fractional Caputo-type simultaneous scheme for finding all roots of polynomial equations with biomedical engineering applications. *Fractals* **2023**, *31*, 2340075. [[CrossRef](#)]
35. Gupta, A.; Kumar, S. A multiplicative Gauss–Newton minimization algorithm: Theory and application to exponential functions. *Appl. Math.-J. Chin. Univ.* **2021**, *36*, 370–389. [[CrossRef](#)]
36. Unal, E.; Cumhur, I.; Gokdogan, A. Multiplicative Newton’s Methods with Cubic Convergence. *New Trends Math. Sci.* **2017**, *5*, 299–307. [[CrossRef](#)]
37. Singh, G.; Bhalla, S.; Behl, R. Higher-order multiplicative derivative iterative scheme to solve the nonlinear problems. *Math. Comput. Appl.* **2023**, *28*, 23. [[CrossRef](#)]
38. Waseem, M.; Noor, M.A.; Shah, F.A.; Noor, K.I. An efficient technique to solve nonlinear equations using multiplicative calculus. *Turk. J. Math.* **2018**, *42*, 679–691. [[CrossRef](#)]
39. Chicharro, F.I.; Cordero, A.; Garrido, N.; Torregrosa, J.R. Stability and applicability of iterative methods with memory. *J. Math. Chem.* **2019**, *57*, 1282–1300. [[CrossRef](#)]
40. 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. Prob. Eng.* **2020**, *2020*, 3524324. [[CrossRef](#)]
41. Kennes, R. Computational aspects of the Mobius transformation of graphs. *IEEE Trans. Syst. Man Cybern.* **1992**, *22*, 201–223. [[CrossRef](#)]
42. Cordero, A.; Reyes, J.A.; Torregrosa, J.R.; Vassileva, M.P. Stability Analysis of a New Fourth-Order Optimal Iterative Scheme for Nonlinear Equations. *Axioms* **2023**, *13*, 34. [[CrossRef](#)]
43. Herceg, D.; Tričković, S.; Petković, M. On the fourth order methods of Weierstrass’ type. *Nonlinear Anal. Theory Methods Appl.* **1997**, *30*, 83–88. [[CrossRef](#)]
44. Anourein, A.W.M. An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial. *Int. J. Comput. Math.* **1977**, *6*, 241–252. [[CrossRef](#)]
45. Petkovic, M.S.; Petkovic, L.D.; Dcunic, J. On an efficient method for the simultaneous approximation of polynomial multiple roots. *Appl. Anal. Discret. Math.* **2014**, *1*, 73–94. [[CrossRef](#)]
46. Petkovic, M.S.; Petkovic, L.D.; Dzunic, J. On an efficient simultaneous method for finding polynomial zeros. *Appl. Math. Lett.* **2014**, *28*, 60–65. [[CrossRef](#)]
47. Kou, J.; Li, Y.; Wang, X. Some variants of Ostrowski’s method with seventh-order convergence. *J. Comput. Appl. Math.* **2007**, *209*, 153–159. [[CrossRef](#)]
48. Diethelm, K.; Siegmund, S.; Tuan, H.T. Asymptotic behavior of solutions of linear multi-order fractional differential systems. *Fract. Calc. Appl. Anal.* **2017**, *20*, 1165–1195. [[CrossRef](#)]
49. Shams, M.; Rafiq, N.; Kausar, N.; Agarwal, P.; Park, C.; Mir, N.A. On highly efficient derivative-free family of numerical methods for solving polynomial equation simultaneously. *Adv. Differ. Equ.* **2021**, *2021*, 465. [[CrossRef](#)]
50. 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. Differ. Equ.* **2021**, *2021*, 495. [[CrossRef](#)]
51. Yang, Y. Solving a nonlinear multi-order fractional differential equation using Legendre pseudo-spectral method. *Appl. Math.* **2013**, *4*, 113–118. [[CrossRef](#)]
52. Uwaheren, O.A.; Adebisi, A.F.; Ishola, C.Y.; Raji, M.T.; Yekeen, A.O.; Peter, O.J. Numerical Solution of Volterra integro-differential Equations by Akbari-Ganji’s Method. *BAREKENG: Jurnal Ilmu Mat. Terap.* **2022**, *16*, 1123–1130. [[CrossRef](#)]
53. Ziada, E.A.A. Solution of Nonlinear Fractional Differential Equations Using Adomain Decomposition Method. *J. Syst. Sci. Appl. Math.* **2021**, *6*, 111–119.
54. Ray, S.S.; Bera, R.K. An approximate solution of a nonlinear fractional differential equation by Adomian decomposition method. *Appl. Math. Comput.* **2005**, *167*, 561–571.
55. Cordero, A.; Hueso, J.L.; Martínez, E.; Torregrosa, J.R. Increasing the convergence order of an iterative method for nonlinear systems. *Appl. Math. Lett.* **2012**, *25*, 2369–2374. [[CrossRef](#)]
56. Rafiq, N.; Akram, S.; Shams, M.; Mir, N.A. Computer geometries for finding all real zeros of polynomial equations simultaneously. *Comput. Math. Contin.* **2021**, *69*, 2636–2651. [[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.