axioms

MDPI

Article

Achieving Optimal Order in a Novel Family of Numerical Methods: Insights from Convergence and Dynamical Analysis Results

Marlon Moscoso-Martínez [1,2,3], Francisco I. Chicharro [1], Alicia Cordero [1,*], Juan R. Torregrosa [1] and Gabriela Ureña-Callay [2,3]

[1] Institute for Multidisciplinary Mathematics, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain; marmosma@doctor.upv.es (M.M.-M.); frachilo@mat.upv.es (F.I.C.); jrtorre@mat.upv.es (J.R.T.)

[2] Faculty of Sciences, Escuela Superior Politécnica de Chimborazo (ESPOCH), Panamericana Sur km 1 1/2, Riobamba 060106, Ecuador; gabriela.urena@espoch.edu.ec

[3] Higher School of Engineering and Technology, Universidad Internacional de la Rioja (UNIR), Avda. de la Paz 137, 26006 Logroño, Spain

* Correspondence: acordero@mat.upv.es

Abstract: In this manuscript, we introduce a novel parametric family of multistep iterative methods designed to solve nonlinear equations. This family is derived from a damped Newton's scheme but includes an additional Newton step with a weight function and a "frozen" derivative, that is, the same derivative than in the previous step. Initially, we develop a quad-parametric class with a first-order convergence rate. Subsequently, by restricting one of its parameters, we accelerate the convergence to achieve a third-order uni-parametric family. We thoroughly investigate the convergence properties of this final class of iterative methods, assess its stability through dynamical tools, and evaluate its performance on a set of test problems. We conclude that there exists one optimal fourth-order member of this class, in the sense of Kung–Traub's conjecture. Our analysis includes stability surfaces and dynamical planes, revealing the intricate nature of this family. Notably, our exploration of stability surfaces enables the identification of specific family members suitable for scalar functions with a challenging convergence behavior, as they may exhibit periodical orbits and fixed points with attracting behavior in their corresponding dynamical planes. Furthermore, our dynamical study finds members of the family of iterative methods with exceptional stability. This property allows us to converge to the solution of practical problem-solving applications even from initial estimations very far from the solution. We confirm our findings with various numerical tests, demonstrating the efficiency and reliability of the presented family of iterative methods.

Keywords: nonlinear equations; optimal iterative methods; convergence analysis; dynamical study; stability

MSC: 65H05

Citation: Moscoso-Martínez, M.; Chicharro, F.I.; Cordero, A.; Torregrosa, J.R.; Ureña-Callay, G. Achieving Optimal Order in a Novel Family of Numerical Methods: Insights from Convergence and Dynamical Analysis Results. *Axioms* **2024**, *13*, 458. https://doi.org/10.3390/axioms13070458

Academic Editor: Behzad Djafari-Rouhani

Received: 20 May 2024
Revised: 25 June 2024
Accepted: 2 July 2024
Published: 7 July 2024

1. Introduction

A multitude of challenges in Computational Sciences and other fields in Science and Technology can be effectively represented as nonlinear equations through mathematical modeling, see for example [1–3]. Finding solutions ξ to nonlinear equations of the form $f(x) = 0$ stands as a classical yet formidable problem in the realms of Numerical Analysis, Applied Mathematics, and Engineering. Here, the function $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ is assumed to be differentiable enough within the open interval I. Extensive overviews of iterative methods for solving nonlinear equations published in recent years can be found in [4–6], and their associated references.

In recent years, many iterative methods have been developed to solve nonlinear equations. The essence of these methods is as follows: if one knows a sufficiently small domain that contains only one root ξ of the equation $f(x) = 0$, and we select a sufficiently close initial estimate of the root x_0, we generate a sequence of iterates $x_1, x_2, \ldots, x_k, \ldots$, by means of a fixed point function $g(x)$, which under certain conditions converges to ξ. The convergence of the sequence is guaranteed, among other elements, by the appropriate choice of the function g and the initial approximation x_0.

The method described by the iteration function $g : I \subseteq \mathbb{R} \to \mathbb{R}$ such that

$$x_{k+1} = g(x_k), \quad k = 0, 1, \ldots, \tag{1}$$

starting from a given initial estimate x_0, includes a large number of iterative schemes. These differ from each other by the way the iteration function g is defined.

Among these methods, Newton's scheme is widely acknowledged as the most renowned approach for locating a solution $\xi \in I$. This scheme is defined by the iterative formula:

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

where $k = 0, 1, 2, \ldots$, and $f'(x_k)$ denotes the derivative of the function f evaluated in the kth iteration.

A very important concept of iterative methods is their order of convergence, which provides a measure of the speed of convergence of the iterates. Let $\{x_k\}_{k \geq 0}$ be a sequence of real numbers such that $\lim_{k\to\infty} x_k = \xi$. The convergence is called (see [7]):

(a) Linear, if there exist C, $0 < C < 1$ and $k_0 \in \mathbb{N}$ such that

$$\frac{|x_k - \xi|}{|x_{k-1} - \xi|} \leq C, \text{ for all } k > k_0;$$

(b) Is of order p, if there exist $C > 0$ and $k_0 \in \mathbb{N}$ such that

$$\frac{|x_k - \xi|}{|x_{k-1} - \xi|^p} \leq C, \text{ for all } k > k_0.$$

We denote by $e_k = x_k - \xi$ the error of the k-th iteration. Moreover, equation $e_{k+1} = Ce_k^p + O(e_k^{p+1})$ is called the error equation of the iterative method, where p is its order of convergence and C is called the asymptotic error constant.

It is known (see, for example, [4]) that if $x_{k+1} = g(x_k)$ is an iterative point-to-point method with d functional evaluations per step, then the order of convergence of the method is, at most, $p = d$. On the other hand, Traub proves in [4] that to design a point-to-point method of order p, the iterative expression must contain derivatives of the nonlinear function whose zero we are looking for, at least of order $p - 1$. This is why point-to-point methods are not efficient if we seek to simultaneously increase the order of convergence and computational efficiency.

These restrictions of point-to-point methods are the starting point for the growing interest of researchers in multipoint methods, see for example [4–6]. In such schemes, also called predictor–corrector, the $(k + 1)$-th iterate is obtained by using functional evaluations of the k-th iterate and also of other intermediate points. For example, a two-step multipoint method has the expression

$$\begin{aligned} y_k &= \Psi(x_k), \\ x_{k+1} &= \Phi(x_k, y_k), \; k = 0, 1, 2, \ldots \end{aligned}$$

Thus, the main motivation for designing new iterative schemes is to increase the order of convergence without adding many functional evaluations. The first multipoint schemes were designed by Traub in [4]. At that time the concept of optimality had not yet been defined and the fact of designing multipoint schemes with the same order as classical schemes such as Halley or Chebyshev, but with a much simpler iterative expression and without using second derivatives, was of great importance. The techniques used then have been the seed of those that allowed the appearance of higher-order methods.

In recent years, different authors have developed a large number of optimal schemes for solving nonlinear equations [6,8]. A common way to increase the convergence order of an iterative scheme is to use the composition of methods, based on the following result (see [4]).

Theorem 1. *Let $g_1(x)$ and $g_2(x)$ be the fixed-point functions of orders p_1 and p_2, respectively. Then, the iterative method resulting from composing them, $x_{k+1} = g_1(g_2(x_k))$, $k = 0, 1, 2, \ldots$, has an order of convergence $p_1 p_2$.*

However, this composition necessarily increases the number of functional evaluations. So, to preserve optimality, the number of evaluations must be reduced. There are many techniques used for this purpose by different authors, such as approximating some of the evaluations that have appeared with the composition by means of interpolation polynomials, Padé approximants, inverse interpolation, Adomian polynomials, etc. (see, for example, [6,9,10]). If after the reduction of functional evaluations the resulting method is not optimal, the weight function technique, introduced by Chun in [11], can be used to increase its order of convergence.

There are also other ways in the literature to compare different iterative methods with each other. Traub in [4] defined the information efficiency of an iterative method as

$$I(M) = \frac{p}{d},$$

where p is the order of convergence and d is the number of functional evaluations per iteration. On the other hand, Ostrowski in [12] introduced the so-called efficiency index,

$$EI(M) = p^{1/d},$$

which, in turn, gives rise to the concept of optimality of an iterative method.

Regarding the order of convergence, Kung and Traub in their conjecture (see [13]) establish what is the highest order that a multipoint iterative scheme without memory can reach. Schemes that attain this limit are called optimal methods. Such a conjecture states that the order of convergence of any memoryless multistep method cannot exceed 2^{d-1} (called optimal order), where d is the number of functional evaluations per iteration, with efficiency index $2^{(d-1)/d}$ (called optimal index). In this sense, Newton is an optimal method.

Furthermore, in order to numerically test the behavior of the different iterative methods, Weerakoon and Fernando in [14] introduced the so-called computational order of convergence (COC),

$$p \approx COC = \frac{\ln \frac{|x_{k+1} - \xi|}{|x_k - \xi|}}{\ln \frac{|x_k - \xi|}{|x_{k-1} - \xi|}}, \ k = 1, 2, \ldots,$$

where x_{k+1}, x_k and x_{k-1} are three consecutive approximations of the root of the nonlinear equation, obtained in the iterative process. However, the value of the zero ξ is not known in practice, which motivated the definition in [15] of the approximate computational convergence order ACOC,

$$p \approx ACOC = \frac{\ln \frac{|x_{k+1} - x_k|}{|x_k - x_{k-1}|}}{\ln \frac{|x_k - x_{k-1}|}{|x_{k-1} - x_{k-2}|}}, \ k = 2, 3, \ldots. \quad (2)$$

On the other hand, the dynamical analysis of rational operators derived from iterative schemes, particularly when applied to low-degree nonlinear polynomial equations, has emerged as a valuable tool for assessing the stability and reliability of these numerical methods. This approach is detailed, for instance, in Refs. [16–20] and their associated references.

Using the tools of complex discrete dynamics, it is possible to compare different algorithms in terms of their basins of attraction, the dynamical behavior of the rational functions associated with the iterative method on low-degree polynomials, etc. Varona [21], Amat et al. [22], Neta et al. [23], Cordero et al. [24], Magreñán [25], Geum et al. [26], among others, have analyzed many schemes and parametric families of methods under this point of view, obtaining interesting results about their stability and reliability.

The dynamical analysis of an iterative method focuses on the study of the asymptotic behavior of the fixed points (roots, or not, of the equation) of the operator, as well as on the basins of attraction associated with them. In the case of parametric families of iterative methods, the analysis of the free critical points (points where the derivative of the operator cancels out that are not roots of the nonlinear function) and stability functions of the fixed points allows us to select the most stable members of these families. Some of the existing works in the literature related to this approach are Refs. [27,28], among others.

In this paper, we introduce a novel parametric family of multistep iterative methods tailored for solving nonlinear equations. This family is constructed by enhancing the traditional Newton's scheme, incorporating an additional Newton step with a weight function and a frozen derivative. As a result, the family is characterized by a two-step iterative expression that relies on four arbitrary parameters.

Our approach yields a third-order uni-parametric family and a fourth-order member. However, in the course of developing these iterative schemes, we initially start with a first-order quad-parametric family. By selectively setting just one parameter, we manage to accelerate its convergence to a third-order scheme, and for a specific value of this parameter, we achieve an optimal member. To substantiate these claims, we conduct a comprehensive convergence analysis for all classes.

The stability of this newly introduced family is rigorously examined using dynamical tools. We construct stability surfaces and dynamical planes to illustrate the intricate behavior of this class. These stability surfaces help us to identify specific family members with exceptional behavior, making them well-suited for practical problem-solving applications. To further demonstrate the efficiency and reliability of these iterative schemes, we conduct several numerical tests.

The rest of the paper is organized as follows. In Section 2, we present the proposed class of iterative methods depending on several parameters, which is step-by-step modified in order to achieve the highest order of convergence. Section 3 is devoted to the dynamical study of the uni-parametric family; by means of this analysis, we find the most stable members, less dependent from their initial estimation. In Section 4, the previous theoretical results are checked by means of numerical tests on several nonlinear problems, using a wide variety of initial guesses and parameter values. Finally, some conclusions are presented.

2. Convergence Analysis of the Family

In this section, we conduct a convergence analysis of the newly introduced quad-parametric iterative family, with the following iterative expression:

$$\begin{cases} y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta \left(\dfrac{f(y_k)}{f(x_k)}\right)^2\right) \dfrac{f(x_k)}{f'(x_k)}, \end{cases} \tag{3}$$

where α, β, γ, δ are arbitrary parameters and $k = 0, 1, 2, \ldots$.

Additionally, we present a strategy for simplifying it into a uni-parametric class to enhance convergence speed. Consequently, even though the quad-parametric family has a first-order convergence rate, we employ higher-order Taylor expansions in our proof, as they are instrumental in establishing the convergence rate of the uni-parametric subfamily. In Appendix A, the Mathematica code used for checking it is available.

Theorem 2 (quad-parametric family). *Let $f : I \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function in an open interval I and $\xi \in I$ a simple root of the nonlinear equation $f(x) = 0$. Let us suppose that $f'(x)$ is continuous and nonsingular at ξ, and x_0 is an initial estimate close enough to ξ. Then, the sequence $\{x_k\}_{k \geq 0}$ obtained by using the expression (3) converges to ξ with order one, being its error equation*

$$e_{k+1} = \left(-\alpha^2\delta + \alpha(\gamma + 2\delta - 1) - \beta - \gamma - \delta + 1\right)e_k + \mathcal{O}\left(e_k^2\right),$$

where $e_k = x_k - \xi$, and α, β, γ, and δ are free parameters.

Proof. Let us consider ξ as the simple root of nonlinear function $f(x)$ and $x_k = \xi + e_k$. We calculate the Taylor expansion of $f(x_k)$ and $f'(x_k)$ around the root ξ, we get

$$\begin{aligned} f(x_k) &= f(\xi) + f'(\xi)e_k + \frac{1}{2!}f''(\xi)e_k^2 + \frac{1}{3!}f'''(\xi)e_k^3 + \frac{1}{4!}f^{(iv)}(\xi)e_k^4 + \mathcal{O}(e_k^5) \\ &= f'(\xi)\left[e_k + \frac{1}{2!}\frac{f''(\xi)}{f'(\xi)}e_k^2 + \frac{1}{3!}\frac{f'''(\xi)}{f'(\xi)}e_k^3 + \frac{1}{4!}\frac{f^{(iv)}(\xi)}{f'(\xi)}e_k^4\right] + \mathcal{O}(e_k^5) \\ &= f'(\xi)\left[e_k + C_2e_k^2 + C_3e_k^3 + C_4e_k^4\right] + \mathcal{O}(e_k^5), \end{aligned} \tag{4}$$

and

$$\begin{aligned} f'(x_k) &= f'(\xi) + f''(\xi)e_k + \frac{1}{2!}f'''(\xi)e_k^2 + \frac{1}{3!}f^{(iv)}(\xi)e_k^3 + \mathcal{O}(e_k^4) \\ &= f'(\xi)\left[1 + \frac{f''(\xi)}{f'(\xi)}e_k + \frac{1}{2!}\frac{f'''(\xi)}{f'(\xi)}e_k^2 + \frac{1}{3!}\frac{f^{(iv)}(\xi)}{f'(\xi)}e_k^3\right] + \mathcal{O}(e_k^4) \\ &= f'(\xi)\left[1 + 2C_2e_k + 3C_3e_k^2 + 4C_4e_k^3\right] + \mathcal{O}(e_k^4), \end{aligned} \tag{5}$$

where $C_p = \dfrac{1}{p!}\dfrac{f^{(p)}(\xi)}{f'(\xi)}$, $p = 2, 3, \ldots$

By a direct division of (4) and (5),

$$\frac{f(x_k)}{f'(x_k)} = e_k - C_2e_k^2 + 2\left(C_2^2 - C_3\right)e_k^3 - \left(4C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{6}$$

Replacing (6) in (3), we have

$$y_k = \xi + (1-\alpha)e_k + \alpha C_2e_k^2 - 2\alpha\left(C_2^2 - C_3\right)e_k^3 + \alpha\left(4C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{7}$$

Again a Taylor expansion of $f(y_k)$ around ξ allows us to get

$$\begin{aligned} f(y_k) = f'(\xi)\Big[&(1-\alpha)e_k + \left(\alpha^2 - \alpha + 1\right)C_2e_k^2 + \left(-2\alpha^2C_2^2 - \left(\alpha^3 - 3\alpha^2 + \alpha - 1\right)C_3\right)e_k^3 \\ &+ \left(5\alpha^2C_2^3 + \alpha^2(3\alpha - 10)C_2C_3 + \left(\alpha^4 - 4\alpha^3 + 6\alpha^2 - \alpha + 1\right)C_4\right)e_k^4\Big] + \mathcal{O}\left(e_k^5\right). \end{aligned} \tag{8}$$

Dividing (8) by (4), we obtain

$$\begin{aligned} \frac{f(y_k)}{f(x_k)} &= (1-\alpha) + \alpha^2C_2e_k - \alpha^2\left((\alpha - 3)C_3 + 3C_2^2\right)e_k^2 \\ &+ \alpha^2\left(\left(\alpha^2 - 4\alpha + 6\right)C_4 + 2(2\alpha - 7)C_2C_3 + 8C_2^3\right)e_k^3 + \mathcal{O}\left(e_k^4\right). \end{aligned} \tag{9}$$

Finally, substituting (6), (7) and (9), in the second step of family (3), we have

$$x_{k+1} = \xi + A_1 e_k + A_2 e_k^2 + A_3 e_k^3 + A_4 e_k^4 + \mathcal{O}\left(e_k^5\right), \tag{10}$$

where

$$\begin{aligned}
A_1 &= -\alpha^2\delta + \alpha(\gamma + 2\delta - 1) - \beta - \gamma - \delta + 1,\\
A_2 &= \left(2\alpha^3\delta - \alpha^2(\gamma+\delta) - \alpha(\gamma+2\delta-1) + \beta + \gamma + \delta\right)C_2,\\
A_3 &= \left(-2\alpha^4\delta + \alpha^3(\gamma+8\delta) - \alpha^2(3\gamma+4\delta) - 2\alpha(\gamma+2\delta-1) + 2(\beta+\gamma+\delta)\right)C_3\\
&\quad - \left(\alpha^4\delta + 8\alpha^3\delta - 2\alpha^2(2\gamma+3\delta) - 2\alpha(\gamma+2\delta-1) + 2(\beta+\gamma+\delta)\right)C_2^2,\\
A_4 &= \left(7\alpha^4\delta + 26\alpha^3\delta - \alpha^2(13\gamma+22\delta) - 4\alpha(\gamma+2\delta-1) + 4(\beta+\gamma+\delta)\right)C_2^3\\
&\quad + \left(2\alpha^5\delta + 4\alpha^4\delta - \alpha^3(5\gamma+48\delta) + \alpha^2(19\gamma+31\delta) + 7\alpha(\gamma+2\delta-1) - 7(\beta+\gamma+\delta)\right)C_2C_3\\
&\quad + \left(2\alpha^5\delta - \alpha^4(\gamma+10\delta) + 4\alpha^3(\gamma+5\delta) - 3\alpha^2(2\gamma+3\delta) - 3\alpha(\gamma+2\delta-1) + 3(\beta+\gamma+\delta)\right)C_4,
\end{aligned} \tag{11}$$

being the error equation

$$\begin{aligned}
e_{k+1} &= A_1 e_k + A_2 e_k^2 + A_3 e_k^3 + A_4 e_k^4 + \mathcal{O}\left(e_k^5\right)\\
&= \left(-\alpha^2\delta + \alpha(\gamma+2\delta-1) - \beta - \gamma - \delta + 1\right)e_k + \mathcal{O}\left(e_k^2\right),
\end{aligned} \tag{12}$$

and the proof is finished. □

From Theorem 2, it is evident that the newly introduced quad-parametric family exhibits a convergence order of one, irrespective of the values assigned to α, β, γ, and δ. Nevertheless, we can expedite convergence by holding only two parameters constant, effectively reducing the family to a bi-parametric iterative scheme. In Appendix B, the Mathematica code used for checking it is available.

Theorem 3 (bi-parametric family). *Let $f : I \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function in an open interval I and $\xi \in I$ a simple root of the nonlinear equation $f(x) = 0$. Let us suppose that $f'(x)$ is continuous and nonsingular at ξ, and x_0 is an initial estimate close enough to ξ. Then, the sequence $\{x_k\}_{k\geq 0}$ obtained by using the expression (3) converges to ξ with order three, provided that $\beta = \frac{(\alpha-1)^2(\alpha^2\delta-\alpha-1)}{\alpha^2}$ and $\gamma = \frac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}$, being its error equation*

$$e_{k+1} = \left(-\left(\alpha^4\delta - 2\right)C_2^2 + (\alpha-1)C_3\right)e_k^3 + \mathcal{O}\left(e_k^4\right),$$

where $e_k = x_k - \xi$, $C_q = \frac{1}{q!}\frac{f^{(q)}(\xi)}{f'(\xi)}$, $q = 2, 3, \ldots$, and α, δ are arbitrary parameters.

Proof. Using the results of Theorem 2 to cancel A_1 and A_2 accompanying e_k and e_k^2 in (12), respectively, it must be satisfied that

$$\begin{cases} -\alpha^2\delta + \alpha(\gamma+2\delta-1) - \beta - \gamma - \delta + 1 = 0,\\ 2\alpha^3\delta - \alpha^2(\gamma+\delta) - \alpha(\gamma+2\delta-1) + \beta + \gamma + \delta = 0. \end{cases} \tag{13}$$

It is clear that system (13) has infinite solutions for

$$\beta = \frac{(\alpha-1)^2(\alpha^2\delta-\alpha-1)}{\alpha^2} \quad \text{and} \quad \gamma = \frac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}, \tag{14}$$

where α and δ are free parameters. Therefore, replacing (14) in (11), we obtain that

$$\begin{aligned}
A_1 &= 0,\\
A_2 &= 0,\\
A_3 &= -\left(\alpha^4\delta - 2\right)C_2^2 + (\alpha - 1)C_3,\\
A_4 &= \left(7\alpha^4\delta - 9\right)C_2^3 + \left(2(\alpha - 3)\alpha^4\delta - 5\alpha + 12\right)C_2C_3 - (\alpha - 3)(\alpha - 1)C_4,
\end{aligned} \tag{15}$$

being the error equation

$$\begin{aligned}
e_{k+1} &= A_3 e_k^3 + \mathcal{O}(e_k^4)\\
&= \left(-\left(\alpha^4\delta - 2\right)C_2^2 + (\alpha - 1)C_3\right)e_k^3 + \mathcal{O}(e_k^4),
\end{aligned} \tag{16}$$

and the proof is finished. □

According to the findings in Theorem 3, it is evident that the newly introduced bi-parametric family

$$\begin{cases}
y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)},\\
x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta\left(\dfrac{f(y_k)}{f(x_k)}\right)^2\right)\left(\dfrac{f(x_k)}{f'(x_k)}\right),
\end{cases} \tag{17}$$

where $k = 0, 1, 2, \ldots$, $\beta = \dfrac{(\alpha - 1)^2\left(\alpha^2\delta - \alpha - 1\right)}{\alpha^2}$ and $\gamma = \dfrac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}$ consistently exhibits a third-order convergence across all values of α and δ. Nevertheless, it is noteworthy that by restricting one of the parameters while transitioning to a uni-parametric iterative scheme, not only can we sustain convergence, but we can also enhance performance. This improvement arises from the reduction in the error equation complexity, resulting in more efficient computations.

Corollary 1 (uni-parametric family). *Let $f : I \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function in an open interval I and $\xi \in I$ a simple root of the nonlinear equation $f(x) = 0$. Let us suppose that $f'(x)$ is continuous and nonsingular at ξ and x_0 is an initial estimate close enough to ξ. Then, the sequence $\{x_k\}_{k\geq 0}$ obtained by using the expression (17) converges to ξ with order three, provided that $\epsilon = \alpha^4\delta = 2$, being its error equation*

$$e_{k+1} = (\alpha - 1)C_3 e_k^3 + \mathcal{O}\left(e_k^4\right),$$

where $e_k = x_k - \xi$, $C_q = \dfrac{1}{q!}\dfrac{f^{(q)}(\xi)}{f'(\xi)}$, $q = 2, 3, \ldots$, and α is an arbitrary parameter. Indeed, $\alpha = 1$ and, therefore, $\delta = \epsilon = 2$ provides the only member of the family of the optimal fourth-order of convergence.

Proof. Using the results of Theorem 3 to reduce the expression of A_3 accompanying e_k^3 in (15), it must be satisfied that $\alpha^4\delta - 2 = 0$ and/or $\alpha - 1 = 0$. It is easy to show that the first equation has infinite solutions for

$$\epsilon = \alpha^4\delta = 2. \tag{18}$$

Therefore, replacing (18) in (15), we obtain that

$$\begin{aligned} e_{k+1} &= A_3 e_k^3 + \mathcal{O}(e_k^4) \\ &= (\alpha - 1) C_3 e_k^3 + \mathcal{O}\left(e_k^4\right), \end{aligned} \tag{19}$$

and the proof is finished. □

Based on the outcomes derived from Corollary 1, it becomes apparent that the recently introduced uni-parametric family, which we will call MCCTU(α),

$$\begin{cases} y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta \left(\dfrac{f(y_k)}{f(x_k)}\right)^2\right)\left(\dfrac{f(x_k)}{f'(x_k)}\right), \end{cases} \tag{20}$$

where $k = 0, 1, 2, \ldots$, $\beta = \dfrac{(\alpha - 1)^2(\alpha^2\delta - \alpha - 1)}{\alpha^2}$, $\gamma = \dfrac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}$ and $\delta = \dfrac{2}{\alpha^4}$ consistently exhibits a convergence order of three, regardless of the chosen value for α. Nevertheless, a remarkable observation emerges when $\alpha = 1$: in such a case, a member of this family attains an optimal convergence order of four.

Due to the previous results, we have chosen to concentrate our efforts solely on the MCCTU(α) class of iterative schemes moving forward. To pinpoint the most effective members within this family, we will utilize dynamical techniques outlined in Section 3.

3. Stability Analysis

This section delves into the examination of the dynamical characteristics of the rational operator linked to the iterative schemes within the MCCTU(α) family. This exploration provides crucial insights into the stability and dependence of the members of the family with respect to the initial estimations used. To shed light on the performance, we create rational operators and visualize their dynamical planes. These visualizations enable us to discern the behavior of specific methods in terms of the attraction basins of periodic orbits, fixed points, and other relevant dynamics.

Now, we introduce the basic concepts of complex dynamics used in the dynamical analysis of iterative methods. The texts [29,30], among others, provide extensive and detailed information on this topic.

Given a rational function $R : \hat{\mathbb{C}} \to \hat{\mathbb{C}}$, where $\hat{\mathbb{C}}$ is the Riemann sphere, the orbit of a point $z_0 \in \hat{\mathbb{C}}$ is defined as:

$$\{z_0,\, R^1(z_0),\, R^2(z_0), \ldots, R^n(z_0), \ldots\}.$$

We are interested in the study of the asymptotic behavior of the orbits depending on the initial estimate z_0, analyzed in the dynamical plane of the rational function R defined by the different iterative methods.

To obtain these dynamical planes, we must first classify the fixed or periodic points of the rational operator R. A point $z_0 \in \hat{\mathbb{C}}$ is called a fixed point if it satisfies $R(z_0) = z_0$. If the fixed point is not a solution of the equation, it is called a strange fixed point. z_0 is said to be a periodic point of period $p > 1$ if $R^p(z_0) = z_0$ and $R^k(z_0) \neq z_0$, $k < p$. A critical point z_C is a point where $R'(z_C) = 0$.

On the other hand, a fixed point z_0 is called attracting if $|R'(z_0)| < 1$, superattracting if $|R'(z_0)| = 0$, repulsive if $|R'(z_0)| > 1$, and parabolic if $|R'(z_0)| = 1$.

The basin of attraction of an attractor $\bar{z}$ is defined as the set of pre-images of any order:

$$\mathcal{A}(\bar{z}) = \{z_0 \in \hat{\mathbb{C}} \;:\; R^n(z_0) \to \bar{z},\; n \to \infty\}.$$

The Fatou set consists of the points whose orbits have an attractor (fixed point, periodic orbit or infinity). Its complementary in $\hat{\mathbb{C}}$ is the Julia set, $\mathcal{J}$. Therefore, the Julia set includes all the repulsive fixed points and periodic orbits, and also their pre-images. So, the basin of attraction of any fixed point belongs to the Fatou set. Conversely, the boundaries of the basins of attraction compose the Julia set.

The following classical result, which is due to Fatou [31] and Julia [32], includes both periodic points (of any period) and fixed points, considered as periodic points of the unit period.

Theorem 4 ([31,32])**.** *Let R be a rational function. The immediate basins of attraction of each attracting periodic point contain at least one critical point.*

By means of this key result, all the attracting behavior can be found using the critical points as a seed.

3.1. Rational Operator

While the fixed-point operator can be formulated for any nonlinear function, our focus here lies on constructing this operator for low-degree nonlinear polynomial equations, in order to get a rational function. This choice stems from the fact that the stability or instability criteria applied to methods on these equations can often be extended to other cases. Therefore, we introduce the following nonlinear equation represented by $f(x)$:

$$f(x) = (x-a)(x-b) = 0, \tag{21}$$

where $a, b \in \mathbb{R}$ are the roots of the polynomial.

Let us remark that when MCCTU(α) is directly applied to $f(x)$, parameter α disappears in the resulting rational expression; so, no dynamical analysis can be made. However, if we use parameter $\epsilon = \alpha^4\delta$ appearing in Corollary 1 the same class of iterative methods can be expressed as MCCTU(ϵ) and the dynamical analisis can be made depending on ϵ.

Proposition 1 (rational operator R_f). *Let the polynomial equation $f(x)$ given in (21), for $a, b \in \mathbb{C}$. Rational operator R_f related to the MCCTU(ϵ) family given in (20) on $f(x)$ is*

$$R_f(x,\epsilon) = \frac{x^3\left(\epsilon - x^3 - 4x^2 - 5x - 2\right)}{x^3(\epsilon-2) - 5x^2 - 4x - 1}, \tag{22}$$

with $\epsilon \in \mathbb{C}$ being an arbitrary parameter.

Proof. Let $f(x)$ be a generic quadratic polynomial function with roots $a, b \in \mathbb{C}$. We apply the iterative scheme MCCTU(ϵ) given in (20) on $f(x)$ and obtain a rational function $A_f(x,\epsilon)$ that depends on the roots $a, b \in \mathbb{C}$ and the parameters $\epsilon \in \mathbb{C}$. Then, by using a Möbius transformation (see [22,33,34]) on $A_f(x,\epsilon)$ with

$$h(w) = \frac{w-a}{w-b},$$

satisfying $h(\infty) = 1, h(a) = 0$ and $h(b) = \infty$, we get

$$R_f(x,\epsilon) = \left(h \circ A_f(x,\epsilon) \circ h^{-1}\right)(x) = \frac{x^3\left(\epsilon - x^3 - 4x^2 - 5x - 2\right)}{x^3(\epsilon-2) - 5x^2 - 4x - 1}, \tag{23}$$

which depends on two arbitrary parameters $\epsilon \in \mathbb{C}$, thus completing the proof. □

From Proposition 1, if we set $\epsilon - 2 = 0$, we obtain

$$\delta = \frac{2}{\alpha^4}, \tag{24}$$

and, then, it is easy to show that the rational operator $R_f(x,\epsilon)$ simplifies to the expression

$$R_f(x) = \frac{x^4(x^2+4x+5)}{5x^2+4x+1}, \tag{25}$$

which does not depend on any free parameters.

3.2. Fixed Points

Now, we calculate all the fixed points of $R_f(x,\epsilon)$ given by (22), to afterwards analyze their character (attracting, repulsive, or neutral or parabolic).

Proposition 2. *The fixed points of $R_f(x,\epsilon)$ are $x=0$, $x=\infty$, and also five strange fixed points:*

- $ex_1 = 1$,
- $ex_{2,3}(\epsilon) = -\frac{5}{4} - \frac{1}{4}\sqrt{1-4\epsilon} \pm \frac{1}{2}\sqrt{\frac{5}{2} - \epsilon - \frac{20(\epsilon+8)-165}{2\sqrt{1-4\epsilon}}}$, *and*
- $ex_{4,5}(\epsilon) = -\frac{5}{4} + \frac{1}{4}\sqrt{1-4\epsilon} \pm \frac{1}{2}\sqrt{\frac{5}{2} - \epsilon + \frac{20(\epsilon+8)-165}{2\sqrt{1-4\epsilon}}}$.

By using Equation (24), the strange fixed points $ex_{2,3}(\epsilon)$ and $ex_{4,5}(\epsilon)$ do not depend on any free parameter,

- $ex_{2,3}(2) = -2.1943 \pm 1.5370i$, *and*
- $ex_{4,5}(2) = -0.3057 \pm 0.2142i$.

Morover, strange fixed points depending on ϵ are conjugated, $ex_{2,3}(\epsilon)$ and $ex_{4,5}(\epsilon)$. If $\epsilon = \frac{1}{4}$, $ex_1(\epsilon) = ex_3(\epsilon)$ and $ex_2(\epsilon) = ex_2(\epsilon)$, so the amount of strange fixed points is three. Indeed, $ex_3(-20) = ex_4(-20) = 1$ and $ex_3(0) = ex_4(0) = -1$.

From Proposition 2, we establish that there are seven fixed points. Among these, 0 and ∞ come from the roots a and b of $f(x)$. $ex_1 = 1$ comes from the divergence of the original scheme, previous to the Möbius transformation.

Proposition 3. *The strange fixed point $ex_1 = 1$, $\forall \epsilon \in \mathbb{C}$, has the following character:*

(i) *If $|\epsilon - 12| > 32$, then ex_1 is an attractor.*
(ii) *If $|\epsilon - 12| < 32$, then ex_1 is a repulsor.*
(iii) *If $|\epsilon - 12| = 32$, then ex_1 is parabolic.*

Moreover, ex_1 can be attracting but not superattracting. The superattracting fixed points of R_f are $x=0$, $x=\infty$, and the strange fixed points $ex_{4,5}(\epsilon)$ for $\epsilon = \frac{1}{9}\left(-5\sqrt{97}-47\right)$ and $\epsilon = \frac{1}{9}\left(5\sqrt{97}-47\right)$.

In the particular case of $\epsilon = 2$ (using the Equation (24)), all the strange fixed points are repulsive.

Proof. We prove this result by analyzing the stability of the fixed points found in Proposition 2. It must be done by evaluating $\left|R'_f(x,\epsilon)\right|$ at each fixed point and, if it is lower, equal, or greater than one it is called attracting, neutral, or repulsive, respectively.

The cases of $x=0$ and ∞ are straightforward from the expression of $R_f(x,\epsilon)$. When $ex_1(\epsilon)$ is studied, then

$$\left|R'_f(1,\epsilon)\right| = \left|\frac{32}{12-\epsilon}\right|,$$

so it is attracting, repelling or neutral if $|\epsilon - 12|$ is greater, lower, or equal to 32. It can be graphically viewed in Figure 1.

By a graphical and numerical study of $\left|R'_f(ex_i(\epsilon),\epsilon)\right|$, $i = 1,2,3,4$, it can be deduced that $ex_{2,3}(\epsilon)$ are repulsive for all ϵ, meanwhile $ex_{4,5}(\epsilon)$ are superattracting for

$\epsilon = \frac{1}{9}\left(-5\sqrt{97}-47\right) \approx -10.6938$ or $\epsilon = \frac{1}{9}\left(5\sqrt{97}-47\right) \approx 0.249365$. Their stability function is presented in Figure 2a,b. Moreover, ex_1 can not be a superattractor as $\left|R'_f(1,\epsilon)\right| \neq 0$. □

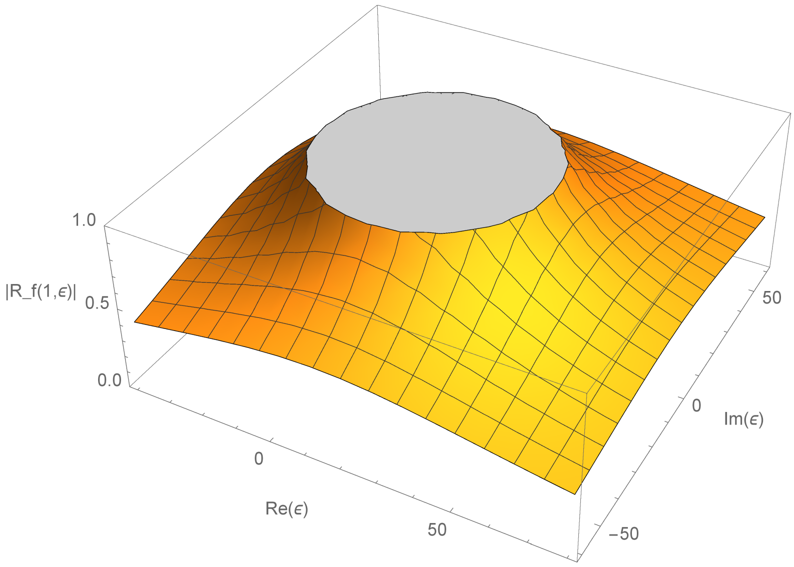
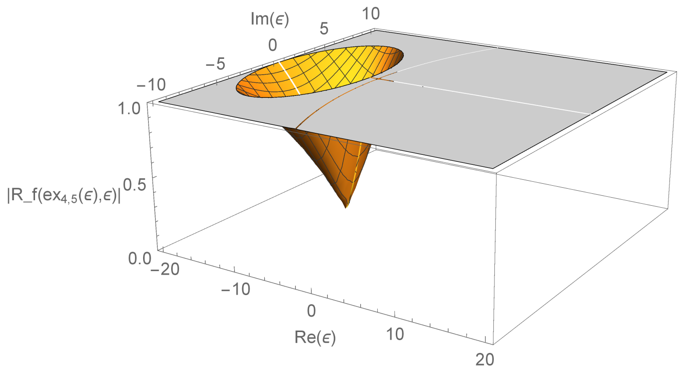
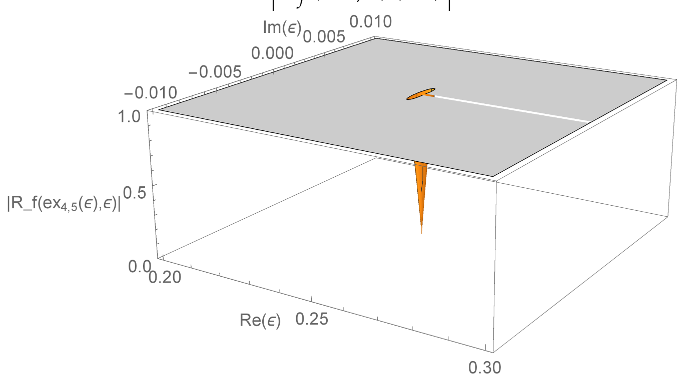


Figure 1. Stability function of $ex_1 = 1$, $\left|R'_f(1,\epsilon)\right|$ for a complex ϵ.



(a) $\left|R'_f(ex_{4,5}(\epsilon),\epsilon)\right|$



(b) $\left|R'_f(ex_{4,5}(\epsilon),\epsilon)\right|$

Figure 2. Stability surfaces of $ex_{4,5}(\epsilon)$ for different complex regions.

It is clear that 0 and ∞ are always superattracting fixed points, but the stability of the remaining fixed points depends on the values of ϵ. According to Proposition 3, two strange fixed points can become superattractors. This implies that there would exist basins of attraction for them, potentially causing the method to fail to converge to the solution. However, even when they are only attracting (that can be the case of ex_1), these basins of attraction exist.

As we have stated previously, Figure 1 represents the stability function of the strange fixed point ex_1. In this figure, the zones of attraction are the yellow area and the repulsion zone corresponds to the grey area. For values of ϵ within the disk, ex_1 is repulsive; whereas for values of ϵ outside the grey disk, ex_1 becomes attracting. So, it is natural to select values within the grey disk, as a repulsive divergence improves the performance of the iterative scheme.

Similar conclusions can be stated from the stability region of strange fixed points $ex_{4,5}(\epsilon)$, appearing in Figure 2. When a value of parameter ϵ is taken in the yellow area of Figure 2, both points are simultaneously attracting, so there are at least four different basins of attraction.

However, the basins of attraction also appear when there exist attracting periodic orbits of any period. To detect this kind of behavior, the role of critical points is crucial.

3.3. Critical Points

Now, we obtain the critical points of $R_f(x, \epsilon)$.

Proposition 4. *The critical points of $R_f(x, \epsilon)$ are $x = 0$, $x = \infty$ and also:*

- $cr_1 = -1$, *and*
- $cr_{2,3}(\epsilon) = \dfrac{2\epsilon + 6 \pm \sqrt{5}\sqrt{12\epsilon - \epsilon^2}}{3(\epsilon - 2)}$.

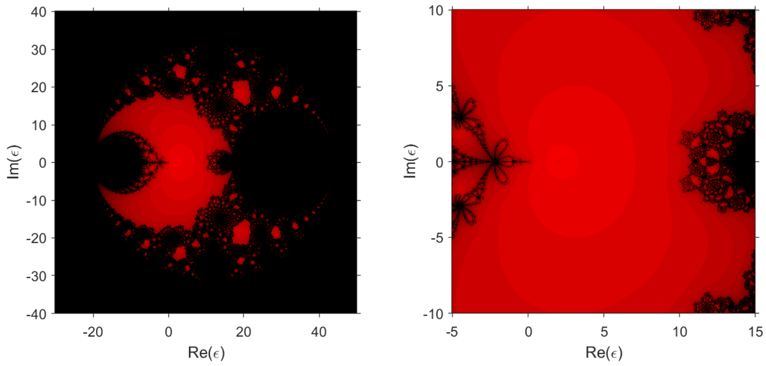
Morover, if $\epsilon = 2$, critical points are not free $cr_{2,3}(2) = 0$. In any other case, $cr_{2,3}(\epsilon)$ are conjugated free critical points.

From Proposition 4, we establish that, in general, there are five critical points. The free critical point $cr_1 = -1$ is a pre-image of the strange fixed point $ex_1 = 1$. Therefore, the stability of cr_1 corresponds to the stability of ex_1 (see Section 3.2). Note that if the Equation (24) is satisfied, the only remaining free critical point is cr_1. Since cr_1 is the pre-image of ex_1, it would be a repulsor.

Then, we use the only independent free critical point $cr_2(\epsilon)$ (conversely, $cr_3(\epsilon)$, as they are conjugates) to generate the parameter plane. This a graphical representation of the global stability performance of the member of the class of iterative methods. In a definite area of the complex plane, a mesh of 500×500 points is generated. Each one of these points is used as a value of parameter ϵ, i.e., we get a particular element of the family. For each one of these values, we get as our initial guess the critical point $cr_2(\epsilon)$ and calculate its orbit. If it converges to $x = 0$ or $x = \infty$, then the point corresponding to this value of ϵ is represented using a red color. In other case, it is left in black. So, convergent schemes to the original roots of the quadratic equations appear in the red stable area and the black area corresponds to schemes of the classes that are not able to converge to them, by reason of an attracting strange fixed point or periodic orbit. This performance can be seen in Figure 3, representing the domain $D_1 = [-30, 50] \times [-40, 40]$, where a wide area of stable performance can be found around the origin, $D_2 = [-5, 15] \times [-10, 10]$ (Figure 3b).

3.4. Dynamical Planes

A dynamical plane is defined as a mesh in a limited domain of the complex plane, where each point corresponds to a different initial estimate x_0. The graphical representation shows the method's convergence starting from x_0 within a maximum of 80 iterations and 10^{-3} as the tolerance. Fixed points appear as a white circle '○', critical points are '□', and a white asterisk '∗' symbolizes an attracting point. Additionally, the basins of attraction are depicted in different colors. To generate this graph, we use MATLAB R2020b with a resolution of 400×400 pixels.

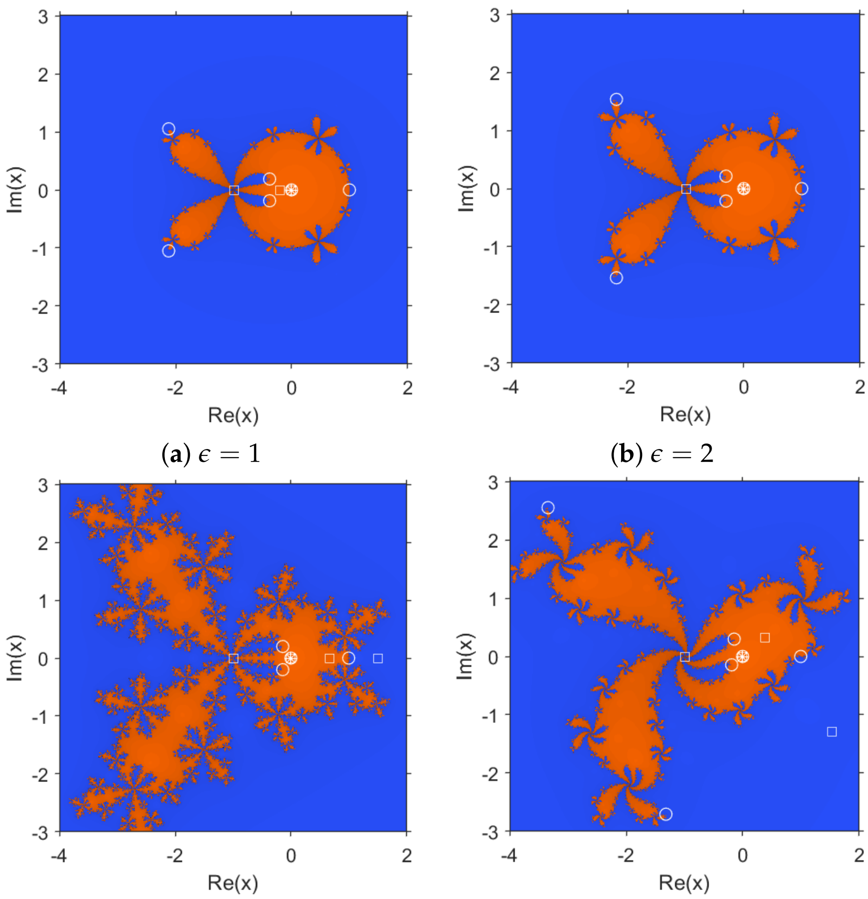


(**a**) D_1 (**b**) D_2

Figure 3. Parameter plane of $cr_2(\epsilon)$ on domain D_1 and a detail on D_2.

Here, we analyze the stability of various MCCTU(ϵ) methods using dynamical planes. We consider methods with ϵ values both inside and outside the stability surface of ex_1, specifically, in the red and black areas of the parameter plane represented in Figure 3a.

Firstly, examples of methods within the stability region are provided for $\epsilon \in \{1, 2, 10, 5 + 5i\}$. Their dynamical planes, along with their respective basins of attraction, are shown in Figure 4. Let us remark that all selected values of ϵ lie in the red area of the parameter plane and have only two basins of attraction, corresponding to $x = 0$ (in orange color in the figures) and $x = \infty$ (blue in the figures).

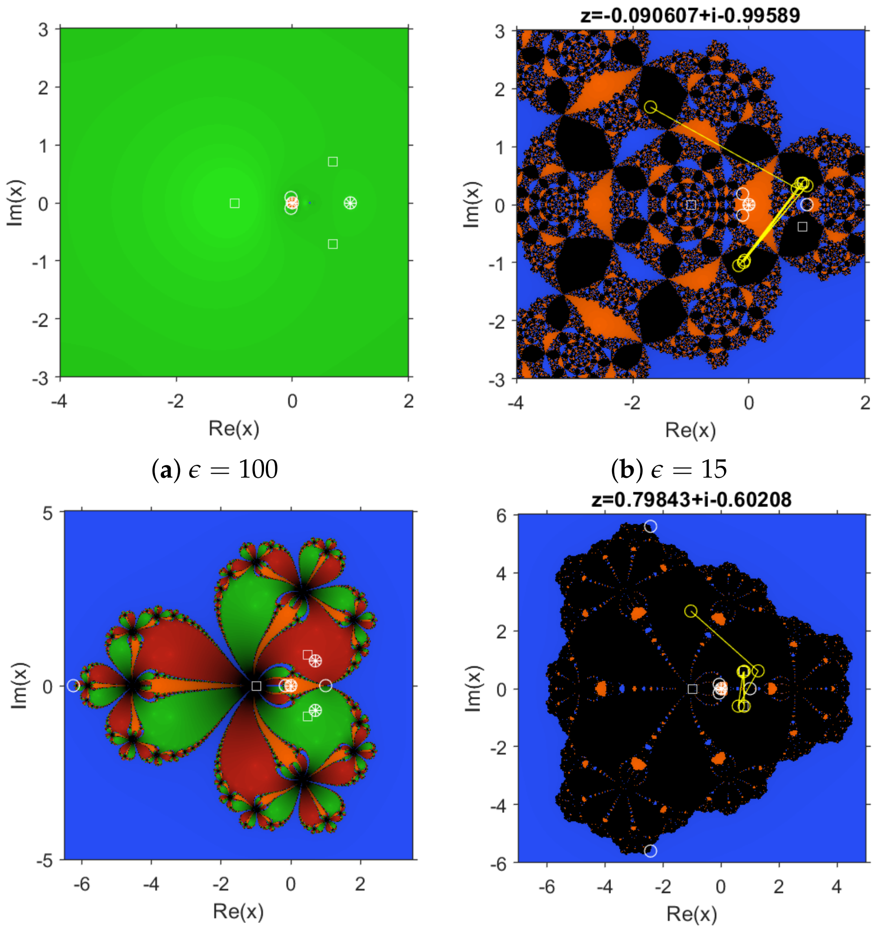


(**a**) $\epsilon = 1$ (**b**) $\epsilon = 2$

(**c**) $\epsilon = 10$ (**d**) $\epsilon = 5 + 5i$

Figure 4. Dynamical planes for some stable methods.

Secondly, some schemes outside the stability region (in black in the parameter plane) are provided for $\epsilon \in \{100, 15, -15, 30\}$. Their dynamical planes are shown in Figure 5. Each of these members have specific characteristics: in Figure 5a, the widest basin of attraction (in green color) corresponds to $ex_1 = 1$, which is attracting for this value of ϵ, the basin of $x = 0$ is a very narrow area around the point; for $\epsilon = 15$, we observe in Figure 5b three different basins of attraction, the third of the two being attracting periodic orbits of period 2 (one of them is plotted in yellow in the figure); Figure 5c corresponds to $\epsilon = -15$, inside the stability area of $ex_{4,5}(\epsilon)$ (see Figure 2), where both are simultaneously attracting; finally, for $\epsilon = 30$, the widest basin of attraction corresponds to an attracting periodic orbit of period 2, see Figure 5d.



(**a**) $\epsilon = 100$ (**b**) $\epsilon = 15$

(**c**) $\epsilon = -15$ (**d**) $\epsilon = 30$

Figure 5. Unstable dynamical planes.

4. Numerical Results

In this section, we conduct several numerical tests to validate the theoretical convergence and stability results of the MCCTU(α) family obtained in previous sections. We use both stable and unstable methods from (20) and apply them to ten nonlinear test equations, with their expressions and corresponding roots provided in Table 1.

We aim to demonstrate the theoretical results by testing the MCCTU(α) family. Specifically, we evaluate three representative members of the family with $\delta = \frac{2}{\alpha^4}$ and $\alpha = 1$, $\alpha = 2$, and $\alpha = 100$. Therefore, in all cases, $\epsilon = 2$.

We conduct two experiments. In the first experiment, we analyze the stability of the MCCTU(α) family using two of its methods, chosen based on stable and unstable values of the parameter α. In the second experiment, we perform an efficiency analysis of the MCCTU(α) family through a comparative study between its optimal stable member and fifteen different fourth-order methods from the literature: Ostrowski (OS) in [12,35], King (KI) in [35,36], Jarratt (JA) in [35,37], Özban and Kaya (OK1, OK2, OK3) in [8], Chun (CH) in [38], Maheshwari (MA) in [39], Behl et al. (BMM) in [40], Chun et al. (CLND1, CLND2) in [41], Artidiello et al. (ACCT1, ACCT2) in [42], Ghanbari (GH) in [43], and Kou et al. (KLW) in [44].

Table 1. Nonlinear test equations and corresponding roots.

Nonlinear Test Equations	Roots
$f_1(x) = \sin(x) - x^2 + 1 = 0$	$\xi \approx -0.63673$
$f_2(x) = x^2 - e^x - 3x + 2 = 0$	$\xi \approx 0.25753$
$f_3(x) = \cos(x) - xe^x + x^2 = 0$	$\xi \approx 0.63915$
$f_4(x) = e^x - 1.5 - \arctan(x) = 0$	$\xi \approx -14.10127$
$f_5(x) = x^3 + 4x^2 - 10 = 0$	$\xi \approx 1.36523$
$f_6(x) = 8x - \cos(x) - 2x^2 = 0$	$\xi \approx 0.12808$
$f_7(x) = xe^{x^2} - \sin^2(x) + 3\cos(x) + 5 = 0$	$\xi \approx -1.20765$
$f_8(x) = \sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3 = 0$	$\xi \approx 2.33197$
$f_9(x) = x^4 + \sin\left(\frac{\pi}{x^2}\right) - 5 = 0$	$\xi \approx -1.41421$
$f_{10}(x) = \sqrt{x^4} + \sin\left(\frac{\pi}{x^2}\right) - \frac{3}{16} = 0$	$\xi \approx -0.90599$

While performing these numerical tests, we start the iterations with different initial estimates: close ($x_0 \approx \xi$), far ($x_0 \approx 3\xi$), and very far ($x_0 \approx 10\xi$) from the root ξ. This approach allows us to evaluate how sensitive the methods are to the initial estimation when finding a solution.

The calculations are performed using the MATLAB R2020b programming package with variable precision arithmetic set to 200 digits of mantissa (in Appendix C, an example with double-precision arithmetics is included). For each method, we analyze the number of iterations (iter) required to converge to the solution, with stopping criteria defined as $|x_{k+1} - x_k| < 10^{-100}$ or $|f(x_{k+1})| < 10^{-100}$. Here, $|x_{k+1} - x_k|$ represents the error estimation between two consecutive iterations, and $|f(x_{k+1})|$ is the residual error of the nonlinear test function.

To check the theoretical order of convergence (p), we calculate the approximate computational order of convergence (ACOC) as described by Cordero and Torregrosa in [15]. In the numerical results, if the ACOC values do not stabilize throughout the iterative process, it is marked as '-'; and if any method fails to converge within a maximum of 50 iterations, it is marked as 'nc'.

4.1. First Experiment: Stability Analysis of MCCTU(α) Family

In this experiment, we conducted a stability analysis of the MCCTU(α) family by considering values of α both within the stability regions ($\alpha = 2$) and outside of them ($\alpha = 100$), setting $\delta = \frac{2}{\alpha^4}$. The methods analyzed are of order 3, consistent with the theoretical convergence results. A special case occurs when $\alpha = 0$, where the associated method never converges to the solution because the denominator in the relation $\delta = 2/\alpha^4$ becomes zero, causing δ to grow indefinitely.

The numerical performance of the iterative methods MCCTU(2) and MCCTU(100) is presented in Tables 2 and 3, using initial estimates that are close, far, and very far from the root. This approach enables us to assess the stability and reliability of the methods under various initial conditions.

Table 2. Numerical performance of MCCTU(2) method on nonlinear equations ("nc" means non-convergence).

Function	x_0	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
	Close to ξ				
f_1	-0.6	2.2252×10^{-54}	1.4765×10^{-162}	4	3
f_2	0.2	1.8447×10^{-50}	1.3536×10^{-150}	4	3
f_3	0.6	2.3846×10^{-44}	1.4235×10^{-131}	4	3
f_4	-14.1	5.1414×10^{-36}	3.3633×10^{-111}	3	3
f_5	1.3	1.6295×10^{-53}	4.3267×10^{-159}	4	3
f_6	0.1	4.6096×10^{-78}	2.4334×10^{-208}	4	3
f_7	-1.2	3.6237×10^{-54}	1.9349×10^{-159}	4	3
f_8	2.3	3.0861×10^{-54}	6.9791×10^{-162}	4	3
f_9	-1.4	7.0858×10^{-51}	3.8746×10^{-150}	4	3
f_{10}	-0.9	8.9456×10^{-45}	9.7874×10^{-131}	4	3
	Far from ξ				
f_1	-1.8	1.5223×10^{-92}	0	5	3
f_2	0.6	6.6012×10^{-87}	0	5	3
f_3	1.8	3.8851×10^{-45}	6.1565×10^{-134}	6	3
f_4	-42.3	nc	nc	nc	nc
f_5	3.9	1.0792×10^{-59}	1.2569×10^{-177}	6	3
f_6	0.3	1.0805×10^{-48}	2.6855×10^{-146}	4	3
f_7	-3.6	2.2394×10^{-55}	4.5662×10^{-163}	14	3
f_8	6.9	1.1722×10^{-41}	3.8248×10^{-124}	6	3
f_9	-4.2	1.3408×10^{-101}	0	8	3
f_{10}	-2.7	4.3149×10^{-78}	3.1147×10^{-207}	8	3
	Very far from ξ				
f_1	-6.0	1.5491×10^{-52}	4.9812×10^{-157}	6	3
f_2	2.0	1.6192×10^{-89}	0	6	3
f_3	6.0	7.1447×10^{-57}	3.8290×10^{-169}	10	3
f_4	-141.0	nc	nc	nc	nc
f_5	13.0	1.6531×10^{-82}	0	8	3
f_6	1.0	1.6423×10^{-56}	9.4291×10^{-170}	5	3
f_7	-12.0	nc	nc	nc	nc
f_8	23.0	1.2648×10^{-44}	4.8043×10^{-133}	7	3
f_9	-14.0	2.3358×10^{-43}	1.3880×10^{-127}	10	3
f_{10}	-9.0	3.0298×10^{-44}	1.2080×10^{-128}	6	3

From the analysis of the first experiment, it is evident that the MCCTU(2) method exhibits robust performance. For initial estimates close to the root ($x_0 \approx \xi$), the method consistently converges to the solution with very low errors, achieving convergence in three or four iterations, and the ACOC value stabilizes at 3. For initial estimates that are far ($x_0 \approx 3\xi$), the number of iterations increases, but the method still converges to the solution in nine out of ten cases. For initial estimates that are very far ($x_0 \approx 10\xi$), the method holds a similar performance, converging to the solution in eight out of ten cases. It is notable that as the initial condition moves further away, the method shows a slight difficulty in finding the solution. This slight dependence is understandable given the complexity of the nonlinear functions f_4 and f_7. Nonetheless, the method is shown to be stable and robust, with a convergence order of 3, verifying the theoretical results.

On the other hand, MCCTU(100) method encounters significant difficulties in finding the solution. As the initial conditions move further away, the number of iterations increases. Despite lacking good stability characteristics, the method converges to the solution for initial estimates close to the root. However, for initial estimates that are far and very far from the root, it fails to converge in four out of ten cases. Additionally, the method never stabilizes the ACOC value in any case. These results confirm the theoretical instability of the method, as $\alpha = 100$ lies outside the stability surface studied in Section 3.

Table 3. Numerical performance of MCCTU(100) method on nonlinear equations ("nc" means non-convergence).

Function	x_0	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
	Close to ξ				
f_1	-0.6	6.1808×10^{-99}	7.6768×10^{-113}	9	-
f_2	0.2	2.1827×10^{-88}	4.9309×10^{-102}	9	-
f_3	0.6	6.0791×10^{-94}	8.8104×10^{-108}	9	-
f_4	-14.1	4.5379×10^{-95}	1.3573×10^{-111}	8	-
f_5	1.3	4.9631×10^{-94}	4.8998×10^{-107}	9	-
f_6	0.1	3.0953×10^{-100}	1.4092×10^{-113}	9	-
f_7	-1.2	8.7126×10^{-95}	1.0578×10^{-107}	9	-
f_8	2.3	2.1622×10^{-95}	3.1373×10^{-109}	9	-
f_9	-1.4	4.0458×10^{-95}	2.7366×10^{-108}	9	-
f_{10}	-0.9	6.2830×10^{-95}	3.1368×10^{-108}	9	-
	Far from ξ				
f_1	-1.8	2.7746×10^{-92}	3.4462×10^{-106}	10	-
f_2	0.6	6.8191×10^{-99}	1.5405×10^{-112}	10	-
f_3	1.8	8.0835×10^{-90}	1.1715×10^{-103}	12	-
f_4	-42.3	nc	nc	nc	nc
f_5	3.9	nc	nc	nc	nc
f_6	0.3	4.0669×10^{-95}	1.8516×10^{-108}	9	-
f_7	-3.6	nc	nc	nc	nc
f_8	6.9	1.5980×10^{-88}	2.3186×10^{-102}	11	-
f_9	-4.2	nc	nc	nc	nc
f_{10}	-2.7	1.5127×10^{-97}	3.0929×10^{-110}	11	-
	Very far from ξ				
f_1	-6.0	1.2947×10^{-94}	1.6081×10^{-108}	11	-
f_2	2.0	3.5429×10^{-94}	8.0036×10^{-108}	11	-
f_3	6.0	4.5426×10^{-97}	6.5836×10^{-111}	18	-
f_4	-141.0	nc	nc	nc	nc
f_5	13.0	nc	nc	nc	nc
f_6	1.0	1.4843×10^{-94}	6.7580×10^{-108}	10	-
f_7	-12.0	nc	nc	nc	nc
f_8	23.0	7.4725×10^{-92}	1.0842×10^{-105}	12	-
f_9	-14.0	nc	nc	nc	nc
f_{10}	-9.0	$6.5629\text{e} \times 10^{-95}$	3.2765×10^{-108}	12	-

4.2. Second Experiment: Efficiency Analysis of MCCTU(α) Family

In this experiment, we conduct a comparative study between an optimal method of the MCCTU(α) family and the fifteen fourth-order methods mentioned in the introduction of Section 4, to contrast their numerical performances on nonlinear equations. We consider the method associated with $\alpha = 1$ and $\delta = 2$, denoted as MCCTU(1), as the optimal stable member of the MCCTU(α) family with fourth-order of convergence.

Thus, in Tables 4–14, we present the numerical results for the sixteen known methods, considering initial estimates that are close, far, and very far from the root, as well as the ten test equations.

Table 4. Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (1/4).

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
f_1	MCCTU(1)	8.4069×10^{-27}	2.2344×10^{-105}	3	4.0111
$x_0 = -0.6$	OS	1.2193×10^{-29}	2.7787×10^{-117}	3	4.0062
	KI	3.9435×10^{-29}	3.8183×10^{-115}	3	4.0070
	JA	1.3498×10^{-29}	4.2651×10^{-117}	3	4.0061
	OK1	5.0547×10^{-32}	2.9443×10^{-127}	3	3.9991
	OK2	4.0266×10^{-30}	2.6729×10^{-119}	3	4.0052
	OK3	2.5735×10^{-30}	4.5908×10^{-120}	3	3.9937
	CH	1.6691×10^{-28}	1.6213×10^{-112}	3	4.0081
	MA	3.0371×10^{-27}	3.1217×10^{-107}	3	4.0103
	BMM	1.2299×10^{-28}	4.4824×10^{-113}	3	4.0084

Table 4. *Cont.*

Function	Method	$\vert x_{k+1} - x_k \vert$	$\vert f(x_{k+1}) \vert$	Iter	ACOC
	CLND1	8.643×10^{-27}	2.5116×10^{-105}	3	4.0110
	CLND2	1.6691×10^{-28}	1.6213×10^{-112}	3	4.0081
	ACCT1	8.4069×10^{-27}	2.2344×10^{-105}	3	4.0111
	ACCT2	7.4417×10^{-32}	1.0756×10^{-126}	3	4.0294
	GH	1.9739×10^{-26}	8.0112×10^{-104}	3	4.0119
	KLW	8.4441×10^{-28}	1.4567×10^{-109}	3	4.0092
f_2	MCCTU(1)	4.0916×10^{-36}	1.3257×10^{-144}	3	3.9624
$x_0 = 0.2$	OS	2.6718×10^{-32}	8.6963×10^{-129}	3	3.9998
	KI	1.7333×10^{-32}	1.4291×10^{-129}	3	3.9987
	JA	1.1553×10^{-31}	4.1074×10^{-126}	3	3.9990
	OK1	2.4295×10^{-31}	9.1464×10^{-125}	3	4.0008
	OK2	1.4863×10^{-31}	1.1754×10^{-125}	3	3.9997
	OK3	1.3844×10^{-31}	8.8054×10^{-126}	3	3.9988
	CH	5.002×10^{-32}	1.2502×10^{-127}	3	3.9969
	MA	2.1425×10^{-34}	1.6464×10^{-137}	3	3.9844
	BMM	5.5838×10^{-31}	2.8585×10^{-123}	3	4.0057
	CLND1	9.7338×10^{-34}	9.6229×10^{-135}	3	3.9830
	CLND2	5.002×10^{-32}	1.2502×10^{-127}	3	3.9969
	ACCT1	4.0916×10^{-36}	1.3257×10^{-144}	3	3.9624
	ACCT2	1.4832×10^{-31}	1.1243×10^{-125}	3	4.0029
	GH	2.5248×10^{-38}	6.6868×10^{-154}	3	3.9675
	KLW	1.8553×10^{-33}	1.2914×10^{-133}	3	3.9925
f_3	MCCTU(1)	2.2096×10^{-83}	0	4	4
$x_0 = 0.6$	OS	1.7622×10^{-27}	3.3439×10^{-108}	3	3.9992
	KI	1.6297×10^{-100}	0	4	4
	JA	2.9708×10^{-27}	2.989×10^{-107}	3	3.9996
	OK1	6.1743×10^{-100}	1.9467×10^{-208}	4	4
	OK2	1.0137×10^{-33}	6.7493×10^{-135}	3	4.0975
	OK3	1.0148×10^{-27}	3.9188×10^{-110}	3	4.2357
	CH	2.1262×10^{-94}	0	4	4
	MA	6.9765×10^{-86}	0	4	4
	BMM	1.6076×10^{-85}	1.9467×10^{-208}	4	4
	CLND1	2.5512×10^{-83}	0	4	4
	CLND2	2.1262×10^{-94}	0	4	4
	ACCT1	2.2096×10^{-83}	6.8135×10^{-208}	4	4
	ACCT2	2.5202×10^{-91}	0	4	4
	GH	2.2217×10^{-81}	0	4	4
	KLW	2.4336×10^{-89}	0	4	4

Table 5. Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (2/4).

Function	Method	$\vert x_{k+1} - x_k \vert$	$\vert f(x_{k+1}) \vert$	Iter	ACOC
f_4	MCCTU(1)	2.4812×10^{-61}	0	3	4
$x_0 = -14.1$	OS	5.7494×10^{-76}	0	3	4
	KI	2.6178×10^{-66}	0	3	4
	JA	4.5662×10^{-69}	3.8934×10^{-208}	3	4
	OK1	1.6181×10^{-64}	0	3	4
	OK2	1.2341×10^{-67}	0	3	4
	OK3	4.782×10^{-68}	3.8934×10^{-208}	3	3.9998
	CH	4.1273×10^{-64}	0	3	4
	MA	5.9003×10^{-62}	0	3	4
	BMM	2.4555×10^{-61}	3.8934×10^{-208}	3	4
	CLND1	2.8374×10^{-61}	0	3	4
	CLND2	4.1273×10^{-64}	0	3	4
	ACCT1	2.4812×10^{-61}	0	3	4
	ACCT2	7.6144×10^{-63}	0	3	4
	GH	7.562×10^{-61}	0	3	4

Table 5. *Cont.*

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
	KLW	7.8025×10^{-63}	0	3	4
f_5	MCCTU(1)	1.5146×10^{-80}	0	4	4
$x_0 = 1.3$	OS	4.0399×10^{-98}	0	4	4
	KI	4.6142×10^{-94}	0	4	4
	JA	4.0399×10^{-98}	0	4	4
	OK1	1.6263×10^{-26}	3.9339×10^{-104}	3	4.0265
	OK2	3.7251×10^{-26}	1.5538×10^{-102}	3	4.0049
	OK3	2.6244×10^{-29}	4.1697×10^{-115}	3	3.8563
	CH	5.0966×10^{-90}	0	4	4
	MA	8.4188×10^{-83}	0	4	4
	BMM	8.6757×10^{-85}	0	4	4
	CLND1	1.5146×10^{-80}	0	4	4
	CLND2	5.0966×10^{-90}	0	4	4
	ACCT1	1.5146×10^{-80}	0	4	4
	ACCT2	1.0557×10^{-91}	0	4	4
	GH	1.0682×10^{-78}	0	4	4
	KLW	8.3547×10^{-86}	0	4	4
f_6	MCCTU(1)	1.1439×10^{-32}	5.0948×10^{-129}	3	3.9969
$x_0 = 0.1$	OS	5.058×10^{-36}	4.1154×10^{-143}	3	3.9980
	KI	2.4554×10^{-35}	3.1386×10^{-140}	3	3.9979
	JA	7.2379×10^{-36}	1.8516×10^{-142}	3	3.9981
	OK1	8.6178×10^{-41}	3.6529×10^{-163}	3	4.0021
	OK2	1.3158×10^{-36}	1.4361×10^{-145}	3	3.9982
	OK3	2.2299×10^{-36}	1.2384×10^{-144}	3	4.0031
	CH	1.6189×10^{-34}	8.6635×10^{-137}	3	3.9977
	MA	3.8478×10^{-33}	5.2367×10^{-131}	3	3.9971
	BMM	7.9902×10^{-34}	7.003×10^{-134}	3	3.9985
	CLND1	1.2375×10^{-32}	7.0855×10^{-129}	3	3.9970
	CLND2	1.6189×10^{-34}	8.6635×10^{-137}	3	3.9977
	ACCT1	1.1439×10^{-32}	5.0948×10^{-129}	3	3.9969
	ACCT2	2.0595×10^{-36}	9.7992×10^{-145}	3	3.9952
	GH	2.7869×10^{-32}	2.1488×10^{-127}	3	3.9967
	KLW	9.5356×10^{-34}	1.49×10^{-133}	3	3.9974

Table 6. Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (3/4).

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
f_7	MCCTU(1)	8.109×10^{-29}	1.224×10^{-110}	3	4.0025
$x_0 = -1.2$	OS	1.0259×10^{-36}	8.588×10^{-144}	3	3.9987
	KI	2.2×10^{-33}	8.266×10^{-130}	3	4.0003
	JA	1.3275×10^{-35}	3.987×10^{-139}	3	3.9993
	OK1	2.8995×10^{-32}	4.1375×10^{-125}	3	4.0011
	OK2	4.5559×10^{-36}	4.3528×10^{-141}	3	4.0014
	OK3	3.3899×10^{-35}	9.9763×10^{-138}	3	4.0602
	CH	1.5282×10^{-31}	4.4541×10^{-122}	3	4.0011
	MA	1.9806×10^{-29}	3.2969×10^{-113}	3	4.0021
	BMM	5.13×10^{-29}	1.8531×10^{-111}	3	3.9988
	CLND1	8.8475×10^{-29}	1.7657×10^{-110}	3	4.0024
	CLND2	1.5282×10^{-31}	4.4541×10^{-122}	3	4.0011
	ACCT1	8.109×10^{-29}	1.224×10^{-110}	3	4.0025
	ACCT2	1.7685×10^{-30}	1.2708×10^{-117}	3	4.0037
	GH	2.4542×10^{-28}	1.2766×10^{-108}	3	4.0029
	KLW	2.7616×10^{-30}	8.4579×10^{-117}	3	4.0014
f_8	MCCTU(1)	3.2362×10^{-36}	1.278×10^{-144}	3	4.0010
$x_0 = 2.3$	OS	4.7781×10^{-35}	1.1082×10^{-139}	3	3.9959
	KI	3.867×10^{-35}	4.5395×10^{-140}	3	3.9962
	JA	6.4886×10^{-36}	2.6103×10^{-143}	3	3.9934

Table 6. *Cont.*

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	Iter	ACOC
	OK1	1.3631×10^{-35}	5.9439×10^{-142}	3	3.9927
	OK2	8.3354×10^{-36}	7.4954×10^{-143}	3	3.9931
	OK3	8.2958×10^{-36}	7.3117×10^{-143}	3	3.9935
	CH	2.8017×10^{-36}	7.5934×10^{-145}	3	3.9943
	MA	7.3689×10^{-36}	4.1437×10^{-143}	3	3.9992
	BMM	2.6822×10^{-34}	1.5979×10^{-136}	3	3.9934
	CLND1	5.1248×10^{-38}	3.528×10^{-152}	3	4.0005
	CLND2	2.8017×10^{-36}	7.5934×10^{-145}	3	3.9943
	ACCT1	3.2362×10^{-36}	1.278×10^{-144}	3	4.0010
	ACCT2	1.2316×10^{-34}	5.9984×10^{-138}	3	3.9946
	GH	1.2035×10^{-36}	1.9404×10^{-146}	3	4.0036
	KLW	1.4928×10^{-35}	8.1719×10^{-142}	3	3.9978
f_9	MCCTU(1)	1.2504×10^{-28}	6.0286×10^{-111}	3	3.9982
$x_0 = -1.4$	OS	2.2297×10^{-33}	5.9539×10^{-131}	3	4.0107
	KI	3.571×10^{-39}	4.8453×10^{-155}	3	3.9663
	JA	6.6365×10^{-33}	5.9006×10^{-129}	3	4.0095
	OK1	1.7043×10^{-30}	8.4881×10^{-119}	3	4.0019
	OK2	8.1242×10^{-32}	2.3078×10^{-124}	3	4.0049
	OK3	1.3061×10^{-31}	1.4689×10^{-123}	3	4.0184
	CH	5.6961×10^{-33}	3.922×10^{-129}	3	3.9887
	MA	2.4063×10^{-29}	5.9988×10^{-114}	3	3.9973
	BMM	2.911×10^{-28}	2.1169×10^{-109}	3	3.9971
	CLND1	1.0887×10^{-28}	3.3751×10^{-111}	3	3.9980
	CLND2	5.6961×10^{-33}	3.922×10^{-129}	3	3.9887
	ACCT1	1.2504×10^{-28}	6.0286×10^{-111}	3	3.9982
	ACCT2	1.7434×10^{-29}	1.473×10^{-114}	3	4.0025
	GH	4.3546×10^{-28}	1.1301×10^{-108}	3	3.9989
	KLW	2.0248×10^{-30}	1.8702×10^{-118}	3	3.9955

Table 7. Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (4/4).

Function	Method	$\|x_{k+1} - x_k\|$	1c$\|f(x_{k+1})\|$	Iter	ACOC
f_{10}	MCCTU(1)	1.3096×10^{-27}	1.0557×10^{-105}	3	4.0263
$x_0 = -0.9$	OS	1.2157×10^{-28}	5.2236×10^{-110}	3	4.0178
	KI	1.6268×10^{-28}	1.7588×10^{-109}	3	4.0189
	JA	2.5808×10^{-28}	1.2592×10^{-108}	3	4.0158
	OK1	1.2566×10^{-28}	6.3023×10^{-110}	3	4.0126
	OK2	2.0733×10^{-28}	5.0608×10^{-109}	3	4.0149
	OK3	1.9638×10^{-28}	4.0898×10^{-109}	3	4.0133
	CH	4.7545×10^{-28}	1.6033×10^{-107}	3	4.0184
	MA	7.9356×10^{-28}	1.3045×10^{-106}	3	4.0246
	BMM	1.3934×10^{-30}	4.5027×10^{-118}	3	3.9969
	CLND1	2.1208×10^{-27}	8.164×10^{-105}	3	4.0242
	CLND2	4.7545×10^{-28}	1.6033×10^{-107}	3	4.0184
	ACCT1	1.3096×10^{-27}	1.0557×10^{-105}	3	4.0263
	ACCT2	1.9256×10^{-29}	2.4651×10^{-113}	3	4.0090
	GH	2.0723×10^{-27}	7.171×10^{-105}	3	4.0278
	KLW	4.546×10^{-28}	1.2771×10^{-107}	3	4.0226

In Tables 4–7, we observe that MCCTU(1) consistently converges to the solution for initial estimates close to the root ($x_0 \approx \xi$), with a similar number of iterations as other methods across all equations. The theoretical convergence order is confirmed by the ACOC, which is close to 4. However, what about the dependence of MCCTU(1) on initial estimates? To answer this, we analyze the method for initial estimates far and very far from the solution, specifically for $x_0 \approx 3\xi$ and $x_0 \approx 10\xi$, respectively. The results are shown in Tables 8–15.

Table 8. Numerical performance of iterative methods on nonlinear equations for x_0 **far from** ξ **("nc" means non-convergence)**. (1/4)

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
f_1	MCCTU(1)	3.15×10^{-28}	4.4044×10^{-111}	4	3.9913
$x_0 = -1.8$	OS	5.8375×10^{-36}	1.46×10^{-142}	4	3.9979
	KI	1.5765×10^{-34}	9.7538×10^{-137}	4	3.9972
	JA	5.3832×10^{-35}	1.0789×10^{-138}	4	3.9976
	OK1	9.9392×10^{-40}	4.4016×10^{-158}	4	4.0001
	OK2	2.4525×10^{-36}	3.6785×10^{-144}	4	3.9982
	OK3	1.1878×10^{-33}	2.0829×10^{-133}	4	4.0017
	CH	2.7866×10^{-32}	1.2595×10^{-127}	4	3.9958
	MA	2.0068×10^{-29}	5.9514×10^{-116}	4	3.9929
	BMM	1.3126×10^{-31}	5.814×10^{-125}	4	4.0050
	CLND1	7.4349×10^{-28}	1.3753×10^{-109}	4	3.9907
	CLND2	2.7866×10^{-32}	1.2595×10^{-127}	4	3.9958
	ACCT1	3.15×10^{-28}	4.4044×10^{-111}	4	3.9913
	ACCT2	6.7276×10^{-44}	7.1842×10^{-175}	4	3.9954
	GH	2.7189×10^{-27}	2.8837×10^{-107}	4	3.9896
	KLW	7.9363×10^{-31}	1.1367×10^{-121}	4	3.9946
f_2	MCCTU(1)	6.8509×10^{-86}	0	4	4
$x_0 = 0.6$	OS	7.8707×10^{-82}	0	4	4
	KI	4.1628×10^{-82}	0	4	4
	JA	5.9451×10^{-78}	7.7869×10^{-208}	4	4
	OK1	1.7391×10^{-77}	7.7869×10^{-208}	4	4
	OK2	8.4717×10^{-78}	0	4	4
	OK3	8.9827×10^{-78}	0	4	4
	CH	1.8951×10^{-78}	0	4	4
	MA	1.8733×10^{-84}	0	4	4
	BMM	1.2206×10^{-79}	0	4	4
	CLND1	2.1212×10^{-80}	0	4	4
	CLND2	1.8951×10^{-78}	0	4	4
	ACCT1	6.8509×10^{-86}	0	4	4
	ACCT2	1.366×10^{-80}	0	4	4
	GH	1.4879×10^{-88}	0	4	4
	KLW	2.1154×10^{-83}	0	4	4
f_3	MCCTU(1)	6.0868×10^{-31}	6.9016×10^{-121}	5	3.9978
$x_0 = 1.8$	OS	7.2812×10^{-73}	0	5	4
	KI	8.2846×10^{-59}	0	5	4
	JA	6.0259×10^{-71}	0	5	4
	OK1	1.1879×10^{-82}	0	5	4
	OK2	6.2111×10^{-27}	9.5138×10^{-108}	4	4.1522
	OK3	7.5783×10^{-53}	0	5	4.0205
	CH	9.6923×10^{-49}	1.3715×10^{-192}	5	3.9999
	MA	1.3275×10^{-34}	1.1979×10^{-135}	5	3.9989
	BMM	nc	nc	nc	nc
	CLND1	9.5034×10^{-31}	4.1315×10^{-120}	5	3.9978
	CLND2	9.6923×10^{-49}	1.3715×10^{-192}	5	3.9999
	ACCT1	6.0868×10^{-31}	6.9016×10^{-121}	5	3.9978
	ACCT2	6.1271×10^{-31}	2.8102×10^{-121}	4	3.9953
	GH	1.4039×10^{-28}	2.4077×10^{-111}	5	3.9965
	KLW	4.5965×10^{-39}	1.1996×10^{-153}	5	3.9996

Table 9. Numerical performance of iterative methods on nonlinear equations for x_0 **far from** ξ ("nc" means non-convergence) (2/4).

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
f_4	MCCTU(1)	nc	nc	nc	nc
$x_0 = -42.3$	OS	2.602×10^{-54}	0	6	4.0004
	KI	nc	nc	nc	nc
	JA	1.0645×10^{-51}	0	6	4

Table 9. *Cont.*

Function	Method	$\lvert x_{k+1}-x_k\rvert$	$\lvert f(x_{k+1})\rvert$	Iter	ACOC
	OK1	nc	nc	nc	nc
	OK2	nc	nc	nc	nc
	OK3	nc	nc	nc	nc
	CH	nc	nc	nc	nc
	MA	nc	nc	nc	nc
	BMM	nc	nc	nc	nc
	CLND1	nc	nc	nc	nc
	CLND2	nc	nc	nc	nc
	ACCT1	nc	nc	nc	nc
	ACCT2	nc	nc	nc	nc
	GH	nc	nc	nc	nc
	KLW	nc	nc	nc	nc
f_5	MCCTU(1)	5.1192×10^{-33}	6.3445×10^{-129}	5	3.9976
$x_0 = 3.9$	OS	1.8922×10^{-60}	0	5	4
	KI	1.6925×10^{-53}	0	5	3.9999
	JA	1.8922×10^{-60}	0	5	4
	OK1	4.3746×10^{-85}	0	5	4
	OK2	8.1261×10^{-70}	0	5	4
	OK3	8.7491×10^{-49}	5.1503×10^{-193}	5	4.0015
	CH	1.68×10^{-47}	2.7094×10^{-187}	5	3.9998
	MA	3.351×10^{-36}	9.1961×10^{-142}	5	3.9986
	BMM	nc	nc	nc	nc
	CLND1	5.1192×10^{-33}	6.3445×10^{-129}	5	3.9976
	CLND2	1.68×10^{-47}	2.7094×10^{-187}	5	3.9998
	ACCT1	5.1192×10^{-33}	6.3445×10^{-129}	5	3.9976
	ACCT2	1.7037×10^{-75}	0	5	4
	GH	8.0066×10^{-31}	4.5963×10^{-120}	5	3.9964
	KLW	5.3477×10^{-40}	4.373×10^{-157}	5	3.9993
f_6	MCCTU(1)	2.8249×10^{-77}	1.2167×10^{-208}	4	4
$x_0 = 0.3$	OS	4.615×10^{-92}	1.2167×10^{-208}	4	4
	KI	4.375×10^{-89}	1.2167×10^{-208}	4	4
	JA	1.7544×10^{-91}	1.2167×10^{-208}	4	4
	OK1	3.5822×10^{-29}	1.0907×10^{-116}	3	3.9593
	OK2	1.0602×10^{-94}	1.2167×10^{-208}	4	4
	OK3	1.3907×10^{-101}	1.2167×10^{-208}	4	4
	CH	1.6778×10^{-85}	1.2167×10^{-208}	4	4
	MA	2.2127×10^{-79}	1.2167×10^{-208}	4	4
	BMM	3.3893×10^{-83}	1.2167×10^{-208}	4	4
	CLND1	3.6933×10^{-77}	1.2167×10^{-208}	4	4
	CLND2	1.6778×10^{-85}	1.2167×10^{-208}	4	4
	ACCT1	2.8249×10^{-77}	2.4334×10^{-208}	4	4
	ACCT2	3.1138×10^{-91}	1.2167×10^{-208}	4	4
	GH	1.6004×10^{-75}	1.2167×10^{-208}	4	4
	KLW	3.9889×10^{-82}	1.2167×10^{-208}	4	4

Table 10. Numerical performance of iterative methods in nonlinear equations for x_0 **far from** ξ ("nc" means non-convergence) (3/4).

Function	Method	$\lvert x_{k+1}-x_k\rvert$	$\lvert f(x_{k+1})\rvert$	Iter	ACOC
f_7	MCCTU(1)	2.1695×10^{-40}	6.2709×10^{-157}	12	3.9997
$x_0 = -3.6$	OS	8.7445×10^{-42}	4.5328×10^{-164}	9	4.0005
	KI	3.5832×10^{-59}	0	10	4
	JA	1.445×10^{-33}	5.5984×10^{-131}	9	4.0010
	OK1	4.5904×10^{-56}	0	9	4
	OK2	2.2993×10^{-100}	0	9	4
	OK3	4.4822×10^{-55}	0	11	3.9955
	CH	2.5759×10^{-93}	0	11	4
	MA	1.0752×10^{-70}	0	12	4
	BMM	nc	nc	nc	nc
	CLND1	5.1735×10^{-38}	2.0643×10^{-147}	12	3.9995
	CLND2	2.5759×10^{-93}	0	11	4

Table 10. *Cont.*

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
	ACCT1	2.1695×10^{-40}	6.2709×10^{-157}	12	3.9997
	ACCT2	7.4341×10^{-57}	0	8	4
	GH	2.2938×10^{-28}	9.7412×10^{-109}	12	3.9971
	KLW	6.4489×10^{-31}	2.515×10^{-119}	11	3.9988
f_8 $x_0 = 6.9$	MCCTU(1)	8.9717×10^{-34}	7.5485×10^{-135}	5	3.9964
	OS	3.5465×10^{-42}	3.3638×10^{-168}	5	3.9988
	KI	3.0788×10^{-46}	1.8241×10^{-184}	5	3.9994
	JA	3.8134×10^{-44}	3.1142×10^{-176}	5	3.9984
	OK1	4.9365×10^{-41}	1.0225×10^{-163}	5	3.9972
	OK2	1.6379×10^{-42}	1.1175×10^{-169}	5	3.9979
	OK3	9.2522×10^{-52}	1.0123×10^{-206}	5	4.0004
	CH	6.7803×10^{-74}	1.5574×10^{-207}	5	4
	MA	4.1752×10^{-36}	4.2707×10^{-144}	5	4.0002
	BMM	1.4528×10^{-98}	1.5574×10^{-207}	5	4
	CLND1	2.6243×10^{-36}	2.4259×10^{-145}	5	3.9960
	CLND2	6.7803×10^{-74}	1.5574×10^{-207}	5	4
	ACCT1	8.9717×10^{-34}	7.5485×10^{-135}	5	3.9964
	ACCT2	7.0924×10^{-38}	6.5962×10^{-151}	5	3.9970
	GH	9.3051×10^{-33}	6.9333×10^{-131}	5	3.9867
	KLW	1.1619×10^{-39}	2.9996×10^{-158}	5	4.0009
f_9 $x_0 = -4.2$	MCCTU(1)	1.9461×10^{-27}	3.5371×10^{-106}	6	4.0014
	OS	4.9716×10^{-100}	0	6	4
	KI	2.3907×10^{-73}	0	6	4.0003
	JA	1.5785×10^{-91}	0	6	4
	OK1	7.3314×10^{-101}	0	6	4
	OK2	5.2967×10^{-94}	0	6	4
	OK3	3.3512×10^{-48}	6.3655×10^{-190}	6	3.9987
	CH	1.4014×10^{-54}	0	6	4.0003
	MA	2.5559×10^{-32}	7.6354×10^{-126}	6	4.0012
	BMM	nc	nc	nc	nc
	CLND1	1.866×10^{-27}	2.9131×10^{-106}	6	4.0016
	CLND2	1.4014×10^{-54}	0	6	4.0003
	ACCT1	1.9461×10^{-27}	3.5371×10^{-106}	6	4.0014
	ACCT2	7.4372×10^{-40}	4.8779×10^{-156}	5	3.9995
	GH	3.5153×10^{-95}	0	7	4
	KLW	1.7919×10^{-38}	1.147×10^{-150}	6	4.0009

Table 11. Numerical performance of iterative methods on nonlinear equations for x_0 **far from** ξ (4/4).

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
f_{10}	MCCTU(1)	1.4573×10^{-79}	1.0707×10^{-207}	6	3.9998
$x_0 = -2.7$	OS	5.9731×10^{-31}	3.7644×10^{-111}	10	4.0223
	KI	3.5019×10^{-84}	3.3094×10^{-207}	6	4.0006
	JA	2.7724×10^{-41}	1.3193×10^{-154}	5	3.9916
	OK1	4.672×10^{-37}	3.3675×10^{-141}	5	4.0067
	OK2	6.9502×10^{-38}	2.3984×10^{-144}	5	3.9545
	OK3	4.1808×10^{-48}	2.2923×10^{-179}	5	4.0014
	CH	1.8136×10^{-97}	3.8389×10^{-205}	6	4
	MA	2.2689×10^{-93}	3.8934×10^{-208}	5	4
	BMM	2.6823×10^{-41}	6.1829×10^{-161}	6	3.9999
	CLND1	8.2625×10^{-101}	2.7254×10^{-207}	5	4
	CLND2	1.8136×10^{-97}	3.8389×10^{-205}	6	4
	ACCT1	1.4573×10^{-79}	1.0707×10^{-207}	6	3.9998
	ACCT2	3.3917×10^{-58}	2.3908×10^{-204}	6	4.0003
	GH	2.2337×10^{-72}	1.0707×10^{-207}	5	3.9994
	KLW	2.4673×10^{-32}	6.302×10^{-122}	4	3.8791

The results presented in Tables 8–11 are promising. MCCTU(1) converges to the solution in nine out of the ten nonlinear equations, even when the initial estimate is far from the root ($x_0 \approx 3\xi$). In these cases, the ACOC consistently stabilizes and approaches 4. Only in one instance, for the function f_4, does MCCTU(1) fail to converge, similar to the other thirteen methods. For this particular equation, only two methods successfully approximate the root. In the remaining equations, MCCTU(1) converges to the solution with a comparable number of iterations to other methods and even requires fewer iterations than Ostrowski's method, as seen with function f_{10}. Therefore, we confirm that this method is robust, consistent with the stability results shown in previous sections.

Table 12. Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ ("nc" means non-convergence) (1/4).

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	Iter	ACOC
f_1	MCCTU(1)	3.9494×10^{-95}	0	6	4
$x_0 = -6.0$	OS	7.5454×10^{-40}	4.0753×10^{-158}	5	3.9989
	KI	2.2846×10^{-36}	4.3008×10^{-144}	5	3.9980
	JA	4.2789×10^{-41}	4.3067×10^{-163}	5	3.9992
	OK1	2.8437×10^{-53}	0	5	4
	OK2	2.0962×10^{-46}	1.9632×10^{-184}	5	3.9997
	OK3	1.9553×10^{-33}	1.5296×10^{-132}	5	4.0018
	CH	1.6161×10^{-33}	1.4248×10^{-132}	5	3.9966
	MA	2.632×10^{-26}	1.7609×10^{-103}	5	3.9875
	BMM	nc	nc	nc	nc
	CLND1	5.8255×10^{-95}	0	6	4
	CLND2	1.6161×10^{-33}	1.4248×10^{-132}	5	3.9966
	ACCT1	3.9494×10^{-95}	0	6	4
	ACCT2	5.5395×10^{-58}	0	5	3.9996
	GH	6.2374×10^{-89}	0	6	4
	KLW	1.0186×10^{-28}	3.0849×10^{-113}	5	3.9921
f_2	MCCTU(1)	1.0368×10^{-34}	5.4646×10^{-139}	4	4.0222
$x_0 = 2.0$	OS	2.3862×10^{-93}	0	5	4
	KI	1.2873×10^{-25}	4.3485×10^{-102}	4	3.9933
	JA	4.1797×10^{-95}	0	5	4
	OK1	8.2892×10^{-82}	0	5	4
	OK2	5.051×10^{-87}	0	5	4
	OK3	4.2138×10^{-33}	7.557×10^{-132}	4	3.9991
	CH	1.639×10^{-29}	1.4412×10^{-117}	4	3.9949
	MA	9.1807×10^{-43}	5.5512×10^{-171}	4	4.0028
	BMM	2.9675×10^{-52}	0	6	3.9998
	CLND1	9.0974×10^{-32}	7.3425×10^{-127}	4	4.0155
	CLND2	1.639×10^{-29}	1.4412×10^{-117}	4	3.9949
	ACCT1	1.0368×10^{-34}	5.4646×10^{-139}	4	4.0222
	ACCT2	1.9358×10^{-73}	0	5	4
	GH	6.4242×10^{-33}	2.803×10^{-132}	4	4.0791
	KLW	1.6753×10^{-39}	8.5848×10^{-158}	4	3.9976
f_3	MCCTU(1)	1.0037×10^{-74}	0	9	4
$x_0 = 6.0$	OS	1.4888×10^{-45}	1.7033×10^{-180}	7	4
	KI	3.4193×10^{-27}	1.1139×10^{-106}	7	3.9978
	JA	2.4788×10^{-42}	1.4488×10^{-167}	7	4
	OK1	7.5531×10^{-64}	0	7	4
	OK2	5.2399×10^{-91}	0	7	4
	OK3	2.024×10^{-56}	0	8	4.0130
	CH	3.2307×10^{-66}	6.8135×10^{-208}	8	4
	MA	2.7511×10^{-26}	2.21×10^{-102}	8	3.9951
	BMM	nc	nc	nc	nc
	CLND1	5.7924×10^{-73}	0	9	4
	CLND2	3.2307×10^{-66}	6.8135×10^{-208}	8	4
	ACCT1	1.0037×10^{-74}	0	9	4
	ACCT2	1.643×10^{-31}	1.4531×10^{-123}	6	4.0042
	GH	1.8548×10^{-60}	0	9	4
	KLW	1.1156×10^{-35}	4.1633×10^{-140}	8	3.9992

Table 13. Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ ("nc" means non-convergence) (2/4).

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	Iter	ACOC
f_4	MCCTU(1)	nc	nc	nc	nc
$x_0 = -141.0$	OS	nc	nc	nc	nc
	KI	nc	nc	nc	nc
	JA	nc	nc	nc	nc
	OK1	nc	nc	nc	nc
	OK2	nc	nc	nc	nc
	OK3	nc	nc	nc	nc
	CH	nc	nc	nc	nc
	MA	nc	nc	nc	nc
	BMM	nc	nc	nc	nc
	CLND1	nc	nc	nc	nc
	CLND2	nc	nc	nc	nc
	ACCT1	nc	nc	nc	nc
	ACCT2	nc	nc	nc	nc
	GH	nc	nc	nc	nc
	KLW	nc	nc	nc	nc
f_5	MCCTU(1)	1.2254×10^{-58}	0	7	4
$x_0 = 13.0$	OS	4.3174×10^{-43}	5.0572×10^{-170}	6	3.9996
	KI	5.4113×10^{-35}	1.9154×10^{-137}	6	3.9985
	JA	4.3174×10^{-43}	5.0572×10^{-170}	6	3.9996
	OK1	1.2884×10^{-87}	0	6	4
	OK2	1.1488×10^{-54}	0	6	4
	OK3	1.3547×10^{-27}	2.9602×10^{-108}	6	4.0338
	CH	1.1569×10^{-28}	6.0929×10^{-112}	6	3.9948
	MA	2.1036×10^{-69}	6.2295×10^{-207}	7	4
	BMM	nc	nc	nc	nc
	CLND1	1.2254×10^{-58}	0	7	4
	CLND2	1.1569×10^{-28}	6.0929×10^{-112}	6	3.9948
	ACCT1	1.2254×10^{-58}	0	7	4
	ACCT2	7.7193×10^{-68}	0	6	4
	GH	6.6848×10^{-52}	2.2302×10^{-204}	7	3.9999
	KLW	1.4904×10^{-82}	6.2295×10^{-207}	7	4
f_6	MCCTU(1)	1.4106×10^{-84}	1.2167×10^{-208}	5	4
$x_0 = 1.0$	OS	1.0011×10^{-40}	6.3155×10^{-162}	4	3.9991
	KI	6.4033×10^{-37}	1.4516×10^{-146}	4	3.9984
	JA	2.3288×10^{-40}	1.9843×10^{-160}	4	3.9991
	OK1	1.8287×10^{-53}	1.2167×10^{-208}	4	3.9997
	OK2	1.2544×10^{-44}	1.1864×10^{-177}	4	3.9995
	OK3	7.2892×10^{-32}	1.4139×10^{-126}	4	3.9897
	CH	9.2308×10^{-32}	9.1584×10^{-126}	4	3.9962
	MA	1.5346×10^{-95}	1.2167×10^{-208}	5	4
	BMM	2.958×10^{-31}	1.3155×10^{-123}	4	4.0024
	CLND1	1.5451×10^{-84}	1.2167×10^{-208}	5	4
	CLND2	9.2308×10^{-32}	9.1584×10^{-126}	4	3.9962
	ACCT1	1.4106×10^{-84}	1.2167×10^{-208}	5	4
	ACCT2	5.8091×10^{-73}	1.2167×10^{-208}	5	4
	GH	4.0743×10^{-77}	1.2167×10^{-208}	5	4
	KLW	3.7745×10^{-28}	3.658×10^{-111}	4	3.9931

Table 14. Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ ("nc" means non-convergence) (3/4).

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_7	MCCTU(1)	nc	nc	nc	nc
$x_0 = -12.0$	OS	nc	nc	nc	nc
	KI	nc	nc	nc	nc

Table 14. *Cont.*

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC
	JA	nc	nc	nc	nc
	OK1	nc	nc	nc	nc
	OK2	nc	nc	nc	nc
	OK3	nc	nc	nc	nc
	CH	nc	nc	nc	nc
	MA	nc	nc	nc	nc
	BMM	nc	nc	nc	nc
	CLND1	nc	nc	nc	nc
	CLND2	nc	nc	nc	nc
	ACCT1	nc	nc	nc	nc
	ACCT2	1.2624×10^{-39}	3.2997×10^{-154}	50	4.0007
	GH	nc	nc	nc	nc
	KLW	nc	nc	nc	nc
f_8	MCCTU(1)	7.1071×10^{-32}	2.9726×10^{-127}	6	3.9944
$x_0 = 23.0$	OS	3.9961×10^{-44}	5.4218×10^{-176}	6	3.9991
	KI	1.9961×10^{-46}	3.223×10^{-185}	6	3.9995
	JA	2.8208×10^{-93}	1.5574×10^{-207}	6	4
	OK1	1.2245×10^{-30}	3.8716×10^{-122}	5	3.9812
	OK2	1.3604×10^{-44}	5.3174×10^{-178}	5	3.9985
	OK3	1.477×10^{-56}	1.5574×10^{-207}	6	3.9998
	CH	3.6894×10^{-61}	1.5574×10^{-207}	6	3.9999
	MA	5.2575×10^{-34}	1.0738×10^{-135}	6	4.0001
	BMM	3.7259×10^{-45}	3.1549×10^{-179}	7	4.0007
	CLND1	1.0076×10^{-36}	5.271×10^{-147}	6	3.9964
	CLND2	3.6894×10^{-61}	1.5574×10^{-207}	6	3.9999
	ACCT1	7.1071×10^{-32}	2.9726×10^{-127}	6	3.9944
	ACCT2	1.1468×10^{-52}	1.5574×10^{-207}	6	3.9997
	GH	3.8246×10^{-31}	1.9787×10^{-124}	6	3.9800
	KLW	1.5819×10^{-37}	1.0306×10^{-149}	6	4.0012
f_9	MCCTU(1)	7.6712×10^{-52}	8.5422×10^{-204}	9	4
$x_0 =$ -14.0	OS	2.3748×10^{-99}	0	8	4
	KI	7.9344×10^{-69}	0	8	4.0006
	JA	1.0139×10^{-94}	0	8	4
	OK1	3.2135×10^{-50}	1.0728×10^{-197}	7	3.9999
	OK2	1.5049×10^{-32}	2.7173×10^{-127}	7	3.9952
	OK3	7.1691×10^{-34}	1.3332×10^{-132}	8	3.9822
	CH	8.1205×10^{-44}	1.62×10^{-172}	8	4.0015
	MA	5.7604×10^{-70}	0	9	4
	BMM	nc	nc	nc	nc
	CLND1	9.8978×10^{-52}	2.306×10^{-203}	9	4
	CLND2	8.1205×10^{-44}	1.62×10^{-172}	8	4.0015
	ACCT1	7.6712×10^{-52}	8.5422×10^{-204}	9	4
	ACCT2	7.9723×10^{-43}	6.4405×10^{-168}	7	4.0003
	GH	3.8751×10^{-42}	7.0871×10^{-165}	9	4.0001
	KLW	1.1099×10^{-94}	0	9	4

The results presented in Tables 12–15 confirm the exceptional robustness of the MCCTU(1) method for initial estimates that are very far from the root ($x_0 \approx 10\xi$), as the method converges in eight out of ten cases. A slight dependence on the initial estimate is observed for functions f_4 and f_7, where the method does not converge; however, in these two cases, the other methods also fail to approximate the solution, except for the ACCT2 method, which converges to the root of function f_7 with 50 iterations. The complexity of the nonlinear equations plays a significant role in finding their solutions. Moreover, in the cases where the MCCTU(1) method converges to the roots, it does so with a comparable number of iterations to other methods and often with fewer iterations, as seen in function f_2. Additionally, for these cases, the ACOC consistently stabilizes at values close to 4.

Table 15. Numerical performance of iterative methods in nonlinear equations for x_0 **very far from** ξ ("nc" means non-convergence) (4/4).

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	Iter	ACOC
f_{10}	MCCTU(1)	1.2776×10^{-70}	1.0707×10^{-207}	6	4.0008
$x_0 = -9.0$	OS	2.9225×10^{-29}	1.7446×10^{-112}	5	3.9821
	KI	7.9476×10^{-94}	1.5574×10^{-207}	8	4
	JA	6.1519×10^{-29}	1.4803×10^{-108}	8	4.0703
	OK1	2.3282×10^{-63}	9.7336×10^{-208}	6	4
	OK2	2.4314×10^{-75}	1.9467×10^{-208}	7	4
	OK3	1.9369×10^{-93}	4.9203×10^{-206}	6	4
	CH	1.0491×10^{-37}	1.9596×10^{-135}	8	4.0092
	MA	9.5564×10^{-89}	3.8934×10^{-208}	6	4
	BMM	nc	nc	nc	nc
	CLND1	1.5067×10^{-28}	4.9584×10^{-107}	12	3.7072
	CLND2	1.0491×10^{-37}	1.9596×10^{-135}	8	4.0092
	ACCT1	1.2776×10^{-70}	1.0707×10^{-207}	6	4.0008
	ACCT2	1.463×10^{-49}	5.2801×10^{-191}	5	4.0594
	GH	2.4513×10^{-45}	6.3226×10^{-174}	7	4.0044
	KLW	5.7956×10^{-39}	1.4646×10^{-150}	6	4.0153

Therefore, based on the results of the second experiment, we conclude that the MCCTU(α) family demonstrates impressive numerical performance when using the optimal stable member with $\alpha = 1$ as a representative, highlighting its robustness and efficiency even with challenging initial conditions. Overall, the selected MCCTU(1) method exhibits low errors and requires a similar or fewer number of iterations compared to other methods. In certain cases, as the complexity of the nonlinear equation increases, the MCCTU(1) method outperforms Ostrowski's method and others. The theoretical convergence order is also confirmed by the ACOC, which is always close to 4.

5. Conclusions

The development of the parametric family of multistep iterative schemes MCCTU(α) based on the damped Newton scheme has proven to be an effective strategy for solving nonlinear equations. The inclusion of an additional Newton step with a weight function and a "frozen" derivative significantly improved the convergence speed from a first-order class to a uniparametric third-order family.

The numerical results confirm the robustness of the MCCTU(2) method for initial estimates close to the root ($x_0 \approx \xi$), with very low errors and convergence in three or four iterations. As the initial estimates move further away ($x_0 \approx 3\xi$) and ($x_0 \approx 10\xi$), the method continues to show solid performance, converging in most cases and confirming its theoretical stability and robustness.

Through the analysis of stability surfaces and dynamical planes, specific members of the MCCTU(α) family with exceptional stability were identified. These members are particularly suitable for scalar functions with challenging convergence behavior, exhibiting attractive periodic orbits and strange fixed points in their corresponding dynamical planes. The MCCTU(1) member stood out for its optimal and stable performance.

In the comparative analysis, the MCCTU(1) method demonstrated superior numerical performance in many cases, requiring a similar or fewer number of iterations compared to well-established fourth-order methods such as Ostrowski's method. This superior performance is especially notable in more complex nonlinear equations, where MCCTU(1) outperforms several alternative methods.

The theoretical convergence order of the MCCTU(α) family was confirmed by calculating the approximate computational order of convergence (ACOC). In most cases, the ACOC value stabilized close to 3, validating the effectiveness and accuracy of the proposed methods both theoretically and practically. Additionally, it was confirmed that the convergence order of the method associated with $\alpha = 1$ is optimal, achieving a fourth-order convergence.

Finally, the analysis revealed that certain members of the MCCTU(α) family, particularly those with α values outside the stability surface, exhibited significant instability. These methods struggled to converge to the solution, especially when initial estimates were far or very far from the root. For instance, the method with $\alpha = 100$ failed to stabilize and did not meet the convergence criteria in four out of ten cases. Additionally, the ACOC values for this method did not stabilize, confirming its theoretical instability. This highlights the importance of selecting appropriate parameter values within the stability regions to ensure reliable performance.

Author Contributions: Conceptualization, A.C. and J.R.T.; methodology, G.U.-C. and M.M.-M.; software, M.M.-M. and G.U.-C.; validation, M.M.-M.; formal analysis, J.R.T.; investigation, A.C.; writing—original draft preparation, M.M.-M.; writing—review and editing, A.C. and F.I.C.; supervision, J.R.T. All authors have read and agreed to the published version of the manuscript.

Funding: Funded with Ayuda a Primeros Proyectos de Investigación (PAID-06-23), Vicerrectorado de Investigación de la Universitat Politècnica de València (UPV).

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Detailed Computation of Theorem 2

The comprehensive proof of Theorem 2, methodically detailed step-by-step in Section 2, is further validated in Wolfram Mathematica software v13.2 using the following code:

```
fx = dFa SeriesData[Subscript[e, k], 0, {0, 1, Subscript[C, 2], Subscript[C, 3],
Subscript[C, 4], Subscript[C, 5]}, 0, 5, 1];
dfx = D[fx, Subscript[e, k]];
fx/dfx // Simplify;
(*Error in the first step*)
Subscript[y, e] = Simplify[Subscript[e, k] - \[Alpha]*fx/dfx];
fy = fx /. Subscript[e, k] -> Subscript[y, e] // Simplify;
(*Error in the second step*)
Subscript[x, e] = Subscript[y, e] - (\[Beta] + \[Gamma]*fy/fx +
\[Delta]*(fy/fx)^2)*(fx/dfx) // Simplify
```

Appendix B. Detailed Computation of Theorem 3

The comprehensive proof of Theorem 3, methodically detailed step-by-step in Section 2, is further validated in Wolfram Mathematica software v13.2 using the following code:

```
fx = dFa SeriesData[Subscript[e, k], 0, {0, 1, Subscript[C, 2], Subscript[C, 3],
Subscript[C, 4], Subscript[C, 5]}, 0, 5, 1];
dfx = D[fx, Subscript[e, k]];
fx/dfx // Simplify;
(*Error in the first step*)
Subscript[y, e] = Simplify[Subscript[e, k] - \[Alpha]*fx/dfx];
fy = fx /. Subscript[e, k] -> Subscript[y, e] // Simplify;
(*Error in the second step*)
Subscript[x, e] = Subscript[y, e] - (\[Beta] + \[Gamma]*fy/fx +
\[Delta]*(fy/fx)^2)*(fx/dfx) // Simplify;
Solve[1 - \[Beta] - \[Gamma] - \[Delta] - \[Alpha]^2 \[Delta] + \[Alpha]
(-1 + \[Gamma] + 2 \[Delta]) == 0 && \[Beta] + \[Gamma] + \[Delta] + 2 \[Alpha]^3
\[Delta] - \[Alpha]^2 (\[Gamma] + \[Delta]) - \[Alpha] (-1 + \[Gamma] + 2 \[Delta])
== 0, {\[Alpha], \[Beta], \[Gamma], \[Delta]}];
Subscript[x, e] = FullSimplify[Subscript[x, e] /. {\[Beta] -> ((-1 + \[Alpha])^2
(-1 - \[Alpha] + \[Alpha]^2 \[Delta]))/\[Alpha]^2, \[Gamma] -> (1 - 2 \[Alpha]^2
\[Delta] + 2 \[Alpha]^3 \[Delta])/\[Alpha]^2}]
```

Appendix C. Additional Experiment Focused on Practical Calculations

In this comprehensive experiment, we conduct an in-depth efficiency analysis of the MCCTU(1) method, set with $\epsilon = \alpha^4\delta = 2$, specifically tailored for practical calculations. This analysis begins with initial estimates that closely approximate the roots ($x_0 \approx \xi$). All computations are carried out using the MATLAB R2020b software package with standard floating-point arithmetic. We assess the number of iterations (iter) each method requires to reach the solution, with sttopping criteria of $|x_{k+1} - x_k| < 10^{-10}$. We also calculate the Approximate Computational Order of Convergence (ACOC) to verify the theoretical order of convergence (p). Our findings indicate that fluctuating ACOC values are marked with a '-', and methods that do not converge within 50 iterations are labeled as 'nc'. Additionally, this study aims to examine how the convergence order is influenced by the number of digits in the variable precision arithmetic employed in the experiments, using the same ten nonlinear test equations listed in Table 1. Thus, the numerical results are presented in Table A1.

Table A1. Numerical results of MCCTU(1) in practical calculations for x_0 **close to** ξ.

Function	x_0	$\|x_{k+1} - x_k\|$	Iter	ACOC	ξ
f_1	-0.6	8.4069×10^{-27}	3	4.0111	-0.6367
f_2	0.2	4.0915×10^{-36}	3	3.9624	0.2575
f_3	0.6	1.8066×10^{-21}	3	4.0121	0.6392
f_4	-14.1	3.6467×10^{-15}	2	-	-14.1013
f_5	1.3	1.2827×10^{-20}	3	4.0226	1.3652
f_6	0.1	1.1439×10^{-32}	3	3.9969	0.1281
f_7	-1.2	8.1090×10^{-29}	3	4.0025	-1.2076
f_8	2.3	3.2363×10^{-36}	3	4.0010	2.3320
f_9	-1.4	1.2504×10^{-28}	3	3.9982	-1.4142
f_{10}	-0.9	1.3096×10^{-27}	3	4.0263	-0.9060

From the analysis of this experiment, it is confirmed that convergence to the solution is achieved in all cases, with errors smaller than the set threshold, reaching convergence within 2 or 3 iterations. The value of the ACOC stabilizes at 4, thus verifying the theoretical results. Furthermore, it is clear that the convergence order is not affected by the number of digits in the variable precision arithmetic used. The number of digits plays a crucial role when higher precision is required, particularly for smaller errors, preventing divisions by zero in this case. Additionally, it is noted that the ACOC for function f_4 cannot be calculated, due to convergence to the solution in just 2 iterations, while (2) requires at least 3 iterations to calculate the approximate order of convergence.

References

1. Danchick, R. Gauss meets Newton again: How to make Gauss orbit determination from two position vectors more efficient and robust with Newton–Raphson iterations. *Appl. Math. Comput.* **2008**, *195*, 364–375.
2. Tostado-Véliz, M.; Kamel, S.; Jurado, F.; Ruiz-Rodriguez, F.J. On the Applicability of Two Families of Cubic Techniques for Power Flow Analysis. *Energies* **2021**, *14*, 4108. https://doi.org/10.3390/en14144108.
3. Arroyo, V.; Cordero, A.; Torregrosa, J.R. Approximation of artificial satellites' preliminary orbits: The efficiency challenge. *Math. Comput. Model.* **2011**, *54*, 1802–1807. https://doi.org/10.1016/j.mcm.2010.11.063.
4. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1964.
5. Petković, M.; Neta, B.; Petković, L.; Džunić, J. *Multipoint Methods for Solving Nonlinear Equations*; Academic Press: Boston, MA, USA, 2013.
6. Amat, S.; Busquier, S. *Advances in Iterative Methods for Nonlinear Equations*; Springer: Cham, Switzerland, 2017.
7. Ostrowski, A.M. *Solution of Equations in Euclidean and Banach Spaces*; Academic Press: New York, NY, USA, 1973.
8. Özban, A.Y.; Kaya, B. A new family of optimal fourth-order iterative methods for nonlinear equations. *Results Control Optim.* **2022**, *8*, 1–11. https://doi.org/10.1016/j.rico.2022.100157.
9. Adomian, G. *Solving Frontier Problem of Physics: The Decomposition Method*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1994.
10. Petković, M.; Neta, B.; Petković, L.; Džunić, J. Multipoint methods for solving nonlinear equations: A survey. *Appl. Math. Comput.* **2014**, *226*, 635–660. https://doi.org/10.1016/j.amc.2013.10.072.

11. Chun, C. Some fourth-order iterative methods for solving nonlinear equations. *Appl. Math. Comput.* **2008**, *195*, 454–459.
12. Ostrowski, A.M. *Solution of Equations and Systems of Equations*; Academic Press: New York, NY, USA, 1960.
13. Kung, H.T.; Traub, J.F. Optimal Order of One-Point and Multipoint Iteration. *J. Assoc. Comput. Mach.* **1974**, *21*, 643–651.
14. Weerakoon, S.; Fernando, T. A variant of Newton's method with accelerated third-order convergence. *Appl. Math. Lett.* **2000**, *13*, 87–93. https://doi.org/10.1016/S0893-9659(00)00100-2.
15. Cordero, A.; Torregrosa, J.R. Variants of Newton's Method using fifth-order quadrature formulas. *Appl. Math. Comput.* **2007**, *190*, 686–698. https://doi.org/10.1016/j.amc.2007.01.062.
16. Kansal, M.; Cordero, A.; Bhalla, S.; Torregrosa, J.R. New fourth- and sixth-order classes of iterative methods for solving systems of nonlinear equations and their stability analysis. *Numer. Algorithms* **2021**, *87*, 1017–1060.
17. Cordero, A.; Soleymani, F.; Torregrosa, J.R. Dynamical analysis of iterative methods for nonlinear systems or how to deal with the dimension? *Appl. Math. Comput.* **2014**, *244*, 398–412.
18. Cordero, A.; Moscoso-Martínez, M.; Torregrosa, J.R. Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations. *Algorithms* **2021**, *14*, 101.
19. Husain, A.; Nanda, M.N.; Chowdary, M.S.; Sajid, M. Fractals: An Eclectic Survey, Part I. *Fractal Fract.* **2022**, *6*, 89. https://doi.org/10.3390/fractalfract6020089.
20. Husain, A.; Nanda, M.N.; Chowdary, M.S.; Sajid, M. Fractals: An Eclectic Survey, Part II. *Fractal Fract.* **2022**, *6*, 379. https://doi.org/10.3390/fractalfract6070379.
21. Varona, J.L. Graphic and numerical comparison between iterative methods. *Math. Intell.* **2002**, *24*, 37–46. https://doi.org/10.1007/BF03025310.
22. Amat, S.; Busquier, S.; Plaza, S. Review of some iterative root–finding methods from a dynamical point of view. *SCI. A Math. Sci.* **2004**, *10*, 3–35.
23. Neta, B.; Chun, C.; Scott, M. Basins of attraction for optimal eighth order methods to find simple roots of nonlinear equations. *Appl. Math. Comput.* **2014**, *227*, 567–592. https://doi.org/10.1016/j.amc.2013.11.017.
24. Cordero, A.; García-Maimó, J.; Torregrosa, J.R.; Vassileva, M.P.; Vindel, P. Chaos in King's iterative family. *Appl. Math. Lett.* **2013**, *26*, 842–848. https://doi.org/10.1016/j.aml.2013.03.012.
25. Magreñán, A.; Argyros, I. *A Contemporary Study of Iterative Methods*; Academic Press: Cambridge, MA, USA, 2018.
26. Geum, Y.H.; Kim, Y.I. Long–term orbit dynamics viewed through the yellow main component in the parameter space of a family of optimal fourth-order multiple-root finders. *Discrete Contin. Dyn. Syst. B* **2020**, *25*, 3087–3109. https://doi.org/10.3934/dcdsb.2020052.
27. Cordero, A.; Torregrosa, J.; Vindel, P. Dynamics of a family of Chebyshev-Halley type method. *Appl. Math. Comput.* **2012**, *219*, 8568–8583. https://doi.org/10.1016/j.amc.2013.02.042.
28. Magreñán, A. Different anomalies in a Jarratt family of iterative root-finding methods. *Appl. Math. Comput.* **2014**, *233*, 29–38.
29. Devaney, R. *An Introduction to Chaotic Dynamical Systems*; Addison-Wesley Publishing Company: Boston, MA, USA, 1989.
30. Beardon, A. *Iteration of Rational Functions*; Graduate Texts in Mathematics; Springer: New York, NY, USA, 1991.
31. Fatou, P. Sur les équations fonctionelles. *Bull. Soc. Mat. Fr.* **1919**, *47*, 161–271.
32. Julia, G. Mémoire sur l'iteration des fonctions rationnelles. *Mat. Pur. Appl.* **1918**, *8*, 47–245.
33. Scott, M.; Neta, B.; Chun, C. Basin attractors for various methods. *Appl. Math. Comput.* **2011**, *218*, 2584–2599. https://doi.org/10.1016/j.amc.2011.07.076.
34. Blanchard, P. Complex analytic dynamics on the Riemann sphere. *Bull. Am. Math. Soc.* **1984**, *11*, 85–141.
35. José L. Hueso, E.M.; Teruel, C. Multipoint efficient iterative methods and the dynamics of Ostrowski's method. *Int. J. Comput. Math.* **2019**, *96*, 1687–1701. https://doi.org/10.1080/00207160.2015.1080354.
36. King, R.F. A Family of Fourth Order Methods for Nonlinear Equations. *SIAM J. Numer. Anal.* **1973**, *10*, 876–879. https://doi.org/10.1137/0710072.
37. Jarratt, P. Some fourth order multipoint iterative methods for solving equations. *Math. Comput.* **1966**, *20*, 434–437. https://doi.org/10.1090/S0025-5718-66-99924-8.
38. Chun, C. Construction of Newton-like iteration methods for solving nonlinear equations. *Numer. Math.* **2006**, *104*, 297–315. https://doi.org/10.1007/s00211-006-0025-2.
39. Maheshwari, A.K. A fourth order iterative method for solving nonlinear equations. *Appl. Math. Comput.* **2009**, *211*, 383–391. https://doi.org/10.1016/j.amc.2009.01.047.
40. Behl, R.; Maroju, P.; Motsa, S. A family of second derivative free fourth order continuation method for solving nonlinear equations. *J. Comput. Appl. Math.* **2017**, *318*, 38–46. https://doi.org/10.1016/j.cam.2016.12.008.
41. Chun, C.; Lee, M.Y.; Neta, B.; Džunić, J. On optimal fourth-order iterative methods free from second derivative and their dynamics. *Appl. Math. Comput.* **2012**, *218*, 6427–6438. https://doi.org/10.1016/j.amc.2011.12.013.
42. Artidiello, S.; Chicharro, F.; Cordero, A.; Torregrosa, J.R. Local convergence and dynamical analysis of a new family of optimal fourth-order iterative methods. *Int. J. Comput. Math.* **2013**, *90*, 2049–2060. https://doi.org/10.1080/00207160.2012.748900.

43. Ghanbari, B. A new general fourth-order family of methods for finding simple roots of nonlinear equations. *J. King Saud Univ. Sci.* **2011**, *23*, 395–398. https://doi.org/10.1016/j.jksus.2010.07.018.
44. Kou, J.; Li, Y.; Wang, X. A composite fourth-order iterative method for solving non-linear equations. *Appl. Math. Comput.* **2007**, *184*, 471–475. https://doi.org/10.1016/j.amc.2006.05.181.