UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTMENT OF MATHEMATICS

Efficient numerical methods for solving nonlinear problems

September 2024

Author: Marlon Ernesto Moscoso Martínez

Supervisors: Juan Ramón Torregrosa Sánchez
Alicia Cordero Barbero
Francisco Israel Chicharro López

Juan Ramón Torregrosa Sánchez, Full Professor of the Department of Mathematics at the Universitat Politècnica de València, Alicia Cordero Barbero, Full Professor of the Department of Mathematics at the Universitat Politècnica de València, and Francisco Israel Chicharro López, Associate Professor of the Department of Mathematics at the Universitat Politècnica de València,

CERTIFY:

That Mr. Marlon Ernesto Moscoso Martínez, Master in Mathematical Research, has carried out, under our supervision, the work included in this dissertation to qualify for the degree of PhD in Mathematics from the Universitat Politècnica de València.

Furthermore, we authorize the presentation of this work before the Universitat Politècnica de València to comply with the corresponding procedures.

For legal purposes, we sign this document in València, on September 12, 2024.

Sgd. Juan R. Torregrosa   Sgd. Alicia Cordero   Sgd. Francisco I. Chicharro

Acknowledgements

To Marco, Anita, Nycolai, Santiago, and Mabela,

for their permanent and unconditional support.

To Alicia, Juan Ramón, and Francisco,

for their guidance, patience, and wisdom that have been fundamental in this academic journey.

València, September 12, 2024.

"Mathematics is the language in which God has written the universe."
Galileo Galilei

Resumen

La resolución de ecuaciones y sistemas de ecuaciones no lineales es fundamental en muchas disciplinas científicas y de ingeniería, incluyendo la física, la química, la biología, la economía y la informática. Los métodos numéricos son cruciales para resolver estas ecuaciones debido a su complejidad, que a menudo resulta en múltiples soluciones o en la ausencia de ellas, lo que hace que los métodos analíticos tradicionales sean inadecuados. Esta investigación se centra en el desarrollo y análisis de nuevos esquemas iterativos para resolver ecuaciones y sistemas de ecuaciones no lineales, enfatizando la convergencia, la estabilidad y la eficiencia computacional. Como parte de esta investigación se publicaron tres artículos clave. El primer artículo introduce una novedosa familia de métodos iterativos de dos pasos derivada de un esquema de Newton amortiguado, que incluye un paso adicional de Newton con una función de peso y una derivada "congelada". Esta familia, inicialmente una clase de cuatro parámetros con convergencia de primer orden, se convierte en una familia de un solo parámetro con convergencia de tercer orden, que además muestra una estabilidad y eficiencia excepcionales, validadas mediante pruebas numéricas. El segundo artículo presenta un nuevo método iterativo de tres pasos, inicialmente una familia de tres parámetros de cuarto orden que acelera a una familia de un solo parámetro de sexto orden. La convergencia, la dinámica compleja y el comportamiento numérico de este método son estudiados a fondo, identificando miembros estables adecuados para problemas prácticos. El tercer artículo extiende la familia de sexto orden a sistemas de ecuaciones no lineales, creando un esquema de un solo parámetro altamente eficiente. Los análisis dinámicos y numéricos confirman la convergencia, estabilidad y aplicabilidad de esta familia extendida para problemas de gran escala. La investigación tiene como objetivo superar las limitaciones de algunos métodos existentes, ofreciendo soluciones robustas y eficientes para ecuaciones y sistemas no lineales. El documento está estructurado para cubrir el desarrollo, análisis y validación de estos métodos, proporcionando recomendaciones específicas para su aplicación práctica en varios dominios científicos y de ingeniería.

Resum

La resolució d'equacions i sistemes d'equacions no lineals és fonamental en moltes disciplines científiques i d'enginyeria, incloent la física, la química, la biologia, l'economia i la informàtica. Els mètodes numèrics són crucials per a resoldre aquestes equacions a causa de la seua complexitat, que sovint resulta en múltiples solucions o en l'absència d'elles, la qual cosa fa que els mètodes analítics tradicionals siguen inadequats. Aquesta investigació se centra en el desenvolupament i anàlisi de nous esquemes iteratius per a resoldre equacions i sistemes d'equacions no lineals, emfatitzant la convergència, l'estabilitat i l'eficiència computacional. Com a part d'aquesta investigació es van publicar tres articles clau. El primer article introdueix una nova família de mètodes iteratius de dos passos derivada d'un esquema de Newton esmorteït, que inclou un pas addicional de Newton amb una funció de pes i una derivada "congelada". Aquesta família, inicialment una classe de quatre paràmetres amb convergència de primer ordre, es converteix en una família d'un sol paràmetre amb convergència de tercer ordre, que a més mostra una estabilitat i eficiència excepcionals, validats mitjançant proves numèriques. El segon article presenta un nou mètode iteratiu de tres passos, inicialment una família de tres paràmetres de quart ordre que accelera a una família d'un sol paràmetre de sisè ordre. La convergència, la dinàmica complexa i el comportament numèric d'aquest mètode són estudiats a fons, identificant membres estables adequats per a problemes pràctics. El tercer article amplia la família de sisè ordre a sistemes d'equacions no lineals, creant un esquema d'un sol paràmetre altament eficient. Els anàlisis dinàmics i numèrics confirmen la convergència, estabilitat i aplicabilitat d'aquesta família ampliada per a problemes de gran escala. La investigació té com a objectiu superar les limitacions d'alguns mètodes existents, oferint solucions robustes i eficients per a equacions i sistemes no lineals. El document està estructurat per a cobrir el desenvolupament, anàlisi i validació d'aquests mètodes, proporcionant recomanacions específiques per a la seua aplicació pràctica en diversos dominis científics i d'enginyeria.

Abstract

The resolution of non-linear equations and systems is fundamental in various scientific and engineering fields, including physics, chemistry, biology, economics, and computer science. Numerical methods are crucial for solving these equations due to their complexity, which often results in multiple or no solutions, rendering traditional analytical methods inadequate. This research focuses on developing and analyzing new iterative schemes for solving non-linear equations and systems, emphasizing convergence, stability, and computational efficiency. Three key papers were published as part of this research. The first paper introduces a novel family of two-step iterative methods derived from a damped Newton scheme, which includes an additional Newton step with a weight function and a "frozen" derivative. This family, initially a four-parameter class with first-order convergence, becomes a single-parameter family with third-order convergence, which also exhibits exceptional stability and efficiency, validated through numerical tests. The second paper presents a new three-step iterative method, initially a three-parameter fourth-order family, which accelerates to a single-parameter sixth-order family. This method's convergence, complex dynamics, and numerical behavior are thoroughly studied, identifying stable members suitable for practical problems. The third paper extends the sixth-order family to systems of non-linear equations, creating a highly efficient single-parameter family. Dynamic and numerical analyses confirm the convergence, stability, and applicability of this extended family for large-scale problems. The research aims to overcome the limitations of some existing methods, offering robust and efficient solutions for non-linear equations and systems. The document is structured to cover the development, analysis, and validation of these methods, providing specific recommendations for their practical application in various scientific and engineering domains.

Contents

Chapter 1

Introduction

The resolution of non-linear equations and systems of equations represents a central problem in various scientific and engineering disciplines. These equations arise in a wide range of contexts, including physics, chemistry, biology, economics, and computer science. For example, in physics, non-linear equations are crucial for modeling phenomena such as fluid dynamics, field theory, and quantum mechanics [1]. In engineering, these systems are essential for the analysis and design of control systems, electronic circuits, and mechanical structures [2].

Numerical methods are essential for solving non-linear equations due to the intrinsic complexity of these problems. Unlike linear equations, non-linear equations can exhibit multiple solutions, complex solutions, or even have no solution at all. This unpredictable nature renders traditional analytical techniques inadequate, necessitating the development of robust and efficient numerical methods [3].

The study of numerical methods for solving non-linear equations has evolved significantly, with advancements enabling the tackling of increasingly complex and large-scale problems. Iterative methods, in particular, have proven to be powerful tools due to their ability to handle large systems of equations and their applicability to problems where analytical methods are not viable [4].

The importance of numerical methods also extends to biology and medicine, where they are used to model cell growth, disease spread, and the interaction of complex biological systems [5]. In economics, these methods are fundamental for modeling financial markets and analyzing macroeconomic dynamics [6]. In quantum mechanics, the non-linear Schrödinger equation is essential for describing quantum systems in the presence of non-linear potentials. Numerical methods enable the solution of these equations to obtain wave functions and system energies, which are crucial for understanding material properties and interactions at the quantum level [7]. In control engineering, non-linear systems are ubiquitous. The modeling and control of dynamic systems, such as robots, autonomous vehicles, and industrial processes, often require solving non-linear systems of equations to design controllers that ensure system stability and performance [8].

Despite the diversity and sophistication of existing methods, several challenges and limitations related to the convergence, stability, and computational efficiency of these algorithms persist [9]. Additionally, many iterative methods, such as Newton-Raphson, require a good initial approximation to ensure convergence to a solution. Without an adequate initial estimate, these methods may diverge or converge to undesired solutions [10]. Furthermore, some methods are extremely sensitive to initial conditions, which can lead to slow convergence or a lack of convergence. This sensitivity is a significant problem in practical applications where precise initial conditions are not always available [4].

The computation of derivatives and, in the case of large systems, of the Jacobian matrix can be costly and, in some cases, impractical [11]. Computational cost is another significant challenge. Methods such as Quasi-Newton and Homotopy can be computationally intensive, especially for large or high-dimensional systems. Computational efficiency is crucial for the practical applicability of these methods in real-world problems [12]. Additionally, numerical stability is a constant concern in solving non-linear equations. Rounding and truncation errors can be amplified through iterations, leading to incorrect or non-convergent solutions [13].

Non-linear equations often have multiple solutions, and many methods are not designed to identify all possible solutions. This can be problematic in applications where all solutions are of interest [9]. This issue is particularly relevant in biology and medicine, where different solutions may represent different physiological or pathological states [14].

The need for more robust and efficient numerical methods that are applicable to a wider variety of problems drives ongoing development in this field. New methods must overcome the limitations of existing approaches and offer solutions that are both theoretically sound and practically viable. Recent advances in research involve the combination of different approaches, such as fixed-point methods, including Newton's method, which have been shown to improve convergence and robustness across a variety of problems [15]. Additionally, adaptive algorithms that dynamically adjust their parameters during iteration are gaining popularity. These algorithms can automatically modify the iteration step and other parameters to enhance convergence and stability [4].

The integration of machine learning techniques and numerical methods is emerging as a promising trend. These techniques can be used to predict good initial approximations or adjust method parameters in real-time [16]. In this regard, hybrid approaches that combine machine learning with traditional numerical methods are showing great potential to improve the efficiency and accuracy of solving non-linear equations [17].

With the progress in high-performance computing, parallel and distributed methods are being developed to handle large-scale systems of equations more effectively [6]. These recent advancements promise to significantly enhance the applicability and efficiency of numerical methods for solving non-linear equations and systems. Parallel and distributed computing allows for the utilization of multiple processors to divide and solve the problem, thereby reducing computation time and enabling the tackling of larger and more complex problems [18].

In this context, the present research focuses on the development and analysis of new families of iterative schemes for solving non-linear equations and systems of equations, with an emphasis on their convergence, stability, and computational efficiency. The aim is to contribute to the field of numerical analysis by proposing new iterative methods that optimize these aspects. As part of this research, the following three papers have been published in JCR impact factor journals:

- 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. IF 1.9 / JCR - Q1 (Mathematics, Applied).

- Cordero, A.; Moscoso-Martínez, M.; Torregrosa, J.R. *Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations*. Algorithms 2021, 14, 101. IF 1.8 / JCR - Q2 (Computer Science, Theory and Methods) / CiteScore - Q1 (Numerical Analysis).

- Moscoso-Martínez, M.; Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. *Performance of a New Sixth-Order Class of Iterative Schemes for Solving Non-Linear Systems of Equations*. Mathematics 2023, 11, 1374. IF 2.3 / JCR - Q1 (Mathematics) / CiteScore - Q1 (General Mathematics).

The article *"Achieving Optimal Order in a Novel Family of Numerical Methods: Insights from Convergence and Dynamical Analysis Results"* presents a new parametric family of two-step iterative methods for solving non-linear equations. This family is derived from a damped Newton scheme but includes an additional Newton step with a weight function and a "frozen" derivative, i.e., the same derivative as the previous step. Initially, a four-parameter class with first-order convergence is developed, which, by fixing one of its parameters, becomes a single-parameter family of third-order:

$$\begin{cases} y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta \left(\dfrac{f(y_k)}{f(x_k)} \right)^2 \right) \left(\dfrac{f(x_k)}{f'(x_k)} \right), \end{cases} \tag{1.1}$$

where α is an arbitrary parameter, $\beta = \dfrac{(\alpha-1)^2\left(\alpha^2\delta - \alpha - 1\right)}{\alpha^2}$, $\gamma = \dfrac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}$, $\delta = \dfrac{2}{\alpha^4}$, and $k = 0, 1, 2, \ldots$ The convergence and stability properties are thoroughly investigated, identifying an optimal fourth-order member according to the Kung-Traub's conjecture. The analysis reveals the complexity of the family and allows the identification of members with exceptional stability, capable of converging to practical solutions even from initial estimates distant to the solution. These results are validated with numerical tests, demonstrating the efficiency and reliability of the proposed methods.

The article *"Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations"* introduces a new parametric family of three-step iterative methods for solving non-linear equations. Initially, a three-parameter fourth-order family is designed, which, by fixing one of its parameters, accelerates its convergence, resulting in a single-parameter sixth-order family:

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \dfrac{f(y_k)}{2f[x_k, y_k] - f'(x_k)}, \\ x_{k+1} = z_k - (\alpha + \beta u_k + \gamma v_k) \dfrac{f(z_k)}{f'(x_k)}, \end{cases} \tag{1.2}$$

where $u_k = 1 - \dfrac{f[x_k, y_k]}{f'(x_k)}$, $v_k = \dfrac{f'(x_k)}{f[x_k, y_k]}$, $k = 0, 1, 2, \ldots$, α is an arbitrary parameter, $\beta = 1 + \alpha$, and $\gamma = 1 - \alpha$. The divided difference operator $f[\cdot, \cdot] : I \times I \subset \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ satisfies

$f[x,y](x-y) = f(x) - f(y)$, $\forall x, y \in I$. The convergence, complex dynamics, and numerical behavior of this latter family are studied. From the dynamical analysis, members with particularly stable behavior, suitable for solving practical problems, are identified. Several numerical tests illustrate the efficiency and stability of the proposed family.

The article *"Performance of a New Sixth-Order Class of Iterative Schemes for Solving Non-Linear Systems of Equations"* presents an extension of the single-parameter sixth-order family (1.2), initially designed to solve non-linear equations, to systems of equations. Based on the Ostrowski scheme, the class is constructed by adding a Newton step with a Jacobian matrix from the previous step and using a divided difference operator, resulting in a three-parameter scheme with fourth-order convergence. By adjusting two parameters, the convergence order is increased to six, forming a single-parameter family:

$$\begin{cases} y^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} = y^{(k)} - [2[x^{(k)}, y^{(k)}; F] - F'(x^{(k)})]^{-1}F(y^{(k)}), \\ x^{(k+1)} = z^{(k)} - (\alpha I + \beta u^{(k)} + \gamma v^{(k)})[F'(x^{(k)})]^{-1}F(z^{(k)}), \end{cases} \tag{1.3}$$

where $u^{(k)} = I - [F'(x^{(k)})]^{-1}[x^{(k)}, y^{(k)}; F]$, $v^{(k)} = [x^{(k)}, y^{(k)}; F]^{-1}F'(x^{(k)})$, $k = 0, 1, 2, \ldots$, α is an arbitrary parameter, $\beta = 1+\alpha$, and $\gamma = 1-\alpha$. The divided difference operator $[x, y; F]$ is the map $[\cdot, \cdot; F] : D \times D \subset \mathbb{R}^n \times \mathbb{R}^n \to \mathcal{L}(\mathbb{R}^n)$, satisfying $[x, y; F](x-y) = F(x) - F(y), \forall x, y \in D$. Dynamical studies and numerical developments are carried out to analyze the stability of the sixth-order family designed to solve systems of non-linear equations. Additionally, previous investigations on scalar functions allow the identification of those family members with stable performance for solving practical problems.

Thus, the objectives of this research focused on the development, analysis, and validation of new families of iterative methods for solving non-linear equations and systems of equations. Firstly, new families of iterative methods were developed to solve non-linear equations. Two classes were designed: a two-step family with third-order convergence and a three-step family with sixth-order convergence.

The second objective was to investigate the convergence and stability properties of the new families of iterative methods developed for solving non-linear equations. To achieve this, theoretical and experimental analyses were conducted, including the use of Taylor series approximations and dynamical tools. This objective aimed to ensure that the proposed methods are theoretically sound and practical for application to real-world problems, ensuring their performance and reliability.

The third objective was to extend one of the two families designed for solving non-linear equations to systems. The family that demonstrated the best characteristics in terms of convergence and stability, in this case, was family (1.2). The original scheme was adapted to address the solution of more complex and large-scale systems. This extension aims to improve the applicability and versatility of the constructed family of iterative methods.

The fourth objective of the research was to evaluate the numerical performance of the new family of iterative methods designed to solve systems of non-linear equations. Extensive numerical tests were conducted using a set of standard benchmark problems, and the results were analyzed in terms of accuracy, efficiency, and stability. This objective aims to validate the effectiveness of the proposed methods and position them as an improvement over current techniques in the field of numerical analysis.

Therefore, the research focused on designing new families of iterative methods for solving non-linear equations and systems, identifying members with the highest order of convergence and exceptional stability. Using dynamical tools and numerical tests, the values of the free parameters of the new families that offer the best performance were explored, aiming to provide specific recommendations for the practical use of the new iterative methods in various applications.

The remainder of the document is structured as follows. Chapter 2 addresses the development of the article *"Achieving Optimal Order in a Novel Family of Numerical Methods: Insights from Convergence and Dynamical Analysis Results."* Chapter 3 presents the article *"Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations."* Chapter 4 focuses on the article *"Performance of a New Sixth-Order Class of Iterative Schemes for Solving Non-Linear Systems of Equations."* Chapter 5 discusses the general results obtained. And, finally, Chapter 6 presents the most relevant conclusions and outlines possible future lines of research.

Chapter 2

Achieving optimal order in a novel family of numerical methods

Reference: *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*

Abstract: In this manuscript, we introduce a novel parametric family of multistep iterative methods designed for solving nonlinear equations. This family is derived from 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 exist 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 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.

2.1 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 [19, 20, 21]. Finding solutions ξ to nonlinear equations of the form $f(x) = 0$ stands as a classical yet formidable problem in the realm of Numerical Analysis, Applied Mathematics, and Engineering. Here, the function $f : I \subset \mathbb{R} \to \mathbb{R}$ 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 [22], [23] and [24], 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

$$x_{k+1} = g(x_k), \quad k = 0, 1, \ldots. \tag{2.1}$$

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 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 [25]):

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, [22]), 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 [22] 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 of the growing interest of researchers in recent years in multipoint methods, see for example [24, 22, 23]. 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 [22]. 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 [26, 24]. 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 [22]).

Theorem 2.1. *Let $g_1(x)$ and $g_2(x)$ be 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 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, [27], [28] or [24]). If after the reduction of functional evaluations the resulting method is not optimal, the weight function technique, introduced by Chun in [29], 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 [22] 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 [30] 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 [31]) 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 multistep method without memory 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 scheme.

Furthermore, in order to numerically test the behavior of the different iterative methods, Weerakoon and Fernando in [32] 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 [33] of the approximate computational order of convergence (ACOC),

$$p \approx ACOC = \frac{\ln \frac{|x_{k+1}-x_k|}{|x_k-x_{k-1}|}}{\ln \frac{|x_k-x_{k-1}|}{|x_{k-1}-x_{k-2}|}},\ \ k = 2, 3, \ldots. \tag{2.2}$$

On the other hand, the dynamical analysis of rational operators derived from iterative schemes, particularly when applied to low-degree 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 [34, 35, 36, 37, 38] 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 [39], Amat et al. [40], Neta et al. [41], Cordero et al. [42], Magreñán [43], Geum et al. in [44], 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 [45] and [46], 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.

2.2 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:

$$\begin{cases} y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta \left(\dfrac{f(y_k)}{f(x_k)}\right)^2\right) \dfrac{f(x_k)}{f'(x_k)}, \end{cases} \tag{2.3}$$

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.

Theorem 2.2 (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 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* (2.3) *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, 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

$$\begin{aligned} f(x_k) &= f(\xi) + f'(\xi)e_k + \frac{1}{2!}f''(\xi)e_k^2 + \frac{1}{3!}f'''(\xi)e_k^3 + \frac{1}{4!}f^{(iv)}(\xi)e_k^4 + \mathcal{O}(e_k^5) \\ &= f'(\xi)\left[e_k + \frac{1}{2!}\frac{f''(\xi)}{f'(\xi)}e_k^2 + \frac{1}{3!}\frac{f'''(\xi)}{f'(\xi)}e_k^3 + \frac{1}{4!}\frac{f^{(iv)}(\xi)}{f'(\xi)}e_k^4\right] + \mathcal{O}(e_k^5) \\ &= f'(\xi)\left[e_k + C_2e_k^2 + C_3e_k^3 + C_4e_k^4\right] + \mathcal{O}(e_k^5), \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} f'(x_k) &= f'(\xi) + f''(\xi)e_k + \frac{1}{2!}f'''(\xi)e_k^2 + \frac{1}{3!}f^{(iv)}(\xi)e_k^3 + \mathcal{O}(e_k^4) \\ &= f'(\xi)\left[1 + \frac{f''(\xi)}{f'(\xi)}e_k + \frac{1}{2!}\frac{f'''(\xi)}{f'(\xi)}e_k^2 + \frac{1}{3!}\frac{f^{(iv)}(\xi)}{f'(\xi)}e_k^3\right] + \mathcal{O}(e_k^4) \\ &= f'(\xi)\left[1 + 2C_2e_k + 3C_3e_k^2 + 4C_4e_k^3\right] + \mathcal{O}(e_k^4), \end{aligned} \tag{2.5}$$

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

By a direct division of (2.4) and (2.5),

$$\frac{f(x_k)}{f'(x_k)} = e_k - C_2e_k^2 + 2\left(C_2^2 - C_3\right)e_k^3 - \left(4C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{2.6}$$

Replacing (2.6) in (2.3), we have

$$y_k = \xi + (1-\alpha)e_k + \alpha C_2 e_k^2 - 2\alpha\left(C_2^2 - C_3\right)e_k^3 + \alpha\left(4C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{2.7}$$

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

$$\begin{aligned} f(y_k) = f'(\xi)\Big[&(1-\alpha)e_k + \left(\alpha^2-\alpha+1\right)C_2e_k^2 + \left(-2\alpha^2C_2^2 - \left(\alpha^3-3\alpha^2+\alpha-1\right)C_3\right)e_k^3 \\ &+ \left(5\alpha^2C_2^3 + \alpha^2(3\alpha-10)C_2C_3 + \left(\alpha^4-4\alpha^3+6\alpha^2-\alpha+1\right)C_4\right)e_k^4\Big] + \mathcal{O}\left(e_k^5\right). \end{aligned} \tag{2.8}$$

Dividing (2.8) by (2.4), we obtain

$$\begin{aligned} \frac{f(y_k)}{f(x_k)} &= (1-\alpha) + \alpha^2C_2e_k - \alpha^2\left((\alpha-3)C_3 + 3C_2^2\right)e_k^2 \\ &\quad + \alpha^2\left(\left(\alpha^2-4\alpha+6\right)C_4 + 2(2\alpha-7)C_2C_3 + 8C_2^3\right)e_k^3 + \mathcal{O}\left(e_k^4\right). \end{aligned} \tag{2.9}$$

Finally, substituting (2.6), (2.7) and (2.9), in the second step of family (2.3), we have

$$x_{k+1} = \xi + A_1e_k + A_2e_k^2 + A_3e_k^3 + A_4e_k^4 + \mathcal{O}\left(e_k^5\right), \tag{2.10}$$

where

$$\begin{aligned} A_1 &= -\alpha^2\delta + \alpha(\gamma+2\delta-1) - \beta - \gamma - \delta + 1, \\ A_2 &= \left(2\alpha^3\delta - \alpha^2(\gamma+\delta) - \alpha(\gamma+2\delta-1) + \beta+\gamma+\delta\right)C_2, \\ A_3 &= \left(-2\alpha^4\delta + \alpha^3(\gamma+8\delta) - \alpha^2(3\gamma+4\delta) - 2\alpha(\gamma+2\delta-1) + 2(\beta+\gamma+\delta)\right)C_3 \\ &\quad - \left(\alpha^4\delta + 8\alpha^3\delta - 2\alpha^2(2\gamma+3\delta) - 2\alpha(\gamma+2\delta-1) + 2(\beta+\gamma+\delta)\right)C_2^2, \\ A_4 &= \left(7\alpha^4\delta + 26\alpha^3\delta - \alpha^2(13\gamma+22\delta) - 4\alpha(\gamma+2\delta-1) + 4(\beta+\gamma+\delta)\right)C_2^3 \\ &\quad + \left(2\alpha^5\delta + 4\alpha^4\delta - \alpha^3(5\gamma+48\delta) + \alpha^2(19\gamma+31\delta) + 7\alpha(\gamma+2\delta-1) - 7(\beta+\gamma+\delta)\right)C_2C_3 \\ &\quad + \left(2\alpha^5\delta - \alpha^4(\gamma+10\delta) + 4\alpha^3(\gamma+5\delta) - 3\alpha^2(2\gamma+3\delta) - 3\alpha(\gamma+2\delta-1) + 3(\beta+\gamma+\delta)\right)C_4, \end{aligned} \tag{2.11}$$

being the error equation

$$
\begin{aligned}
e_{k+1} &= A_1 e_k + A_2 e_k^2 + A_3 e_k^3 + A_4 e_k^4 + \mathcal{O}\left(e_k^5\right) \\
&= \left(-\alpha^2\delta + \alpha(\gamma + 2\delta - 1) - \beta - \gamma - \delta + 1\right) e_k + \mathcal{O}\left(e_k^2\right),
\end{aligned} \tag{2.12}
$$

and the proof is finished. □

From Theorem 2.2, 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.

Theorem 2.3 (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 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* (2.3) *converges to ξ with order three, provided that $\beta = \dfrac{(\alpha-1)^2\left(\alpha^2\delta - \alpha - 1\right)}{\alpha^2}$ and $\gamma = \dfrac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}$, 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 $e_k = x_k - \xi$, $C_q = \dfrac{1}{q!}\dfrac{f^{(q)}(\xi)}{f'(\xi)}$, $q = 2, 3, \ldots$, and α, δ are arbitrary parameters.

Proof. Using the results of Theorem 2.2 to cancel A_1 and A_2 accompanying e_k and e_k^2 in (2.12), respectively; it must be satisfied that

$$
\begin{cases}
-\alpha^2\delta + \alpha(\gamma + 2\delta - 1) - \beta - \gamma - \delta + 1 = 0, \\
2\alpha^3\delta - \alpha^2(\gamma + \delta) - \alpha(\gamma + 2\delta - 1) + \beta + \gamma + \delta = 0.
\end{cases} \tag{2.13}
$$

It is clear that system (2.13) has infinite solutions for

$$
\beta = \frac{(\alpha-1)^2\left(\alpha^2\delta - \alpha - 1\right)}{\alpha^2} \quad \text{and} \quad \gamma = \frac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}, \tag{2.14}
$$

where α and δ are free parameters. Therefore, replacing (2.14) in (2.11), we obtain that

$$
\begin{aligned}
A_1 &= 0, \\
A_2 &= 0, \\
A_3 &= -\left(\alpha^4\delta - 2\right) C_2^2 + (\alpha - 1) C_3, \\
A_4 &= \left(7\alpha^4\delta - 9\right) C_2^3 + \left(2(\alpha - 3)\alpha^4\delta - 5\alpha + 12\right) C_2 C_3 - (\alpha - 3)(\alpha - 1) C_4,
\end{aligned} \tag{2.15}
$$

being the error equation

$$\begin{aligned} e_{k+1} &= A_3 e_k^3 + \mathcal{O}(e_k^4) \\ &= \left(-\left(\alpha^4 \delta - 2 \right) C_2^2 + (\alpha - 1) C_3 \right) e_k^3 + \mathcal{O}(e_k^4), \end{aligned} \tag{2.16}$$

and the proof is finished. □

According to the findings in Theorem 2.3, it is evident that the newly introduced bi-parametric family

$$\begin{cases} y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta \left(\dfrac{f(y_k)}{f(x_k)} \right)^2 \right) \left(\dfrac{f(x_k)}{f'(x_k)} \right), \end{cases} \tag{2.17}$$

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 2.3.1 (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 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* (2.17) *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 an only member of the family of optimal fourth-order of convergence.

Proof. Using the results of Theorem 2.3 to reduce the expression of A_3 accompanying e_k^3 in (2.15), 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

$$\epsilon = \alpha^4 \delta = 2, \tag{2.18}$$

Therefore, replacing (2.18) in (2.15), we obtain that

$$\begin{aligned} e_{k+1} &= A_3 e_k^3 + \mathcal{O}(e_k^4) \\ &= (\alpha-1)C_3 e_k^3 + \mathcal{O}\left(e_k^4\right), \end{aligned} \tag{2.19}$$

and the proof is finished. □

Based on the outcomes derived from Corollary 2.3.1, it becomes apparent that the recently introduced uni-parametric family, which we will call MCCTU(α),

$$\begin{cases} y_k = x_k - \alpha \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \left(\beta + \gamma \dfrac{f(y_k)}{f(x_k)} + \delta \left(\dfrac{f(y_k)}{f(x_k)}\right)^2\right) \left(\dfrac{f(x_k)}{f'(x_k)}\right), \end{cases} \tag{2.20}$$

where $k = 0, 1, 2, \ldots$, $\beta = \dfrac{(\alpha-1)^2\left(\alpha^2\delta - \alpha - 1\right)}{\alpha^2}$, $\gamma = \dfrac{2\alpha^3\delta - 2\alpha^2\delta + 1}{\alpha^2}$ 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 2.3.

2.3 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 respect 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 [47] and [48], 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 fixed point if it satisfies $R(z_0) = z_0$. If the fixed point is not a solution of the equation, it is called 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 [49] and Julia [50], includes both periodic points (of any period) and fixed points, considered as periodic points of unit period.

Theorem 2.4 (Fatou, Julia)**.** *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.

2.3.1 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 $f(x)$:

$$f(x) = (x - a)(x - b) = 0, \tag{2.21}$$

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

Let us remark that when MCCTU(α) is directly applied on $f(x)$, parameter α dissapears in the resulting rational expression; so, no dynamical analysis can be made. However, if we use parameter $\epsilon = \alpha^4\delta$ appearing in Corollary 2.3.1 the same class of iterative methods can be expressed as MCCTU(ϵ) and the dynamical analyisis can be made depending on ϵ.

Proposition 2.1 (rational operator R_f). *Let the polynomial equation $f(x)$ given in* (2.21)*, for $a, b \in \mathbb{C}$. Rational operator R_f related to MCCTU(ϵ) family given in* (2.20) *on $f(x)$, is*

$$R_f(x,\epsilon) = \frac{x^3\left(\epsilon - x^3 - 4x^2 - 5x - 2\right)}{x^3(\epsilon-2) - 5x^2 - 4x - 1}, \tag{2.22}$$

with $\epsilon \in \mathbb{C}$ 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 (2.20) 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 [51, 40, 52]) 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

$$R_f(x,\epsilon) = \left(h \circ A_f(x,\epsilon) \circ h^{-1}\right)(x) = \frac{x^3\left(\epsilon - x^3 - 4x^2 - 5x - 2\right)}{x^3(\epsilon-2) - 5x^2 - 4x - 1}, \tag{2.23}$$

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

From Proposition 2.1, if we set $\epsilon - 2 = 0$, we obtain

$$\delta = \frac{2}{\alpha^4}, \tag{2.24}$$

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

$$R_f(x) = \frac{x^4\left(x^2 + 4x + 5\right)}{5x^2 + 4x + 1}, \tag{2.25}$$

which does not depend on any free parameter.

2.3.2 Fixed Points

Now, we calculate all the fixed points of $R_f(x,\epsilon)$ given by (2.22), to afterwards analyze their character (attracting, repulsive or neutral or parabolic).

Proposition 2.2. *The fixed points of $R_f(x,\epsilon)$ are $x = 0$, $x = \infty$ and also five strange fixed points:*

- $ex_1 = 1$,

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

By using Equation (2.24)*, the strange fixed points* ex_2*,* ex_3*,* ex_4 *and* ex_5 *do not depend on any free parameter,*

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

Moreover, as can be observed, strange fixed points depending on ϵ *are conjugated,* $ex_{2,4}(\epsilon)$ *and* $ex_{3,5}(\epsilon)$*. If* $\epsilon = \frac{1}{4}$*, then* $ex_2 = ex_4 = -\frac{1}{2}$ *and* $ex_3 = ex_5 = -2$*, resulting in three strange fixed points. Also, when* $\epsilon = -20$*,* $ex_4 = ex_5 = 1$*, and when* $\epsilon = 0$*,* $ex_4 = ex_5 = -1$*, indicating the presence of four strange fixed points in each case.*

From Proposition 2.2, 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, previously to the Möbius transformation.

Proposition 2.3. *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* (2.24)*), all the strange fixed points are repulsive.*

Proof. We prove this result analyzing the stability of the fixed points found in Proposition 2.2. 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 afttracting, 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 2.1.

By a graphical and numerical study of $\left|R'_f(ex_i(\epsilon), \epsilon)\right|$, $i = 2, 3, 4, 5$, 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 Figures 2.2(a) and 2.2(b). Moreover, ex_1 can not be a superattractor as $\left|R'_f(1, \epsilon)\right| \neq 0$. □

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 2.3, 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 2.1 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 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.

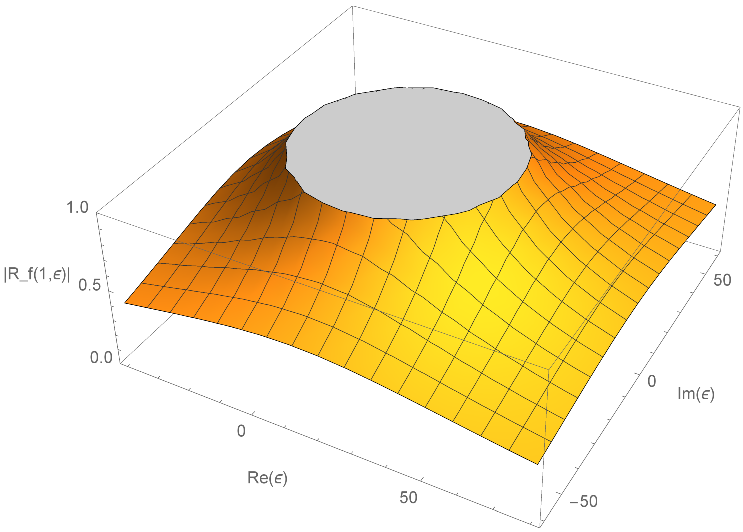
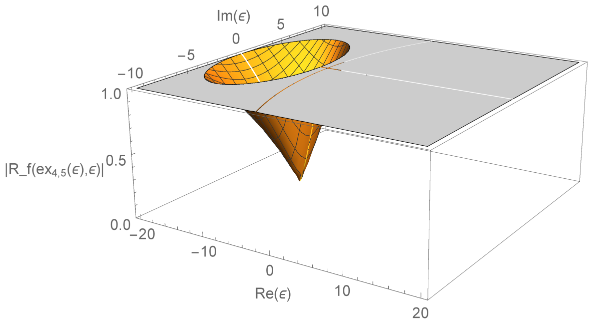
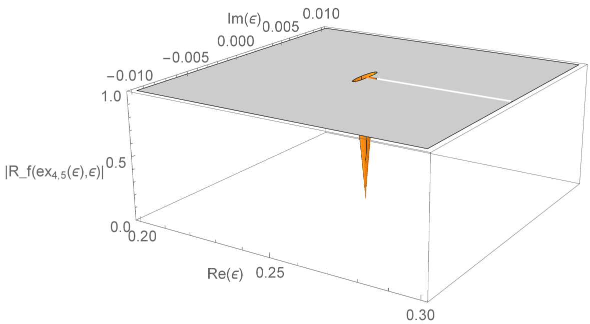


Figure 2.1: Stability function of $ex_1 = 1$, $\left|R'_f(1, \epsilon)\right|$ for a complex ϵ

Similar conclusions can be stated from the stability region of strange fixed points $ex_{4,5}(\epsilon)$, appearing in Figure 2.2. When a value of parameter ϵ is taken in the yellow area of Figure 2.2, both points are simultaneously attracting, so there are at least four different basins of attraction.



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



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

Figure 2.2: Stability surfaces of $ex_{4,5}(\epsilon)$ for different complex regions

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.

2.3.3 Critical Points

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

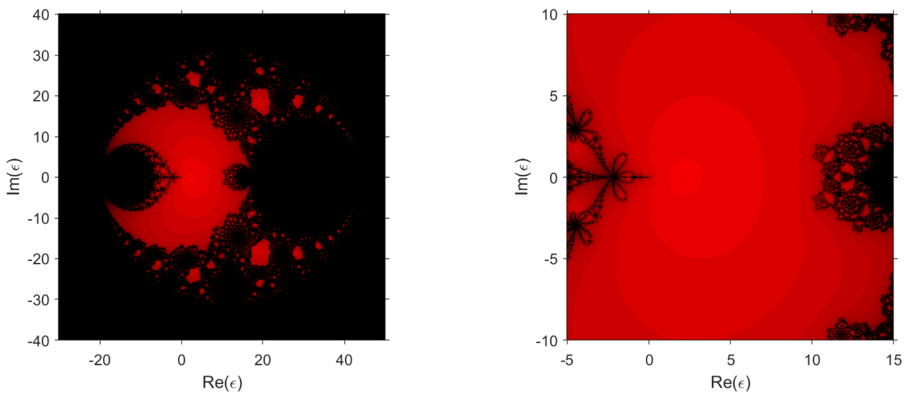
Proposition 2.4. *The critical points of $R_f(x,\epsilon)$ are $x=0$, $x=\infty$ and also:*

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

Moreover, 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 2.4, 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 2.3.2). Note that if the Equation (2.24) is satisfied, the only remaining free critical point is cr_1. Since cr_1 is the preimage 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 conjugate) 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 initial guess the critical point $cr_2(\epsilon)$ and calculate its orbit. If it converge to $x = 0$ or $x = \infty$, then the point corresponding to this value of ϵ is represented in 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 black area corresponds to schemes of the class 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 2.3 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 2.3(b)).



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

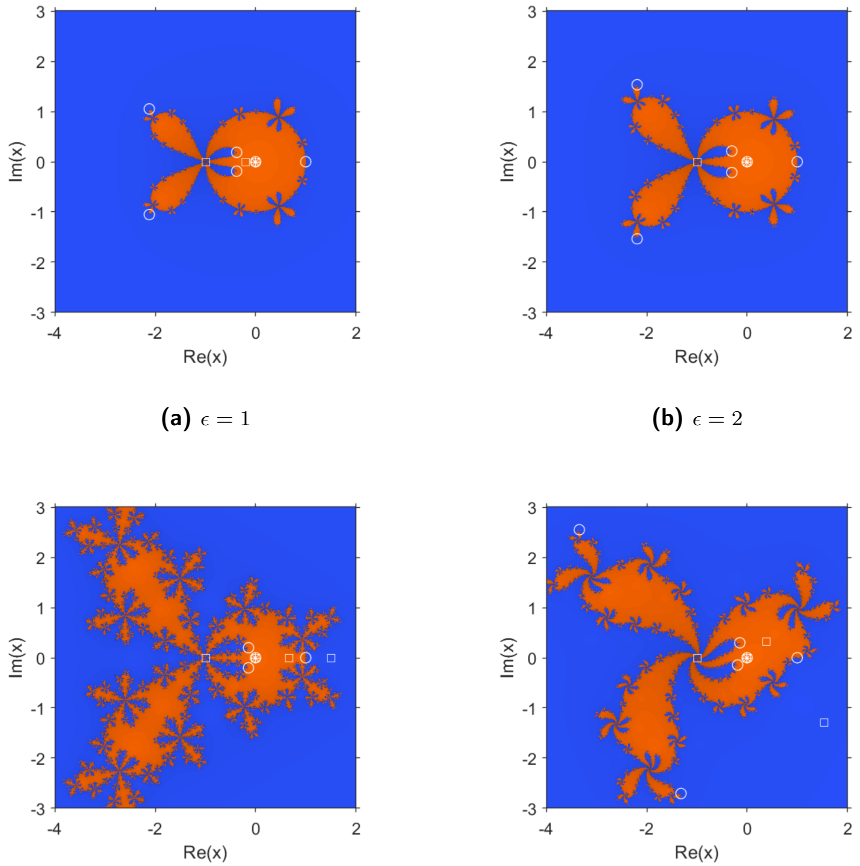
Figure 2.3: Parameter plane of $cr_2(\epsilon)$ on domain D_1 and a detail on D_2

2.3.4 Dynamical Planes

A dynamical plane is defined as a mesh in a limited domian 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 '$\bigcirc$', critical points are '$\square$', 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.

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 2.3(a).

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 2.4. 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).

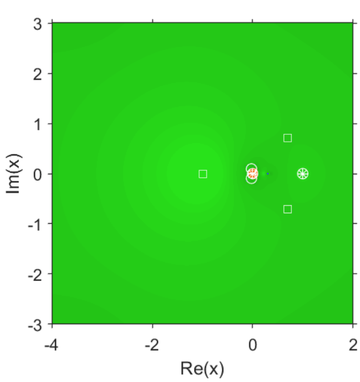


(a) $\epsilon = 1$ **(b)** $\epsilon = 2$

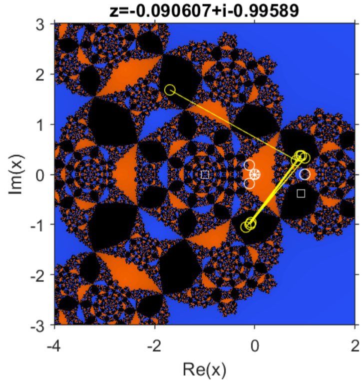
(c) $\epsilon = 10$ **(d)** $\epsilon = 5 + 5i$

Figure 2.4: Dynamical planes for some stable methods

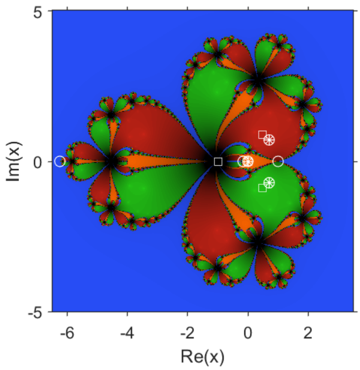
Secondly, some schemes outside the stability region (in black in the parameter plane) are provided for $\epsilon \in \{100, 15, -15, 30\}$. Their dynamical planes are shown in Figure 2.5. Each of these members have specific characteristics: in Figure 2.5(a), the widest basing of attraction (in green color) corresponds to $ex_1 = 1$, which is attracting for this value of ϵ, the basing of $x = 0$ is a very narrow area around the point; for $\epsilon = 15$, we observe in Figure 2.5(b) three different basins of attraction, being the third of two attracting periodic orbits of period 2 (one of them is plotted in yellow in the figure); Figure 2.5(c) corresponds to $\epsilon = -15$, inside the stability area of $ex_{4,5}(\epsilon)$ (see Figure 2.2), 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 2.5(d).



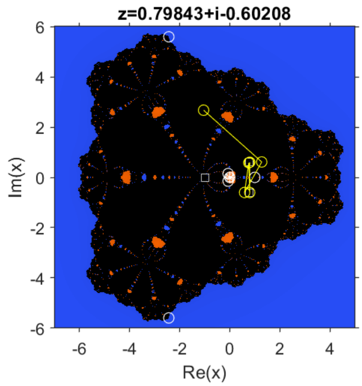
(a) $\epsilon = 100$



(b) $\epsilon = 15$



(c) $\epsilon = -15$



(d) $\epsilon = 30$

Figure 2.5: Unstable dynamical planes

2.4 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 (2.20) and apply them to ten nonlinear test equations, with their expressions and corresponding roots provided in Table 2.1.

We aim to confirm 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, $\epsilon = 2$.

Table 2.1: Nonlinear test equations and corresponding roots

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

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 [30, 53], King (KI) in [53, 54], Jarratt (JA) in [53, 55], Özban and Kaya (OK1, OK2, OK3) in [26], Chun (CH) in [56], Maheshwari (MA) in [57], Behl, Maroju, and Motsa (BMM) in [58], Chun et al. (CLND1, CLND2) in [59], Artidiello et al. (ACCT1, ACCT2) in [60], Ghanbari (GH) in [61], and Kou, Li, and Wang (KLW) in [62].

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. 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 [33]. 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'.

2.4.1 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 $\delta = \frac{2}{\alpha^4}$ becomes zero, causing δ to grow indefinitely.

The numerical performance of the iterative methods MCCTU(2) and MCCTU(100) is presented in Tables 2.2 and 2.3, 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 2.2: Numerical performance of MCCTU(2) method on nonlinear equations

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

Table 2.3: Numerical performance of MCCTU(100) method on nonlinear equations

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

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 2.3.

2.4.2 Second Experiment: Efficiency Analysis of MCCTU(α) Family

In this experiment, we conducted a comparative study between an optimal method of the MCCTU(α) family and the fifteen fourth-order methods mentioned in the introduction of Section 2.4, 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 2.4 to 2.18, 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 2.4: Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (1/5)

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC
f_1	MCCTU(1)	8.4069e-27	2.2344e-105	3	4.0111
$x_0 =$ -0.6	OS	1.2193e-29	2.7787e-117	3	4.0062
	KI	3.9435e-29	3.8183e-115	3	4.0070
	JA	1.3498e-29	4.2651e-117	3	4.0061
	OK1	5.0547e-32	2.9443e-127	3	3.9991
	OK2	4.0266e-30	2.6729e-119	3	4.0052
	OK3	2.5735e-30	4.5908e-120	3	3.9937
	CH	1.6691e-28	1.6213e-112	3	4.0081
	MA	3.0371e-27	3.1217e-107	3	4.0103
	BMM	1.2299e-28	4.4824e-113	3	4.0084
	CLND1	8.643e-27	2.5116e-105	3	4.0110
	CLND2	1.6691e-28	1.6213e-112	3	4.0081
	ACCT1	8.4069e-27	2.2344e-105	3	4.0111
	ACCT2	7.4417e-32	1.0756e-126	3	4.0294
	GH	1.9739e-26	8.0112e-104	3	4.0119
	KLW	8.4441e-28	1.4567e-109	3	4.0092
f_2	MCCTU(1)	4.0916e-36	1.3257e-144	3	3.9624
$x_0 =$ 0.2	OS	2.6718e-32	8.6963e-129	3	3.9998
	KI	1.7333e-32	1.4291e-129	3	3.9987
	JA	1.1553e-31	4.1074e-126	3	3.9990
	OK1	2.4295e-31	9.1464e-125	3	4.0008
	OK2	1.4863e-31	1.1754e-125	3	3.9997
	OK3	1.3844e-31	8.8054e-126	3	3.9988
	CH	5.002e-32	1.2502e-127	3	3.9969
	MA	2.1425e-34	1.6464e-137	3	3.9844
	BMM	5.5838e-31	2.8585e-123	3	4.0057
	CLND1	9.7338e-34	9.6229e-135	3	3.9830
	CLND2	5.002e-32	1.2502e-127	3	3.9969
	ACCT1	4.0916e-36	1.3257e-144	3	3.9624
	ACCT2	1.4832e-31	1.1243e-125	3	4.0029
	GH	2.5248e-38	6.6868e-154	3	3.9675
	KLW	1.8553e-33	1.2914e-133	3	3.9925

Table 2.5: Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (2/5)

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC
f_3	MCCTU(1)	2.2096e-83	0	4	4
$x_0 =$ 0.6	OS	1.7622e-27	3.3439e-108	3	3.9992
	KI	1.6297e-100	0	4	4
	JA	2.9708e-27	2.989e-107	3	3.9996
	OK1	6.1743e-100	1.9467e-208	4	4
	OK2	1.0137e-33	6.7493e-135	3	4.0975
	OK3	1.0148e-27	3.9188e-110	3	4.2357
	CH	2.1262e-94	0	4	4
	MA	6.9765e-86	0	4	4
	BMM	1.6076e-85	1.9467e-208	4	4
	CLND1	2.5512e-83	0	4	4
	CLND2	2.1262e-94	0	4	4
	ACCT1	2.2096e-83	6.8135e-208	4	4
	ACCT2	2.5202e-91	0	4	4
	GH	2.2217e-81	0	4	4
	KLW	2.4336e-89	0	4	4
f_4	MCCTU(1)	2.4812e-61	0	3	4
$x_0 =$ -14.1	OS	5.7494e-76	0	3	4
	KI	2.6178e-66	0	3	4
	JA	4.5662e-69	3.8934e-208	3	4
	OK1	1.6181e-64	0	3	4
	OK2	1.2341e-67	0	3	4
	OK3	4.782e-68	3.8934e-208	3	3.9998
	CH	4.1273e-64	0	3	4
	MA	5.9003e-62	0	3	4
	BMM	2.4555e-61	3.8934e-208	3	4
	CLND1	2.8374e-61	0	3	4
	CLND2	4.1273e-64	0	3	4
	ACCT1	2.4812e-61	0	3	4
	ACCT2	7.6144e-63	0	3	4
	GH	7.562e-61	0	3	4
	KLW	7.8025e-63	0	3	4

Table 2.6: Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (3/5)

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

Table 2.7: Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (4/5)

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_7	MCCTU(1)	8.109e-29	1.224e-110	3	4.0025
$x_0 =$ -1.2	OS	1.0259e-36	8.588e-144	3	3.9987
	KI	2.2e-33	8.266e-130	3	4.0003
	JA	1.3275e-35	3.987e-139	3	3.9993
	OK1	2.8995e-32	4.1375e-125	3	4.0011
	OK2	4.5559e-36	4.3528e-141	3	4.0014
	OK3	3.3899e-35	9.9763e-138	3	4.0602
	CH	1.5282e-31	4.4541e-122	3	4.0011
	MA	1.9806e-29	3.2969e-113	3	4.0021
	BMM	5.13e-29	1.8531e-111	3	3.9988
	CLND1	8.8475e-29	1.7657e-110	3	4.0024
	CLND2	1.5282e-31	4.4541e-122	3	4.0011
	ACCT1	8.109e-29	1.224e-110	3	4.0025
	ACCT2	1.7685e-30	1.2708e-117	3	4.0037
	GH	2.4542e-28	1.2766e-108	3	4.0029
	KLW	2.7616e-30	8.4579e-117	3	4.0014
f_8	MCCTU(1)	3.2362e-36	1.278e-144	3	4.0010
$x_0 =$ 2.3	OS	4.7781e-35	1.1082e-139	3	3.9959
	KI	3.867e-35	4.5395e-140	3	3.9962
	JA	6.4886e-36	2.6103e-143	3	3.9934
	OK1	1.3631e-35	5.9439e-142	3	3.9927
	OK2	8.3354e-36	7.4954e-143	3	3.9931
	OK3	8.2958e-36	7.3117e-143	3	3.9935
	CH	2.8017e-36	7.5934e-145	3	3.9943
	MA	7.3689e-36	4.1437e-143	3	3.9992
	BMM	2.6822e-34	1.5979e-136	3	3.9934
	CLND1	5.1248e-38	3.528e-152	3	4.0005
	CLND2	2.8017e-36	7.5934e-145	3	3.9943
	ACCT1	3.2362e-36	1.278e-144	3	4.0010
	ACCT2	1.2316e-34	5.9984e-138	3	3.9946
	GH	1.2035e-36	1.9404e-146	3	4.0036
	KLW	1.4928e-35	8.1719e-142	3	3.9978

Table 2.8: Numerical performance of iterative methods on nonlinear equations for x_0 **close to** ξ (5/5)

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC
f_9	MCCTU(1)	1.2504e-28	6.0286e-111	3	3.9982
$x_0 =$ -1.4	OS	2.2297e-33	5.9539e-131	3	4.0107
	KI	3.571e-39	4.8453e-155	3	3.9663
	JA	6.6365e-33	5.9006e-129	3	4.0095
	OK1	1.7043e-30	8.4881e-119	3	4.0019
	OK2	8.1242e-32	2.3078e-124	3	4.0049
	OK3	1.3061e-31	1.4689e-123	3	4.0184
	CH	5.6961e-33	3.922e-129	3	3.9887
	MA	2.4063e-29	5.9988e-114	3	3.9973
	BMM	2.911e-28	2.1169e-109	3	3.9971
	CLND1	1.0887e-28	3.3751e-111	3	3.9980
	CLND2	5.6961e-33	3.922e-129	3	3.9887
	ACCT1	1.2504e-28	6.0286e-111	3	3.9982
	ACCT2	1.7434e-29	1.473e-114	3	4.0025
	GH	4.3546e-28	1.1301e-108	3	3.9989
	KLW	2.0248e-30	1.8702e-118	3	3.9955
f_{10}	MCCTU(1)	1.3096e-27	1.0557e-105	3	4.0263
$x_0 =$ -0.9	OS	1.2157e-28	5.2236e-110	3	4.0178
	KI	1.6268e-28	1.7588e-109	3	4.0189
	JA	2.5808e-28	1.2592e-108	3	4.0158
	OK1	1.2566e-28	6.3023e-110	3	4.0126
	OK2	2.0733e-28	5.0608e-109	3	4.0149
	OK3	1.9638e-28	4.0898e-109	3	4.0133
	CH	4.7545e-28	1.6033e-107	3	4.0184
	MA	7.9356e-28	1.3045e-106	3	4.0246
	BMM	1.3934e-30	4.5027e-118	3	3.9969
	CLND1	2.1208e-27	8.164e-105	3	4.0242
	CLND2	4.7545e-28	1.6033e-107	3	4.0184
	ACCT1	1.3096e-27	1.0557e-105	3	4.0263
	ACCT2	1.9256e-29	2.4651e-113	3	4.0090
	GH	2.0723e-27	7.171e-105	3	4.0278
	KLW	4.546e-28	1.2771e-107	3	4.0226

In Tables 2.4 to 2.8, 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 2.9 to 2.13 and 2.14 to 2.18.

Table 2.9: Numerical performance of iterative methods on nonlinear equations for x_0 **far from** ξ (1/5)

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_1	MCCTU(1)	3.15e-28	4.4044e-111	4	3.9913
$x_0 =$ -1.8	OS	5.8375e-36	1.46e-142	4	3.9979
	KI	1.5765e-34	9.7538e-137	4	3.9972
	JA	5.3832e-35	1.0789e-138	4	3.9976
	OK1	9.9392e-40	4.4016e-158	4	4.0001
	OK2	2.4525e-36	3.6785e-144	4	3.9982
	OK3	1.1878e-33	2.0829e-133	4	4.0017
	CH	2.7866e-32	1.2595e-127	4	3.9958
	MA	2.0068e-29	5.9514e-116	4	3.9929
	BMM	1.3126e-31	5.814e-125	4	4.0050
	CLND1	7.4349e-28	1.3753e-109	4	3.9907
	CLND2	2.7866e-32	1.2595e-127	4	3.9958
	ACCT1	3.15e-28	4.4044e-111	4	3.9913
	ACCT2	6.7276e-44	7.1842e-175	4	3.9954
	GH	2.7189e-27	2.8837e-107	4	3.9896
	KLW	7.9363e-31	1.1367e-121	4	3.9946
f_2	MCCTU(1)	6.8509e-86	0	4	4
$x_0 =$ 0.6	OS	7.8707e-82	0	4	4
	KI	4.1628e-82	0	4	4
	JA	5.9451e-78	7.7869e-208	4	4
	OK1	1.7391e-77	7.7869e-208	4	4
	OK2	8.4717e-78	0	4	4
	OK3	8.9827e-78	0	4	4
	CH	1.8951e-78	0	4	4
	MA	1.8733e-84	0	4	4
	BMM	1.2206e-79	0	4	4
	CLND1	2.1212e-80	0	4	4
	CLND2	1.8951e-78	0	4	4
	ACCT1	6.8509e-86	0	4	4
	ACCT2	1.366e-80	0	4	4
	GH	1.4879e-88	0	4	4
	KLW	2.1154e-83	0	4	4

Table 2.10: Numerical performance of iterative methods on nonlinear equations for x_0 **far from** ξ (2/5)

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_3	MCCTU(1)	6.0868e-31	6.9016e-121	5	3.9978
$x_0 =$ 1.8	OS	7.2812e-73	0	5	4
	KI	8.2846e-59	0	5	4
	JA	6.0259e-71	0	5	4
	OK1	1.1879e-82	0	5	4
	OK2	6.2111e-27	9.5138e-108	4	4.1522
	OK3	7.5783e-53	0	5	4.0205
	CH	9.6923e-49	1.3715e-192	5	3.9999
	MA	1.3275e-34	1.1979e-135	5	3.9989
	BMM	nc	nc	nc	nc
	CLND1	9.5034e-31	4.1315e-120	5	3.9978
	CLND2	9.6923e-49	1.3715e-192	5	3.9999
	ACCT1	6.0868e-31	6.9016e-121	5	3.9978
	ACCT2	6.1271e-31	2.8102e-121	4	3.9953
	GH	1.4039e-28	2.4077e-111	5	3.9965
	KLW	4.5965e-39	1.1996e-153	5	3.9996
f_4	MCCTU(1)	nc	nc	nc	nc
$x_0 =$ -42.3	OS	2.602e-54	0	6	4.0004
	KI	nc	nc	nc	nc
	JA	1.0645e-51	0	6	4
	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

Table 2.11: Numerical performance of iterative methods on nonlinear equations for x_0 **far from** ξ (3/5)

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

Table 2.12: Numerical performance of iterative methods in nonlinear equations for x_0 **far from** ξ (4/5)

Function	Method	$\lvert x_{k+1}-x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_7	MCCTU(1)	2.1695e-40	6.2709e-157	12	3.9997
$x_0 =$ -3.6	OS	8.7445e-42	4.5328e-164	9	4.0005
	KI	3.5832e-59	0	10	4
	JA	1.445e-33	5.5984e-131	9	4.0010
	OK1	4.5904e-56	0	9	4
	OK2	2.2993e-100	0	9	4
	OK3	4.4822e-55	0	11	3.9955
	CH	2.5759e-93	0	11	4
	MA	1.0752e-70	0	12	4
	BMM	nc	nc	nc	nc
	CLND1	5.1735e-38	2.0643e-147	12	3.9995
	CLND2	2.5759e-93	0	11	4
	ACCT1	2.1695e-40	6.2709e-157	12	3.9997
	ACCT2	7.4341e-57	0	8	4
	GH	2.2938e-28	9.7412e-109	12	3.9971
	KLW	6.4489e-31	2.515e-119	11	3.9988
f_8	MCCTU(1)	8.9717e-34	7.5485e-135	5	3.9964
$x_0 =$ 6.9	OS	3.5465e-42	3.3638e-168	5	3.9988
	KI	3.0788e-46	1.8241e-184	5	3.9994
	JA	3.8134e-44	3.1142e-176	5	3.9984
	OK1	4.9365e-41	1.0225e-163	5	3.9972
	OK2	1.6379e-42	1.1175e-169	5	3.9979
	OK3	9.2522e-52	1.0123e-206	5	4.0004
	CH	6.7803e-74	1.5574e-207	5	4
	MA	4.1752e-36	4.2707e-144	5	4.0002
	BMM	1.4528e-98	1.5574e-207	5	4
	CLND1	2.6243e-36	2.4259e-145	5	3.9960
	CLND2	6.7803e-74	1.5574e-207	5	4
	ACCT1	8.9717e-34	7.5485e-135	5	3.9964
	ACCT2	7.0924e-38	6.5962e-151	5	3.9970
	GH	9.3051e-33	6.9333e-131	5	3.9867
	KLW	1.1619e-39	2.9996e-158	5	4.0009

Table 2.13: Numerical performance of iterative methods in nonlinear equations for x_0 **far from** ξ (5/5)

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_9	MCCTU(1)	1.9461e-27	3.5371e-106	6	4.0014
$x_0 =$ -4.2	OS	4.9716e-100	0	6	4
	KI	2.3907e-73	0	6	4.0003
	JA	1.5785e-91	0	6	4
	OK1	7.3314e-101	0	6	4
	OK2	5.2967e-94	0	6	4
	OK3	3.3512e-48	6.3655e-190	6	3.9987
	CH	1.4014e-54	0	6	4.0003
	MA	2.5559e-32	7.6354e-126	6	4.0012
	BMM	nc	nc	nc	nc
	CLND1	1.866e-27	2.9131e-106	6	4.0016
	CLND2	1.4014e-54	0	6	4.0003
	ACCT1	1.9461e-27	3.5371e-106	6	4.0014
	ACCT2	7.4372e-40	4.8779e-156	5	3.9995
	GH	3.5153e-95	0	7	4
	KLW	1.7919e-38	1.147e-150	6	4.0009
f_{10}	MCCTU(1)	1.4573e-79	1.0707e-207	6	3.9998
$x_0 =$ -2.7	OS	5.9731e-31	3.7644e-111	10	4.0223
	KI	3.5019e-84	3.3094e-207	6	4.0006
	JA	2.7724e-41	1.3193e-154	5	3.9916
	OK1	4.672e-37	3.3675e-141	5	4.0067
	OK2	6.9502e-38	2.3984e-144	5	3.9545
	OK3	4.1808e-48	2.2923e-179	5	4.0014
	CH	1.8136e-97	3.8389e-205	6	4
	MA	2.2689e-93	3.8934e-208	5	4
	BMM	2.6823e-41	6.1829e-161	6	3.9999
	CLND1	8.2625e-101	2.7254e-207	5	4
	CLND2	1.8136e-97	3.8389e-205	6	4
	ACCT1	1.4573e-79	1.0707e-207	6	3.9998
	ACCT2	3.3917e-58	2.3908e-204	6	4.0003
	GH	2.2337e-72	1.0707e-207	5	3.9994
	KLW	2.4673e-32	6.302e-122	4	3.8791

The results presented in Tables 2.9 to 2.13 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 2.14: Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ (1/5)

Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC
f_1	MCCTU(1)	3.9494e-95	0	6	4
$x_0 =$ -6.0	OS	7.5454e-40	4.0753e-158	5	3.9989
	KI	2.2846e-36	4.3008e-144	5	3.9980
	JA	4.2789e-41	4.3067e-163	5	3.9992
	OK1	2.8437e-53	0	5	4
	OK2	2.0962e-46	1.9632e-184	5	3.9997
	OK3	1.9553e-33	1.5296e-132	5	4.0018
	CH	1.6161e-33	1.4248e-132	5	3.9966
	MA	2.632e-26	1.7609e-103	5	3.9875
	BMM	nc	nc	nc	nc
	CLND1	5.8255e-95	0	6	4
	CLND2	1.6161e-33	1.4248e-132	5	3.9966
	ACCT1	3.9494e-95	0	6	4
	ACCT2	5.5395e-58	0	5	3.9996
	GH	6.2374e-89	0	6	4
	KLW	1.0186e-28	3.0849e-113	5	3.9921
f_2	MCCTU(1)	1.0368e-34	5.4646e-139	4	4.0222
$x_0 =$ 2.0	OS	2.3862e-93	0	5	4
	KI	1.2873e-25	4.3485e-102	4	3.9933
	JA	4.1797e-95	0	5	4
	OK1	8.2892e-82	0	5	4
	OK2	5.051e-87	0	5	4
	OK3	4.2138e-33	7.557e-132	4	3.9991
	CH	1.639e-29	1.4412e-117	4	3.9949
	MA	9.1807e-43	5.5512e-171	4	4.0028
	BMM	2.9675e-52	0	6	3.9998
	CLND1	9.0974e-32	7.3425e-127	4	4.0155
	CLND2	1.639e-29	1.4412e-117	4	3.9949
	ACCT1	1.0368e-34	5.4646e-139	4	4.0222
	ACCT2	1.9358e-73	0	5	4
	GH	6.4242e-33	2.803e-132	4	4.0791
	KLW	1.6753e-39	8.5848e-158	4	3.9976

Table 2.15: Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ (2/5)

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC
f_3	MCCTU(1)	1.0037e-74	0	9	4
$x_0 = 6.0$	OS	1.4888e-45	1.7033e-180	7	4
	KI	3.4193e-27	1.1139e-106	7	3.9978
	JA	2.4788e-42	1.4488e-167	7	4
	OK1	7.5531e-64	0	7	4
	OK2	5.2399e-91	0	7	4
	OK3	2.024e-56	0	8	4.0130
	CH	3.2307e-66	6.8135e-208	8	4
	MA	2.7511e-26	2.21e-102	8	3.9951
	BMM	nc	nc	nc	nc
	CLND1	5.7924e-73	0	9	4
	CLND2	3.2307e-66	6.8135e-208	8	4
	ACCT1	1.0037e-74	0	9	4
	ACCT2	1.643e-31	1.4531e-123	6	4.0042
	GH	1.8548e-60	0	9	4
	KLW	1.1156e-35	4.1633e-140	8	3.9992
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

Table 2.16: Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ (3/5)

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

Table 2.17: Numerical performance of iterative methods on nonlinear equations for x_0 **very far from** ξ (4/5)

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
	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	1.2624e-39	3.2997e-154	50	4.0007
	GH	nc	nc	nc	nc
	KLW	nc	nc	nc	nc
f_8	MCCTU(1)	7.1071e-32	2.9726e-127	6	3.9944
$x_0 =$ 23.0	OS	3.9961e-44	5.4218e-176	6	3.9991
	KI	1.9961e-46	3.223e-185	6	3.9995
	JA	2.8208e-93	1.5574e-207	6	4
	OK1	1.2245e-30	3.8716e-122	5	3.9812
	OK2	1.3604e-44	5.3174e-178	5	3.9985
	OK3	1.477e-56	1.5574e-207	6	3.9998
	CH	3.6894e-61	1.5574e-207	6	3.9999
	MA	5.2575e-34	1.0738e-135	6	4.0001
	BMM	3.7259e-45	3.1549e-179	7	4.0007
	CLND1	1.0076e-36	5.271e-147	6	3.9964
	CLND2	3.6894e-61	1.5574e-207	6	3.9999
	ACCT1	7.1071e-32	2.9726e-127	6	3.9944
	ACCT2	1.1468e-52	1.5574e-207	6	3.9997
	GH	3.8246e-31	1.9787e-124	6	3.9800
	KLW	1.5819e-37	1.0306e-149	6	4.0012

Table 2.18: Numerical performance of iterative methods in nonlinear equations for x_0 **very far from** ξ (5/5)

Function	Method	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC
f_9	MCCTU(1)	7.6712e-52	8.5422e-204	9	4
$x_0 =$ -14.0	OS	2.3748e-99	0	8	4
	KI	7.9344e-69	0	8	4.0006
	JA	1.0139e-94	0	8	4
	OK1	3.2135e-50	1.0728e-197	7	3.9999
	OK2	1.5049e-32	2.7173e-127	7	3.9952
	OK3	7.1691e-34	1.3332e-132	8	3.9822
	CH	8.1205e-44	1.62e-172	8	4.0015
	MA	5.7604e-70	0	9	4
	BMM	nc	nc	nc	nc
	CLND1	9.8978e-52	2.306e-203	9	4
	CLND2	8.1205e-44	1.62e-172	8	4.0015
	ACCT1	7.6712e-52	8.5422e-204	9	4
	ACCT2	7.9723e-43	6.4405e-168	7	4.0003
	GH	3.8751e-42	7.0871e-165	9	4.0001
	KLW	1.1099e-94	0	9	4
f_{10}	MCCTU(1)	1.2776e-70	1.0707e-207	6	4.0008
$x_0 =$ -9.0	OS	2.9225e-29	1.7446e-112	5	3.9821
	KI	7.9476e-94	1.5574e-207	8	4
	JA	6.1519e-29	1.4803e-108	8	4.0703
	OK1	2.3282e-63	9.7336e-208	6	4
	OK2	2.4314e-75	1.9467e-208	7	4
	OK3	1.9369e-93	4.9203e-206	6	4
	CH	1.0491e-37	1.9596e-135	8	4.0092
	MA	9.5564e-89	3.8934e-208	6	4
	BMM	nc	nc	nc	nc
	CLND1	1.5067e-28	4.9584e-107	12	3.7072
	CLND2	1.0491e-37	1.9596e-135	8	4.0092
	ACCT1	1.2776e-70	1.0707e-207	6	4.0008
	ACCT2	1.463e-49	5.2801e-191	5	4.0594
	GH	2.4513e-45	6.3226e-174	7	4.0044
	KLW	5.7956e-39	1.4646e-150	6	4.0153

The results presented in Tables 2.14 to 2.18 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.

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.

2.5 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 3 or 4 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.

2.6 Appendix

2.6.1 Detailed Computation of Theorem 2.2

The comprehensive proof of Theorem 2.2, methodically detailed step-by-step in Section 2.2, is further validated in Wolfram Mathematica software 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
```

2.6.2 Detailed Computation of Theorem 2.3

The comprehensive proof of Theorem 2.3, methodically detailed step-by-step in Section 2.2, is further validated in Wolfram Mathematica software 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}]
```

2.6.3 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 2.1. Thus, the numerical results are presented in Table 2.19.

Table 2.19: Numerical results of MCCTU(1) in practical calculations for x_0 **close to** ξ

Function	x_0	$\lvert x_{k+1} - x_k \rvert$	iter	ACOC	ξ
f_1	-0.6	8.4069e-27	3	4.0111	-0.6367
f_2	0.2	4.0915e-36	3	3.9624	0.2575
f_3	0.6	1.8066e-21	3	4.0121	0.6392
f_4	-14.1	3.6467e-15	2	-	-14.1013
f_5	1.3	1.2827e-20	3	4.0226	1.3652
f_6	0.1	1.1439e-32	3	3.9969	0.1281
f_7	-1.2	8.1090e-29	3	4.0025	-1.2076
f_8	2.3	3.2363e-36	3	4.0010	2.3320
f_9	-1.4	1.2504e-28	3	3.9982	-1.4142
f_{10}	-0.9	1.3096e-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 (2.2) requires at least 3 iterations to calculate the approximate order of convergence.

Chapter 3

Chaos and stability in a new iterative family for solving nonlinear equations

Reference: *Cordero, A.; Moscoso-Martínez, M.; Torregrosa, J.R. Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations. Algorithms 2021, 14, 101. https://doi.org/10.3390/a14040101*

Abstract: In this paper, we present a new parametric family of three-steps iterative for solving nonlinear equations. Firstly, we design a fourth-order triparametric family that, by holding only one of its parameters, we get to accelerate its convergence and finally obtain a sixth-order uniparametric family. With this last family we study its convergence, its complex dynamics (stability) and its numerical behavior. The parameter spaces and dynamical planes are presented showing the complexity of the family. From the parameter spaces we have been able to determine different members of the family that have bad convergence properties, since attracting periodic orbits and attracting strange fixed points appear in their dynamical planes. Moreover, this same study has allowed us to detect family members with especially stable behavior and suitable for solving practical problems. Several numerical tests are performed to illustrate the efficiency and stability of the presented family.

Keywords: Nonlinear equations; multistep iterative methods; convergence analysis; complex dynamics; chaos and stability.

3.1 Introduction

Many problems in Computational Sciences and other disciplines can be stated in the form of a nonlinear equation or nonlinear systems using mathematical modelling. In particular, a large number of problems in Applied Mathematics and Engineering are solved by finding the solutions of these equations.

In the literature there are many methods and families of iterative schemes, that have been designed by using different procedures, to approximate the simple roots of a nonlinear equation $f(x) = 0$, where $f : I \subseteq \mathbb{R} \to \mathbb{R}$ is a real function defined in an open interval I. We can find in [63, 23, 64] several surveys and overviews of the iterative schemes published in the last years. Each method has a different behavior. This behavior is characterized with the efficiency criteria and the complex dynamics tools.

In this paper, we introduce a new family of multistep iterative schemes to solve nonlinear equations, which contains as an element of this family, a particular method presented in [65]. This family is built from the Ostrowski's scheme, adding a Newton step with a "frozen" derivative and using a divided difference operator. So, the family has a three-step iterative expression. Furthermore, it has three arbitrary parameters named α, β and γ, which can take real or complex values, and an order of convergence of at least four. The order of convergence will be discussed in Section 3.2.

From the error equation we observe, by fixing two parameters in function of the third one, an uniparametric family of sixth-order iterative methods is obtained. We analyze the dynamical behavior of this family in terms of values of the parameter, in order to detect its elements with good stability properties and others with chaotic behavior. The concept of chaos has been widely discussed (see, for example, [66]) and it is commonly understood as the presence of complex orbit structure and extreme sensitivity of orbits to small perturbations. Moreover, the presence of unstable periodic orbits of all periods, is also included in the concept of chaotic system. For this study, we use tools of discrete complex dynamics that we introduce in Section 3.3.

In Section 3.4 we present the performance of the presented schemes on several test functions. These numerical tests allow us to confirm the results obtained in the dynamical section and to compare our schemes with other known ones. The manuscript finishes with some conclusions and the references used in it.

The parametric family object of study in this manuscript has the following iterative expression:

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \dfrac{f(y_k)}{2f[x_k, y_k] - f'(x_k)}, \\ x_{k+1} = z_k - (\alpha + \beta u_k + \gamma v_k)\, \dfrac{f(z_k)}{f'(x_k)}, \end{cases} \tag{3.1}$$

where $u_k = 1 - \frac{f[x_k, y_k]}{f'(x_k)}$, $v_k = \frac{f'(x_k)}{f[x_k, y_k]}$, $k = 0, 1, 2, \ldots$, and α, β, γ are arbitrary parameters. The divided difference operator $f[\cdot,\cdot] : I \times I \subset \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defined by Ortega and Rheinboldt in [67], satisfies

$$f[x, y](x - y) = f(x) - f(y), \quad \forall x, y \in I. \tag{3.2}$$

3.2 Convergence of the New Family

In this section, we perform the convergence analysis of the new triparametric iterative family. Furthermore, we propose a strategy to reduce the triparametric scheme to an uniparametric scheme in order to accelerate the convergence.

Theorem 3.1. *Let $f : I \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function on an open interval I and $\xi \in I$ a simple root of the nonlinear equation $f(x) = 0$. Suppose that $f(x)$ is continuous and sufficiently differentiable in a neighborhood of the simple root 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* (3.1) *converges to ξ with an order of convergence of four, being its error equation*

$$e_{k+1} = (1 - \alpha - \gamma) C_2 \left(C_2^2 - C_3 \right) e_k^4 + \mathcal{O}\left(e_k^5 \right),$$

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

Proof. Let ξ be a simple root of $f(x)$ (that is, $f(\xi) = 0$ and $f'(\xi) \neq 0$) and $x_k = \xi + e_k$. Using Taylor expansion of $f(x_k)$ and $f'(x_k)$ around ξ, we have

$$\begin{aligned} f(x_k) &= f(\xi + e_k) \\ &= f(\xi) + f'(\xi) e_k + \frac{1}{2!} f''(\xi) e_k^2 + \frac{1}{3!} f'''(\xi) e_k^3 + \frac{1}{4!} f^{(iv)}(\xi) e_k^4 + \mathcal{O}(e_k^5) \\ &= f'(\xi) \left[e_k + \frac{1}{2!} \frac{f''(\xi)}{f'(\xi)} e_k^2 + \frac{1}{3!} \frac{f'''(\xi)}{f'(\xi)} e_k^3 + \frac{1}{4!} \frac{f^{(iv)}(\xi)}{f'(\xi)} e_k^4 + \mathcal{O}(e_k^5) \right] \\ &= f'(\xi) \left[e_k + C_2 e_k^2 + C_3 e_k^3 + C_4 e_k^4 + \mathcal{O}(e_k^5) \right], \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} f'(x_k) &= f'(\xi + e_k) \\ &= f'(\xi) + f''(\xi)e_k + \frac{1}{2!}f'''(\xi)e_k^2 + \frac{1}{3!}f^{(iv)}(\xi)e_k^3 + \mathcal{O}(e_k^4) \\ &= f'(\xi)\left[1 + \frac{f''(\xi)}{f'(\xi)}e_k + \frac{1}{2!}\frac{f'''(\xi)}{f'(\xi)}e_k^2 + \frac{1}{3!}\frac{f^{(iv)}(\xi)}{f'(\xi)}e_k^3 + \mathcal{O}(e_k^4)\right] \\ &= f'(\xi)\left[1 + 2C_2e_k + 3C_3e_k^2 + 4C_4e_k^3 + \mathcal{O}(e_k^4)\right], \end{aligned} \tag{3.4}$$

where $C_q = \dfrac{1}{q!}\dfrac{f^{(q)}(\xi)}{f'(\xi)}$, $q = 2, 3, ...$

Dividing (3.3) by (3.4), we get

$$\frac{f(x_k)}{f'(x_k)} = e_k - C_2e_k^2 + 2\left(C_2^2 - C_3\right)e_k^3 - \left(4C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{3.5}$$

Replacing (3.5) in the first step of family (3.1), we have

$$y_k = \xi + C_2e_k^2 - 2\left(C_2^2 - C_3\right)e_k^3 + \left(4C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{3.6}$$

Using Taylor expansion again, similar to (3.3), to develop $f(y_k)$ around ξ, we get

$$f(y_k) = f'(\xi)\left[C_2e_k^2 - 2\left(C_2^2 - C_3\right)e_k^3 + \left(5C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 + \mathcal{O}\left(e_k^5\right)\right]. \tag{3.7}$$

With (3.3), (3.6) and (3.7), we calculate the divided difference operator defined in (3.2), obtaining

$$f[x_k, y_k] = f'(\xi)\left[1 + C_2e_k + \left(C_2^2 + C_3\right)e_k^2 - \left(2C_2^3 - 3C_2C_3 - C_4\right)e_k^3 + \mathcal{O}\left(e_k^4\right)\right]. \tag{3.8}$$

Then, substituting (3.3), (3.4), (3.6) and (3.8) in the second step of family (3.1), we have

$$z_k = \xi + \left(C_2^3 - C_2C_3\right)e_k^4 + \mathcal{O}\left(e_k^5\right). \tag{3.9}$$

Using Taylor series once again, similar to (3.3), to expand $f(z_k)$ around ξ, we get

$$f(z_k) = f'(\xi)\left[\left(C_2^3 - C_2C_3\right)e_k^4 + \mathcal{O}\left(e_k^5\right)\right]. \tag{3.10}$$

Replacing (3.4) and (3.8) in u_k and v_k of family (3.1), we have

$$u_k = C_2e_k - \left(3C_2^2 - 2C_3\right)e_k^2 + \left(8C_2^3 - 10C_2C_3 + 3C_4\right)e_k^3 + \mathcal{O}\left(e_k^4\right), \tag{3.11}$$

$$v_k = 1 + C_2e_k - 2\left(C_2^2 - C_3\right)e_k^2 + 3\left(C_2^3 - 2C_2C_3 + C_4\right)e_k^3 + \mathcal{O}\left(e_k^4\right). \tag{3.12}$$

Finally, substituting (3.4), (3.9), (3.10), (3.11) and (3.12) in the third step of family (3.1), we get

$$x_{k+1} = \xi + (1-\alpha-\gamma)C_2\left(C_2^2 - C_3\right)e_k^4 + \mathcal{O}\left(e_k^5\right), \tag{3.13}$$

being the error equation

$$e_{k+1} = (1-\alpha-\gamma)C_2\left(C_2^2 - C_3\right)e_k^4 + \mathcal{O}\left(e_k^5\right), \tag{3.14}$$

and the proof is finished. □

From Theorem 3.1, it follows that the new triparametric family of iterative methods has an order of convergence of four for any real or complex values of the parameters α, β and γ. However, convergence can be speed-up if only one parameter is held and the family is reduced to an uniparametric iterative scheme. The latter can be seen in Theorem 3.2.

Theorem 3.2. *Let $f : I \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function on an open interval I and $\xi \in I$ a simple root of the nonlinear equation $f(x) = 0$. Suppose that $f(x)$ is continuous and sufficiently differentiable in a neighborhood of the simple root 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* (3.1) *converges to ξ with an order of convergence of six, provided that $\beta = 1+\alpha$ and $\gamma = 1-\alpha$, being its error equation*

$$e_{k+1} = \left(6C_2^5 - 7C_2^3C_3 + C_2C_3^2\right)e_k^6 + \mathcal{O}\left(e_k^7\right),$$

where $e_k = x_k - \xi$, $C_q = \dfrac{1}{q!}\dfrac{f^{(q)}(\xi)}{f'(\xi)}$, $q = 2, 3, ...$

Proof. Let ξ be a simple root of $f(x)$ (that is, $f(\xi)=0$ and $f'(\xi)\neq 0$) and $x_k=\xi+e_k$. Using Taylor expansion of $f(x_k)$ and $f'(x_k)$ around ξ, we have

$$\begin{aligned} f(x_k) &= f(\xi+e_k)\\ &= f(\xi)+f'(\xi)e_k+\frac{1}{2!}f''(\xi)e_k^2+\cdots+\frac{1}{6!}f^{(vi)}(\xi)e_k^6+\mathcal{O}(e_k^7)\\ &= f'(\xi)\left[e_k+\frac{1}{2!}\frac{f''(\xi)}{f'(\xi)}e_k^2+\cdots+\frac{1}{6!}\frac{f^{(vi)}(\xi)}{f'(\xi)}e_k^6+\mathcal{O}(e_k^7)\right]\\ &= f'(\xi)\left[e_k+C_2e_k^2+C_3e_k^3+C_4e_k^4+C_5e_k^5+C_6e_k^6+\mathcal{O}(e_k^7)\right], \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} f'(x_k) &= f'(\xi+e_k)\\ &= f'(\xi)+f''(\xi)e_k+\frac{1}{2!}f'''(\xi)e_k^2+\cdots+\frac{1}{5!}f^{(vi)}(\xi)e_k^5+\mathcal{O}(e_k^6)\\ &= f'(\xi)\left[1+\frac{f''(\xi)}{f'(\xi)}e_k+\frac{1}{2!}\frac{f'''(\xi)}{f'(\xi)}e_k^2+\cdots+\frac{1}{5!}\frac{f^{(vi)}(\xi)}{f'(\xi)}e_k^5+\mathcal{O}(e_k^6)\right]\\ &= f'(\xi)\left[1+2C_2e_k+3C_3e_k^2+4C_4e_k^3+5C_5e_k^4+6C_6e_k^5+\mathcal{O}(e_k^6)\right], \end{aligned} \tag{3.16}$$

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

Dividing (3.15) by (3.16), we get

$$\begin{aligned} \frac{f(x_k)}{f'(x_k)} = {}& e_k-C_2e_k^2+2\left(C_2^2-C_3\right)e_k^3-\left(4C_2^3-7C_2C_3+3C_4\right)e_k^4+\\ &\left(8C_2^4-20C_2^2C_3+6C_3^2+10C_2C_4-4C_5\right)e_k^5-\left(16C_2^5-52C_2^3C_3+\right.\\ &\left.28C_2^2C_4-17C_3C_4+C_2\left(33C_3^2-13C_5\right)+5C_6\right)e_k^6+\mathcal{O}\left(e_k^7\right). \end{aligned} \tag{3.17}$$

Replacing (3.17) in the first step of family (3.1), we have

$$\begin{aligned} y_k = {}& \xi+C_2e_k^2-2\left(C_2^2-C_3\right)e_k^3+\left(4C_2^3-7C_2C_3+3C_4\right)e_k^4-\\ &\left(8C_2^4-20C_2^2C_3+6C_3^2+10C_2C_4-4C_5\right)e_k^5+\left(16C_2^5-52C_2^3C_3+\right.\\ &\left.28C_2^2C_4-17C_3C_4+C_2\left(33C_3^2-13C_5\right)+5C_6\right)e_k^6+\mathcal{O}\left(e_k^7\right). \end{aligned} \tag{3.18}$$

Using Taylor expansion again, similar to (3.15), to expand $f(y_k)$ around ξ, we get

$$\begin{aligned} f(y_k) = f'(\xi)\Big[&C_2 e_k^2 - 2\left(C_2^2 - C_3\right)e_k^3 + \left(5C_2^3 - 7C_2C_3 + 3C_4\right)e_k^4 - \\ &2\left(6C_2^4 - 12C_2^2C_3 + 3C_3^2 + 5C_2C_4 - 2C_5\right)e_k^5 + \Big(28C_2^5 - 73C_2^3C_3 + \\ &34C_2^2C_4 - 17C_3C_4 + C_2\left(37C_3^2 - 13C_5\right) + 5C_6\Big)e_k^6 + \mathcal{O}\left(e_k^7\right)\Big]. \end{aligned} \tag{3.19}$$

With (3.15), (3.18) and (3.19), we calculate the divided difference operator defined in (3.2), obtaining

$$\begin{aligned} f[x_k, y_k] = f'(\xi)\Big[&1 + C_2 e_k + \left(C_2^2 + C_3\right)e_k^2 - \left(2C_2^3 - 3C_2C_3 - C_4\right)e_k^3 + \\ &\left(4C_2^4 - 8C_2^2C_3 + 2C_3^2 + 4C_2C_4 + C_5\right)e_k^4 + \Big(-8C_2^5 + 20C_2^3C_3 - \\ &11C_2^2C_4 + 5C_3C_4 + C_2\left(-9C_3^2 + 5C_5\right) + C_6\Big)e_k^5 + \mathcal{O}\left(e_k^6\right)\Big]. \end{aligned} \tag{3.20}$$

Then, substituting (3.15), (3.16), (3.18) and (3.20) in the second step of family (3.1), we have

$$\begin{aligned} z_k = \xi + &\left(C_2^3 - C_2C_3\right)e_k^4 - 2\left(2C_2^4 - 4C_2^2C_3 + C_3^2 + C_2C_4\right)e_k^5 + \\ &\left(10C_2^5 - 30C_2^3C_3 + 12C_2^2C_4 - 7C_3C_4 + 3C_2\left(6C_3^2 - C_5\right)\right)e_k^6 + \mathcal{O}\left(e_k^7\right). \end{aligned} \tag{3.21}$$

Using Taylor series once again, similar to (3.15), to expand $f(z_k)$ around ξ, we get

$$\begin{aligned} f(z_k) = f'(\xi)\Big[&\left(C_2^3 - C_2C_3\right)e_k^4 - 2\left(2C_2^4 - 4C_2^2C_3 + C_3^2 + C_2C_4\right)e_k^5 + \\ &\left(10C_2^5 - 30C_2^3C_3 + 12C_2^2C_4 - 7C_3C_4 + 3C_2\left(6C_3^2 - C_5\right)\right)e_k^6 + \mathcal{O}\left(e_k^7\right)\Big]. \end{aligned} \tag{3.22}$$

Replacing (3.16) and (3.20) in u_k and v_k of family (3.1), we have

$$\begin{aligned} u_k = C_2 e_k - &\left(3C_2^2 - 2C_3\right)e_k^2 + \left(8C_2^3 - 10C_2C_3 + 3C_4\right)e_k^3 + \Big(-20C_2^4 + 37C_2^2C_3 - \\ &8C_3^2 - 14C_2C_4 + 4C_5\Big)e_k^4 + \Big(48C_2^5 - 118C_2^3C_3 + 51C_2^2C_4 - 22C_3C_4 + \\ &C_2\left(55C_3^2 - 18C_5\right) + 5C_6\Big)e_k^5 + \mathcal{O}\left(e_k^6\right), \end{aligned} \tag{3.23}$$

$$\begin{aligned} v_k = {} & 1 + C_2 e_k - 2\left(C_2^2 - C_3\right)e_k^2 + 3\left(C_2^3 - 2C_2C_3 + C_4\right)e_k^3 + \left(-3C_2^4 + 11C_2^2C_3 - \right. \\ & \left. 4C_3^2 - 8C_2C_4 + 4C_5\right)e_k^4 + \left(-10C_2^3C_3 + 14C_2^2C_4 + C_2\left(11C_3^2 - 10C_5\right) + \right. \\ & \left. 5\left(-2C_3C_4 + C_6\right)\right)e_k^5 + \mathcal{O}\left(e_k^6\right). \end{aligned} \tag{3.24}$$

Finally, substituting (3.16), (3.21), (3.22), (3.23) and (3.24) in the third step of family (3.1), we get

$$\begin{aligned} x_{k+1} = {} & \xi + (1-\alpha-\gamma)C_2\left(C_2^2 - C_3\right)e_k^4 + \left((-4+6\alpha-\beta+5\gamma)C_2^4 + \right. \\ & \left. (8-10\alpha+\beta-9\gamma)C_2^2C_3 - 2(1-\alpha-\gamma)C_3^2 - 2(1-\alpha-\gamma)C_2C_4\right)e_k^5 + \\ & \left((10-22\alpha+9\beta-14\gamma)C_2^5 - (30-53\alpha+15\beta-39\gamma)C_2^3C_3 + \right. \\ & 2(6-8\alpha+\beta-7\gamma)C_2^2C_4 - 7(1-\alpha-\gamma)C_3C_4 + \\ & \left. C_2\left((18-25\alpha+4\beta-21\gamma)C_3^2 - 3(1-\alpha-\gamma)C_5\right)\right)e_k^6 + \mathcal{O}\left(e_k^7\right), \end{aligned} \tag{3.25}$$

being the error equation

$$\begin{aligned} e_{k+1} = {} & (1-\alpha-\gamma)C_2\left(C_2^2 - C_3\right)e_k^4 + \left((-4+6\alpha-\beta+5\gamma)C_2^4 + \right. \\ & \left. (8-10\alpha+\beta-9\gamma)C_2^2C_3 - 2(1-\alpha-\gamma)C_3^2 - 2(1-\alpha-\gamma)C_2C_4\right)e_k^5 + \\ & \left((10-22\alpha+9\beta-14\gamma)C_2^5 - (30-53\alpha+15\beta-39\gamma)C_2^3C_3 + \right. \\ & 2(6-8\alpha+\beta-7\gamma)C_2^2C_4 - 7(1-\alpha-\gamma)C_3C_4 + \\ & \left. C_2\left((18-25\alpha+4\beta-21\gamma)C_3^2 - 3(1-\alpha-\gamma)C_5\right)\right)e_k^6 + \mathcal{O}\left(e_k^7\right). \end{aligned} \tag{3.26}$$

To cancel the factors accompanying e_k^4 and e_k^5 in (3.26), it must be satisfied that $\alpha + \gamma = 1$, $6\alpha - \beta + 5\gamma = 4$ and $10\alpha - \beta + 9\gamma = 8$. It is easy to show that this system of equations has infinite solutions for

$$\beta = 1 + \alpha \quad \text{and} \quad \gamma = 1 - \alpha, \tag{3.27}$$

where α is a free parameter. Therefore, replacing (3.27) in (3.26), we obtain

$$e_{k+1} = \left(6C_2^5 - 7C_2^3C_3 + C_2C_3^2\right)e_k^6 + \mathcal{O}\left(e_k^7\right), \tag{3.28}$$

and the proof is finished. □

From Theorem 3.2, it follows that, if we only hold parameter α in (3.1), the new triparametric family of iterative methods is reduced to an uniparametric family with an order of convergence of six for any real or complex values of the parameters α, β and γ, as long as (3.27) is satisfied. Therefore, the iterative expression of the new uniparametric family, dependent only on parameter α and which we will call CMT(α) family, is defined as

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \dfrac{f(y_k)}{2f[x_k, y_k] - f'(x_k)}, \\ x_{k+1} = z_k - (\alpha + (1+\alpha)u_k + (1-\alpha)v_k)\,\dfrac{f(z_k)}{f'(x_k)}, \end{cases} \tag{3.29}$$

where $u_k = 1 - \dfrac{f[x_k, y_k]}{f'(x_k)}$, $v_k = \dfrac{f'(x_k)}{f[x_k, y_k]}$ and $k = 0, 1, 2, \ldots$

Because of the results obtained with the convergence analysis carried out, from now on we will only work with CMT(α) family of iterative methods and, to select the best members of this family, we will use the complex dynamical tools discussed in Section 3.3.

3.3 Complex Dynamical Behavior

This topic refers to the study of the behavior of a rational function associated with an iterative family or method. From the numerical point of view, the dynamical properties of the referred rational function give us important information about its stability and reliability. The parameter spaces of a family of methods, built from the critical points, allow us to understand the performance of the different members of the family, helping us in the election of a particular one. The dynamical planes show the behavior of these particular methods in terms of the basins of attraction of their fixed points, periodic points, etc. A basin of attraction provides us to visually interpret how a method works based on several initial estimates.

In this section, we present the study of the complex dynamics of CMT(α) family given in (3.29). To do this, we construct a rational operator associated with the family, on a generic low-degree nonlinear polynomial, and we analyze the stability and convergence of the corresponding fixed and critical points. Then, we construct the parameter spaces of the free critical points and generate dynamical planes of some methods of the family for good and bad values of α, in terms of stability.

3.3.1 Rational Operator

The rational operator can be built on any nonlinear function; however, we construct this operator on quadratic polynomials, since the criterion of stability or instability of a method applied to these polynomials can be generalized for other nonlinear functions.

Proposition 3.1. *Let $p(x)=(x-a)(x-b)$ be a generic quadratic polynomial with roots $a,b\in\mathbb{R}$. So, the rational operator $R_\alpha(x)$ associated with CMT(α) family given in* (3.29) *and applied on $p(x)$, is*

$$R_\alpha(x)=\frac{x^6\left(x^6+5x^5+12x^4+19x^3+21x^2+14x+\alpha+5\right)}{(\alpha+5)x^6+14x^5+21x^4+19x^3+12x^2+5x+1}, \tag{3.30}$$

with $\alpha\in\mathbb{C}$ an arbitrary parameter. Also, if $\alpha\in\{-77,-1,1,5\}$, $R_\alpha(x)$ is simplified as shown

$$R_{-77}(x)=-\frac{x^6\left(x^5+6x^4+18x^3+37x^2+58x+72\right)}{72x^5+58x^4+37x^3+18x^2+6x+1}, \tag{3.31}$$

$$R_{-1}(x)=\frac{x^6\left(x^4+3x^3+5x^2+6x+4\right)}{4x^4+6x^3+5x^2+3x+1}, \tag{3.32}$$

$$R_{1}(x)=\frac{x^6\left(x^4+4x^3+7x^2+8x+6\right)}{6x^4+8x^3+7x^2+4x+1}, \tag{3.33}$$

$$R_{5}(x)=\frac{x^6\left(x^4+5x^3+11x^2+14x+10\right)}{10x^4+14x^3+11x^2+5x+1}. \tag{3.34}$$

Proof. Let $p(x)=(x-a)(x-b)$ be a generic quadratic polynomial with roots $a,b\in\mathbb{R}$. We apply the iterative scheme given in (3.29) on $p(x)$ and obtain a rational function $A_{p,\alpha}(x)$ which depends on the roots $a,b\in\mathbb{R}$ and a parameter $\alpha\in\mathbb{C}$. Then, if we use Möbius transformation (see [51, 40, 52]) in $A_{p,\alpha}(x)$ with

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

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

$$R_\alpha(x)=\left(h\circ A_{p,\alpha}\circ h^{-1}\right)(x)=\frac{x^6\left(x^6+5x^5+12x^4+19x^3+21x^2+14x+\alpha+5\right)}{(\alpha+5)x^6+14x^5+21x^4+19x^3+12x^2+5x+1}, \tag{3.35}$$

which only depends on an arbitrary parameter $\alpha\in\mathbb{C}$. Also, if we factor numerator and denominator of (3.35), it is easy to show that for $\alpha\in\{-77,-1,1,5\}$ some roots coincide and simplify $R_\alpha(x)$, as it is observed in Equations (3.31) to (3.34), and the proof is finished. □

From Proposition 3.1, for four values of α the rational operator $R_\alpha(x)$ is simpler, so there will be fewer fixed and critical points that can improve the stability of the associated methods. This will be seen in Sections 3.3.2 and 3.3.3.

3.3.2 Analysis and Stability of Fixed Points

We calculate the fixed points of the rational operator $R_\alpha(x)$ given in (3.30), and analyze their stability.

Proposition 3.2. *The fixed points of $R_\alpha(x)$ are the roots of the equation $R_\alpha(x) = x$. That is, $x = 0$, $x = \infty$ and the following strange fixed points:*

- $ex_1 = 1$ *(if $\alpha \neq -77$), and*
- $ex_i(\alpha)$ *that correspond to the 10 roots of the polynomial* $x^{10} + 6x^9 + 18x^8 + 37x^7 + 58x^6 - (\alpha - 67)x^5 + 58x^4 + 37x^3 + 18x^2 + 6x + 1$, *where* $i = 2, ..., 11$.

The total number of different fixed points varies with the value of α, as shown

- *If $\alpha \in \mathbb{C}$ and $\alpha \notin \{-77, -1, 1, 5, 307\}$, then $R_\alpha(x)$ has 13 fixed points.*
- *If $\alpha = -77$, then $ex_1 = 1$ is not a fixed point and $R_\alpha(x)$ has 12 fixed points.*
- *If $\alpha \in \{-1, 1, 5\}$, then $R_\alpha(x)$ has 11 fixed points.*
- *If $\alpha = 307$, then $ex_1 = ex_2 = ex_3 = 1$ and $R_\alpha(x)$ has 11 fixed points.*

The pairs of conjugated strange fixed points, satisfying $ex_i = \dfrac{1}{ex_j}$ for $i \neq j$, are ex_2 and ex_3, ex_4 and ex_5, ex_6 and ex_9, ex_7 and ex_8, ex_{10} and ex_{11}.

From Proposition 3.2, we establish there are a minimum of 11 and a maximum of 13 fixed points. Of these, 0 and ∞ correspond to the roots of the original quadratic polynomial $p(x)$; and the strange fixed point $ex_1 = 1$ (if $\alpha \neq -77$) corresponds to the divergence of the original method, before Möbius transformation.

Proposition 3.3. *The stability of the strange fixed point $ex_1 = 1$, $\forall \alpha \in \mathbb{C} \setminus \{-77\}$, verifies:*

i) If $\left|\dfrac{384}{77+\alpha}\right| < 1$, *then ex_1 is an attractor.*

ii) If $\left|\dfrac{384}{77+\alpha}\right| > 1$, *then ex_1 is a repulsor.*

iii) If $\left|\dfrac{384}{77+\alpha}\right| = 1$, *then ex_1 is parabolic.*

ex_1 is never a superattractor because $\left|\frac{384}{77+\alpha}\right| \neq 0$. The superattracting fixed points that satisfy $|R'_\alpha(x)| = 0$ are $x = 0$, $x = \infty$ and the following strange fixed points:

- *ex_4, ex_5 for $\alpha = -0.949874 \pm 0.16946i$,*
- *ex_6, ex_9 for $\alpha = 2.40285 \pm 1.11088i$, and*
- *ex_{10}, ex_{11} for $\alpha = 178.653$.*

The repulsive fixed points, which always satisfy $|R'_\alpha(x)| > 1$, are the strange fixed points ex_2 and ex_3.

It is clear that 0 and ∞ are always superattracting fixed points, but the stability of the rest of fixed points depends on the values of the parameter α. From Proposition 3.3, there are 6 strange fixed points that can become superattractors for certain values of α. This means that there would be a basin of attraction of the strange fixed point, and it could cause the method not to converge to the solution.

Figure 3.1 shows the stability surface of the strange fixed point ex_1. In this figure, the zones of attraction (yellow surface) and repulsion (grey surface) are observed, being the first one much greater than the second one. Note that for values of α inside disk, ex_1 is a repulsor; and, for off-disk values of α, ex_1 is an attractor. So, it is in our interest to always work inside the disk because the strange fixed point $ex_1 = 1$ comes from the divergence of the original method and, therefore, it is better for the performance of the iterative method that the divergence is repulsive.

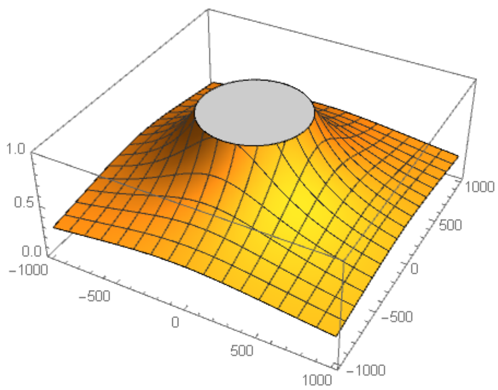
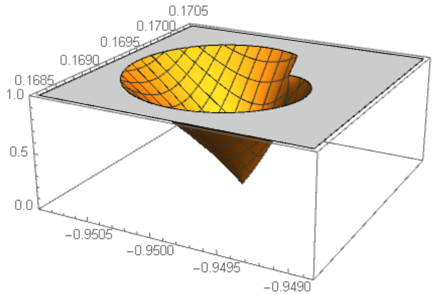


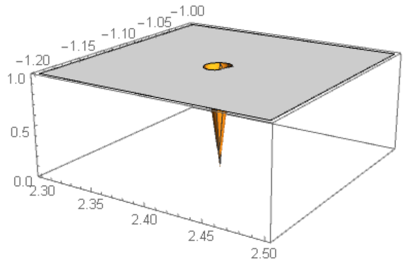
Figure 3.1: Stability surface of $ex_1 = 1$ (in grey color, the complex area where the fixed point is repulsive, being attracting in the rest)

From Proposition 3.2, the study of the stability of strange fixed points is reduced by a half. This is because each pair of conjugated strange fixed points exhibits the same stability characteristics.

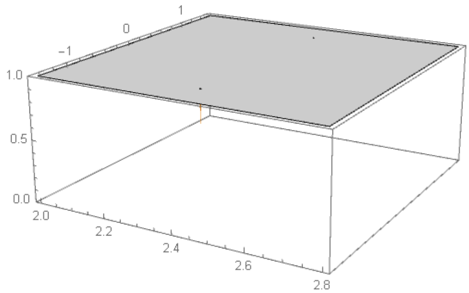
Also, due to Proposition 3.3, ex_2 and ex_3 are always repulsors regardless of the value of α. Thus, Figure 3.2 shows the stability surfaces of the remaining 8 strange fixed points, which can be attracting or repulsive depending on the value of α, for analysis.



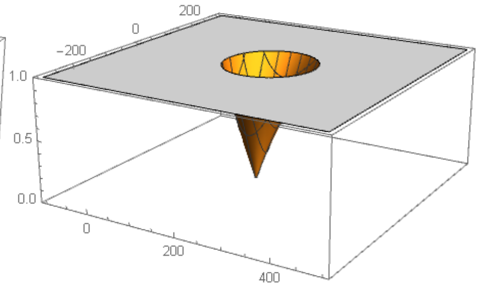
(a) ex_4 and ex_5



(b) ex_6 and ex_9



(c) ex_7 and ex_8



(d) ex_{10} and ex_{11}

Figure 3.2: Stability surfaces of 8 strange fixed points (in grey color, the complex area where each fixed point is repulsive, being attracting in the rest)

3.3.3 Analysis of Critical Points

We calculate the critical points of the rational operator $R_\alpha(x)$ given in (3.30).

Proposition 3.4. *The critical points of $R_\alpha(x)$ are the roots of the equation $R'_\alpha(x) = 0$. That is, $x = 0$, $x = \infty$ and the following free critical points:*

- $cr_1 = -1$,
- $cr_2 = -i$,
- $cr_3 = i$, *and*

- $cr_i(\alpha)$ *that correspond to the 6 roots of polynomial* $(6\alpha+30)x^6+(\alpha+103)x^5+(2\alpha+206)x^4+(-6\alpha+246)x^3+(2\alpha+206)x^2+(\alpha+103)x+6\alpha+30$, *where* $i=4,...,9$.

The total number of different critical points varies with the value of α, *as shown*

- *If* $\alpha \in \mathbb{C}$ *and* $\alpha \notin \{-77,-5,-1,1,5\}$, *then* $R_\alpha(x)$ *has 11 critical points.*
- *If* $\alpha \in \{-77,-5,-1\}$, *then* $R_\alpha(x)$ *is simplified or reduced and has 9 critical points.*
- *If* $\alpha \in \{1,5\}$, *then* $R_\alpha(x)$ *is simplified and has 7 critical points.*

The pairs of conjugated free critical points, satisfying $cr_i = \dfrac{1}{cr_j}$ *for* $i \neq j$, *are* cr_2 *and* cr_3, cr_4 *and* cr_5, cr_6 *and* cr_7, cr_8 *and* cr_9.

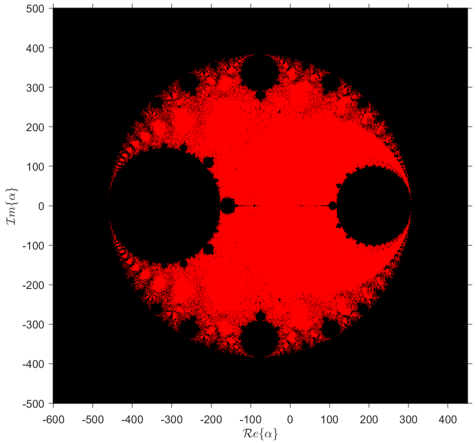
From Proposition 3.4, we establish there are a minimum of 7 and a maximum of 11 critical points. Of these, 0 and ∞ correspond to the roots of the original quadratic polynomial $p(x)$. The free critical points $cr_1=-1$, $cr_2=-i$ and $cr_3=i$ are preimages of the strange fixed point $ex_1=1$. Therefore, the stability of cr_1, cr_2 and cr_3 will correspond to the stability of ex_1 (see Section 3.3.2). Also, the dynamical study of the free critical points is reduced by a half because each pair of conjugated free critical points presents the same stability characteristics. This will be seen in Section 3.3.4.

3.3.4 Parameter Spaces

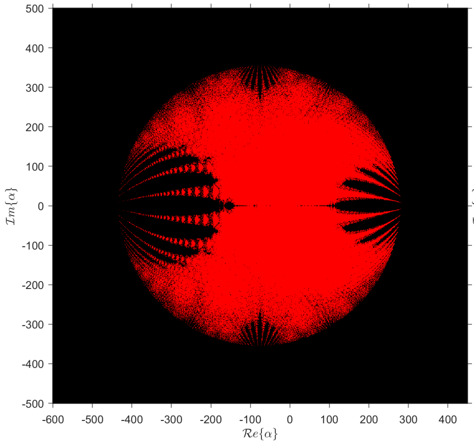
The dynamical behavior of the operator $R_\alpha(x)$ depends on the values of parameter α. The parameter space is defined as a mesh in the complex plane, where each point of this mesh corresponds to a different value of α. Its graphical representation shows the convergence analysis of a method of CMT(α) family associated with this α using one of the free critical points $cr(\alpha)$ given in Proposition 3.4 as initial estimate. The resulting graphic is made in Matlab R2020a programming package with a resolution of 1000x1000 pixels. If a method converges to any of the roots starting from $cr(\alpha)$ in a maximum of 80 iterations with a tolerance of 10^{-3}, the pixel is colored red; in other cases, the pixel is colored black.

Each value of α that belongs to the same connected component of the parameter space results in subsets of schemas with similar dynamical behavior. Therefore, it is interesting to find regions of the parameter space as stable as possible (red regions), because these values of α will give us the best members of the family in terms of numerical stability.

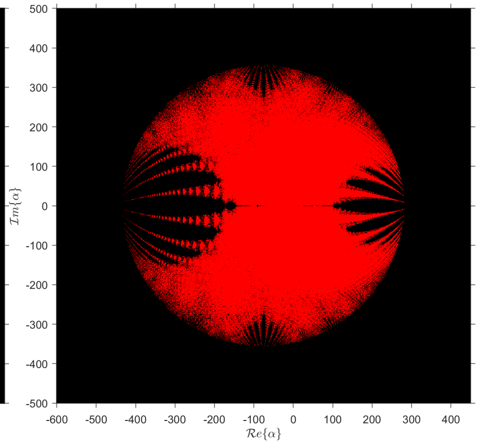
CMT(α) family has a maximum of 9 free critical points. Of these, cr_1, cr_2 and cr_3 have the same parameter space which corresponds to the stability surface of ex_1 (see Figure 3.1), because they are preimages of this point. The remaining free critical points, cr_4 to cr_9, are conjugated in pairs (see Proposition 3.4), which gives rise to 3 different parameter spaces. These parameter spaces named P_1 (for $x=cr_4,cr_5$), P_2 (for $x=cr_6,cr_7$) and P_3 (for $x=cr_8,cr_9$) are shown in Figure 3.3.



(a) P_1 for $x = cr_4, cr_5$



(b) P_2 for $x = cr_6, cr_7$



(c) P_3 for $x = cr_8, cr_9$

Figure 3.3: Parameter spaces of free critical points (in red color, the complex area where the corresponding critical point converges to 0 or ∞, that is, the stability region)

From 3.3(b) and 3.3(c), we observe that the parameter spaces P_2 and P_3 have similar characteristics; then, we can select any of them for analysis.

On the one hand, if we choose values of α inside the stability regions (red regions) of the parameter spaces, for example $\alpha = -1, 0, 1$, the methods associated with these parameters will show good dynamical behavior in terms of numerical stability. Also, note that these particular values of α simplify the iterative scheme of CMT(α) family given in (3.29) by canceling a term in its third step. This is especially useful to improve the computational efficiency of the associated method because the processing times required to reach the solution are reduced (see Section 3.4).

On the other hand, if we choose values of α outside the stability regions (black regions) of the parameter spaces, for example $\alpha = -300, 200, 400$, the methods associated with these parameters will show poor dynamical behavior in terms of numerical stability.

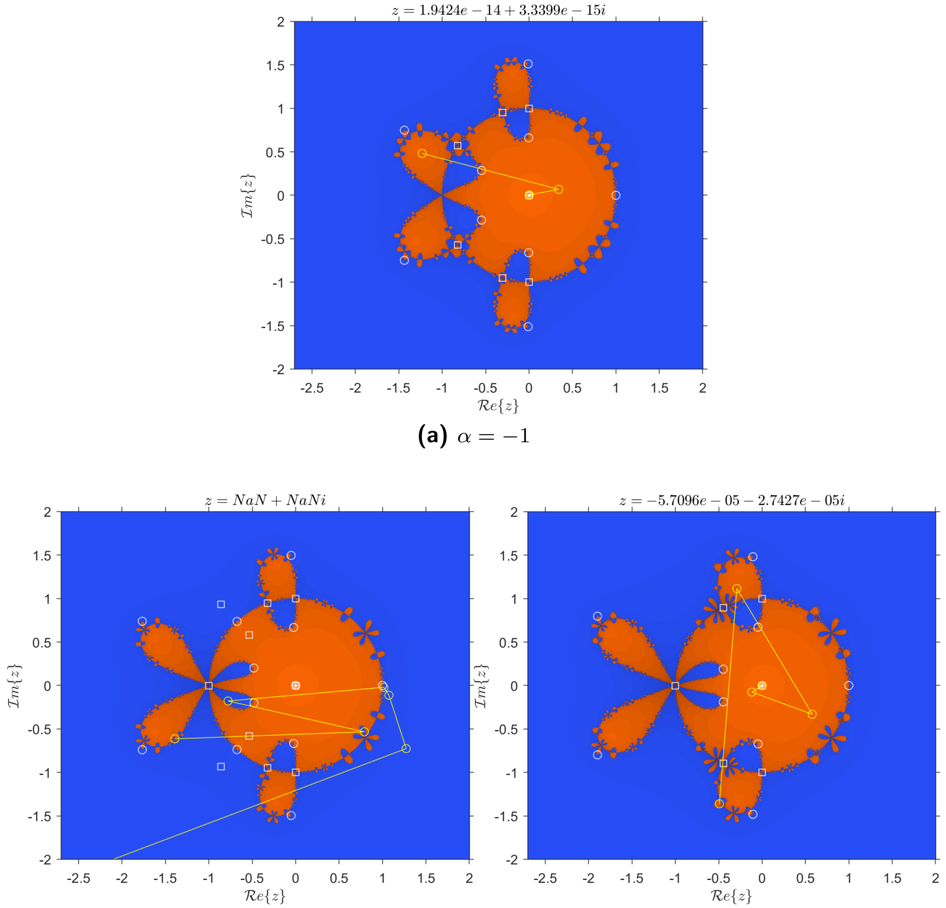
The methods associated with the values of α treated above are discussed in Section 3.3.5.

3.3.5 Dynamical Planes

We begin this section by presenting how we generate a dynamical plane that will allow us to see the stability of a method for a specific value of α. This is defined as a mesh in the complex plane where each point of this mesh corresponds to a different value of the initial estimate x_0. Its graphical representation shows the convergence of the method to any of the roots starting from x_0 with a maximum of 50 iterations and a tolerance of 10^{-3}. Fixed points are illustrated with a white circle '○', critical points with a white square '□' and attractors with a white asterisk '$*$'. Also, the basins of attraction are depicted in different colors. The resulting graphic is made in Matlab R2020a with a resolution of 1000x1000 pixels.

Here, we study the stability of some CMT(α) family methods through the use of dynamical planes. We will consider the methods proposed in Section 3.3.4 for values of α inside and outside the stability regions of the parameter spaces.

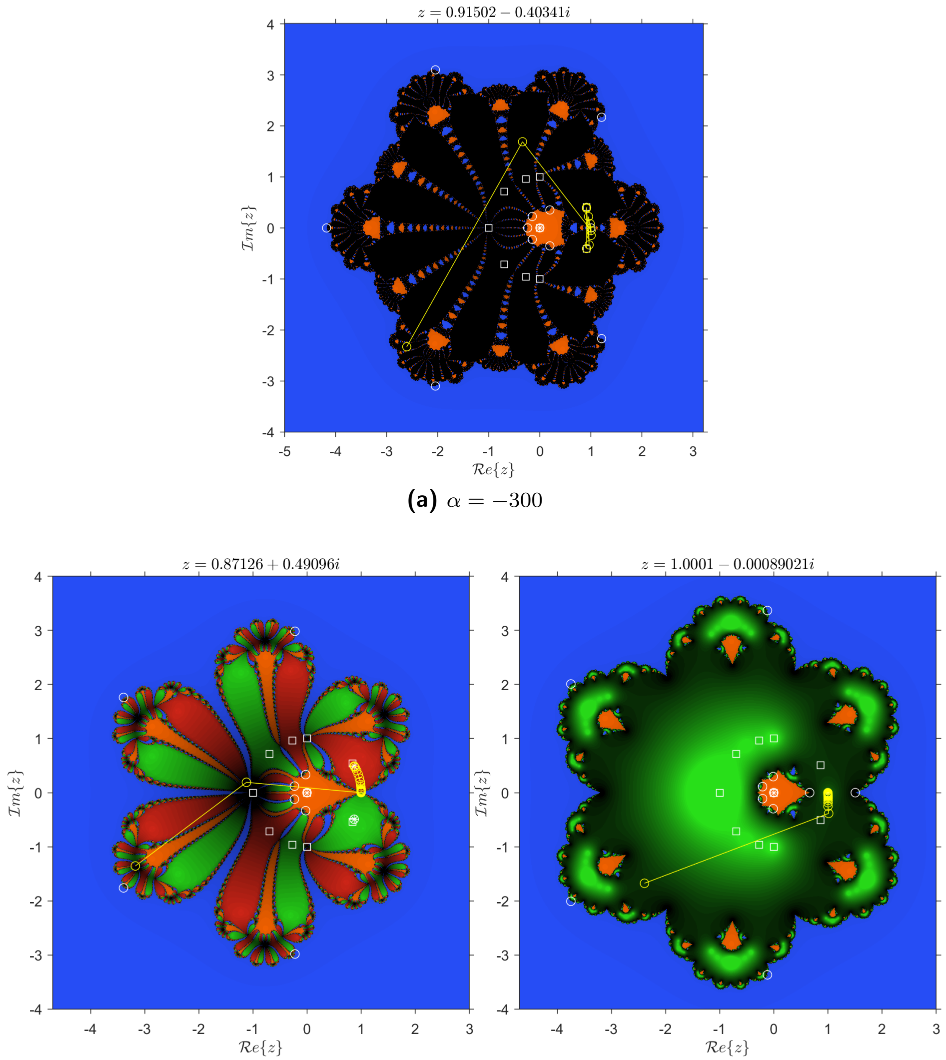
On the one hand, examples of methods inside the stability region are given for $\alpha = -1, 0, 1.$ Their dynamical planes with some convergence orbits in yellow are shown in Figure 3.4. Note that all three methods present only two basins of attraction associated with the roots: the basin of 0 colored in orange and the basin of ∞ colored in blue. Also, there are no black areas of non-convergence to the solution. Consequently, these methods show good dynamical behavior: they are very stable. Of these methods, the best member of CMT(α) family is for $\alpha = 1$, since it has fewer strange fixed points and free critical points.



(b) $\alpha = 0$ (c) $\alpha = 1$

Figure 3.4: Dynamical planes for methods inside the stability region (basin of attraction of 0 in orange color; in blue color, the basin of ∞)

On the other hand, examples of methods outside the stability region are given for $\alpha = -300$, 200, 400. Their dynamical planes with some convergence orbits in yellow are shown in Figure 3.5. Note that all three methods present more than two basins of attraction, that is, there are other basins of attraction that do not correspond to the roots. The basins of 0 and ∞ are colored in orange and blue, respectively; and the other basins are colored in black, red and green. Figure 3.5(a) shows the convergence to an attracting periodic orbit of period 2; Figures 3.5(b) and 3.5(c) show the convergence to an attracting strange fixed point. Also, let us remark that in the three figures the basin of 0 is very small, due to the presence of the other basins of attraction, which reduces the chances of convergence to the solution. Likewise, there are black areas of slow convergence of the methods. Consequently, these methods have poor dynamical behavior: they are unstable.



(a) $\alpha = -300$

(b) $\alpha = 200$ **(c)** $\alpha = 400$

Figure 3.5: Dynamical planes for methods outside the stability region (basin of attraction of 0 in orange color; in blue color the basin of ∞; in green or red color, the basins of attracting strange fixed points)

3.4 Numerical Results

Here, we perform several numerical tests in order to check the theoretical convergence and stability results of CMT(α) family obtained in previous sections. To do this, we use some stable and unstable methods of (3.29). These methods are applied on 5 nonlinear test functions, whose expressions and corresponding roots are:

$$
\begin{aligned}
f_1(x) &= \sin(x) - x^2 + 1, \quad \xi \approx -0.6367326508,\\
f_2(x) &= \cos(x) - x\exp(x) + x^2, \quad \xi \approx 0.6391540963,\\
f_3(x) &= x^3 + 4x^2 - 10, \quad \xi \approx 1.3652300134,\\
f_4(x) &= \sqrt{x^2+2x+5} - 2\sin(x) - x^2 + 3, \quad \xi \approx 2.3319676559,\\
f_5(x) &= \sqrt{x^4} + \sin\left(\frac{\pi}{x^2}\right) - \frac{3}{16}, \quad \xi \approx -0.9059869793.
\end{aligned}
$$

Thus, we performed two experiments. In a first experiment, we carried out an efficiency analysis of CMT(α) family through a comparative study between one of its stable methods and five different methods given in the literature: Newton of order 2, Ostrowski of order 4 and three other methods of order 6 proposed by Alzahrani, Behl, and Alshomrani in [68] (ABA), Chun and Ham in [69] (CH) and Amat, Hernández, and Romero in [70] (AHR). In a second experiment, we carried out a stability analysis of CMT(α) family using six of its methods obtained with three good and three bad values of parameter α, in terms of stability.

In the development of the numerical tests, we start the iterations with different initial estimates: close ($x_0 \approx \xi$), far ($x_0 \approx 10\xi$) and very far ($x_0 \approx 100\xi$) to the root ξ, respectively. This allows us to measure, up to some extent, how demanding the methods are relative to the initial estimation for finding a solution.

The calculations are developed in Matlab R2020a programming package using variable precision arithmetics with 200 digits of mantissa. For each method, we analyze the number of iterations (iter) required to converge to the solution, so that the stopping criteria $|x_{k+1} - x_k| < 10^{-100}$ or $|f(x_{k+1})| < 10^{-100}$ are satisfied. Note that $|x_{k+1} - x_k|$ represents the error estimation between two consecutive iterations and $|f(x_{k+1})|$ is the residual error of the nonlinear test function. This stopping criterium does not need the exact solution, on the contrary of absolute error, and differs from recent ones as CESTAC (see [71]) in the absence of additional calculations or functional evaluations, as $f(x_{k+1})$ is needed for the following iteration and its absolute values is an efficient control element of the proximity to the exact root, where f is zero. Indeed, although a precision of one hundred exact digits is not usually necessary in the applications, we employ this value in the stopping criterium as it is useful to check the robustness and effectiveness of the numerical methods.

To check the theoretical order of convergence (p), we calculate the approximate computational order of convergence (ACOC) given by Cordero and Torregrosa in [33]. In the numerical results presented below, if the ACOC vector inputs do not stabilize their values throughout the iterative process, it is marked as '-'; and, if any of the methods used does not reach convergence in a maximum of 50 iterations, it is marked as 'nc'.

To illustrate the computational efficiency of each used method, the processing time (tcpu) in seconds required by the iterative scheme to converge to the solution is measured. This value is determined as the arithmetic mean of 10 runs of the method.

3.4.1 First Experiment: Efficiency Analysis of CMT(α) Family

In this experiment, we carried out a comparative study between a stable method of CMT(α) family and the methods of Newton, Ostrowski, ABA, CH and AHR, in order to contrast their numerical performances in nonlinear equations. We consider as a stable member of CMT(α) family the method associated with $\alpha = 1$, that is, CMT(1).

Thereby, in Tables 3.1-3.3 we show the numerical results of the six known methods, considering close, far and very far initial estimates. Furthermore, in Figure 3.6 we show graphics that summarize these results for the number of iterations (iter) and the processing time (tcpu).

Table 3.1: Numerical performance of iterative methods in nonlinear equations for x_0 **close to** ξ

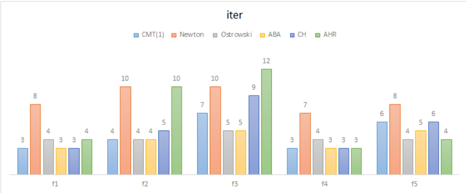
Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC	tcpu
f_1	CMT(1)	7.6395e-19	1.8769e-110	3	5.5148	0.1257
$x_0 =$ -1.6	Newton	3.2063e-84	7.2243e-168	8	2	0.1225
	Ostrowski	3.6277e-39	2.1775e-155	4	3.9988	0.1036
	ABA	6.3941e-19	5.2542e-111	3	5.5472	0.1201
	CH	3.9619e-19	3.095e-112	3	5.5336	0.1173
	AHR	6.9779e-86	0	4	5.9989	0.1381
f_2	CMT(1)	1.1915e-19	3.2336e-114	4	6.0717	0.2913
$x_0 =$ -0.4	Newton	6.977e-101	9.2573e-201	10	2	0.2747
	Ostrowski	3.6009e-28	5.8295e-111	4	3.9993	0.1899
	ABA	6.593e-46	0	4	6.0055	0.4278
	CH	4.0133e-50	6.8135e-208	5	5.9951	0.5636
	AHR	1.4561e-73	0	10	5.9991	0.7038
f_3	CMT(1)	5.868e-64	0	7	5.9957	0.4654
$x_0 =$ 0.4	Newton	3.2665e-83	8.6382e-165	10	2	0.2818
	Ostrowski	1.3665e-51	5.077e-204	5	3.9999	0.1682
	ABA	2.5625e-27	4.3729e-160	5	5.8933	0.224
	CH	2.4971e-24	4.0266e-142	9	5.8498	0.4374
	AHR	2.1589e-36	0	12	5.9521	0.5912
f_4	CMT(1)	1.2572e-32	3.2096e-195	3	5.717	0.6075
$x_0 =$ 1.3	Newton	7.2803e-95	1.2821e-189	7	2	0.4947
	Ostrowski	1.0395e-64	1.5574e-207	4	4	0.535
	ABA	4.7735e-26	8.9685e-156	3	5.9419	0.598
	CH	1.6112e-32	1.7919e-194	3	5.6961	0.6046
	AHR	3.0816e-22	1.1423e-131	3	5.6812	0.4497
f_5	CMT(1)	2.5535e-53	6.4242e-207	6	5.9132	1.2222
$x_0 =$ -1.9	Newton	3.4167e-84	8.1562e-167	8	2	0.6295
	Ostrowski	4.1408e-38	2.4627e-142	4	4.0146	0.5521
	ABA	5.637e-65	1.9467e-208	5	6.0107	0.9561
	CH	1.0828e-43	1.0707e-207	6	6.2212	1.1314
	AHR	5.6988e-26	4.6285e-106	4	5.7855	0.5939

Table 3.2: Numerical performance of iterative methods in nonlinear equations for x_0 **far from** ξ

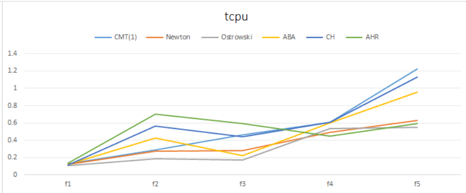
Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC	tcpu
f_1	CMT(1)	4.3721e-23	6.595e-136	4	5.7093	0.163
$x_0 =$ -6	Newton	4.549e-85	1.4542e-169	10	2	0.1527
	Ostrowski	7.5454e-40	4.0753e-158	5	3.9989	0.1487
	ABA	6.5662e-25	6.1621e-147	4	5.775	0.1607
	CH	9.4464e-24	5.6868e-140	4	5.7326	0.1811
	AHR	9.9786e-85	0	5	5.9988	0.1578
f_2	CMT(1)	6.0086e-60	0	16	5.9975	0.9939
$x_0 =$ -6	Newton	2.9103e-57	1.6107e-113	12	2	0.2714
	Ostrowski	1.7318e-82	6.8135e-208	8	4	0.3234
	ABA	2.9737e-18	6.4713e-106	10	-	0.6234
	CH	4.8167e-51	0	14	5.9955	0.8618
	AHR	1.0711e-58	0	6	5.9971	0.3006
f_3	CMT(1)	4.2145e-24	1.1268e-140	10	5.8416	0.4353
$x_0 =$ -14	Newton	nc	nc	nc	nc	nc
	Ostrowski	2.3325e-76	0	37	4	0.9868
	ABA	9.1479e-18	9.0509e-103	24	6.2542	1.1023
	CH	5.0027e-98	0	17	5.9997	0.7088
	AHR	nc	nc	nc	nc	nc
f_4	CMT(1)	4.6353e-98	2.3361e-207	5	5.9995	0.978
$x_0 =$ -23	Newton	9.6577e-79	1.3216e-156	10	2	0.683
	Ostrowski	8.1672e-31	5.8293e-122	5	3.9956	0.6646
	ABA	1.3364e-69	2.3361e-207	5	5.9961	0.9691
	CH	4.543e-99	2.3361e-207	5	5.9996	0.9597
	AHR	1.7793e-56	2.3361e-207	5	5.9898	0.6951
f_5	CMT(1)	3.9117e-41	9.6363e-207	5	6.1766	0.9564
$x_0 =$ -9	Newton	1.2423e-55	1.9722e-109	9	2	0.615
	Ostrowski	2.9225e-29	1.7446e-112	5	3.9821	0.6514
	ABA	2.0254e-31	5.3498e-181	5	5.5153	0.9702
	CH	6.524e-29	3.5709e-159	6	6.2558	1.1451
	AHR	1.6141e-41	9.7687e-148	12	5.8222	1.592

Table 3.3: Numerical performance of iterative methods in nonlinear equations for x_0 **very far from** ξ

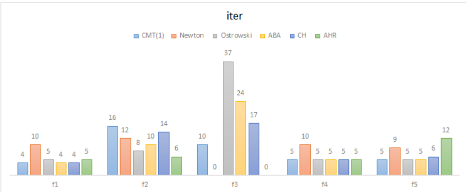
Function	Method	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC	tcpu
f_1	CMT(1)	6.8586e-80	0	6	5.9981	0.2413
$x_0 =$ -60	Newton	3.1826e-73	7.1179e-146	13	2	0.2003
	Ostrowski	1.2267e-100	0	7	4	0.1793
	ABA	1.3417e-77	0	6	5.9978	0.273
	CH	1.4971e-82	0	6	5.9984	0.2776
	AHR	8.7686e-61	3.8934e-208	7	5.992	0.2246
f_2	CMT(1)	5.9893e-27	5.2167e-158	6	6.0379	0.3503
$x_0 =$ -60	Newton	1.6537e-59	5.201e-118	15	2	0.3125
	Ostrowski	8.0088e-72	0	8	4	0.2956
	ABA	2.9305e-56	0	10	6.0024	0.5679
	CH	8.4413e-48	0	7	5.994	0.399
	AHR	6.4484e-60	0	7	5.9974	0.318
f_3	CMT(1)	3.7145e-76	0	13	5.9983	0.6398
$x_0 =$ -140	Newton	nc	nc	nc	nc	nc
	Ostrowski	6.9267e-37	3.3507e-145	49	3.999	1.216
	ABA	7.5885e-54	0	11	5.9907	0.4246
	CH	4.8283e-28	2.1045e-164	21	5.8989	0.8005
	AHR	3.4494e-58	6.2295e-207	12	5.9928	0.3997
f_4	CMT(1)	9.2602e-68	2.3361e-207	6	5.9954	1.0547
$x_0 =$ -230	Newton	8.9492e-96	1.1348e-190	14	2	0.8454
	Ostrowski	7.8874e-37	5.0705e-146	7	3.9985	0.8196
	ABA	2.5587e-21	9.9754e-126	6	6.2382	1.0537
	CH	2.2055e-60	2.3361e-207	6	6.0079	1.0555
	AHR	nc	nc	nc	nc	nc
f_5	CMT(1)	2.8545e-38	1.0707e-207	6	6.2665	1.0249
$x_0 =$ -90	Newton	9.6307e-58	6.4804e-114	12	2	0.7181
	Ostrowski	6.1241e-52	6.9183e-202	8	3.9999	0.9291
	ABA	1.2306e-20	2.1729e-114	6	6.8491	1.0378
	CH	3.4946e-26	1.6995e-147	6	5.7567	1.0301
	AHR	8.5778e-51	1.0901e-182	25	5.902	3.0345



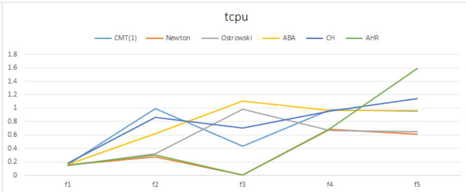
(a) Number of iterations for x_0 close to ξ



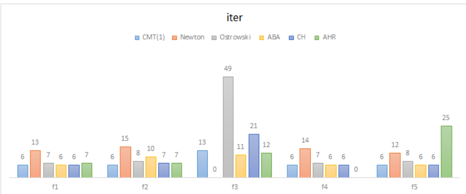
(b) Processing time for x_0 close to ξ



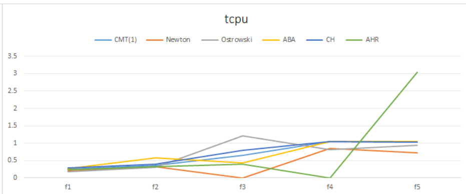
(c) Number of iterations for x_0 far from ξ



(d) Processing time for x_0 far from ξ



(e) Number of iterations for x_0 very far from ξ



(f) Processing time for x_0 very far from ξ

Figure 3.6: Numerical results of the first experiment

Therefore, from the results of the first experiment, we conclude that CMT(α) family has an excellent numerical performance considering a stable member ($\alpha = 1$) as a representative. This conclusion has been made based on the following aspects from Tables 3.1-3.3: CMT(1) method has the lowest error and lowest number of iterations (iter). However, the mean of the execution time (tcpu) varies according to the nonlinear test function used and the inherent complexity that the iterative scheme of the method presents on the nonlinear function. In several cases, the tcpu of the CMT(1) method is significantly lower than the 6th order ABA, CH and AHR methods. The theoretical convergence order is also verified by the ACOC, which is close to 6.

3.4.2 Second Experiment: Stability Analysis of CMT(α) Family

In this experiment, we carried out a stability analysis of CMT(α) family considering some values of α inside the stability regions of the parameter spaces ($\alpha = -1, 0, 1$) and outside of them ($\alpha = -300, 200, 400$).

Thus, in Tables 3.4-3.9 we show the numerical performance of iterative methods associated with these values of α for close, far and very far initial estimations. The results for $\alpha = 1$ were already presented in the first experiment; however, these are presented again due to the different conditions in which each experiment was performed.

Table 3.4: Numerical performance of CMT(-1) method in nonlinear equations

Function	x_0	$\lvert x_{k+1} - x_k\rvert$	$\lvert f(x_{k+1})\rvert$	iter	ACOC	tcpu
	Close to ξ					
f_1	-1.6	1.8646e-19	2.7591e-114	3	5.5559	0.1216
f_2	-0.4	1.3898e-46	0	4	6.0038	0.2775
f_3	0.4	9.0583e-50	0	5	5.9873	0.2321
f_4	1.3	1.9771e-32	7.3778e-194	3	5.6791	0.6628
f_5	-1.9	4.057e-47	1.606e-206	6	6.0586	1.2462
	Far from ξ					
f_1	-6	1.4965e-24	7.3749e-145	4	5.7594	0.1606
f_2	-6	7.3835e-26	1.1396e-151	14	-	0.8807
f_3	-14	1.009e-18	1.3833e-108	22	5.7241	0.9937
f_4	-23	3.2059e-100	2.3361e-207	5	5.9996	1.0545
f_5	-9	4.5305e-85	1.168e-207	7	6.0034	1.446
	Very far from ξ					
f_1	-60	1.264e-85	0	6	5.9988	0.2385
f_2	-60	8.3236e-19	2.3391e-109	9	6.0055	0.5682
f_3	-140	6.8807e-19	1.3913e-109	10	5.7297	0.4723
f_4	-230	1.1069e-48	2.3361e-207	6	6.0195	1.2992
f_5	-90	2.3226e-65	1.168e-207	6	5.9808	1.3969

Table 3.5: Numerical performance of CMT(0) method in nonlinear equations

Function	x_0	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC	tcpu
	Close to ξ					
f_1	-1.6	3.9254e-19	2.9279e-112	3	5.5334	0.1219
f_2	-0.4	1.0637e-28	1.328e-168	4	6.0263	0.2689
f_3	0.4	4.828e-30	2.1036e-176	6	5.9174	0.2482
f_4	1.3	1.6112e-32	1.7919e-194	3	5.6961	0.6771
f_5	-1.9	3.0345e-27	1.8831e-154	6	6.5022	1.2896
	Far from ξ					
f_1	-6	8.7386e-24	3.5638e-140	4	5.7334	0.1602
f_2	-6	6.7903e-26	8.9867e-152	9	-	0.6206
f_3	-14	8.1206e-24	4.7631e-139	11	5.8407	0.491
f_4	-23	4.2612e-99	2.3361e-207	5	5.9996	1.0585
f_5	-9	4.1362e-41	2.3361e-207	6	5.9265	1.2619
	Very far from ξ					
f_1	-60	1.1445e-82	0	6	5.9985	0.2395
f_2	-60	3.277e-54	0	7	5.9966	0.4971
f_3	-140	3.695e-64	0	37	5.9959	1.6934
f_4	-230	5.2233e-59	2.3361e-207	6	6.0088	1.2644
f_5	-90	8.2696e-19	2.984e-103	6	5.5602	1.2865

Table 3.6: Numerical performance of CMT(1) method in nonlinear equations

Function	x_0	$\lvert x_{k+1} - x_k\rvert$	$\lvert f(x_{k+1})\rvert$	iter	ACOC	tcpu
	Close to ξ					
f_1	-1.6	7.6395e-19	1.8769e-110	3	5.5148	0.124
f_2	-0.4	1.1915e-19	3.2336e-114	4	6.0717	0.2474
f_3	0.4	5.868e-64	0	7	5.9957	0.3128
f_4	1.3	1.2572e-32	3.2096e-195	3	5.717	0.7052
f_5	-1.9	2.5535e-53	6.4242e-207	6	5.9132	1.3006
	Far from ξ					
f_1	-6	4.3721e-23	6.595e-136	4	5.7093	0.1619
f_2	-6	6.0086e-60	0	16	5.9975	1.0008
f_3	-14	4.2145e-24	1.1268e-140	10	5.8416	0.446
f_4	-23	4.6353e-98	2.3361e-207	5	5.9995	1.0401
f_5	-9	3.9117e-41	9.6363e-207	5	6.1766	1.0393
	Very far from ξ					
f_1	-60	6.8586e-80	0	6	5.9981	0.2654
f_2	-60	5.9893e-27	5.2167e-158	6	6.0379	0.3777
f_3	-140	3.7145e-76	0	13	5.9983	0.5816
f_4	-230	9.2602e-68	2.3361e-207	6	5.9954	1.2349
f_5	-90	2.8545e-38	1.0707e-207	6	6.2665	1.2801

Table 3.7: Numerical performance of CMT(-300) method in nonlinear equations

Function	x_0	$\lvert x_{k+1} - x_k \rvert$	$\lvert f(x_{k+1}) \rvert$	iter	ACOC	tcpu
	Close to ξ					
f_1	-1.6	1.454e-49	3.8934e-208	4	6.0127	0.1743
f_2	-0.4	7.7e-75	0	40	6.0006	2.5385
f_3	0.4	nc	nc	nc	nc	nc
f_4	1.3	4.1603e-29	3.3384e-172	3	5.3365	0.621
f_5	-1.9	1.6341e-58	2.5794e-206	5	5.7418	1.1787
	Far from ξ					
f_1	-6	nc	nc	nc	nc	nc
f_2	-6	nc	nc	nc	nc	nc
f_3	-14	3.9697e-26	4.0419e-151	7	6.0709	0.328
f_4	-23	nc	nc	nc	nc	nc
f_5	-9	2.8218e-76	6.0348e-207	8	5.9788	1.5886
	Very far from ξ					
f_1	-60	4.4607e-32	3.3463e-187	9	6.0453	0.3717
f_2	-60	nc	nc	nc	nc	nc
f_3	-140	1.7723e-57	0	21	6.0044	0.9822
f_4	-230	1.249e-29	2.4449e-175	39	5.1386	7.4938
f_5	-90	6.6349e-43	4.8668e-209	22	6.0131	4.4952

Table 3.8: Numerical performance of CMT(200) method in nonlinear equations

Function	x_0	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC	tcpu
	Close to ξ					
f_1	-1.6	2.1496e-56	0	4	5.9921	0.1499
f_2	-0.4	nc	nc	nc	nc	nc
f_3	0.4	nc	nc	nc	nc	nc
f_4	1.3	4.0045e-33	1.6998e-196	3	5.3325	0.6325
f_5	-1.9	1.3149e-70	9.6363e-207	4	6.0496	0.8213
	Far from ξ					
f_1	-6	6.1599e-40	3.8934e-208	7	5.9673	0.2711
f_2	-6	nc	nc	nc	nc	nc
f_3	-14	nc	nc	nc	nc	nc
f_4	-23	4.3946e-33	2.9689e-196	7	5.339	1.4742
f_5	-9	8.369e-63	2.9162e-205	11	5.964	2.0915
	Very far from ξ					
f_1	-60	1.9877e-20	1.8239e-118	14	5.7565	0.5598
f_2	-60	nc	nc	nc	nc	nc
f_3	-140	nc	nc	nc	nc	nc
f_4	-230	2.7541e-49	1.5574e-207	15	5.9586	3.1228
f_5	-90	7.8278e-51	9.6363e-207	15	6.1663	3.2771

Table 3.9: Numerical performance of CMT(400) method in nonlinear equations

Function	x_0	$\|x_{k+1} - x_k\|$	$\|f(x_{k+1})\|$	iter	ACOC	tcpu
	Close to ξ					
f_1	-1.6	2.9103e-44	0	4	5.9805	0.1439
f_2	-0.4	nc	nc	nc	nc	nc
f_3	0.4	nc	nc	nc	nc	nc
f_4	1.3	1.139e-35	1.5574e-207	3	5.2494	0.6218
f_5	-1.9	5.8131e-53	3.1147e-207	4	5.754	0.8023
	Far from ξ					
f_1	-6	nc	nc	nc	nc	nc
f_2	-6	nc	nc	nc	nc	nc
f_3	-14	nc	nc	nc	nc	nc
f_4	-23	nc	nc	nc	nc	nc
f_5	-9	nc	nc	nc	nc	nc
	Very far from ξ					
f_1	-60	nc	nc	nc	nc	nc
f_2	-60	nc	nc	nc	nc	nc
f_3	-140	nc	nc	nc	nc	nc
f_4	-230	nc	nc	nc	nc	nc
f_5	-90	nc	nc	nc	nc	nc

On the one hand, from Tables 3.4-3.6 we observe that the methods associated with $\alpha = -1, 0, 1$ always converge to the solution, although the number of iterations (iter) needed differs for any initial estimate and nonlinear test function. Thus, in estimations close to the root, the methods converge to ξ with a minimum iter of 3 and a maximum of 7. When the initial guess is far from the root, they converge to ξ with a minimum iter of 4 and a maximum of 22. And, when the starting estimations are very far from the root, the iterative schemes converge to ξ with a minimum iter of 6 and a maximum of 37.

On the other hand, from the results shown in Tables 3.7-3.9, we see that the methods associated with $\alpha = -300, 200, 400$ do not always converge to the solution, confirming the conclusions obtained in the dynamical analysis. The convergence highly depends on the initial estimation and the nonlinear test function used. Thus, for estimations close to the root, these methods do not converge to the solution in up to 2 test functions. And, for estimations far and very far from the root, they do not converge to the solution even for any function.

Consequently, we conclude that the methods for $\alpha = -1, 0, 1$ are stable, have the lowest processing times (tcpu), and always converge to the solution for any initial estimate and nonlinear test function used. The methods for $\alpha = -300, 200, 400$ are unstable, chaotic, have the highest tcpu, and tend not to converge to the solution according to the initial estimate and the nonlinear test function used. With this, the theoretical results obtained in previous sections about the dynamical behavior of CMT(α) family are verified.

3.5 Conclusions

In this paper, a new family of iterative methods was designed to solve nonlinear equations from Ostrowski scheme, adding a Newton step with a "frozen" derivative and using a divided difference operator. This family named CMT(α, β, γ) has a three-step iterative expression and three arbitrary parameters which can take any real or complex value.

In the convergence analysis of the new family, we obtained an order of convergence of four just like the order of the Ostrowski method. However, we managed to speed-up the convergence to six by setting the parameters β and γ as a function of α, resulting in an uniparametric CMT(α) family.

In the dynamical study we constructed parameters spaces of the free critical points of the rational operator associated with the uniparametric family. These parameter spaces allowed us to understand the performance of the different members of the family, helping us to choose stable (for $\alpha = -1, 0, 1, ...$) and unstable (for $\alpha = -300, 200, 400, ...$) methods. Also, we generated dynamical planes to show the behavior of these particular methods.

From numerical results, the order of convergence is verified by the ACOC, which is close to 6. The CMT(α) family proved to have an excellent numerical performance considering stable members as representatives. In general, this family has low errors and number of iterations to converge to the solution. However, the processing time (tcpu) varies depending on the nonlinear test functions used and the inherent complexity that the iterative schemes of the methods present when they are applied to said functions. In several cases, the tcpu of stable methods is significantly lower than other sixth-order methods developed so far. Also, the methods for $\alpha = -1, 0, 1$ proved to be stable, have the lowest tcpu, and always converge to the solution for any initial estimate and nonlinear test function used. The methods for $\alpha = -300, 200, 400$ proved to be unstable, chaotic, have the highest tcpu, and tend not to converge to the solution according to the initial estimate and the nonlinear test function used. This verifies the theoretical results obtained in convergence analysis and dynamical study of CMT(α) family.

Chapter 4

Performance of a new sixth-order class of iterative schemes for solving non-linear systems

Reference: *Moscoso-Martínez, M.; Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. Performance of a New Sixth-Order Class of Iterative Schemes for Solving Non-Linear Systems of Equations. Mathematics 2023, 11, 1374. https://doi.org/10.3390/math11061374*

Abstract: This manuscript is focused on a new parametric class of multi-step iterative procedures to find the solutions of systems of nonlinear equations. Starting from Ostrowski's scheme, the class is constructed by adding a Newton step whose Jacobian matrix is taken from the previous step and employing a divided difference operator, resulting in a triparametric scheme with order of convergence four. We can accelerate the convergence of the family to 6 by setting two parameters, resulting in a uniparametric family. We perform a dynamic and numerical development to analyze the stability of the sixth-order family. Previous studies for scalar functions allowed us to isolate elements of the family with stable performance for solving practical problems. In this regard, we present dynamical planes showing the complexity of the family. The numerical properties of the class are analyzed with several test problems.

Keywords: Nonlinear systems of equations; multipoint iterative methods; analysis of convergence; real discrete dynamics; chaos and stability.

4.1 Introduction

A large number of problems in Computer Science and related disciplines are mathematically characterized by a nonlinear equation or a nonlinear system of equations $F(x) = 0$, where $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is a sufficiently Frechet differentiable function over an open convex set D. Finding the value of a solution ξ is a problem that has been tackled with multiple strategies in fields such as Numerical Analysis, Applied Mathematics or Engineering.

Newton's scheme is the best known scheme for finding the zero $\xi \in D$ of F,

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}),$$

where $k \geq 0$ and the Jacobian matrix of F at $x^{(k)}$ is denoted by $F'(x^{(k)})$.

In recent years, this problem has attracted the attention of many scientists, highlighting the following techniques. The extension of scalar to vector iterative methods [72, 73, 74, 75] is a common practice – provided the extension is feasible – that affords solutions to n-dimensional problems. To improve the convergence order without compromising the computational cost, new steps are included with only one new evaluation of F, keeping F' frozen [76, 34, 77, 78].

We propose in this manuscript a new parametric class of multi-step iterative procedure (4.1) for solving systems of nonlinear equations. This family is a multidimesnional extension of the set of methods defined in [36], for nonlinear equations. The starting point of this family is Ostrowski's scheme, appending an step of Newton-type with a "frozen" Jacobian matrix. Thus, it has an iterative expression with three arbitrary parameters and three steps,

$$\begin{cases} y^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} = y^{(k)} - [2[x^{(k)}, y^{(k)}; F] - F'(x^{(k)})]^{-1}F(y^{(k)}), \\ x^{(k+1)} = z^{(k)} - (\alpha I + \beta u^{(k)} + \gamma v^{(k)})[F'(x^{(k)})]^{-1}F(z^{(k)}), \end{cases} \tag{4.1}$$

where α, β and γ are arbitrary parameters, $v^{(k)} = [x^{(k)}, y^{(k)}; F]^{-1}F'(x^{(k)})$ and $u^{(k)} = I - [F'(x^{(k)})]^{-1}[x^{(k)}, y^{(k)}; F]$, $k = 0, 1, 2, ...$ The definition of the divided difference operator can be found in [67]: it is the map $[\cdot, \cdot; F] : D \times D \subset \mathbb{R}^n \times \mathbb{R}^n \to \mathcal{L}(\mathbb{R}^n)$ that satisfies

$$[x, y; F](x - y) = F(x) - F(y), \quad \forall x, y \in D. \tag{4.2}$$

Starting from (4.1), an uniparametric family is constructed that reaches order of convergence six, which is corroborated supported by a convergence analysis. The objective of the new family is to increase the convergence order without increasing the computational cost significantly.

The dynamic behavior of the rational operator obtained from iterative schemes applied to low-degree nonlinear polynomial systems is an effective tool to analyze the stability and reliability of these numerical methods [34, 35]. The stability of the family is analyzed using a real multidimensional discrete dynamical system. We construct dynamical planes that show the complexity of this class. It should be noted that the complex analysis presented in [36] for scalar functions is extended to vector functions to choose stable members from parameter spaces. Several numerical tests are performed to illustrate the efficiency and stability of the iterative schemes.

The outline of the manuscript is as follows: we introduce the proposed class of iterative procedures in Section 1; its convergence is analyzed in Section 2, finding that with an appropriate selection of the parameters, a one-parametric family of sixth-order of convergence can be found. Section 3 is devoted to the dynamical analysis of the family, finding those with best and worst performance, in terms of their stability. The numerical performance is checked in Section 4 and some conclusions are stated in Section 5.

4.2 Convergence Analysis of the Family

Now, we analyze the convergence properties of the new triparametric iterative family. Although the order of the triparametric family is four, in the proof we use higher-order Taylor expansions since they will be useful to prove the order of the uniparametric family.

Theorem 4.1 (Tri-parametric class). *Let us consider a sufficiently differentiable function $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ in an convex open set D. Let $\xi \in D$ be a solution of nonlinear system $F(x) = 0$. Also, let us assume that $F'(x)$ is continuous and nonsingular at ξ, and $x^{(0)}$ is an seed close enough to ξ. Then, sequence $\{x^{(k)}\}_{k\geq 0}$ obtained by using expression* (4.1) *converges with order of convergence four to solution ξ. Under this hypothesis, its error equation is*

$$e^{(k+1)} = (1-\alpha-\gamma)\left(C_2^3 - C_3C_2\right)e^{(k)^4} + \mathcal{O}(e^{(k)^5}),$$

being α, β and γ are arbitrary parameters, $C_q = \frac{1}{q!}[F'(\xi)]^{-1}F^{(q)}(\xi)$, $q = 2, 3, \ldots$, and $e^{(k)} = x^{(k)} - \xi$.

Proof. Let us consider ξ such that $F(\xi) = 0$ and $F'(\xi)$ nonsingular. Also, let be $x^{(k)} = \xi + e^{(k)}$. By using Taylor expansion series of $F(x^{(k)})$ and $F'(x^{(k)})$ around ξ, we get

$$F(x^{(k)}) = F'(\xi)\left[e^{(k)} + C_2e^{(k)^2} + C_3e^{(k)^3} + C_4e^{(k)^4}\right] + \mathcal{O}(e^{(k)^5}), \tag{4.3}$$

and

$$F'(x^{(k)}) = F'(\xi)\left[I + 2C_2e^{(k)} + 3C_3e^{(k)^2} + 4C_4e^{(k)^3}\right] + \mathcal{O}(e^{(k)^4}), \tag{4.4}$$

where coefficients C_q are defined as $C_q = \dfrac{1}{q!}[F'(\xi)]^{-1}F^{(q)}(\xi)$, $q = 2, 3, \ldots$

Now, the Taylor expansion of the inverse $[F'(x^{(k)})]^{-1}$ is stated as follows

$$\begin{aligned}[F'(x^{(k)})]^{-1} = {} & \left[I + X_2e^{(k)} + X_3e^{(k)^2} + X_4e^{(k)^3} + X_5e^{(k)^4} + X_6e^{(k)^5}\right]\left[F'(\xi)\right]^{-1} \\ & + \mathcal{O}(e^{(k)^7}),\end{aligned} \tag{4.5}$$

where $X_2, X_3, ..., X_6$ are unknowns such that

$$[F'(x^{(k)})]^{-1}F'(x^{(k)}) = I. \tag{4.6}$$

Then, we get

$$\begin{aligned}
X_2 &= -2C_2,\\
X_3 &= 4C_2^2 - 3C_3,\\
X_4 &= -8C_2^3 + 6C_2C_3 + 6C_3C_2 - 4C_4,\\
X_5 &= 16C_2^4 - 12C_2^2C_3 - 12C_2C_3C_2 + 8C_2C_4 + 9C_3^2 - 12C_3C_2^2 + 8C_4C_2 - 5C_5,\\
X_6 &= -32C_2^5 + 24C_2^3C_3 + 24C_2^2C_3C_2 - 16C_2^2C_4 + 24C_2C_3C_2^2 - 16C_2C_4C_2\\
&\quad - 18C_2C_3^2 + 10C_2C_5 - 18C_3^2C_2 + 24C_3C_2^3 - 18C_3C_2C_3\\
&\quad + 12C_3C_4 - 16C_4C_2^2 + 12C_4C_3 + 10C_5C_2 - 6C_6.
\end{aligned} \tag{4.7}$$

Thus, multiplying (4.5) by (4.3) and replacing them in the first step of (4.1),

$$\begin{aligned}
y^{(k)} &= \xi - \left[-C_2e^{(k)^2} + (-2C_3 + 2C_2^2)e^{(k)^3} + A_4e^{(k)^4} + A_5e^{(k)^5} + A_6e^{(k)^6}\right]\\
&\quad + \mathcal{O}(e^{(k)^7}),
\end{aligned} \tag{4.8}$$

where

$$\begin{aligned}
A_4 &= -3C_4 + 4C_2C_3 - 4C_2^3 + 3C_3C_2,\\
A_5 &= -4C_5 + 6C_2C_4 - 8C_2^2C_3 + 6C_3^2 + 8C_2^4 - 6C_2C_3C_2 - 6C_3C_2^2 + 4C_4C_2,\\
A_6 &= -5C_6 + 8C_2C_5 - 12C_2^2C_4 + 9C_3C_4 + 16C_2^3C_3 - 12C_2C_3^2 - 12C_3C_2C_3\\
&\quad + 8C_4C_3 - 16C_2^5 + 12C_2^2C_3C_2 + 12C_2C_3C_2^2 - 8C_2C_4C_2 - 9C_3^2C_2\\
&\quad + 12C_3C_2^3 - 8C_4C_2^2 + 5C_5C_2.
\end{aligned} \tag{4.9}$$

Again, by means of Taylor series, we develop $F(y^{(k)})$ around ξ, with $e_y^{(k)} = y^{(k)} - \xi$, we get

$$\begin{aligned}
F(y^{(k)}) &= F'(\xi)\left[C_2e^{(k)^2} + (2C_3 - 2C_2^2)e^{(k)^3} + B_4e^{(k)^4} + B_5e^{(k)^5} + B_6e^{(k)^6}\right]\\
&\quad + \mathcal{O}(e^{(k)^7}),
\end{aligned} \tag{4.10}$$

where

$$\begin{aligned}
B_4 &= -A_4 + C_2A_2^2,\\
B_5 &= -A_5 + C_2A_2A_3 + C_2A_3A_2,\\
B_6 &= -A_6 + C_2A_2A_4 + C_2A_3^2 + C_2A_4A_2 - C_3A_2^3.
\end{aligned} \tag{4.11}$$

In order to proof the order of convergence of the second step of (4.1), we use the Genocchi-Hermite formula (see [79])

$$[x, x+h; F] = \int_0^1 F'(x+th)\,dt \tag{4.12}$$

Expanding $F'(x+th)$ in Taylor series around x,

$$\begin{aligned}\int_0^1 F'(x+th)\,dt = F'(x) + \frac{1}{2!}F''(x)h + \frac{1}{3!}F'''(x)h^2 + \frac{1}{4!}F^{(iv)}(x)h^3 \\ + \frac{1}{5!}F^{(v)}(x)h^4 + \mathcal{O}(h^5).\end{aligned} \tag{4.13}$$

Denoting by $e = x - \xi$ and taking into account that $F'(\xi)$ is nonsingular, we get

$$[x^{(k)}, y^{(k)}; F] = F'(\xi)\left[I + P_1 e^{(k)} + P_2 {e^{(k)}}^2 + P_3 {e^{(k)}}^3 + P_4 {e^{(k)}}^4\right] + \mathcal{O}({e^{(k)}}^5), \tag{4.14}$$

being the error at the first step denoted by $e_y^{(k)} = y^{(k)} - \xi$. In this expression,

$$\begin{aligned} P_1 &= C_2, \\ P_2 &= C_2^2 + C_3, \\ P_3 &= C_4 + 2C_2C_3 + C_3C_2 - 2C_2^3, \\ P_4 &= C_5 + 3C_2C_4 - 4C_2^2C_3 + 4C_2^4 - 3C_2C_3C_2 + 2C_3^2 - C_3C_2^2 + C_4C_2. \end{aligned} \tag{4.15}$$

Now, by denoting $M = 2[x^{(k)}, y^{(k)}; F] - F'(x^{(k)})$ and we get

$$M = F'(\xi)\left[I + M_2 {e^{(k)}}^2 + M_3 {e^{(k)}}^3 + M_4 {e^{(k)}}^4\right] + \mathcal{O}({e^{(k)}}^5), \tag{4.16}$$

where

$$\begin{aligned} M_2 &= 2C_2^2 - C_3, \\ M_3 &= 2\left(-C_4 + 2C_2C_3 + C_3C_2 - 2C_2^3\right), \\ M_4 &= -3C_5 + 6C_2C_4 - 8C_2^2C_3 + 8C_2^4 - 6C_2C_3C_2 + 4C_3^2 - 2C_3C_2^2 + 2C_4C_2. \end{aligned} \tag{4.17}$$

The inverse of M must satisfy

$$M^{-1}M = I, \tag{4.18}$$

being

$$M^{-1} = \left[I + Y_1 e^{(k)} + Y_2 {e^{(k)}}^2 + Y_3 {e^{(k)}}^3 + Y_4 {e^{(k)}}^4\right]\left[F'(\xi)\right]^{-1} + \mathcal{O}({e^{(k)}}^5), \tag{4.19}$$

where $Y_1, ..., Y_4$ are unknowns. Then, replacing M^{-1} and M in (4.18), we have

$$\begin{aligned} Y_1 &= 0, \\ Y_2 &= -2C_2^2 + C_3, \\ Y_3 &= 2C_4 - 4C_2C_3 - 2C_3C_2 + 4C_2^3, \\ Y_4 &= 3C_5 - 6C_2C_4 + 6C_2^2C_3 - 4C_2^4 + 6C_2C_3C_2 - 3C_3^2 - 2C_4C_2. \end{aligned} \tag{4.20}$$

Next, we denote $L = M^{-1}F(y^{(k)})$ and we obtain

$$L = C_2e^{(k)^2} + 2\left(C_3 - C_2^2\right)e^{(k)^3} + L_4e^{(k)^4} + L_5e^{(k)^5} + L_6e^{(k)^6} + \mathcal{O}(e^{(k)^7}), \tag{4.21}$$

where

$$\begin{aligned} L_4 &= 3C_4 - 4C_2C_3 - 2C_3C_2 + 3C_2^3, \\ L_5 &= 4C_5 - 6C_2C_4 + 6C_2^2C_3 - 4C_3^2 - 4C_2^4 + 4C_2C_3C_2 + 2C_3C_2^2 - 2C_4C_2, \\ L_6 &= 5C_6 - 8C_2C_5 + 9C_2^2C_4 - 6C_3C_4 - 8C_2^3C_3 + 8C_2C_3^2 + 4C_3C_2C_3 - 4C_4C_3 \\ &\quad + 6C_2^5 - 7C_2^2C_3C_2 - 5C_2C_3C_2^2 + 5C_2C_4C_2 + 3C_3^2C_2 - 2C_3C_2^3 + 2C_4C_2^2 \\ &\quad - 2C_5C_2. \end{aligned} \tag{4.22}$$

Therefore,

$$z^{(k)} = y^{(k)} - L = \xi - \left[K_4e^{(k)^4} + K_5e^{(k)^5} + K_6e^{(k)^6}\right] + \mathcal{O}(e^{(k)^7}), \tag{4.23}$$

where

$$\begin{aligned} K_4 &= -C_2^3 + C_3C_2, \\ K_5 &= -2C_2^2C_3 + 2C_3^2 + 4C_2^4 - 2C_2C_3C_2 - 4C_3C_2^2 + 2C_4C_2, \\ K_6 &= -3C_2^2C_4 + 3C_3C_4 + 8C_2^3C_3 - 4C_2C_3^2 - 8C_3C_2C_3 + 4C_4C_3 - 10C_2^5 \\ &\quad + 5C_2^2C_3C_2 + 7C_2C_3C_2^2 - 3C_2C_4C_2 - 6C_3^2C_2 + 10C_3C_2^3 - 6C_4C_2^2 + 3C_5C_2. \end{aligned} \tag{4.24}$$

Similarly, and denoting by $e_z^{(k)} = z^{(k)} - \xi$,

$$F(z^{(k)}) = F'(\xi)\left[-K_4e^{(k)^4} - K_5e^{(k)^5} - K_6e^{(k)^6}\right] + \mathcal{O}(e^{(k)^7}). \tag{4.25}$$

Using (4.5) and (4.25), and denoting by $N = [F'(x^{(k)})]^{-1}F(z^{(k)})$, we get

$$N = (C_2^3 - C_3C_2)e^{(k)^4} + N_5e^{(k)^5} + N_6e^{(k)^6} + \mathcal{O}(e^{(k)^7}), \tag{4.26}$$

$$\begin{aligned} N_5 &= 2C_2^2C_3 - 2C_3^2 - 6C_2^4 + 4C_2C_3C_2 + 4C_3C_2^2 - 2C_4C_2, \\ N_6 &= 3C_2^2C_4 - 3C_3C_4 - 12C_2^3C_3 + 8C_2C_3^2 + 8C_3C_2C_3 - 4C_4C_3 + 22C_2^5 \\ &\quad - 13C_2^2C_3C_2 - 15C_2C_3C_2^2 + 7C_2C_4C_2 + 9C_3^2C_2 - 13C_3C_2^3 + 6C_4C_2^2 \\ &\quad - 3C_5C_2. \end{aligned} \tag{4.27}$$

Then, replacing (4.5) and (4.14) in $u^{(k)}$,

$$u^{(k)} = C_2 e^{(k)} + (-3C_2^2 + 2C_3) e^{(k)^2} + \mathcal{O}(e^{(k)^3}), \tag{4.28}$$

Now, we find the Taylor series expansion of $[x^{(k)}, y^{(k)}; F]^{-1}$ as follows

$$[x^{(k)}, y^{(k)}; F]^{-1} = \left[I + R_1 e^{(k)} + R_2 e^{(k)^2}\right] \left[F'(\xi)\right]^{-1} + \mathcal{O}(e^{(k)^3}), \tag{4.29}$$

where R_1 and R_2 are unknowns such that

$$[x^{(k)}, y^{(k)}; F]^{-1} [x^{(k)}, y^{(k)}; F] = I. \tag{4.30}$$

So, we get

$$\begin{aligned} R_1 &= -C_2, \\ R_2 &= -C_3. \end{aligned} \tag{4.31}$$

Thus, substituting (4.29) and (4.4) in $v^{(k)}$,

$$v^{(k)} = I + v_1 e^{(k)} + v_2 e^{(k)^2} + \mathcal{O}(e^{(k)^3}), \tag{4.32}$$

where

$$\begin{aligned} v_1 &= C_2, \\ v_2 &= 2C_3 - 2C_2^2. \end{aligned} \tag{4.33}$$

Denoting by $T = (\alpha I + \beta u^{(k)} + \gamma v^{(k)}) N$, and using (4.28) and (4.32), we get

$$T = (\alpha + \gamma)(C_2^3 - C_3 C_2) e^{(k)^4} + T_5 e^{(k)^5} + T_6 e^{(k)^6} + \mathcal{O}(e^{(k)^7}), \tag{4.34}$$

$$\begin{aligned} T_5 &= (\alpha + \gamma) N_5 + (\beta + \gamma) u_1 N_4, \\ T_6 &= (\alpha + \gamma) N_6 + (\beta + \gamma) u_1 N_5 + (\beta u_2 + \gamma v_2) N_4. \end{aligned} \tag{4.35}$$

Finally, using (4.23) and (4.34),

$$x^{(k+1)} = \xi - \left[W_4 e^{(k)^4} + W_5 e^{(k)^5} + W_6 e^{(k)^6}\right] + \mathcal{O}(e^{(k)^7}), \tag{4.36}$$

where

$$\begin{aligned}
W_4 &= (\alpha+\gamma-1)\left(C_2^3 - C_3C_2\right),\\
W_5 &= 2(\alpha+\gamma-1)\left(C_2^2C_3 - C_3^2 + 2C_3C_2^2 - C_4C_2\right) - (6\alpha-\beta+5\gamma-4)C_2^4\\
&\quad + (4\alpha-\beta+3\gamma-2)C_2C_3C_2,\\
W_6 &= -3C_2^2C_4 + 3C_3C_4 + 8C_2^3C_3 - 4C_2C_3^2 - 8C_3C_2C_3 + 4C_4C_3 - 10C_2^5 + 5C_2^2C_3C_2\\
&\quad + 7C_2C_3C_2^2 - 3C_2C_4C_2 - 6C_3^2C_2 + 10C_3C_2^3 - 6C_4C_2^2 + 3C_5C_2\\
&\quad + (\alpha+\gamma)\Big(3C_2^2C_4 - 3C_3C_4 - 12C_2^3C_3 + 8C_2C_3^2 + 8C_3C_2C_3 - 4C_4C_3 + 22C_2^5\\
&\quad - 13C_2^2C_3C_2 - 15C_2C_3C_2^2 + 7C_2C_4C_2 + 9C_3^2C_2 - 13C_3C_2^3 + 6C_4C_2^2 - 3C_5C_2\Big)\\
&\quad + (\beta+\gamma)C_2\left(2C_2^2C_3 - 2C_3^2 - 6C_2^4 + 4C_2C_3C_2 + 4C_3C_2^2 - 2C_4C_2\right)\\
&\quad + \left(\beta\left(-3C_2^2 + 2C_3\right) + \gamma\left(2C_3 - 2C_2^2\right)\right)\left(C_2^3 - C_3C_2\right),
\end{aligned} \tag{4.37}$$

being the error equation

$$\begin{aligned}
e^{(k+1)} &= -W_4 e^{(k)^4} - W_5 e^{(k)^5} - W_6 e^{(k)^6} + \mathcal{O}(e^{(k)^7})\\
&= (1-\alpha-\gamma)\left(C_2^3 - C_3C_2\right)e^{(k)^4} + \mathcal{O}(e^{(k)^5}).
\end{aligned} \tag{4.38}$$

This finishes the proof. □

From Theorem 4.1, the triparametric family is fourth-order convergent for any α, β and γ. Nevertheless, the order of convergence can be accelerated by reducing the number of parameters, resulting in an uniparametric family.

Theorem 4.2 (Uni-parametric family). *Let us consider a sufficiently differentiable function $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ defined in a convex open set D. Let also $\xi \in D$ be a solution of the nonlinear system $F(x) = 0$. Assuming that $F'(x)$ is nonsingular and continuous at ξ and $x^{(0)}$ is a seed close enough to ξ. Under these hypotheses, the sequence $\{x^{(k)}\}_{k\geq 0}$ obtained by using* (4.1)*, converges to ξ with sixth order of convergence, only if $\gamma = 1-\alpha$ and $\beta = 1+\alpha$. Therefore, its error equation*

$$e^{(k+1)} = \left(C_3^2C_2 - C_3C_2^3 + 6C_2^5 - 6C_2^2C_3C_2\right)e^{(k)^6} + \mathcal{O}(e^{(k)^7}),$$

where $C_q = \frac{1}{q!}[F'(\xi)]^{-1}F^{(q)}(\xi)$, $q = 2, 3, \ldots$, and $e^{(k)} = x^{(k)} - \xi$.

Proof. Using the results of Theorem 4.1, to cancel W_4 and W_5, coefficients of $e^{(k)^4}$ and $e^{(k)^5}$ in (4.38), respectively, $\alpha+\gamma = 1$, $6\alpha-\beta+5\gamma = 4$ and $4\alpha-\beta+3\gamma = 2$ must be satisfied. This system has infinite solutions for

$$\beta = 1+\alpha \quad \text{and} \quad \gamma = 1-\alpha, \tag{4.39}$$

being α is a disposable parameter. Then, replacing (4.39) in (4.37), we get that

$$W_4 = 0, \quad W_5 = 0, \quad \text{and} \quad W_6 = -C_3^2C_2 + C_3C_2^3 - 6C_2^5 + 6C_2^2C_3C_2, \tag{4.40}$$

being the error equation

$$\begin{aligned} e^{(k+1)} &= -W_6 {e^{(k)}}^6 + \mathcal{O}({e^{(k)}}^7) \\ &= \left(C_3^2C_2 - C_3C_2^3 + 6C_2^5 - 6C_2^2C_3C_2\right) {e^{(k)}}^6 + \mathcal{O}({e^{(k)}}^7). \end{aligned} \tag{4.41}$$

This finishes the proof. □

As follows from Theorem 4.2, replacing $\beta = 1+\alpha$ and $\gamma = 1-\alpha$ in (4.1), the tri-parametric family becomes a uniparametric family with sixth-order of convergence. Thus, the iterative expression of the new three-step family dependent on α – denoted henceforth as MCCT(α) – is

$$\begin{cases} y^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} = y^{(k)} - [2[x^{(k)}, y^{(k)}; F] - F'(x^{(k)})]^{-1}F(y^{(k)}), \\ x^{(k+1)} = z^{(k)} - (\alpha I + (1+\alpha)u^{(k)} + (1-\alpha)v^{(k)})[F'(x^{(k)})]^{-1}F(z^{(k)}), \end{cases} \tag{4.42}$$

being $v^{(k)} = [x^{(k)}, y^{(k)}; F]^{-1}F'(x^{(k)})$, $u^{(k)} = I - [F'(x^{(k)})]^{-1}[x^{(k)}, y^{(k)}; F]$, $k = 0, 1, 2, ...$, and α an arbitrary parameter.

The stability of MCCT(α) family is analyzed to select its best members. This study is carried out using the real dynamical tools appearing in Section 4.3.

4.3 Real Dynamics for Stability

This section refers to the analysis of the dynamical behavior of the rational operator related with iterative schemes of MCCT(α) family. It provides significative information about the reliability and stability of the class. We construct rational operators and their dynamical planes in order to see the performance of particular schemes from the different basins of attraction.

4.3.1 Rational operator

Rational operators are built on low-degree non-linear polynomial systems, since the criterion of stability of a method applied to these systems can be generalized for other multidimensional cases. Thus, we propose the following two non-linear systems: one of separated variables $F(x_1, x_2)$ and another of non-separated variables $G(x_1, x_2)$, as shown

$$F(x_1, x_2) = \left(x_1^2 - 1, x_2^2 - 1\right) = (0, 0), \tag{4.43}$$

$$G(x_1, x_2) = \left(x_1^2 + x_2^2 - 1, x_1^2 - x_2^2 - \frac{1}{2}\right) = (0, 0)\,. \tag{4.44}$$

Proposition 4.1 (rational operator R_F). *Let us consider the polynomial system $F(x_1, x_2)$, given in (4.43), with roots $(-1,-1)$, $(-1,1)$, $(1,-1)$, $(1,1) \in \mathbb{R}^2$. The rational operator associated with MCCT(α) family and applied on $F(x_1, x_2)$, with $\alpha \in \mathbb{R}$ an arbitrary parameter, is*

$$R_F(x_1, x_2, \alpha) = \left(R_{F_{11}}, R_{F_{12}}\right), \tag{4.45}$$

where

$$\begin{aligned} R_{F_{11}} &= \frac{1}{32}\left(\frac{\left(x_1^2-1\right)^4\left(\alpha+(\alpha-19)x_1^4-2(\alpha-1)x_1^2+1\right)}{4x_1^5\left(x_1^2+1\right)^2\left(3x_1^2+1\right)} + \frac{8\left(x_1^4+6x_1^2+1\right)}{x_1^3+x_1}\right. \\ &\qquad \left. - \frac{\alpha\left(x_2^2-1\right)^4}{x_2^3\left(x_2^2+1\right)^2}\right), \\ R_{F_{12}} &= \frac{1}{32}\left(\frac{\left(x_2^2-1\right)^4\left(\alpha+(\alpha-19)x_2^4-2(\alpha-1)x_2^2+1\right)}{4x_2^5\left(x_2^2+1\right)^2\left(3x_2^2+1\right)} + \frac{8\left(x_2^4+6x_2^2+1\right)}{x_2^3+x_2}\right. \\ &\qquad \left. - \frac{\alpha\left(x_1^2-1\right)^4}{x_1^3\left(x_1^2+1\right)^2}\right). \end{aligned}$$

From Proposition 4.1, note that the rational operator $R_F(x_1, x_2, \alpha)$ is obtained by substituting the nonlinear system $F(x_1, x_2)$ into the iterative scheme of the MCCT(α) family. To simplify R_F, we can select a value of α that cancels terms of the expression and reduces it. It is easy to show that for $\alpha = 0$ the rational operator is simpler and there will be fewer fixed and critical points that can improve the performance of the associated method. In addition, the components of $R_F(x_1, x_2, 0)$ will be of separate variables, as shown

$$\begin{aligned} R_F(x_1, x_2, 0) = \Bigg(& \frac{77x_1^{12}+782x_1^{10}+775x_1^8+404x_1^6+11x_1^4-2x_1^2+1}{128x_1^5\left(x_1^2+1\right)^2\left(3x_1^2+1\right)}, \\ & \frac{77x_2^{12}+782x_2^{10}+775x_2^8+404x_2^6+11x_2^4-2x_2^2+1}{128x_2^5\left(x_2^2+1\right)^2\left(3x_2^2+1\right)}\Bigg). \end{aligned} \tag{4.46}$$

Proposition 4.2 (rational operator R_G). *Let us consider the polynomial system $G(x_1, x_2)$, given in (4.44), with roots $\left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$, $\left(-\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$, $\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$, $\left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right) \in \mathbb{R}^2$. The rational operator associated with MCCT(α) family and applied on $G(x_1, x_2)$, with $\alpha \in \mathbb{R}$ an arbitrary parameter, is*

$$R_G(x_1, x_2, \alpha) = \left(R_{G_{11}}, R_{G_{12}}\right), \tag{4.47}$$

where

$$R_{G_{11}} = \frac{\left(3-4x_1^2\right)^4\left(9(\alpha+1)+16(\alpha-19)x_1^4-24(\alpha-1)x_1^2\right)}{24576x_1^5\left(4x_1^2+1\right)\left(4x_1^2+3\right)^2} + \frac{16x_1^4+72x_1^2+9}{64x_1^3+48x_1} - \frac{\alpha\left(1-4x_2^2\right)^4}{512x_2^3\left(4x_2^2+1\right)^2},$$

$$R_{G_{12}} = \frac{\left(1-4x_2^2\right)^4\left(\alpha+16(\alpha-19)x_2^4-8(\alpha-1)x_2^2+1\right)}{8192x_2^5\left(4x_2^2+1\right)^2\left(12x_2^2+1\right)} + \frac{16x_2^4+24x_2^2+1}{64x_2^3+16x_2} - \frac{\alpha\left(3-4x_1^2\right)^4}{512x_1^3\left(4x_1^2+3\right)^2}.$$

From Proposition 4.2, note that the rational operator $R_G(x_1, x_2, \alpha)$ is also obtained by substituting the nonlinear system $G(x_1, x_2)$ into the iterative scheme of the MCCT(α) family. In the same way as before for R_F, it is easy to prove that for $\alpha = 0$ the rational operator R_G is simpler. Moreover, the components of $R_G(x_1, x_2, 0)$ will be of separate variables, as shown

$$R_G(x_1,x_2,0) = \left(\frac{\left(-304x_1^4+24x_1^2+9\right)\left(3-4x_1^2\right)^4}{24576x_1^5\left(4x_1^2+1\right)\left(4x_1^2+3\right)^2} + \frac{16x_1^4+72x_1^2+9}{64x_1^3+48x_1}, \frac{\left(-304x_2^4+8x_2^2+1\right)\left(1-4x_2^2\right)^4}{8192x_2^5\left(4x_2^2+1\right)^2\left(12x_2^2+1\right)} + \frac{16x_2^4+24x_2^2+1}{64x_2^3+16x_2}\right). \tag{4.48}$$

With these two rational operators, $R_F(x_1, x_2, \alpha)$ and $R_G(x_1, x_2, \alpha)$, we study the stability of the family MCCT(α) by means of dynamical planes built for different values of α. These planes show the complexity of the iterative class.

4.3.2 Fixed points and their stability

The fixed points are calculated from the rational operators $R_F(x_1, x_2, \alpha)$ and $R_G(x_1, x_2, \alpha)$ given in (4.45) and (4.47), respectively. With these points, we analyze their stability.

Proposition 4.3 (R_F fixed points). *The real fixed points of $R_F(x_1, x_2, \alpha)$ are the roots of the equation $R_F(x_1, x_2, \alpha) = (x_1, x_2)$. That is*

$$fp_1 = (-1,-1), fp_2 = (-1,1), fp_3 = (1,-1), fp_4 = (1,1),$$

that correspond to the roots of the polynomial system $F(x_1, x_2)$ given in (4.43)*, and they are also superattracting. Other strange fixed points may appear, but their components are roots of polynomials of degree 120.*

Proposition 4.4 (R_G fixed points). *The real fixed points of $R_G(x_1, x_2, \alpha)$ are the roots of the equation $R_G(x_1, x_2, \alpha) = (x_1, x_2)$. That is*

$$fp_1 = \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), fp_2 = \left(-\frac{\sqrt{3}}{2}, \frac{1}{2}\right), fp_3 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), fp_4 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right),$$

that correspond to the roots of the polynomial system $G(x_1, x_2)$ given in (4.44)*, and they are also superattracting. Other strange fixed points may appear, but their components are roots of polynomials of degree 120.*

From Propositions 4.3 and 4.4, we establish there are a minimum of 4 fixed points for $F(x_1, x_2)$ and $G(x_1, x_2)$ polynomial systems. Of these, from fp_1 to fp_4 correspond to the roots of the original systems and are attractive and critical points.

4.3.3 Dynamical planes

We perform the stability analysis of the MCCT(α) family by representing dynamical planes of the rational operators $R_F(x_1, x_2, \alpha)$ and $R_G(x_1, x_2, \alpha)$. Two values of α of different behavior in the parameter space of the Figure 4.1 have been chosen: the value $\alpha = 0$ is in the red zone which implies convergence, and the value $\alpha = 200$ is in the black zone which does not guarantee convergence. This parameter space was obtained from the MCCT(α) family for scalar cases [36] and their results have been extrapolated for vector cases.

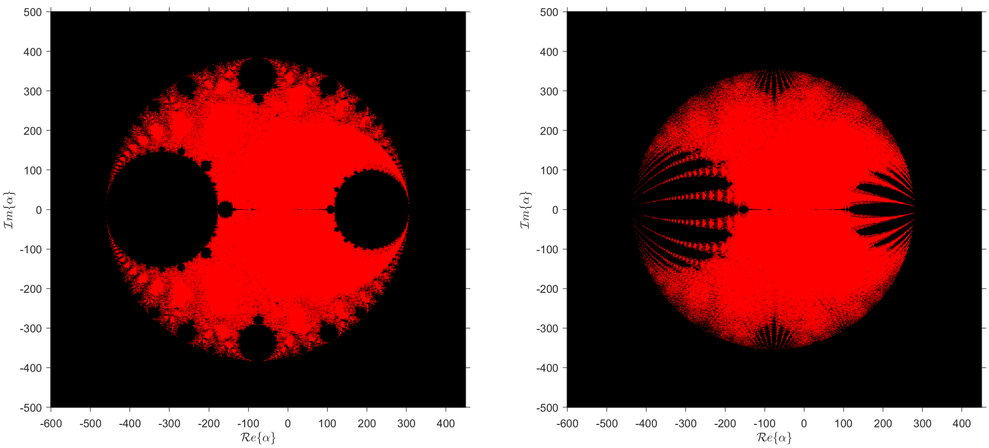


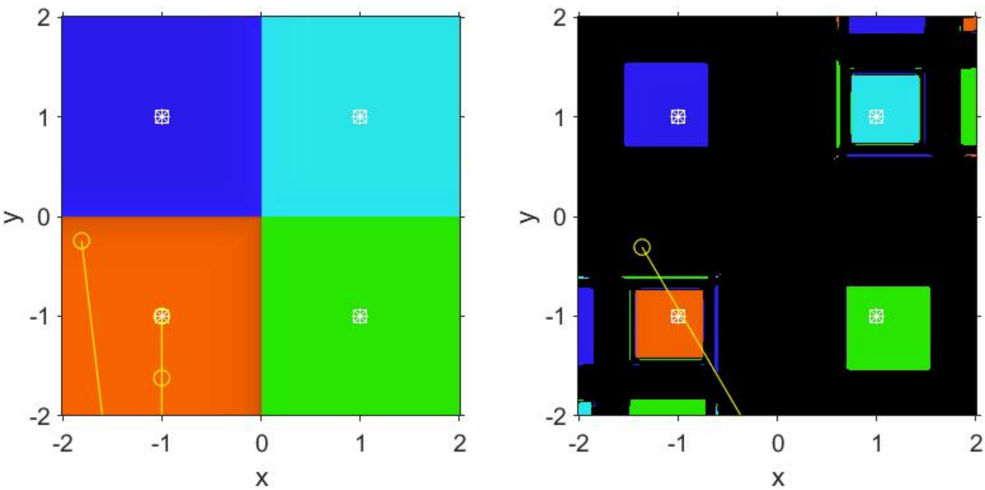
Figure 4.1: Parameter spaces of some free critical points of the family MCCT(α) applied to a non-linear polynomial equation $(x-a)(x-b) = 0$, where $a, b \in \mathbb{C}$

A dynamical plane is represented by a mesh of 400×400 points in $\mathbb{R}^2$. Each point of the mesh is a seed of the iterative process. It shows the convergence of the scheme with a maximum of

50 iterations and a stopping criterion of $||x^{(k+1)} - x^{(k)}|| < 10^{-3}$. Each root is color assigned. The color of the mesh points indicates which root it converges to, with black being the points at which the maximum number of iterations is reached; the brighter the color, the fewer the number of iterations. Fixed points are represented in white color by a circle '$\bigcirc$', critical points by a square '$\square$', and attractors by an asterisk '$*$'. The resulting plane is represented using Matlab R2020b.

The dynamical planes corresponding to $R_F(x_1, x_2, 0)$ and $R_F(x_1, x_2, 200)$, on the one hand, and for $R_G(x_1, x_2, 0)$ and $R_G(x_1, x_2, 200)$, on the other, are shown in Figure 4.2 and Figure 4.3, respectively. In both cases, some yellow convergence orbits are observed.

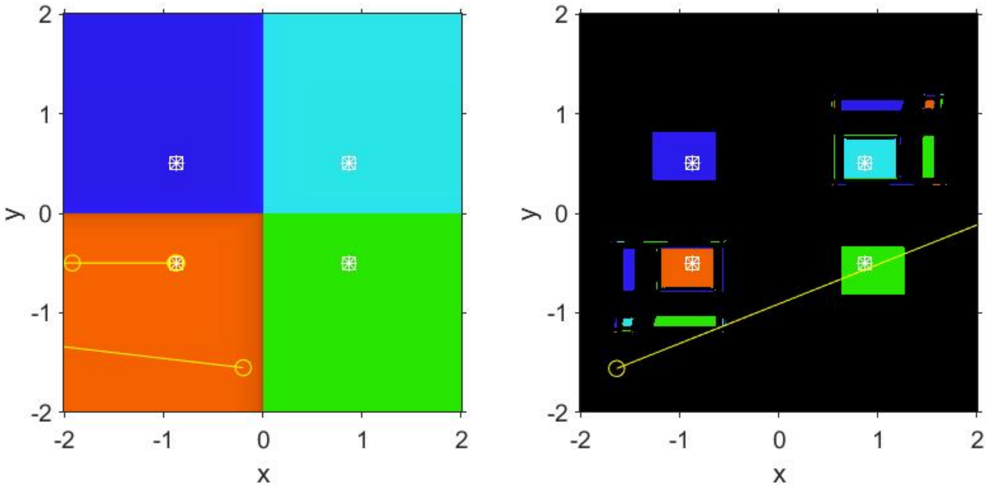
The method for $\alpha = 0$ presents in both cases four basins of attraction associated with the roots. No black areas are observed. Consequently, this method shows a good dynamic behavior. In contrast, the method for $\alpha = 200$ presents in R_F and R_G the same four basins of attraction associated to the roots, but of reduced size, which minimizes the possibilities of convergence to the solution. Black areas of slow convergence of the method are observed. In consequence, this method performs poorly dynamically.



(a) Convergence to $pf_1 = (-1, -1)$ for $\alpha = 0$ and an initial estimation $x^{(0)}$ close to the roots

(b) No convergence to any pf for $\alpha = 200$ and an initial estimation $x^{(0)}$ close to the roots

Figure 4.2: Dynamical planes for $R_F(x_1, x_2, \alpha)$



(a) Convergence to $pf_1 \approx (-0.87, -0.5)$ for $\alpha = 0$ and an initial estimation $x^{(0)}$ close to the roots

(b) No convergence to any pf for $\alpha = 200$ and an initial estimation $x^{(0)}$ close to the roots

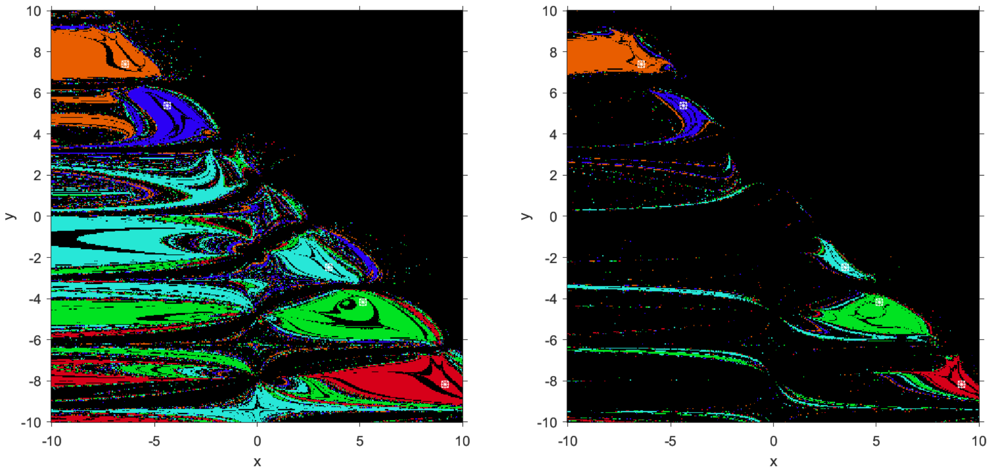
Figure 4.3: Dynamical planes for $R_G(x_1, x_2, \alpha)$

From Figure 4.2 and Figure 4.3, the basins of attraction have a similar behavior for the rational operators R_F and R_G with $\alpha = 0$. However, for $\alpha = 200$, these basins are reduced, and the associated iterative methods do not easily converge to the solution.

If we consider non-linear systems that involve logarithmic, trigonometric and exponential functions, as well as polynomial functions, the behavior of the representative members of the MCCT(α) family, for $\alpha = 0$ and $\alpha = 200$, is similar to what has already been studied. For example, if we analyze the systems shown in Table 4.1, we observe in their dynamical planes (see Figure 4.4, Figure 4.5 and Figure 4.6) that the regions of the basins of attraction for $\alpha = 0$ are much larger than for $\alpha = 200$, increasing the chances of converging to the solution for the first case. In addition, more regions of slow convergence or non-convergence are observed for the MCCT(200) iterative method, compared to the MCCT(0) method.

Table 4.1: Tested non-linear systems for dynamical analysis.

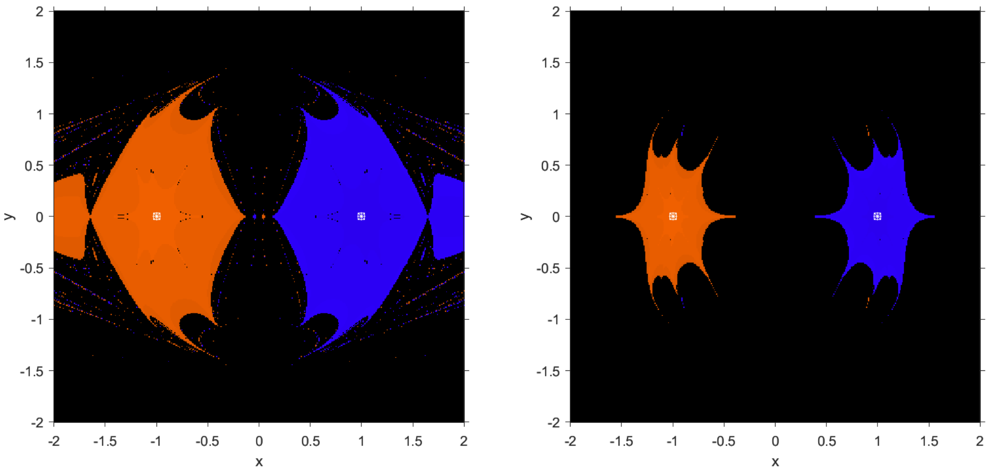
Non-linear system	Some roots
$M(x_1, x_2)$: $(e^{x_1}e^{x_2} + x_1 \cos(x_2), x_1 + x_2 - 1) = (0, 0)$	$\xi \approx (-6.4165, 7.4165; -4.3816, 5.3816; 3.4706, -2.4706; 5.1572, -4.1572; 9.1554, -8.1554)^T$
$N(x_1, x_2)$: $\left(\ln(x_1^2) - 2\ln(\cos(x_2)), x_1 \tan(x_2)\right) = (0, 0)$	$\xi = (-1, 0; 1, 0)^T$
$O(x_1, x_2)$: $(x_1 + e^{x_2} - \cos(x_2) + 0.5, 3x_1 - x_2 - \sin(x_2)) = (0, 0)$	$\xi \approx (-0.2535, -0.3851; -0.9389, -1.8576; -1.0935, -4.0974)^T$



(a) Considering the MCCT(0) method

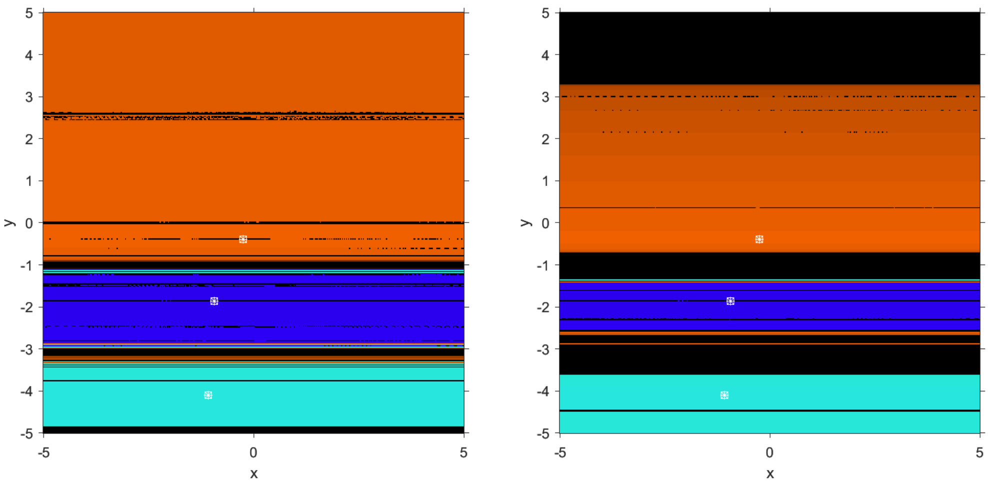
(b) Considering the MCCT(200) method

Figure 4.4: Dynamical planes for $M(x_1, x_2)$ system



(a) Considering the MCCT(0) method **(b)** Considering the MCCT(200) method

Figure 4.5: Dynamical planes for $N(x_1, x_2)$ system



(a) Considering the MCCT(0) method **(b)** Considering the MCCT(200) method

Figure 4.6: Dynamical planes for $O(x_1, x_2)$ system

4.4 Numerical Results

Several numerical tests are carried out to check the performance of MCCT(α) family. We are going to prove the theoretical results of convergence and stability. We employ two members of the class used before as representatives: MCCT(0) and MCCT(200). These methods are applied to the same two-by-two non-linear test systems seen above and to new three-by-three and four-by-four systems. They and their corresponding roots are summarized in Table 4.2.

Table 4.2: Non-linear test systems and their roots

Non-linear test system	Roots
$F(x_1,x_2)$: $\left(x_1^2-1, x_2^2-1\right)=(0,0)$	$\xi=(1,1)^T$
$G(x_1,x_2)$: $\left(x_1^2+x_2^2-1, x_1^2-x_2^2-\frac{1}{2}\right)=(0,0)$	$\xi=\left(\frac{\sqrt{3}}{2},\frac{1}{2}\right)^T$
$M(x_1,x_2)$: $(e^{x_1}e^{x_2}+x_1\cos(x_2), x_1+x_2-1)=(0,0)$	$\xi\approx(3.4706,-2.4706)^T$
$N(x_1,x_2)$: $\left(\ln(x_1^2)-2\ln(\cos(x_2)), x_1\tan(x_2)\right)=(0,0)$	$\xi=(1,0)^T$
$O(x_1,x_2)$: $(x_1+e^{x_2}-\cos(x_2)+0.5, 3x_1-x_2-\sin(x_2))=(0,0)$	$\xi\approx(-0.2535,-0.3851)^T$
$P(x_1,x_2,x_3)$: $\left(\cos(x_2)-\sin(x_1), x_3^{x_1}-\frac{1}{x_2}, e^{x_1}-x_3^2\right)=(0,0)$	$\xi\approx(0.9096,0.6612,1.5758)^T$
$Q(x_1,x_2,x_3,x_4)$: $(x_2x_3+x_4(x_2+x_3), x_1x_3+x_4(x_1+x_3),$ $x_1x_2+x_4(x_1+x_2), x_1x_2+x_1x_3+x_2x_3-1)=(0,0)$	$\xi\approx(0.5774,0.5774,$ $0.5774,-0.2887)^T$

A comparison of MCCT(0) is conducted against three methods from the literature: Newton's [67], Ostrowski's [30], and HMT's method [80]. Table 4.3 collects the numerical results, taking initial guesses $x^{(0)}$ close to ξ solutions.

The computations have been performed in Matlab R2020b with variable precision arithmetics, with a mantissa of 200 digits. For each scheme, the amount of iterations (iter) needed to converge to the solution has been analyzed, in such a way that the stopping criteria $|||x^{(k+1)} - x^{(k)}|| < 10^{-100}$ or $|||F(x^{(k+1)})|| < 10^{-100}$ are satisfied.

The approximate computational order of convergence (ACOC) [33] is obtained. The ACOC column is 'nc' if the number of iterations reaches 50, or '-' if the ACOC does not stabilize.

Table 4.3: Numerical results of MCCT(0) and known schemes on test problems for $x^{(0)} \approx \xi$

System	Method	$\|\|x^{(k+1)} - x^{(k)}\|\|$	$\|\|F(x^{(k+1)})\|\|$	iter	ACOC
$F(x_1, x_2)$	MCCT(0)	4.1590e-41	1.0578e-162	3	6.0326
$x^{(0)} = (0.90, 0.90)^T$	Newton	4.0862e-82	1.1806e-163	7	2.0000
	Ostrowski	2.3572e-61	6.5488e-183	4	-
	HMT	2.1362e-52	5.5061e-208	3	-
$G(x_1, x_2)$	MCCT(0)	1.6140e-29	3.8389e-115	3	5.9785
$x^{(0)} = (0.80, 0.40)^T$	Newton	8.4816e-62	1.0174e-122	7	2.0000
	Ostrowski	3.1433e-46	8.7844e-137	4	-
	HMT	2.5671e-36	1.9467e-208	3	-
$M(x_1, x_2)$	MCCT(0)	1.1444e-49	1.3224e-131	3	5.5845
$x^{(0)} = (3.40, -2.40)^T$	Newton	2.4421e-57	2.1989e-114	6	2.0000
	Ostrowski	3.9750e-66	3.0486e-148	4	-
	HMT	8.1589e-54	7.7869e-208	3	5.9851
$N(x_1, x_2)$	MCCT(0)	1.0160e-76	5.8602e-308	4	-
$x^{(0)} = (0.90, 0.10)^T$	Newton	1.6691e-73	1.5673e-146	7	2.0000
	Ostrowski	6.5957e-87	1.4347e-259	5	-
	HMT	3.0359e-41	3.8934e-208	3	6.1133
$O(x_1, x_2)$	MCCT(0)	1.2709e-37	6.3818e-107	3	5.9417
$x^{(0)} = (-0.20, -0.30)^T$	Newton	2.3676e-73	3.4799e-146	7	2.0000
	Ostrowski	7.1310e-54	3.2193e-123	4	-
	HMT	2.3769e-43	4.0133e-208	3	5.9289
$P(x_1, x_2, x_3)$	MCCT(0)	1.3178e-66	1.2841e-162	4	-
$x^{(0)} = (0.80, 0.60, 1.50)^T$	Newton	1.4817e-63	2.0520e-126	7	1.9802
	Ostrowski	2.3811e-82	2.8239e-179	5	-
	HMT	6.1154e-24	5.9378e-139	3	6.2016
$Q(x_1, x_2, x_3, x_4)$	MCCT(0)	2.4839e-22	2.0342e-128	3	5.6492
$x^{(0)} = (0.50, 0.50, 0.50, -0.20)^T$	Newton	3.2002e-72	7.8134e-145	7	2.0156
	Ostrowski	4.1778e-49	3.0779e-157	4	4.0962
	HMT	2.0321e-44	1.6859e-208	3	-

Table 4.3 notices that MCCT(0) converges to ξ, even with fewer iterations than the other methods in five of the seven nonlinear systems. The theoretical order of convergence is also achieved by ACOC, being close to 6. This method has been analyzed for seeds near and far from the solution, i.e., for $x^{(0)} \approx 3\xi$ and $x^{(0)} > 10\xi$, respectively. The obtained results are collected in Tables 4.4 and 4.5.

Table 4.4: Numerical performance of MCCT(0) on test problems for $x^{(0)} \approx 3\xi$

System	$x^{(0)}$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	$\|\|F(x^{(k+1)})\|\|$	iter	ACOC
$F(x_1, x_2)$	$(3.00, 3.00)^T$	2.9511e-49	2.6815e-195	4	5.8233
$G(x_1, x_2)$	$(2.60, 1.50)^T$	1.0829e-49	6.7038e-196	4	5.8689
$M(x_1, x_2)$	$(10.41, -7.41)^T$	2.3053e-41	1.0439e-113	4	5.9611
$N(x_1, x_2)$	$(3.00, 0.00)^T$	nc	nc	nc	nc
$O(x_1, x_2)$	$(-0.76, -1.16)^T$	7.0387e-71	7.9136e-174	5	-
$P(x_1, x_2, x_3)$	$(2.73, 1.98, 4.73)^T$	6.8830e-58	4.1779e-146	5	-
$Q(x_1, x_2, x_3, x_4)$	$(1.73, 1.73, 1.73, -0.87)^T$	1.2880e-33	8.1032e-180	4	-

Table 4.5: Numerical performance of MCCT(0) on test problems for $x^{(0)} > 10\xi$

System	$x^{(0)}$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	$\|\|F(x^{(k+1)})\|\|$	iter	ACOC
$F(x_1, x_2)$	$(11.00, 11.00)^T$	3.4914e-55	0	5	-
$G(x_1, x_2)$	$(9.53, 5.50)^T$	1.2350e-55	0	5	-
$M(x_1, x_2)$	$(38.18, -27.18)^T$	4.9654e-57	3.4199e-145	5	5.4814
$N(x_1, x_2)$	$(11.00, 0.00)^T$	nc	nc	nc	nc
$O(x_1, x_2)$	$(-2.79, -4.24)^T$	3.7780e-39	2.3868e-110	3	-
$P(x_1, x_2, x_3)$	$(10.01, 7.27, 17.33)^T$	1.6228e-61	2.8246e-153	14	-
$Q(x_1, x_2, x_3, x_4)$	$(6.35, 6.35, 6.35, -3.18)^T$	1.0412e-45	1.9467e-208	5	-

Results in Tables 4.4 and 4.5 evidence that MCCT(0) converges to the solution in six of the seven non-linear test systems, regardless of the initial estimates used. The ACOC does not stabilize its value in several cases but, when it does, it approaches to 6.

The analysis of MCCT(200) method is shown below. Numerical results, for $x^{(0)} \approx \xi$) and $x^{(0)} \approx 3\xi$), are presented in Tables 4.6 and 4.7.

Table 4.6: Numerical performance of MCCT(200) on test problems for $x^{(0)} \approx \xi$

System	$x^{(0)}$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	$\|\|F(x^{(k+1)})\|\|$	iter	ACOC
$F(x_1, x_2)$	$(0.90, 0.90)^T$	1.9038e-29	4.6447e-116	3	6.0626
$G(x_1, x_2)$	$(0.80, 0.40)^T$	5.1091e-67	1.9467e-208	4	-
$M(x_1, x_2)$	$(3.40, -2.40)^T$	1.6761e-43	2.8365e-119	3	5.9400
$N(x_1, x_2)$	$(0.90, 0.10)^T$	2.0202e-48	7.7869e-208	4	-
$O(x_1, x_2)$	$(-0.20, -0.30)^T$	3.8365e-85	6.7625e-202	4	-
$P(x_1, x_2, x_3)$	$(0.80, 0.60, 1.50)^T$	3.2604e-41	5.4472e-112	4	-
$Q(x_1, x_2, x_3, x_4)$	$(0.50, 0.50, 0.50, -0.20)^T$	8.7884e-87	1.9467e-208	4	5.6358

Table 4.7: Numerical performance of MCCT(200) on test problems for $x^{(0)} \approx 3\xi$

System	$x^{(0)}$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	$\|\|F(x^{(k+1)})\|\|$	iter	ACOC
$F(x_1, x_2)$	$(3.00, 3.00)^T$	9.6219e-49	3.0304e-193	5	5.7669
$G(x_1, x_2)$	$(2.60, 1.50)^T$	3.4103e-49	7.5761e-194	5	5.8239
$M(x_1, x_2)$	$(10.41, -7.41)^T$	5.0005e-75	1.0922e-182	9	-
$N(x_1, x_2)$	$(3.00, 0.00)^T$	nc	nc	nc	nc
$O(x_1, x_2)$	$(-0.76, -1.16)^T$	nc	nc	nc	nc
$P(x_1, x_2, x_3)$	$(2.73, 1.98, 4.73)^T$	nc	nc	nc	nc
$Q(x_1, x_2, x_3, x_4)$	$(1.73, 1.73, 1.73, -0.87)^T$	5.9657e-32	4.5520e-179	5	5.8489

MCCT(200) presents convergence problems for $x^{(0)} \approx 3\xi$, since it does not converge to the solution in three of the seven cases, establishing a dependence on the initial estimate and the nonlinear test system used. In addition, the number of iterations increases for the systems in which the solution is reached, with respect to the MCCT(0) method for the same conditions.

Consequently, we conclude that the method for $\alpha = 0$ is robust, converges to the solution with few iterations and for any seed and system used. Nevertheless, the method for $\alpha = 200$ is unstable, since it does not tend to the solution according to the seed and the system used. Observe that both methods converge to the solution with order 6. Therefore, the theoretical results from the dynamical behavior and convergence analysis of the MCCT(α) family are verified.

4.5 Conclusions

The designed class MCCT(α) for solving systems of nonlinear equations, proving to be a highly efficient class with sixth order of convergence.

The convergence of the class of iterative schemes, the stability using a real multidimensional discrete dynamical system and the numerical throughput performance have been analyzed through several test problems.

The stable members of the MCCT(α) family exhibited outstanding numerical performance. The method for $\alpha = 0$ proved to be robust (stable), according to the real dynamics analysis performed. The method for $\alpha = 200$ has been shown to be unstable, chaotic and may not converge to the searched solution. The theoretical order of convergence is verified by ACOC, which is close to 6. Numerical experiments confirm the theoretical results.

Future lines of research consist of introducing a new step with similar characteristics to increase the order of convergence without considerably penalizing its computational cost and analyzing its effect on the stability of the resulting family of methods.

Chapter 5

General discussion of the findings

The research conducted on new iterative schemes for solving non-linear equations and systems has yielded significant results, validated through three key publications. This discussion focuses on the performance, stability, and applicability of these new methods.

The first paper, *"Achieving Optimal Order in a Novel Family of Numerical Methods: Insights from Convergence and Dynamical Analysis Results"*, introduces a novel family of two-step iterative methods. This family is derived from a damped Newton scheme and includes an additional Newton step with a weight function and a "frozen" derivative. Initially, a four-parameter class with first-order convergence is developed, which, by fixing one parameter, becomes a single-parameter family with third-order convergence. The convergence and stability properties were thoroughly investigated, identifying an optimal fourth-order member according to the Kung-Traub's conjecture. The analysis revealed the complexity of the family and allowed the identification of members with exceptional stability. Numerical tests validated the efficiency and reliability of the proposed methods, demonstrating their capability to converge to solutions even from distant initial estimates.

The second paper, *"Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations"*, presents a new parametric family of three-step iterative methods for solving non-linear equations. Initially, a three-parameter fourth-order family is designed, which, by fixing one of its parameters, accelerates its convergence to a single-parameter sixth-order family. The convergence, complex dynamics, and numerical behavior of this latter family were studied extensively. From the dynamic analysis, members with particularly stable behavior, suitable for solving practical problems, were identified. Several numerical tests illustrated the efficiency and stability of the proposed family, confirming its robustness in practical applications.

The third paper, *"Performance of a New Sixth-Order Class of Iterative Schemes for Solving Non-Linear Systems of Equations"*, extends the sixth-order family to systems of equations. Based on the Ostrowski scheme, the class is constructed by adding a Newton step with a Jacobian matrix from the previous step and using a divided difference operator, resulting in a three-parameter scheme with fourth-order convergence. By adjusting two parameters, the convergence order is accelerated to six, forming a single-parameter family. Dynamical and numerical analyses confirmed the convergence, stability, and applicability of this extended family for solving large-scale problems. Numerical tests on several benchmark problems demonstrated the method's efficiency and stability, validating the theoretical findings.

Therefore, the designed iterative schemes have shown remarkable improvements in convergence, stability, and computational efficiency. The robust performance of the proposed methods, particularly the stability observed in dynamic analysis and validated through extensive numerical tests, signifies a substantial contribution to the field of numerical analysis. Future research should focus on optimizing the free parameters within these families to maximize their performance across a broader range of non-linear problems. Additionally, integrating these methods with machine learning techniques and exploring parallel and distributed computing environments could further enhance their applicability and efficiency in solving complex large-scale systems.

Chapter 6

Conclusions and future research directions

"Everything should be made as simple as possible, but not simpler."
Albert Einstein

6.1 Conclusions

The research concentrated on developing and analyzing new families of iterative methods for solving non-linear equations and systems, yielding substantial results that were validated through publications in high-impact journals.

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 non-linear 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 third-order family. Numerical results confirm the robustness of the MCCTU(2) method for initial estimates close to the root, with very low errors and convergence within 3 or 4 iterations. Even as initial estimates move further away, the method continues to perform solidly, demonstrating 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. The MCCTU(1) member stood out for its optimal and stable performance, especially in complex non-linear equations where it outperformed several well-established methods. The theoretical convergence order of the MCCTU(α) family was validated by the approximate computational order of convergence (ACOC), with most cases stabilizing close to a third-order convergence. However, the analysis also revealed significant instability in certain members with α values outside the stability surface, emphasizing the importance of selecting appropriate parameter values to ensure reliable performance.

The development of the new family of iterative methods, CMT(α, β, γ), based on the Ostrowski scheme, has demonstrated significant improvements in solving non-linear equations. By incorporating a Newton step with a "frozen" derivative and utilizing a divided difference operator, the family achieved a fourth-order convergence, similar to the Ostrowski method. By parameterizing β and γ as functions of α, the convergence order was accelerated to six, resulting in the uniparametric CMT(α) family. Numerical results confirmed the high efficiency and low error rates of this family, particularly for stable members ($\alpha = -1, 0, 1$), which consistently converged to the solution with fewer iterations and lower CPU time compared to other sixth-order methods of the literature.

The dynamical study of the CMT(α) family revealed critical insights into the stability and performance of its members. By constructing parameter spaces for the free critical points and generating dynamical planes, stable and unstable methods were identified. Stable members for $\alpha = -1, 0, 1$ showed excellent numerical performance, with low errors, rapid convergence, and the lowest CPU time across various initial estimates and non-linear test functions. Conversely, unstable members for $\alpha = -300, 200, 400$ exhibited chaotic behavior, higher CPU times, and a tendency to fail convergence, verifying the theoretical predictions from the convergence and dynamical analysis. These findings underscore the importance of selecting appropriate parameter values to ensure the reliability and efficiency of iterative methods for solving non-linear equations.

The development of the MCCT(α) family, a highly efficient sixth-order iterative method for solving non-linear systems of equations, has demonstrated substantial improvements in numerical performance. Built from Ostrowski's scheme with the addition of a Newton step using a "frozen" Jacobian matrix and a divided difference operator, this family offers a three-step iterative process with one arbitrary parameter. The convergence analysis and numerical experiments have confirmed the theoretical order of convergence close to six, validating the method's effectiveness. Notably, the method for $\alpha = 0$ proved to be robust and stable, consistently converging to solutions across various test problems.

The stability analysis of the MCCT(α) family through a real multidimensional discrete dynamical system has highlighted significant differences in performance based on the parameter α. Stable members, particularly the method for $\alpha = 0$, demonstrated excellent numerical performance, low errors, and reliable convergence. In contrast, the method for $\alpha = 200$ exhibited chaotic behavior and instability, failing to converge to solutions for the given initial estimates and non-linear test systems. These findings underscore the importance of parameter selection within the MCCT(α) family to ensure robust and efficient problem-solving capabilities for non-linear systems of equations.

6.2 Future research directions

Based on the promising results obtained from the development and analysis of the MCCT(α) family for solving non-linear systems of equations, several future research directions are proposed. These avenues aim to further enhance the robustness, efficiency, and applicability of the iterative methods, ensuring their practical utility across a broad spectrum of complex problems. The following suggestions outline potential areas for extending and refining the current methodologies.

Investigate the applicability of the MCCT(α) family of methods to solve non-stationary non-linear systems, where the system parameters or equations change over time. This could provide valuable insights into the dynamic behavior and stability of the methods under varying conditions.

Develop adaptive algorithms that dynamically adjust the parameter α during iterations to optimize convergence speed and stability. This approach could enhance the robustness and efficiency of the MCCT(α) methods across a broader range of non-linear systems.

Explore the implementation of the MCCT(α) family in parallel and distributed computing environments to handle large-scale non-linear systems more effectively. Leveraging high-performance computing resources can significantly reduce computation times and enable the solution of more complex problems.

Apply the MCCT(α) family to non-linear systems in various fields such as biology, economics, and physics. Evaluating the performance and stability of these methods in different contexts could reveal new opportunities for interdisciplinary applications.

Investigate the integration of the MCCT(α) methods with other numerical techniques or machine learning approaches to create hybrid methods. These hybrids could potentially combine the strengths of different approaches, leading to even more powerful and versatile solution strategies for non-linear systems.

Tailor the MCCT(α) methods to specific types of non-linear systems, such as stiff systems or systems with multiple solutions. Developing specialized versions of these methods for particular problem classes could further enhance their applicability and effectiveness.

Enhance the convergence order of the MCCT(α) family by employing weight functions or modifying one of the iterative scheme steps. This improvement aims to achieve convergence to the solution with a reduced number of iterations.

Bibliography

[1] George B. Arfken, Hans J. Weber, and Frank E. Harris. *Mathematical Methods for Physicists*. Seventh Edition. Boston: Academic Press, 2013. ISBN: 978-0-12-384654-9.

[2] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, 2010.

[3] William E. Boyce and Richard C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. Wiley, 2017.

[4] John E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.

[5] James D. Murray. *Mathematical Biology I: An Introduction*. Springer, 2002.

[6] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[7] Ramamurti Shankar. *Principles of Quantum Mechanics*. Springer, 2011.

[8] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[9] James M. Ortega and Werner C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, 2000.

[10] Peter Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer, 2011.

[11] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[12] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2013.

[13] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM, 1997.

[14] James Keener and James Sneyd. *Mathematical Physiology*. Springer, 2009.

[15] Nobuo Yamashita, Hiroshi Yabe, and Masao Fukushima. "A globally and superlinearly convergent newton method for strongly monotone variational inequalities". *Mathematical Programming* 91.1 (2001), pp. 51–64.

[16] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics* 378 (2019), pp. 686–707.

[17] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. "Machine learning for fluid mechanics". *Annual Review of Fluid Mechanics* 52 (2020), pp. 477–508.

[18] Jack Dongarra and Aad Van Der Steen. "High-performance computing systems: Status and outlook". *Acta Numerica* 21 (2012), pp. 379–474. DOI: 10.1017/S0962492912000050.

[19] Roy Danchick. "Gauss meets Newton again: How to make Gauss orbit determination from two position vectors more efficient and robust with Newton–Raphson iterations". *Applied Mathematics and Computation* 195.2 (2008), pp. 364–375. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2007.03.053.

[20] Marcos Tostado-Véliz, Salah Kamel, Francisco Jurado, and Francisco J. Ruiz-Rodriguez. "On the Applicability of Two Families of Cubic Techniques for Power Flow Analysis". *Energies* 14.14 (2021). ISSN: 1996-1073. DOI: 10.3390/en14144108.

[21] Víctor Arroyo, Alicia Cordero, and Juan R. Torregrosa. "Approximation of artificial satellites' preliminary orbits: The efficiency challenge". *Mathematical and Computer Modelling* 54.7 (2011). Mathematical models of addictive behaviour, medicine & engineering, pp. 1802–1807. ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2010.11.063.

[22] Joseph F. Traub. *Iterative Methods for the Solution of Equations.* Englewood Cliffs, NJ, USA: Prentice-Hall, 1964.

[23] Miodrag S. Petković, Beny Neta, Ljiljana D. Petković, and Jovana Džunić. *Multipoint Methods for Solving Nonlinear Equations*. Boston, USA: Academic Press, 2013.

[24] Sergio Amat and Sonia Busquier. *Advances in Iterative Methods for Nonlinear Equations*. Switzerland: Springer, 2017.

[25] Alexander M. Ostrowski. *Solution of equations in Euclidean and Banach spaces*. New York, USA: Academic Press, 1973.

[26] Ahmet Yaşar Özban and Bahar Kaya. "A new family of optimal fourth-order iterative methods for nonlinear equations". *Results in Control and Optimization* 8 (2022), pp. 1–11. ISSN: 2666-7207. DOI: `https://doi.org/10.1016/j.rico.2022.100157`.

[27] George Adomian. *Solving Frontier Problem of Physics: The Decomposition Method*. Kluwer Academic Publishers, Dordrecht, 1994.

[28] Miodrag S. Petković, Beny Neta, Ljiljana D. Petković, and Jovana Džunić. "Multipoint methods for solving nonlinear equations: A survey". *Applied Mathematics and Computation* 226 (2014), pp. 635–660. ISSN: 0096-3003. DOI: `https://doi.org/10.1016/j.amc.2013.10.072`.

[29] Changbum Chun. "Some fourth-order iterative methods for solving nonlinear equations". *Applied Mathematics and Computation* 195 (2008), pp. 454–459.

[30] Alexander M. Ostrowski. *Solution of Equations and Systems of Equations*. New York, USA: Academic Press, 1960.

[31] Hsiang-Tsung Kung and Joseph F. Traub. "Optimal Order of One-Point and Multipoint Iteration". *Journal of the Association for Computing Machinery* 21.4 (1974), pp. 643–651.

[32] S. Weerakoon and T.G.I. Fernando. "A variant of Newton's method with accelerated third-order convergence". *Applied Mathematics Letters* 13.8 (2000), pp. 87 –93. ISSN: 0893-9659. DOI: `https://doi.org/10.1016/S0893-9659(00)00100-2`.

[33] Alicia Cordero and Juan R. Torregrosa. "Variants of Newton's Method using fifth-order quadrature formulas". *Applied Mathematics and Computation* 190.1 (2007), pp. 686 –698. ISSN: 0096-3003. DOI: `https://doi.org/10.1016/j.amc.2007.01.062`.

[34] Munish Kansal, Alicia Cordero, Sonia Bhalla, and Juan R. Torregrosa. "New fourth- and sixth-order classes of iterative methods for solving systems of nonlinear equations and their stability analysis". *Numerical Algorithms* 87 (2021), pp. 1017–1060.

[35] Alicia Cordero, Fazlollah Soleymani, and Juan R. Torregrosa. "Dynamical analysis of iterative methods for nonlinear systems or how to deal with the dimension?" *Applied Mathematics and Computation* 244 (2014), pp. 398–412.

[36] Alicia Cordero, Marlon Moscoso-Martínez, and Juan R. Torregrosa. "Chaos and Stability in a New Iterative Family for Solving Nonlinear Equations". *Algorithms* 14.4 (2021), pp. 1–24.

[37] Akhlaq Husain, Manikyala Navaneeth Nanda, Movva Sitaram Chowdary, and Mohammad Sajid. "Fractals: An Eclectic Survey, Part-I". *Fractal and Fractional* 6.2 (2022). ISSN: 2504-3110. DOI: 10.3390/fractalfract6020089.

[38] Akhlaq Husain, Manikyala Navaneeth Nanda, Movva Sitaram Chowdary, and Mohammad Sajid. "Fractals: An Eclectic Survey, Part II". *Fractal and Fractional* 6.7 (2022). ISSN: 2504-3110. DOI: 10.3390/fractalfract6070379.

[39] Juan Luis Varona. "Graphic and numerical comparison between iterative methods". *The Mathematical Intelligencer* 24 (Mar. 2002), pp. 37–46. DOI: 10.1007/BF03025310.

[40] Sergio Amat, Sonia Busquier, and Sergio Plaza. "Review of some iterative root–finding methods from a dynamical point of view". *SCIENTIA Series A: Mathematical Sciences* 10 (Jan. 2004), pp. 3–35.

[41] Beny Neta, Changbum Chun, and Melvin Scott. "Basins of attraction for optimal eighth order methods to find simple roots of nonlinear equations". *Applied Mathematics and Computation* 227 (2014), pp. 567–592. DOI: 10.1016/j.amc.2013.11.017.

[42] Alicia Cordero, Javier García-Maimó, Juan R. Torregrosa, Maria P. Vassileva, and Pura Vindel. "Chaos in King's iterative family". *Applied Mathematics Letters* 26.8 (2013), pp. 842–848. ISSN: 0893-9659. DOI: https://doi.org/10.1016/j.aml.2013.03.012.

[43] Ángel A. Magreñán and Ioannis Argyros. *A Contemporary Study of Iterative Methods*. Academic Press, 2018.

[44] Young Hee Geum and Young Ik Kim. "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 and Continuous Dynamical Systems–B* 25.8 (2020), pp. 3087–3109. DOI: 10.3934/dcdsb.2020052.

[45] Alicia Cordero, Juan R. Torregrosa, and Pura Vindel. "Dynamics of a family of Chebyshev-Halley type method". *Applied Mathematics and Computation* 219 (July 2012). DOI: 10.1016/j.amc.2013.02.042.

[46] Ángel A. Magreñán. "Different anomalies in a Jarratt family of iterative root-finding methods". *Applied Mathematics and Computation* 233 (2014), pp. 29–38.

[47] Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley Publishing Company, 1989.

[48] Alan F. Beardon. *Iteration of rational functions*. Graduate Texts in Mathematics. Springer-Verlag New York, USA, 1991.

[49] Pierre J. L. Fatou. "Sur les équations fonctionelles". *Bull. Soc. Mat. Fr.* 47 (1919), pp. 161–271.

[50] Gaston M. Julia. "Mémoire sur l'iteration des fonctions rationnelles". *Mat. Pur. Appl.* 8 (1918), pp. 47–245.

[51] Melvin Scott, Beny Neta, and Changbum Chun. "Basin attractors for various methods". *Applied Mathematics and Computation* 218 (Nov. 2011), 2584–2599. DOI: 10.1016/j.amc.2011.07.076.

[52] Paul Blanchard. "Complex analytic dynamics on the Riemann sphere". *Bull. Amer. Math. Soc. (N.S.)* 11.1 (July 1984), pp. 85–141.

[53] José L. Hueso, Eulalia Martínez, and Carles Teruel. "Multipoint efficient iterative methods and the dynamics of Ostrowski's method". *International Journal of Computer Mathematics* 96.9 (2019), pp. 1687–1701. DOI: 10.1080/00207160.2015.1080354.

[54] Richard F. King. "A Family of Fourth Order Methods for Nonlinear Equations". *SIAM Journal on Numerical Analysis* 10.5 (1973), pp. 876–879. DOI: 10.1137/0710072.

[55] P. Jarratt. "Some fourth order multipoint iterative methods for solving equations". *Mathematics of Computation* 20.95 (1966), pp. 434–437. DOI: 10.1090/S0025-5718-66-99924-8.

[56] Changbum Chun. "Construction of Newton-like iteration methods for solving nonlinear equations". *Numerische Mathematik* 104.95 (2006), pp. 297–315. ISSN: 0945-3245. DOI: 10.1007/s00211-006-0025-2.

[57] Amit Kumar Maheshwari. "A fourth order iterative method for solving nonlinear equations". *Applied Mathematics and Computation* 211.2 (2009), pp. 383–391. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2009.01.047.

[58] R. Behl, P. Maroju, and S.S. Motsa. "A family of second derivative free fourth order continuation method for solving nonlinear equations". *Journal of Computational and Applied Mathematics* 318 (2017). Computational and Mathematical Methods in Science and Engineering CMMSE-2015, pp. 38–46. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2016.12.008.

[59] Changbum Chun, Mi Young Lee, Beny Neta, and Jovana Džunić. "On optimal fourth-order iterative methods free from second derivative and their dynamics". *Applied Mathematics*

and Computation 218.11 (2012), pp. 6427–6438. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2011.12.013.

[60] Santiago Artidiello, Francisco I. Chicharro, Alicia Cordero, and Juan R. Torregrosa. "Local convergence and dynamical analysis of a new family of optimal fourth-order iterative methods". *International Journal of Computer Mathematics* 90.10 (2013), pp. 2049–2060. DOI: 10.1080/00207160.2012.748900.

[61] Behzad Ghanbari. "A new general fourth-order family of methods for finding simple roots of nonlinear equations". *Journal of King Saud University - Science* 23.4 (2011), pp. 395–398. ISSN: 1018-3647. DOI: https://doi.org/10.1016/j.jksus.2010.07.018.

[62] Jisheng Kou, Yitian Li, and Xiuhua Wang. "A composite fourth-order iterative method for solving non-linear equations". *Applied Mathematics and Computation* 184.2 (2007), pp. 471–475. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2006.05.181.

[63] Beny Neta. *Numerical methods for the solution of equations.* California: Net-A-Sof, 1983.

[64] Sergio Amat and Sonia Busquier. *Advances in Iterative Methods for Nonlinear Equations.* Switzerland: Springer, 2017.

[65] Santiago Artidiello, Alicia Cordero, Juan R. Torregrosa, and María P. Vassileva. "Design and multidimensional extension of iterative methods for solving nonlinear problems". *Applied Mathematics and Computation* 293 (Jan. 2017), pp. 194–203. DOI: 10.1016/j.amc.2016.08.034.

[66] Brian R. Hunt and Edward Ott. "Defining chaos". *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.9 (2015). DOI: 10.1063/1.4922973.

[67] James M. Ortega and Werner C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.

[68] Abdullah K. H. Alzahrani, Ramandeep Behl, and Ali S. Alshomrani. "Some higher-order iteration functions for solving nonlinear models". *Applied Mathematics and Computation* 334 (Oct. 2018), pp. 80–93. DOI: 10.1016/j.amc.2018.03.120.

[69] Changbum Chun and YoonMee Ham. "Some sixth-order variants of Ostrowski root-finding methods". *Applied Mathematics and Computation* 193 (Nov. 2007), pp. 389–394. DOI: 10.1016/j.amc.2007.03.074.

[70] Sergio Amat, M. A. Hernández, and N. Romero. "Semilocal convergence of a sixth order iterative method for quadratic equations". *Applied Numerical Mathematics* 62 (July 2012), 833–841. DOI: 10.1016/j.apnum.2012.03.001.

[71] Samad Noeiaghdam, Denis Sidorov, Alyona Zamyshlyaeva, Aleksandr Tynda, and Aliona Dreglea. "A valid dynamical control on the reverse osmosis system using the CESTAC method". *Mathematics* 9.48 (2021). DOI: 10.3390/math9010048.

[72] Ramandeep Behl, Sonia Bhalla, Ángel A. Magreñán, and Sanjeev Kumar. "An efficient high order iterative scheme for large nonlinear systems with dynamics". *Journal of Computational and Applied Mathematics* 404 (2022), p. 113249. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2020.113249.

[73] Obadah S. Solaiman and Ishak Hashim. "An iterative scheme of arbitrary odd order and its basins of attraction for nonlinear systems". *Comput. Mater. Contin.* 66 (2020), pp. 1427–1444.

[74] Rajni Sharma, Janak Sharma, and Nitin Kalra. "A Modified Newton–Özban Composition for Solving Nonlinear Systems". *International Journal of Computational Methods* 17 (2019), p. 1950047. DOI: 10.1142/S0219876219500476.

[75] Saima Yaseen and Fiza Zafar. "A new sixth-order Jarratt-type iterative method for systems of nonlinear equations". *Arabian Journal of Mathematics* 11 (2022), pp. 585–599.

[76] Harmandeep Singh, Janak R. Sharma, and Sunil Kumar. "A simple yet efficient two-step fifth-order weighted-Newton method for nonlinear models". *Numerical Algorithms* (2022). DOI: https://doi.org/10.1007/s11075-022-01412-w.

[77] R. H. Al-Obaidi and M. T. Darvishi. "Constructing a Class of Frozen Jacobian Multi-Step Iterative Solvers for Systems of Nonlinear Equations". *Mathematics* 10.16 (2022), p. 2952.

[78] M. K. Singh and A. K. Singh. "Study of Frozen-Type Newton-Like Method in a Banach Space with Dynamics". *Ukrainian Mathematical Journal* 74 (2022), pp. 266–288.

[79] Charles Hermite. "Sur la formule d'interpolation de Lagrange". *Reine Angew. Math.* 84 (1878), pp. 70–79.

[80] José L. Hueso, Eulalia Martínez, and Carles Teruel. "Convergence, effiency and dinamics of new fourth and sixth order families of iterative methods for nonlinear systems". *Comput. Appl. Math.* 275 (2015), pp. 412–420.