Turnitin Originality Report

Processed on: 15-Feb-2025 17:59 IST
ID: 2589212705
Word Count: 60592
Submitted: 1

THesis-Himani By Munish Kansal

Similarity Index	Similarity by Source	
42%	Internet Sources:	21%
	Publications:	40%
	Student Papers:	4%

9% match (Himani Sharma, Munish Kansal. "Stability analysis and dynamical behavior of optimal mean-based iterative methods", Journal of Mathematical Chemistry, 2024)
Himani Sharma, Munish Kansal. "Stability analysis and dynamical behavior of optimal mean-based iterative methods", Journal of Mathematical Chemistry, 2024

6% match (Munish Kansal, Himani Sharma. "Analysis of optimal iterative methods from a dynamical point of view by studying their stability properties", Journal of Mathematical Chemistry, 2023)
Munish Kansal, Himani Sharma. "Analysis of optimal iterative methods from a dynamical point of view by studying their stability properties", Journal of Mathematical Chemistry, 2023

5% match (Himani Sharma, Munish Kansal. "A modified Chebyshev–Halley-type iterative family with memory for solving nonlinear equations and its stability analysis", Mathematical Methods in the Applied Sciences, 2023)
Himani Sharma, Munish Kansal. "A modified Chebyshev–Halley-type iterative family with memory for solving nonlinear equations and its stability analysis", Mathematical Methods in the Applied Sciences, 2023

3% match (Internet from 14-Jul-2023)
https://WWW.MDPI.COM/2297-8747/27/6/97

3% match (Internet from 27-Mar-2024)
https://gredos.usal.es/bitstream/handle/10366/156078/Sharma,%20Behl,%20Kansal,%20Ramos-2024-JCAM%20A%20robust%20iterative%20family%20for%20multiple%20roots%20of%20nonlinear%20equations.pdf.tx
sequence=4

3% match (Himani Sharma, Munish Kansal, Ramandeep Behl. "An Efficient Two-Step Iterative Family Adaptive with Memory for Solving Nonlinear Equations and Their Applications", Mathematical and Computational Applications, 2022)
Himani Sharma, Munish Kansal, Ramandeep Behl. "An Efficient Two-Step Iterative Family Adaptive with Memory for Solving Nonlinear Equations and Their Applications", Mathematical and Computational Applications, 2022

3% match (Himani Sharma, Ramandeep Behl, Munish Kansal, Higinio Ramos. "A robust iterative family for multiple roots of nonlinear equations: Enhancing accuracy and handling critical points", Journal of Computational and Applied Mathematics, 2024)
Himani Sharma, Ramandeep Behl, Munish Kansal, Higinio Ramos. "A robust iterative family for multiple roots of nonlinear equations: Enhancing accuracy and handling critical points", Journal of Computational and Applied Mathematics, 2024

2% match (Internet from 14-Jul-2023)
https://WWW.MDPI.COM/2297-8747/28/2/48

2% match (Himani Sharma, Munish Kansal, Ramandeep Behl. "An Efficient Optimal Derivative-Free Fourth-Order Method and Its Memory Variant for Non-Linear Models and Their Dynamics", Mathematical and Computational Applications, 2023)
Himani Sharma, Munish Kansal, Ramandeep Behl. "An Efficient Optimal Derivative-Free Fourth-Order Method and Its Memory Variant for Non-Linear Models and Their Dynamics", Mathematical and Computational Applications, 2023

1% match (Internet from 21-Nov-2022)
http://uilis.unsyiah.ac.id/oer/files/original/d982fe4e0f518c32c9f254dfe7bbaa11.pdf

1% match (Internet from 25-Nov-2022)

https://downloads.hindawi.com/journals/specialissues/127152.pdf

1% match (student papers from 21-Aug-2013)
Submitted to Panjab University on 2013-08-21

< 1% match (Internet from 26-Feb-2024)
https://gredos.usal.es/bitstream/handle/10366/156078/Sharma%2c%20Behl%2c%20Kansal%2c%20Ramos-2024-JCAM%20A%20robust%20iterative%20family%20for%20multiple%20roots%20of%20nonlinear%20equations.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 05-Sep-2024)
https://ebin.pub/symmetry-with-operator-theory-and-equations-303921666x-9783039216666.html

< 1% match (Internet from 02-Dec-2023)
https://ebin.pub/iterative-methods-for-solving-nonlinear-equations-and-systems-3039219405-9783039219407.html

< 1% match (Internet from 11-Aug-2024)
https://ebin.pub/contemporary-algorithms-theory-and-applications-volume-iii-3-9781685079949-9798886971095-9798886979060-9798886974256-9798886974614-9798886979855.html

< 1% match (Paula Triguero Navarro. "High Performance Multidimensional Iterative Processes for Solving Nonlinear Equations", Universitat Politecnica de Valencia, 2023)
Paula Triguero Navarro. "High Performance Multidimensional Iterative Processes for Solving Nonlinear Equations", Universitat Politecnica de Valencia, 2023

< 1% match (Fabricio Alfredo Cevallos Alarcón. "Métodos iterativos para la resolución de problemas aplicados transformados a sistemas no lineales", Universitat Politecnica de Valencia, 2023)
Fabricio Alfredo Cevallos Alarcón. "Métodos iterativos para la resolución de problemas aplicados transformados a sistemas no lineales", Universitat Politecnica de Valencia, 2023

< 1% match (Petković, Miodrag S., Beny Neta, Ljiljana D. Petković, and Jovana Džunić. "Two-point methods", Multipoint Methods, 2013.)
Petković, Miodrag S., Beny Neta, Ljiljana D. Petković, and Jovana Džunić. "Two-point methods", Multipoint Methods, 2013.

< 1% match (student papers from 17-Apr-2012)
Submitted to Higher Education Commission Pakistan on 2012-04-17

< 1% match (student papers from 20-May-2013)
Submitted to Higher Education Commission Pakistan on 2013-05-20

< 1% match (student papers from 19-Mar-2012)
Submitted to Higher Education Commission Pakistan on 2012-03-19

< 1% match (student papers from 12-Aug-2011)
Submitted to Higher Education Commission Pakistan on 2011-08-12

< 1% match ("Advances in Iterative Methods for Nonlinear Equations", Springer Science and Business Media LLC, 2016)
"Advances in Iterative Methods for Nonlinear Equations", Springer Science and Business Media LLC, 2016

< 1% match (student papers from 15-Jan-2024)
Submitted to Universitas Dian Nuswantoro on 2024-01-15

< 1% match (Tugal Zhanlav, Ochbadrakh Chuluunbaatar. "New Developments of Newton-Type Iterations for Solving Nonlinear Problems", Springer Science and Business Media LLC, 2024)
Tugal Zhanlav, Ochbadrakh Chuluunbaatar. "New Developments of Newton-Type Iterations for Solving Nonlinear Problems", Springer Science and Business Media LLC, 2024

< 1% match ("Finite Difference Methods. Theory and Applications", Springer Science and Business Media LLC, 2019)
"Finite Difference Methods. Theory and Applications", Springer Science and Business Media LLC, 2019

< 1% match (D. K. Gupta. "An improved class of regula falsi methods of third order for solving nonlinear equations in R", Journal of Applied Mathematics and Computing, 04/01/2009)
D. K. Gupta. "An improved class of regula falsi methods of third order for solving nonlinear equations in R", Journal of Applied Mathematics and Computing, 04/01/2009

< 1% match (publications)
Andrei D. Polyanin, Alexander V. Manzhirov. "Handbook of Mathematics for Engineers and Scientists", Chapman and Hall/CRC, 2019

< 1% match ()
Malik, Zaka Ullah. "Numerical iterative methods for nonlinear problems.", Italy, 2015

< 1% match (publications)
Abdelwahab Kharab, Abdelwahab Kharab, Ronald B. Guenther. "An Introduction to Numerical Methods - A MATLAB Approach, Third Edition", Chapman and Hall/CRC, 2019

< 1% match (publications)
Steven G. Krantz. "Dictionary of Algebra, Arithmetic, and Trigonometry", CRC Press, 2019

< 1% match (Internet from 15-Jul-2024)
https://docslib.org/doc/6197892/symmetry-with-operator-theory-and-equations

< 1% match (Chein-Shan Liu, Tsung-Lin Lee. "A New Family of Fourth-Order Optimal Iterative Schemes and Remark on Kung and Traub’s Conjecture", Journal of Mathematics, 2021)
Chein-Shan Liu, Tsung-Lin Lee. "A New Family of Fourth-Order Optimal Iterative Schemes and Remark on Kung and Traub’s Conjecture", Journal of Mathematics, 2021

< 1% match (Xiaofeng Wang. "A family of Newton-type iterative methods using some special self-accelerating parameters", International Journal of Computer Mathematics, 2017)
Xiaofeng Wang. "A family of Newton-type iterative methods using some special self-accelerating parameters", International Journal of Computer Mathematics, 2017

< 1% match (Sepehri, Amir. "Non-Parametric Goodness-Of-Fit Testing and Applications.", Stanford University, 2021)
Sepehri, Amir. "Non-Parametric Goodness-Of-Fit Testing and Applications.", Stanford University, 2021

< 1% match (publications)
Lawrence Turyn. "Advanced Engineering Mathematics", CRC Press, 2019

< 1% match (Chein-Shan Liu, Essam R. El-Zahar, Chih-Wen Chang. "A Two-Dimensional Variant of Newton’s Method and a Three-Point Hermite Interpolation: Fourth- and Eighth-Order Optimal Iterative Schemes", Mathematics, 2023)
Chein-Shan Liu, Essam R. El-Zahar, Chih-Wen Chang. "A Two-Dimensional Variant of Newton’s Method and a Three-Point Hermite Interpolation: Fourth- and Eighth-Order Optimal Iterative Schemes", Mathematics, 2023

< 1% match (Moin-ud-Din Junjua, Fiza Zafar, Nusrat Yasmin. "Optimal Derivative-Free Root Finding Methods Based on Inverse Interpolation", Mathematics, 2019)
Moin-ud-Din Junjua, Fiza Zafar, Nusrat Yasmin. "Optimal Derivative-Free Root Finding Methods Based on Inverse Interpolation", Mathematics, 2019

< 1% match (publications)
Christopher Apelian, Steve Surace. "Real and Complex Analysis", Chapman and Hall/CRC, 2019

< 1% match (publications)
James K. Peterson. "Basic Analysis I - Functions of a Real Variable", CRC Press, 2020

< 1% match (publications)
Nicholas Loehr. "Advanced Linear Algebra", Chapman and Hall/CRC, 2019

< 1% match (Цветковић, Марија С.. "Fixed Point Theorems of Perov Type", University of Nis (Serbia), 2024)
Цветковић, Марија С.. "Fixed Point Theorems of Perov Type", University of Nis (Serbia), 2024

< 1% match (publications)
Justin Solomon. "Numerical Algorithms - Methods for Computer Vision, Machine Learning, and Graphics", A K Peters/CRC Press, 2019

< 1% match (publications)
Elias Camouzis, G. Ladas. "Dynamics of Third-Order Rational Difference Equations with Open Problems and Conjectures", Chapman and Hall/CRC, 2019

< 1% match (publications)
Waldemar Dos Passos. "Numerical Methods, Algorithms and Tools in C#", CRC Press, 2019

< 1% match (Dixit, Rishabh. "Gradient Based Methods for Nonconvex Optimization: A Dynamical Systems Viewpoint.", Rutgers The State University of New Jersey, School of Graduate Studies, 2024)
Dixit, Rishabh. "Gradient Based Methods for Nonconvex Optimization: A Dynamical Systems Viewpoint.", Rutgers The State University of New Jersey, School of Graduate Studies, 2024

< 1% match (Calado, Ricardo Manuel Santana. "Capacitação dos Alunos de 9º ano do Agrupamento de Escolas de Ponte de sor no Algoritmo de Suporte Básico de Vida e Avaliar a Aquisição de Competências na Atuação em Paragem Cardiorrespiratória", Universidade de Evora (Portugal), 2024)
Calado, Ricardo Manuel Santana. "Capacitação dos Alunos de 9º ano do Agrupamento de Escolas de Ponte de sor no Algoritmo de Suporte Básico de Vida e Avaliar a Aquisição de Competências na Atuação em Paragem Cardiorrespiratória", Universidade de Evora (Portugal), 2024

< 1% match (publications)
John Bird, John Bird. "Higher Engineering Mathematics", Routledge, 2019

< 1% match (publications)
Mustafa R.S. Kulenovic, G. Ladas. "Dynamics of Second Order Rational Difference Equations - With Open Problems and Conjectures", Chapman and Hall/CRC, 2019

< 1% match (Ayyildiz, Ezgi. "Gaussian Graphical Approaches in Estimation of Biological Systems.", Middle East Technical University (Turkey), 2024)
Ayyildiz, Ezgi. "Gaussian Graphical Approaches in Estimation of Biological Systems.", Middle East Technical University (Turkey), 2024

< 1% match (Kúkelová, Zuzana. "Algebraic Methods in Computer Vision", Czech Technical University, 2022)
Kúkelová, Zuzana. "Algebraic Methods in Computer Vision", Czech Technical University, 2022

< 1% match (Zeng, Qili. "Learning Temporal Variations for Action Recognition.", Boston University, 2021)
Zeng, Qili. "Learning Temporal Variations for Action Recognition.", Boston University, 2021

Some Efficient Numerical Methods for Solving Nonlinear Equations and Their Dynamics A Thesis submitted in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Mathematics by Himani Sharma (Reg. No. 902011013) under the supervision of Dr. Munish Kansal (Assistant Professor) Thapar Institute of Engineering and Technology, Patiala, India Dr. Ramandeep Behl (Associate Professor) King Abdulaziz University, Jeddah, Saudi Arabia DEPARTMENT OF MATHEMATICS THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY (Deemed to be University) Patiala - 147004, India February, 2025 CERTIFICATE I hereby certify that the work, which is being presented in the thesis entitled “Some Efficient Numerical Methods for Solving Nonlinear Equations and Their Dynamics” in partial fulfillment of the requirements for the award of degree of Doctor of Philosophy and submitted to the institution is an authentic record of my own work carried out during the period February, 2021 to February, 2025 under the supervision of Dr. Munish Kansal, Assistant Professor, Department of Mathematics, Thapar Institute of Engineering and Technology, Patiala, India and Dr. Ramandeep Behl, Associate Professor, Department of Mathematics, King Abdulaziz University, Jeddah, Saudi Arabia. Date: Himani Sharma (Reg. No. 902011013) It is certified that the above statement made by the candidate is correct to the best of my knowledge. Date: Dr. Munish Kansal (Assistant Professor) Thapar Institute of Engineering and Technology, Patiala, India Dr. Ramandeep Behl (Associate Professor) King Abdulaziz University, Jeddah, Saudi Arabia iii ....dedicated to my beloved family Acknowledgments The completion of this doctoral thesis was possible with the inspiration and support of several people. I want to extend my sheer appreciation to all of them. I also want to convey my sincere thanks to Thapar Institute of Engineering and Technology, Patiala, for providing me with the logistics, basic infrastructural facilities, and professional environment for my research. Finally, I thank the ALMIGHTY for granting me perseverance and faith during the dif- ferent phases of life. (Himani Sharma) vii Abstract ix List of Published Pap ers 1. H. Sharma, M. Kansal, A modified Chebyshev–Halley-type iterative family with memory for solving nonlinear equations and its stability analysis. Mathematical Methods in the Applied Sciences (Wiley), 46(12), 12549–12569, 2023 (SCIE, Q1, Impact Factor: 2.1). (Covered under Objective 2) 2. H. Sharma, M. Kansal, Stability analysis and dynamical behavior of optimal mean-based iterative methods. Journal of Mathematical Chemistry (Springer), 1-23, 2024 (SCIE, Q2, Impact Factor: 1.7). (Covered under Objective 2) 3. H. Sharma, R. Behl, M. Kansal, H. Ramos, A robust iterative family for multiple roots of nonlinear equations: enhancing accuracy and handling critical points. Journal of Computational and Applied Mathematics (Elsevier), 444, 115795, 2024 (SCIE, Q1, Impact Factor: 2.1). (Covered under Objective 1) 4. M. Kansal, H. Sharma, Analysis of optimal iterative methods from a dynamical point of view by studying their stability properties. Journal of Mathematical Chemistry (Springer), 62(1), 198–221, 2024 (SCIE, Q2, Impact Factor: 1.7). (Covered under Objective 2) 5. H. Sharma, M. Kansal, R. Behl, An efficient two-step iterative family adaptive with memory for solving nonlinear equations and their applications. Mathematical and Com- putational Applications (MDPI), 27(6), 97, 2022 (Scopus, Q2, Impact Factor: 1.9).

(Covered under Objective 1) 6. H. Sharma, M. Kansal, R. Behl, An efficient optimal derivative-free fourth-order method and its memory variant for non-linear models and their dynamics. Mathematical and Computational Applications (MDPI), 28(2), 48, 2023 (Scopus, Q2, Impact Factor: 1.9). (Covered under Objective 1) x Contents Certificate Acknowledgments Abstract List of Published Papers List of Figures List of Tables 1 Introduction 1.1 1.2 1.3 1.4 1.5 General introduction . . . . Classification of iterative methods . . . . Fundamental concepts . . . . 1.3.1 Order of convergence . . . . 1.3.2 Computational order of convergence (COC) . . . . 1.3.3 Computational efficiency of iterative methods . . . . 1.3.4 Kung-Traub’s conjecture . . . . 1.3.5 R-order of convergence . . . . 1.3.6 Banach space . . . . 1.3.7 Fréchet derivative . . . . Stability analysis . . . . Literature survey . . . . 1.5.1 Iterative methods for simple roots of a scalar nonlinear equation . . . . 1.5.2 Iterative methods for multiple roots of a scalar nonlinear equation . . . 1.5.3 Iterative methods for simple roots of system of nonlinear equations . . 1.6 1.5.4 Complex dynamical analysis . . . . Structure of the thesis . . . . iii vii ix x xvi xix 1 1 2 3 3 3 4 4 4 4 5 5 6 7 16 17 18 22 2 An Efficient Iterative Family Adaptive with Memory and Their Applica- tions 25 2.1 Introduction . . . . 25 xi An iterative family without memory and its convergence analysis . . . . 2.2.1 Convergence analysis . . . . 2.3 An iterative family with memory and its convergence analysis . . . . 2.4 2.5 2.6 2.3.1 Convergence analysis . . . . Numerical results . . . . Basins of attraction . . . . Conclusions . . . . 27 28 29 30 32 39 42 3 Chebyshev-Halley Type Variants with Memory and Their Stability Analy- sis 47 3.1 Introduction . . . . 47 3.2 An iterative family with memory and its convergence analysis . . . . 48 3.2.1 Convergence analysis . . . . 50 3.3 Complex dynamics . . . . 51 3.3.1 Dynamical planes and parameter spaces . . . . 55 3.3.2 Dynamical planes . . . . 55 3.4 Numerical results . . . . 57 3.5 Basins of attraction . . . . 62 3.6 Conclusions . . . . 65 4 Optimal Iterative Family Involving First Order Derivative and its Complex Dynamics 67 4.1 Introduction . . . . 67 4.2 An optimal iterative family and its convergence analysis . . . . 69 4.2.1 Convergence analysis . . . . 69 4.3 Complex dynamics . . . . 70 4.3.1 Scaling theorem . . . . 70 4.3.2 Rational operator . . . . 73 4.3.3 Fixed and critical points . . . . 74 4.3.4 Dynamical planes and parameter spaces . . . . 78 4.4 Numerical results . . . . 80 4.5 Conclusions . . . . 86 5 Complex Dynamics of a Mean-Based Optimal Iterative Family and its Applications to Chemical Models 89 5.1 Introduction . . . . 89 5.2 An optimal iterative family and its convergence analysis . . . . 90 5.2.1 Convergence analysis . . . . 90 5.3 Complex dynamics . . . . 91 5.3.1 Scaling theorem . . . . 91 5.3.2 Rational operator . . . . 93 5.3.3 Fixed points and their stability . . . . 94 5.3.4 Critical points . . . . 96 xii 5.3.5 Dynamical planes and parameter spaces . . . . 99 5.4 Numerical results . . . . 101 5.5 Conclusions . . . . 106 6 An Optimal Derivative-free Fourth Order Method and its Memory Variant 6.1 for Nonlinear Models 111 Introduction . . . . 111 6.2 An optimal iterative method without memory and its convergence analysis . . 112 6.2.1 Convergence analysis . . . . 113 6.3 An iterative method with memory and its convergence analysis . . . . 115 6.3.1 Convergence analysis . . . . 116 6.4 Numerical results . . . . 117 6.5 Basins of attraction . . . . 123 6.6 Conclusions . . . . 123 7 A Robust Iterative Family for Multiple Roots of Nonlinear Equations 127 7.1 Introduction . . . . 127 7.2 An optimal iterative family and its convergence analysis . . . . 128 7.2.1 Convergence analysis . . . . 128 7.2.2 Some selective weight functions . . . . 130 7.3 Numerical results . . . . 131 7.4 Conclusions . . . . 141 8 An Iterative Family for System of Nonlinear Equations 143 8.1 Introduction . . . . 143 8.2 An iterative method and its convergence analysis . . . . 144 8.2.1 Convergence analysis . . . . 145 8.3 Numerical results . . . . 146 8.4 Conclusions . . . . 153 Bibliography 153 xiii List of Figures 2.1 Basins of attraction for PM1, PM2, PM3, HPF, SHM, respectively for p1(z) 2.2 Basins of attraction for PMM1, PMM2, PMM3, TM1, TM2, respectively for p1(z) . . . . 2.3 Basins of attraction for PM1, PM2, PM3, HPF, SHM, respectively for p2(z) 2.4 Basins of attraction for PMM1, PMM2, PMM3, TM1, TM2, respectively for p2(z) . . . .

Chapter 1 Intro duction 1.1 General introduction Numerical analysis is an area of Mathematics that focuses on creating effective methodolo- gies for obtaining numerical solutions to intricate mathematical problems Burden and Faires (2010); Gautschi (2011). The term ‘numerical methods’ refers to the approach of solving mathematical issues through basic arithmetic operations, including addition, subtraction, multiplication, division, and comparison. Given that these operations are precisely what computers can perform, there is a significant connection between numerical analysis and computer technology. In other words, numerical analysis is a technique for approximating solutions to complex problems by utilizing various algorithms. This discipline encompasses both mathematics and computer science, as it involves the design, evaluation, and application of algorithms aimed at producing numerical solutions for problems with continuous variables. Therefore, numerical analysis is relevant in numerous sectors, including business, engineering, natural sciences, social sciences, medical sciences, and applied sciences, where the demand for accurate solutions following calculations is prevalent. Large systems of equations, whether linear, nonlinear, or characterized by complex ge- ometries, can also be effectively addressed through numerical methods, which are widely used in the realms of science and engineering. Consider a continuously differentiable function F : D ⊆ Rn → Rn, D being a convex set such that F (X) = 0, (1.1.1) where F (X) = (f1(X), f2(X), . . . , fn(X))T , X = (x1, x2, . . . , xn)T . Here, the functions f1, f2, . . . , fn are the coordinate functions of F . (1.1.1) reduces to scalar nonlinear equation, when n = 1 and it can be rewritten as follows: f (x) = 0, (1.1.2) where f : D ⊆ R → R is a scalar function defined on a domain D. Such problems typically stem from real-world applications within four key engineering fields: civil, chemical, electrical, and mechanical. These issues can be modeled using a vari- ety of mathematical equations. Examples include the quest to find the roots of an auxiliary 1 equation related to higher-order homogeneous differential equations with constant coefficients, the determination of eigenvalues for a square matrix, and the solution of integral and differ- ential equations through finite difference techniques. These scenarios can often be framed as nonlinear equations or systems of nonlinear equations, potentially involving exponential, trigonometric, and hyperbolic functions, or they may be entirely transcendental in nature. There are two ways for finding the solutions of given problems, 1. Direct methods or analytical methods 2. Indirect methods (generally called as iterative methods). Direct or analytical methods are capable of determining the exact roots of an equation in a limited number of steps and can identify all roots simultaneously. Unfortunately, these methods are seldom applicable to the types of problems encountered. As a result, one must often rely on numerical techniques or indirect methods. Although these indirect methods do not guarantee a finite number of steps for completion, they provide approximate solutions for every linear and nonlinear equation. The process involves starting with an initial guess and producing successive approximations at each stage, which converge towards the accurate solution. An iterative method (IM) is of the form xn+1 = ϕ(xn), n = 0, 1, . . . , (1.1.3) where xn is an approximation to the root κ isolated in a real domain D, xn+1 is the next approximation and ϕ is a suitable continuous function defined on D. IM starts with an initial approximation x0 ∈ D to κ. The function ϕ is called iteration function (IF). 1.2 Classification of iterative methods At this point, we will outline a classification of IMs utilizing Traub’s framework Traub (1964). 1. Formula (1.1.3) defines the simplest IM where only one previous approximation xn is required for evaluating the next approximation xn+1. Such scheme is called one-point IM. The most commonly used IM of this type is given by xn+1 = xn − f′(xn), n = 0,1,..., f(xn) (1.2.1) known as Newton’s method (NM) or Newton-Raphson’s method. 2. Assume that real numbers xn−k, . . . , xn−1, xn are approximations to κ and let us define the mapping xn+1 = ϕ(xn; xn−1, . . . , xn−k). (1.2.2) The approximation xn+1 is determined by ϕ on the basis of the previous k + 1 ap- proximations. However, only xn is a new information, while xn−k,...,xn−1 are reused 2 information, which is indicated by the inserted semicolon. The IF ϕ of the form (1.2.2) is called a one-point iteration function with memory. The best known iteration function with memory is defined by the secant method, xn+1 = xn − f (xn) − f (xn−1) f (xn), n = 1, 2, . . . xn − xn−1 (1.2.3) 3. Another type of IF is constructed by introducing w1(xn), w2(xn), . . . , wk(xn), where xn is the common argument. The IF ϕ defined as xn+1 = ϕ(xn, w1(xn), . . . , wk(xn)), (1.2.4) is called a multipoint iteration function without memory. We see from (1.2.4) that the new approximation xn+1 is obtained by the use of only previous approximation xn, but through the k expressions wi. 4. Let the IF has arguments zj, where each such argument represents k + 1 quantities xj,w1(xj),...,wk(xj), k ≥ 1. Then, this IF can be represented in the general form as xn+1 = ϕ(zn; zn−1, . . . , zn−k). (1.2.5) Such iteration function is called a multipoint iteration function with memory. Namely, in each iterative step, we must preserve information of the last n approximations xj, and for each approximation, we must calculate k expressions w1(xj),...,wk(xj). 1.3 Fundamental concepts Some fundamental definitions and findings that serve as the foundation for the examination of numerical methods are included in this part. 1.3.1 Order of convergence Let ϕ : R → R be an IF which defines the iterative procedure xn+1 = ϕ(xn). If a real number ρ and a nonzero constant A exist such that | ϕ(xn) − κ| nl→im∞ |xn − κ|ρ = A, (1.3.1) then ρ is termed as the order of convergence (see Traub (1964)) and A is the factor of conver- gence or the asymptotic error constant (AEC). 1.3.2 Computational order of convergence (COC) Let xn−1, xn, and xn+1 be the last three successive approximations to the zero κ obtained in the iterative procedure xn+1 = ϕ(xn) supposedly of order ρ. The computational order of convergence as rediscovered by Weerakoon and Fernando (2000) is as follows: ρc = log|f(xn)/f(xn−1)|, n = 1,2,... log|f(xn+1)/f(xn)| (1.3.2) 1.3.3 Computational efficiency of iterative methods Let Y denotes the number of functional evaluations per iteration. Ostrowski (1960) introduced the efficiency index of an iterative method (IM) by E(IM) = ρY1, (1 .3.3) where ρ is the order of the method. 1.3.4 Kung-Traub’s conjecture Multipoint iterative methods without memory, requiring n + 1 function evaluations per iter- ation, have order of convergence at most 2n (see Kung and Traub (1974)). 1.3.5 R-order of convergence In some scenarios, it is not feasible to determine the order of convergence from the definition given in (1.3.1). Moreover, there are instances where the limit in (1.3.1) may be nonexistent. To remedy this situation, Ortega and Rheinboldt (1970) established a more generalized notion of convergence. For finding the R-order of convergence of method with memory, we make use of the concept by Ortega and Rheinboldt (1970) and Theorem 1.3.1 given by Traub (1964). Theorem 1.3.1. Consider an iterative method with memory that generates a sequence {xm} of approximations (converging to the root κ). If there exists a nonzero constant ζ and nonnegative numbers sj, 0 ≤ j ≤ k, such that the inequality k | ϵm+1 |≤ ζ | ϵm−j |sj ∏j=0 holds, then the R-order of convergence (denoted by OR) of this iterative method satisfies the inequality OR ≥ t∗, where t∗ is the unique positive root of the equation, k tk+1 − sj tk−j = 0. (1.3.4) ∑j=0 1.3.6 Banach space A Banach space (see Beauzamy (2011)) is a complete normed space (X, ∥ · ∥). A normed space is a pair (X, ∥ · ∥) consisting of a vector space X over a scalar field K (where K is commonly R or C) together with a distinguished norm ∥ · ∥ : X → R. Like all norms, this norm induces a translation invariant distance function, called the canonical or (norm) induced metric, defined by d(x, y) := ∥y − x∥ = ∥x − y∥ (1.3.5) for all vectors x,y ∈ X. This makes X into a metric space (X,d). A sequence (xn)∞n=1 is called d−Cauchy or Cauchy in (X,d) if for every real τ > 0, there exists some index N such that d (xn, xm) = ∥xn − xm∥ < τ, (1.3.6) whenever m, n > N . 1.3.7 Fréchet derivative The Fréchet derivative (see Narici and Beckenstein (2010) is a derivative defined on normed spaces. It is commonly used to generalize the derivative of a real-valued function of a single real variable to the case of a vector-valued function of multiple real variables, and to define the functional derivative used widely in the calculus of variations. Generally, it extends the idea of the derivative from real-valued functions of one real variable to functions on normed spaces. Let V and W be normed vector spaces, and U ⊆ V be an open subset of V . A function f : U → W is called Fréchet differentiable at x ∈ U if there exists a bounded linear operator A : V → W such that ∥f (x + h) − f (x) − Ah∥W lim∥h∥→0 ∥h∥V = 0. (1.3.7) If there exists such an operator A, it is unique, so we write Df (x) = A and call it the Fréchet derivative of f at x. 1.4 Stability analysis The study of the dynamical behaviour of iterative methods for solving nonlinear equations in the complex plane is a subject that has drawn the attention of researchers in the last decades. Papers by Roberts and Horgan-Kobelski (2004); Amat et al. (2004); Varona (2002); Kneisl (2001) and the references therein are a good evidence of this fact. The seminal works of Cayley and Schröder at the end of the nineteenth century, dealing with Newton’s method applied to quadratic polynomials, were the beginning of a theory (iteration of rational functions) that has been in continuous evolution. The application of root-finding methods to polynomial equations leads to rational maps defined in the extended complex plane. Therefore, the theory and concepts related with the iteration of rational maps (see Curry et al. (1983); Beardon (1991)) can be applied in this situation. Firstly, we will recall the basic theory concerning the complex dynamics (see Blanchard (1984)) that we will be using throughout. Assume a rational map R : Ĉ → Ĉ, where Ĉ denotes a Riemann sphere. The objective of this study is to analyze the asymptotic nature of the orbit based on the starting point x0. Precisely, the efficient procedures are designed to investigate the dynamical plane of the function R. We obtain these phase spaces by categorizing the initial points according to the asymptotic nature of their orbits. A point x0 ∈ Ĉ in the complex plane is termed as fixed point if R(x0) = x0. A point x0 is called periodic with period p > 1 whenever Rp(x0) = x0, but Rk(x0) ≠ x0, for 1 < k < p. Furthermore, the point x0 is critical if R′(x0) = 0. In terms of classification of fixed points, x0 is considered to be attractor if |R′(x0)| < 1, a superattractor if |R′(x0)| = 0, a repulsor if |R′(x0)| > 1 and a parabolic point if |R′(x0)| = 1. The basin of attraction (BoA) of κ is defined by the set containing all pre-images of arbitrary order B(κ) = {x0 ∈ Ĉ : Rn(x0) → κ, n → ∞}. (1.4.1) The set in which the orbits converge to an attractor is referred to as Fatou set, denoted as F (R). Its complement in the complex plane is known as the Julia set, denoted as J (R). Consequently, J (R) encompasses all repelling fixed points, periodic orbits and their pre- images. This implies that the BoAs of any fixed point are part of F (R), while the boundaries of these BoAs reside in J (R). 1.5 Literature survey Identifying the zeros of a nonlinear equation represented by (1.1.2) is considered one of the most critical challenges in both theoretical and applied aspects of various scientific and tech- nological fields. To address this equation, a variety of qualitative methods are utilized. Many mathematical issues encountered in science and engineering are complex and at times, cannot be solved exactly. Consequently, approximating these challenging mathematical problems is essential to facilitate their resolution. In essence, the application of analytical methods to intricate functions may not be applicable in a broader context. Therefore, iterative methods emerge as a powerful approach for deriving approximate solutions. Traub (1964) provided a qualitative and quantitative analysis of these iterative techniques, which has encouraged a multitude of researchers to delve into this concept. Moreover, Traub (1964) made a distinction between one-point and multipoint methods within the realm of iterative techniques. The multipoint iterative methods are considered to be among the most proficient methods for solving nonlinear equations. This group of methods was the subject of extensive research in Traub's book and in numerous articles and books published in the latter part of the twentieth century. The past few years have witnessed an increased interest in multipoint methods, primarily driven by two main considerations: ˆ The application of multipoint methods in root-solving is attracting significant interest at present, as these methods transcend the theoretical boundaries of one-point approaches, especially regarding their convergence order and computational efficiency. ˆ Significant enhancement of computer hardware, particularly through the development of powerful processors, alongside improvements in software such as multi-precision arith- metic and symbolic computation, has enabled the implementation and convergence anal- ysis of multipoint methods that can produce root approximations with exceptional ac- curacy. Multipoint methods with memory draw upon information from both the current iteration and those preceding it. While the initial concepts for the formulation of this class of methods originated in 1964, as presented in Traub's book, there is a scarcity of contributions on this subject in the existing literature. The convergence order of innovative multipoint methods with memory exceeds that of their optimal counterparts without memory. This accelerated convergence is facilitated by the modification of a self-correcting parameter, which is recur- sively determined as the iterations advance, utilizing data from both the current and previous iterations. The significant enhancement in the convergence rate, achieved without requiring additional function evaluations, constitutes a major advantage of multipoint methods with memory. The foremost goal and inspiration behind the formulation of new techniques is to enhance computational efficiency, which involves attaining the highest feasible convergence order while limiting the number of function evaluations per iteration. Methods that incorporate memory generally demonstrate a high degree of stability concerning the variety of initial estimations that yield convergence. By judiciously choosing parameters, iterative methods without mem- ory can markedly elevate their convergence order, thereby functioning similarly to memory- based approaches. 1.5.1 Iterative methods for simple roots of a scalar nonlinear equa- tion The most basic examples of iterative techniques for numerically determining the simple roots of nonlinear equations are the bisection method and the regula-falsi method. These approaches are not recommended in reality when accuracy in multi-precision digits is needed because of their poor convergence speed. The inability of these methods to identify multiple roots with even multiplicity is another disadvantage. Because locally convergent methods converge more quickly than globally convergent ones, one chooses to utilize them. From the last decade, hybrid algorithms, beginning with a globally convergent approach and transition- ing to a locally convergent one as the approximations approach the desired zero, are being employed. One can consult various top-notch textbooks for additional information and a thorough analysis of these techniques given by Traub (1964); Ostrowski (1960); Ortega and Rheinboldt (1970); Kantorovich and Akilov (2014); Chapra and Canale (2015); McNamee and Pan (2013); Petkovic et al. (2012). There are numerous ways to develop computationally ef- ficient techniques to find zeros. A number of iterative approaches have been developed using geometrical approach Ostrowski (1960); Traub (1964); Amat et al. (2003); Chun (2007b); Kan- war et al. (2008), functional approach Kanwar et al. (2006); Chun (2007d); Sharma (2007), Adomian decomposition approach Adomian and Rach (1985); Adomian (1988, 2013); Abbaoui and Cherruault (1994); Abbasbandy (2003); Chun (2005); Petkovic et al. (2012), quadrature approach Weerakoon and Fernando (2000); Frontini and Sormani (2003); Homeier (2005); Kou et al. (2006); Kanwar (2006); Mir and Zaman (2007), weight function approach Geum and Kim (2011b); Zhou et al. (2011); Soleymani et al. (2012b,d); Sharifi et al. (2012); Soleymani et al. (2012a), inverse interpolation approach Traub (1964), sampling approach Traub (1964); Jarratt (1966, 1969); Neta et al. (1979); Chun and Neta (2012), etc. some of which we will discuss as we proceed through the chapter. The most well-known iterative method for solving nonlinear equations is NM Traub (1964); Petkovic et al. (2012), defined in (1.2.1). This method exhibits quadratic convergence for simple roots and linear convergence for multiple roots. The efficiency index of NM is 2 2 1 ≈ 1.414. While this method is optimal, its reliance on the derivative can be a limitation in certain applications. It is often the case that evaluating derivatives can be prohibitively expensive in various practical contexts. To mitigate this challenge, a number of researchers have proposed the approach of omitting derivatives from iterative functions, relying exclusively on the function that characterizes the problem for the calculation of iterates. Geometrically, NM substitutes f with its tangent line close to x to obtain each new approximation xn+1 to the zero. The approximation f ′(x) ≈ f (x + h) − f (x) h (1.5.1) is accurate for small h. The consecutive approximations xn−1, xn are taken from (1.5.1), the derivative can be approximated as f ′(xn) ≈ f (xn) − f (xn−1) xn − xn−1 . (1.5.2) Using the estimate (1.5.2) in (1.2.1) leads to the formulation of the secant's method described in (1.2.3). Visually, this method is the intersection of x−axis and the secant line along xn−1, xn. The error relation for the same is |en+1| ≤ η|en||en−1|. In accordance with Theorem 1.3.1, the order of this method is approximately calculated to be 1.618. The secant method is classified as a method with memory. It exhibits superlinear convergence and eliminates the need for evaluating the derivatives of the function. Furthermore, taking h to be f (x) for its small values in (1.5.1) and replacing in (1.2.1), the derivative free optimal Steffensen's method (see Steffensen (1933)) of order 2 is formulated as follows: xn+1 = xn − f (xn)2 f (xn + f (xn)) − f (xn) , n = 0, 1, 2, . . . (1.5.3) The efficiency index of this method is 2 12 ≈ 1.414, the same as that of Newton's method. Among the various third-order methods, Halley's method (see Petkovic et al. (2012)) stands out as the most widely utilized given by xn+1 = xn − f ′(xn) f (xn) 1 f (xn)f ′′(xn) , n = 0, 1, 2, . . . (1.5.4) 1 − 2f ′(xn)2 The above-mentioned expression was formulated by Schröder (1870a) which has been derived in several manners. For instance, Salehov (1952) used the osculatory rational function, (x + c)/(ax + b) and named it the method of tangent hyperbolas. Numerous contemporary and traditional iterative methods can likewise be developed through geometric techniques. For example, the derivation of Euler-Cauchy's method involves identifying the intersection of a quadratic parabola with the x-axis. Chun and Kim (2010) developed a method of order 3 utilizing the circle of curvature, an essential concept within the field of differential geometry. New iteration functions are formulated from those of order two, drawing upon geometric insights for their construction by Chun (2007c). Another notable third-order scheme is the widely recognized Chebyshev's method (see Schröder (1870a); Gutiérrez and Hernández (1997)), which is given as follows: f (xn) f (xn)f ′′(xn) xn+1 = xn − f ′(xn) 1 + 2f ′(xn)2 , n = 0, 1, 2, . . . (1.5.5) ( ) This method can be derived from (1.5.4) using the estimate 1 1 − t ≈ 1 + t (for small t). Potra and Pták (1984) have introduced the following modified version of NM that exhibits third-order convergence: xn+1 = xn − f (xn)f ′(xn − w(xn)) f ′(xn) , n = 0, 1, 2, . . . (1.5.6) Further, Laguerre (1878), a French mathematician, developed a third-order method as follows: xn+1 = xn − αf(xn) f′(xn) ± (α − 1)2 f′(xn)2 − αα−1f(xn)f′′(xn) 1/2 , α ≠ 0, 1. (1.5.7) It is one of the most preferred(method(because of the fact that it)h)as a great feature of global convergence in the case of polynomial zeros (real) and is nearly insensitive to initial estimates. It is noteworthy that a number of techniques can be derived from Laguerre's method as special cases, namely, NM for α = 1 (a limiting case), Euler-Cauchy's method for α = 2, Halley's method for α = 0 (a limiting case), square-root method (Ostrowski (1960)) as α → ∞, method of Hansen and Patrick (1976) by taking α = β1 + 1, β ≠ 0, etc. Schröder (1870a) introduced two general methods for finding zeros that possess an arbitrary order of convergence, which he termed the methods of the first and second kind. These families of methods have been rediscovered multiple times between 1946 and 1997, as noted by Petković and Herceg (1999); Petković and Petković (2008); Petković et al. (2010c). The method of the first kind of order ρ is as follows: ρ−1 Eρ(xn) = xn + (−1)j f (xn)j f(j)(f (xn)); (1.5.8) f(j)(f (xn)) = (f ′)2j−1 pj Σj=1, p1 = 1,j!pj+1 = f ′p′j − (2j − 1)pjf ′′, j = 1, 2, . . . Here, f is inverse of f and pj is a polynomial in f ′, f ′ ′, . . . , f (j). Schróder's basic sequence, {Eρ} is frequently employed to determine the convergence order associated with iterative techniques. The method of the second kind of order ρ is as follows: Iρ(xn) = xn − Oρ−1(xn) Oρ−2(xn) ; O0(xn) = 1 ρ f (xn) , Oρ(xn) = Σj=1 (−1)j−1 fj!(fj)((xxnn)) Oρ−j(xn), ρ = 1, 2, . . . (1.5.9) Many iterative techniques can be developed employing this relation. Traub (1964) used interpolation as one of the approach to derive some two-point methods. An interpolation function Θ(x) is constructed such that Θ(ρ)(x) = f (ρ)(x), ρ = 0, 1, . . . , n. (1.5.10) The goal is to use lower derivatives of f assessed at a specific number of points to replace the higher order derivatives. Traub (1964) developed the following method of order 3: xn+1 = xn − f (xn) f ′ xn − 12 w(xn) , n = 0, 1, . . . (1.5.11) Using quadrature rule of midpoints, Fron(tini and Sorm)ani (2003) rediscovered (1.5.11). Home- ier (2003) and Özban (2004) also derived the same after Traub. Weerakoon and Fernando (2000) used numerical integration to derive another third-order method by Traub. Several methods can be developed by employing distinct formulae for numerical integration (see Home- ier (2005)). Moving further, it is achievable to use certain suitable approximations in place of higher derivatives. For example, Chun (2007f) introduced several modifications of the Chebyshev- Halley methods that do not rely on the second derivative. Kou et al. (2006) introduced a modification of the NM which is particularly advantageous when the computational expenses associated with calculating the first derivative are equal to or exceed those of evaluating the function itself. Sharma and Guha (2011) formulated second derivative free multipoint methods from the one-point Hansen–Patrick family, which involves the second derivative. A notable progress in the development of efficient two-point methods has been made with the introduction of methods that utilize only three function evaluations while achieving a fourth-order convergence. This indicates that these methods are in alignment with the Kung- Traub hypothesis, i.e., they attain the optimal order of convergence. Their computational efficiency is 41/3 ≈ 1.587. Well before Traub conducted his in-depth studies in this domain, Ostrowski (1960) was the pioneer in creating the first optimal two-point method, employing the interpolation of f via a linear fraction, which is given as follows: yn = xn − f′(xn) f(xn) , xn+1 = yn − f′(xn) (f(xn) − 2f(yn)) f(yn) f(xn) (1.5.12) , n = 0, 1, . . . The fourth-order Ostrowski's method (OM) have been derived using various manners. It can be considered a fundamental technique that serves as the foundation for numerous families of two-point methods. Given that OM frequently yields the most favorable outcomes among those with comparable attributes, this claim is largely substantiated. It is important to acknowledge that the favorable convergence characteristics of OM have been established by Yun and Petković (2009), who utilized initial approximations derived from a technique by Yun (2008) involving the numerical integration of sigmoid-like functions. One of the derivation of OM starts with double NM given by yn = xn − f′(xn) f(xn) , xn+1 = yn − f′(yn), n = 0,1,... f(yn) (1.5.13) Using a suitable estimate for f ′(yn), OM can be deduced. It is important to highlight that (1.5.13) does not qualify as a proper two-point method. Rather, we utilize it as an auxiliary scheme in the development of optimal two-point methods. A different formulation of OM was suggested by King (1973a) (KF) as a generalization which is fourth-order optimal given by xn+1 = xn − w(xn) − f (xn − w(xn)) f ′(xn) (f (xn) + (β − 2)f (xn − w(xn))) , n = 0, 1, . . . , β ∈ R. f(xn) + βf(xn − w(xn)) (1.5.14) OM is a special case of (1.5.14) when β = 0. Also, the methods by Kou et al. (2007a); Chun (2008); Chun and Ham (2008a) are also some of the special cases of (1.5.14). Various forms of KF, derived from distinct approximations to f ′(x), were introduced by Chun (2007g,a); Kou and Li (2007a). OM is also constructed as a special case by Chun and Ham (2007a). The King-Ostrowski type methods were also developed by Chun and Neta (2009) employing the method of undetermined coefficients, Kou et al. (2007a) combining the methods by Potra and Pták (1984) and Sharma (2005), etc. Proceeding ahead, the computational efficiency of the iterative scheme (1.5.13) (order and number of evaluations are 4) is evidently equivalent to that of NM. To decrease the number of evaluations and thereby improve computational efficiency, Chun (2008) introduced a valuable technique for estimating f ′ through the use of an appropriately chosen weight function D, f′(yn) ≈ f′(xn)D(tn). Here, tn = ff((xynn)). So, (1.5.13) takes the form: yn = xn − f′(xn) f(xn) , xn+1 = yn − f′(xn)D(tn), n = 0,1,..., f(yn) (1.5.15) D being a real function to be selected so as to achieve order 4. The conditions for the same are D(0) = 1, D′(0) = −2, and |D′′(0)| < ∞. The proposed concept is both straightforward and effective. At times, it becomes essential to expand D into Taylor or geometric series in order to create appropriate iteration functions, whether they are new or already established. A more straightforward strategy that does not change the weight function, as considered by Petković and Petković (2010), is based on Chun's idea by using f ′(yn) ≈ f ′(xn) D(tn) . This approach relies on the assumption that D, along with its derivatives D′ and D′′, remains continuous around 0. So, (1.5.13) takes the form: yn = xn − f ′(xn) f (xn) , f (yn) f (yn) (1.5.16) xn+1 = yn − D(tn) f ′(xn) , tn = f (xn) , n = 0, 1, . . . , The conditions for order 4 are D(0) = 1, D′(0) = 2, and |D′′(0)| < ∞. We outline a few variants of (1.5.16) as we proceed ahead. KF is obtained from (1.5.16) for D(t) = 1+(β−2)t, β ∈ R. 1+βt Cordero et al. (2010b) introduced an optimal fourth-order method for D(t) = 1 + 2t + t2 and so on. As noted previously, derivative-free methods prove to be advantageous for identifying the zeros of f, particularly when the computation of its derivatives is complex and costly. Petković et al. (2010b) made (1.5.13) free from derivative by taking h to be τf(x) in (1.5.1) and using the subsequent expression as f′(xn), and a specific weight function to estimate f′(yn) in (1.5.13). Such kind of derivative-free techniques can be found in Thukral and Petković (2010); Peng et al. (2011); Sharma and Goyal (2006); Ren et al. (2009); Cordero and Torregrosa (2011); Liu et al. (2010). The majority of the multipoint methods discussed thus far utilize two or more evaluations of f, along with a single evaluation of f′. Following OM (1.5.12), which achieved optimal order four in 1960, Jarratt (1966) introduced the subsequent two-point method of fourth order given as follows: yn = xn − 3 f ′(xn) 2 f (xn) , xn+1 = xn − p1w(xn) − p2 f ′(yn) p3f ′(xn) + p4f ′(yn) , n = 0, 1, 2, . . . f (xn) − f (xn) (1.5.17) Here, p1 = 1 + , p2 = 1 − 1 3 3 1 4 , p3 = − (1 − δ)2, p4 = 8δ 8δ2 (2δ) 4 (2(δ − 1)) 3 3 (δ − 1). It is based on an expanded version of Traub's form (see Traub (1964)), and it incorporates one function evaluation and two derivative evaluations for each iteration. Consequently, optimal two-point methods that adhere to this evaluation model are frequently referred to as Jarratt- type methods. It is important to highlight that, among the two-point methods, only Jarratt's model has the capability to produce two-point methods of optimal order four for the purpose of locating multiple roots. A comparable methodology was employed by Jarratt (1969) to develop a two-point tech- nique of fourth order, which requires the same number of evaluations. Using expansion of ge- ometric series, an inverse-free Jarratt's method is formulated. Amat et al. (2004) and Varona (2002) proposed its convergence. Several such instances can be seen in Chun and Ham (2008b); Kou et al. (2007b); Sharma et al. (2009); Basu (2008). Also, Chun et al. (2012) employed an alternative methodology to establish a generalized family of Jarratt's type through which various methods can be formulated. Next, the study of non-optimal three-point methods is undertaken because their design is informed by a wide range of diverse, inspiring, and genuine developmental techniques that have systematically led to the emergence of methods of optimal order. It has been found that these techniques, which involve various interpolation methods, Taylor series approximations, and weight functions, play a crucial role in the formulation of higher-order optimal multipoint methods. After Ostrowski, King and Jarratt, the further optimal methods were introduced by Kung and Traub (1974) which exhibit arbitrary order 2n for any integer n ≥ 1. In the time frame from 1974 to 2007, the four-point method by Neta (1983) was the sole method to achieve an optimal order of 16, while none of the three-point methods developed during this period managed to reach the optimal order of 8. In the years following the establishment of Kung-Traub's families, a range of n−point methods (n ≥ 3) surfaced in the scholarly articles of Neta et al. (1979); Neta (1981, 1983); Popovski (1981); King (1973a,b). Certain methods outlined in these publications were created through alternative methodologies or utilized more than one initial estimation, thus compli- cating their direct comparison to optimal multipoint methods. Following KF, Neta Neta et al. (1979) formulated the first three-point sixth-order method with 4 function evaluations. The subsequent three-point methods achieving optimal order 8 were introduced in an explicit manner many years after the families proposed by Kung-Traub, specifically in the works of Milovanović and Cvetković (2007) and the articles authored by Bi et al. (2009b,a). It is important to highlight that a significantly more efficient four-point method of optimal order 16 was developed by Neta (1983). This particular method is based on an implicitly defined three-point method of eighth-order. This raises the question of precedence between Neta's findings and the results presented from 2007 to 2009. Furthermore, between 1974 and 2007, numerous three-point methods with orders lower than 8 were established. Following the works of Bi et al., several eighth-order three-point methods were created employing a variety of techniques and concepts. Further, using a parametric function in the third step, Chun and Ham (2007b) formu- lated a sixth-order family of three-point methods following (1.5.12), efficiency index being 61/4 ≈ 1.565, less than that of optimal two-point methods (41/3 ≈ 1.587). Special cases include methods by Sharma and Guha (2007); Grau and Díaz-Barrero (2006), considering specific weight function and parameter values. Similarly, following (1.5.17), Jarratt’s method was accelerated by Kou and Li (2007b) using linear interpolation, and by Chun (2007e) using equation for parabola. Chun and Neta (2008) employed the method of undetermined coeffi- cients to formulate another sixth-order method. Using linear interpolation, Parhi and Gupta (2008) formulated the same kind of scheme. A category of third-order techniques known as Chebyshev-Halley methods (see Amat et al. (2003); Gutiérrez and Hernández (1997)) is given as follows: xn+1 = xn − 1 + 1 − αLf(xn)) f′(xn) Lf (xn) f (xn) , n = 0, 1, . . . , α ∈ R, (Here, Lf (xn) = f ′ ′2(fx′n()xfn()x2n) . This family features Halley’s method when α = 12 , Chebyshev’s method when α = 0, and super-Halley’s method when α = 1. Estimation of Lf (xn) formulates a method of order 3 which in turn led to the first seventh-order method by Kou et al. (2007c). Another method of order 7 was constructed by Bi et al. (2008). Using an estimation for derivative in third step and choosing suitable weight function, Bi et al. (2009b) formulated an optimal method of order 8. Employing a few modifications, Bi et al. (2009a) formulated another optimal eighth-order method. Further, some optimal families can be seen in the work of Petković (2010, 2011); Petković and Petković (2010); Kou et al. (2010); Petković et al. (2010a) which utilize advanced two-point methods of order four and incorporate interpolation techniques to approximate f ′(z) in the third stage. These methods have efficiency index 23/4 ≈ 1.682. The subsequent category of optimal three-point methods is based on the application of optimal two-point methods in the initial two steps, followed by the utilization of a third- degree inverse interpolating polynomial in the final step, as discussed by Neta and Petković (2010). Thukral and Petković (2010) proposed a family of three-point methods, where the first two steps are aligned with KF (1.5.14). The third step is constructed through the application of three weight functions, arranged in such a manner that the overall order of convergence for the three-step method reaches eight. Several instances can be seen in the work by Thukral (2010); Džunić (2011); Liu and Wang (2010). Next, Džunić et al. (2012) formulated an optimal eighth-order derivative-free method by applying third-degree Newton’s interpolating polynomial. Our next focus is on multipoint methods without memory and have an order exceeding eight. These methods generate approximations of remarkable accuracy, which are infrequently necessary in practical scenarios. The formulation of n−point methods with an optimal order of 2n for any n ≥ 1 is justified from a theoretical standpoint; these methods are particularly significant for smaller values of n. Higher-order multipoint methods with this property include the family by Kung and Traub (1974) and the family by Zheng et al. (2011), both of which are derivative-free. Furthermore, a sixteenth-order method Neta (1983), based on inverse interpolation is optimized through a judicious choice of initial approximations. Also, with some suitable choices of weight functions, Geum and Kim (2011a) derived optimal methods of order 16. Petković (2010) has established a general class of optimal methods with arbitrary convergence orders through the use of Hermite’s interpolation. The work of Zheng et al. (2011) presents a one-parameter family of n−point methods that do not rely on derivatives and can attain an arbitrary order of convergence of 2n (for n ≥ 1). These methods are developed through the application of Newton’s interpolation and forward divided differences, which necessitate n + 1 function evaluations. Thus, they are deemed optimal according to the principles outlined in the Kung-Traub conjecture. As discussed earlier, the increase in the convergence order can be brought out with no further functional evaluation by making use of self accelerating parameter(s). Traub (1964) was the first one to introduce this idea and these methods were termed as methods with memory, the reason being that these methods use the previous information to calculate the next iterate. This idea by Traub was later developed by Petković et al. (2014) and is being used by many authors (see, for instance Cordero et al. (2015)) over these years. By employing two innovative strategies for the calculation of a self-correcting parameter, specifically a ‘sliding’ secant technique and Newton’s interpolation with divided differences, one can achieve extremely rapid convergence of new methods with memory, all without the need for additional function evaluations. As a result, these multipoint methods demonstrate a high level of computational efficiency. Additionally, another variant of multipoint methods with memory is founded on inverse interpolation and a strategic selection of initial approxi- mations, which significantly improves the accuracy of the approximations to the roots. The already existing Steffensen’s method had been improved in this sense by Traub (1964) and the first method with memory be given which is as follows: w0 = x0 + γ0f (x0), γ0, x0 are suitably given, ⎧xn+1 = xn⎫ − f [xn, wn] , 0 ≠ γn ∈ R, n = 0, 1, 2, . . . , f (xn) N1(x) = f (xn) + (x − xn)f [xn, wn], (1.5.18) ⎪ ⎪ γn+1 = −1 N1′(xn) , wn+1 = xn+1 + γn+1f (xn+1). Here, f [s, t] = f (s) ⎪ ⎩− f (t) s − t denotes a first-order divided difference. This method has order of convergence 2.414. Therefore, the order of the method with memory is higher than that of Newton’s method, which also requires two F.E. per iteration. It is still possible to increase the convergence order using a better self accelerating parameter. Petković et al. (2011) developed a two-point iterative method with memory having R-order of convergence at least 4.561 given as wn = xn + γnf (xn), γn = f (xn) − f (xn−1) xn−1 − xn , yn = xn − f [xn, wn] f (xn) , (1.5.19) xn+1 = yn − f [xn, wn] 1 + f (xn) f (wn)) , 0 ≠ γn ∈ R, n = 0, 1, 2, . . . f (yn) f (yn) f (yn) + (Further, Neta (1983) formulated a sixth-order three-point method with memory requiring 3 function and one derivative computation per step of iteration. Several iterative schemes without and with memory can be found in the literature, pub- lished in recent years, highlighting ongoing advancements in the field. 1.5.2 Iterative methods for multiple roots of a scalar nonlinear equation Locating multiple roots of f (x) = 0 with multiplicity m is one of the most important challenges in science and engineering. We must therefore research numerical techniques in this regard. Iterative methods intended for simple roots are either unsuitable or their convergence rate diminishes to linear when confronted with multiple roots. Therefore, it is crucial to either create new iterative algorithms or modify existing ones to effectively find multiple roots.

There are essentially two main types of challenges related to multiple zeros: ˆ When the multiplicity m associated with the multiple root is clearly defined. ˆ When the multiplicity m of the multiple root is not clearly defined, one can approximate both the multiple roots and their multiplicity in these cases. The modified NM by Rall (1966), one of the prime numerical methods to approximate a multiple root κ with multiplicity m, is given by xn+1 = xn − mf′(xn) f(xn) . (1.5.20) It is quadratically convergent, but the multiplicity m must be determined beforehand. This method can be derived from NM for the function (f (x))1/m. To enhance the local order of convergence, Laguerre (see Bodewig (1946)) introduced a family of methods that converge cubically, all derived from the aforementioned function. Notable techniques such as the Euler- Cauchy method, Halley's method, Ostrowski's square-root method, and the Hansen-Patrick family are recognized as specific instances of this family (see Hansen and Patrick (1976)). Numerous higher order methods have been developed by several authors to compute mul- tiple roots by using weight functions or the NM as a starting step. The said methods need prior knowledge about the multiplicity m (see Hansen and Patrick (1976); Victory Jr and Neta (1983); Dong (1987); Osada (1994); Neta (2008); Li et al. (2010); Shengguo et al. (2009); Zafar et al. (2020); Sharma and Sharma (2010); Zhou et al. (2011); Sharifi et al. (2012); Soleymani et al. (2013); Kansal et al. (2020); Cordero et al. (2016); Kansal et al. (2015b); Behl et al. (2019); Behl and Al-Hamdan (2019)). Many variations of King's family King (1973a) have been constructed and studied in order to approximate simple roots of nonlinear functions. Behl et al. (2020) used derivatives for multiple roots to extend King's family. We now turn our attention to the situation where the multiplicity m of the multiple root is not clearly specified. Traub (1964) recommended a particular approach to estimate the multiplicity m of the root as follows: m ≈ log |f (x)/f ′(x)| log |f (x)| , (1.5.21) m being sufficiently near to the zero κ. Lagouanelle (1966) used the following expression for the same: m≈ f ′(x)2 f ′(x)2 − f (x)f ′′(x) . (1.5.22) In contrast, Traub (1964) adopted a basic transformation rather than the function f (x) for the purpose of locating multiple zeros that possess an unknown multiplicity given as follows: T (x) = f′(x) f(x) , f′(x) ≠ 0 {0, f′(x) = 0 (1.5.23) If NM is applied to T (x), the following method by Schröder (1870a) of order 2 is formulated: xn+1 = xn − f (xn)f ′(xn) f ′(xn)2 − f (xn)f ′′(xn) (1 .5.24) which holds crucial for locating multiple roots (with unknown m). 1.5.3 Iterative methods for simple roots of system of nonlinear equations Tackling the problem of nonlinear equation systems is a vital and widespread concern in the realms of science and engineering Ortega and Rheinboldt (1970), i.e., determining a vector κ = (κ1, κ2, . . . , κn) for a nonlinear function F : D ⊆ Rn → Rn such that, F (κ) = 0. One can attain this solution as a fixed point of a specific function G : Rn → Rn utilizing the fixed point iteration, x(k+1) = G(x(k)), k = 0, 1, , . . . (1.5.25) A fundamental approach for addressing the system of nonlinear equations F (x) = 0 is the quadratically convergent Newton's method (see Kelley (2003)) given as follows: x(k+1) = x(k) − F ′(x(k))−1F (x(k)), k = 0, 1, 2, . . . , (1.5.26) where F ′(x(k))−1 is the inverse of first Fréchet derivative F ′(x) of the function F (x). To enhance the convergence rate of Newton's method, several alternative approaches have been suggested (see Cordero and Torregrosa (2006, 2007); Darvishi and Barati (2007a); Frontini and Sormani (2004); Amat et al. (2003); Grau-Sánchez et al. (2011)). For a system of n equations in n unknowns, the first Fréchet derivative F ′ is a matrix with n2 evaluations while the second Fréchet derivative F ′′ has n2(n + 1)/2 evaluations. The meth- ods like Halley and Chebyshev, despite their cubic convergence, are considered less practical from a computational point of view because of costly second-order derivative. For the scalar equation f (x) = 0, some authors have proposed third-order methods without using second- order derivative. Some of the instances include Frontini and Sormani (2003); Homeier (2003); Özban (2004); Weerakoon and Fernando (2000), each requiring three evaluations, namely, one f and two f ′. The extensions of these methods for systems of equations have been developed in Cordero and Torregrosa (2006); Frontini and Sormani (2004); Homeier (2004). In the last few years, multiple iterative methods have been established to resolve nonlinear systems of equations, predominantly employing Taylor's polynomial Ortega and Rheinboldt (1970), decomposition Darvishi and Barati (2007b), homotopy perturbation method Golbabai and Javidi (2007), quadrature formulas Cordero and Torregrosa (2006, 2007); Darvishi and Barati (2007a); Frontini and Sormani (2004); Babajee et al. (2008) and other techniques Homeier (2004); Kou (2007). Also, some new higher-order methods have been formulated without second Fréchet derivative. For example, Cordero and Torregrosa (2007) developed two variants of NM with at most third-order convergence. One of the methods requires n+3n2 evaluations per iteration whereas, the other requires n+4n2 evaluations. Cordero et al. (2009) also presented fourth and fifth-order methods requiring 2n + 2n2 and 3n + 2n2 evaluations, respectively. Darvishi and Barati (2007a) obtained a fourth-order method which uses 2n+3n2 evaluations. Grau-Sánchez et al. (2011) proposed third, fourth and fifth-order methods which require n + 2n2, 3n + n2 and 2n + 2n2 evaluations, respectively. Noor and Waseem (2009) developed two third-order methods, each requires n + 2n2 evaluations. Another family of third and fourth-order iterative methods was proposed by Nedzhibov (2008), which is defined by 1 (2 β I− λ Ω(x(k)) Ω(x(k)) h(x(k)), − 1 x(k+1) = x(k) − I + (β)) where Ω(x(k)) = I − F ′(x(k))−1F ′(y(k)), y(k) = x(k) − βh(x(k)), h(x(k)) = F ′(x(k))−1F (x(k)), k = 0, 1, 2, . . . (1.5.27) (1.5.28) Here, λ and β are arbitrary real parameters. This family is proved to have order of convergence 3 for (λ,β) ≠ (1, 23) and order of convergence 4 for (λ,β) = (1, 23). One can notice that the two-parameter family requires one F, two F′ per iteration and depending on the value of λ would require one matrix inversion or two matrix inversions per iteration. The development of such kind of methods has been increasing over the years. In general, methods intended for one-dimensional equations are ineffective for solving non- linear systems. Moreover, the concept of optimality related to functional evaluations and convergence order is not applicable to nonlinear equation systems. Specifically, the estab- lished Kung-Traub conjecture Kung and Traub (1974) does not hold true for such systems. In multidimensional scenarios, a method is deemed efficient if it requires the least number of matrix inversions, functional evaluations, and matrix multiplications per iteration to achieve the necessary accuracy. 1.5.4 Complex dynamical analysis The iteration of rational mappings of a complex variable has a rich historical background. Foundational contributions to this field were made by Fatou (1919) and Julia (1918) in the early 20th. In the 60s, notable advancements were achieved through the work of Brolin (1965); Guckenheimer (1970); Jakobson (1968). A significant progress had also been made by Sullivan (1982); Mané et al. (1983). While these studies had provided substantial theoretical insights, experimental investigations into the iterates of rational maps were also done. Among the few experimental studies, the work of Mandelbrot (1980) stands out as a singular achieve- ment. He took into consideration the transformation, PC(x) = x2 + C, C ∈ C, and generated remarkable images characterized by widespread self-similarity. Douady (1982) had provided a major breakthrough in comprehending Mandelbrot's bifurcation diagram. The abovemen- tioned quadratic transformation is arguably the simplest nontrivial rational map. Here, we focus on another set of examples: rational maps derived from applying Newton's method to a polynomial P. As stated earlier, one of the most widely recognized techniques for determining the roots (or zeros) of a polynomial P with real or complex coefficients is the Newton's iteration, also known as the Newton's method (NM) given by NP(x) = x − P′(x) P(x) , (1.5.29) having the following properties: ˆ If x0 is sufficiently close to a simple root κ of P, the sequence xn = NPn(x0) = NP(NPn−1(x0)) exhibits a quadratic convergence to κ. Specifically, there exists a constant τ > 0 such that |xn+1 − κ| < τ|xn − κ|2. ˆ If x0 is close to a multiple root κ, then |xn+1 − κ| < τ |xn − κ|2 for some τ (0 < τ < 1). So, the main emphasis is on selecting suitable initial conditions such that xn+1 → κ. The clas- sical theory by Fatou and Julia provides significant understanding of the potential behaviors of {xn}. A major motivation comes from the following theorem by Fatou: Theorem 1.5.1. If R(x) possesses an attracting periodic cycle, the orbit of at least one critical point will eventually converge to it. Schröder (1870a,b), and later, Cayley (1879b,a, 1890), proposed extending Newton's method for a polynomial P to the complex plane. Both scholars aimed to investigate the BoA associ- ated with a zero κ of P. From a local perspective, the problem is relatively straightforward. xn converges very fast to simple root κ (x0 close to κ), which implies that κ is an attractive fixed point for NP. In case of multiple root κ, the same scenario occurs. The relaxed New- ton's method is employed to enhance the convergence of NP towards a multiple root given as follows: NP,m(x) = x − mP′(x) P(x) . (1.5.30) For simple root κ, NP (κ + x) = κ + 12 NP′′ (κ)x2 + . . . and for multiple root κ of order q > 1, NP (κ + x) = κ + q−1 q x + ... If m = q is chosen, NP′,m(κ) = 0 resulting in rapid convergence of xn towards κ. However, it is important to note that NP,m does not converge to a simple root when m > 2, the root being a repelling fixed point. For m = 2, the convergence is very slow, the root being a neutral fixed point. Schröder (1870a) generalized Newton's method by introducing algorithms G for finding zeros of P, satisfying G(κ + x) = κ + xn G(n)(κ) + . . . , n ∈ N. 1 n! Specifically, xn = Gn(x0) converges to κ with order n (see Vrscay (1986); Vrscay and Gilbert (1987)). As we continue further, the Julia set, J (R), an invariant and perfect set associated with a rational map, is the closure of the set of repelling periodic points, where the map exhibits chaotic behavior. Under iteration, any neighborhood of a point in J (R) eventually maps to cover the entire Ĉ, with the exception of at most two points. The Fatou set, F(R), which is the complement of J (R), is where tame dynamics takes place, such as attracting cycles and the associated BoAs. F(R) contains the zeros of P(x) and their BoAs. These BoAs provide examples of locations where the numerical technique is effective. The numerical approach can go wrong in two ways. First, if the initial seed x0 is chosen in J (R), it will never converge to a zero of P(x). Nevertheless, a slight perturbation of x0 will result in convergence since any neighborhood of such an x0 contains points that do converge to one of the zeros. A more problematic situation arises when an attracting cycle exists that is not associated with the zeros. This cycle must reside in F(R), and its BoA forms an entire region in Ĉ, for which initial seeds fail to converge to a zero. In such a scenario, even a small perturbation of a failing seed may not result in convergence to a zero. We now give a mathematical interpretation of this discussion as follows: Definition 1.5.2. The set of 'good' initial points for the real NM is as Q = {x ∈ R : NPn(x) → κ, n → ∞, P(κ) = 0} and the set of 'bad' initial points is R\Q. NM relies on approximating the polynomial P(x) with a linear approximation. If P(x) is a degree-d polynomial with different zeros, then NP(x) becomes a rational map of degree d. It is evident that fixed points of NP(x) correspond to zeros of P(x). Actually, ∞ is the only fixed point of NP(x) that is not a root. Now, we compute NP′ (x) = P((Px)′(Px′)′)(2x) depicting the zeros to be superattracting. It becomes crucial to recognize that the inflection points of P(x), where P′′(x) = 0, correspond to the critical points of NP(x). Since, ∞ is repelling for NM, it is worth noting that the poles of NP(x) are the critical points of P(x). As a result, orbits that steer clear of the critical points of P(x) are most likely to converge quickly to a zero. The following theorem shows the relation of the critical points and the zeros of P(x): Theorem 1.5.3. The critical points of P(x) are located within the convex hull formed by the roots of it. Further, a critical point is referred to as free if it is an inflection point of P(x). To identify polynomials that contain extraneous attracting cycles, the orbit of the free critical point is traced. Any such cycle must attract this orbit, as the other critical points, which correspond to the roots, are fixed. This approach is grounded in Theorem 1.5.1 by Fatou and Julia (see Blanchard (1984)). Before undertaking further study, the following result should be mentioned: Lemma 1.5.1. Consider an affine map A(x) = ξ1x + ξ2, ξ1 ≠ 0. If V(x) = P ◦ A(x), then NP is analytically conjugate to NV via A, that is, A ◦ NV ◦ A−1(x) = NP. The concept of conjugation appears to be crucial for understanding the iteration of NP(x), and in the modern theory of iteration. In fact, Schröder (1870a) recognized the significance of conjugations, primarily to simplify calculations by obtaining a more convenient form. He proposed transforming an iteration xn+1 = R(xn) using a bijective mapping Y such that (i) Y−1 ◦ R ◦ Y(x) = x + a (Abel's equation) (ii) Y−1 ◦ R ◦ Y(x) = ax (Schröder's equation) (iii) Y−1 ◦ R ◦ Y(x) = xn (Boettcher's equation) A characteristic geometric property of methods like NM is the nearest root principal, where initial guesses converge to the closest zero. This principal accurately describes the dynamics when the technique is applied to a quadratic polynomial. The technique succeeds if the initial guess is closer to one zero than the other but fails if the initial seed lies equidistant from the two zeros. In such cases, the method leaves the perpendicular bisector between the two zeros unchanged, as if unable to decide among them. In case of NM, the dynamics on the line of points that are equally spaced from the roots are chaotic. These results on NM can be found in Schröder (1870a); Cayley (1879a) where the König iteration functions Kn applied to x2 − 1 are shown to be conjugate to x↦ → xn. Theorem 1.5.4. Assume P(x) to be a polynomial of degree 2 having distinct zeros. Then, NM for P(x) is globally, analytically conjugate to x↦ → x2. Furthermore, J (R) consists of all points located on the perpendicular bisector of the line segment connecting the two roots. Proof. Given a polynomial P(x) = (x − ξ1)(x − ξ2) containing roots ξ1, ξ2 ∈ R. The following Möbius transformation can be employed, mc(x) = x − ξ2 x − ξ1 , (1.5.31) acquiring the properties, (i) mc(ξ1) = 0, (ii) mc(∞) = 1, (iii) mc(ξ2) = ∞. Then, mc ◦ NP ◦ m−c1 is a rational map of degree 2 having fixed points at 0 and ∞ (both superattracting) and fixes 1. Under the conjugacy map mc, J (R) for x↦ → x2 corresponds to the perpendicular bisector of the line segment connecting ξ1 and ξ2. Along this bisector, NP exhibits the angle doubling dynamics of this map restricted to the unit circle. The analysis of NM becomes dramatically more complicated as on the degree of the poly- nomial is greater than 2 which is thoroughly explained in Blanchard et al. (1994). In the past decade, Amat et al. (2004) and Varona (2002) described the dynamical behavior of several well-known iterative methods. Moreover Devaney (2018); Robinson (2012) studied the dynamics of different iterative families. In most of these studies, interesting dynamical planes, including some periodical behavior and other anomalies, have been obtained. In a few cases, the parameter planes have also been analyzed. The iterative methods without memory have been overlooked by complex discrete dynamics in recent years (see previous studies Cordero et al. (2021, 2013d); Chicharro et al. (2013a); Cordero et al. (2022); Padilla et al. (2022); Cordero et al. (2013a); Chicharro et al. (2019)). 1.6 Structure of the thesis The present thesis is organized into eight chapters, which are outlined below: Chapter 1 of the thesis presents the motivation, and literature review for the research work, focusing on the development of the iterative methods to solve scalar as well as system of nonlinear equations in order to improve the order of convergence and analysis of stability properties of the rational functions associated to the iterative methods. Chapter 2 deals with introducing a new iterative family without memory for solving nonlinear equations which is based on a cubically convergent Hansen-Patrick type method. The beauty of our techniques is that they work even though the derivative is very small in the vicinity of the required root or f ′(x) = 0. On the contrary, the previous modifications either diverge or fail to work. In addition, we also extended the same idea for iterative method with memory. Numerical examples and comparisons with some of the existing methods are included to confirm the theoretical results. Furthermore, basins of attraction are included to describe a clear picture of the convergence of the proposed as well as some of the existing methods. Chapter 3 presents a new iterative family with memory for solving nonlinear equations numerically in order to achieve higher order of convergence in comparison to the cubically convergent Chebyshev-Halley type method. The acceleration of convergence speed has been attained using a self accelerating parameter which is estimated from the current and previous iterations using divided differences. Therefore, the order of convergence increases from 3 to 3.30 without any further functional evaluation. In addition, the complex and real dynamics of the proposed family have been studied. The parameter spaces and dynamical planes are presented. This study helps to determine the family members with stable behavior which in turn are suitable for practical problems. Chapter 4 presents an optimal fourth-order iterative family and the dynamical analysis of the family with the help of complex dynamics tools. This study allows us to find those parametric values for which the corresponding family variant's behavior is stable or unsta- ble. Furthermore, we calculate critical and fixed points associated with the rational operator linked to this iterative family. To visualize our findings, we draw dynamical and parameter planes. Hence, we can select the regions where the corresponding method is more efficient or shows chaotic behavior. The conclusions obtained from this stability analysis are used in the numerical section, where some academic and real-life problems are solved. Chapter 5 performs stability analysis of an optimal mean-based family of iterative meth- ods of order four. Taking into consideration the stability aspect of the specified method, one can describe the method's sensitivity to the initial guesses. A rational function corresponding to the iterative family is developed. The convergence and stability of a certain method can be analyzed upon finding the fixed points, critical points, periodic points, etc. of the rational function. Furthermore, the dynamical and parametric planes are drawn which help us to de- tect the stable as well as non-stable regions. It has been observed that stable iterative methods generally yield better performance on complex problems compared to unstable methods. This observation has been supported by numerical experiments that compare our proposed family with some existing methods for representing some chemistry problems, like conversion in a chemical reactor, equations of state, and continuous stirred tank reactor problem. Chapter 6 displays an optimal iterative scheme without memory free from derivatives for solving nonlinear equations. There are many iterative schemes existing in the literature which either diverge or fail to work when f ′(x) = 0. But, our proposed scheme works even in those cases. In addition, we also extended the same idea for iterative method with memory with the help of self-accelerating parameters estimated from the current and previous approximations. As a result, the order of convergence increased from 4 to 7 without the addition of any further functional evaluation . Chapter 7 introduces an iterative method that exhibits an optimal fourth-order conver- gence rate, ensuring rapid and accurate approximation of the multiple roots. Unlike conven- tional methods, the proposed algorithm can successfully converge even when the derivative is zero or approaches zero in the vicinity of the desired root. Chapter 8 deals with introducing a new iterative family for solving system of nonlinear equations. The effectiveness and performance of the new iterative techniques are demon- strated by a number of numerical examples. The current approaches to solving the system of nonlinear equations can be seen as being expanded upon and generalized by these new iterative techniques. Lastly, we conclude with a summary of the thesis contributions. In addition, future areas of research for new and related fields of study are also suggested. Chapter 2 An Efficient Iterative Family Adaptive with Memory and Their Applications This chapter deals with introducing a new iterative family without memory for solving nonlinear equations which is based on a cubically convergent Hansen-Patrick type method. The beauty of our techniques is that they work even though the derivative is very small in the vicinity of the required root or f ′(x) = 0. On the contrary, the previous modifications either diverge or fail to work. In addition, we also extended the same idea for iterative methods with memory. Numerical examples and comparisons with some of the existing methods are included to confirm the theoretical results. Furthermore, basins of attraction are included to describe a clear picture of the convergence of the proposed as well as some of the existing methods. 2.1 Introduction Determining the zeros of a nonlinear function promptly and accurately has become a very crucial task in many branches of science and technology. The most used technique in this regard is NM of order 2 for simple roots. Various higher order schemes have also been presented in Amat et al. (2003); Chen et al. (1993); Argyros et al. (2017); Cordero et al. (2021). One amongst them is the family of Hansen and Patrick (1976) of order 3 given by xn+1 = xn − α + 1 f(xn) [α ± (1 − (α + 1)Lf (xn))1/2] f′(xn), n = 0,1,2,..., (2.1.1) where Lf (xn) = f ′′(xn)f (xn) f ′2(xn) and α ∈ R\{−1}. This family comprises of Euler’s method for (α = 1), Ostrowski’s square-root method for (α = 0), Laguerre’s method for (α = ν−11,ν ≠ 1) and Newton’s method as a limiting case. Despite the fact that it has cubic convergence, the The contents of this chapter are published in: Mathematical and Computational Applications (MDPI), 27(6), 97, 2022 (Scopus, Q2, Impact Factor: 1.9) involvement of second order derivative is limiting its computational applications. This factor inspired many researchers to concentrate on multipoint methods Petković et al. (2014), since they overcome the drawbacks of one-point iterative methods with respect to the convergence order and efficiency. The main motive in the development of new iterative methods is to achieve order of convergence as high as possible with certain number of functional evaluations per iteration. Sharma et al. (2009) had modified (2.1.1), which is given as follows: yn = xn − α f ′(xn) f (xn) , xn+1 = xn − β + 1 f (xn) (2.1.2) [β ± (1 − (β + 1)Hf (xn))1/2] f ′(xn) , where Hf (xn) = f ′′(yn)f (xn) the authors calculated second-order derivative of f at yn. Moreover, several developmentsf ′2(xn) and α, β are free parameters. Here, β ≠ −1. Instead at xn, of Hansen-Patrick type methods have been presented and examined in the work by Kansal et al. (2015a) in order to eradicate the second-order derivative. Using some appropriate approximation for f ′′(xn), Kansal et al. (2015a) presented the following method: yn = xn − f′(xn) f(xn) , xn+1 = xn − ⌈ α + 1 f(xn) , (2.1.3) f(xn)2 + (β − 2α − 2)f(xn)f(yn) − β(α + 1)f(yn)2 1/2 ⌉ f′(xn) ⎢ α ± ⌊ (f(xn)2 + βf(xn)f(yn)) ⎥ where α and β are free parameters. The prominent problem when using such kind of methods ⌋ is that they fail to work in the case f ′(x) = 0 and diverge or fail when the derivative is very small in the vicinity of the required root. That’s why, our main goal is to develop a method that is globally convergent. As discussed earlier, the convergence order can possibly be increased with no further functional evaluation by making use of a self accelerating parameter. Traub (1964) was the first one to introduce this idea of the methods with memory. He made minute alterations in the already existing Steffensen’s method (see Zheng et al. (2011)) and presented the first method with memory as follows: γ0, x0 are suitably given, wn = xn + γnf (xn), 0 ≠ γn ∈ R, xn+1 = xn − f [xn, wn] , n = 0, 1, 2, . . . , f (xn) (2.1.4) where γn is a self-accelerating parameter given as −1 γn+1 = N1′(xn) , N1(x) = f (xn) + (x − xn)f [xn, wn], n = 0, 1, 2, . . . This method has order of convergence 2.414. Still, if we use a better self-accelerating param- eter, there are apparent chances that the order of convergence will increase. Also, using secant approach, by the reuse of information from the previous iteration, Traub refined a Steffensen-like method and presented the following method: γ0 is given, γn = f (xn) − f (xn−1) , n ∈ N, xn − xn−1 (2.1.5) xn+1 = xn − γnf (xn)2 f (xn + γnf (xn)) − f (xn) , having R-order of convergence at least 2.414. To this end, we firstly develop a new iterative family without memory and extend it to one with memory. We carry out the convergence analysis of the new families in order to demonstrate their order of convergence. To illustrate our theoretical results, numerical results for the proposed families and comparisons with some of the existing methods are then given. Lastly, we present the basins of attraction of the new families depicting clearly the convergence or divergence of the new as well as the existing methods. 2.2 An iterative family without memory and its conver- gence analysis We aim to construct a new two-point Hansen-Patrick type method without memory in this section. Suppose yn = xn − f ′(xn) be the Newton’s iterate. Expanding f (yn) about a point x = xn f (xn) by Taylor series, we get f(yn) ≈ f(xn) + f′(xn) (yn − xn) + 1f′′(xn)(yn − xn)2, 2 ⇒f′′(xn) ≈ 2f′(xn)2f(yn) f(xn)2 . Also, if we expand the function f ′(yn) = f ′ xn − ff′((xxnn)) about x = xn by Taylor series, we have f′(yn) ≈ f′(xn) + f′′(xn)(yn − xn), ( ) ⇒ f′′(xn) ≈ f′(xn)(f′(xn) − f′(yn)). f(xn) Using previous developments, we have 2f′(2fxf(n′ x()fnx2(nf)x2)(n2y)fn2()y)n) (ff′(f(fxx′((nnxx))nn())f(′f(′x(nx)n)−−ff′(′y(ny)n))) 2 2 f ′′(xn) ≈ + (+) . As we can see, this estimation for f ′′(xn) uses 4 functional evaluations per iteration, f (xn), f (yn), f ′(xn) and f ′(yn). To decrease the number of functional evaluations, King's approxi- mation King (1973a) may be used which is f′(yn) = f′(xn)f(xn) + γf(yn) , f(xn) + βf(yn) when γ = β − 2, where β is a free parameter. Now, using this new approximation for f ′′(xn) in (2.1.1), Kansal et al. (2015a) presented the following family: yn = xn − f′(xn) f(xn) , xn+1 = xn − ⌈ α + 1 f(xn)2 + (β − 2α − 2)f(xn)f(yn) − β(α + 1)f(yn)2 1/2 ⌉ (2.2.1) α ± f′(xn) f(xn) , ⌉ ⌊ (f(xn)2 + βf(xn)f(yn)) ⌉ where α and β are free parameters. ⌋ Now, in order to extend to the method with memory, we come up with an idea of intro- ducing a parameter η in the family (2.2.1) and we present a modification in this family as follows: yn = xn − f(xn) f′(xn) + ηf(xn) , xn+1 = xn − ⌈ α + 1 ⌉ f (xn) , α ± f (xn) 2 + (β − 2α − 2)f (xn)f (yn) − β(α + 1)f (yn)2 1/2 f ′(xn) + ηf (xn) n =⌊ 0, 1,(2, . . . ,⌉ f (xn)2 + βf (xn)f (yn)) ⌋ ⌉ (2.2.2) where α and β are free parameters. Next, we establish the convergence results for our proposed family without memory given by (2.2.2). 2.2.1 Convergence analysis Theorem 2.2.1. Suppose that f : D ⊂ R → R be a real function suitably differentiable in a domain D. If κ ∈ D is a simple root of f (x) = 0 and an initial guess x0 is sufficiently close to κ, then the iterative family (2.2.2) converges to κ with convergence order p = 3 having the following error relation, en+1 = − 12 (η + d2)(η(1 + α − β) + (−1 + α − β)d2)e3n + O(en)4, 1 f (n)(κ) where en = xn − κ and dn = n! f ′(κ) , n = 2, 3, . . . Proof. Expanding f (xn) about xn = κ by Taylor series, we have f (xn) = f ′(κ)(en + d2e2n + d3e3n + d4e4n) + O(en)5. Then, f ′(xn) = f ′(κ)(1 + 2d2en + 3d3e2n + 4d4e3n) + O(en)4. Using (2.2.3) and (2.2.4), we have f ′(xnf) (+xnη)f (xn) = en − (η + d2)e2n + (η2 + 2ηd2 + 2d22 − 2d3)e3n + O(en)4. Using (2.2.5) in the first step of (2.2.2), we have en,y = yn − κ = (η + d2)e2n + (−η2 − 2ηd2 − 2d22 + 2d3)e3n + O(en)4. Also, the Taylor's expansion of f(yn) is f(yn) = f′(κ)(en,y + e2n,y + e3n,y + e4n,y) + O(en,y)5. Using (2.2.3)−(2.2.7), we have ⌈ α + 1 f(xn) α ± f(xn)2+(β−f2α(x−n2))2f+(βxnf)(fxn(y)nf)(−ynβ)(α+1)f(yn)2 1/2⌉ f′(xn) + ηf(xn) ⌉ ⌊ = en + 1(η + d2)(d2(−1 + α − β) + η(1 + α − β))e3n + O(en)4, ( ) ⌉ ⌋ 2 Finally, putting (2.2.8) in the second step of (2.2.2), we get en+1 = −1(η + d2) (η(1 + α − β) + (−1 + α − β)d2)e3n + O(en)4, 2 (2.2.3) (2.2.4) (2.2.5) (2.2.6) (2.2.7) (2.2.8) (2.2.9) which is the error equation for the proposed family (2.2.2) giving convergence order three. This completes the proof. 2.3 An iterative family with memory and its conver- gence analysis Now, we present an extension to the family (2.2.2) by inclusion of memory having improved convergence order without the addition of any new functional evaluation. If we observe clearly, it can be seen from the error relation (2.2.9), if η = −d2 = −2f′(κ) f′′(κ) , then the order of convergence of our family (2.2.2) can possibly be improved, but this value can't be reached because the values of f ′(κ) and f ′′(κ) are not practically available. Instead, we can use approximations calculated by already available information. So, to improve the convergence order, we give an estimation using first order divided difference, given by bn = − 1 f ′[xn, xn−1] 2 f [xn, xn−1] , where f [s, t] = s − t f (s) − f (t) denotes a first-order divided difference. So, by replacing η by ηn in (2.2.2), we obtain a new family with memory using the two previous iterations x0, x1 as follows: ηn = − 2 f [xn, xn−1] 1 f ′ [xn, xn−1] , yn = xn − f (xn) f′(xn) + ηnf(xn) , xn+1 = xn − ⌈ α + 1 ⌉ (2.3.1) α ± f (xn)2 + (β − 2α − 2)f (xn)f (yn) − β(α + 1)f (yn)2 1/2⌉ ⌉ ⌊ (f (xn)2 + βf (xn)f (yn)) ⌉ × f ′(xn) + ηnf (xn) , n ∈ N, f (xn) ⌋ where α and β are free parameters. Next, we establish the convergence results for our proposed family with memory (2.3.1). 2.3.1 Convergence analysis Theorem 2.3.1. Suppose that f : D ⊂ R → R be a real function suitably differentiable in a domain D. If κ ∈ D is a simple root of f (x) = 0 and an initial guess x0 is sufficiently close to κ, then the iterative family (2.3.1) converges to κ with convergence order at least 3.30. Proof. Using Taylor series expansion about xn = κ, we get f (xn−1) = f ′(κ)(en−1 + d2e2n−1 + d3e3n−1 + d4e4n−1 + d5e5n−1) + O(en−1)6, f (xn) = f ′(κ) (en + d2e2n + d3e3n + d4e4n + d5e5n) + O(en)6. Then, f ′(xn−1) = f ′(κ)(1 + 2d2en−1 + 3d3e2n−1 + 4d4e3n−1 + 5d5e4n−1) + O(en−1)5, f ′(xn) = f ′(κ)(1 + 2d2en + 3d3e2n + 4d4e3n + 5d5e4n) + O(en)5. (2.3.2) (2.3.3) (2.3.4) (2.3.5) Now, using previous developments, we have ηn = − 2 f [xn, xn−1] 1 f ′[xn, xn−1] = − d2 + d22 − 3 d3 en−1 + −d32 + 5d2d3 − 2d4 e2n−1 + d42 − 72d22d3 + 3d23 ((2) (2) (2 + 3d2d4 − 5d5 e3n−1 d22 − 3d3 − 2 d32 − 2d2d3 + d4 en−1 + 3d42 − 17 d22d3 2 +)) ((2) − 2d4 + 3d42 − 17 d22d3 + 3d23) (2 + 3d23 + 5d2d4 − 5d5 e2n−1 2 en + − d32 +(5d2d3)) ((2) (2 + 5d2d4 − 5d5 en−1 e2n + O3(en−1en). 2)) (2.3.6) Using (2.3.3),(2.3.5) and (2.3.6) in the second step of (2.3.1), we get yn − κ = d22 − 3d3 − d32 + 5d2d3 − 2d4 e2n−1 e2n + d3 − 2 d32 − 2d2d3 ((2 en−1 + 1) (2)) ((2 + d4 en−1 + 8d42 − 22d22d3 + 3d23 + 20d2d4 − 10d5 e2n−1 e3n + O4(en−1en). (4)) ( )) (2.3.7) Then, using (2.3.7) in (2.3.3), we get f (yn) = f ′(κ) d22 − 3d3 en−1 + − d32 + 5d2d3 − 2d4 e2n−1 e2n + d3 − 2 d32 (((2) (2)) (2 − 2d2d3 + d4 en−1 + 8d42 − 22d22d3 + 3d23 + 20d2d4 − 10d5 e2n−1 e3n + O4(en−1en). 1 (4) ( ))) (2.3.8) Using (2.3.3),(2.3.5)−(2.3.8) in the third step of (2.3.1), we finally get en+1 = d32 − 3d2d3 en−1e3n + d2d3 e4n + O5(en−1en) 2 (2.3.9) ( ) 2 Now, we can see the lowest term of the error equation is d32 − 3d2d3 en−1e3n, therefore, by Theorem 1.3.1, the unique positive root √of the polynomial (s2 − 3s − 1 gives the R-order of the 2) proposed family (2.3.1), which is s = 3+2 13 ≈ 3.30. This completes our proof. Remark 2.3.1. It can be seen that the order of convergence has increased without the addition of any further functional evaluation. Moreover, the presented scheme with memory has a very simple body structure in comparison to the general complex structures of iterative schemes with memory. 2.4 Numerical results This section lays out the comparison of our families (2.2.2) and (2.3.1) with several existing schemes. The initial values of α, β and η (or η0) are assumed to be chosen beforehand to begin with the computations. Also, a suitable x0 must be fixed. The following members of the families (2.2.2) and (2.3.1) are chosen in order to perform the calculations: 1. P M1 and P M M1 for α = β = 12 . 2. P M2 and P M M2 for α = 1 and β = 12 . 3. P M3 and P M M3 for α = 1 and β = 1. We have taken η (or η0)= 0.01 in our computations. The following existing methods have been selected to facilitate comparisons with our methods: 1. Hansen-Patrick's family

(H P M) without memory Hansen and Patrick (1976): xn+1 = xn − α + 1 [α ± (1 − (α + 1)Lf (xn))1/2] f′(xn), n = 0,1,2,..., f(xn) where Lf (xn) = f ′′(xn)f (xn) f ′2(xn) and α ∈ R\{−1}. The results are obtained for α = . 1 2 2. Sharma et al. family (SHM) without memory Sharma et al. (2009): yn = xn − αf′(xn) f(xn) , xn+1 = xn − β + 1 f(xn) [β ± (1 − (β + 1)Hf(xn))1/2] f′(xn) , where Hf(xn) = f′′(yn)f(xn) f′2(xn) and α and β(≠ −1) are free parameters. The results are obtained for α = 1 and β = 12. 3. Halley's method (HM) without memory Chen et al. (1993): xn+1 = xn − 1 + 1 Lf (xn) f (xn) 2 1 − αLf (xn))] f ′(xn) , n = 0, 1, 2, . . . , [(f ′′(xn)f (xn) 1 where Lf (xn) = f ′2(xn) and α = . The results are obtained for α = . 1 2 2 32 (2.4.1) (2.4.2) (2.4.3) 4. Traub's method (T M1) with memory Traub (1964): γ0, x0 are suitably given, wn = xn + γnf (xn), 0 ≠ γn ∈ R, xn+1 = xn − f [xn, wn] , n = 0, 1, 2, . . . , f (xn) where γn is a self-accelerating parameter given as γn+1 = N1′(xn) , N1(x) = f (xn) + (x − xn)f [xn, wn], n = 0, 1, 2, . . . −1 The results are obtained for γ0 = 0.01. 5. Traub's method (T M2) with memory Traub (1964): γ0 is given, γn = f (xn) − f (xn−1) , n ∈ N, xn − xn−1 xn+1 = xn − γnf (xn)2 f (xn + γnf (xn)) − f (xn) . The results are obtained for γ0 = 0.01. (2.4.4) (2.4.5) Further, Table 2.1 displays some nonlinear functions (f1 to f5) used to carry out the compu- tations. Table 2.1: Test functions, associated zeros and the initial approximations (x0) Function Real zero x0 f1(x) = x2 − ex − 3x + 2 f2(x) = sin(πx)ex2+x cos x−1 + x log(x sin x + 1) f3(x) = (x − 2)(x10 + x + 2)e−5x f4(x) = x2 − 1 f5(x) = sin x 0.2575 0 2 −1 2π 0.70 0.50 2.20 0 1.69 In addition, some real-life problems are also solved after transforming them to nonlinear functions (f6 to f10). COC (computational order of convergence) and the errors of approxi- mations to the desired roots, | xn − κ | for n = 1, 2, 3 of ft(x), t = 1, 2, . . . , 10 are outlined in Tables 2.2−2.11. Remark 2.4.1. We have tested our proposed family of iterative methods for several values of the parameters α and β out of which the best ones (the values for which we got best results) are selected for numerical computations. Table 2.2: Numerical outcomes for f1(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f1 (x) Without memory P M1 P M2 P M3 H P M S H M H M 1.2 × 10−4 1.0 × 10−4 1.2 × 10−4 7.2 × 10−3 1.2 × 10−2 7.2 × 10−3 6.9 × 10−15 2.9 × 10−15 7.0 × 10−15 2.1 × 10−8 1.9 × 10−7 1.8 × 10−8 1.9 × 10−35 1.9 × 10−35 1.9 × 10−35 4.9 × 10−25 7.6 × 10−22 2.9 × 10−25 3.0000 3.0000 3.0000 2.9993 3.0008 2.9990 0.344 0.312 0.329 0.343 0.344 0.281 With memory P M M1 1.2 × 10−4 P M M2 1.0 × 10−4 P M M3 1.2 × 10−4 T M1 6.8 × 10−3 T M2 6.8 × 10−3 5.0 × 10−15 2.9 × 10−15 5.0 × 10−15 2.0 × 10−7 9.0 × 10−6 1.9 × 10−35 1.9 × 10−35 1.9 × 10−35 2.4 × 10−18 1.5 × 10−11 3.3435 3.3362 3.3434 2.4151 2.0037 0.407 0.407 0.344 0.343 0.312 Table 2.3: Numerical outcomes for f2(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f2 (x) Without memory P M1 P M2 P M3 H P M S H M H M 9.0 × 10−3 8.7 × 10−3 9.0 × 10−3 1.4 × 10−1 5.4 × 10−1 1.6 × 10−1 3.6 × 10−7 1.6 × 10−7 3.6 × 10−7 5.1 × 10−4 1.8 × 10−1 3.5 × 10−3 2.4 × 10−20 9.7 × 10−22 2.3 × 10−20 6.4 × 10−11 1.2 × 10−2 1.2 × 10−8 2.9961 2.9946 2.9950 2.7546 2.7518 3.1314 0.578 0.656 0.749 0.594 0.843 0.751 With memory P M M1 9.0 × 10−3 P M M2 8.7 × 10−3 P M M3 9.0 × 10−3 T M1 2.6 × 10−2 T M2 2.6 × 10−2 3.2 × 10−7 2.2 × 10−7 3.1 × 10−7 2.2 × 10−4 1.1 × 10−3 1.6 × 10−23 5.3 × 10−24 1.5 × 10−23 1.6 × 10−9 2.4 × 10−6 3.6558 3.6119 3.6550 2.4624 1.9291 0.859 0.875 0.938 0.672 0.657 Remark 2.4.2. In Table 2.5, for the function f4(x), as the derivative of the function becomes zero, the existing methods HPM, SHM and HM fail. Also, TM1 and TM2 converge to the desired root but in 11 and 14 number of iterations, respectively as we can see the errors of approximations are large in these cases. Further, for the function f5(x), HPM, SHM and HM converge to undesired root ′π′ and TM1, TM2 both converge to undesired root ′3π′ which can be seen in Table 2.6. Table 2.4: Numerical outcomes for f3(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f3 (x) Without memory P M1 P M2 P M3 H P M S H M H M 7.5 × 10−5 3.0 × 10−4 1.8 × 10−4 6.9 × 10−3 2.4 × 10−2 9.4 × 10−3 5.2 × 10−17 2.6 × 10−15 7.9 × 10−16 4.0 × 10−7 3.2 × 10−5 1.0 × 10−6 1.7 × 10−53 1.9 × 10−48 5.9 × 10−50 7.6 × 10−20 7.7 × 10−14 1.2 × 10−18 3.0008 2.9971 3.0019 2.9994 3.0020 2.9993 0.345 0.344 0.359 0.358 0.328 0.390 With memory P M M1 7.5 × 10−5 P M M2 3.0 × 10−4 P M M3 1.8 × 10−4 T M1 2.0 × 10−2 T M2 2.0 × 10−2 2.7 × 10−14 2.5 × 10−12 3.9 × 10−13 1.4 × 10−7 4.8 × 10−5 4.9 × 10−47 1.6 × 10−40 3.7 × 10−43 4.6 × 10−20 8.7 × 10−11 3.4661 3.4917 3.4628 2.4298 2.1886 0.516 0.468 0.422 0.313 0.297 Table 2.5: Numerical outcomes for f4(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f4 (x) Without memory P M1 P M2 P M3 H P M S H M H M 2.2 × 10−1 4.0 × 10−1 4.0 × 10−1 − − − 1.0 × 10−3 2.5 × 10−3 4.5 × 10−3 − − − 1.3 × 10−10 1.0 × 10−9 1.1 × 10−8 − − − 2.8943 2.7963 2.7547 F F F 0.250 0.266 0.328 - - - With memory P M M1 2.2 × 10−1 P M M2 4.0 × 10−1 P M M3 4.0 × 10−1 T M1 − T M2 − 2.5 × 10−3 1.2 × 10−2 1.1 × 10−2 − − 3.7 × 10−10 6.8 × 10−8 5.4 × 10−8 − − 3.4326 3.2771 3.2331 NC NC 0.250 0.250 0.296 - - F- Method fails NC- Not converging in desired iterations Real-life problems: Next, we describe a few real-life problems together with the computa- tional outcomes: Example 2.4.1. Planck's radiation law problem: Firstly, we analyze the well-known Table 2.6: Numerical outcomes for f5(x). |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time Without memory 2.4 × 10−1 4.4 × 10−1 4.8 × 10−1 − − − 4.9 × 10−6 3.2 × 10−4 2.0 × 10−4 − − − 6.0 × 10−21 2.5 × 10−15 3.8 × 10−16 − − − 3.1821 3.5597 3.4779 UR UR UR 0.422 0.406 0.328 - - - With memory P M M1 2.4 × 10−1 P M M2 4.4 × 10−1 P M M3 7.9 × 10−1 T M1 − T M2 − 3.8 × 10−3 1.1 × 10−1 8.7 × 10−2 − − 1.0 × 10−10 1.1 × 10−5 1.6 × 10−5 − − 4.2117 6.8399 4.0907 UR UR 0.360 0.421 0.485 - - UR- Converging to undesired root Planck's radiation law problem (see Jain (2013)): ψ(λ) = 8πcchphpλ−5 , (2.4.6) eλBkT − 1 where λ is the wavelength of radiation, hp is the Planck's constant, T is the absolute temper- ature of the blackbody, c is the speed of light and Bk is the Boltzmann constant. It computes the energy density within an isothermal blackbody. We intend to obtain wavelength λ corre- sponding to maximum energy density ψ(λ). To obtain maximum value of ψ, we take ψ′(λ) = 0 which gives λcBhkpT e λcBhkpT chp = 5. eλBkT − 1 chp Let x = λBkT . Then, (2.4.7) becomes f6(x) = e−x + −

1 = 0. x 5 (2.4.7) (2.4.8) As we find the solutions of f6(x) = 0, we get the maximum wavelength of radiation λ. As stated in Bradie (2006), the L.H.S. of (2.4.8) is zero when x = 5. Also, e−5 ≈ 6.738e − 3. Thus, other root could appear close to x = 5. The desired zero is κ ≈ 4.9651142317442763. Table 2.7: Numerical outcomes for Example 2.4.1. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f6 (x) Without memory P M1 P M2 P M3 H P M S H M H M 2.4 × 10−6 2.2 × 10−6 2.5 × 10−6 1.4 × 10−4 2.5 × 10−4 1.5 × 10−4 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 1.6 × 10−14 1.8 × 10−13 2.0 × 10−14 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 3.0000 3.0000 3.0000 3.0000 3.0000 3.0000 0.392 0.390 0.375 0.313 0.328 0.329 With memory P M M1 P M M2 P M M3 T M1 T M2 2.4 × 10−6 2.2 × 10−6 2.5 × 10−6 1.5 × 10−3 1.5 × 10−3 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 4.8 × 10−10 8.6 × 10−8 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 5.5 × 10−16 3.3297 3.3318 3.3293 2.4006 2.0001 0.359 0.344 0.297 0.282 0.297 Example 2.4.2. Van der Waals state equation: Van der Waals equation of state (see Argyros et al. (2017)) is given by the following equation: an2 (2.4.9) (P + V 2 (V − nb) = nGT .) The following nonlinear equation needs to be solved to attain the volume V of the gas in terms of another parameters: P V 3 − (nbP + nGT)V 2 + an2V − an2b = 0. (2.4.10) Here, G is the universal gas constant, P is the pressure and T is the absolute temperature. If the parameters a and b of a specific gas are given, the values of n, P and T can be calculated. Using certain values, the following nonlinear equation can be obtained: f7(x) = 0.986x3 − 5.181x2 + 9.067x − 5.289 = 0, (2.4.11) having three roots, out of which one is real and two are complex. Though our required zero is κ ≈ 1.9298462428478622. Example 2.4.3. Chemical reactor problem: Here, we consider a problem of fractional conversion in a chemical reactor (see Shacham (1986)) described by the following equation: f8(x) = 1 − x − 5 log 0.4 − 0.5x + 4.45977 = 0. x 0.4(1 − x) (2.4.12) Here, x denotes the fractional conversion of quantities in a chemical reactor. If x is less than zero or greater than one, then above fractional conversion will be of no physical meaning. Hence, x is taken to be bounded in the region 0 ≤ x ≤ 1. Moreover, desired root is κ ≈ 0.7573962462537538. Table 2.8: Numerical outcomes for Example 2.4.2. |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time Without memory 3.1 × 10−3 7.3 × 10−4 2.2 × 10−3 4.1 × 10−4 5.7 × 10−3 4.6 × 10−3 5.6 × 10−7 3.6 × 10−9 1.8 × 10−7 1.4 × 10−10 5.7 × 10−6 2.3 × 10−6 7.3 × 10−18 4.1 × 10−18 4.2 × 10−18 4.1 × 10−18 6.0 × 10−15 3.3 × 10−16 2.9887 2.9976 2.9912 3.0008 2.9774 2.9830 0.234 0.218 0.250 0.282 0.234 0.234 With memory P M M1 3.1 × 10−3 P M M2 7.3 × 10−4 P M M3 2.2 × 10−3 T M1 1.9 × 10−2 T M2 1.9 × 10−2 2.0 × 10−7 2.4 × 10−9 6.4 × 10−8 8.6 × 10−4 3.0 × 10−3 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 6.4 × 10−7 9.9 × 10−5 3.2847 3.3337 3.3027 2.2611 1.7706 0.125 0.187 0.171 0.157 0.203 Table 2.9: Numerical outcomes for Example 2.4.3. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f8 (x) Without memory P M1 P M2 P M3 H P M S H M H M 8.0 × 10−4 5.9 × 10−4 1.1 × 10−3 5.8 × 10−4 2.9 × 10−3 1.9 × 10−5 6.6 × 10−8 1.4 × 10−8 1.7 × 10−7 4.0 × 10−8 1.4 × 10−5 1.0 × 10−13 8.8 × 10−17 8.8 × 10−17 8.8 × 10−17 8.8 × 10−17 1.7 × 10−12 8.8 × 10−17 3.0049 3.0056 3.0096 3.0000 3.0246 3.0000 0.359 0.360 0.376 0.344 0.313 0.296 With memory P M M1 8.0 × 10−4 P M M2 5.9 × 10−4 P M M3 1.1 × 10−3 T M1 1.8 × 10−3 T M2 1.8 × 10−3 1.4 × 10−8 5.5 × 10−9 3.4 × 10−8 2.2 × 10−5 1.1 × 10−4 8.8 × 10−17 8.8 × 10−17 8.8 × 10−17 1.2 × 10−10 4.0 × 10−7 3.2477 3.2580 3.2322 2.7340 2.0345 0.390 0.374 0.297 0.296 0.344 Example 2.4.4. Multi-factor effect: The path traversed by an electron in the air gap between two parallel plates considering the multi-factor effect (see Argyros et al. (2017)) is given by u(t) =u0 + v0 + c0 mω sin ωt0 + β (t − t0) + c0 mω2 (cos(ωt + β) + sin(ωt + β)) , E E0 ( ) (2.4.13) where u0 and v0 are the position and velocity of the electron at time t0, m and c0 are the mass and the charge of the electron at rest and E0 sin(ωt + β) is the RF electric field between the plates. If particular parameters are chosen, (2.4.13) can be simplified as f9(x) = x − cos x + = 0. 1 π 2 4 (2.4.14) The desired root of (2.4.14) is κ ≈ −0.3090932715417949. Table 2.10: Numerical outcomes for Example 2.4.4. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f9 (x) Without memory P M1 P M2 P M3 H P M S H M H M 2.1 × 10−3 8.4 × 10−4 1.8 × 10−3 1.2 × 10−3 2.6 × 10−3 4.2 × 10−3 3.7 × 10−10 1.1 × 10−11 2.5 × 10−10 1.8 × 10−11 1.5 × 10−9 3.6 × 10−9 2.4 × 10−30 3.0 × 10−31 9.2 × 10−31 3.0 × 10−31 2.5 × 10−28 2.4 × 10−27 2.9998 2.9999 2.9998 2.9994 2.9996 3.0001 0.390 0.375 0.376 0.328 0.406 0.375 With memory P M M1 P M M2 P M M3 T M1 T M2 2.1 × 10−3 8.4 × 10−4 1.8 × 10−3 4.0 × 10−2 4.0 × 10−2 5.3 × 10−11 3.2 × 10−12 3.5 × 10−11 8.2 × 10−5 8.3 × 10−4 3.0 × 10−31 3.0 × 10−31 3.0 × 10−31 2.5 × 10−11 3.9 × 10−7 3.3204 3.3363 3.3231 2.4205 1.9767 0.390 0.391 0.359 0.328 0.312 Example 2.4.5. Embedment of a wall: The following nonlinear equation results from the embedment x of a sheet-pile wall: f10(x) = x3 + 2.87x2 − 10.28 4.62 − x = 0. (2.4.15) The required zero of (2.4.15) is κ ≈ 2.0021. Remark 2.4.3. The proposed family with memory (2.3.1) has been compared to some other methods and it is noted that it gives better outcomes in terms of COC and errors as depicted in the tables. There is an obvious increase in the order of convergence. 2.5 Basins of attraction The basins of attraction (see Ardelean (2011)) of the root t∗ of u(t) = 0 is the set of all initial points t0 in the complex plane that converge to t∗ on the application of the given iterative Table 2.11: Numerical outcomes for Example 2.4.5. |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time Without memory 1.1 × 10−3 6.2 × 10−4 1.3 × 10−3 2.6 × 10−6 1.5 × 10−3 1.7 × 10−3 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 3.0002 3.0001 3.0003 3.0000 3.0002 3.0003 0.298 0.327 0.328 0.313 0.297 0.234 With memory P M M1 P M M2 P M M3 T M1 T M2 1.1 × 10−3 6.2 × 10−4 1.3 × 10−3 2.3 × 10−2 2.3 × 10−2 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 4.3 × 10−5 5.4 × 10−4 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 3.2964 3.3102 3.2930 2.5636 2.0023 0.218 0.187 0.234 0.235 0.220 scheme. Our objective is to make use of basins of attraction to examine the comparison of several root finding iterative methods in

the complex plane in terms of convergence and stability of the method. On this front, we have taken a 512 × 512 grid of the rectangle S = [−2,2] × [−2,2] ⊂ C. A color is assigned to each point t0 ∈ S on the basis of the convergence of the corresponding method starting from t0 to the simple root and if the method diverges, black color is assigned to that point. Thus, distinct colors are assigned to the distinct roots of the corresponding problem. It is decided that an initial point t0 converges to a root t∗ when | t∗ − t0 |<10−4. Then, the point t0 is said to belong to the basins of attraction of t∗. Likewise, the method beginning from the initial point t0 is said to diverge if no root is located in a maximum of 25 iterations. We have used MATLAB R2021a software to draw the presented basins of attraction (see Zachary (1996)). Furthermore, Tables 2.12 and 2.13 list the average of iterations (AI) denoted by Avg Iter, percentage of non-converging points denoted by PNC and the total CPU time taken by the methods to generate the basins of attraction. To carry out the desired comparisons, we have considered the test problems given below: Problem 2.5.1. The first function considered is p1(z) = z2+1. The roots of this function are i, −i. The basins corresponding to our proposed methods and the mentioned existing methods are shown in Figures 2.1 and 2.2. It is observed that PM1, PM2, PM3, PMM1, PMM2 and PMM3 converge to the root with no diverging points but HPM, SHM, TM1 and TM2 have some points painted as black. Problem 2.5.2. Second function taken is p2(z) = z3 + 1 having roots −1, 0.5 + 0.866i, 0.5−0.866i. Figures 2.3 and 2.4 show the basins for p2(z) in which it can be seen that SHM, TM1 and TM2 have wider regions of divergence. Table 2.12: Comparison of different methods without memory in terms of Avg Iter, PNC and CPU time Without memory methods Avg Iter PN C CPU time p1(z) P M1 P M2 P M3 H P M S H M p2(z) P M1 P M2 P M3 H P M S H M p3(z) P M1 P M2 P M3 H P M S H M p4(z) P M1 P M2 P M3 H P M S H M 2.7165 2.2822 2.6060 2.5187 2.5187 3.1603 2.7433 2.8910 2.2958 7.1570 3.9678 3.4784 3.6042 2.9863 8.9566 3.5381 3.3973 3.4684 3.4316 4.7686 0 0 0 4.0 × 10−6 4.0 × 10−6 0 0 0 4.0 × 10−6 1.5 × 10−1 1.3 × 10−2 8.2 × 10−3 7.8 × 10−3 4.0 × 10−6 1.1 × 10−1 0 0 0 0 3.0 × 10−4 4.0501 3.5181 3.8734 2.7963 3.0479 5.6380 4.9187 5.0015 2.7867 7.9437 7.2858 6.3408 6.7897 3.4503 10.4044 6.2320 6.0172 6.0730 3.7895 6.3023 Problem 2.5.3. The third function considered is p3(z) = z4 − 1 having roots ±1, ±i. Fig- ures 2.5 and 2.6 show that SHM and TM2 have smaller basins. Although PM1, PM2, PM3 and PMM1 have some diverging points, yet they converge faster than the existing methods. Problem 2.5.4. The fourth function we have taken is p4(z) = z5 − z whose roots are 0, ±1, ±i. Figures 2.7 and 2.8 show that PM1, PM2, PM3, HPM, PMM1, PMM2 and PMM3 depict convergence to the root for any initial point as they have no diverging points. Remark 2.5.1. One can see from Figures 2.1−2.8 and Tables 2.12 and 2.13 that there is a marginal increase in the average number of iterations per point of the existing methods, as they have more number of divergent points than that of the proposed methods. Special mention to the fact that our proposed with memory methods have negligible number of divergent points in the specified mesh of points and hence, larger basins of attraction. Consequently, our proposed family with memory shows faster convergence in comparison to the existing methods. Table 2.13: Comparison of different methods with memory in terms of Avg Iter, PNC and CPU time With memory methods Avg Iter PN C CPU time p1(z) P M M1 P M M2 P M M3 T M1 T M2 p2(z) P M M1 P M M2 P M M3 T M1 T M2 p3(z) P M M1 P M M2 P M M3 T M1 T M2 p4(z) P M M1 P M M2 P M M3 T M1 T M2 2.7089 2.4002 2.5916 4.3642 8.3338 3.1132 2.7755 2.8498 6.0252 11.9211 3.6980 3.3119 3.3200 8.3110 15.0794 3.7034 3.6684 3.6692 5.4714 9.1054 0 0 0 1.9 × 10−3 1.4 × 10−1 0 0 0 1.1 × 10−3 2.8 × 10−1 3.0 × 10−5 0 0 4.7 × 10−2 4.1 × 10−1 0 0 0 1.5 × 10−3 8.0 × 10−2 5.2390 4.6904 5.1093 3.2522 5.6000 6.8290 6.2775 6.3416 5.2045 9.6673 8.2979 7.5358 7.7738 6.7431 11.8306 8.2241 8.2776 8.2856 4.5927 7.0686 2.6 Conclusions In this chapter, we have contributed further to the literature by introducing a new family of iterative methods with memory having higher order of convergence in comparison to the Hansen-Patrick’s family and Traub’s method. For verification, we have carried out numerical experiments on a few test functions and some real life problems. It is clearly visible from our results that the proposed family improves the convergence order. This increase in the convergence order has been achieved with no additional functional evaluation. Furthermore, we have also presented the basins of attraction for the proposed as well as some existing methods, which point to the very fact that our proposed methods converge largely to the desired zeros over a specified region much faster. Finally, to conclude we would say that the proposed family can be significantly used for solving nonlinear equations. Figure 2.1: Basins of attraction for PM1, PM2, PM3, HPF, SHM, respectively for p1(z) Figure 2.2: Basins of attraction for P M M1, P M M2, P M M3, T M1, T M2, respectively for p1(z) Figure 2.3: Basins of attraction for PM1, PM2, PM3, HPF, SHM, respectively for p2(z) Figure 2.4: Basins of attraction for P M M1, P M M2, P M M3, T M1, T M2, respectively for p2(z) Figure 2.5: Basins of attraction for PM1, PM2, PM3, HPF, SHM, respectively for p3(z) Figure 2.6: Basins of attraction for P M M1, P M M2, P M M3, T M1, T M2, respectively for p3(z) Figure 2.7: Basins of attraction for PM1, PM2, PM3, HPF, SHM, respectively for p4(z) Figure 2.8: Basins of attraction for P M M1, P M M2, P M M3, T M1, T M2, respectively for p4(z) Chapter 3 Chebyshev-Halley Typ e Variants with Memory and Their Stability Analysis In this chapter, a new iterative family with memory has been presented for solving nonlinear equations numerically in order to achieve higher order of convergence in comparison to the cubically convergent Chebyshev-Halley type method. The acceleration of convergence speed has been attained using a self accelerating parameter which is estimated from the current and previous iterations using divided differences. Therefore, the order of convergence increases from 3 to 3.30 without any further functional evaluation. In addition, the complex and real dynamics of the proposed family have been studied. The parameter spaces and dynamical planes are presented. This study helps to determine the family members with stable behavior which in turn are suitable for practical problems. 3.1 Introduction One of the various higher order schemes which we have considered is the Chebyshev- Halley's family of order 3 given by xn+1 = xn − 1+ 1 Lf (xn) f(xn) 2 (1 − αLf (xn)) (3.1.1) [ )] f′(xn), α ∈ R, n = 0,1,2,..., where Lf (xn) = f′′(xn)f(xn) f′2(xn) . This family comprises of Chebyshev's method for (α = 0), Halley's method for (α = 12) and super-Halley's method for (α = 1). As stated in the previous chapter, the involvement of second order derivative makes the family less practical in terms of computations. Due to these restrictions, the researchers are being interested towards The contents of this chapter are published in: Mathematical Methods in the Applied Sciences (Wiley), 46(12), 12549–12569, 2023 (SCIE, Q1, Impact Factor: 2.1) the development of multipoint methods for solving nonlinear equations. Many second order derivative free variants of Chebyshev-Halley's method have been introduced. Thus, by taking into consideration this motivation, we further intend to increase the order of convergence of the proposed family with the help of a self accelerating parameter in each iteration. As a result, the R-order of convergence of the proposed family increases from 3 to 3.30 without any further functional evaluation. However, another important aspect of an iterative scheme to be considered is its stability which is the analysis that tells us how dependent the scheme of the initial guesses used is. The dynamical performance of the rational functions associated to iterative schemes is a very useful element to study their dependence on initial estimations. In recent years, complex discrete dynamics has been widely used on iterative methods without memory (see Cordero et al. (2021, 2013d); Chicharro et al. (2013a)). Nevertheless, it is known that iterative schemes with memory cannot be analyzed by means of these techniques. This is the reason why the authors focused on their qualitative study by transforming them into multidimensional dynamical systems (see Campos et al. (2017); Chicharro et al. (2019); Campos et al. (2015)). We make use of the dynamical tools on iterative schemes with and without memory for solving nonlinear equations. We carry out the study of the behavior of a rational function associated with an iterative scheme. The dynamical properties of the rational function provide us with important infor- mation about the stability and reliability of the corresponding method. The dynamical planes show this behavior by taking into consideration the basins of attraction of the fixed points, periodic points, etc. of the rational function considered. A basin of attraction allows us to visually interpret how a method works based on several initial estimates. To this end, we firstly develop a new iterative family without memory and extend it to one with memory. We carry out the convergence analysis of the new families in order to demonstrate their order of convergence. Stability analysis of the proposed iterative family has also been carried out. To illustrate our theoretical results, numerical results for the proposed families and comparisons with some of the existing methods are then given. Lastly, we present the basins of attraction of the new families depicting clearly the convergence or divergence of the new as well as the existing methods. 3.2 An iterative family with memory and its conver- gence analysis We aim to construct a new two-point Chebyshev-Halley type method with memory. Suppose yn = xn − f ′(xn) be the Newton's iterate. Expanding f (yn) about a point x = xn f (xn) by Taylor series, we get f(yn) ≈ f(xn) + f′(xn)(yn − xn) + 1f′′(xn)(yn − xn)2, 2 ⇒f′′(xn) ≈ 2f′(xn)2f(yn) f(xn)2 . Now, using this new approximation for f ′′(xn) in (3.1.1), Li et al. (2014) presented the following scheme, yn = xn − f ′(xn) f (xn) , (3.2.1) xn+1 = yn − (f (xn)f−(y2nα)f (yn)) ff′((xxnn)) , α ∈ R. Now, in order to extend to the method with memory, we come up with an idea of intro- ducing a parameter t in (3.2.1) and we present a modification in this method as follows: yn = xn − f (xn) f ′(xn) + tf (xn) , xn+1 = yn − f (yn) f (xn) (f (xn) − 2αf (yn)) (f ′(xn) + tf (xn)) , n = 0, 1, 2, . . . , where α, t ∈ R are free parameters. This scheme yields the convergence order 3 having the following error relation, en+1 = −(t + d2) t(−1 + 2α) + 2(−1 + α)d2 e3n + O(en)4, (3.2.2) (3.2.3) ( ) where en = xn − κ, κ is a simple root of f (x) = 0 and dn = n! f ′(κ) , n = 2, 3, . . . 1 f (n)(κ) If we observe clearly, it can be seen from the error equation (3.2.3), if t = −d2 = − 2f ′(κ) f ′′(κ) , then the order of convergence of the presented scheme (3.2.2) can possibly be improved, but this value can't be reached because the values of f ′(κ) and f ′′(κ) are not practically available. Instead, we can use approximations calculated by already available information. So, to im- prove the convergence order, we give an estimation for tn given by tn = − 1 f ′(xn) − f ′(xn−1) 2 (f (xn) − f (xn−1) .) So, by replacing t by tn in the method (3.2.2), we obtain a new family with memory using the two previous iterations x0, x1 as follows: tn = − 1 f ′(xn) − f ′(xn−1) 2 , (f (xn) − f (xn−1) yn = xn − f (xn)) f ′(xn) + tnf (xn) , (3.2.4) xn+1 = yn − (f (xn) − 2αf (yn)) (f ′(xn) + tnf (xn)) , α ∈ R, n = 1, 2, . . . f (yn) f (xn) Next, we establish the convergence results for our proposed family with memory (3.2.4). 3.2.1 Convergence analysis Theorem 3.2.1. Suppose that f : D ⊂ R → R be a real function suitably differentiable in a domain D. If κ ∈ D is a simple root of f(x) = 0 and an initial guess x0 is sufficiently close to κ, then the iterative method (3.2.4) converges to κ with convergence order at least 3.30 having the following error relation, en+1 = d32 − 3d2d3 en−1e3n + d2d3 e4n + O5(en−1en), (2) 2 where en = xn − κ and dn = n! f ′(κ) , n = 2, 3, . . . 1 f (n)(κ) Proof. Using Taylor series expansion about xn = κ, we get f (xn−1) = f ′(κ)(en−1 + d2e2n−1 + d3e3n−1 + d4e4n−1 + d5e5n−1) + O(en−1)6, f (xn) = f ′(κ)(en + d2e2n + d3e3n + d4e4n + d5e5n) + O(en)6. Then, f ′(xn−1) = f ′(κ)(1 + 2d2en−1 + 3d3e2n−1 + 4d4e3n−1 + 5d5e4n−1) + O(en−1)5, f ′(xn) = f ′(κ)(1 + 2d2en + 3d3e2n + 4d4e3n + 5d5e4n) + O(en)5. Now, using previous developments, we have (3.2.5) (3.2.6) (3.2.7) (3.2.8) (3.2.9) tn = − 1 f ′(xn) − f ′(xn−1) 2 = − d2 + d22 − 3d3 (f (xn) − f (xn−1)) + (2 en−1 + − d32 + 5d2d3 e2n−1 d22 − 3d3 + − 2d32 + 4d2d3 − 2d4 en−1 + 3d42 − 17d22d3 + 3d23 + 5d2d4 − 5d5 e2n−1 ( ) (2)) ((2) en + − 2d4 + 3d42 − 17d22d3 + 3d23 + 5d2d4 − 5d5 en−1 + − 6d52 + 21d32d3 − 15d2d23 − 12d22d4) (2 2 − d32 (+ 5d2d3)) ((2) (2 2) + 9d3d4 + 6d2d5 e2n−1 e2n + O3(en−1en). ( )) (3.2.10) Using (3.2.7), (3.2.9), (3.2.10) in the second step of (3.2.4), we get yn − κ = d22 − 3d3 en−1 + − d32 + 5d2d3 − 2d4 e2n−1 e2n + d3 − 2 d32 − 2d2d3 ((2) (2)) (2 + d4 en−1 + 8d42 − 22d22d3 + 3d23 + 20d2d4 − 10d5 e2n−1 e3n + O4(en−1en). 1 (4) ( )) (3.2.11) Then, using (3.2.11) in (3.2.7), we get f (yn) =f ′(κ) d22 − 3d3 en−1 + ((2) (− d32 + 5d2d3 − 2d4 e2n−1 e2n + d23 − 2 d32 − 2d2d3 2 1)) (+ d4 en−1 + 8d42 − 22d22d3 + 3d23 + 20d2d4 − 10d5 e2n−1 e3n + O4(en−1en). (4) ( ))) (3.2.12) Using (3.2.7), (3.2.9)-(3.2.12) in the third step of (3.2.4), we finally get en+1 = d32 − 3d2d3 en−1e3n + d2d3 e4n + O5(en−1en). (2) 2 (3.2.13) Now, we can see the lowest term of the error equation is d32 − 3d2d3 en−1e3n, therefore, by Theorem 1.3.1, the unique positive root o√f the polynomial (s2 − 3s − 1 gives the R-order of the 2) proposed scheme (3.2.4), which is s = 3+2 13 ≈ 3.30. This completes our proof. Remark 3.2.1. It is observed that the order of convergence has increased without the addition of any further functional evaluation. Moreover, our family with memory has a very simple body structure in comparison to the general complex structures of iterative schemes with memory. 3.3 Complex dynamics To present the complex dynamical analysis of our proposed scheme without memory (3.2.2), we construct a rational operator associated with the scheme on a low-degree nonlinear polynomial and analyze the stability and convergence of the corresponding fixed points and critical points. Then, we generate the parameter planes of the free critical points and generate dynamical planes of the method for some optimal choices of the parameters involved. The fixed point operator corresponding to the proposed family (3.2.2) is Sp(z) = y − f (y) f (z) f (z) − 2αf (y) f ′(z) + tf (z) . (3.3.1) ( ) ( ) Here, y = z − f (z) f ′(z) + tf (z) , the iteration xn is denoted by z. Also, α, t ∈ C are parameters. Here, we study the dynamics of this operator when it is applied to a quadratic polynomial p(z) = z2 − 3. It is known that the roots of a polynomial can be transformed by an affine map with no qualitative changes on the dynamics of the concerned family. By some previous propositions, we can use the conjugacy map, z − √ 3 h(z) = √ , z + 3 (3.3.2) with the following properties: i) h(∞) = 1, ii) h(3) = 0, iii) h(− 3) = ∞, √ √ For z = C1 For z = C2 For z = C3 For z = C4 For z = C5, C6 For z = C7, C8 Figure 3.1: Parameter planes for Ci, i = 1, 2, . . . , 8 to prove that the operator Sp(z) can be conjugated to the operator Gp(z) = h ◦ Sp ◦ h−1 (z) = n1(z; α, t) , (3.3.3) n2(z; α, t) where n1 and n2 are rational polynomials whose coefficients are dependent on parameters α ( ) and t. One of our goal is to make them minimally dependent on parameters. We also take a particular case for the parameter t which is t = 1, which corresponds to the following fixed point operator: Gp(z; α) − z = z(z − 1)2n3(z; α) n4(z; α) , (3.3.4) where n3(z; α) =1 + z(1 + 6 3 − 2(1 + 2 3)α) − 20z2(−1 + α) + z3(1 − 6 3 + (−2 + 4 3)α) + z4 √ √ √ √ n4(z; α) = − 1 + z(1 − 6 3 + (2 + 4 3)α) + z2(−33 + 6 3 + 42α) + z3(19 − 18 3 + 18(−1 √ √ √ √ + 2 3)α) + 2z4(7 − 3 3 + (−13 + 4 3)α). √ √ √ Cl√early, 0 and ∞ are two super-attracting fixed points of Gp corresponding to roots 3 and √ − 3 of the polynomial p(z). Also, they are free from parameter α. Fixed points excluding these two are strange fixed points which are different from the roots of p(z). In order to find further strange fixed points, we will solve Gp(z; α) − z = 0 for z with given values of α. Theorem 3.3.1. (a) For α = , Gp(z) has 3 different fixed points which are z = 1, z = 1 2 + 3 and z = −2 + 3. √ √ 2 (b) For α = 1, Gp(z) has 4 different fixed points which are z = 1, z = 4.319637, z = −0.427768 ± 0.220263i. (c) For α ≠ 1, , Gp(z) has 5 different fixed points which are z = 1 and the four roots of 1 n3(z; α) = 0. { 2 } Proof. (a) Let n3(z) = 0 and n4(z)√= 0 for some values of z ∈ C. By elim√inating α√between these two equations, we get −2 3z + z2 − 1 = 0 which gives z = 2 + 3, −2 + 3. Now, n3(2 + 3) = 0 and n4(2 + 3) = 0 which gives α = . When we put α = in Gp(z), √ √ 1 1 we get 3 strange fixed points. 2 2 (b) We will check if z − 1 is factor of n3(z) and n4(z). If we put√z = 1 in n3(z), we get n3(1) = −24(−1 + α) = 0 which gives α = 1. Also, n4(1) = 24 3(−1 + 2α) = 0 which gives α = . So, n3(z) has factor z − 1 for α = 1 and n4(z) has factor z − 1 for α = 1 1 2 . When we put α = 1 in Gp(z), we get 4 strange fixed points. 2 (c) The proof for this is straightforward. We can numerically find the fixed points for α ≠ 1, as stated earlier. 1 { 2 } Further, to study the stability of the fixed points, we will find the derivative of the operator Gp from (3.3.4) as Gp′(z; α) = 2zn26n(5z(;zα;)α) , (3.3.5) where n5(z; α) =(21 + 9 3 − 3(13 + 4 3)α + 2z(66 + 57 3 − (127 + 144 3)α + (74 + 60 3)α2) √ √ √ √ √ + 6z2(92 + 31 3 − 4(78 + 19 3)α + 16(13 + 4 3)α2) + 6z3(26 + 19 3 − (91 √ √ √ √ + 176 3)α + 98(1 + 2 3)α2) − 2z4(429 − 983α + 256α2) − 6z5(−26 + 19 3 √ √ √ + (91 − 176 3)α + 98(−1 + 2 3)α2) − 6z6(−92 + 31 3 + (312 − 76 3)α √ √ √ √ + 16(−13 + 4 3)α2) − 2z7(−66 + 57 3 + (127 − 144 3)α + (−74 + 60 3)α2) √ √ √ √ + 3z8(7 − 3 3 + (−13 + 4 3)α)) and √ √ n6(z; α) =(−1 + z2(−33 + 6 3 + 42α) + z3(19 − 18 3 + 18(−1 + 2 3)α) + 2z4(7 − 3 3 √ √ √ √ + (−13 + 4 3)α) + z(1 − 6 3 + (2 + 4 3)α))2. √ √ √ Now, for the strange fixed point z = 1, we have |G′p(1)| = 1 which implies that z = 1 is parabolic point. The stability of the remaining strange fixed points depends on parameter α. A classical result establishes that there is atleast one critical point associated with each invariant Fatou component. The critical points of Gp(z; α) are the roots of G′p(z; α) = 0 which are z = 0, z = ∞ and the roots of equation n5(z; α) = 0. As z = 0 and z = ∞ are both superattractive fixed points of Gp(z; α), they are critical points too and give rise to their respective Fatou components. For the other critical points, we can establish the following result: Theorem 3.3.2. (a) For α = , Gp(z) has 2 critical points, −0.5974 and 3.0324. 1 2 (b) For α = 1, Gp(z) has 8 critical points, denoted by Ci, i = 1, 2, . . . , 8 given by C1 = −0.9282, C2 = −0.3648, C3 = 0.3442, C4 = 2.5866, C5 = −0.4600 + 0.3338i, C6 = −0.4600 − 0.3338i, C7 = 4.3089 + 0.1932i and C8 = 4.3089 − 0.1932i. (c) For α ≠ 1, , Gp(z) has 8 critical points which are the roots of n5(z; α) = 0. 1 { 2 } Proof. (a) When we put α = , we get 1 2 1 = 9z2(1 + 2 3 + 6z + (1 − 2 3)z2) √ √ Gp′ z; √ (2 3 − (−6 + 3)z √ 2 .) (54) Now, on solving the equation G′p z; 1 z = 0 which are z = −0.5974 and z(= 3.)0324. 2 = 0, we obtain two critical points other than (b) On the similar lines,when we put α = 1, we get Gp′ (z; 1) = (−1 + (3 − 2 3)z + (9 + 6 3)z2 + (1 + 18 3)z3 + 2(−6 + 3)z4)2 √ √ 2z2S1 √ √ , S1 = (−3(6 + 3) + (26 − 54 3)z + 6(−12 + 19 3)z2 + 18(11 + 13 3)z3 + 596z4 + (198 √ √ √ √ − 234 3)z5 − 6(12 + 19 3)z6 + (26 + 54 3)z7 + 3(−6 + 3)z8). √ √ √ √ On solving the equation G′p (z; 1) = 0, we obtain eight critical points other than z = 0 which are C1 = −0.9282, C2 = −0.3648, C3 = 0

.3442, C4 = 2.5866, C5 = −0.4600 + 0.3338i, C6 = −0.4600 − 0.3338i, C7 = 4.3089 + 0.1932i and C8 = 4.3089 − 0.1932i. (c) The critical points for any value of α ≠ 1, can also be found numerically as stated 1 earlier. { 2 } 3.3.1 Dynamical planes and parameter spaces The dynamical behavior of operator Gp depends on the values of parameter α. A parame- ter plane 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 family (3.2.2) associated with this α using one of the free critical points Ci as initial estimate. The resulting figure is made in MATLAB R2021a programming package with a resolution of 400 × 400 pixels. If a method converges to any of the roots starting from Ci in a maximum of 40 iterations with a tolerance of 10−3, the pixel is colored red; in other cases, the pixel is colored black which can be seen in Figure 3.1. Each value of α that belongs to the same connected component of the parameter plane 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. 3.3.2 Dynamical planes The behavior of the fixed points can be illustrated through the dynamical planes. The dynamical planes for some of the values of the parameter α are shown in Figure 3.2. These figures have been generated using MATLAB R2021a with a resolution of 400 × 400 pixels. We have taken maximum iterations 40 with a tolerance of 10−3. Based on these figures, we can analyze that for α = , there are no black areas of non-convergence to the solution. Hence, this 1 2 method shows good dynamical behavior. It is very stable. Further, for α = 1, 5 11 2 , 3 3 6 , 5 and , 5 For α = 12 For α = 1 For α = 53 For α = 161 For α = 25 For α = 35 Figure 3.2: Dynamical planes there are black areas of slow convergence of the methods. Hence, these methods show poor dynamical behavior. 3.4 Numerical results This section lays out the comparison of our family (3.2.4) with several existing schemes. The initial values of α and t (or t0) are assumed to be chosen beforehand to begin with the computations. Also, a suitable x0 must be fixed. The following members of the family (3.2.4) are chosen in order to perform the calculations: 1. P M1 for α = 12 . 2. P M2 for α = 19050 . 3. P M3 for α = 1. We have taken t (or t0)= 0.01 in our computations. The following existing methods with memory have been selected to facilitate comparisons with our methods: 1. Chicharro-Cordero method (M M1) Chicharro et al. (2019): tn = 2N ′(xn) −N ′′(xn) , yn = xn − f (xn) f ′(xn) + tnf (xn) , (3.4.1) xn+1 = yn − f′(xn), n ∈ N, f(yn) where N (x) = f (xn) + f [xn, xn−1](x − xn) + f [xn, xn−1, yn−1](x − xn)(x − xn−1) is the Newton's interpolating polynomial of second degree. 2. Modified Traub's method (M M2) Soleymani et al. (2015): wn = xn + tnf (xn), yn = xn − f [xn, wn] f (xn) , xn+1 = yn − f [xn, wn] , n = 0, 1, 2, . . . f (yn) (3.4.2) −1 tn = f[xn,xn−1], n = 1,2,... 3. Modified parametric family (M M3) Soleymani et al. (2015): wn = xn + tnf (xn), yn = xn − f [xn, wn] αf (xn) , vn = xn − f (yn) + αf (xn) f [xn, wn] , (3.4.3) xn+1 = xn − f (vn) + f (yn) + αf (xn) f [xn, wn] , n = 0, 1, 2, . . . , α ∈ R, −2 tn = f [xn, xn−1] , n = 1, 2, . . . 15 We have taken a particular case for this family when α = 10 . Further, Table 3.1 displays some nonlinear functions (f1 to f4) used to carry out the compu- tations. Table 3.1: Test functions, associated zeros and the initial approximations (x0) Function Real zero x0 f1(x) = x3 − sin2 x + 3 cos x + 5 f2(x) = ex2+7x−30 − 1 f3(x) = x3 + 1 + ex3−x − cos(x2 − 1) f4(x) = e−x2(x − 2)(x6 + x3 + 1) −1.5827 3 −1 2 −0.9 3.15 −0.9 1.9 In addition, some real-life problems are also solved after transforming them to nonlinear functions (f5 to f8). COC (computational order of convergence) and the errors of approxima- tions to the desired roots, | xn − κ | for n = 1, 2, 3 of ft(x), t = 1, 2, . . . , 10 are outlined in Tables 3.2–3.9. Table 3.2: Numerical outcomes for f1(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f1 (x) With memory P M1 P M2 P M3 M M1 M M2 M M3 7.3 × 10−2 1.0 × 10−2 1.7 × 10−2 2.7 × 10−1 3.4 × 10−1 1.5 × 10−1 1.1 × 10−5 3.1 × 10−8 1.5 × 10−7 6.4 × 10−4 1.2 × 10−3 1.4 × 10−5 3.3 × 10−17 3.6 × 10−17 3.6 × 10−17 2.1 × 10−12 2.2 × 10−11 2.5 × 10−17 3.3495 3.4001 3.3918 3.2889 3.1911 2.9844 0.0049 0.0048 0.0048 0.0059 0.0036 0.0068 Table 3.3: Numerical outcomes for f2(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f2 (x) With memory P M1 5.3 × 10−2 P M2 2.7 × 10−2 P M3 2.1 × 10−2 M M1 6.4 × 10−2 M M2 8.7 × 10−2 M M3 6.3 × 10−2 5.8 × 10−4 3.7 × 10−5 1.3 × 10−5 1.7 × 10−2 1.3 × 10−2 5.0 × 10−3 2.7 × 10−11 4.6 × 10−15 1.7 × 10−16 1.7 × 10−4 5.4 × 10−5 2.3 × 10−6 3.4625 3.3623 3.3326 2.3424 2.2481 2.5584 0.0028 0.0022 0.0021 0.0029 0.0025 0.0037 Table 3.4: Numerical outcomes for f3(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f3 (x) With memory P M1 3.2 × 10−4 P M2 3.4 × 10−5 P M3 7.1 × 10−5 M M1 7.7 × 10−4 M M2 8.6 × 10−4 M M3 1.0 × 10−3 2.1 × 10−12 3.3 × 10−15 3.2 × 10−14 3.1 × 10−11 5.1 × 10−12 3.9 × 10−12 2.0 × 10−39 7.6 × 10−49 1.4 × 10−45 1.8 × 10−35 7.2 × 10−39 2.0 × 10−39 3.3065 3.3620 3.3542 3.2798 3.2616 3.2368 0.0036 0.0032 0.0034 0.0048 0.0030 0.0046 Table 3.5: Numerical outcomes for f4(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f4 (x) With memory P M1 8.0 × 10−4 P M2 2.3 × 10−4 P M3 1.6 × 10−4 M M1 1.4 × 10−3 M M2 1.4 × 10−3 M M3 1.4 × 10−3 1.6 × 10−10 3.1 × 10−12 1.0 × 10−12 6.3 × 10−10 4.2 × 10−10 3.2 × 10−10 1.0 × 10−32 2.2 × 10−38 5.9 × 10−40 6.6 × 10−31 1.8 × 10−31 4.0 × 10−32 3.3103 3.3212 3.3260 3.3131 3.2780 3.2856 0.0035 0.0036 0.0036 0.0038 0.0036 0.0038 Remark 3.4.1. We have tested our proposed family of iterative methods for several values of the parameter α out of which the best ones (the values for which we got best results) are selected for numerical computations. Real-life problems: Next, we describe a few real-life problems together with the computa- tional outcomes: Example 3.4.1. Planck's radiation law problem: As described in Example 2.4.1, the nonlinear equation for the said problem is as follows: f5(x) = e−x + − 1 = 0. x 5 (3.4.4) The desired zero is κ ≈ 4.9651142317442763. Table 3.6: Numerical outcomes for Example 3.4.1. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f5 (x) With memory P M1 P M2 P M3 M M1 M M2 M M3 2.4 × 10−3 1.7 × 10−3 2.1 × 10−3 6.6 × 10−3 5.3 × 10−3 5.6 × 10−3 4.6 × 10−12 2.8 × 10−12 5.9 × 10−12 9.6 × 10−11 1.6 × 10−12 8.6 × 10−13 2.8 × 10−16 2.8

× 10−16 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 2.8 × 10−16 3.3297 3.3768 3.3774 3.3004 3.2655 3.2513 0.0022 0.0022 0.0021 0.0025 0.0020 0.0028 Example 3.4.2. Van der Waals state equation: As described in Example 2.4.2, the nonlinear equation for the said problem is as follows: f6(x) = 0.986x3 − 5.181x2 + 9.067x − 5.289 = 0, (3.4.5) having three roots, out of which one is real and two are complex. Though our required zero is κ ≈ 1.9298462428478622. Table 3.7: Numerical outcomes for Example 3.4.2. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f6 (x) With memory P M1 P M2 P M3 M M1 M M2 M M3 1.4 × 10−10 1.3 × 10−11 1.3 × 10−13 2.7 × 10−10 2.7 × 10−10 3.4 × 10−10 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 4.1 × 10−18 3.2821 3.3008 3.3298 3.2766 3.2795 3.2734 0.0007 0.0006 0.0006 0.0009 0.0006 0.0010 Example 3.4.3. Multi-factor effect: As described in Example 2.4.4, the nonlinear equation for the said problem is as follows: f7(x) = x − cos x + = 0. 1 π 2 4 (3.4.6) The desired root of (3.4.6) is κ ≈ −0.309093271541794952741986808924. Table 3.8: Numerical outcomes for Example 3.4.3. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f7 (x) With memory P M1 P M2 P M3 M M1 M M2 M M3 4.3 × 10−3 6.8 × 10−4 2.4 × 10−4 7.5 × 10−3 7.4 × 10−3 6.5 × 10−3 4.6 × 10−10 1.7 × 10−12 7.2 × 10−14 1.9 × 10−9 3.6 × 10−9 1.6 × 10−9 3.1 × 10−31 3.0 × 10−31 3.0 × 10−31 1.2 × 10−30 8.3 × 10−30 6.2 × 10−31 3.3075 3.3389 3.3539 3.2366 3.2772 3.2900 0.0024 0.0026 0.0027 0.0030 0.0024 0.0037 Example 3.4.4. Embedment of a wall: The following nonlinear equation results from the embedment x of a sheet-pile wall, as described in Example 2.4.5: x3 + 2.87x2 − 10.28 f8(x) = 4.62 − x = 0. (3.4.7) The required zero of (3.4.7) is κ ≈ 2.0021. Table 3.9: Numerical outcomes for Example 3.4.4. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f8 (x) With memory P M1 P M2 P M3 M M1 M M2 M M3 2.3 × 10−3 1.2 × 10−5 2.4 × 10−4 5.4 × 10−3 5.6 × 10−3 8.6 × 10−3 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 3.2769 3.3716 3.3299 3.2195 3.2496 3.1871 0.0005 0.0006 0.0006 0.0008 0.0005 0.0015 Remark 3.4.2. The proposed with memory scheme (3.2.4) has been compared to some other methods and it is noted that our proposed scheme with memory gives better outcomes in terms of COC and errors as depicted in the tables. It can be seen from Tables 3.2–3.5 that for the functions f1, f2, f3 and f4, the proposed methods P M1, P M2 and P M3 converge to the desired root with error of approximations much lower than the existing methods M M1, M M2 and M M3. For the function f2, the methods M M1, M M2 and M M3 have low convergence order in comparison to the methods P M1, P M2 and P M3. Table 3.10: Comparison of different methods with memory in terms of Avg Iter, PNC and CPU time Methods Avg Iter PN C CPU time p1(z) P M1 P M2 P M3 M M1 M M2 M M3 p2(z) P M1 P M2 P M3 M M1 M M2 M M3 p3(z) P M1 P M2 P M3 M M1 M M2 M M3 p4(z) P M1 P M2 P M3 M M1 M M2 M M3 3.0368 2.9180 2.8824 4.2048 3.6256 6.0583 3.6516 3.5149 3.4624 6.5860 6.6945 8.4984 4.1059 3.7720 3.6980 8.0397 8.4718 11.5663 3.8285 3.7316 3.6943 5.7324 4.5185 5.5617 0 1.6 × 10−3 6.6 × 10−4 4.0 × 10−6 7.5 × 10−3 1.4 × 10−1 1.6 × 10−4 1.3 × 10−2 1.1 × 10−2 1.2 × 10−2 1.2 × 10−1 1.1 × 10−1 2.6 × 10−3 1.6 × 10−2 1.5 × 10−2 6.3 × 10−2 2.0 × 10−1 3.7 × 10−1 1.7 × 10−3 2.2 × 10−3 1.9 × 10−3 5.4 × 10−3 2.4 × 10−2 7.0 × 10−2 4.0254 3.9590 3.7584 7.8023 4.4682 5.9984 5.6410 5.3589 5.5212 13.5069 8.8731 16.5921 6.6177 6.1735 6.1986 17.0593 10.5184 9.8958 6.1362 6.0548 6.0116 11.8286 6.2708 10.6208 3.5 Basins of attraction Now, we will study the dynamics of the proposed as well as some existing methods by analyzing the behavior of their basins of attraction in the complex plane. Table 3.10 lists the average number of iterations denoted by Avg Iter, percentage of non- converging points denoted by PNC and the total CPU time taken by the methods to generate the basins of attraction. We have taken the initial approximation for the accelerating param- eter t0 = 0.01 while plotting the basins of attraction. To carry out the desired comparisons, we have considered the test problems given below: Figure 3.3: Basins of attraction for P M1, P M2, P M3, M M1, M M2, M M3, respectively for p1(z) Figure 3.4: Basins of attraction for P M1, P M2, P M3, M M1, M M2, M M3, respectively for p2(z) Problem 3.5.1. The first function considered is p1(z) = z2 − 1. The roots of this function are ±1. The basins corresponding to our proposed method and the mentioned existing methods are shown in Figure 3.3. It is observed that P M1 converge to the root with no diverging point, P M2, P M3 and M M1 converge to the root with a small number of diverging points but M M2 and M M3 have many points painted as black. Problem 3.5.2. Second function taken is p2(z) = z3 − 1 having roots 1, −0.5 ± 0.866i. Figure 3.4 shows the basins for p2(z) in which it can be seen that M M1, M M2 and M M3 have wider regions of divergence. Problem 3.5.3. The third function considered is p3(z) = z4 + 1 having roots −0.707 ± 0.707i, 0.707 ± 0.707i. Figure 3.5 shows that M M1, M M2 and M M3 have smaller basins. Although P M1, P M2 and P M3 have some diverging points, yet they converge faster than the existing methods. Problem 3.5.4. The fourth function we have taken is p4(z) = z5 + z whose roots are 0, −0.707 ± 0.707i, 0.707 ± 0.707i. Figure 3.6 shows the basins for p4(z). Figure 3.5: Basins of attraction for P M1, P M2, P M3, M M1, M M2, M M3, respectively for p3(z) Remark 3.5.1. One can see from the Figures 3.3–3.6 that the existing methods have some- what darker basins as they have more number of divergent points than that of the proposed method in the specified mesh of points. Hence, our proposed method with memory has larger basins of attraction. Remark 3.5.2. As we concentrate on the number of iterations, it can be seen from the Table 3.10 that there is a marginal increase in the average number of iterations per point of the existing methods as they have more number of divergent points than that of the proposed method. Moreover, the existing methods require more CPU time to generate the basins as compared to our proposed method. Consequently, our proposed method with memory shows faster convergence in comparison to the existing methods. Figure 3.6: Basins of attraction for P M1, P M2, P M3, M M1, M M2, M M3, respectively for p4(z) 3.6 Conclusions A new method with memory has been introduced. The proposed method has higher order of convergence in comparison to the Chebyshev Halley's family and Traub's method. For verification, we have carried out numerical experiments on a few test functions and some real life problems. It is clearly visible from our results that the proposed method improves the convergence order. This increase in the convergence order has been achieved with no additional functional evaluation. Furthermore, we have also presented the basins of attraction for the proposed as well as some existing methods, which point to the very fact that our proposed method converges largely to the desired zeros over a specified region much faster. Also, we have adapted some tools of the dynamical analysis of the complex as well as multivariate real discrete problems to analyze the stability of the fixed points of iterative methods without memory and with memory, respectively on a quadratic polynomial. This study aids in determining the family members with stable behavior which in turn are suitable for practical problems. Finally, to conclude we would say that the proposed method can be significantly used for solving nonlinear equations. Chapter 4 Optimal Iterative Family Involving First Order Derivative and its Complex Dynamics In the previous chapter, we carried out the dynamical analysis of a non-optimal third- order iterative family of methods. In the current chapter, we have proposed an optimal fourth-order iterative family and performed the dynamical analysis of the family with the help of complex dynamics tools. This study allows us to find those parametric values for which the corresponding family variant's behavior is stable or unstable. Furthermore, we calculate critical and fixed points associated with the rational operator linked to this iterative family. To visualize our findings, we draw dynamical and parameter planes. Hence, we can select the regions where the corresponding method is more efficient or shows chaotic behavior. The conclusions obtained from this stability analysis are used in the numerical section, where some academic and real-life problems are solved. 4.1 Introduction In recent decades, academics have become interested in studying the dynamical behavior of iterative procedures to solve $f(x) = 0$ within the complex plane. The papers Amat et al. (2004); Varona (2002) and the references therein provide substantial proof of this. The concept of iterated rational functions has continuously evolved since the pioneering contributions of Cayley and Schröder in the late 19th century. Their work primarily focused on applying Newton's solver to quadratic polynomials. The extended complex plane hosts rational maps, which emerge from operating root-finding procedures to polynomial equations. Consequently, the principles and concepts related to iterated rational maps find relevance in this context. The contents of this chapter are published in: Journal of Mathematical Chemistry (Springer), 62(1), 198–221, 2024 (SCIE, Q2, Impact Factor: 1.7) Numerous researchers have delved into the dynamics of Newton's procedure when applied to complex polynomials, mainly focusing on those of lower degrees. The paper by Curry et al. (1983) was one of the early works on this subject, and it eventually inspired numerous subsequent studies (see Roberts and Horgan-Kobelski (2004) as an example). Halley's method and other iterative techniques have also been taken into consideration (see Roberts and Horgan-Kobelski (2004)). There are numerous iterative techniques to take into account, each with its unique characteristics and properties. The Schröder iteration functions, encompassing both the first and second kinds, are associated with well-recognized procedures like Halley's and Chebyshev's techniques. Within each of these families, we can find an iterative approach characterized by a convergence order m, corresponding to its mth member. The dynamical behavior of Chebyshev's method reveals surprising results. Notably, Chebyshev's technique demonstrates a distinct dynamical behavior, setting it apart from the previously mentioned Newton's and Halley's solvers, even when applied to polynomials of lower degrees. This particular trait, initially established by Cayley for quadratic polynomials when employing Newton's procedure, asserts that the trajectory initiated from an initial estimate converges towards the nearest root while leaving the perpendicular bisector between the two roots as an invariant set, where divergence occurs. However, Chebyshev's procedure does not adhere to this property, even in the simplest scenario involving quadratic polynomials. Furthermore, one must outlook the stability prospect of the considered method, which aids in telling us the dependence of the method on the initial guesses used. Investigating the dynamical behavior of the rational functions is highly helpful when examining how iterative techniques depend on the initial estimates. We can learn vital details about the associated scheme's stability and dependability from the rational function's dynamical characteristics. By taking into account the fixed, periodic points, etc., of the rational function under study, the dynamical planes exhibit this behavior. Based on various initial hypotheses, an attraction basin enables us to comprehend how an approach operates visually. This study explores the stability of the presented family via complex dynamics tools. To begin with, we derive the rational map for the proposed family through a Möbius conjugate transformation applied to the Riemann sphere. Then, we study its critical and fixed points based on the rational map. The most stable member of the family is then acquired by examining the appropriate parameter and dynamical planes beginning from the critical points. In this study, we firstly develop a new iterative family of optimal nature. We carry out the convergence analysis of the new family in order to demonstrate their order of convergence. A stability analysis of the proposed family is also made. To highlight our theoretical results, numerical computations for the proposed family and comparisons with some of the existing methods are then done. 4.2 An optimal iterative family and its convergence analysis We propose an optimal iterative family of order four whose iterative expression is given as $y_n = x_n - f'(x_n)\ f(x_n)$, $x_{n+1} = x_n - \alpha\ f'(x_n)(f(x_n) - f(y_n)) - (1 - \alpha)\ (\gamma_2 f(y_n) - f(x_n))\ f'(x_n)\ f(x_n)^2\ (\gamma_1 f(y_n) - f(x_n))\ f(x_n)$, (4.2.1) $n = 0, 1, 2, \ldots$, where $\alpha \in \mathbb{R}\backslash\{1\}$ denotes a free parameter, whereas γ_1, γ_2 are the parameters depending on α. Theorem 4.2.1 describes the circumstances in which our proposed family (4.2.1) will reach the optimal convergence rate. 4.2.1 Convergence analysis Theorem 4.2.1. Consider a function $f : D \subset \mathbb{R} \to \mathbb{R}$ that is sufficiently differentiable in a domain D. If an initial estimate x0 is close enough to κ ∈ D which is a simple root of f (x) = 0, then the sequence {xn}n≥0 obtained from scheme (4.2.1) converges to κ with convergence order at least four when γ1 = 1 α − 1 and γ2 = α − 1 , where α ∈ R\{1}, α giving the error relation, en+1 = (2αα−−11 d32 − d2d3 e4n + O(en)5,) where en = xn − κ and dn = n! f ′(κ) , n = 2, 3, . . . 1 f (n) (κ) Proof. Expanding f(xn) and f′(xn) about xn = κ by Taylor series, we have f(xn) = f′(κ)(en + d2e2n + d3e3n + d4e4n) + O(en)5 and f′(xn) = f′(κ)(1 + 2d2en + 3d3e2n + 4d4e3n + 5d5e4n) + O(en)5. Substituting (4.2.3) and (4.2.4) in the first substep of scheme (4.2.1), one arrive at en,y = yn − κ = d2e2n + (−2d22 + 2d3)e3n + (4d32 − 7d2d3 + 3d4)e4n + O(en)5. Also, the Taylor's expansion of f(yn) around κ, similar to (4.2.3) is f(yn) = f′(κ)(en,y + d2e2n,y + d3e3n,y + d4e4n,y) + O(en,y)5. (4.2.2) (4.2.3) (4.2.4) (4.2.5) (4.2.6) On substituting (4.2.2)−(4.2.6) in the second substep of scheme (4.2.1), one can have en+1 = xn+1 − κ = − (α − 1)(γ1 − γ2 + 1)d2e2n + ((−2 + γ1(−4 + γ2) + 4γ2 − γ22+ α(3 − γ1(−4 + γ2) − 4γ2 + γ22))d22 − 2(α − 1)(γ1 − γ2 + 1)d3)e3n+ ((4 − 13γ2 + 7γ22 − γ23 + γ1(13 − 7γ2 + γ22) + α(−7 + 13γ2 − 7γ22+ γ23 − γ1(13 − 7γ2 + γ22)))d32 + (−7 + 14γ2 − 4γ22 + 2γ1(−7 + 2γ2)+ 2α(5 + γ1(7 − 2γ2) − 7γ2 + 2γ22))d2d3 − 3(α − 1)(γ1 − γ2 + 1)d4)e4n+ O(en)5. (4.2.7) Now, to attain fourth-order convergence, coefficients of e2n and e3n must vanish simultaneously. Hence, substituting coefficients of e2n and e3n to zero, we get the following conditions: γ1 = 1 α−1 and γ2 = α α−1 , where α ∈ R\{1}. (4.2.8) Therefore, replacing these conditions (4.2.8) in (4.2.7), we obtain en+1 = (2αα−−11d32 − d2d3 e4n + O(en)5, (4.2.9)) where α ∈ R\{1} denotes a free parameter. This is the error equation depicting the conver- gence order four of our proposed optimal family (4.2.1). This finishes the proof. 4.3 Complex dynamics Let us discuss the stability of our proposed iterative family (4.2.1) using complex dynamics tools. 4.3.1 Scaling theorem Theorem 4.3.1. Let the map f (x) be analytic on the Riemann sphere, and consider an affine map A(x) = ξ1x + ξ2, ξ1 ≠ 0. If V (x) = f ◦ A(x), then the fixed point operator obtained from the scheme (4.2.1) denoted by Sf is analytically conjugated to SV via A, that is, A ◦ SV ◦ A−1(x) = Sf (x). Proof. The fixed point operator obtained from the proposed family (4.2.1) denoted by Sf is given as Sf (x) = x − α f (x)2 f ′(x)(f (x) − f (y)) − (1 − α) f (x)(f (x) − γ1f (y)) f ′(x)(f (x) − γ2f (y)) , (4.3.1) f(x) where y = x − f′(x). With this iteration function, we obtain SV(A−1(x)) = A−1(x) − α V(A−1(x))2 V (A−1(x)) V(A−1(x)) V(A−1(x)) − γ1 V A−1(x) − V(A−1(x)) V′(A−1(x)) V(A−1(x)) − V A−1(x) − V′(A−1(x)))) (((4.3.2) −(1 − α) ((V′(A−1(x)) V(A−1(x)) − γ2 V A−1(x) − V′(A−1(x)))) VV′((AA−−11((xx)))))) . ((Now, since V (x) = f ◦ A(x), we have V ◦ A−1(x) =f (x), (V ◦ A−1)′(x) = V′(A−1 (x)). 1 ξ1 (4.3.3) (4.3.4) By using (4.3.3) and (4.3.4), we get V′(A−1(x)) =ξ1(V ◦ A−1)′(x) = ξ1f′(x), (4.3.5) and V′′(A−1(x)) = ξ12f′′(x). (4.3.6) Therefore, by (4.3.3) and (4.3.5), we obtain the expression SV(A−1(x)) = A−1(x) − α (f(x))2 ξ1f′(x) f(x) − V A−1(x) − f(x) ((f(x) ξ1f′(x))) −(1 − α) f(x) f(x) − γ1 V A−1(x) − ξ1f′(x))) . ((ξ1f′(x) f(x)−γ2 V A−1(x)− f(x) ((ξ1f′(x))) Further, we have A ◦ SV ◦ A−1(x) =A(SV(A−1(x))) = ξ1SV(A−1(x)) + ξ2 =ξ1A−1(x) − α (f(x))2 f′(x) f(x) − V A−1(x) − f(x) ((ξ1f′(x))) f(x) f(x)−γ1 V A−1(x)− f(x) −(1−α) ((ξ1f′(x))) +ξ2 f′(x) f(x) −γ2 V A−1(x)− f(x) ((f(x))2 (ξ1f′(x))) =x−α f′(x) f(x)−V A−1(x)− f(x) ((ξ1f′(x))) f (x) f (x) − γ1 V A−1(x) − f (x) − (1 − α) ((ξ1f′(x))) . f ′(x) f (x) − γ2 V A−1(x) − f (x) ((ξ1f ′(x))) Now, to prove A ◦ SV ◦ A−1(x) = Sf (x), we just need to prove V A−1(x) − ξ1f ′(x) f (x) = f (x) ( ) f (y). Expanding V A−1(x) − (4.3.6), we have (ξ1f′(x)) using Taylor's expansion about A−1(x) and Equation V A−1(x) − f(x) =V(A−1(x)) − V′(A−1(x)) f(x) + V′′(A−1(x)) f (x)2 (ξ1f′(x)) ξ1f′(x) 2ξ12f ′(x)2 − V ′′′(A−1(x)) 6ξf13f(x′(x)3)3 + . . . =f ′′(x) 2ff(′x(x)2)2 − f ′′′(x) 6ff(′(xx))33 + . . . =f x − f ′(x) f (x) =f (y). ( ) Thus, we have A ◦ SV ◦ A−1(x) = Sf (x) which finishes our proof. Remark 4.3.1. The aforementioned theorem proves that we can conjugate the dynamical traits of an operator with that of another via an affine application. According to the scaling theorem, studying the dynamical properties of a conjugated map that has been simplified through conjugacy A, is definitely of value. 4.3.2 Rational operator Any nonlinear function can be used to generate the rational operator. In this context, our focus lies on quadratic polynomials, as their stability or instability characteristics in relation to a given procedure can be extrapolated to other nonlinear functions too. Proposition 4.3.1. Consider a quadratic polynomial P(x) = (x − ξ1)(x − ξ2) having roots ξ1, ξ2 ∈ R. So, the rational operator Mγ1,γ2(x) obtained from the family (4.2.1) applied to P(x), is Mγ1,γ2(x) = σ2 σ1 , (4.3.7) where σ1 = x2(α(γ2 − 1)(x + 1)2 − αγ1(1 + x + x2) + (1 + x + x2)(γ1 − (−1 + γ2 − x)(x + 1))) and σ2 = 1 − (γ2 − 3)x + (4 + γ1 − 2γ2 + α(−1 − γ1 + γ2))x2 + (3 + γ1 − α(2 + γ1 − 2γ2) − 2γ2)x3 − (α − 1)(1 + γ1 − γ2)x4. Making use of the conditions described in Theorem 4.2.1 for γ1 and γ2, we obtain the operator Mα(x) given as Mα(x) = −1 − x − x2 + α(1 + 2x + 2x2) x4(−1 − x − x2 + α(2 + 2x + x2)) , (4.3.8) where α ∈ C is an arbitrary parameter. Proof. Given a polynomial P(x) = (x − ξ1)(x − ξ2) containing roots ξ1, ξ2 ∈ R. By applying (4.2.1) on P(x), we can get a rational map H(x) which depends upon the parameters γ1, γ2 and the roots ξ1, ξ2. Now, the following conjugate transformation can be employed, mc(x) = xx −− ξξ12 , (4.3.9) acquiring the properties, (i) mc(ξ1) = 0, (ii) mc(∞) = 1, (iii) mc(ξ2) = ∞, proving that the function H(x) can be transformed into the conjugated operator, Mγ1,γ2(x) = σ2 σ1 , where σ1 = x2(α(γ2 − 1)(x + 1)2 − αγ1(1 + x + x2) + (1 + x + x2)(γ1 − (−1 + γ2 − x)(x + 1))) and σ2 = 1 − (γ2 − 3)x + (4 + γ1 − 2γ2 + α(−1 − γ1 + γ2))x2 + (3 + γ1 − α(2 + γ1 − 2γ2) − 2γ2)x3 − (α − 1)(1 + γ1 − γ2)x4. Making use of the conditions described in Theorem 4.2.1 for γ1 and γ2, we obtain the operator Mα(x) given as Mα(x) = −1 − x − x2 + α(1 + 2x + 2x2) x4(−1 − x − x2 + α(2 + 2x + x2)) (4.3.10) depending only on the arbitrary parameter α and the proof is finished. It can be observed from Proposition 4.3.1 and in view of the scaling theorem that analyzing the rational operator (4.3.10) is similar to analyzing the family (4.2.1). 4.3.3 Fixed and critical points It can be obtained from (4.3.10), Mα(x) − x = x(x − 1)ψα(x) ϕα (x) , (4.3.11) where ψα(x) = −(1 + x + x2)2 + α(1 + 3x + 5x2 + 3x3 + x4) and ϕα(x) = −1 − x − x2 + α(1 + 2x +

2x2). Proposition 4.3.2. For ψα(x) and ϕα(x), we have ˆ When α = 0, ψα(x) and ϕα(x) have common factor (x2 + x + 1). ˆ When α = , ψα(x) has a factor (x − 1)2. 9 13 ˆ When α = , ϕα(x) has a factor (x − 1). 3 5 Proof. Upon solving the equations ψα(x) = 0 and ϕα(x) = 0 simultaneously, one can get: ψα(x) and ϕα(x) have common factor (x2 + x + 1) when α = 0. At this value, ψ0(x) = (x2 + x + 1)2 and ϕ0(x) = −(x2 + x + 1). Then, we put x = 1 into ψα(x) and ϕα(x) and get ψα(1) = −9 + 13α and ϕα(x) = −3 + 5α. As we solve ψα(1) = 0 and ϕα(1) = 0, we find that ψα(x) has a factor (x − 1)2 when α = 9 and ϕα(x) has a factor (x − 1) when α = . This completes the proof. 3 13 5 Proposition 4.3.3. To find FPs, we solve Mα(x) − x = 0 for x with given values of α. The FPs of Mα(x) are x = 0, x = ∞ and the following strange FPs (SFPs): ˆ F1 = 1 when α ≠ and Fi(α) corresponding to the 4 roots of polynomial ψα(x), 3 where i =(2, 3, 4, 5. 5) As we choose different values of α, we obtain different number of FPs as follows: ˆ Mα(x) has 6 FPs when α ∈ C\ 0, 3 5 . { } ˆ Mα(x) has 5 FPs when α = excluding F1 = 1. 3 5 ˆ Mα(x) has 4 FPs when α = 0. ˆ Mα(x) has 6 FPs when α = and F1 = 1 is a triple root in this case. 9 13 74 ˆ The SFPs of Mα(x) satisfy Fi = for i ≠ j, that is, each pair is conjugate to each 1 Fj other. Proof. From (4.3.11), we have Mα(x) − x = x(x − 1)ψα(x) ϕα (x) = 0. (4.3.12) Now, M0(x) − x = x(x3 − 1), M 9 (x) − x = −x(x − 1)3(4 + 7x + 4x2) and 13 −4 + 5x + 5x2 (4.3.13) M 3 (x) − x = − x(2 + x + x3 + 2x4) 5 2+x . We can obtain the corresponding FPs for varying α upon solving these equations which are displayed in Table 4.1. According to Proposition 4.3.3, we obtained that there exist a minimum of 5 and a max- imum of 7 FPs, where 0 and ∞ are two superattracting FPs of Mα that correspond to the roots a and b of the polynomial g(x), respectively. They do not involve the parameter α. If we exclude these two FPs, then the remaining points are termed as the SFPs. Also, F1 = 1 when α ≠ 3 (5 is indicating a point of divergence within the original approach.) Proposition 4.3.4. The FP F1 = 1, α ∈ C\ 3 has its stability properties as follows: { } 5 ˆ F1 is a repulsor when 201 α − 299 > 6 299 . ˆ F1 is an attractor when| α − 20|1 < 6 299 . 299 ˆ F1 is parabolic when α −| 201 = | 6 . 299 299 ˆ F1 is a superattractor| for α = |2 . 3 Proof. We compute the derivative of the operator Mα from (4.3.11) as Mα′(x) = (xϕ3αλ(αx()x))2 , (4.3.14) where λα(x) = 4(1 + x + x2)2 − 3α(4 + 9x + 12x2 + 9x3 + 4x4) + α2(8 + 22x + 30x2 + 22x3 + 8x4). Substituting x = 1 in (4.3.14), we get |M′α(1)| = 6 2 − 3α 3 − 5α . It is easy to obtain | | 6 2 − 3α 3 − 5α ≤ 1 ⟺ 6|2 − 3α| ≤ |3 − 5α|. Let α = p + iq be an arbitrary complex number. Then, we have the following: | |2 − 3α|2 = (2 − 3p)2 + 9q2 | and |3 − 5α|2 = (3 − 5p)2 + 25q2. So, 36(4 + 9p2 − 12p + 9q2) ≤ 9 + 25p2 − 30p + 25q2. Upon simplifying, we obtain 299p2 + 29 − 402p + 135 = 299 p − 220919 + 299q2 − 23969 ≤ 0 2 ( ) =⇒ p − 201 + q2 ≤ 2 6 2 (4.3.15) . (299) (299) Thus, |M′α(1)| ≤ 1 iff 201 α − 299 ≤ 6 . Further, |M′α(1)| = 0 iff α = ||3; F1 = 1 becomes superattractor in this case. Hence the 2 299 | proof. Further, the stability of the SFPs is based on the parameter α. The superattracting FPs other than 0, ∞ and F1 = 1 for α = 2 3 are as follows: ˆ F2 and F3 for α = 1.32991. ( ) ˆ F4 and F5 for α = 0.70855. As a result, the method might not converge to the root because of a basin of attraction of the SFP. Table 4.1 displays the stability outcomes of the SFPs for varying values of α. As outlined in Proposition 4.3.3, the stability analysis of the SFPs becomes more efficient since each set of conjugate SFPs shares identical stability characteristics, effectively reducing the stability by fifty percent. Further, established classical findings indicate that each invariant Fatou component is associated with a minimum of one CP linked to it. This implies that each attraction basin contains at least one CP, other than the points 0 and ∞, which are categorized as FCPs. Our focus lies in analyzing these points dependent on the parameter α, due to their varying orbital dynamics. Table 4.1: Stability of SFPs Fi for specific α−values α Fi |M′α(Fi)| Behavior No. of Fi 0 −0.5 ± 0.866025i 4 Repulsor 2 9 13 −0.875 ± 0.484123i 5.69 Repulsor 2 3 5 −0.84307 ± 0.537803i 5.84 Repulsor 4 0.59307 ± 0.805151i 4.41 Repulsor Table 4.2: FCPs Ci for special α−values α Ci No. of Ci 0 − 0 9 13 1.37429, 0.727649 4 −0.530922, −1.88352 3 5 −0.404831, −2.47017 2 Proposition 4.3.5. The CPs of Mα(x) can be seen as solutions of equation M′α(x) = 0 or roots of the equation λα(x) = 0. As x = 0 and ∞ are both superattractive FPs of Mα(x), they are also CPs giving their specific Fatou elements. The FCPs are as follows: 1 8 √ √ √ √ √ √ ˆ C1 = − 11α 16 α − 1 α−1 + 3 αu1 − √ 2 u3 − 8 3u2 αu1 + 11 3αu2 αu1 u2 √ (α − 1)3(2α − 1) , √ √ − 11α √ √ √ √ √ ˆ C2 = 1 8 16 α − 1 α−1 − 3 αu1 + 2 u3 − 8 3u2 αu1 + 11 3αu2 αu1 u2 √ (α − 1)3(2α − 1) , √ √ 1 8 11α 3 αu1 √ √ √ √ √ ˆ C3 = 16 α − 1 − α−1 + u2 − 2 u3 + 8 3u2 αu1 − 11 3αu2 αu1 √ (α − 1)3(2α − 1) , 1 8 √ √ − 11α 3 αu1 √ u3 + 8 3u2 αu1 − 11 3αu2 αu1 √ √ √ √ ˆ C4 = 16 α − 1 α − 1 + u2 + 2 √ (α − 1)3(2α − 1) , where u1 = 16 − 19α + 6α2, u2 = (−1 + α)2(−1 + 2α) and u3 = −96 + 432α − 711α2 + 501α3 − 126α4. As we choose different values of α, we obtain different number of CPs as follows: √ ˆ Mα(x) has 7 CPs when α ∈ C/ 0, 5 3 . { } ˆ Mα(x) has 2 CPs when α = 0 which are 0 only. ˆ Mα(x) has 7 CPs when α = and 0 denotes the triple root in this case. 9 13 ˆ Mα(x) has 5 CPs when α = and 0 denotes the triple root in this case too. 3 5 ˆ The FCPs of Mα(x) satisfy Ci = for i ≠ j, that is, each pair is conjugate to each 1 Cj other which are C1 and C2, and C3 and C4. As indicated by Proposition 4.3.5, it has been established that the count of CPs falls within the range of 3 to 8. Notably, within this set, the points 0 and ∞ correspond respectively to the roots ξ1 and ξ2 of the equation P(x) = 0. It's important to reiterate that any CP other than these two is categorized as an FCP. It's worth noting that, similar to the SFPs, the analysis of FCPs is simplified since examination of just half of them is required, given that each pair of conjugate FCPs shares identical stability characteristics. Refer to Table 4.2 for a comprehensive display of the FCPs across varying values of the parameter α. 4.3.4 Dynamical planes and parameter spaces The parametric values of α determine how dynamically the operator Mα behaves. In the complex plane, the term 'parameter plane' (see Chicharro et al. (2013b)) refers to a mesh where every point represents an independent value of α. Taking one of the FCPs Ci as a starting measure, the graphical representation of this plane depicts the convergence analysis of a variant in our proposed family (4.2.1) incurred with α. The final image has a resolution of 500 × 500 pixels. The pixel is colored red whenever a method converges to one of the solutions beginning with Ci having an iteration count of 25 as its upper limit and a tolerance of 0.001, otherwise, black color is assigned to the pixel as can be illustrated through Figure 4.1. Subsets of schemes with comparable dynamical behavior are produced for every α-value that corresponds to the similar connected component of the parameter space. Discovering possible stable regions in the parameter plane (i.e., red colored regions) is interesting since these values will provide us the best members in the family (4.2.1) in terms of numerical stability. As displayed in Table 4.2, the family (4.2.1) has a maximum of four FCPs. Proposition 4.3.5 makes it clear that the study of two distinct parameter planes will be sufficient as shown in Figure 4.1 because the CPs C1 to C4 are conjugated in pairs. The major region of the figure is red, which indicates that it converges to the roots in most of the cases. In terms of numerical stability, the methods associated with the parametric values of α inside the stability areas (i.e., red colored regions), for instance, α = 3 , 11 33 , , −2, 7 , 8 show good dynamical behavior. Additionally, with these particular values of α, the iterative 5 20 50 5 5 scheme of the proposed family (4.2.1) is simplified. This helps in reducing the desired time to arrive at the solution. On the other hand, the methods associated with the parametric values of α giving black shaded areas are outside the stability regions in the parameter spaces, for instance α = 9 , 7 , −30, −35, showing poor dynamical behavior in context of numerical stability. 13 10 Furthermore, the dynamical planes can also be used to explain how FPs behave. We can check the method's stability for a given value of α by drawing a dynamical plane (see Chicharro et al. (2013b)) (represented by a mesh in the complex plane) where each point represents a different value of the initial estimate x0. With an upper limit of 25 iterations and a tolerance of 10−3, its graphical depiction demonstrates the convergence of the method to any of the solutions beginning with x0. Different colors are used to represent the different basins of attraction. These images were created with a resolution of 500 × 500 pixels. Now, through dynamical planes, we examine the stability of some methods of our proposed family. This analysis encompasses instances where the values of α fall within and outside the stability regions of the parameter spaces. Inside the stability region are the methods pertaining to some values of α, namely, α = 20 50 , −2, 3 , 11 33 5 , 7 5 , 8 5 . Their dynamical planes are displayed in Figure 4.2. It is to be noted here that all the methods display two basins of attraction only, related to the roots namely 0 and ∞ by red and blue colors, respectively. In addition, no black areas are found representing the non-convergence to the root. As a result, the said methods exhibit good dynamical behavior, leading to being very stable. However, some black areas denote the slow convergence of the techniques considered here that can be spotted outside the stability regions for α = 9 , 7 , −30, −35 as shown in Figure 13 10 4.3. In these figures, due to the presence of another basins, the region corresponding to root 0 is of small size, which reduces the possibilities of convergence to the root. Thus, the said methods show poor dynamical traits. Remark 4.3.2. The parameter planes allow us to recognize the performance of the various members in scheme (4.2.1), leading us to select stable and unstable methods of the family. Moreover, the dynamical planes help us to describe the behavior of these methods. Figure 4.1: Parameter planes for x = C1, C2 and x = C3, C4 4.4 Numerical results This section lays out the comparison of our family (4.2.1) with several existing schemes. The initial value of α is assumed to be chosen beforehand to begin with the computations. Also, a suitable x0 must be fixed. The following members of the family (4.2.1) are chosen in order to perform the calculations: 1. P M1 for α = 9 . 13 2. P M2 for α = . 3 5 3. P M3 for α = 23 15 . 4. P M4 for α = 50 33 . The following existing methods without memory have been selected to facilitate comparisons with our methods: 1. Panday et al. method (SM1 and SM2) Panday et al. (2023): yn = xn − f ′(xn) f (xn) , xn+1 = xn − γ f ′(xn)(f (xn) − f (yn)) − (1 − γ) f ′(xn)(f (xn) − 3f (yn)) f (xn)2 f (xn)2 − 2f (xn)f (yn) , (4.4.1) n = 0, 1, 2, . . . , γ ∈ R. The parameter values taken are as γ = 1 for SM1 and γ = 20 for SM2. 11 2. Cordero et al. method (AM) Cordero et al. (2013c): yn = xn − f [xn, wn] , wn = xn + f (xn), f (xn) xn+1 = yn − f [xn, yn]f [yn, wn] , n = 0, 1, 2, . . . f (yn)f [xn, wn] 3. Chun method (CM) Chun (2007g): yn = xn − f ′(xn) f (xn) , xn+1 = xn − ff′((xxnn))(1 + u + 2u2), u = ff((xynn)), n = 0, 1, 2, . . . 4. Kou et al. method (KM) Kou et al. (2007a): yn = xn − ff′((xxnn)) , xn+1 = xn − f ′(xn)(f (xn) − f (yn)) , n = 0, 1, 2, . . . (f (xn))2 + (f (yn))2 (4.4.2) (4.4.3) (4.4.4) Further, Table 4.3 displays some nonlinear functions (f1 to f5) used to carry out the compu- tations. Table 4.3: Test functions, associated zeros and the initial approximations (x0) Function Real zero x0 f1(x) = sin2 x + x f2(x) = ex2+7x−30 − 1 f3(x) = 5i=1(x − i) f4(x) = (x − 2)(x10 + x + 1)e−x−1 f5(x) = (∏x − 1)3 − 1 0 3 2 2 2 0.05 2.90 1.8 1.925 1.8 In addition, some real-life problems are also solved after transforming them to nonlinear functions (f6 to f9). COC (computational order of convergence) and the errors of approxi- mations to the desired roots, | xn − κ | for n = 1, 2, 3 of ft(x), t = 1, 2, . . . , 9 are outlined in Tables 4.4−4.12. Remark 4.4.1. It can be seen from Tables 4.4-4.8 that for the function f1, the proposed methods PM1, PM2, PM3 and PM4 converge to the respective solution with a significantly reduced approximation error compared to the previous methods. For the function f2, AM and CM diverge to the solution. However, KM needs more than 3 iterations in order to converge to the solution. AM converges to undesired root, 4 for the function f3 and diverges for the function f4. Due to a derivative flaw, CM and KM cannot operate at points where the function is zero or nearly zero. Table 4.4: Numerical outcomes for f1(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f1 (x) S M1 1.1 × 10−4 S M2 1.1 × 10−5 AM 3.7 × 10−5 C M 2.2 × 10−5 K M 1.4 × 10−5 P M1 6.9 × 10−6 P M2 2.5 × 10−6 P M3 4.7 × 10−6 P M4 5.1 × 10−6 1.2 × 10−12 1.3 × 10−16 1.5 × 10−17 1.2 × 10−18 1.2 × 10−19 2.9 × 10−21 2.1 × 10−23 4.2 × 10−22 6.2 × 10−22 1.9 × 10−36 2.3 × 10−49 4.2 × 10−67 1.1 × 10−71 6.3 × 10−76 8.4 × 10−83 9.2 × 10−92 2.8 × 10−86 1.4 × 10−85 3.0000 3.0000 4.0000 4.0000 4.0000 4.0000

4.0000 4.0000 4.0000 0.0106 0.0021 0.0018 0.0012 0.0013 0.0022 0.0029 0.0031 0.0029 Table 4.5: Numerical outcomes for f2(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f2 (x) 6.0 × 10−2 1.4 × 10−2 – – – 2.3 × 10−2 1.6 × 10−2 2.0 × 10−2 2.0 × 10−2 1.3 × 10−2 1.9 × 10−5 – – – 1.3 × 10−4 1.9 × 10−5 5.8 × 10−5 6.8 × 10−5 9.5 × 10−5 3.0 × 10−14 – – – 1.4 × 10−13 3.9 × 10−17 4.9 × 10−15 9.6 × 10−15 3.7890 3.1164 D D NC 4.0762 4.0484 4.0639 4.0663 0.0027 0.0025 – – – 0.0038 0.0035 0.0033 0.0031 D- Divergent NC- Not converging in desired iterations Real-life problems: Next, we describe a few real-life problems together with the computa- tional outcomes: Example 4.4.1. Chemical reactor problem: As described in Example 2.4.3, the nonlinear equation for the said problem is as follows: x 0.4(1 − x) f6(x) = 1 − x − 5 log 0.4 − 0.5x + 4.45977 = 0. The desired root is κ ≈ 0.7573962462537538. After comparing the new methods with existing ones taking initial value x0 = 0.72, the results are displayed in Table 4.9. Example 4.4.2. Channel flow problem: In this example, we consider a problem of open channel flow, which involves finding the depth of water in a rectangular channel which is Table 4.6: Numerical outcomes for f3(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f3 (x) SM1 1.4 × 10−3 SM2 6.8 × 10−4 AM – CM 8.2 × 10−4 KM 7.4 × 10−4 PM1 5.1 × 10−4 PM2 5.7 × 10−4 PM3 5.4 × 10−4 PM4 5.4 × 10−4 2.0 × 10−9 2.2 × 10−11 – 1.6 × 10−12 7.4 × 10−13 1.9 × 10−15 4.1 × 10−14 1.6 × 10−14 1.3 × 10−14 5.6 × 10−27 7.2 × 10−34 – 2.6 × 10−47 7.1 × 10−49 3.6 × 10−61 1.2 × 10−54 1.3 × 10−56 3.7 × 10−57 2.9992 3.0000 UR 3.9995 3.9996 3.9970 4.0000 4.0003 4.0004 0.0016 0.0014 – 0.0008 0.0007 0.0012 0.0012 0.0011 0.0011 UR- Converging to undesired root Table 4.7: Numerical outcomes for f4(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f4 (x) 1 .1 × 10−2 1.9 × 10−3 – 6.6 × 10−2 2.1 × 10−2 3.1 × 10−3 1.9 × 10−3 2.5 × 10−3 2.6 × 10−3 2.3 × 10−5 1.0 × 10−8 – 1.8 × 10−3 2.3 × 10−5 9.6 × 10−9 7.2 × 10−10 3.3 × 10−9 4.1 × 10−9 2.0 × 10−13 1.8 × 10−24 – 3.2 × 10−9 4.7 × 10−17 9.1 × 10−31 1.5 × 10−35 1.0 × 10−32 2.5 × 10−32 3.0342 3.0052 D 3.4574 3.9064 4.0035 4.0022 4.0029 4.0031 0.0042 0.0041 – 0.0027 0.0030 0.0041 0.0056 0.0040 0.0041 D- Divergent represented by the nonlinear equation given as follows (see Rehman et al. (2021)): √ tcx cx 2 3 f (x) = n (c + 2x) − F = 0, √ F being the water flow, which is given by F = tcx 2 n r3, n is the Manning’s roughness coeffi- cient, t is the slope, r is the hydraulic radius and c is the width of the channel. The following equation is obtained if the parameter values are as F = 14.15 m3/s, c = 4.572 m, t = 0.017 and n = 0.0015: 0.5961x 4.572x 2 3 − 14.15 = 0. The desired solution being κ = 0.13839748098511792; for the initial guess x0 = 1.2, the f7(x) = 0.0015 (4.572 + 2x) numerical outcomes are displayed in Table 4.10. Table 4.8: Numerical outcomes for f5(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f5 (x) S M1 1.2 × 10−2 S M2 1.7 × 10−3 AM 1.4 × 10−1 C M 2.8 × 10−2 K M 1.1 × 10−2 P M1 2.6 × 10−3 P M2 1.5 × 10−3 P M3 2.1 × 10−3 P M4 2.2 × 10−3 1.7 × 10−6 5.3 × 10−10 3.8 × 10−3 2.3 × 10−6 4.1 × 10−8 7.3 × 10−11 4.3 × 10−12 2.3 × 10−11 2.9 × 10−11 5.2 × 10−18 1.5 × 10−29 5.3 × 10−9 1.4 × 10−22 7.9 × 10−30 4.5 × 10−41 2.8 × 10−46 3.3 × 10−43 8.8 × 10−43 3.0068 3.0009 3.6288 3.9726 3.9925 4.0006 4.0004 4.0005 4.0005 0.0004 0.0004 0.0007 0.0004 0.0003 0.0004 0.0004 0.0007 0.0007 Table 4.9: Numerical outcomes for Example 4.4.1. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f6 (x) 1.3 × 10−2 4.6 × 10−3 – – 3.6 × 10−2 6.0 × 10−3 4.8 × 10−3 5.4 × 10−3 5.6 × 10−3 5.4 × 10−4 3.8 × 10−6 – – 1.3 × 10−2 9.0 × 10−6 2.6 × 10−6 5.3 × 10−6 5.9 × 10−6 4.0 × 10−8 1.3 × 10−15 – – 2.2 × 10−4 2.9 × 10−17 8.8 × 10−17 8.2 × 10−17 7.9 × 10−17 3.1678 3.0950 D D 2.4506 4.0109 4.0130 4.0126 4.0123 0.0046 0.0048 – – 0.0038 0.0045 0.0048 0.0038 0.0036 D- Divergent Example 4.4.3. Multi-factor effect: As described in Example 2.4.4, the nonlinear equation for the said problem is as follows: f8(x) = x − cos x + = 0. 1 π 2 4 (4.4.5) The desired root of f8(x) is κ ≈ −0.3090932715417949 and the numerical results are ob- tained by taking x0 = −2.25 in Table 4.11. Example 4.4.4. Embedment of a wall: As described in Example 2.4.5, the nonlinear equation for the said problem is as follows: f9(x) = x3 + 2.87x2 − 10.28 4.62 − x = 0. (4.4.6) The required zero of f9(x) is κ ≈ 2.0021 and the initial guess taken for the results is x0 = 1.2. The results are displayed in Table 4.12. Table 4 .10: Numerical outcomes for Example 4.4.2. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f7 (x) S M1 1.1 × 10−1 S M2 6.6 × 10−2 AM 1.6 × 10−1 C M 9.0 × 10−2 K M 8.2 × 10−2 P M1 3.5 × 10−2 P M2 5.3 × 10−2 P M3 4.6 × 10−2 P M4 4.4 × 10−2 2.4 × 10−3 1.3 × 10−4 2.1 × 10−2 8.3 × 10−4 4.8 × 10−4 7.8 × 10−6 4.6 × 10−6 8.5 × 10−6 9.2 × 10−6 6.2 × 10−8 9.7 × 10−13 5.6 × 10−4 2.7 × 10−11 2.0 × 10−12 2.7 × 10−18 2.7 × 10−18 2.7 × 10−18 2.7 × 10−18 2.6490 2.9298 1.5936 3.5588 3.6497 3.9222 4.0035 3.8879 3.8940 0.0047 0.0045 0.0018 0.0031 0.0116 0.0044 0.0043 0.0043 0.0046 Table 4.11: Numerical outcomes for Example 4.4.3. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f8 (x) 2.5 × 10−2 2.8 × 10−2 4.0 × 10−1 2.8 × 10−2 2.8 × 10−2 2.9 × 10−2 2.9 × 10−2 2.9 × 10−2 2.9 × 10−2 1.2 × 10−6 1.7 × 10−7 2.2 × 10−3 5.8 × 10−8 3.5 × 10−8 2.4 × 10−8 1.3 × 10−8 1.9 × 10−8 2.0 × 10−8 1.2 × 10−19 3.8 × 10−23 2.8 × 10−12 1.5 × 10−30 3.9 × 10−31 2.9 × 10−31 3.0 × 10−31 3.0 × 10−31 3.0 × 10−31 2.9966 2.9946 3.8518 3.9948 3.9958 3.9970 3.9961 3.9966 3.9967 0.0033 0.0031 0.0032 0.0018 0.0020 0.0030 0.0031 0.0030 0.0034 Remark 4.4.2. From Tables 4.9-4.12, we can determine that AM and C M diverge to the root in case of Example 4.4.1. For Example 4.4.4, AM converges to the undesired root, −1.5417 . . . and C M do not converge to the respective solution in 3 number of iterations. In addition, in case of Examples 4.4.2 and 4.4.3, our methods converge to the respective solution acquiring minimum error compared to the previous methods. Remark 4.4.3. The proposed methods from the family (4.2.1) and some existing methods have been compared and the results clarify that our methods perform well in several situations where the existing methods fall short when the COC and errors are concerned as demonstrated in Tables 4.4-4.12. In addition, our methods exhibit a noticeable reduction in the error in approximations as shown in above mentioned tables. Table 4.12: Numerical outcomes for Example 4.4.4. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc

CPU Time f9 (x) SM1 2.7 × 10−1 5.7 × 10−3 SM2 2.0 × 10−2 1.9 × 10−5 AM – – CM – – KM 1.5 × 100 1.4 × 10−1 PM1 6.4 × 10−2 1.6 × 10−5 PM2 2.6 × 10−2 1.9 × 10−5 PM3 4.7 × 10−2 1.8 × 10−5 PM4 5.0 × 10−2 1.8 × 10−5 1.9 × 10−5 1.9 × 10−5 – – 1.1 × 10−4 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 1.9 × 10−5 3.1800 3.0060 UR NC 2.5644 4.0137 4.0048 4.0095 4.0103 0.0007 0.0008 – – 0.0005 0.0009 0.0009 0.0007 0.0007 UR- Converging to undesired root NC- Not converging in desired iterations 4.5 Conclusions In this work, we proposed an optimal iterative family of methods. We explored the complex dynamic behavior of our proposed family (4.2.1) for solving nonlinear equations. In accordance with the theory of Möbius conjugate transformation and scaling theorem, we were able to extract the equivalent rational operator by the application of a general quadratic polynomial upon the iterative family. We computed its fixed and critical points and analyzed that the two strange points were conjugate, same with the case of free critical points. Consequently, we study about half of them for stability. Further, the parameter planes and the dynamical planes were drawn. This study helps identify family members which exhibit stable behavior and are thus appropriate for solving practical issues. Numerical results are displayed on comparison of our methods with some existing schemes. which reveal that our methods perform better compared to the existing methods, since they provide lower absolute errors. Furthermore, our methods have proven their versatility and applicability in diverse problem domains, including the chemical reactor problem, a channel flow problem, multi-factor effect and embedment of a sheet-pile wall. Thus, our methods are highly valuable tools for researchers and practitioners in fields where an accurate root determination is crucial. Figure 4.2: Dynamical planes for α = 5 20 50 , −2, 5 5 , respectively 3 11 33 , , 7 8 , Figure 4.3: Dynamical planes for α = 9 , 7 , −30, −35, respectively 13 10 Chapter 5 Complex Dynamics of a Mean-Based Optimal Iterative Family and its Applications to Chemical Mo dels This chapter performs stability analysis of an optimal mean-based family of itera- tive methods of order four. Taking into consideration the stability aspect of the specified method, one can describe the method's sensitivity to the initial guesses. A rational function corresponding to the iterative family is developed. The convergence and stability of a certain method can be analyzed upon finding the fixed points, critical points, periodic points, etc. of the rational function. Furthermore, the dynamical and parametric planes are drawn which help us to detect the stable as well as non-stable regions. It has been observed that stable iterative methods generally yield better performance on complex problems compared to un- stable methods. This observation has been supported by numerical experiments that compare our proposed family with some existing methods for representing some chemistry problems, like conversion in a chemical reactor, equations of state, and continuous stirred tank reactor problem. 5.1 Introduction It is very well-known that the fixed point iterative methods (IMs) contribute significantly in the field of applied mathematics and scientific computations. Nonlinear models are com- mon in many areas of chemistry, such as reaction kinetics, thermodynamics, quantum chem- istry, and molecular dynamics (see Wilczek-Vera and Vera (2015),Constantinides and Mostoufi (1999),Douglas (1972)). These models often describe complex behaviors, including reaction mechanisms, phase transitions, and molecular interactions, where minimal changes in the ini- tial state may produce significantly distinct outcomes. Through the integration of numerical The contents of this chapter are published in: Journal of Mathematical Chemistry (Springer), 1-23, 2024 (SCIE, Q2, Impact Factor: 1.7) techniques, computational tools, and theoretical methods, researchers can effectively solve nonlinear models, leading to deeper insights into complex chemical phenomena. Whether through iterative methods, optimization techniques, or computational simulations, the ability to solve nonlinear models is crucial for advancing both theoretical and applied chemistry. Ostrowski proposed the coefficient p1/n, known as the efficiency index, to quantify the effectiveness of these techniques. Here, p is the convergence order and n refers to the number of times the function computations are performed during each iterative step. However, as the number of evaluations of functions and the order of a method among a set of IMs do not differ, it becomes relevant to analyze the methods by taking into consideration their dynamical behavior in the argand plane through which the convergence and consistency of the IMs could be examined. The perspective of stability of a method facilitates explaining the method's sensitivity to the initial guesses. The investigation of the rational functions and calculating their fixed points, critical points, periodic points becomes relevant in this context. In the study, a rational operator is constructed and by the use of affine transformation, the concerned iterative family (IF) can be conjugated to that operator so that it can be investigated how the family behaves dynamically. Further, we focus upon its fixed and critical points. Then, the parametric spaces are drawn which help us to detect the stable as well as non-stable regions via the parameter values. By taking into account these parameter values, we are aware of the regions where the concerned method is efficient by drawing the dynamical planes. In this manner, we obtain the most-stable member in the family. 5.2 An optimal iterative family and its convergence anal- ysis In this section, we propose an IF in two free parameters γ1 and γ2 as follows: yn = xn − wn, xn+1 = xn − (f (xn))2 + γ1f (xn)f (yn) + γ2(f (yn))2 ((f (xn))2 + (γ1 − 1)f (xn)f (yn) + (γ2 − γ1 − 1)(f (yn))2) wn , (5.2.1) f (xn) for n = 0,1,2,... Here, wn = f′(xn). This family is optimal as it uses three functional evaluations and shows fourth-order convergence. Theorem 5.2.1 establishes the convergence analysis of the family (5.2.1). 5.2.1 Convergence analysis Theorem 5.2.1. Assume f : D ⊂ R → R to be a sufficiently differentiable function in D. If x0 is close enough to its simple root κ ∈ D, then the sequence {xn}n≥0 derived from (5.2.1) converges to κ with order at least four whose error relation is as follows: en+1 = ((2 + γ1 + γ2)d32 − d2d3)e4n + O(en)5. (5.2.2) Here, γ1 and γ2 are parameters, en = xn − κ and dn = n! f ′(κ) , n = 2, 3, . . . 1 f (n)(κ) Proof. Employing Taylor series expansion to f (xn) and f ′(xn) about xn = κ, we attain f (xn) = f ′(κ)(en + d2e2n + d3e3n + d4e4n) + O(en)5 (5.2.3) and f′(xn) = f′(κ)(1

+ 2d2en + 3d3e2n + 4d4e3n + 5d5e4n) + O(en)5. (5.2.4) Now, using (5.2.3) and (5.2.4), the first substep of (5.2.1) becomes en,y = yn − κ = d2e2n + (2d3 − 2d22)e3n + (3d4 − 7d2d3 + 4d32)e4n + O(en)5. (5.2.5) Further, expanding f(yn) around κ, similar to (5.2.3), we get f(yn) = f′(κ)(en,y + d2e2n,y + d3e3n,y + d4e4n,y) + O(en,y)5. (5.2.6) (f (xn))2 + γ1f (xn)f (yn) + γ2(f (yn))2 ((f (xn))2 + (γ1 − 1)f (xn)f (yn) + (γ2 − γ1 − 1)(f (yn))2) wn (5.2.7) = en − ((2 + γ1 + γ2)d32 − d2d3)e4n + O(en)5. Finally, using the expression (5.2.7), the second substep of (5.2.1) yields en+1 = xn+1 − κ = ((2 + γ1 + γ2)d32 − d2d3)e4n + O(en)5. (5.2.8) (5.2.8) is the error relation for our proposed optimal family (5.2.1), showing convergence order four which brings our proof to an end. 5.3 Complex dynamics This section lays out the stability analysis of the IF (5.2.1). To this end, using a low-degree nonlinear polynomial, a rational operator associated with (5.2.1) is designed. In this manner, the convergence and stability of the corresponding fixed and critical points are examined. Then, the parameter planes are drawn and the dynamical planes for specific techniques of the IF are generated for the optimal choice of the associated parameters. 5.3.1 Scaling theorem Theorem 5.3.1. Assume f(x) to be a holomorphic function on Ĉ, and A(x) = ξ1x+ξ2,ξ1 ≠ 0, to be an affine mapping. If V(x) = f◦A(x), then the fixed point operator denoted by Sf, derived from the family (5.2.1) is analytically conjugated to SV via A, that is, A◦SV◦A−1(x) = Sf(x). Proof. Sf (the fixed point operator) derived from (5.2.1) is as follows: Sf (x) = x − (f (x))2 + γ1f (x)f (y) + γ2(f (y))2 ((f (x))2 + (γ1 − 1)f (x)f (y) + (γ2 − γ1 − 1)(f (y))2) w(x). Here, y = x − w(x) and w(x) = f ′(x) f (x) . Using the recurrence relation (5.3.1), we get SV (A−1(x)) = A−1(x) − V (A−1(x)) a1(x) V ′(A−1(x)) (a2(x)) , where a1(x) = (V(A−1(x)))2 + γ1V(A−1(x))V(θ1(x)) + γ2(V(θ1(x)))2, a2(x) = (V(A−1(x)))2 + (γ1 − 1)V(A−1(x))V(θ1(x)) + (γ2 − γ1 − 1)(V(θ1(x)))2, θ1(x) = A−1(x) − V(A−1(x)) V′(A−1(x)) . Now, since V(x) = f ◦ A(x), we have (V ◦ A−1)(x) = f(x), (V ◦ A−1)′(x) = V′(A−1(x)). 1 ξ1 By using (5.3.3) and (5.3.4), we get V′(A−1(x)) = ξ1(V ◦ A−1)′(x) = ξ1f′(x), and V′′(A−1(x)) = ξ12f′′(x). Thus, (5.3.3) and (5.3.5) lead to the following expression: SV(A−1(x)) = A−1(x) − w(x) a3(x) 1 ξ , 1 (a4(x)) where a3(x) = (f(x))2 + γ1f(x)V(θ2(x)) + γ2(V(θ2(x)))2, a4(x) = (f(x))2 + (γ1 − 1)f(x)V(θ2(x)) + (γ2 − γ1 − 1)(V(θ2(x)))2, θ2(x) = A−1(x) − w(x). 1 ξ1 Further, we have A ◦ SV ◦ A−1(x) = A(SV(A−1(x))) = ξ1SV(A−1(x)) + ξ2 = ξ1A−1(x) − w(x)(aa34((xx))) + ξ2 = x − w(x) a3(x) (a4(x)) . (5.3.1) (5.3.2) (5.3.3) (5.3.4) (5.3.5) (5.3.6) Now, the last point is to verify A ◦ SV ◦ A−1(x) = Sf (x). For this, we just need to prove V A−1(x) − w(x) = f (y). Expanding V A−1(x) − w(x) by Taylor's form about 1 1 A(−1(x) and Equation (5.3.6), we have ξ 1) (ξ 1) V A−1(x) − w(x) = V (A−1(x)) − V ′(A−1(x)) w(x) + 1 V ′′(A−1(x)) 1 1 1 2 ξ w(x) (1)) 2 3 (ξ1 (ξ1) − 1 V ′′′(A−1(x)) 1 6 w(x) + . . . (ξ1) = f ′′(x) 1 w(x)2 − f ′′′(x) 1 w(x)3 + . . . 2 6 = f (x − w(x)) = f (y). Therefore, we get A ◦ SV ◦ A−1(x) = Sf (x) which brings this proof to an end. Remark 5.3.1. It is proven through Theorem 5.3.1 that by an affine application, the dynami- cal behaviors of two operators can be matched via conjugation. The scaling theorem states that it is valuable to examine the dynamics of a conjugated map if it is made simpler by conjugacy A. 5.3.2 Rational operator When developing a rational operator, any nonlinear mapping can be utilized. Nevertheless, we make use of quadratic polynomials since the stability or instability criterion associated with a technique on these polynomials can similarly be implemented to distinct nonlinear mappings. We therefore construct the operator on a quadratic polynomial that corresponds to the IF (5.2.1). Proposition 5.3.1. Let us take a generic quadratic polynomial P(x) = (x − ξ1)(x − ξ2) with roots ξ1, ξ2 ∊ R. The rational operator Mγ1(x) related to (5.2.1) applied to P(x) is Mγ1(x) = 1 + (γ1 + 3)x + (2γ1 + 3)x2 x4(3 + 3x + x2 + γ1(x + 2)) , for all γ1 ∊ C. Proof. Assume P(x) = (x − ξ1)(x − ξ2) to be any quadratic polynomial with roots ξ1, ξ2 ∊ R. The expression (5.2.1) is applied on P(x) resulting in a function H(x) which is rational, dependent on the roots ξ1, ξ2 and the parameters γ1, γ2 ∊ C. The following Möbius conjugacy map can be used: mc(x) = x − ξ1 x − ξ2 , (mc(ξ1) = 0, mc(ξ2) = ∞, mc(∞) = 1), which conjugates H(x) to the operator Mγ1,γ2(x) which will depend on the parameters γ1, γ2 only, whose expression is given as Mγ1,γ2(x) = (γ1 + γ2 + 2)x2 + (γ1 + 3)x + 1 x4(x2 + 3x + γ1x + γ1 + γ2 + 2) . Now, we need to reduce the dependence of our operator on the parameters. So, we will opt for a particular case for γ2 given by γ2 = γ1 + 1 which reduces the operator M as Mγ1(x) = (2γ1 + 3)x2 + (γ1 + 3)x + 1 x4(x2 + γ1(x + 2) + 3x + 3) . (5.3.7) This completes our proof. Proposition 5.3.1 asserts that the rational operator (5.3.7) and the IF (5.2.1) can be ana- lyzed interchangeably. 5.3 .3 Fixed points and their stability Proceeding further, we will find fixed points of Mγ1(x) by solving Mγ1(x) − x = 0 for x with given values of γ1. Now, from (5.3.7), the following can be deduced: Mγ1 (x) − x = ψγ1 (x)(x − 1)x ϕγ1 (x) . (5.3.8) Here, ψγ1 (x) = x4 + (γ1 + 4)x3 + (3γ1 + 7)x2 + (γ1 + 4)x + 1, ϕγ1 (x) = (2γ1 + 3)x2 + (γ1 + 3)x + 1. Firstly, it will be investigated whether values of γ1 exist for common divisors of ψγ1(x) and ϕγ1(x). Also, it will be examined whether they have a factor (x − 1). This examination is carried out in Proposition 5.3.2. Proposition 5.3.2. When checking for common divisors of ψγ1(x) and ϕγ1(x), the subsequent outcomes are: (a) When γ1 = −1, ψγ1(x) and ϕγ1(x) have common factor (x + 1)2. (b) When γ1 = − 157 , ψγ1 (x) has (x − 1)2 as a factor and when γ1 = − 73 , ϕγ1 (x) has (x − 1) as a factor. Proof. (a) On solving ψγ1 (x) = 0 and ϕγ1 (x) = 0 in parallel, we obtain: As γ1 appears in both the polynomials, eliminating γ1 between ψγ1 = 0 and ϕγ1 = 0, we get an equation (x + 1)3 = 0. So, (x + 1)j, j = 1, 2, 3 are the candidates for common divisors of ψγ1 (x) and ϕγ1 (x). If we put x = −1, the equations ψγ1 (−1) = 0 and ϕγ1 (−1) = 0 imply γ1 = −1. Then, in this case, ψγ1 (x) and ϕγ1 (x), respectively reduce to (x + 1)2(x2 + x + 1) and (x + 1)2. (b) Then, we put x = 1 in ψγ1 (x) and ϕγ1 (x) to attain ψγ1 (1) = 5γ1 + 17 and ϕγ1 (1) = 3γ1 + 7, as a result of which ψγ1 (x) has (x − 1)2 as a factor when γ1 = − 157 (as ψγ′1 (1) = 0 in this case) and ϕγ1 (x) has (x − 1) as a factor when γ1 = − 73 . This finishes the proof. The subsequent proposition details the fixed points of Mγ1. Proposition 5.3.3. Fixed points of Mγ1(x) are 0,∞ and the strange fixed points (apart from 0 and ∞) described as: ˆ

F1 = 1 when γ1 ≠ − and the four zeros of ψγ1(x), denoted by Fi(γ1), i = 2, 3, 4, 5. 7 (3) Distinct values for γ1 give us fixed points in varying counts, as detailed below: ˆ Mγ1 (x) has 7 fixed points when γ1 ∈ C\ − 1, − 7 . { 3 } ˆ Mγ1 (x) has 5 fixed points when γ1 = −1 including F1 = 1. ˆ Mγ1 (x) has 7 fixed points when γ1 = − 17 5 and F1 = 1 occurs three times in this case. ˆ Mγ1 (x) has 6 fixed points when γ1 = − not including F1 = 1. 7 3 ˆ Every pair of strange fixed points of Mγ1(x) is mutually conjugate which means that they comply with the condition Fi = , i ≠ j. 1 Fj Proposition 5.3.3 affirms that there are at most 7 and at least 5 fixed points, where 0 and ∞ are superattractors of Mγ1 corresponding to ξ1 and ξ2, respectively. Also, they do not involve the parameter γ1. F1 = 1 is a fixed point which indicates the divergence of the method. Its stability characteristics are detailed in the following result: Proposition 5.3.4. F1 = 1 when γ1 ∈ C\ − 7 has its stability properties as follows: ({ 3 }) ˆ F1 is a repulsor when 171 > 16 γ1 + 55 55 . ˆ F1 is an attractor when| γ1 + 17|1 < 16 55 55 . ˆ F1 is parabolic when γ1 |+ 15751 =|16 . 55 ˆ F1 is a superattractor| for γ1 = |−3. Proof. The derivative M′γ1 from (5.3.7) is computed to gain insight into the stability of the strange fixed points, given by Mγ′1(x) = 2x3((xϕ+γ1(1x))2)ζ2γ1(x) , (5.3.9) where ζγ1(x) = 6x2 + 4γ1(x + 1)2 + 9x + 3γ12x + 6. Substituting x = 1 in (5.3.9), we get |M′γ1 (1)| = 8 γ1 + 3 3γ1 + 7 . It is easy to obtain that | | 8 γ1 + 3 3γ1 + 7 ≤ 1 ⇐⇒ 8|γ1 + 3| ≤ |3γ1 + 7|. (5.3.10) Let γ1 = p + iq be an arbitr|ary com|plex number. Therefore, upon simplifying (5.3.10), we get the following relation: (p + 15751 + q2 ≤ 2 16 2) 55 171 (16) =⇒ γ1 + 55 ≤ 55 . Thus, |M′γ1 (1)| ≤ 1 iff γ1 + 55 171 ≤ 16 55 . | | Further, |M′γ1(1)| = 0 iff γ1 = −3; the case for which F1 is a superattractor. Thus, the proof is complete. | | Furthermore, excluding the above-mentioned points, stability of the remaining ones is dependent on γ1. These stability results with distinct γ1−values are depicted in Table 5.1. Proposition 5.3.3 also asserts that there exists conjugation in pairs of strange fixed points. Each such pair acquires the similar stability properties. That’s why, only half of them are sufficient to analyze. Superattractors other than 0 and ∞ is F1 = 1 for γ1 = −3 (as stated in Proposition 5.3.4). This implies that the fixed point would have an attractor basin, which would prevent the method from converging to the root. The remaining points are never superattractors. Moreover, the stability zones of Fi, i = 1, 2, . . . , 5 are shown in Figures 5.1-5.2. These figures have been generated via Mathematica 11.1. 5.3.4 Critical points The zeros of M′γ1(x) = 0 will be the critical points. We know that to every Fatou component (invariant), at least one critical point is linked, that is, each basin involves at least a free critical point (apart from 0 and ∞). We are interested in examining these free critical points that vary with γ1, in light of their orbital behavior. Table 5.1: Strange fixed points Fi for special γ1−values and their stability γ1 Fi Behavior Number of Fi −1 −(−1)1/3 (−1)2/3 Repulsor Repulsor 2 − 17 5 −2.1307 −0.4693 Repulsor 2 Repulsor − 7 3 −1.9662 −0.5086 0.4041 ± 0.9147i Repulsor 4 Repulsor Repulsor Figure 5.1: Stability surface for F1 = 1 Prior to finding these points, values of γ1 for common divisors of ζγ1(x) and ϕγ1(x) will be investigated for existence. Also, it will be examined whether they have a factor xk,k ∈ N. Proposition 5.3.5 demonstrates this analysis. Proposition 5.3.5. When checking for common factors of ζγ1(x) and ϕγ1(x), the subsequent outcomes are: (a) When γ1 = −1, ζγ1 (x) and ϕγ1 (x) have common factor (x + 1)2, when γ1 = − 73 , they have (x − 1) in common and when γ1 = 3, they have (3x + 1) in common. (b) When γ1 = − 32 , ζγ1 (x) has x as a factor, but ϕγ1 (x) doesn’t have any such factor. Figure 5.2: Stability surfaces for F2, F3 and F4, F5, respectively Proof. (a) On solving ζγ1(x) = 0 and ϕγ1(x) = 0, simultaneously, we obtain: Eliminating γ1 from the two equations, we get a relation (x − 1)(x + 1)4(3x + 1) = 0. So, (x − 1), (x + 1)j, j = 1,2,3,4 and (3x + 1) are the candidates for common divisors of ζγ1(x) and ϕγ1(x). If we put x = 1, ζγ1(1) = 0 implies γ1 = −3,−73, and ϕγ1(1) = 0 implies γ1 = −73. If we put x = −1, the equations ζγ1(−1) = 0 and ϕγ1(−1) = 0 imply γ1 = 1,−1, and γ1 = −1, respectively. Finally, putting x = −13, the equations ζγ1(−13) = 0 and ϕγ1(−13) = 0 imply γ1 = 3,−191, and γ1 = 3, respectively. (b) Then, we put x = 0 into ζγ1(x) and ϕγ1(x) to obtain ζγ1(0) = 2(2γ1 + 3) and ϕγ1(0) = 1, as a result of which ζγ1(x) has x as a factor when γ1 = −32. This brings our proof to an end. The next result details the critical points of the operator Mγ1. Proposition 5.3.6. Critical points of Mγ1(x) are listed as x = 0,∞,−1 and two roots of ζγ1 (x) = 0 denoted by Ci (i = 1, 2). As we have the term x3 in M′γ1 (x), so 0 occurs three times except when γ1 = − 3 Selecting distinct values for γ1 gives us critical points in varying counts, as detailed below: 2 as stated in Proposition 5.3.5. ˆ Mγ1 (x) has 8 critical points when γ1 ∈ C\ − 1, − , 3, − 7 3 . { 3 2 } ˆ Mγ1 (x) has 4 critical points when γ1 = −1, where 0 occurs three times. ˆ Mγ1(x) has 6 critical points when γ1 = − , and 7 critical points when γ1 = 3. In these 7 cases, 0 is a triple root as well. 3 ˆ Mγ1 (x) has 7 critical points when γ1 = − and 0 occurs four times in this case. 3 2 ˆ Each pair of free critical points is mutually conjugate which means that they comply with the condition Ci = , i ≠ j. 1 Cj Proposition 5.3.6 affirms that there are at most 8 and at least 4 critical points, where 0 and ∞ are corresponding to ξ1, ξ2, respectively. Additionally, as with the case of strange fixed points, the dynamical traits of only half of the free critical points are sufficient to analyze. These points are depicted in Table 5.2 for varying values of γ1. Table 5.2: Free critical points Ci for distinct γ1−values γ1 Ci Number of Ci −1 − − − 7 3 −1, −1 2 3 −1, −1, −3 3 − 3 2 −1, −1 2 5.3.5 Dynamical planes and parameter spaces The values of parameter γ1 determine how the operator Mγ1 behaves dynamically. A mesh in the argand plane with distinct values of γ1 assigned to each point is called a parameter plane. The analysis of convergence of a technique of the IF (5.2.1) linked to this γ1 is depicted graphically, with one among the free critical points Ci serving as the primary guess. The package of MATLAB R2022a with a resolution of 500 × 500 pixels has been used to generate the figures. In this instance, red color is applied to the pixel if a method, starting from Ci, reaches to a zero in at most 25 iterations with a tolerance value of 10−3; in rest of the cases, it is painted black as demonstrated in Figure 5.3. Subsets of schemas with dynamically analogous behavior are deduced for any value of γ1 that is a member of the similar connected component of the parameter plane. That’s why, we are interested to find the possible stable areas, i.e. red areas of the parameter plane, as the values of γ1 in these areas will give us the best technique of the IF as far as stability is concerned. 8 6 4 2 } 1 Im{ 0 -2 -4 -6 -8 -8 -6 -4 -2 0 2 4 6 8 Re{ 1 } Figure 5.3: Parameter plane for x = C1 and C2 The affirmations in Table 5.2 and Proposition 5.3.6 directs us to visualize one parameter plane as shown in Figure 5.3 because C1 and C2 are conjugated in pairs. It can be seen that the large domain of the figure is red, which implies that it converges to zeros for majority of the points. However, observing the parameter plane’s details carefully near the imaginary part of γ1, we can see the black patches. The members of the IF associated with γ1 inside the stability areas of the parameter plane will show advantageous dynamical behavior, for γ1 = 1, 3, −1, − 17 , − , − 3 9 as instance. Additionally, taking specific values of γ1 simplify the structure of the IF (5.2.1) which cuts 10 2 10 down the time required in processing to achieve the solution to some extent. In contrast, the members associated with γ1 beyond the areas of stability will display unsatisfactory dynamical behavior, for γ1 = − , − , − , −3 as instance. 7 17 7 Moreover, the dynamical planes provide an illustration of the behavior of fixed points. 3 5 2 After drawing a dynamical plane, the stability of a member for any γ1 can be evaluated. In the complex plane, this is described as a mesh in which every point refers to a distinct initial guess x0. The method’s graphical depiction illustrates how, with 10−3 as tolerance and at most 25 iterations, it can converge from x0 to one of the roots. The attractor basins are depicted in various colors. The software specifications are same as the previous one. Now, through dynamical planes, certain members of the IF (5.2.1) are examined for sta- bility. The members are taken for values of γ1 within and beyond the stability areas of the parametric space. The associated dynamical planes are shown in Figures 5.4 and 5.5. Note that the red and blue basins correspond to 0 and ∞, respectively. For methods in Figure 5.4, there are no regions (black) of divergence. Hence, the mentioned methods are considered stable. For methods in Figure 5.5, the black areas can be seen depicting slow convergence. It makes the basin of 0 small, which lessens the probability of converging to the solution. Hence, these methods display poor behavior. 5.4 Numerical results This section lays out the comparison of our family (5.2.1) with several existing schemes. The initial values of γ1 and γ2 and a suitable x0 are assumed to be chosen beforehand to begin with the computations. The following members of the IF (5.2.1) are chosen in order to perform the calculations: 1. P M1 for γ1 = − and γ2 = 9 1 . 10 10 2. P M2 for γ1 = − and γ2 = − . 3 1 2 2 3. P M3 for γ1 = −1 and γ2 = 0. 4. P M4 for γ1 = − 10 and γ2 = − 10 17 7 . The following existing methods without memory have been selected to facilitate comparisons with our methods: 1. Kung and Traub method (KTM) Kung and Traub (1974): yn = xn − f [xn, un] , un = xn + γ3f (xn), γ3 ∈ R\{0}, f (xn) xn+1 = yn − f (yn)f (un) f [xn, yn](f (un) − f (yn)) , n = 0, 1, 2, . . . The results are obtained for γ3 = −0.01. 2. Soleimani et al. method (SSM) Soleimani et al. (2013): yn = xn − f [xn , un] , un = xn + γ3f (xn), γ3 ∈ R\{0}, f (xn) xn+1 = xn − (f (xn))2 f [xn, un] f (xn) − f (yn) − (ff((yunn)))2 , n = 0, 1, 2, . . . The results are obtained for γ3 = −0.01. ( ) (5.4.1) (5.4.2) 3. Kou and Li method (KLM1) Kou and Li (2007a): yn = xn − wn, xn+1 = xn − 1 + Kf (xn) + 1 1 (Kf (xn))2 wn, γ4 ∈ R, (5.4.3) (2 2 1 − γ4Kf (xn)) where Kf (xn) = f ′′(xn − 13 wn)wn , n = 0, 1, 2, . . . f ′(xn) This method is a variant of the famous super-Halley’s method. The results are obtained for γ4 = 19. 4. Kou et al. method (KLM2) Kou et al. (2007a): yn = xn − wn, xn+1 = xn − 1 + s2n (5.4.4) (1 − sn) wn, n = 0, 1, 2, . . . Here, sn = f (xn) f (yn) . Further, Table 5.3 displays some nonlinear functions (f1 to f4) used to carry out the compu- tations. Table 5.3: Test functions, associated zeros and the initial approximations (x0) Function Real zero x0 f1(x) = x + sin2 x 0 0.05 f2(x) = sin x 0 0.50 f3(x) = 5i=1(x − i) 2 1.80 f4(x) = (x10 + x + 2)(x − 2)e−5x 2 1.80 ∏ In addition, some real-life chemistry problems are also solved after transforming them to nonlinear functions (f5 to f8). COC (computational order of convergence) and the errors of approximations to the desired roots, | xn − κ | for n = 1, 2, 3 of ft(x), t = 1, 2, . . . , 8 are outlined in Tables 5.4- 5.11. Real-life problems: Next, we describe a few real-life problems together with the computa- tional outcomes: Example 5.4.1. Virial state equation: Virial state equation, (see Wilczek-Vera and Vera (2015)) is given as V = UU1U32 1.0 + p1 + p 2 V V 2 . Here, U1 is the universal gas constant, U2 i(s the absolute te)mperature and U3 is the pressure. Using some specific gas parameters values , U1 = 82.05, U2 = 430.85, U3 = 75, p1 = −159, p2 = 9000, and taking V as variable x, the attained nonlinear equation is f5(x) = x − 471.3499 + 749444.6310 4.242149100 x − x2 = 0, (5.4.5) Table 5.4: Numerical outcomes for f1(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f1 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 9.7 × 10−6 9.6 × 10−6 2.0 × 10−5 1.4 × 10−5 6.2 × 10−6 5.4 × 10−7 5.3 × 10−6 1.4 × 10−6 1.8 × 10−20 1.7 × 10−20 7.0 × 10−19 1.2 × 10−19 1.8 × 10−21 1.0 × 10−31 7.9 × 10−22 1.6 × 10−24 1.9 × 10−79 1.5 × 10−79 1.0 × 10−72 6.3 × 10−76 1.2 × 10−83 2.5 × 10−155 3.8 × 10−85 2.9 × 10−96 4.0000 4.0000 4.0000 4.0000 4.0000 5.0000 4.0000 4.0000 Table 5.5: Numerical outcomes for f2(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f2 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 2.4 × 10−3 2.4 × 10−3 3.3 × 10−3 2.8 × 10−3 2.2 × 10−3 1.8 × 10−3 2.1 × 10−3 1.6 × 10−3 4.1 × 10−15 4.4 × 10−15 1.8 × 10−14 1.0 × 10−14 2.6 × 10−15 9.5 × 10−16 2.2 × 10−15 6.6 × 10−16 6.1 × 10−74 9.1 × 10−74 8.2 × 10−71 6.7 × 10−72 7.2 × 10−75 4.3 × 10−77 3.2 × 10−75 6.7 × 10−78 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 Table 5.6: Numerical outcomes for f3(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f3 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 7.5 × 10−4 7.5 × 10−4 4.1 × 10−4 7.4 × 10−4 6.6 × 10−4 6.0 × 10−4 6.5 × 10−4 5.8 × 10−4 6.6 × 10−13 6.3 × 10−13 8.7 × 10−14 7.4 × 10−13 2.6 × 10−13 9.1 × 10−14 2.3 × 10−13 5.2 × 10−14 4.0 × 10−49 3.2 × 10−49 1.8 × 10−52 7.1 × 10−49 6.8 × 10−51 4.7 × 10−53 3.5 × 10−51 3.5 × 10−54 3.9996 3.9996 3.9998 3.9996 3.9997 3.9999 3.9998 4.0000 with one real and two complex zeros. We choose the zero κ ≈ 5.6604 × 10−6 and the initial guess is taken to be 4.2 × 10−6. The computational results are outlined in Table 5.8. Table 5.7: Numerical outcomes for f4(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f4 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 1.5 × 10−3 1.5 × 10−3 2.1 × 10−3 1.8 × 10−3 1.3 × 10−3 1.0 × 10−3 1.2 × 10−3 9.5 × 10−4 1.2 × 10−13 1.4 × 10−13 1.9 × 10−12 2.6 × 10−13 7.3 × 10−14 2.8 × 10−14 6.3 × 10−14 2.0 × 10−14 5.4 × 10−54 7.5 × 10−54 1.1 × 10−48 1.0 × 10−52 6.3 × 10−55 1.4 × 10−56 3.4 × 10−55 3.6 × 10−57 4.0075 4.0077 3.9976 4.0091 4.0066 4.0051 4.0063 4.0047 Table 5.8: Numerical outcomes for Example 5.4.1. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f5 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 − − 2.2 × 10−7 1.6 × 10−7 5.9 × 10−8 1.1 × 10−7 4.0 × 10−8 2.2 × 10−7 − − 3.3 × 10−10 6.3 × 10−11 2.4 × 10−13 5.4 × 10−12 2.8 × 10−14 1.4 × 10−10 − − 1.9 × 10−21 2.8 × 10−22 2.8 × 10−22 2.8 × 10−22 2.8 × 10−22 3.0 × 10−22 D D 3.9174 3.9539 3.9916 4.0207 3.9955 4.0528 D- Divergent Example 5.4.2. Van der Waals state equation: As described in Example 2.4.2, the nonlinear equation for the said problem after taking specific parameters' values is as follows: f6(x) = x3 − 5.22x2 + 9.0825x − 5.2675 = 0, (5.4.6) with three roots. Among them, the simple root κ = 1.72 is chosen and the initial guess is taken to be 1.70. Table 5.9 outlines the results. Example 5.4.3. Chemical reactor problem: As described in Example 2.4.3, the nonlinear equation for the said problem is as follows: f7(x) = 1 − x − 5 log 0.4 − 0.5x + 4.45977 = 0. x 0.4(1 − x) (5.4.7) The desired root is κ ≈ 0.7573962462537538. The outcomes are presented in Table 5.10 taking x0 = 0.72. Table 5.9: Numerical outcomes for Example 5.4.2. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f6 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 3.3 × 10−3 3.1 × 10−3 4.2 × 10−3 3.7 × 10−3 2.6 × 10−3 1.0 × 10−3 2.4 × 10−3 4.0 × 10−5 2.9 × 10−5 2.3 × 10−5 1.1 × 10−4 5.6 × 10−5 7.8 × 10−6 4.4 × 10−8 5.0 × 10−6 5.1 × 10−13 3.4 × 10−13 1.4 × 10−13 1.7 × 10−10 7.6 × 10−12 1.0 × 10−15 2.7 × 10−17 1.1 × 10−16 2.7 × 10−17 3.6801 3.6933 3.4211 3.5685 3.8014 3.9227 3.8333 3.9987 Table 5.10: Numerical outcomes for Example 5.4.3. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f7 (x) K T M S S M K LM1 K LM2 P M1 P M2 P M3 P M4 1.7 × 10−2 − − 3.6 × 10−2 2.2 × 10−3 9.9 × 10−3 1.9 × 10−4 1.3 × 10−2 2.0 × 10−3 − − 1.3 × 10−2 1.3 × 10−8 3.7 × 10−5 2.7 × 10−13 1.4 × 10−4 1.9 × 10−7 − − 2.2 × 10−4 8.8 × 10−17 7.5 × 10−15 8.8 × 10−17 2.5 × 10−12 3.7523 D D 2.4506 3.9858 4.0888 4.0001 4.1410 D- Divergent Example 5.4.4. Continuous stirred tank reactor (CSTR) problem: Here, we consider the isothermal CSTR (see Constantinides and Mostoufi (1999)). The re- actor receives inputs A1 and U4 at rates of Q and q − Q, respectively. In the reactor, the following reaction process is then achieved: A1 + U4 → A2, A2 + U4 → A3, A3 + U4 → A4, A3 + U4 → A5. While designing a simple model for feedback control systems, Douglas (1972) examined the above-mentioned model which was converted into the following expression: U5 2.98(x + 2.25) (x + 1.45)(x + 4.35)(x + 2.85)2 = −1. Here, U5 is the proportional controller gain. For values of U5 that result in zeros of the transfer function with a negative real part, the control system is stable. Upon choosing U5 = 0, the poles of the feedback-free transfer function can be acquired as zeros of the equation given as f8(x) = x4 + 11.50x3 + 47.49x2 + 83.06325x + 51.23266875 = 0. (5.4.8) (5.4.8) has four zeros but our desired simple zero being κ = −4.35. Table 5.11 outlines the results with x0 = −4.13. Table 5.11: Numerical outcomes for Example 5.4.4. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρ f8 (x) KTM 7.9 × 10−2 SSM − KLM1 8.0 × 10−1 KLM2 2.6 × 10−1 PM1 3.7 × 10−2 PM2 4.2 × 10−2 PM3 2.1 × 10−2 PM4 6.3 × 10−2 2.3 × 10−4 − 1.9 × 10−1 1.3 × 10−2 6.4 × 10−6 7.4 × 10−6 5.9 × 10−7 8.3 × 10−5 2.4 × 10−14 − 6.8 × 10−3 2.8 × 10−7 3.6 × 10−16 3.6 × 10−16 3.6 × 10−16 1.9 × 10−16 3.8413 NC 1.6352 3.1329 3.9520 4.0822 3.9782 4.1370 NC- Not converging in desired iterations Remark 5.4.1. It can be seen from Tables 5.4-5.7 that for f1 to f4, in comparison to the previous techniques, we get a noticeably lower approximation error with the proposed methods P M1, P M2, P M3 and P M4 as they converge to the solution. Remark 5.4.2. From Tables 5.8-5.11, it can be confirmed that for Example 5.4.1, KTM and SSM diverge. Similarly, for Example 5.4.3, SSM and KLM1 diverge. For Example 5.4.4, SSM is not converging to the zero in required iterations and methods KLM1 and KLM2 exhibit a very slow convergence in this case. In addition, for Example 5.4.2, a least amount of error can be observed for our methods. Remark 5.4.3. A comparison has been made between our methods and several existing meth- ods. The results show that our methods outperform the existing methods in numerous scenarios in terms of COC and errors as revealed in Tables 5.4-5.11. Furthermore, the aforementioned tables demonstrate a significant decrease in approximation error for our methods. 5.5 Conclusions In this work, the complex dynamic behavior of an optimal IF has been explored by analyzing its stability properties. We can better understand how a given method of an IF depends on initial guesses if the aforementioned aspect is taken into consideration. A rational operator is constructed by applying a generic quadratic polynomial to the IF, taking into account the scaling theorem and the Möbius transformation theory in conjugation on the Riemann sphere. The fixed and critical points are evaluated, and it is inspected that each pair of strange fixed points as well as free critical points were conjugate to each other. Thereupon, we examine the stability of roughly half of them. Additionally, we generated the dynamical and parameter planes whose study helps to distinguish the stable or unstable family members which in turn can be employed to several practical problems. When our methods are compared to certain previous techniques, numerical outcomes in- dicate that our methods perform better since they produce smaller absolute errors than the existing IMs. Furthermore, the chemical reactor problem, equations of state and continuous stirred tank reactor problems are only a few domains in which our methods have demonstrated their adaptability and usefulness. As a result, researchers working in domains where precise root finding is essential will find great value in our methods. 3 3 2 2 1 1 Im{x} 0 Im{x} 0 -1 -1 -2 -2 -3 -3 -3 -2 -1 0 1 2 3 -3 -2 -1 0 1 2 3 Re{x} Re{x} 3 3 2 2 1 1 Im{x} 0 Im{x} 0 -1 -1 -2 -2 -3 -3 -3 -2 -1 0 1 2 3 -3 -2 -1

0 1 2 3 Re{x} Re{x} 3 3 2 2 1 1 Im{x} 0 Im{x} 0 -1 -1 -2 -2 -3 -3 -3 -2 -1 0 1 2 3 -3 -2 -1 0 1 2 3 Re{x} Re{x} Figure 5.4: Dynamical planes for γ1 = 1, 3, −1, − 1170 , − , − 10 , respectively 3 9 2 3 3 2 2 1 1 Im{x} 0 Im{x} 0 -1 -1 -2 -2 -3 -3 -3 -2 -1 0 1 2 3 -3 -2 -1 0 1 2 3 Re{x} Re{x} 3 3 2 2 1 1 Im{x} 0 Im{x} 0 -1 -1 -2 -2 -3 -3 -3 -2 -1 0 1 2 3 -3 -2 -1 0 1 2 3 Re{x} Re{x} Figure 5.5: Dynamical planes for γ1 = − , − 17 7 3 5 , − , −3, respectively 7 2 Chapter 6 An Optimal Derivative-free Fourth Order Metho d and its Memory Variant for Nonlinear Mo dels An optimal iterative scheme without memory free from derivatives has been presented in this chapter for solving nonlinear equations. There are many iterative schemes existing in the literature which either diverge or fail to work when f ′(x) = 0. But, our proposed scheme works even in those cases. In addition, we also extended the same idea for iterative method with memory with the help of self-accelerating parameters estimated from the current and previous approximations. As a result, the order of convergence increased from 4 to 7 without the addition of any further functional evaluation . 6.1 Introduction The previous chapters dealt with iterative scheme involving first order derivatives. One drawback of such kind of methods is that when f ′(xn) = 0, the methods fail, which confines their applications. Therefore, a lot of researchers are interested towards building optimal multipoint methods free from derivatives. These methods prove to be very helpful in the cases where the derivative of the function is cumbersome to evaluate or is expensive to compute. In order to circumvent the clause of derivatives, Traub-Steffensen used the approximation, f′(xn) ≡ f(xn + βf(xn)) − f(xn), β ∈ R\{0} βf(xn) which when replaced in the Newton’s method taking the form, xn+1 = xn − f [wn, xn] f (xn) , (6.1.1) (6.1.2) The contents of this chapter are published in: Mathematical and Computational Applications (MDPI), 28(2), 48, 2023 (Scopus, Q2, Impact Factor: 1.9) where wn = xn + βf(xn), termed as Traub-Steffensen method. This method is free from derivative and is a significant improvement of Newton’s method because it retains its quadratic convergence without the addition of the derivative. In literature, various procedures have been used for approximating the first order deriva- Gp(x) tives. For example, one of the procedure uses Padé approximant of the form Hq(x), where Gp(x) and Hq(x) are polynomials of degree p and q, respectively. Cordero et al. Cordero et al. (2013c) used the first degree Padé approximant to approximate the first order deriva- tive. Another procedure uses interpolating polynomial of degree n that interpolates n + 1 already known values of the specified function Cordero et al. (2013b). Additionally, procedure by using the divided difference can also be used. Following the steps of Traub, many authors are constructing higher order methods with- out derivatives. Among many others, Chicharro et al. Chicharro et al. (2017) presented a biparametric family of order four and then developed a family of methods with memory having higher order of convergence without further increasing the number of functional evaluations per iteration. In Sharifi et al. (2016), the authors presented a derivative free form of King’s family with memory. Kansal et al. (2016) developed a tri-parametric derivative-free family of Hansen-Patrick type methods which requires only three functional evaluations to achieve optimal fourth order of convergence. Then, they extended the idea to with memory as a result of which the R-order of convergence is increased from 4 to 7, without any additional functional evaluation. Thus, by taking into consideration these developments, we further attempt to propose an iterative method without memory free from derivatives and then convert it into more efficient method with memory such that the order of convergence is increased without any further functional evaluation. We carry out the convergence analysis of the new methods in order to demonstrate their order of convergence. To illustrate our theoretical results, numerical results for the proposed methods and comparisons with some of the existing methods are then given. Lastly, we present the basins of attraction of the new methods which demonstrate the convergence or divergence of the new as well as the existing methods. 6.2 An optimal iterative method without memory and its convergence analysis We aim to construct a new two-point derivative-free optimal scheme without memory in this section and extend it to with memory scheme. If the well-known Steffensen’s method is composed with Newton’s method, we get the following fourth order scheme: yn = xn − f (wn) − f (xn) f (xn)2 , xn+1 = yn − f ′(yn) f (yn) (6.2.1) , where wn = xn + f (xn). To avoid the computation of f ′(yn), Cordero et al. (2013c) approxi- mated it by the derivative m′(yn) of the following first degree Padé approximant: a1 + a2(t − yn) m(t) = 1 + a3(t − yn) , (6.2.2) where a1, a2 and a3 are real parameters to be determined satisfying the following conditions: m(xn) = f (xn), (6.2.3) m(yn) = f (yn), (6.2.4) m(wn) = f (wn). (6.2.5) Using these conditions, the derivative of the Padé approximant evaluated in yn is given as m′(yn) = f [xn, yn]f [yn, wn] . (6.2.6) f [xn, wn] Using (6.2.6) in second step of (6.2.1), they presented the following scheme: yn = xn − f (wn) − f (xn) f (xn)2 , f (yn)f [xn, wn] (6.2.7) xn+1 = yn − f [xn, yn]f [yn, wn] , where wn = xn+f(xn). This scheme is optimal in the sense of Kung−Traub conjecture having order of convergence 4 with three functional evaluations per iteration, f(xn), f(yn) and f(wn). Now, in order to extend to the method with memory, we come up with an idea of intro- ducing two parameters γ and ξ in (6.2.7) and we present a modification in this method as follows: yn = xn − f (xn) , f [xn, wn] + ξf (wn) (6.2.8) xn+1 = yn − (f [xn, yn] + ξf (wn))f [yn, wn] f (yn)(f [xn, wn] + ξf (wn)) , where wn = xn + γf(xn). This modified scheme yields the optimal order of convergence 4 having three functional evaluations per iteration, f(xn), f(yn) and f(wn). Next, we establish the convergence results for our proposed family without memory given by Equation (6.2.8). 6.2.1 Convergence analysis Theorem 6.2.1. Suppose that f : D ⊂ R → R be a real function suitably differentiable in a domain D. If κ ∈ D is a simple root of f (x) = 0 and an initial guess x0 is sufficiently close to κ, then the iterative method given by Equation (6.2.8) converges to κ with convergence order p = 4 having the following error relation, en+1 = (1 + f′(κ) γ)2(ξ + d2) (2 + f′(κ)γ)ξd2 + 2d22 − d3 e4n + O(en)5, (where en = xn − κ, κ is a simple root of f (x) = 0 and dn = 1 f (n))(κ) n! f ′(κ) , n = 2, 3, . . . Proof. Expanding f (xn) about xn = κ by

Taylor series, we have f (xn) = f ′(κ)(en + d2e2n + d3e3n + d4e4n) + O(en)5. (6.2.9) Using Equation (6.2.9) in the first step of Equation (6.2.8), we have en,y = yn − κ = (1 + f′(κ)γ)(ξ + d2) e2n + (−(2 + 2f′(κ)γ + f′(κ)2γ2)ξd2 − (2 + 2f′(κ)γ + f′(κ)2γ2)d22 − (1 + f′(κ)γ)((1 + f′(κ)γ)ξ2 − (2 + f′(κ)γ)d3))e3n + ((5 + 7f′(κ)γ + 4f′(κ)2γ2 + f′(κ)3γ3)ξd22 + (4 + 5f′(κ)γ + 3f′(κ)2γ2 + f′(κ)3γ3)d32 − (4 + 7f′(κ)γ + 5f′(κ)2γ2 + f′(κ)3γ3)ξd3 − d2(−(3 + 5f′(κ)γ + 3f′(κ)2γ2 + f′(κ)3γ3)ξ2 + (7 + 10f′(κ)γ + 7f′(κ)2γ2 + 2f′(κ)3γ3)d3) + (1 + f′(κ)γ)((1 + f′(κ)γ)2ξ3 + (3 + 3f′(κ)γ + f′(κ)2γ2)d4))e4n + O(en)5. (6.2.10) Also, the Taylor's expansion of f(yn) is f(yn) = f′(κ)(en,y + d2e2n,y + d3e3n,y + d4e4n,y) + O(en,y)5. (6.2.11) Using Equations (6.2.9)−(6.2.11), we have (ff[(xynn,)y(nf][x+nξ,wf(nw]+n))ξff[(ywn,nw))n] = (1 + f′(κ)γ)(ξ + d2)e2n + (−(2 + 2f′(κ)γ + f′(κ)2γ2)ξd2 − (2 + 2f′(κ)γ + f′(κ)2γ2)d22 − (1 + f′(κ)γ)((1 + f′(κ)γ)ξ2 − (2 + f′(κ)γ)d3))e3n + ((1 − 2f′(κ)γ − 2f′(κ)2γ2)ξd22 + (2 + f′(κ)γ + f′(κ)2γ2 + f′(κ)3γ3)d32 − (3 + 5f′(κ)γ + 4f′(κ)2γ2 + f′(κ)3γ3)ξd3 − d2((−1 + f′(κ)2γ2)ξ2 + 2(3 + 4f′(κ)γ + 3f′(κ)2γ2 + f′(κ)3γ3)d3) + (1 + f′(κ)γ)((1 + f′(κ)γ)2ξ3 + (3 + 3f′(κ)γ + f′(κ)2γ2)d4))e4n + O(en)5. (6.2.12) Finally, putting Equation (6.2.12) in the second step of Equation (6.2.8), we get en+1 = (1 + f′(κ)γ)2(ξ + d2) (2 + f′(κ)γ)ξd2 + 2d22 − d3 e4n + O(en)5, (6.2.13) which is the error equation for the (proposed optimal scheme g)iven by Equation (6.2.8) with convergence order four. This completes the proof. 6.3 An iterative method with memory and its conver- gence analysis Now, we present an extension to the method given by Equation (6.2.8) by inclusion of memory having improved convergence order without the addition of any new functional evaluation. If we observe clearly, it can be seen from the error relation given in Equation (6.2.13) that the order of convergence of the proposed family given by Equation (6.2.8) is 4 if γ ≠ f′(κ) −1 and ξ ≠ −d2. Therefore, if γ = f′(κ) and ξ = −d2 = −2f′(κ), then the order of convergence −1 f′′(κ) of our proposed family can possibly be improved, but this value can't be reached because the values of f ′(κ) and f ′′(κ) are not practically available. Instead, we can use approximations calculated by already available information. Hence, the main idea in constructing methods with memory consists of the calculation of the parameters γ = γn and ξ = ξn as the iteration proceeds by the formulaes, γn = f′(κ) and ξn = −d2 = −2f′(κ) −1 f′′(κ) for n = 1, 2, . . . Further, it is also assumed that the initial estimates γ0 and ξ0 must be chosen before starting the iterations. Thus, we give an estimation for γn and ξn given by γn = N3−′(x1n) and ξn = −2NN44′′′((wwnn)) , (6.3.1) where N3(k) = N3(k; xn, xn−1, yn−1, wn−1) and N4(k) = N4(k; wn, xn, wn−1, yn−1, xn−1) are Newton's interpolating polynomials of third and fourth degrees, respectively which are set through best available nodal points, (xn, xn−1, yn−1, wn−1) for N3 and (wn, xn, wn−1, yn−1, xn−1) for N4. So, by replacing γ by γn and ξ by ξn in the method given by Equation (6.2.8), we obtain a new family with memory as follows: γ0, ξ0, x0 are given, w0 = x0 + γ0f (x0) γn = N3−′(x1n) , wn = xn + γnf (xn), ξn = −2NN44′′′((wwnn)) , n = 1, 2, . . . , yn = xn − f (xn) , (6.3.2) f [xn, wn] + ξnf (wn) xn+1 = yn − (f [xn, yn] + ξnf (wn))f [yn, wn] f (yn)(f [xn, wn] + ξnf (wn)) . Next, we establish the convergence results for our proposed family with memory given by Equation (6.3.2). 6.3.1 Convergence analysis Theorem 6.3.1. Suppose that f : D ⊂ R → R be a real function suitably differentiable in a domain D. If κ ∈ D is a simple root of f (x) = 0 and an initial guess x0 is sufficiently close to κ, then the iterative method given by Equation (6.3.2) converges to κ with convergence order at least 7. Proof. Let {xn} be a sequence of approximations generated by an iterative method (IM). If this sequence converges to the zero κ of f with the R-order (≥ r) of IM, then we write en+1 ∼ Dn,rern, en = xn − κ, where Dn,r tends to the asymptotic error constant Dr of IM, when n → ∞. Thus, en+1 ∼ Dn,r(Dn−1,rern−1)r = Dn,rDnr−1,rern2−1 (6.3.3) (6.3.4) Let the iterative sequences {wn} and {yn} have R-orders r1 and r2, respectively. Therefore, we obtain en,w = wn − κ ∼ Dn,r1ern1 ∼ Dn,r1(Dn−1,rern−1)r1 = Dn,r1Dnr1−1,rernr−11 and en,y = yn − κ ∼ Dn,r2ern2 ∼ Dn,r2(Dn−1,rern−1)r2 = Dn,r2Dnr2−1 ,rernr−21. Using (6.3.5), (6.3.6) and a lemma stated in Kansal et al. (2016), we obtain 1 + γnf′(κ) ∼ ψ1en−1,wen−1,yen−1 = ψ1Dn−1,r1Dn−1,r2ern1−+1r2+1, ξn + d2 ∼ ψ2en−1,wen−1,yen−1 = ψ2Dn−1,r1Dn−1,r2ern1−+1r2+1. (6.3.5) (6.3.6) (6.3.7) In view of our proposed family of methods without memory given by Equation (6.2.8), we have the following error relations, en,w = (1 + γf′(κ))en + O(en)2, en,y = (1 + γf′(κ))(ξ + d2)e2n + O(en)3, en+1 = ϕ1(1 + γf′(κ))2(ξ + d2)e4n + O(en)5, (6.3.8) (6.3.9) (6.3.10) where ϕ1 = (2 + f′(κ)γ)ξd2 + 2d22 − d3. According to the error relations given by Equations (6.3.8)−(6.3.10) with self-accelerating parameters, γ = γn and ξ = ξn, we can write the corresponding error relations for the methods given by Equation (6.3.2) with memory as follows: en,w ∼ (1 + γnf ′(κ))en, en,y ∼ (1 + γnf ′(κ))(ξn + d2)e2n, en+1 ∼ ϕ2(1 + γnf ′(κ))2(ξn + d2)e4n, (6.3.11) (6.3.12) (6.3.13) where ϕ2 = (2 + f ′(κ)γn)ξnd2 + 2d22 − d3 depending on iteration index since γn and ξn are re-calculated in each step. Now using Equations (6.3.7) and (6.3.11)−(6.3.13), we get the following relations: en,w ∼ (1 + γnf′(κ))en ∼ ψ1Dn−1,r1Dn−1,r2Dn−1,rern+−r11+r2+1, (6.3.14) en,y ∼ (1 + γnf′(κ))(ξn + d2)e2n ∼ ψ1ψ2Dn2−1,r1Dn2−1,r2Dn2−1,re2nr−+12r1+2r2+2, (6.3.15) en+1 ∼ ϕ2(1 + γnf′(κ))2(ξn + d2)e4n ∼ ϕ2ψ12ψ2D3n−1,r1Dn3−1,r2Dn4−1,re4nr−+13r1+3r2+3. (6.3.16) Now, comparing the error exponents of en−1 on the right hand sides of pairs given by Equations (6.3.5) with (6.3.14), (6.3.6) with (6.3.15) and (6.3.4) with (6.3.16), respectively, we obtain the following system of equations: rr1 − r − r1 − r2 = 1, rr2 − 2r − 2r1 − 2r2 = 2, (6.3.17) r2 − 4r − 3r1 − 3r2 = 3. Solving this system of equations, we get a non-trivial solution as r1 = 2, r2 = 4 and r = 7. Hence, we can conclude that the lower bound of the R-order of our proposed family with memory given by Equation (6.3.2) is 7. This completes our proof. 6.4 Numerical results This section lays out the comparison of our schemes (6.2.8), denoted by P M and (6.3.2), denoted by P M M with several existing schemes. The initial values of γ (or γ0) and ξ (or ξ0) are assumed to be chosen beforehand to begin with the computations. Also, a suitable x0 must be fixed. We have taken γ (or γ0)= −0.1 and ξ (or ξ0)= 0.1 in our computations. The following existing methods have been selected to facilitate comparisons with our methods: 1. Soleymani et al. method (SM) without memory Soleymani et al. (2012c): yn = xn − f [xn, wn] , wn = xn + γf (xn), γ ∈ R\{0}, f (xn) 2 xn+1 = xn − f (xn) + f (yn) − 2f (xn) + αf (yn) f (yn) γf [xn, wn] (f [xn, wn] n = 0, 1, 2, . . .) (f [xn, wn]) (f (xn)) (1 − 2 + 2γf [xn, wn]) , α ∈ R, (6.4.1) We have taken a particular case for this family when α = 10, γ = −0.01. 2. Cordero et al. method (AM1) without memory Cordero et al. (2013c): yn = xn − f [xn, wn] , wn = xn + f (xn), f (xn) (6.4.2) xn+1 = yn − f [xn, yn]f [yn, wn] , n = 0, 1, 2, . . . f (yn)f [xn, wn] 3. Chun method (CM) without memory Chun (2007g): yn = xn − f ′(xn) f (xn) , xn+1 = xn − ff′((xxnn))(1 + u + 2u2), u = (6.4.3) f (xn) , n = 0, 1, 2, . . . f (yn) 4. Cordero et al. method (AM2) with memory Cordero et al. (2015): γ0, ξ0, x0 are given, w0 = x0 + γ0f (x0), γn = N3−′(x1n) , wn = xn + γnf (xn), ξn = −2NN44′′′((wwnn)) , n = 1, 2, . . . , yn = xn − f (xn) , (6.4.4) f [xn, wn] + ξnf (wn) xn+1 = yn − f (yn) f [xn, yn] + (yn − xn)f [xn, wn, yn] , where N3 and N4 are as defined in Section 6.3 and γ0 = ξ0 = 0.1 are used in results. 5. Dˇzunić method (DM1 and DM2) with memory Džunić (2013): γ0, ξ0, x0 are given, w0 = x0 + γ0f (x0), γn = N3−′(x1n) , wn = xn + γnf (xn), ξn = −2NN44′′′((wwnn)) , n = 1, 2, . . . , yn = xn − f (xn) , (6.4.5) f [xn, wn] + ξnf (wn) xn+1 = yn − f [yn, wn] + ξnf (wn) , tn = f (xn) f (yn)g(tn) f (yn) , where N3 and N4 are as defined in Section 6.3 and γ0 = ξ0 = 0.1 are used in results. The particular functions are taken for the methods, which are g(t) = 1 + t for DM1 and g(t) = 1/(1 − t) for DM2. Further, Table 6.1 displays some nonlinear functions (f1 to f5) used to carry out the compu- tations. In addition, some real-life problems are also solved after transforming them to nonlinear functions (f6 and f7). COC (computational order of convergence) and the errors of approxi- mations to the desired roots, | xn − κ | for n = 1, 2, 3 of ft(x), t = 1, 2, . . . , 7 are outlined in Tables 6.2−6.8. Real-life problems: Next, we describe a few real-life problems together with the computa- tional outcomes: Table 6.1: Test functions, associated zeros and the initial approximations (x0) Function Real zero x0 f1(x) = (x − 2)(x10 + x + 1)e−x−1 f2(x) = ex2+7x−30 − 1 f3(x) = sin(πx)ex2+x cos x−1 + x log(x sin x + 1) f4(x) = ex3−x − cos(x2 − 1) + x3 + 1 f5(x) = ex2−3x sin x + log(x2 + 1) 2 3 0 −1 0 1.925 2.90 0.05 −1.10 0.05 Table 6.2: Numerical outcomes for f1(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f1 (x) Without memory P M 1.1 × 10−2 S M 4.6 × 10−2 AM1 – C M 6.6 × 10−2 3.5 × 10−5 1.5 × 10−3 – 1.8 × 10−3 2.4 × 10−15 1.8 × 10−10 – 3.2 × 10−9 4.0308 4.8888 F 3.4574 0.0029 0.0038 – 0.0023 With memory P M M 1.1 × 10−2 AM2 3.8 × 10−1 DM1 – DM2 9.5 × 10−1 2.0 × 10−11 1.8 × 10−2 – 7.7 × 10−2 1.7 × 10−72 7.0 × 10−12 – 3.8 × 10−6 6.9728 4.8678 NC 1.9871 0.0131 0.0141 – 0.0154 F- Method fails NC- Not converging in desired iterations Table 6.3: Numerical outcomes for f2(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f2 (x) Without memory P M 5.3 × 10−3 S M – AM1 – C M – 3.7 × 10−8 – – – 6.3 × 10−29 – – – 4.0108 F F NC 0.0027 – – – With memory P M M AM2 DM1 DM2 5.3 × 10−3 5.2 × 10−2 – – 2.2 × 10−12 3.2 × 10−6 – – 6.5 × 10−78 4.6 × 10−35 – – 6.9741 6.6121 F NC 0.0093 0.0100 – – F- Method fails NC- Not converging in desired iterations Table 6.4: Numerical outcomes for f3(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f3 (x) Without memory P M 7.7 × 10−6 S M 2.2 × 10−5 AM1 3.9 × 10−5 C M 2.3 × 10−5 4.4 × 10−21 1.4 × 10−18 1.3 × 10−17 1.1 × 10−18 4.6 × 10−82 2.1 × 10−71 1.9 × 10−67 7.5 × 10−72 4.0000 4.0000 4.0000 4.0000 0.0084 0.0120 0.0081 0.0069 With memory P M M 7.7 × 10−6 AM2 4.3 × 10−6 DM1 2.2 × 10−5 DM2 1.2 × 10−5 4.8 × 10−38 1.2 × 10−37 7.3 × 10−34 3.6 × 10−36 7.5 × 10−261 2.3 × 10−258 7.0 × 10−232 5.7 × 10−248 6.9199 6.9946 6.9537 6.9365 0.0312 0.0340 0.0379 0.0344 Table 6.5: Numerical outcomes for f4(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f4 (x) Without memory P M 3.7 × 10−6 S M 1.4 × 10−5 AM1 9.0 × 10−5 C M 2.3 × 10−5 1.6 × 10−23 2.5 × 10−21 1.2 × 10−15 1.8 × 10−19 6.7 × 10−93 2.4 × 10−84 3.9 × 10−59 7.5 × 10−76 4.0000 4.0000 4.0001 4.0000 0.0030 0.0054 0.0036 0.0025 With memory P M M AM2 DM1 DM2 3.7 × 10−6 1.2 × 10−5 1.3 × 10−5 1.3 × 10−5 3.0 × 10−39 2.6 × 10−35 2.9 × 10−35 2.8 × 10−35 2.1 × 10−271 6.1 × 10−244 1.1 × 10−243 9.2 × 10−244 7.0152 7.0303 7.0301 7.0301 0.0141 0.0151 0.0168 0.0199 Example 6.4.1. Chemical reactor problem: As described in Example 2.4.3, the nonlinear equation for the said problem is as follows: f6(x) = 1 − x − 5 log 0.4 − 0.5x + 4.45977 = 0. x 0.4(1 − x) (6.4.6) The desired root is κ ≈ 0.7573962462537538. Example 6.4.2. Multi-factor effect: As described in Example 2.4.4, the nonlinear equation for the said problem is as follows: f7(x) = x − cos x + = 0. 1 π 2 4 (6.4.7) The desired root of Equation (6.4.7) is κ ≈ −0.3090932715417949. Table 6.6: Numerical outcomes for f5(x). Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f5 (x) Without memory P M 1.0 × 10−5 S M 3.8 × 10−4 AM1 3.6 × 10−5 C M 1.6 × 10−4 3.6 × 10−20 1.0 × 10−12 3.6 × 10−17 2.1 × 10−14 6.1 × 10−78 5.6 × 10−47 3.5 × 10−65 5.4 × 10−54 4.0000 4.0003 4.0000 3.9999 0.0047 0.0058 0.0040 0.0035 With memory P M M AM2 DM1 DM2 1.0 × 10−5 2.6 × 10−5 6.0 × 10−5 1.3 × 10−5 6.6 × 10−34 6.2 × 10−31 6.9 × 10−28 3.2 × 10−32 9.9 × 10−231 4.5 × 10−211 9.8 × 10−190 4.0 × 10−220 6.9827 7.0310 7.0557 7.0625 0.0169 0.0198 0.0200 0.0192 Table 6.7: Numerical outcomes for Example 6.4.1. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f6 (x) Without memory P M 7.5 × 10−3 S M 1.4 × 10−3 AM1 – C M 1.0 × 10−3 1.0 × 10−3 5.4 × 10−7 – 1.7 × 10−8 3.8 × 10−7 8.8 × 10−17 – 8.8 × 10−17 3.7581 4.0239 F 3.9915 0.0063 0.0052 – 0.0023 With memory P M M 7.4 × 10−3 AM2 3.5 × 10−4 DM1 8.3 × 10−2 DM2 4.4 × 10−2 9.0 × 10−8 1.7 × 10−13 3.3 × 10−2 2.5 × 10−2 8.8 × 10−17 8.8 × 10−17 1.0 × 10−2 4.9 × 10−3 7.1919 7.7953 1.8843 1.0704 0.0326 0.0178 0.0282 0.0296 F- Method fails Remark 6.4.1. It can be seen from Table 6.2 that for the function f1, AM1 fails to give solution and DM1 requires more than 3 iterations to converge to the root. Also, PMM converges to the desired root with error of approximations much lower than AM2 and DM2. For the function f2, SM, AM1 and DM1 fail to provide solution and CM and DM2 do not converge to the desired solution in 3 iterations whihc can be seen in Table 6.3. Furthermore, for functions f3, f4 and f5, the proposed methods PM and PMM converge to the required root with minimum error as compared to the existing methods. Remark 6.4.2. The proposed schemes given by Equations (6.2.8) and (6.3.2) have been com- pared to some already existing methods and it can be seen from the computational results that our proposed schemes give results in many of the cases where the existing methods fail in terms of COC and errors as depicted in Tables 6.2−6.8. Our methods display a noticeable decrease Table 6.8: Numerical outcomes for Example 6.4.2. Methods |x1 − κ| |x2 − κ| |x3 − κ| ρc CPU Time f7 (x) Without memory P M S M AM1 C M 1.1 × 10−3 8.6 × 10−4 2.4 × 10−3 1.6 × 10−3 8.4 × 10−14 6.5 × 10−14 3.9 × 10−12 6.6 × 10−13 3.0 × 10−31 3.0 × 10−31 3.0 × 10−31 3.0 × 10−31 3.9999 4.0001 3.9998 3.9998 0.0028 0.0040 0.0023 0.0021 With memory P M M 1.1 × 10−3 AM2 8.5 × 10−4 DM1 2.5 × 10−3 DM2 1.6 × 10−3 5.2 × 10−26 5.7 × 10−29 1.8 × 10−23 3.6 × 10−25 3.0 × 10−31 3.0 × 10−31 3.0 × 10−31 3.0 × 10−31 6.9718 6.8573 6.9345 6.9245 0.0133 0.0151 0.0159 0.0139 in the error in approximations as shown in above mentioned tables. Remark 6.4.3. From Tables 6.7 and 6.8, one can observe that for the function f6, the existing method AM1 fails to converge. In addition, for the function f7, an obvious decrease in the order of convergence of the existing methods can be noticed. Table 6.9: Comparison of different methods without and with memory in terms of Avg Iter and PNC Without memory methods Avg Iter PN C With memory methods Avg Iter PN C p1(z) P M 3.0552 S M 21.3618 AM1 3.3635 C M 3.8199 0.0067 0.8369 0.0001 0.0021 P M M AM2 DM1 DM2 2.6643 2.5278 4.3746 2.8281 0 0.0002 0.0753 0.0001 p2(z) P M 5.8428 S M 23.1283 AM1 9.8161 C M 6.3409 0.1062 0.9159 0.1125 0.0452 P M M AM2 DM1 DM2 4.8963 4.2219 10.5985 5.1956 0.0456 0.0193 0.3383 0.0070 p3(z) P M 8.4306 S M 23.7348 AM1 10.2165 C M 9.5562 0.2165 0.9430 0.0663 0.1695 P M M AM2 DM2 DM2 5.9777 6.3765 17.1381 7.8478 0.0361 0.0224 0.6309 0.0360 6.5 Basins of attraction Now, we will study the dynamics of the proposed as well as some mentioned existing methods by analyzing the behavior of their basins of attraction in the complex plane. Table 6.9 lists the average number of iterations denoted by Avg Iter and percentage of non-converging points denoted by PNC of the methods to generate the basins of attraction. We have considered the following test problems in order to demonstrate the dynamical behavior of the methods: Problem 6.5.1. The first function considered is p1(z) = z2−1. The roots of this function are 1, −1. The basins corresponding to our proposed method and the mentioned existing methods are shown in Figures 6.1 and 6.2. From Table 6.9, it can be seen that the proposed methods, PM and PMM converge to root in less number of iterations. Also, from the figures, it is observed that PMM converges to the root with no diverging points but the existing methods have some points painted as black. SM, in particular has very small basins. Problem 6.5.2. Second function taken is p2(z) = z3 − 1 having roots −1, 0.5 + 0.866i, 0.5 − 0.866i. Figures 6.3 and 6.4 show the basins for p2(z) in which it can be seen that SM, AM1 and DM1 have wider regions of divergence. Moreover, the average number of iterations taken by the proposed methods is less in each case in comparison to the existing methods. Problem 6.5.3. The third function considered is p3(z) = z4 − 1 having roots ±1, ±i. Fig- ures 6.5 and 6.6 show that SM, CM and DM1 have smaller basins. Although PM and PMM have some diverging points, yet they converge in less number of iterations faster than the existing methods. Remark 6.5.1. One can see from Figures 6.1−6.6 and Table 6.9 that our proposed methods have larger basins of attraction in comparison to the existing ones. In addition, there is a marginal increase in the average number of iterations per point of the existing methods. Consequently, through our proposed methods, the chances of non-convergence to the root are very less when compared to the existing methods. 6.6 Conclusions We have proposed a new fourth order optimal method without memory which is more efficient than several optimal schemes existing in the literature as illustrated through our com- putational results. Further, in order to increase the order of convergence, we have extended the proposed method without memory to with memory method without the addition of any new functional evaluation taking into consideration two self-accelerating parameters. Conse- quently, the order of convergence has increased from 4 to 7. Numerical results demonstrate that the proposed optimal method and its extension to memory are more effective than the other methods of the same order at the point considered, as they are converging to the root with higher rate. In addition, our proposed schemes give results in many of the cases where Figure 6.1: Basins of attraction for PM, SM, AM1, CM, respectively for p1(z) Figure 6.2: Basins of attraction for P M M , AM2, DM1, DM2, respectively for p1(z) the existing methods fail in terms of COC and errors. Moreover, we have also presented the basins of attraction for the proposed as well as some existing methods, which assert that the Figure 6.3: Basins of attraction for PM, SM, AM1, CM, respectively for p2(z) Figure 6.4: Basins of attraction for P M M , AM2, DM1, DM2, respectively for p2(z) chances of non-convergence to the root are very less in our proposed methods when compared to the existing methods. Figure 6.5: Basins of attraction for PM, SM, AM1, CM, respectively for p3(z) Figure 6.6: Basins of attraction for P M M , AM2, DM1, DM2, respectively for p3(z) Chapter 7 A Robust Iterative Family for Multiple Ro ots of Nonlinear Equations This chapter introduces an iterative method that exhibits an optimal fourth-order convergence rate, ensuring rapid and accurate approximation in case of multiple roots. Unlike conventional methods, the proposed algorithm can successfully converge even when the deriva- tive is zero or approaches zero in the vicinity of the desired root. This remarkable feature enhances the applicability of the method, allowing it to handle situations where conventional methods fail due to the presence of critical points such as the roots of f ′(x) = 0. Its ability to converge even in the presence of zero or near-zero derivatives significantly expands the scope of applications, making it a valuable tool for solving complex problems in science and engineering. 7.1 Introduction The previous chapters dealt with iterative schemes for determining simple roots of f (x) = 0. Locating its multiple roots with multiplicity m is one of the most important challenges in science and engineering. We must therefore research numerical techniques in this regard. The modified Newton’s method by Rall (1966), one of the prime numerical methods to approximate a multiple root κ with multiplicity m, is given by (1.5.20). It is quadratically convergent, but the multiplicity m must be determined beforehand. Numerous higher order methods have been developed by several authors to compute multiple roots by using weight functions or the Newton’s method as a starting step. The said methods need prior knowledge about the multiplicity m. Some of the instances include methods by Victory Jr and Neta (1983); Dong (1982) which are of order 3. Each of the aforementioned methods necessitates three function evaluations, The contents of this chapter are published in: Journal of Computational and Applied Mathematics (Elsevier), 444, 115795, 2024 (SCIE, Q1, Impact Factor: 2.1) along with two function evaluations and one derivative evaluation for every iteration. Con- sequently, in light of Traub’s well-known conjecture regarding the optimal order for methods without memory, the anticipated optimal convergence order for these methods should be four rather than three. Recently, there has been a growing number of optimal fourth-order iterative methods (see Zhou et al. (2011); Sharma and Sharma (2010); Li et al. (2010)). Inspired by the research in this area, an attempt has been made to present a new iterative family that only requires three functional evaluations to attain convergence of optimal fourth- order in accordance with the Kung-Traub conjecture Kung and Traub (1974). Numerous numerical problems are analyzed using the proposed family along with comparisons with some of the previous techniques to highlight our theoretical findings. 7.2 An optimal iterative family and its convergence anal- ysis We consider an optimal family of iterative methods of order four for multiple roots having known multiplicity m ≥ 1 as follows: yn = xn − m f (xn) , f ′(xn) + tf (xn) μn f (xn) (7.2.1) xn+1 = yn − m 1 − 2αμn) (f ′(xn) + tf (xn)) (G(μn) − t f(xn) f′(xn) + tf(xn)) , (where α, t ∈ R are free parameters, the weight function G : C → C is an analytic function in 1 the vicinity of the origin with μn = f (yn) m , a multi-valued function for which its principal (f (xn)) analytic branch has been considered (see Ahlfors et al. (1966)). Therefore, the principal root of μn is examined which is given as μn = exp m1log ff((xynn)) , where log ff((xynn)) = log ff((xynn)) + i · arg ff((xynn)) with −π < arg ff((yxnn)) ≤ π. [( )] ( ) | In (the su)bsequent Theorem(7.2.)1, we prove that the proposed family (7.2.1) reaches the | | optimal convergence rate for all α, t ∈ R. 7.2.1 Convergence analysis Theorem 7.2.1. Consider a function f : D ⊆ C → C defined on a ball D containing the desired multiple zero κ ∈ R having multiplicity m. Then, the sequence {xn}n≥0 obtained from family (7.2.1) converges to κ with convergence order at least four giving the error relation, en+1 = −1 ((t + d1)(t2(−2 + G′′(0) + 4α) + 2t(−5 + G′′(0) + 6α)d1 + (−9 − m + G′′(0) 2m3 + 8α)d21 + 2md2))e4n + O(e5n), (7.2.2) where α, t ∈ R, G is a weight function verifying G(0) = 1, G′(0) = 2(1 − α), and dk = (mm+!k)! f (fm(m+)k()κ(κ)) , k = 1, 2, 3, ... Proof. Let us consider κ to be a multiple zero having multiplicity m ≥ 1. Now, by Taylor’s series expansion, f (xn) and f ′(xn) can be expanded about x = κ, obtaining f(xn) =f(mm)(!κ)emn(1 + d1en + d2e2n + d3e3n + d4e4n) + O(e5n), (7.2.3) and f′(xn) =f(m−1)(κ)emn−1 m+1 (m − 1)! 1+ (m d1en + m+2 m d2e2n + m + 3d3e3n + m + 4d4e4n + O(e5n), m m) (7.2.4) respectively. Upon using the expressions (7.2.3) and (7.2.4) in (7.2.1), we obtain eyn = yn − κ = (t + d1)e2n + m2 (t2 + 2td1 + (1 + m)d21 − 2md2)e3n + m3 (t3 + (3 + 2m)td21 1 1 1 m + (1 + m)2d31 − 4mtd2 + d1(3t2 − m(4 + 3m)d2) + 3m2d3)e4n + O(e5n). (7.2.5) From the Taylor’s series expansion of f (yn) and expression (7.2.5), we get f(yn) =f(mm)(!κ)emyn 1 + d1eyn + d2e2yn + d3e3yn + d4e4yn + O(e5yn) . ( ) From expressions (7.2.3) and (7.2.6), we obtain μn = (ff ((xynn))) m (t + d1)en − m2 (t2 + 3td1 + (2 + m)d21 − 2md2)e2n + 1 = 1 1 1 m 2m3 (5(3 + m)td21 + (7 + 7m + 2m2)d31 + 2d1(5t2 − m(7 + 3m)d2) + 2(t3 − 5mtd2 + 3m2d3))e3n + O(e4n). (7.2.6) (7.2.7) From (7.2.7), the order of μn is linear, i.e., μn = O(en). So, we can expand G(μ) by employing expansion by Taylor in the vicinity of origin as G(μn) =G(0) + μnG′(0) + μ2nG′′(0) + μ3nG′′′(0) + O(e4n). 1 1 (7.2.8) 2! 3! By using expressions (7.2.3)–(7.2.5), (7.2.7), (7.2.8) in the scheme (7.2.1), we obtain en+1 = 1 (−1 + G(0))(t + d1)e2n − t2(G′(0) + 2G(0)(−1 + α)) + t(1 + 2G′(0) 1 m m2 + G(0)(−5 + 4α))d1 + (1 + m (− 3G(0) − mG(0) + G′(0) + 2G(0)α)d21 + 2m(−1 + G(0))d2 e3n − 2m3 1 t3(4 + G′′(0) − 4α + G′(0)(−6 + 4α) + 2G(0)(3 − 6α + 4α2)) + t2((8(+ 3G′ ′(0) − 8α + 2G′(0)(−11 + 6α) + 4G(0)(6 − 11α + 6α2))d1) + t((2 − 26G′(0) + 3G′′(0) − 4α + 12αG′(0) (7.2.9) + G(0)(31 − 52α + 24α2) − m(2 + 4G′(0) + G(0)(−9 + 8α)))d21 + 2m(2 + 4G′(0) + G(0)(−9 + 8α))d2) + (−2 + 2m2(−1 + G(0)) − 10G′(0) + G′′(0) + 4G′(0)α − m(4 − 11G(0) + 4G′(0) + 8G(0)α) + G(0)(13 − 20α + 8α2))d31 + 2m(4 − 3m(−1 + G(0)) − 11G(0) + 4G′(0) + 8αG(0))d1d2 + 6m2(−1 + G(0))d3 e4n + O(e5n). Taking G(0) = 1, (7).2.9) becomes en+1 = − (−2 + G′(0) + 2α)(t + d1)2e3n − 1 1 2m3 t3(10 + G′′(0) − 16α + 8α2 m2 + G′(0)(−6 + 4α)) + t2((32 + 3G′′(0) − 52(α + 24α2 + 2G′(0)(−11 + 6α))d1) + t((33 − 26G′(0) + 3G′′(0) + m(7 − 4G′(0) − 8α) − 56α + 12G′(0)α + 24α2)d21 + 2m(−7 + 4G′(0) + 8α)d2) + (11 − 10G′(0) + G′′(0) + m(7 − 4G′(0) − 8α) − 20α + 4G′(0)α + 8α2)d31 + 2m(−7 + 4G′(0) + 8α)d1d2 e4n + O(e5n).) (7.2.10) Further, taking G′(0) = 2(1 − α) in (7.2.10), the final error equation becomes en+1 = −1 ((t + d1)(t2(−2 + G′′(0) + 4α) + 2t(−5 + G′′(0) + 6α)d1 + (−9 − m + G′′(0) 2m3 + 8α)d21 + 2md2))e4n + O(e5n), (7.2.11) which shows at least fourth-order convergence of (7.2.1) for m ≥ 1. Remark 7.2.1. There are very few multiple root-finding methods of order four that work when m = 1. Particularly, the schemes by Shengguo et al. (2009) and Zafar et al. (2020) work only when m ≥ 2 and when m = 1, they fail. We will show that we have developed a scheme in order to solve problems requiring multiple solutions working for m ≥ 1 instead of m ≥ 2. 7.2.2 Some selective weight functions Many methods of the family (7.2.1) can be formed for distinct choices of the weight function G which fulfill the conditions indicated in Theorem 7.2.1. Here, some of them are listed: 1. 1 + (2 − 2α)μ 2. eμ2 + (2 − 2α)μ 3. p(1 + (2 − 2α)μ) p + qμ2 , where p, q ∈ R 4. 1 + 2(2 − α)μ 1 + 2μ + cμ2 , where c ∈ R. 7.3 Numerical results This section lays out the comparison of our family (7.2.1) with several existing schemes. Numerical results have been computed by taking t = 0.5 and α = 1 for our methods. We have made a careful selection of challenging problems, for which many numerical methods encounter difficulties. First, we examined the Oscillation problem, where certain methods showed instability and fluctuating solutions. Next, we focused on the Failure problem, where some methods failed to converge or produced inaccurate results. We also investigated the issue of convergence to undesired roots, noting that certain methods struggled to reach the desired root and instead converged to incorrect solutions. Another challenge we encountered was the so-called 6 − 12 potential problem, where certain methods exhibited slow convergence or even divergence. Additionally, we explored the Channel flow problem, which required specialized methods to handle complex flow dynamics. The Clustered roots problem posed difficulties due to multiple roots in close proximity, requiring careful handling to accurately locate each root. Finally, we addressed the challenge of High-order multiplicity, where some methods struggled to handle roots with high multiplicities, leading to inaccuracies in the computed solutions. Many of the previous optimal Newton-like schemes locating the multiple roots having order four are taken into consideration which are described as follows: 1. Li et al. method (LM1) Shengguo et al. (2009): yn = xn − m + 2 f ′(xn) 2m f (xn) , xn+1 = xn − b1 f ′(xn) b2f ′(xn) + b3f ′(yn) f (xn) − f (xn) , where b1 = m − , b2 = − , and b3 = m2 1 1 2 m m m m. m+2 2. Li et al. method (LM2) Li et al. (2010): ( ) yn = xn − m2m+ 2 ff′((xxnn)) , xn+1 = xn − b1 f ′(yn) b2f ′(xn) + b3f ′(yn) f (xn) − f (xn) , (7.3.1) (7.3.2) where b1 = − 1 mm+2 mm(m4 + 4m3 − 16m − 16) 2 , ( ) m3 − 4m + 8 b2 = − (m3 − 4m + 8)2 m(m4 + 4m3 − 4m2 − 16m + 16)(m2 + 2m − 4) , b3 = m2(m3 − 4m + 8) m+2 (m4 + 4m3 − 4m2 − 16m + 16)(m2 + 2m − 4) m m . ( ) 3. Soleymani and Babajee method (SB1) Soleymani and Babajee (2013): yn = xn − m2+m 2 ff′ ((xxnn)) , xn+1 = xn + 4mpmf (xn)S1 (m2 + 2m − 4)pmf ′(xn) − m2f ′(yn) , where p = m and m + 2 S1 = − (m − 2)m3p−2m f′(yn) − pm−1 1 2 − 1000 f′(yn) − pm−1 + 1. 3 16 (f′(xn)) (f′(xn)) 4. Soleymani and Babajee method (SB2) Soleymani and Babajee (2013): yn = xn − m + 2 f ′(xn) 2m f (xn) , 2 4mpmf (xn) 1 − ((m − 2)m3 f ′(yn) − pm−1 xn+1 = xn + 16p2m (f ′(xn))) (m2 + 2m − 4)pmf ′(xn) − m2f ′(yn) , where p = m . m + 2 5. Sharma et al. method (SM1) Sharma et al. (2019): yn = xn − m f [vn, xn] f (xn) , xn+1 = yn − sn − wn + mwn − m2snwn + 2msnwn f (xn) , (−msn + s2n + 1) f [vn, xn] f (yn) m 1 f (yn) m 1 where vn = xn + βf(xn), sn = (f(xn)) , wn = (f(vn)) and β = 0.5. 132 (7.3.3) (7.3.4) (7.3.5) 6. Sharma and Sharma method (SM2) Sharma and Sharma (2010): yn = xn − m + 2 f ′(xn) 2m f (xn) , xn+1 = xn − (m3 − 4m + 8) − (m + 2)2 m m m f ′(xn) 8 (7.3.6) (m f ′(xn) f (xn) (m + 2 m) f ′(yn) (2(m − 1) − (m + 2) (m + 2) f ′(yn))) f ′(xn) . 7. Sharifi et al. method (SM3) Sharifi et al. (2012): yn = xn − m + 2 f ′(xn) 2m f (xn) , xn+1 = xn + 1 m(m2 + 2m − 4) f (xn) − 1 m(m + 2)2pm f (xn) (4 f ′(xn) 4 f ′(yn) (7.3.7) f′(yn) − pm−1 2 × 1 + m4 + 1 f(xn) 3 , 8(m + 2)p2m f′(xn) m (( ) 81 f′(yn))) (where p = . m + 2 8. Zafar et al. method (ZM1) Zafar et al. (2020): yn = xn − mf′(xn) f(xn) , xn+1 = xn − m + mvn + m2m−21 vn2 + 32.6vn3 ff′((xxnn)) , (7.3.8) ( ) f′(yn) m−1 1 where vn = f′(xn)) . (9. Zhou et al. method (ZM2) Zhou et al. (2011): yn = xn − m + 2 f ′(xn) 2m f (xn) , xn+1 = xn − m3 m + 2 2m f ′(yn) 2 − 2m2(m + 3) m + 2 m f ′(yn) m 8 (7.3.9) ((m) (f ′(xn)) (m) f ′(xn) + (m3 + 6m2 + 8m + 8) f (xn) f ′(xn) .) 10. Zafar et al. method (ZM3) Zafar et al. (2020): yn = xn − (m + 50f (xn)m) f (xn) f ′(xn) , f (xn) (7.3.10) xn+1 = xn − m + mvn + (m2m−21 vn2 + 32.6vn3) f ′(xn) , f′(yn) m−1 1 where vn = f′(xn)) . (We display in Table 7.1 our proposed methods which have been named as P M1, P M2 or P M3 according to the weight function used. Table 7.1: Our proposed methods Weight function G(μ) Parameter values Method 1 + (2 − 2α)μ p(1 + (2 − 2α)μ) p + qμ2 1 + 2(2 − α)μ 1 + 2μ + cμ2 − p=1,q =1 c=1 P M1 P M2 P M3 In Tables 7.2−7.9, we have displayed errors in the consecutive approximations |xn+1 − xn| and the computational order of convergence (see Jay (2001)) (denoted by ρ). We have also included CPU time for all the methods. We have selected the following numerical examples: Example 7.3.1. Oscillatory approximations: Oscillation in iterative methods refers to a phenomenon where the computed solutions alternate between different values without con- verging to a stable result. It can occur due to poor initial guesses, ill-conditioned systems, or inappropriate iteration parameters. We take into consideration the following problem: f1(x) = sin x = 0. Although, the said equation acquires infinite roots in number having multiplicity one, our desired zero being κ = −π. After comparing the new methods with existing ones taking initial guess as x0 = −1.6, we display the numerical results in Table 7.2. In many cases, this test problem gives an oscillatory conduct. Furthermore, the variants of Newton's method on the said problem do not work, in general, even when the initial point is very near to the desired solution. But our methods do not exhibit this kind of behavior. We note that the methods LM1, LM2, SB1, SB2, SM1, SM2, SM3 and ZM2 converge to the undesired roots −18.8496, 0, 3.4872 × 107, 728.8495, −9.4248, −21.9911, −2070.3096 and −28695.3073, respectively. Also, the methods ZM1 and ZM3 diverge. Example 7.3.2. Failure problem: When the first derivative is zero at any stage of an iterative process, usually a failure problem arises. This condition leads to a flat or plateau- like region, preventing further progress towards the solution. Then, some special techniques like our methods are needed to address this challenge and resume convergence. The following problem is therefore taken in this context: 1 3 f2(x) = xe−x − = 0. (10) Table 7.2: Numerical outcomes for Example 7.3.1 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f1 (x) LM1 − LM2 − − SB1 − − SB2 − − SM1 − − SM2 − − SM3 − − ZM1 − − ZM2 − − ZM3 − − PM1 3.3 × 10−1 1.2 × 10−3 PM2 3.7 × 10−1 5.3 × 10−3 PM3 4.8 × 10−1 5.0 × 10−2 − − − − − − − − − − − 1.0 × 10−13 6.5 × 10−11 4.2 × 10−7 UR UR UR UR UR UR UR D UR D 4.1852 4.3230 5.5106 − − − − − − − − − − − 0.003 0.004 0.003 UR−Convergence to undesired root D−Divergent The said equation has roots, finite in number having multiplicity three each, but the root which we have considered is κ = 0.11183255915896296483356945682026584. It can be seen in Table 7.3 that for the initial guess x0 = 1, all the previous methods mentioned, fail to converge to the desired solution. The reason being that the function has its first order derivative equal to zero at the required root, but to guarantee convergence of Newton's like methods, the derivative must be non zero in a vicinity of the desired solution. Nevertheless, after a finite number of iterations, our methods converge to the desired solution. Example 7.3.3. Divergence or convergence to undesired roots: Divergence or con- vergence to an undesired root is a significant issue in iterative methods. It occurs when the iterations fail to converge towards the desired solution and instead approach a different unde- sired root. This may be due to poor initial guesses, ill-conditioned problems, or inappropriate convergence criteria. However, such behavior is not displayed by our methods. The following Van der Waals equation (see Chapra and Canale (2015)) of the ideal gas is the subject of the third illustration, P1 + a1n2 (V1 − nb1) = nR1T1, (V1) which details the behavior of a real gas under certain conditions, where a1 and b1 are two known constants. For specific values, the volume V1 can be computed upon finding the solution for the nonlinear equation given as f3(x) = x3 − 5.22x2 + 9.082x − 5.2675 = 0. Table 7.3: Numerical outcomes for Example 7.3.2 Methods |x8 − x7| |x9 − x8| |x10 − x9| ρ CPU Time f2 (x) LM1 – LM2 – – SB1 – – SB2 – – SM1 – – SM2 – – SM3 – – ZM1 – – ZM2 – – ZM3 – – PM1 7.9 × 10−34 2.6 × 10−133 PM2 1.3 × 10−3 4.1 × 10−12 PM3 6.2 × 10−4 2.0 × 10−13 – – – – – – – – – – 2.9 × 10−531 3.8 × 10−46 2.4 × 10−51 F F F F F F F F F F 4.0000 4.0000 4.0000 – – – – – – – – – – 0.016 0.027 0.036 F–Method fails to converge to the desired root The concerned multiple zero taken is κ = 1.75 with m = 2. Table 7.4 includes the corre- sponding data taking initial guess as x0 = 1.735. We can see that our methods PM1, PM2 and PM3 converge to the desired root with far fewer errors than other existing methods. Fur- thermore, the methods LM1 and SB2 both converge to the undesired root 1.72. On the other hand, the methods LM2, SM2, SM3, ZM1 and ZM3 diverge, while the methods SB1, SM1 and ZM2 show a very slow convergence to the desired root. Example 7.3.4. Potential problem: In the fourth example, the problem of Lenard-Jones potential, known as the 6 − 12 potential has been outlooked (see Özyapıcı et al. (2014)). The said problem examines the interaction in between two neutral atoms or molecules in atomic physics and physical chemistry described through the function, s 12 s 6 V (x) = V0 x − x , (( ) ( )) where x is the distance in between both particles (measured from the center of one to that of another particle), s is the position where the particle interaction becomes zero and V0 repre- sents depth of the potential. We are concerned only about the minimum distance, which is determined after solving the equation V ′ (x) = 0 because the Lenard-Jones potential reflects the potential energy of contact between two nonbonding atoms or molecules dependent on their distance of detachment. This means getting the root of the nonlinear equation: f4(x) = −12x13 + 6x7 = 0. s12 s6 The exact minimum value κ = 2 76 with multiplicity 1 is obtained for s = 2. Taking initial guess x0 = 1.9, the obtained results are displayed in Table 7.5. It can be seen that under these Table 7.4: Numerical outcomes for Example 7.3.3 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f1 (x) LM1 − − LM2 − − SB1 3.8 × 100 5.4 × 10−3 SB2 − − SM1 1.0 × 10−1 9.1 × 10−3 SM2 − − SM3 − − ZM1 − − ZM2 1.7 × 10−1 2.2 × 10−2 ZM3 − − PM1 8.6 × 10−3 2.6 × 10−3 PM2 9.99 × 10−3 3.8 × 10−5 PM3 2.6 × 10−3 3.5 × 10−7 − − 2.3 × 10−5 − 3.1 × 10−5 − − − 5.3 × 10−4 − 2.4 × 10−7 2.5 × 10−14 1.9 × 10−22 UR D 0.6259 UR 1.8508 D D D 1.3913 D 3.9540 3.7074 3.9419 − − 0.001 − 0.001 − − − 0.001 − 0.002 0.002 0.001 UR–Convergence to undesired root D–Divergent conditions, the methods SB1, SM1, ZM1 and ZM3 diverge. Also, the method ZM2 displays a very slow convergence to the desired root. Table 7.5: Numerical outcomes for Example 7.3.4 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f4 (x) LM1 9.5 × 10−2 LM2 1.2 × 10−1 S B1 − S B2 1.3 × 10−1 S M1 − S M2 1.1 × 10−1 S M3 1.2 × 10−1 Z M1 − Z M2 1.3 × 10−1 Z M3 − P M1 9.7 × 10−2 P M2 1.1 × 10−1 P M3 1.0 × 10−1 2.4 × 10−3 2.0 × 10−2 − 2.1 × 10−2 − 8.0 × 10−3 1.6 × 10−2 − 3.3 × 10−2 − 2.6 × 10−3 5.4 × 10−3 4.3 × 10−3 1.5 × 10−9 3.3 × 10−5 − 3.5 × 10−5 − 3.6 × 10−7 1.2 × 10−5 − 2.9 × 10−4 − 2.3 × 10−9 9.4 × 10−8 4.0 × 10−8 3.4210 2.4911 D 2.5209 D 3.0602 2 .6616 D 2.1862 D 3.4019 3.1031 3.1425 0.002 0.001 − 0.002 − 0.001 0.002 − 0.002 − 0.002 0.002 0.002 D–Divergent Example 7.3.5. Chemical reactor problem: Here, we consider a problem of fractional conversion in a chemical reactor, described by the following equation (see Shacham (1986)): f5(x) = 1 − x − 5 log 0.4 − 0.5x + 4.45977 = 0. x 0.4(1 − x) Here, fractional conversion of quantities in a chemical reactor is denoted by x. The above fractional conversion will be of no physical meaning if x is less than zero or greater than one. Thus, x must be from the interval [0,1]. The desired root is κ ≈ 0.7573962462537538. After comparing the new methods with existing ones taking initial guess x0 = 0.73, the numerical outcomes are displayed in Table 7.6. Note that the methods SB1, SM1, ZM1 and ZM3 diverge in this case too. Table 7.6: Numerical outcomes for Example 7.3.5 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f5 (x) 1.7 × 10−4 2.6 × 10−2 − 6.9 × 10−3 − 8.5 × 10−4 5.7 × 10−3 − 2.0 × 10−2 − 1.3 × 10−4 7.9 × 10−4 2.6 × 10−4 2.5 × 10−13 7.6 × 10−3 − 1.8 × 10−5 − 1.1 × 10−9 6.8 × 10−6 − 3.1 × 10−3 − 1.1 × 10−13 2.1 × 10−14 3.4 × 10−13 1.2 × 10−48 2.9 × 10−5 − 9.0 × 10−16 − 2.9 × 10−33 1.4 × 10−17 − 1.4 × 10−6 − 5.9 × 10−50 2.5 × 10−52 6.2 × 10−50 3.9995 11.2100 D 3.9121 D 3.9968 3.9370 D 3.1577 D 4.0002 3.9995 4.0002 0.003 0.003 − 0.003 − 0.004 0.005 − 0.004 − 0.004 0.005 0.009 D–Divergent Example 7.3.6. Channel flow problem: In this example, we consider a problem of open channel flow, which involves finding the depth of water in a rectangular channel which is represented by the nonlinear equation: (see Rehman et al. (2021)) √ sbx bx 2 3 f(x)= −F =0, √ n (b+2x) F being the water flow given by F = sbx 2 r3, n is the Manning's roughness coefficient, s is the slope, r is the hydraulic radius and b is the width of the channel. The following equation n is obtained if the parameter values are as F = 14.15 m3/s, b = 4.572 m, s = 0.017 and n = 0.0015: f6(x) = 0.0015 (4.572 + 2x) 0.5961x 4.572x 2 3 − 14.15 = 0. The desired solution being κ = 0.13839748098511792. For the initial guess x0 = 0.4, the numerical outcomes are displayed in

Table 7.7. In this case, the methods SB1, SM1, ZM1 and ZM3 diverge. Table 7.7: Numerical outcomes for Example 7.3.6 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f6 (x) 2.9 × 10−2 3.7 × 10−2 – 3.6 × 10−2 – 3.1 × 10−2 3.4 × 10−2 – 4.1 × 10−2 – 1.9 × 10−2 3.6 × 10−2 3.1 × 10−2 8.5 × 10−6 4.1 × 10−5 – 3.5 × 10−5 – 1.4 × 10−5 2.5 × 10−5 – 7.5 × 10−5 – 1.7 × 10−6 3.4 × 10−5 1.7 × 10−5 8.9 × 10−20 1.2 × 10−16 – 5.7 × 10−17 – 7.2 × 10−19 1.2 × 10−17 – 1.8 × 10−15 – 1.5 × 10−22 4.8 × 10−17 3.3 × 10−18 3.9229 3.8567 D 3.8709 D 3.9101 3.8859 D 3.8281 D 3.9545 3.8704 3.8891 0.005 0.011 – 0.014 – 0.009 0.012 – 0.005 – 0.009 0.006 0.007 D–Divergent Example 7.3.7. Clustered roots problem: In this example, a problem with clustered roots is considered, which was previously examined by Zeng (2005), given as f7(x) = (x − 1)20(x − 2)15(x − 3)10(x − 4)5 = 0. f7 has four zeros x = 1, 2, 3 and 4 having multiplicities twenty, fifteen, ten and five, respectively. However, the desired root taken here is x = 1, with multiplicity 20. Table 7.8 details the numerical outcomes of the methods with initial guess x0 = 0.8. We note that the methods Z M1 and Z M3 diverge in this case too, and the proposed methods provide the lower errors. Example 7.3.8. High-order multiplicity: In the final example, a classic academic problem is taken, given by the equation: f8(x) = (x − 1)3 − 1 100 = 0. f8 has a root x = 2 having multiplicit(y 100. Table)7.9 details the obtained computed results with x0 = 2.1. The proposed methods P M1, P M2 and P M3 converge to the desired solution with error as minimum when compared to the previous methods. In this example, we have again that the methods ZM1 and ZM3 diverge. Table 7.8: Numerical outcomes for Example 7.3.7 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f7 (x) 2.6 × 10−3 2.6 × 10−3 2.5 × 10−3 2.5 × 10−3 1.2 × 10−2 2.6 × 10−3 2.5 × 10−3 – 2.6 × 10−3 – 9.0 × 10−4 1.4 × 10−3 1.3 × 10−3 2.2 × 10−10 2.2 × 10−10 1.8 × 10−10 1.8 × 10−10 1.3 × 10−6 2.2 × 10−10 1.9 × 10−10 – 2.2 × 10−10 – 7.4 × 10−13 8.6 × 10−12 6.4 × 10−12 1.1 × 10−38 1.1 × 10−38 5.6 × 10−39 5.6 × 10−39 3.4 × 10−12 1.1 × 10−38 6.5 × 10−39 – 1.2 × 10−38 – 3.3 × 10−49 1.2 × 10−44 3.8 × 10−45 3.9982 3.9982 3.9983 3.9983 1.3914 3.9982 3.9983 D 3.9982 D 3.9997 3.9992 3.9992 0.007 0.004 0.008 0.006 0.003 0.012 0.007 – 0.011 – 0.006 0.004 0.005 D–Divergent Table 7.9: Numerical outcomes for Example 7.3.8 Methods |x2 − x1| |x3 − x2| |x4 − x3| ρ CPU Time f8 (x) 2.1 × 10−4 2.1 × 10−4 2.1 × 10−4 2.1 × 10−4 1.6 × 10−3 2.1 × 10−4 2.1 × 10−4 – 2.1 × 10−4 – 5.3 × 10−5 1.1 × 10−4 1.0 × 10−4 6.1 × 10−15 6.1 × 10−15 5.9 × 10−15 5.9 × 10−15 1.6 × 10−9 6.1 × 10−15 5.9 × 10−15 – 6.1 × 10−15 – 5.2 × 10−18 2.5 × 10−16 1.8 × 10−16 4.5 × 10−57 4.5 × 10−57 3.8 × 10−57 3.8 × 10−57 5.2 × 10−18 4.5 × 10−57 3.9 × 10−57 – 4.5 × 10−57 – 4.7 × 10−70 6.3 × 10−63 1.9 × 10−63 3.9999 3.9999 3.9999 3.9999 1.4141 3.9999 3.9999 D 3.9999 D 4.0000 4.0000 4.0000 0.009 0.002 0.001 0.001 0.031 0.002 0.002 – 0.002 – 0.002 0.002 0.001 D–Divergent Remark 7.3.1. The numerical outcomes in Tables 7.2–7.9 clarify that our methods P M1, P M2 and P M3 perform better compared to other existing methods, since they provide lower ab- solute errors. Furthermore, many of the considered methods for comparison failed to converge to the root when f ′(κ) = 0. Remark 7.3.2. In all the eight problems, our methods show consistent behavior, whereas the existing methods do not exhibit this behavior. The existing methods either fail to converge, converge to undesired root, diverge or show slow convergence to the root. Therefore, our methods can be considered superior in this regard as they give results in many of the cases where the existing methods do not work. 7.4 Conclusions A new family of methods iterative in nature, has been developed, exhibiting an exceptional performance with optimal fourth-order convergence for multiple roots. Notably, they demon- strate a remarkable convergence even in scenarios where the derivative of the function is zero or extremely small around the desired zero. Additionally, our methods successfully overcome various challenges that commonly arise in root-finding algorithms. By overcoming challenges such as oscillation, failure, and convergence to undesired roots, our methods provide robust and reliable solutions. Furthermore, our methods have proven their versatility and applica- bility in diverse problem domains, including the 6 − 12 potential problem, a channel flow problem, a clustered roots problem, and high-order multiplicity problems. Thus, our methods are highly valuable tools for researchers and practitioners in fields where an accurate root determination is crucial. The robust convergence properties, coupled with their versatility and ability to handle complex scenarios, position our methods as reliable and efficient options for root-finding tasks across diverse domains. Chapter 8 An Iterative Family for System of Nonlinear Equations This chapter deals with introducing an extension to a family for scalar equations to solve system of nonlinear equations. To validate the effectiveness of the proposed meth- ods, multiple numerical examples are provided, demonstrating their convergence properties and efficiency in comparison to well-established iterative techniques. The results illustrate that the new methods outperform traditional approaches in terms of iteration count, accu- racy, and computational complexity. In particular, the numerical experiments highlight how these iterative schemes successfully navigate issues such as slow convergence and sensitivity to initial conditions, which are common challenges in solving nonlinear systems. The current approaches to solving the system of nonlinear equations can be seen as being expanded upon and generalized by these new iterative techniques. 8.1 Introduction The resolution of equations is a long-established topic within the realms of science and en- gineering, holding significant relevance in various applications. Given this context, a vast array of iterative methods has been developed for addressing scalar nonlinear equations. However, it is important to recognize that many of these methods are not applicable to their correspond- ing systems. Even when such extensions are feasible, several critical factors must be taken into account. Consequently, there exists a limited number of practical iterative methods in this domain. Furthermore, it is noteworthy that while some scalar iterations can be adapted, the resulting increase in computational complexity renders them of little practical value. This issue has been extensively examined in the literature. Consider a nonlinear function F : D ⊆ Rn → Rn having at

least, second-order Fréchet derivatives and continuity on D. We determine a vector κ = (κ1, κ2, . . . , κn) for F such that, F (κ) = 0. The contents of this chapter are communicated. As discussed earlier, a fundamental approach for addressing the system of nonlinear equa- tions F (x) = 0 is NM given in (1.5.26). However, it requires the computation of Jacobian matrices, which can be computationally expensive, motivating the development of derivative- free or quasi-Newton approaches. Various one-point methods, including Chebyshev and Halley, have been adapted to their respective system versions, achieving a convergence order of three. However, these methods rely on the first and second Fréchet derivatives, necessitating n2 and n3 functional evaluations, respectively. Conversely, significant efforts have been made to develop methods that do not depend on the second Fréchet derivative while still attaining a convergence order of three, although these methods are not classified as one-step methods. More recently, multipoint and hybrid methods have gained attention for their ability to accelerate convergence while reducing computational cost. Inspired by the research in this area, this chapter introduces an extension of a family of iterative methods initially designed for scalar equations, adapting them to solve systems of nonlinear equations. The newly proposed methods generalize existing approaches by incor- porating modifications that improve convergence behavior and computational efficiency. To assess their effectiveness, multiple numerical experiments are conducted, demonstrating their superior performance in comparison to conventional methods. 8.2 An iterative method and its convergence analysis We consider the following iterative scheme to find roots of system of nonlinear equations: y(k) = x(k) − F (x(k)) F (x(k)), −1 x(k+1) = x(k) − [y(k), x()k); F] (−1 (8.2.1) F (x(k)), ( ) where [y(k), x(k); F] is a finite difference. Now, we will prove that our scheme (8.2.1) attains third order of convergence. In order to clarify its proof, we recall some notions and results introduced in Cordero et al. (2010a). q ≥ 1, is the q-linear function F (q)(u) : Rn × . . . × Rn → Rn such that F (q)(u)(v1, . . . , vq) ∈ Rn. Let F : D ⊆ Rn → Rn be sufficiently differentiable in D. The qth derivative of F at u ∈ Rn, It is easy to observe that 1. F (q)(u)(v1, . . . , vq−1, ·) ∈ L(Rn) 2. F (q)(u)(vσ(1), . . . , vσ(q)) = F (q)(u)(v1, . . . , vq), for all permutation σ of {1, 2, . . . , q}. From the above properties, we can use the following notations: (a) F (q)(u)(v1, . . . , vq) = F (q)(u)v1 . . . vq (b) F (q)(u)vq−1F (p)vp = F (q)(u)F (p)(u)vq+p−1 On the other hand, for κ + h ∈ Rn lying in a neighborhood of a solution κ of F (x) = 0, we can apply Taylor's expansion and assuming that the Jacobian matrix F ′(κ) is nonsingular, we have p−1 F (κ + h) = F ′(κ) [h + Dqhq] + O(hp), (8.2.2) Σq=2 where Dq = [F ′(κ)]−1F (q)(κ), q ≥ 2. We observe that Dqhq ∈ Rn since F (q)(κ) ∈ L(Rn × 1 . . . × Rn, Rn) and [F ′(x¯)]−1 ∈ L(Rn). In addition, we can express F ′ as q! p−2 F ′(κ + h) = F ′(x¯) [I + qDqhq−1] + O(hp−1), (8.2.3) Σq=2 where I is the identity matrix. Therefore, qDqhq−1 ∈ L(Rn). From (8.2.3), we obtain [F ′(κ + h)]−1 = I + X2h + X3h2 + X4h4 + . . . [F ′(κ)]−1 + O(hp), (8.2.4) where [ ] X2 = −2D2, X3 = 4D22 − 3D3, X4 = −8D23 + 6D2D3 + 6D3D2 − 4D4, . We denote ek = x(k) − κ, the error in the kth iteration in the multidimensional case. The equation ek+1 = M epk + O(epk+1), where M is a p-linear function, M ∈ L(Rn × . . . × Rn, Rn), is called the error equation and p is the order of convergence. Observe that ekp is (ek, ek, . . . , ek). 8.2.1 Convergence analysis Theorem 8.2.1. Let F : D ⊆ Rn → Rn be a sufficiently differentiable function in an open neighborhood D of its zero κ. Let us suppose that F ′(x) is continuous and nonsingular in κ and the initial guess x(0) is close enough to κ. Then, the iterative scheme defined by ((8.2.1)) attains maximum third-order of convergence. Proof. Let ek = x(k) − κ be the error of the kth iteration. Developing F (x(k)) and F ′(x(k)) in a neighborhood of κ, we write F (x(k)) = F ′(κ) ek + D2e2k + D3e3k + O(e4k) , (8.2.5) and [ ] F ′(x(k)) = F ′(κ) I + 2D2ek + 3D3e2k + O(e3k) , (8.2.6) [ ] where I is the identity matrix of size n × n and Dk = F ′(κ)−1F (k)(κ), k ≥ 2. Inversion of 1 F ′(x(k)) yields k! F ′(x(k))−1 = I + X1ek + X2e2k F ′(κ)−1, (8.2.7) where X1 = −2D2 and X2 = 4D22 − 3D[3. By replacing t]hese expressions in the first step of (8.2.1), we obtain y(k) − κ = D2e2k − 2(D22 − D3)e3k + O(e4k). (8.2.8) Therefore, we have F (y(k)) = F ′(κ) D2e2k − 2(D22 − D3)e3k + O(e3k) (8.2.9) and [ ] [y(k), x(k); F] = F ′(κ) I + D2ek + (D22 + D3)e2k + O(e3k) . (8.2.10) Inversion of [y(k), x(k); F] yields [ ] [y(k), x(k); F]−1 = I − D2ek − D3e2k + O(e3k) F ′(κ)−1. (8.2.11) Adopting the expressions (8.2.5) and (8.2.11), we obtain [ ] [y(k), x(k); F]−1F ′(x(k)) = ek − D22e3k + O(e4k). (8.2.12) Finally, using the expression (8.2.12) in the second substep of (8.2.1), we get x(k+1) − κ = ek − [y(k), x(k); F]−1F ′(x(k)) = D22e3k + O(e4k). (8.2.13) Hence, the iterative method (8.2.1) has third-order convergence. 8.3 Numerical results Based on the theoretical results, we performed a computational analysis to demonstrate their practical significance. Therefore, we choose some third-order existing schemes for comparison: 1. Homeier method (HM) Homeier (2004): y(k) =x(k) − 1F′(x(k))−1F(x(k)), 2 (8.3.1) x(k+1) =x(k) − F′(y(k))−1F(x(k)). 2. Frontini and Sormani method (FSM) Frontini and Sormani (2003); Cordero et al. (2012): y(k) =x(k) − F ′(x(k))−1F (x(k)), x(k+1) =x(k) − 2(F ′(y(k)) + F ′(x(k)))−1F (x(k)). (8.3.2) 3. Modified Newton method (M N M) Frontini and Sormani (2004): m −1 Σ ( ) We have chosen m = 2, A1 = A2 = τ1 = τ2 = . 1 2 x(k+1) =x(k) − AiF ′ x(k) − τiF ′(x(k))−1F (x(k))) F (x(k)). (8.3.3) (i=1 4. Newton Simpson method (N SM) Cordero and Torregrosa (2007): y(k) =x(k) − F ′(x(k))−1F (x(k)), x(k+1) =x(k) − 6F (x(k)) F ′(x(k)) + 4F ′ y(k) + x(k) (2 . + F ′(y(k))) (8.3.4) 5. Chebyshev-like method (CLM) Babajee et al. (2010): Xk+1 =XkN+1 − JF−1(Xk)F (XkN+1). (8.3.5) We choose five problems for our computational examinations. In Example (8.3.1), we selected an academic problem that contains a 4 × 4 system of nonlinear equations, and the obtained results are depicted in Table 8.1. In Example 8.3.2, we investigate an applied science problem, namely the Hammerstein integral equation, to demonstrate the applicability and efficacy of our method (8.2.1). The values of the abscissas tj and weights wj are depicted in Table 8.2. The numerical results are presented in Table 8.3 for Example 8.3.2. Table 8.4 provides the numerical results for the boundary value problem, which is given in Example (8.3.3). In the BVP, we choose a larger system of nonlinear equations of order 100 × 100. In Example 8.3.4, we investigate the applicability of our method on another applied science problem, namely the Burger equation, and for this problem, we also used a larger system of nonlinear equations of order 100 × 100. In the last example, we selected another 95 × 95 system of nonlinear equations, which is a mixture of algebraic and trigonometric functions (further information is provided in Example (8.3.5)), with numerical results displayed in Table 8.6. Further, we also compared our method with existing methods based on the number of iterations, and the results are mentioned in Table 8.7, and in Table 8.8, the results based on the residual error at fifth step are mentioned. Additionally, we provide the computational order of convergence (COC), which has been calculated using the following formulae: η = ln ‖x|xk+k−1−x∗x‖∗‖ ln ‖x‖xk−k−1−x∗x‖∗‖ , for k = 1, 2, . . . or approximated computational order of convergence (ACOC) (see Grau-Sánchez et al. (2011)) by: η∗ = ln ‖‖‖xxkk+−1x−k−xk1‖‖‖ ln ‖x‖xk−k−1−xkx−k−1‖2‖ , for k = 2, 3, . . . The termination criteria for programming is given below: (i) ‖xk+1 − xk‖ < ϵ, (ii) ‖F (xk)‖ < ϵ, where ϵ = 10−300. Example 8.3.1. Consider the following system of nonlinear equations in four unknown vari- ables: x2x3 + (x2 + x3) x4 = 0, x2x3 + (x2 + x3) x4 = 0, x1x3 + (x1 + x3) x4 = 0, x1x2 + x3x2 + x1x3 = 1. The approximate solution is (0.5773 . . . , 0.5773 . . . , 0.5773 . . . , −0.2886 . . .)T . The obtained results can be observed in Table 8.1 based on the initial approximation (0.5, 0.5, 0.5, 0.5)T . Table 8.1: Numerical outcomes for Example 8.3.1. Methods n ‖F (xn)‖ ‖xn+1 − xn‖ η CPU Timing HM 1 2 3 1.1 × 10−8 8.4 × 10−29 2.7 × 10−2 5.5 × 10−9 4.2 × 10−29 1.4 × 10−2 3.6234 3.1473 0.0183853 2.7 × 10−2 1.4 × 10−2 F SM 1 2 3 1.1 × 10−8 8.4 × 10−29 2.7 × 10−2 5.5 × 10−9 4.2 × 10−29 1.4 × 10−2 3.6234 3.1473 0.0200469 MNM 1 2 3 1.1 × 10−8 8.4 × 10−29 2.7 × 10−2 5.5 × 10−9 4.2 × 10−29 1.4 × 10−2 3.6234 3.1473 0.0203486 N SM 1 2 3 1.1 × 10−8 8.4 × 10−29 5.5 × 10−9 4.2 × 10−29 3.6234 3.1473 0.028271 1 C LM 2 3 6.2 × 10−2 2.6 × 10−7 2.1 × 10−24 3.2 × 10−2 1.3 × 10−7 1.1 × 10−24 3.8913 0.0178618 3.1474 1 OM 2 3 1.7 × 10−2 4.5 × 10−9 2.3 × 10−29 8.5 × 10−3 2.3 × 10−9 1.2 × 10−29 3.3350 0.0250707 3.089 Example 8.3.2. We investigate a widely recognized problem in applied science, the Hammer- stein integral equation, as detailed in pages 19-20 in the work of Sharma and Gupta (2014). The primary objective is to evaluate and contrast the effectiveness and practicality of our pro- posed methods against those established earlier. The Hammerstein integral equation, presented below, serves as a standard reference for this comparative study: 1 x(s) = 1 + 1 G(s, t)x(t)3dt, 5 ∫0 where x ∈ C [0, 1], s, t ∈ [0, 1] and the kernel G is G(s, t) = (1 − s)t, t ≤ s, {s(1 − t), s ≤ t. To transform the given equation into a finite-dimensional problem, the Gauss-Legendre quadra- ture formula is applied as follows: 1 10 g(t)dt ≃ wjg(tj), ∫0 ∑j=1 where the abscissas tj and the weights wj are computed using the Gauss-Legendre quadrature formula for j = 10. Let xi(i = 1, 2, ..., 10) represent the approximations of x(ti). This leads to a system of nonlinear equations, which is given below: 10 5xi − 5 − aij x3j = 0, i = 1, 2, ..., 10 ∑j=1 where wjtj(1 − ti), j ≤ i, aij = {wjti(1 − tj), i < j. For i = j = 10, the abscissas tj and weights wj are known and shown in the Table 8.2. In Table 8.2: The abscissas tj and weights wj by Gauss Legendre quadrature formula. j tj wj 1 0 .01304673574141413996101799... 2 0.06746831665550774463395165... 3 0.16029521585048779688283632... 4 0.28330230293537640460036703... 5 0.42556283050918439455758700... 6 0.57443716949081560544241300... 7 0.71669769706462359539963297... 8 0.83970478414951220311716368... 9 0.93253168334449225536604834... 10 0.98695326425858586003898201... 0.03333567215434406879678440... 0.07472567457529029657288816... 0.10954318125799102199776746... 0.13463335965499817754561346... 0.14776211235737643508694649... 0.14776211235737643508694649... 0.13463335965499817754561346... 0.10954318125799102199776746... 0.07472567457529029657288816... 0.03333567215434406879678440... Table 8.3, we present COC, CPU timing, residual errors and the difference of errors between two iterations for Example (8.3.2). The convergence approaches towards the root, which is given as a column vector: x∗ = 1.001 . . . , 1.006 . . . , 1.014 . . . , 1.021 . . . , 1.026 . . . , 1.026 . . . , 1.021 . . . , (1.014 . . . , 1.006 . . . , 1.0013 . . . T) In addition, we choose the initial guess x0 = 1.1, 1.1, .1.0., 1.1 T for the computational work. ( ) Table 8.3: Numerical outcomes for Example 8.3.2. Methods n ‖F (xn)‖ ‖xn+1 − xn‖ η CPU Timing HM 1 2 3 1.7 × 10−17 4.7 × 10−55 3.6 × 10−5 3.6 × 10−18 1.0 × 10−55 7.6 × 10−6 2.8844 2.9970 0.140692 5.8 × 10−5 1.2 × 10−5 F SM 1 2 3 2.3 × 10−18 8.0 × 10−58 5.8 × 10−5 5.0 × 10−19 1.7 × 10−58 1.2 × 10−5 2.8949 2.9930 0.154874 MNM 1 2 3 1.7 × 10−17 4.7 × 10−55 2.7 × 10−5 3.6 × 10−18 1.0 × 10−55 5.7 × 10−6 2.8844 2.9970 0.153137 N SM 1 2 3 7.8 × 10−19 2.0 × 10−59 5.4 × 10−5 1.7 × 10−19 4.3 × 10−60 1.2 × 10−5 2.8923 2.9988 0.235745 C LM 1 2 3 1.3 × 10−17 2.0 × 10−55 2.7 × 10−5 2.8 × 10−18 4.2 × 10−56 5.7 × 10−6 2.8867 2.9987 0.150185 OM 1 2 3 7.8 × 10−19 2.0 × 10−59 1.7 × 10−19 4.3 × 10−60 2.8923 2.9988 0.291461 Example 8.3.3. Consider the Van der Pol equation Burden and Faires (2010), presented as follows: y′′ − μ(y2 − 1)y′ + y = 0, μ > 0. (8.3.6) The above expression describes the current flow in a vacuum tube with the boundary conditions y(0) = 0 and y(2) = 1. Additionally, we consider the following partition of the given interval [0,2]: x0 = 0 < x1 < x2 < x3< · · · < xθ, where xi = x0 + ip, p = 2 θ . Furthermore, we suppose that y0 = y(x0) = 0, y1 = y(x1), . . . , yθ−1 = y(xθ−1), yθ = y(xθ) = 1. If we discretize the preceding problem (8.3.6) using the second-order division difference for the first and second derivatives, we get yτ′ = yτ +1 2−h yτ −1 , yτ′′ = yτ −1 − 2hy2τ + yτ +1 , τ = 1, 2, . . . , θ − 1. The result is a (θ − 1) × (θ − 1) system of nonlinear equations, which is defined by 2h2xτ − hμ x2τ − 1 (xτ +1 − xτ −1) + 2 (xτ −1 + xτ +1 − 2xτ) = 0, τ = 1, 2, . . . , θ − 1. 1 ( ) Let μ = , and the initial approximation x0 = 9 2 , 9 T , 9 , .10.0., 9 . For θ = 101, we are dealing with a 100 × 100 system of nonlinear equa(tio1n0s a1n0d

1t0he prov1i0d)ed solution is given below: x∗ = 0.01420 . . . , 0.02833 . . . , 0.04239 . . . , 0.05637 . . . , 0.07028 . . . , 0.08411 . . . , 0(.09787 . . . , 0.1115 . . . , 0.1252 . . . , 0.1387 . . . , 0.1521 . . . , 0.1655 . . . , 0.1788 . . . , 0.1920 . . . , 0.2051 . . . , 0.2181 . . . , 0.2311 . . . , 0.2440 . . . , 0.2568 . . . , 0.2695 . . . , 0.2821 . . . , 0.2946 . . . , 0.3071 . . . , 0.3194 . . . , 0.3317 . . . , 0.3439 . . . , 0.3560 . . . , 0.3680 . . . , 0.3800 . . . , 0.3918 . . . , 0.4036 . . . , 0.4153 . . . , 0.4268 . . . , 0.4383 . . . , 0.4497 . . . , 0.4610 . . . , 0.4723 . . . , 0.4834 . . . , 0.4944 . . . , 0.5054 . . . , 0.5163 . . . , 0.5270 . . . , 0.5377 . . . , 0.5483 . . . , 0.5588 . . . , 0.5693 . . . , 0.5796 . . . , 0.5898 . . . , 0.6000 . . . , 0.6100 . . . , 0.6200 . . . , 0.6299 . . . , 0.6396 . . . , 0.6493 . . . , 0.6589 . . . , 0.6684 . . . , 0.6778 . . . , 0.6872 . . . , 0.6964 . . . , 0.7055 . . . , 0.7146 . . . , 0.7235 . . . , 0.7324 . . . , 0.7412 . . . , 0.7499 . . . , 0.7585 . . . , 0.7669 . . . , 0.7754 . . . , 0.7837 . . . , 0.7919 . . . , 0.8000 . . . , 0.8080 . . . , 0.8160 . . . , 0.8238 . . . , 0.8316 . . . , 0.8392 . . . , 0.8468 . . . , 0.8543 . . . , 0.8617 . . . , 0.8690 . . . , 0.8761 . . . , 0.8832 . . . , 0.8902 . . . , 0.8972 . . . , 0.9040 . . . , 0.9107 . . . , 0.9173 . . . , 0.9238 . . . , 0.9303 . . . , 0.9366 . . . , 0.9429 . . . , 0.9490 . . . , 0.9551 . . . , 0.9610 . . . , 0.9669 . . . , 0.9726 . . . , 0.9783 . . . , 0.9839 . . . , 0.9893 . . . , 0.9947 . . . T .) Table 8.4 shows the data for COC (Coefficient of Convergence), CPU timing, residual errors, and the difference in errors between consecutive iterations for Example (8.3.3). Example 8.3.4. We assume the 2−dimensional Burger's equation Xiao and Yin (2016), which is given by ∂2u ∂t + p∂u + g(u, t) = 0, ∂2p ∂p ∂p − Table 8.4: Numerical outcomes for Example 8.3.3. Methods n ‖F (xn)‖ ‖xn+1 − xn‖ η CPU Timing HM 1 2 3 6.8 × 10−12 8.2 × 10−34 1.7 × 10−3 9.5 × 10−10 1.1 × 10−31 3.8 × 10−2 3.2781 2.9647 3.33992 8.4 × 10−4 2.4 × 10−2 F SM 1 2 3 8.7 × 10−11 6.1 × 10−30 8.4 × 10−4 1.4 × 10−8 6.1 × 10−28 2.4 × 10−2 3.1324 3.0077 3.97881 MNM 1 2 3 6.8 × 10−12 8.2 × 10−34 2.2 × 10−5 9.5 × 10−10 1.1 × 10−31 4.1 × 10−3 3.2781 2.9647 4.13904 N SM 1 2 3 2.2 × 10−14 6.8 × 10−42 5.5 × 10−5 2.1 × 10−12 3.7 × 10−40 1.0 × 10−2 3.0648 2.9923 6.13112 C LM 1 2 3 6.1 × 10−13 3.4 × 10−37 2.8 × 10−5 6.3 × 10−11 1.8 × 10−35 3.0 × 10−3 3.1105 2.9913 3.39971 OM 1 2 3 1.0 × 10−14 2.9 × 10−43 8.1 × 10−13 1.5 × 10−41 3.0289 3.0021 5.34322 where (u,t) ∈ [0,1]×[0,1], g(u,t) = −10e−2t[et(u2−u+2)+10u(2u2−3u+1)]. The p = p(u,t) fulfills the following boundary conditions: p(0, t) = p(1, t) = 0, p(u, 0) = 10(u2 − u), p(u, 1) = 10 (u2 − u). e We assume that pi,j = p(ui,tj) is the approximate root at the grid points of the mesh. In addition, we consider M and N are the number of steps in u and t directions. The h and k are their corresponding step sizes. We can easily deduce a nonlinear system of equations from this partial differential equation by adopting a finite difference discretization. Therefore, we apply the following central difference and backward difference, respectively: ∂2p ∂2p ∂2u ∂2u(ui, tj) = = pi+1,j − 2pi,j + pi−1,j h2 , and ∂p pi+1,j − pi−1,j ∂p pi,j+1 − pi,j−1 ∂u = 2h , ∂t = 2k , in order to obtain solution. For deducing the large system of nonlinear equations 100 × 100, we use M = 11 and N = 11. Moreover, we assume 1 π π 2π 1 T x0 = sin2 , 1 sin sin , .10.0., sin π 2π 1 π sin , sin2 (10 (11) 10 (11) (11) 10 (11) (11) 10 (11)) 152 as initial vector and our required estimated zero is given below: x∗ = 0.001113 . . . , 0.001813 . . . , 0.002260 . . . , 0.002530 . . . , 0.002658 . . . , 0.002658 . . . , 0.002530 . . . , 0(.002260 . . . , 0.001813 . . . , 0.001113 . . . , 0.001813 . . . , 0.003049 . . . , 0.003870 . . . , 0.004374 . . . , 0.004614 . . . , 0.004614 . . . , 0.004374 . . . , 0.003870 . . . , 0.003049 . . . , 0.001813 . . . , 0.002260 . . . , 0.003870 . . . , 0.004969 . . . , 0.005652 . . . , 0.005979 . . . , 0.005979 . . . , 0.005652 . . . , 0.004969 . . . , 0.003870 . . . , 0.002260 . . . , 0.002530 . . . , 0.004374 . . . , 0.005652 . . . , 0.006455 . . . , 0.006841 . . . , 0.006841 . . . , 0.006455 . . . , 0.005652 . . . , 0.004374 . . . , 0.002530 . . . , 0.002658 . . . , 0.004614 . . . , 0.005979 . . . , 0.006841 . . . , 0.007257 . . . , 0.007257 . . . , 0.006841 . . . , 0.005979 . . . , 0.004614 . . . , 0.002658 . . . , 0.002658 . . . , 0.004614 . . . , 0.005979 . . . , 0.006841 . . . , 0.007257 . . . , 0.007257 . . . , 0.006841 . . . , 0.005979 . . . , 0.004614 . . . , 0.002658 . . . , 0.002530 . . . , 0.004374 . . . , 0.005652 . . . , 0.006455 . . . , 0.006841 . . . , 0.006841 . . . , 0.006455 . . . , 0.005652 . . . , 0.004374 . . . , 0.002530 . . . , 0.002260 . . . , 0.003870 . . . , 0.004969 . . . , 0.005652 . . . , 0.005979 . . . , 0.005979 . . . , 0.005652 . . . , 0.004969 . . . , 0.003870 . . . , 0.002260 . . . , 0.001813 . . . , 0.003049 . . . , 0.003870 . . . , 0.004374 . . . , 0.004614 . . . , 0.004614 . . . , 0.004374 . . . , 0.003870 . . . , 0.003049 . . . , 0.001813 . . . , 0.001113 . . . , 0.001813 . . . , 0.002260 . . . , 0.002530 . . . , 0.002658 . . . , 0.002658 . . . , 0.002530 . . . , 0.002260 . . . , 0.001813...,0.001113... T . In Table 8.5, we present COC, CPU timing, residual errors and errors difference between two) iterations for Example (8.3.4). Example 8.3.5. We assume a nonlinear system (selected from Grau-Sánchez et al. (2011)), defined as follows: n F (x) = xk − cos 2xk − xi ( ) , 1 ≤ k ≤ n. (8.3.7) We choose n = 95 and the initial guess x(0) = (0.21, 0.21, 0.21, .9.5.0.21)T for this problem. ∑i=1 The required solution is 0.2172 . . . , 0.2172 . . . , .9.5., 0.2172 . . . . The obtained results can be T observed in Table 8.6. ( ) 8.4 Conclusions Table 8.5: Numerical outcomes for Example 8.3.4. Methods n ‖F (xn)‖ ‖xn+1 − xn‖ η CPU Timing HM 1 2 3 1.0 × 10−7 1.6 × 10−25 6.0 × 10−79 1.9 × 10−7 5.6 × 10−7 8.4 × 10−25 3.0 × 10−78 1.1 × 10−6 2.9929 2.9988 23.6653 F SM 1 2 3 2.2 × 10−24 2.8 × 10−75 1.0 × 10−7 1.1 × 10−23 1.4 × 10−74 2.9921 2.9987 24.7904 MNM 1 2 3 1.6 × 10−25 6.0 × 10−79 2.8 × 10−9 ×10− 8.4 × 10−25 3.0 × 10−78 1.6 × 10−8 2.9929 2.9988 26.1044 N SM 1 2 3 9.2 × 10−32 3.3 × 10−99 5.6 × 10−9 5.3 × 10−31 1.9 × 10−98 3.3 × 10−8 3.0007 2.9998 34.6971 C LM 1 2 3 1.5 × 10−30 3.1 × 10−95 2.8 × 10−9 8.9 × 10−30 1.8 × 10−94 1.6 × 10−8 3.0016 2.9998 22.3362 OM 1 2 3 9.3 × 10−32 3.4 × 10−99 5.4 × 10−31 1.9 × 10−98 3.0008 2.9998 47.8200 Table 8.6: Numerical outcomes for Example 8.3.5. Methods n ‖F (xn)‖ ‖xn+1 − xn‖ η CPU Timing HM 1 2 3 6.7 × 10−5 1.7 × 10−16 1.5 × 10−1 7.3 × 10−7 1.9 × 10−18 1.7 × 10−3 3.5100 3.0074 88.7778 4.7 × 10−1 5.2 × 10−3 F SM 1 2 3 2.8 × 10−6 1.6 × 10−20 4.7 × 10−1 3.0 × 10−8 1.9 × 10−18 5.2 × 10−3 2.9026 3.0074 130.598 MNM 1 2 3 6.7 × 10−5 1.7 × 10−16 2.8 × 10−1 7.3 × 10−7 1.9 × 10−18 3.1 × 10−3 3.5100 3.0074 134.812 N SM 1 2 3 3.3 × 10−6

4.9 × 10−21 3.6 × 10−8 5.3 × 10−23 1.3 × 10−2 3.6710 3.0120 C LM 1 2 3 1.2 7.9 × 10−4 1.3 × 10−13 8.6 × 10−6 1.4 × 10−15 4.8498 3.0858 88.2526 1 OM 2 3 2.8 × 10−1 3.3 × 10−6 4.6 × 10−21 3.0 × 10−3 3.6 × 10−8 5.1 × 10−23 3.6683 202.62 3.0120 Table 8.7: Number of iterations for Examples 8.3.1– 8.3.5. Methods HM F SM MNM NSM CLM OM Ex. 8.3.1 6 6 6 6 6 6 6 Ex. 8.3.2 5 5 5 5 5 5 Ex. 8.3.3 6 6 6 5 5 5 Ex. 8.3.4 5 5 5 4 5 4 Ex. 8.3.5 6 6 6 6 6 6 (Ex. stands for example) Table 8.8: Comparison on basis of ‖F (x5)‖ for Examples 8.3.1– 8.3.5. Methods HM F SM MNM NSM CLM OM Ex. 8.3.1 6.3 × 10−275 3.2 × 10−275 3.2 × 10−275 3.2 × 10−275 1.3 × 10−234 7.7 × 10−276 Ex. 8.3.2 9.7 × 10−506 2.7 × 10−531 9.7 × 10−506 1.6 × 10−546 2.3 × 10−509 1.6 × 10−546 Ex. 8.3.3 3.0 × 10−297 2.9 × 10−720 1.2 × 10−261 4.6 × 10−686 3.0 × 10−297 2.9 × 10−720 2.2 × 10−377 1.5 × 10−908 3.7 × 10−333 1.3 × 10−871 8.9 × 10−389 1.8 × 10−908 Ex. 8.3.4 1.6 × 10−155 2.8 × 10−191 1.6 × 10−155 4.4 × 10−199 4.7 × 10−131 2.9 × 10−199 Ex. 8.3.5 Bibliography Abbaoui, K. and Cherruault, Y. (1994). Convergence of adomian's method applied to nonlin- ear equations. Mathematical and Computer Modelling, 20(9):69–73. Abbasbandy, S. (2003). Improving newton–raphson method for nonlinear equations by modi- fied adomian decomposition method. Applied mathematics and computation, 145(2-3):887– 893. Adomian, G. (1988). Nonlinear stochastic systems theory and applications to physics, vol- ume 46. Springer Science & Business Media. Adomian, G. (2013). Solving frontier problems of physics: the decomposition method, vol- ume 60. Springer Science & Business Media. Adomian, G. and Rach, R. (1985). On the solution of algebraic equations by the decomposition method. Journal of mathematical analysis and applications, 105(1):141–166. Ahlfors, L. V. et al. (1966). Complex analysis: an introduction to the theory of analytic functions of one complex variable. New York: McGraw-Hill,. Amat, S., Busquier, S., and Gutiérrez, J. M. (2003). Geometric constructions of iterative functions to solve nonlinear equations. Journal of Computational and Applied Mathematics, 157(1):197–205. Amat, S., Busquier, S., and Plaza, S. (2004). Review of some iterative root-finding methods from a dynamical point of view. Scientia, 10(3):35. Ardelean, G. (2011). A comparison between iterative methods by using the basins of attrac- tion. Applied Mathematics and computation, 218(1):88–95. Argyros, I. K., Kansal, M., Kanwar, V., and Bajaj, S. (2017). Higher-order derivative-free families of chebyshev–halley type methods with or without memory for solving nonlinear equations. Applied Mathematics and Computation, 315:224–245. Babajee, D., Dauhoo, M. Z., Darvishi, M. T., and Barati, A. (2008). A note on the local convergence of iterative methods based on adomian decomposition method and 3-node quadrature rule. Applied Mathematics and Computation, 200(1):452–458. Babajee, D., Dauhoo, M. Z., Darvishi, M. T., Karami, A., and Barati, A. (2010). Analysis of two chebyshev-like third order methods free from second derivatives for solving systems of nonlinear equations. Journal of Computational and Applied Mathematics, 233(8):2002–2012. Basu, D. (2008). From third to fourth order variant of newton's method for simple roots. Applied mathematics and computation, 202(2):886–892. Beardon, A. F. (1991). Iteration of rational functions (graduate texts in mathematics, 132). Beauzamy, B. (2011). Introduction to Banach spaces and their geometry. Elsevier. Behl, R. and Al-Hamdan, W. M. (2019). A 4th-order optimal extension of ostrowski's method for multiple zeros of univariate nonlinear functions. Mathematics, 7(9):803. Behl, R., Kansal, M., and Salimi, M. (2020). Modified king's family for multiple zeros of scalar nonlinear functions. Mathematics, 8(5):827. Behl, R., Zafar, F., Alshormani, A. S., Junjua, M.-U.-D., and Yasmin, N. (2019). An optimal eighth-order scheme for multiple zeros of univariate functions. International Journal of Computational Methods, 16(04):1843002. Bi, W., Ren, H., and Wu, Q. (2008). New family of seventh-order methods for nonlinear equations. Applied Mathematics and Computation, 203(1):408–412. Bi, W., Ren, H., and Wu, Q. (2009a). Three-step iterative methods with eighth-order conver- gence for solving nonlinear equations. Journal of Computational and Applied Mathematics, 225(1):105–112. Bi, W., Wu, Q., and Ren, H. (2009b). A new family of eighth-order iterative methods for solving nonlinear equations. Applied Mathematics and Computation, 214(1):236–245. Blanchard, P. (1984). Complex analytic dynamics on the riemann sphere. Bulletin of the American mathematical Society, 11(1):85–141. Blanchard, P. et al. (1994). The dynamics of newton's method. In Proceedings of Symposia in Applied Mathematics, volume 49, pages 139–154. American Mathematical Society Prov- idence, RI, USA. Bodewig, E. (1946). Sur la méthode laguerre pour l'approximation des racines de certaines équations algébriques et sur la critique d'hermite. Indag. Math, 8:570–580. Bradie, B. (2006). A friendly introduction to numerical analysis. Pearson Education India. Brolin, H. (1965). Invariant sets under iteration of rational functions. Arkiv för matematik, 6(2):103–144. Burden, R. L. and Faires, J. D. (2010). Numerical analysis. Campos, B., Cordero, A., Torregrosa, J. R., and Vindel, P. (2015). A multidimensional dynam- ical approach to iterative methods with memory. Applied Mathematics and Computation, 271:701–715. Campos, B., Cordero, A., Torregrosa, J. R., and Vindel, P. (2017). Stability of king's family of iterative methods with memory. Journal of Computational and Applied Mathematics, 318:504–514. Cayley, A. (1879a). Application of the newton-fourier method to an imaginary root of an equation. Quart. J. Pure Appl. Math, 16:179–185. Cayley, A. (1890). Sur les racines d'une équation algébrique. CRAS, 110:215–218. Cayley, P. (1879b). Desiderata and suggestions: no. 3. the newton-fourier imaginary problem. American Journal of Mathematics, 2(1):97–97. Chapra, S. C. and Canale, R. P. (2015). Numerical Methods for Engineers. 7th ed. McGraw- Hill, New York. Chen, D., Argyros, I., and Qian, Q. (1993). A note on the halley method in banach spaces. Applied Mathematics and Computation, 58(2-3):215–224. Chicharro, F., Cordero, A., Gutiérrez, J. M., and Torregrosa, J. R. (2013a). Complex dynamics of derivative-free methods for nonlinear equations. Applied Mathematics and Computation, 219(12):7023–7035. Chicharro, F. I., Cordero, A., Garrido, N., and Torregrosa, J. R. (2019). Stability and applica- bility of iterative methods with memory. Journal of Mathematical Chemistry,

57(5):1282– 1300. Chicharro, F. I., Cordero, A., Torregrosa, J. R., et al. (2013b). Drawing dynamical and parameters planes of iterative families and methods. The Scientific World Journal, 2013. Chicharro, F. I., Cordero, A., Torregrosa, J. R., Vassileva, M. P., et al. (2017). King-type derivative-free iterative families: Real and memory dynamics. Complexity, 2017. Chun, C. (2005). Iterative methods improving newton's method by the decomposition method. Computers & Mathematics with Applications, 50(10-12):1559–1568. Chun, C. (2007a). A family of composite fourth-order iterative methods for solving nonlinear equations. Applied mathematics and computation, 187(2):951–956. Chun, C. (2007b). A geometric construction of iterative functions of order three to solve nonlinear equations. Computers & Mathematics with Applications, 53(6):972–976. Chun, C. (2007c). On the construction of iterative methods with at least cubic convergence. Applied mathematics and computation, 189(2):1384–1392. Chun, C. (2007d). A one-parameter family of third-order methods to solve nonlinear equations. Applied mathematics and computation, 189(1):126–130. Chun, C. (2007e). Some improvements of jarratt's method with sixth-order convergence. Applied Mathematics and Computation, 190(2):1432–1437. Chun, C. (2007f). Some variants of chebyshev–halley methods free from second derivative. Applied Mathematics and Computation, 191(1):193–198. Chun, C. (2007g). Some variants of king's fourth-order family of methods for nonlinear equations. Applied mathematics and computation, 190(1):57–62. Chun, C. (2008). Some fourth-order iterative methods for solving nonlinear equations. Applied mathematics and computation, 195(2):454–459. Chun, C. and Ham, Y. (2007a). A one-parameter fourth-order family of iterative methods for nonlinear equations. Applied Mathematics and Computation, 189(1):610–614. Chun, C. and Ham, Y. (2007b). Some sixth-order variants of ostrowski root-finding methods. Applied Mathematics and Computation, 193(2):389–394. Chun, C. and Ham, Y. (2008a). Some fourth-order modifications of newton's method. Applied Mathematics and Computation, 197(2):654–658. Chun, C. and Ham, Y. (2008b). Some second-derivative-free variants of super-halley method with fourth-order convergence. Applied mathematics and computation, 195(2):537–541. Chun, C. and Kim, Y.-I. (2010). Several new third-order iterative methods for solving non- linear equations. Acta applicandae mathematicae, 109:1053–1063. Chun, C., Lee, M. Y., Neta, B., and Džunić, J. (2012). On optimal fourth-order iterative meth- ods free from second derivative and their dynamics. Applied mathematics and computation, 218(11):6427–6438. Chun, C. and Neta, B. (2008). Some modification of newton's method by the method of undetermined coefficients. Computers & Mathematics with Applications, 56(10):2528–2538. Chun, C. and Neta, B. (2009). Certain improvements of newton's method with fourth-order convergence. Applied Mathematics and Computation, 215(2):821–828. Chun, C. and Neta, B. (2012). A new sixth-order scheme for nonlinear equations. Applied mathematics letters, 25(2):185–189. Constantinides, A. and Mostoufi, N. (1999). Numerical methods for chemical engineers with matlab applications with cdrom. Prentice Hall PTR. Cordero, A., García-Maimó, J., Torregrosa, J. R., Vassileva, M. P., and Vindel, P. (2013a). Chaos in king's iterative family. Applied Mathematics Letters, 26(8):842–848. Cordero, A., Hueso, J. L., Martínez, E., and Torregrosa, J. R. (2010a). A modified newton- jarratt's composition. Numerical Algorithms, 55:87–99. Cordero, A., Hueso, J. L., Martínez, E., and Torregrosa, J. R. (2010b). New modifications of potra–pták's method with optimal fourth and eighth orders of convergence. Journal of computational and applied mathematics, 234(10):2969–2976. Cordero, A., Hueso, J. L., Martínez, E., and Torregrosa, J. R. (2012). Increasing the con- vergence order of an iterative method for nonlinear systems. Applied Mathematics Letters, 25(12):2369–2374. Cordero, A., Hueso, J. L., Martínez, E., and Torregrosa, J. R. (2013b). Generating optimal derivative free iterative methods for nonlinear equations by using polynomial interpolation. Mathematical and Computer Modelling, 57(7-8):1950–1956. Cordero, A., Hueso, J. L., Martínez, E., and Torregrosa, J. R. (2013c). A new technique to obtain derivative-free optimal iterative methods for solving nonlinear equations. Journal of Computational and Applied Mathematics, 252:95–102. Cordero, A., Kansal, M., Kanwar, V., and Torregrosa, J. R. (2016). A stable class of im- proved second-derivative free chebyshev-halley type methods with optimal eighth order convergence. Numerical Algorithms, 72(4):937–958. Cordero, A., Leonardo Sepúlveda, M. A., and Torregrosa, J. R. (2022). Dynamics and stability on a family of optimal fourth-order iterative methods. Algorithms, 15(10):387. Cordero, A., Lotfi, T., Bakhtiari, P., and Torregrosa, J. R. (2015). An efficient two-parametric family with memory for nonlinear equations. Numerical Algorithms, 68(2):323–335. Cordero, A., Martínez, E., and Torregrosa, J. R. (2009). Iterative methods of order four and five for systems of nonlinear equations. Journal of Computational and Applied Mathematics, 231(2):541–551. Cordero, A., Moscoso-Martínez, M., and Torregrosa, J. R. (2021). Chaos and stability in a new iterative family for solving nonlinear equations. Algorithms, 14(4):101. Cordero, A. and Torregrosa, J. R. (2006). Variants of newton's method for functions of several variables. Applied Mathematics and Computation, 183(1):199–208. Cordero, A. and Torregrosa, J. R. (2007). Variants of newton's method using fifth-order quadrature formulas. Applied Mathematics and Computation, 190(1):686–698. Cordero, A. and Torregrosa, J. R. (2011). A class of steffensen type methods with optimal order of convergence. Applied Mathematics and Computation, 217(19):7653–7659. Cordero, A., Torregrosa, J. R., and Vindel, P. (2013d). Dynamics of a family of chebyshev– halley type methods. Applied Mathematics and Computation, 219(16):8568–8583. Curry, J. H., Garnett, L., and Sullivan, D. (1983). On the iteration of a rational function: computer experiments with newton's method. Communications in mathematical physics, 91:267–277. Darvishi, M. T. and Barati, A. (2007a). A fourth-order method from quadrature formulae to solve systems of nonlinear equations. Applied Mathematics and Computation, 188(1):257– 261. Darvishi, M. T. and Barati, A. (2007b). A third-order newton-type method to solve systems of nonlinear equations. Applied Mathematics and Computation, 187(2):630–635. Devaney, R. (2018). An introduction to chaotic dynamical systems.

CRC press. Dong, C. (1982). A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation. Math. Numer. Sinica, 11:445–450. Dong, C. (1987). A family of multiopoint iterative functions for finding multiple roots of equations. International Journal of Computer Mathematics, 21(3-4):363–367. Douady, A. (1982). Itération des polynômes quadratiques complexes. CR Acad. Sci. Paris Sér. I Math., 294:123–126. Douglas, J. (1972). Process Dynamics and Control. vol. 2. Prentice Hall, Englewood Cliffs. Džunić, J. (2011). On the similarity of some three-point methods for solving nonlinear equa- tions. Applied mathematics and computation, 217(14):6633–6635. Džunić, J. (2013). On efficient two-parameter methods for solving nonlinear equations. Nu- merical algorithms, 63(3):549–569. Džunić, J., Petković, M., and Petković, L. (2012). Three-point methods with and with- out memory for solving nonlinear equations. Applied Mathematics and Computation, 218(9):4917–4927. Fatou, P. (1919). Sur les équations fonctionnelles premier mémoire. Bulletin de la Société mathématique de France, 47:161–271. Frontini, M. and Sormani, E. (2003). Some variant of newton's method with third-order convergence. Applied Mathematics and Computation, 140(2-3):419–426. Frontini, M. and Sormani, E. (2004). Third-order methods from quadrature formulae for solving systems of nonlinear equations. Applied Mathematics and Computation, 149(3):771– 782. Gautschi, W. (2011). Numerical analysis. Springer Science & Business Media. Geum, Y. H. and Kim, Y. I. (2011a). A biparametric family of optimally convergent sixteenth- order multipoint methods with their fourth-step weighting function as a sum of a rational and a generic two-variable function. Journal of Computational and Applied Mathematics, 235(10):3178–3188. Geum, Y. H. and Kim, Y. I. (2011b). A family of optimal sixteenth-order multipoint methods with a linear fraction plus a trivariate polynomial as the fourth-step weighting function. Computers & Mathematics with Applications, 61(11):3278–3287. Golbabai, A. and Javidi, M. (2007). A new family of iterative methods for solving system of nonlinear algebric equations. Applied mathematics and computation, 190(2):1717–1722. Grau, M. and Díaz-Barrero, J. L. (2006). An improvement to ostrowski root-finding method. Applied Mathematics and Computation, 173(1):450–456. Grau-Sánchez, M., Grau, À., and Noguera, M. (2011). On the computational efficiency in- dex and some iterative methods for solving systems of nonlinear equations. Journal of Computational and Applied Mathematics, 236(6):1259–1266. Guckenheimer, J. (1970). Endomorphisms of the Riemann sphere. University of California, Berkeley. Gutiérrez, J. M. and Hernández, M. A. (1997). A family of chebyshev-halley type methods in banach spaces. Bulletin of the Australian Mathematical Society, 55(1):113–130. Hansen, E. and Patrick, M. (1976). A family of root finding methods. Numerische mathematik, 27(3):257–269. Homeier, H. (2003). A modified newton method for rootfinding with cubic convergence. Journal of Computational and Applied Mathematics, 157(1):227–230. Homeier, H. H. (2004). A modified newton method with cubic convergence: the multivariate case. Journal of Computational and Applied Mathematics, 169(1):161–169. Homeier, H. H. (2005). On newton-type methods with cubic convergence. Journal of compu- tational and applied mathematics, 176(2):425–432. Jain, D. (2013). Families of newton-like methods with fourth-order convergence. International Journal of computer mathematics, 90(5):1072–1082. Jakobson, M. V. (1968). Structure of polynomial mappings on a singular set. Mathematics of the USSR-Sbornik, 6(1):97. Jarratt, P. (1966). Some fourth order multipoint iterative methods for solving equations. Mathematics of computation, 20(95):434–437. Jarratt, P. (1969). Some efficient fourth order multipoint methods for solving equations. BIT Numerical Mathematics, 9(2):119–124. Jay, L. O. (2001). A note on q-order of convergence. BIT Numerical Mathematics, 41:422–429. Julia, G. (1918). Mémoire sur l'itération des applications fonctionnelles. J. Math. Pures et Appl, 8(1):47–245. Kansal, M., Alshomrani, A. S., Bhalla, S., Behl, R., and Salimi, M. (2020). One parameter optimal derivative-free family to find the multiple roots of algebraic nonlinear equations. Mathematics, 8(12):2223. Kansal, M., Kanwar, V., and Bhatia, S. (2015a). New modifications of hansen–patrick's family with optimal fourth and eighth orders of convergence. Applied Mathematics and Computation, 269:507–519. Kansal, M., Kanwar, V., and Bhatia, S. (2015b). On some optimal multiple root-finding methods and their dynamics. Applications and Applied Mathematics: An International Journal (AAM), 10(1):22. Kansal, M., Kanwar, V., and Bhatia, S. (2016). Efficient derivative-free variants of hansen- patrick's family with memory for solving nonlinear equations. Numerical algorithms, 73:1017–1036. Kantorovich, L. V. and Akilov, G. P. (2014). Functional analysis. Elsevier. Kanwar, V. (2006). A family of third-order multipoint methods for solving nonlinear equations. Applied mathematics and computation, 176(2):409–413. Kanwar, V., Singh, S., and Bakshi, S. (2008). Simple geometric constructions of quadrati- cally and cubically convergent iterative functions to solve nonlinear equations. Numerical Algorithms, 47:95–107. Kanwar, V., Singh, S., Guha, R. K., et al. (2006). On method of osculating circle for solving nonlinear equations. Applied mathematics and computation, 176(1):379–382. Kelley, C. T. (2003). Solving nonlinear equations with Newton's method. SIAM. King, R. F. (1973a). A family of fourth order methods for nonlinear equations. SIAM journal on numerical analysis, 10(5):876–879. King, R. F. (1973b). An improved pegasus method for root finding. BIT Numerical Mathe- matics, 13:423–427. Kneisl, K. (2001). Julia sets for the super-newton method, cauchy's method, and halley's method. Chaos: An Interdisciplinary Journal of Nonlinear Science, 11(2):359–370. Kou, J. (2007). A third-order modification of newton method for systems of non-linear equa- tions. Applied mathematics and computation, 191(1):117–121. Kou, J. and Li, Y. (2007a). A family of modified super-halley methods with fourth-order convergence. Applied mathematics and computation, 189(1):366–370. Kou, J. and Li, Y. (2007b). An improvement of the jarratt method. Applied mathematics and Computation, 189(2):1816–1821. Kou, J., Li, Y., and Wang, X. (2006). A modification of newton method with third-order convergence. Applied Mathematics and Computation, 181(2):1106–1111. Kou, J., Li, Y., and Wang, X. (2007a). A composite fourth-order iterative method for solving non-linear equations. Applied Mathematics and Computation, 184(2):471–475. Kou, J., Li, Y., and Wang, X. (2007b). Fourth-order iterative methods free from second derivative. Applied mathematics and computation, 184(2):880–885. Kou, J., Li, Y., and Wang, X. (2007c). Some variants of ostrowski's method with seventh-order convergence. Journal of Computational and Applied Mathematics, 209(2):153–159. Kou, J., Wang, X., and Li, Y. (2010). Some eighth-order root-finding three-step methods. Communications in Nonlinear Science and Numerical Simulation, 15(3):536–544. Kung, H. and Traub, J. F. (1974). Optimal order of one-point and multipoint iteration. Journal of the ACM (JACM), 21(4):643–651. Lagouanelle, J.-L. (1966). Sur une methode de calcul de lordre de multiplicite des zeros dun polynome. COMPTES RENDUS HEBDOMADAIRES DES SEANCES DE L ACADEMIE DES SCIENCES SERIE A, 262(11):626–+. Laguerre, E. (1878). Sur la résolution des équations numériques. Nouvelles Annales de math-ematiques, ser, 2:97–101. Li, D., Liu, P., and Kou, J. (2014). An improvement of chebyshev–halley methods free from second derivative. Applied Mathematics and Computation, 235:221–225. Li, S., Cheng, L., and Neta, B. (2010). Some fourth-order nonlinear solvers with closed formulae for multiple roots. Computers & Mathematics with Applications, 59(1):126–135. Liu, L. and Wang, X. (2010). Eighth-order methods with high efficiency index for solving nonlinear equations. Applied Mathematics and Computation, 215(9):3449–3454. Liu, Z., Zheng, Q., and Zhao, P. (2010). A variant of steffensen's method of fourth-order convergence and its applications. Applied Mathematics and Computation, 216(7):1978–1983. Mandelbrot, B. (1980). Fractal aspects of z λz (1- z) for complex λ and z. Ann. NY Acad. Sci, 357:249–259. Mané, R., Sad, P., and Sullivan, D. (1983). On the dynamics of rational maps. In Annales scientifiques de l'École Normale Supérieure, volume 16, pages 193–217. McNamee, J. M. and Pan, V. (2013). Numerical Methods for Roots of Polynomials-Part II. Newnes. Milovanović, G. V. and Cvetković, A. S. (2007). A note on three-step iterative methods for nonlinear equations. Stud Univ Babes-Bolyai Math, 52(3):137–146. Mir, N. A. and Zaman, T. (2007). Some quadrature based three-step iterative methods for non-linear equations. Applied Mathematics and Computation, 193(2):366–373. Narici, L. and Beckenstein, E. (2010). Topological vector spaces. Chapman and Hall/CRC. Nedzhibov, G. H. (2008). A family of multi-point iterative methods for solving systems of nonlinear equations. Journal of Computational and Applied Mathematics, 222(2):244–250. Neta, B. (1981). On a family of multipoint methods for non-linear equations. International Journal of Computer Mathematics, 9(4):353–361. Neta, B. (1983). A new family of higher order methods for solving equations. International Journal of Computer Mathematics, 14(2):191–195. Neta, B. (2008). New third order nonlinear solvers for multiple roots. Applied Mathematics and Computation, 202(1):162–170. Neta, B. et al. (1979). A sixth-order family of methods for nonlinear equations. Int. J. Comput. Math, 7(1997):157–161. Neta, B. and Petković, M. (2010). Construction of optimal order nonlinear solvers using inverse interpolation. Applied Mathematics and Computation, 217(6):2448–2455. Noor, M. A. and Waseem, M. (2009). Some iterative methods for solving a system of nonlinear equations. Computers & Mathematics with Applications, 57(1):101–106. Ortega, J. M. and Rheinboldt, W. C. (1970). Iterative solution of nonlinear equations in several variables. Academic Press. Osada, N. (1994). An optimal multiple root-finding method of order three. Journal of Com- putational and Applied Mathematics, 51(1):131–133. Ostrowski, A. M. (1960). Solution of equations and systems of equations. Academic press. Özban, A. Y. (2004). Some new variants of newton's method. Applied Mathematics Letters, 17(6):677–682. Özyapıcı, A., Riza, M., Bilgehan, B., and Bashirov, A. E. (2014). On multiplicative and volterra minimization methods. Numerical Algorithms, 67:623–636. Padilla, J. J., Chicharro, F. I., Cordero, A., and Torregrosa, J. R. (2022). Parametric family of root-finding iterative methods: Fractals of the basins of attraction. Fractal and Fractional, 6(10):572. Panday, S., Sharma, A., and Thangkhenpau, G. (2023). Optimal fourth and eighth-order iterative methods for non-linear equations. Journal of Applied Mathematics and Computing, 69(1):953–971. Parhi, S. K. and Gupta, D. K. (2008). A sixth order method for nonlinear equations. Applied Mathematics and Computation, 203(1):50–55. Peng, Y., Feng, H., Li, Q., and Zhang, X. (2011). A fourth-order derivative-free algorithm for nonlinear equations. Journal of computational and applied mathematics, 235(8):2551–2559. Petković, L. and Petković, M. (2008). The link between schröder's iteration methods of the first and second kind. Novi Sad Journal of Mathematics, 38(3):55–63. Petković, L. D., Petković, M. S., and Džunić, J. (2010a). A class of three-point root-solvers of optimal order of convergence. Applied Mathematics and Computation, 216(2):671–676. Petković, M. and Herceg, D. (1999). On rediscovered iteration methods for solving equations. Journal of Computational and Applied Mathematics, 107(2):275–284. Petković, M., Ilić, S., and Džunić, J. (2010b). Derivative free two-point methods with and without memory for solving nonlinear equations. Applied mathematics and computation, 217(5):1887–1895. Petkovic, M., Neta, B., Petkovic, L., and Dzunic, J. (2012). Multipoint methods for solving nonlinear equations. Academic press. Petković, M., Petković, L., and Herceg, D. (2010c). On schröder's families of root-finding methods. Journal of computational and applied mathematics, 233(8):1755–1762. Petković, M. S. (2010). On a general class of multipoint root-finding methods of high compu- tational efficiency. SIAM Journal on Numerical Analysis, 47(6):4402–4414. Petković, M. S. (2011). Remarks on "on a general class of multipoint root-finding methods of high computational efficiency". SIAM journal on numerical analysis, 49(3):1317–1319. Petković, M. S., Džunić, J., and Neta, B. (2011). Interpolatory multipoint methods with mem- ory for solving nonlinear equations. Applied Mathematics and Computation, 218(6):2533– 2541. Petković, M. S., Neta, B., Petković, L. D., and Džunić, J. (2014). Multipoint methods for solving nonlinear equations: A survey. Applied Mathematics and Computation, 226:635–660. Petković, M. S. and Petković, L. D. (2010). Families of optimal multipoint methods for solving nonlinear equations: a survey. Applicable Analysis and Discrete

Mathematics, 4:1–22. Popovski, D. (1981). A note on neta's family of sixth-order methods for solving equations. International Journal of Computer Mathematics, 10(1):91–93. Potra, F. and Pták, V. (1984). Nondiscrete induction and iterative processes. Pitman Pub- lishing. Rall, L. B. (1966). Convergence of the newton process to multiple solutions. Numerische Mathematik, 9:23–37. Rehman, M., Naseem, A., and Abdeljawad, T. (2021). Some novel sixth-order iteration schemes for computing zeros of nonlinear scalar equations and their applications in engi- neering. Journal of Function Spaces, 2021:1–11. Ren, H., Wu, Q., and Bi, W. (2009). A class of two-step steffensen type methods with fourth- order convergence. Applied Mathematics and Computation, 209(2):206–210. Roberts, G. E. and Horgan-Kobelski, J. (2004). Newton's versus halley's method: A dynamical systems approach. International Journal of Bifurcation and Chaos, 14(10):3459–3475. Robinson, R. C. (2012). An introduction to dynamical systems: continuous and discrete, volume 19. American Mathematical Soc. Salehov, G. (1952). On the convergence of the process of tangent hyperbolas. In Dokl. Akad. Nauk SSSR, volume 82, pages 525–528. Schröder, E. (1870a). Über unendlich viele algorithmen zur auflösung der gleichungen. Math- ematische Annalen, 2(2):317–365. Schröder, E. (1870b). Ueber iterirte functionen. Mathematische Annalen, 3(2):296–322. Shacham, M. (1986). Numerical solution of constrained non-linear algebraic equations. Inter- national journal for numerical methods in engineering, 23(8):1455–1481. Sharifi, M., Babajee, D., and Soleymani, F. (2012). Finding the solution of nonlinear equations by a class of optimal methods. Computers & mathematics with applications, 63(4):764–774. Sharifi, S., Siegmund, S., and Salimi, M. (2016). Solving nonlinear equations by a derivative- free form of the king's family with memory. Calcolo, 53:201–215. Sharma, J. (2005). A composite third order newton–steffensen method for solving nonlinear equations. Applied Mathematics and Computation, 169(1):242–246. Sharma, J. (2007). A family of third-order methods to solve nonlinear equations by quadratic curves approximation. Applied mathematics and computation, 184(2):210–215. Sharma, J. R. and Goyal, R. (2006). Fourth-order derivative-free methods for solving non- linear equations. International Journal of Computer Mathematics, 83(1):101–106. Sharma, J. R. and Guha, R. K. (2007). A family of modified ostrowski methods with accel- erated sixth order convergence. Applied Mathematics and Computation, 190(1):111–115. Sharma, J. R. and Guha, R. K. (2011). Second-derivative free methods of third and fourth order for solving nonlinear equations. International Journal of Computer Mathematics, 88(1):163–170. Sharma, J. R., Guha, R. K., and Sharma, R. (2009). Some variants of hansen–patrick method with third and fourth order convergence. Applied mathematics and computation, 214(1):171– 177. Sharma, J. R. and Gupta, P. (2014). An efficient fifth order method for solving systems of nonlinear equations. Computers & Mathematics with Applications, 67(3):591–601. Sharma, J. R., Kumar, S., and Jäntschi, L. (2019). On a class of optimal fourth order multiple root solvers without using derivatives. Symmetry, 11(12):1452. Sharma, J. R. and Sharma, R. (2010). Modified jarratt method for computing multiple roots. Applied Mathematics and Computation, 217(2):878–881. Shengguo, L., Xiangke, L., and Lizhi, C. (2009). A new fourth-order iterative method for finding multiple roots of nonlinear equations. Applied Mathematics and Computation, 215(3):1288–1292. Soleimani, F., Soleymani, F., and Shateyi, S. (2013). Some iterative methods free from derivatives and their basins of attraction for nonlinear equations. Discrete Dynamics in Nature and Society, 2013. Soleymani, F. and Babajee, D. (2013). Computing multiple zeros using a class of quartically convergent methods. Alexandria Engineering Journal, 52(3):531–541. Soleymani, F., Babajee, D., and Lotfi, T. (2013). On a numerical technique for finding multiple zeros and its dynamic. Journal of the Egyptian Mathematical Society, 21(3):346–353. Soleymani, F., Khattri, S. K., and Vanani, S. K. (2012a). Two new classes of optimal jarratt-type fourth-order methods. Applied mathematics letters, 25(5):847–853. Soleymani, F., Lotfi, T., Tavakoli, E., and Haghani, F. K. (2015). Several iterative methods with memory using self-accelerators. Applied Mathematics and Computation, 254:452–458. Soleymani, F., Sharifi, M., and Somayeh Mousavi, B. (2012b). An improvement of ostrowski's and king's techniques with optimal convergence order eight. Journal of Optimization Theory and Applications, 153(1):225–236. Soleymani, F., Sharma, R., Li, X., and Tohidi, E. (2012c). An optimized derivative-free form of the potra–pták method. Mathematical and computer modelling, 56(5-6):97–104. Soleymani, F., Vanani, S. K., Khan, M., and Sharifi, M. (2012d). Some modifications of king's family with optimal eighth order of convergence. Mathematical and Computer Modelling, 55(3-4):1373–1380. Steffensen, J. (1933). Remarks on iteration. Scandinavian Actuarial Journal, 1933(1):64–72. Sullivan, D. (1982). Itération des fonctions analytiques complexes. Thukral, R. (2010). A new eighth-order iterative method for solving nonlinear equations. Applied Mathematics and Computation, 217(1):222–229. Thukral, R. and Petković, M. (2010). A family of three-point methods of optimal order for solv- ing nonlinear equations. Journal of Computational and Applied Mathematics, 233(9):2278– 2284. Traub, J. F. (1964). Iterative methods for the solution of equations. Prentice-Hall, Englewood Cliffs. Varona, J. L. (2002). Graphic and numerical comparison between iterative methods. Mathe- matical Intelligencer, 24(1):37–47. Victory Jr, H. and Neta, B. (1983). A higher order method for multiple zeros of nonlinear functions. International Journal of Computer Mathematics, 12(3-4):329–335. Vrscay, E. R. (1986). Julia sets and mandelbrot-like sets associated with higher order schröder rational iteration functions: a computer assisted study. Mathematics of Computation, 46(173):151–169. Vrscay, E. R. and Gilbert, W. J. (1987). Extraneous fixed points, basin boundaries and chaotic dynamics for schröder and könig rational iteration functions. Numerische Mathematik, 52:1– 16. Weerakoon, S. and Fernando, T. (2000). A variant of newton's method with accelerated third-order convergence. Applied Mathematics Letters, 13(8):87–93. Wilczek-Vera, G. and Vera, J. H. (2015). Understanding cubic equations of state: A search for the hidden clues of their success. AIChE Journal, 61(9):2824–2831. Xiao, X. and Yin, H. (2016). Increasing the order of convergence for

iterative methods to solve nonlinear systems. Calcolo, 53(3):285–300. Yun, B. I. (2008). A non-iterative method for solving non-linear equations. Applied Mathe- matics and Computation, 198(2):691–699. Yun, B. I. and Petković, M. S. (2009). Iterative methods based on the signum function approach for solving nonlinear equations. Numerical Algorithms, 52(4):649–662. Zachary, J. L. (1996). Introduction to scientific programming: computational problem solving using Maple and C. Springer Science & Business Media. Zafar, F., Cordero, A., and Torregrosa, J. R. (2020). A family of optimal fourth-order methods for multiple roots of nonlinear equations. Mathematical Methods in the Applied Sciences, 43(14):7869–7884. Zeng, Z. (2005). Computing multiple roots of inexact polynomials. Mathematics of Compu- tation, 74(250):869–903. Zheng, Q., Li, J., and Huang, F. (2011). An optimal steffensen-type family for solving non- linear equations. Applied Mathematics and Computation, 217(23):9592–9597. Zhou, X., Chen, X., and Song, Y. (2011). Constructing higher-order methods for obtaining the multiple roots of nonlinear equations. Journal of Computational and Applied Mathematics, 235(14):4199–4206. . . . . 2.2 5.5 Methods f5 (x) P M1 P M2 P M3 H P M S H M H M Methods f7 (x) P M1 P M2 P M3 H P M S H M H M Methods f10 (x) P M1 P M2 P M3 H P M S H M H M S M1 S M2 AM C M K M P M1 P M2 P M3 P M4 S M1 S M2 AM C M K M P M1 P M2 P M3 P M4 S M1 S M2 AM C M K M P M1 P M2 P M3 P M4 S M1 S M2 AM C M K M P M1 P M2 P M3 P M4 LM1 LM2 S B1 S B2 S M1 S M2 S M3 Z M1 Z M2 Z M3 P M1 P M2 P M3 LM1 LM2 S B1 S B2 S M1 S M2 S M3 Z M1 Z M2 Z M3 P M1 P M2 P M3 LM1 LM2 S B1 S B2 S M1 S M2 S M3 Z M1 Z M2 Z M3 P M1 P M2 P M3 LM1 LM2 S B1 S B2 S M1 S M2 S M3 Z M1 Z M2 Z M3 P M1 P M2 P M3 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 55 56 57 58 59 60 61 62 63 64 65 67 68 69 70 71 72 73 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 133 134 135 136 137 138 139 140 141 143 144 145 146 147 148 149 150 151 153 154 155 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171