A study of numerical methods for nonlinear problems and their applications

A Thesis

submitted in fulfillment of the requirements
for the award of the degree of

Doctor of Philosophy

in

Mathematics

Submitted by

Litika Rani

(Regd. No. 902011014)

under the guidance of

Dr. Munish Kansal

(Associate Professor)

Department of Mathematics

Thapar Institute of Engineering and Technology

(Deemed to be University)

Patiala 147004, India

May 2026

Certificate

This hereby certify that the work, which is being presented in the thesis, entitled **"A study of numerical methods for nonlinear problems and their applications"**, in fulfillment of the requirement for the award of the degree of **Doctor of Philosophy** in the Department of Mathematics, Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the guidance of **Dr. Munish Kansal**.

It is certified that matter presented in this thesis has not been submitted for the award of any degree at this or any other university.

Litika Rani

Litika Rani

Reg. No. 902011014

It is certified that the above statement, made by the candidate is correct to the best of my knowledge.

Munish Kansal

Dr. Munish Kansal

Associate Professor

Department of Mathematics

Thapar Institute of Engineering and Technology

Patiala 147004, Punjab, India

Dated: 19 May, 2026

Declaration

I hereby declare that thesis entitled **"A study of numerical methods for nonlinear problems and their applications"** submitted for the award of the degree of **Doctor of Philosophy** in the Department of Mathematics, Thapar Institute of Engineering and Technology, Patiala, is true and original record of my own work carried out under the supervision of **Dr. Munish Kansal**, Associate Professor at Department of Mathematics, Thapar Institute of Engineering and Technology, Patiala, India. The matter embodied in this thesis has not been submitted in part or full to any other university or institute for the award of any degree in India or abroad and that the ideas and references cited herein have been duly acknowledged.

Litika Rani

Ms. Litika Rani
Reg. No. 902011014

Dedicated to my parents,
for everything
beyond limits.

Contents

Acknowledgements

Life can be understood as a continuous process of growth marked by important milestones. Along this path, we are fortunate to encounter individuals whose guidance, encouragement, and presence make the journey meaningful. As I complete this significant academic endeavor, I take this opportunity to sincerely acknowledge and appreciate all those who contributed, directly or indirectly, to the successful completion of this research work.

First and foremost, I express my deep gratitude to the Almighty God for granting me strength, clarity of thought, and perseverance throughout this journey. In moments of uncertainty and difficulty, faith gave me resilience and direction. The successful completion of this work stands as a testament to divine grace that guided me from the initial idea to the final submission.

I extend my profound thanks to my supervisor, Dr. Munish Kansal, DOM, Thapar Institute of Engineering & Technology, Patiala, for his constant guidance, patience, and insightful suggestions. His belief in my potential often exceeded my own, seeing the researcher I could become long before I saw it myself. His inspiration was never loud; it was reflected in patient revisions, late-night emails, challenging questions that pushed me beyond my comfort zone, and the quiet reassurance that "this is just part of the process." His guidance strengthened my research skills and shaped my perspective toward professional integrity and humanity. I am deeply grateful for every lesson, every word of encouragement, and for never giving up on me.

I would also like to thank Dr. Deepak Gumber, Professor and Head, DOM, Thapar Institute of Engineering & Technology, Patiala, along with the former heads, Dr. Satish Kumar Sharma and Dr. Mahesh Kumar Sharma, for their administrative support and encouragement, which enabled me to explore diverse academic opportunities during my research tenure.

My sincere thanks to my doctoral committee members, Dr. Paramjeet Singh, Dr. Sanjeev Kumar, and Dr. Ajay Kumar (CSED), for their constructive feedback, thoughtful suggestions, and critical evaluation during progress reviews. I am equally grateful to Dr. Satish Kumar Sharma, Dr. A.K. Lal, and Dr. Harish Garg for their motivating words and positive encouragement during various academic interactions.

I would like to thank TIET for providing me a teaching associateship during the initial two years of my research. I am also grateful to the Council of Scientific and Industrial Research (CSIR), Government of India, for awarding me the CSIR SRF-DIRECT fellowship (File No. 09/0677(18752)/2024-EMR-I) and providing financial support for this research work.

Acknowledgements

I am deeply indebted to my father, Parmod Kumar Bansal, whose unwavering encouragement and belief in my abilities have always inspired me to aim higher. I express my heartfelt gratitude to my mother, Saroj Bansal, whose unconditional love, sacrifices, and prayers have been my greatest source of strength. Whatever I have achieved today is firmly rooted in the values and resilience instilled in me by my parents. I also extend my sincere appreciation to my brother, Umesh Bansal, who has always been my constant support system and closest companion. His encouragement, understanding, and shared moments of joy have made this journey lighter and more meaningful. I would also like to express my heartfelt thanks to my sister-in-law, Vibha Bansal, who has been much more than family; she has been a true friend. Her presence in my life has been a source of positivity and strength, and I am deeply grateful for the bond we share.

I am fortunate to have supportive friends who stood by me during challenges and celebrated my achievements. I thank Dr. Manpreet Kaur for their guidance and support in the initial stage of this journey. I also express my appreciation to close friends Shagun Kaushal, Princy Garg, Ritika, Ramneek Kaur, Rahul Sachdeva, Sarishti Sharma, V. K. Singh, Nancy Phanghal, Huny, Raman Duddi, Anshika Sharma, Jaspreet Kaur, and Priya Garg for their encouragement, meaningful discussions, and companionship. The time we spent together, especially during informal breaks and evening tea, created memories that I will always cherish.

It is not possible to mention every individual who contributed to my growth during this Ph.D. journey; however, I remain sincerely thankful to each one of them. This achievement is a collective outcome of their support, guidance, and goodwill. I extend my heartfelt gratitude to all.

Place: Patiala
Date: 19 May, 2026

Litika Rani

Litika Rani

Abstract

Computational methods have been effectively used to tackle real-world problems like global positioning systems, fluid flow, control systems, chemical reactions, computational economics, and biological and physical phenomena with the advent of modern high-speed digital electronic computers. These applications demonstrate the growing dependence on computational approaches in solving complex practical problems. Because of its practical aspect, many look for solutions that are "good enough," but defining what "good enough" means is more difficult. Determining the required accuracy therefore becomes a central topic in computational problem solving. The methodology and techniques for addressing scientific and engineering challenges have experienced significant transformations. The increasing complexity of the nonlinear problems under study largely drives these changes. This complexity arises from the complex structures analyzed in the mathematical modeling of various real-world instances. As a result, exact analytical solutions are often difficult or impossible to obtain. Simultaneously, approximate numerical solutions to problems are often sufficient. In many cases, such approximations provide results that are both accurate and computationally efficient. An approximation of the solution is necessary, precise to a specified number of decimal places or within a defined tolerance. Various numerical techniques for solving a specific problem generate a sequence of approximations that converge to the desired solution. The quality of these approximations must be evaluated in relation to the problem's requirements. But "good enough" depends on the application area.

The primary objectives include finding the best possible approximate solutions for scalar or systems of nonlinear equations, along with higher-degree polynomials, transcendental functions, extensive multidimensional systems, and both ordinary and partial differential equations. Secondly, it addresses the calculation of the matrix sign function, which is subsequently utilized to compute solutions for nonlinear matrix equations. This function plays an important role in extending numerical techniques to matrix-based problems. The matrix sign function is an axiomatic study of the intuitive concepts that govern the matrix function of non-singular square matrices, which are fundamental concepts in matrix theory. Therefore, performing these computations requires strong theoretical knowledge. The present study requires the knowledge of not only the original problem but also the derivation, error analysis, and performance limits of the numerical methods used to

solve it. This research develops new optimized iterative methods for solving nonlinear scalar, vectorial, and matrix equations, aiming for high convergence orders with minimal computational cost and adhering to theoretical rigor to avoid common numerical pitfalls. This emphasis is particularly important when numerical methods are applied to practical problems. As a fact, some algorithms do not work for real-world problems. Although numerical techniques have improved greatly, some limitations still exist.

Comprehensive numerical experiments consistently validate the theoretical developments. This computational experience is useful, as it connects exact arithmetic theory with finite-precision computation. At the same time, studying dynamical behavior through basins of attraction proves that the proposed schemes are better, efficient, and useful in real-world computing instances.

To present these results in a systematic manner, the thesis is organized as follows:

Chapter 1 provides an introduction and developments in the theory related to nonlinear scalar, vector, and matrix equations. We present a thorough literature review of algorithms for nonlinear problems. The applications of these algorithms for computing the matrix sign function are studied as well. Furthermore, we review basic mathematical concepts that drive the study of nonlinear solvers and the existence of the matrix sign function.

Chapter 2 deals with developing a higher-order optimal family of Chebyshev–Halley type methods to solve a univariate nonlinear equation with multiple roots. The proposed scheme considers weight functions that are selected adequately to optimize the convergence order and demands only four functional evaluations at each iteration. An extensive convergence analysis is also provided that demonstrates the establishment of eighth-order convergence for the developed scheme. As a result, the efficiency index of the proposed scheme is optimal according to the Kung-Traub conjecture. Finally, the theoretical findings are verified through numerical experiments that include real-life and nonlinear academic problems. In addition, the dynamical study of iterative schemes reflects a comprehensive overview of their stability, convergence properties, and graphical aspects by drawing attraction basins in the complex plane.

Chapter 3 aims to develop and analyze a new derivative-free class of higher-order iterative methods for locating multiple roots numerically. The scheme is generated by using King's type iterative method. By employing the Traub-Steffensen technique, a derivative-free family is proposed, which requires three functional evaluations to achieve optimal fourth-order convergence. Moreover, it can be observed that the theoretical convergence results of the family are symmetrical for different multiplicities of roots. Finally, a wide variety of nonlinear problems are included to confirm the

applicability and effectiveness of the proposed methods.

Chapter 4 deals with the development of multi-step vectorial iterative schemes for solving nonlinear systems, achieving fourth and sixth-order convergence. The proposed methods are designed to minimize computational costs by employing a single inverse operator and reducing the number of functional evaluations per iteration. Furthermore, the proposed three-step scheme is generalized into a $(q+1)$-step family, increasing the convergence order to $2q+2$. While standard local convergence analysis based on Taylor series expansion is common, but it has limitations, as it requires the use of higher-order derivatives. To overcome this limitation, the theoretical analysis in a Banach space setting is conducted that relies solely on first-order derivatives. The existence of a unique solution is guaranteed within a specific domain, whose radius of convergence is formally obtained using Lipschitz constants. Furthermore, the theoretical results are tested on several numerical examples to check the performance and stability of the proposed methods in comparison to existing counterparts.

In chapter 5, some new iterative families that utilizes a derivative-free approach, requiring two frozen divided differences and one matrix inversion per iteration are proposed. Additionally, a general multi-point family using a fifth-order scheme as a predictor is derived. The proposed families are analyzed both theoretically and numerically, and experimentation is conducted using a range of numerical problems to confirm the theoretical results on convergence order and computational efficiency. The dynamical properties of each iterative method are studied by means of basins of attraction.

In chapter 6, a novel matrix iteration for numerically computing the matrix sign function is proposed. An extensive convergence analysis is carried out in the matrix case to achieve fourth-order convergence. The basins of attraction are illustrated to demonstrate the global convergence behavior of the proposed method. Analytically, it is shown that the presented scheme is asymptotically stable. Furthermore, theoretical developments are numerically verified by discussing the clustering of eigenvalues around ± 1. Moreover, a class of numerical examples for matrices of various dimensions is assessed to exhibit the efficacy of the proposed method.

In chapter 7, the primary objective is to develop two novel iterative schemes for computing the sign of a matrix that has no pure imaginary eigenvalues. Detailed convergence analysis and asymptotic stability of the proposed methods are discussed. It is shown that the new schemes converge globally by drawing attraction basins. In addition, the obtained results are extended to compute non-trivial solutions of the Yang-Baxter-like equation, provided the given matrix has no

eigenvalues on the imaginary axis. To illustrate the effectiveness of theoretical developments, a class of numerical examples for matrices of various dimensions, including eigenvalue clustering, is worked out.

Chapter 8 discusses the summary and future aspects of the presented work.

List of Publications

- Litika Rani and Munish Kansal, "*An optimal derivative-free King's family for multiple zeros and its dynamics*", Engineering Computations (Emerald), 39(6), 2367–2390, 2022.
  SCIE, Impact Factor: 1.9
  DOI: 10.1108/EC-08-2021-0449

- Litika Rani, Fazlollah Soleymani, Munish Kansal, and Hemant Kumar Nashine, "*An optimized Chebyshev–Halley type family of multiple solvers: Extensive analysis and applications*", Mathematical Methods in the Applied Sciences (Wiley), 48(7), 7037–7055, 2025.
  SCIE, Impact Factor: 1.8
  DOI: 10.1002/mma.8699

- Litika Rani and Munish Kansal, "*Numerically stable iterative methods for computing matrix sign function*", Mathematical Methods in the Applied Sciences (Wiley), 46(8), 8596–8617, 2023.
  SCIE, Impact Factor: 1.8
  DOI: 10.1002/mma.9004

- Munish Kansal and Litika Rani, "*Globally convergent iterative scheme for computing matrix sign function with numerical stability*", Mathematical Methods in the Applied Sciences (Wiley), 48(7), 7580–7594, 2025.
  SCIE, Impact Factor: 1.8
  DOI: 10.1002/mma.9212

- Munish Kansal and Litika Rani, "*An adaptive Steffensen-like families for solving nonlinear systems using frozen divided differences*", Journal of Mathematical Chemistry (Springer), 62, 109–144, 2024.
  SCIE, Impact Factor: 2.0
  DOI: 10.1007/s10910-023-01524-1

- Munish Kansal and Litika Rani, "*A family of multi-step vectorial iterative methods for solving nonlinear systems*", Journal of Mathematical Chemistry (Springer), 64, Article ID 8, 2026.

SCIE, Impact Factor: 2.0

DOI: 10.1007/s10910-025-01770-5

List of Figures

List of Tables

List of Notations

$\mathbb{N}$	the set of natural numbers,
$\hat{\mathbb{N}}_0$	the set of natural numbers including zero,
$\mathbb{R}$	the set of real numbers,
$\mathbb{C}$	the set of complex numbers,
$\mathbb{R}^+$	the set of positive real numbers,
$\mathbb{R}^-$	the set of negative real numbers,
$\mathbb{C}^+$	the set of positive complex numbers,
$\mathbb{C}^-$	the set of negative complex numbers,
$\mathbb{R}^k$	the set of all real k-tuple vectors,
$\mathbb{C}^k$	the set of all complex k-tuple vectors,
$\mathbb{R}^{k\times l}$	the set of all real matrices of order $k \times l$,
$\mathbb{C}^{k\times l}$	the set of all complex matrices of order $k \times l$,
$\mathbb{R}^{k\times k}$	the set of all real square matrices of order k,
$\mathbb{C}^{k\times k}$	the set of all complex square matrices of order k,
T	Transpose,
$\mathbb{D}$	the open interval that is subset of $\mathbb{R}$,
f	the nonlinear function defined on the domain $\mathbb{D}$,
$f^i(x)$	the i^{th} order derivative of the function $f(x)$,
m	the multiplicity of the root,
η	the multiple zero of the function $f(x)$ of multiplicity $m \geq 1$,
n	the iteration number,
x_n	the approximation of the root at the n^{th} iteration,
$f(x_n)$	the approximation of the function $f(x)$ at the n^{th} iteration,
ψ	the rational map defined on extended complex plane (or the Riemann sphere),
$D_1(\alpha^*)$	the attraction basins which is the collection of those initial points z_0 such that their orbits converge to fixed point α^*,
M/P	mean value of iterations required by the algorithm per point,

NC(%)	the percentage of non-convergent points,
M_C/C	mean value of iterations per convergent point,
$[y, x; F]$	the first-order finite difference operator,
$\mathcal{D}$	the open convex domain that is subset of $\mathbb{R}^k$,
F	the nonlinear function defined on the domain $\mathcal{D}$,
F'	the first-order Fréchet derivative (or Jacobian matrix of order k),
$[F'(x)]^{-1}$	the inverse of first-order Fréchet derivative $F'(x)$,
Θ	the domain which is subset of $\mathbb{C}^{m\times m}$,
$\mathcal{F}$	the analytic (or holomorphic) function defined on domain Θ,
$\mathrm{sign}(A)$	the matrix sign function of a matrix A,
S	the matrix sign function or $S = \mathrm{sign}(A)$,
$\mathcal{B}_1$	Banach space
$\mathcal{B}_2$	Banach space
$\mathbb{S}$	Convex domain which is a subset of $\mathcal{B}_1$
$\mathbb{F}$	Field
ρ	Approximated order of convergence via actual solution
ρ_c	Approximated order of convergence via functions
$\hat{\rho}$	Approximated order of convergence via three consecutive iterations
I	Identity matrix
I_p	Identity matrix of order p

List of Abbreviations

NT	Number of iterations
CEI	Computational efficiency index
DD	Divided difference
FE	Functional evaluation
EF	Elementary function
OOC	Order of convergence
ACOC	Approximated computational order of convergence
JCF	Jordan Canonical form
MSF	Matrix sign function
MSR	Matrix square root
MMM	matrix-matrix multiplications
e-time	Elapsed time
IVPs	Initial value problems
BVPs	Boundary value problems
ODEs	Ordinary differential equations
PDEs	Partial differential equations
LCA	Local convergence analysis
SLCA	Semilocal convergence analysis
LTID	Linear time-invariant dynamic
BIBO	Bounded-input bounded output
HPD	Hermitian positive definite
GM	Geometric mean
AGM	Arithmetic-geometric mean
AFP	Attracting fixed point
SFP	Super attracting fixed point
REF	Repelling fixed point

Chapter 1

Introduction

1.1 General introduction

Computational mathematics plays a fundamental role in modern science and engineering by providing effective tools for computing the numerical and symbolic solutions to mathematical models arising in real-world applications. There are several computational aspects in physics, chemistry, economics, fluid dynamics, statistics, functional equations, differential equations, number theory, numerical linear algebra, and many other parts of applied mathematics that lead to equations for which closed-form analytical solutions are either unknown or impossible to obtain. Therefore, one of the most significant and fascinating areas of computational mathematics is to find accurate and efficient numerical solutions to the nonlinear problems presented below:

(i) Scalar nonlinear equations:

$$f(x) = 0, \tag{1.1}$$

where $f : \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$ is a nonlinear sufficiently differentiable function on the open interval $\mathbb{D}$. In general, the solution η to a problem (1.1) is determined such that $f(\eta) = 0$. A root η of the equation (1.1) is called a multiple root of multiplicity m if $f^{(m)}(\eta) \neq 0$ and $f^{(l)}(\eta) = 0$, for $l = 0, 1, \ldots, m-1$. The root with multiplicity $m = 1$ is called a simple root of an equation (1.1).

(ii) Systems of nonlinear equations:

$$F(X) = 0, \tag{1.2}$$

where $X = (X_1, X_2, \ldots, X_k)^T$ is an unknown vector and $F : \mathcal{D} \subseteq \mathbb{R}^k \to \mathbb{R}^k$ is a continuously differentiable nonlinear function defined on a convex set $\mathcal{D}$ as $(F_1(X), F_2(X), \ldots, F_k(X))^T$.

The goal is to find a vector X^* such that $F(X^*) = 0$. When $k = 1$, the equation (1.2) reduces to the aforementioned case, i.e., the scalar nonlinear equation (1.1).

(iii) Nonlinear matrix equations:

$$\mathcal{F}(\mathcal{X}) = 0, \tag{1.3}$$

where $\mathcal{F} : \Theta \subseteq \mathbb{C}^{m\times m} \to \mathbb{C}^{m\times m}$ is an analytic matrix function on some domain Θ and $\mathcal{X} = [x_{ij}], 1 \leq i, j \leq m$ is a $m \times m$ matrix that is to be determined. This definition is formulated to adequately expand the concept of a function of a scalar variable $f(z)$, where $z \in \mathbb{C}$.

These three classes of nonlinear problems constitute the primary focus of many researchers from different areas of science and engineering. In practice, exact analytical techniques are often inadequate for solving the complete mathematical formulation of these problems. These include higher-degree polynomial equations [242, 243, 244, 289], transcendental equations [29, 66, 271], nonlinear systems of equations [229, 263], integral equations [28, 252], ordinary and partial differential equations [30, 31, 32, 239, 268], IVP/BVPs [254, 255, 294, 354], linear systems [185, 205, 281] and nonlinear matrix equations [54, 154, 222, 343, 344]. For instance, solving the Van der Pol oscillator of the form:

$$\begin{cases} \dfrac{d^2u}{d\tau^2} - \mu(u^2-1)\dfrac{du}{d\tau} + u = 0,\ \mu > 0, \\ u(0) = 0,\ u(2) = 1, \end{cases} \tag{1.4}$$

using the discretization technique based on central finite-difference formulae leads to an $(l-1)$ - dimensional nonlinear system:

$$2h^2X_s + h\mu(X_s^2 - 1)(X_{s-1} - X_{s+1}) + 2(X_{s+1} + X_{s-1} - 2X_s) = 0,\ s = 1, 2, \ldots, l-1.$$

The system governing the current flow in a vacuum tube is obtained by partitioning the interval $[0, 2]$ with a step size $h = \frac{2}{l}$. Moreover, the nonlinear wave interactions and viscous flow can be governed by a PDE, also called Burger's equation [289], having a mathematical form:

$$\begin{cases} \dfrac{\partial^2 u}{\partial l^2} - \dfrac{\partial u}{\partial r} + u\dfrac{\partial u}{\partial l} + R(l, r) = 0,\ (l, r) \in [0, 1] \times [0, 1], \\ u(0, r) = u(1, r) = 0, \\ u(l, 0) = 10l(l-1), \quad u(l, 1) = \dfrac{10l(l-1)}{e}, \end{cases} \tag{1.5}$$

where $R(l,r) = -10\left[10l(1-3l+2l^2)+e^r(2-l+l^2)\right]/e^{2r}$ is a source term, and the solution is constrained by mixed boundary conditions. The finite difference discretization leads to a large nonlinear systems that requires numerical iteration for its solution.

In addition to such nonlinear systems, an important class of problems consists of nonlinear matrix equations, where the unknown variable is a matrix and the nonlinearity is induced by matrix products. A representative example is the Yang-Baxter-like matrix equation [112, 116] of the form:

$$A\mathcal{X}A = \mathcal{X}A\mathcal{X}, \tag{1.6}$$

which arises in quantum group theory and knot theory. The nonlinear and matrix-valued nature of the equation (1.6) makes analytical solution techniques impractical. Moreover, direct methods are applicable only to restricted classes of equations (see [6]). As a result, indirect methods have become necessary for obtaining approximate solutions that are close to exact solutions to complex mathematical models.

Under suitable conditions, numerical algorithms generate sequences of approximations that converge to the desired solution. In real-life models, engineers and scientists use a wider range of numerical methods to work with large-scale and complex mathematical models. These techniques use the algebraic and structural properties of the governing equations to create fast and accurate ways to solve a wide range of problems. Typical examples include the problem of studying the flow of current [303], the three-body problem in a gravitational field [251], the problem of eigenvalue clustering [162], the problem of calculating the potential field produced by a specific distribution of electric charge [303], the problem of computing the solution of algebraic Riccati equations [110, 319, 320], the problem of computing invariant subspaces [36], the chemical equilibrium relations combined with mass balance [76, 149], the computation of generalized inverses [342] and geometric mean of Hermitian matrices [167]. Such methodologies also appear naturally in many engineering applications, including linear-time invariant dynamic (LTID) control systems [64], such as the blood glucose control for diabetes patients [267], the three-state continuous stirred tank reactor models [284], the maleic anhydride production in fluidized bed reactors [208], the tenth-order transfer functions [163], and the pneumatic conveying batch dryers [288], and the BIBO stability [123].

Generally, the scope and effectiveness of numerical techniques have expanded considerably due to the rapid development of digital computers. Since computer systems represent one of the most important ways to solve complex mathematical models in fields such as physics [289], chemistry

[69], mechanics [261], economics [66], and biology [4]. As a result, the design and implementation of modern numerical algorithms have undergone a significant transformation.

Newton's method is perhaps the most frequently used technique for scalar problems, as it can be extended from scalar to other variants for solving nonlinear matrix equations or systems of nonlinear equations. The method is characterized by its simple structure and faster convergence. For scalar nonlinear equations, systems of nonlinear equations, or nonlinear matrix equations, computing the derivative of a function or the inverse of a matrix can be too costly or difficult to obtain. As a result, Newton's method and other derivative-based methods have become insufficient. Consequently, the study increasingly concentrates on eliminating its derivatives. The secant method, analogous to Newton's method and free of derivatives, has been developed and continues to captivate academics worldwide. Newton's method is indeed of quadratic order, requiring several iterations to achieve high accuracy, even when the solutions are less precise. Given this, derivative-free methods are formulated to address nonlinear problems. Researchers also emphasize a higher convergence order, resulting in fewer iterations. Even though they require a lot of calculations, computer softwares like MATLAB and MATHEMATICA have made the process easier. These tools make it easy to perform both symbolic and numerical calculations, break large tasks into smaller ones, and help students to understand the concept better by showing them in graphs, which would be hard to do manually.

Nevertheless, one must carefully consider the significant concepts presented in the preceding problems. The key principles of discretization, conditioning, numerical error, and algorithmic stability significantly impact the theoretical structure of any numerical problem. Therefore, when selecting a method, it is essential to evaluate algorithms based on several factors, including accuracy, the number of operations required, and efficiency.

The primary objective of this thesis is to discuss and incorporate our research on the development of efficient numerical algorithms for solving nonlinear problems. We are particularly interested in developing methods that have high-order convergence, reduced computational cost, low execution time, high accuracy, larger basins of attraction, and a simple structure.

1.2 Fundamental concepts of nonlinear problems

We begin by recalling basic definitions and results on iterative methods for solving nonlinear problems, thereby making this chapter self-contained. The notation defined below is used globally for

the rest of this work.

1.2.1 Jacobian matrix

The Jacobian matrix provides the matrix representation for the first-order Fréchet derivative $F'(X)$. Its entries are the first-order partial derivatives of the component functions of $F(X)$, given by

$$F'(X) = \begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \frac{\partial F_1}{\partial X_2} & \cdots & \frac{\partial F_1}{\partial X_k} \\ \frac{\partial F_2}{\partial X_1} & \frac{\partial F_2}{\partial X_2} & \cdots & \frac{\partial F_2}{\partial X_k} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial F_k}{\partial X_1} & \frac{\partial F_k}{\partial X_2} & \cdots & \frac{\partial F_k}{\partial X_k} \end{bmatrix}.$$

1.2.2 First-order divided difference

For a multivariable function F, the first-order divided difference is expressed as a bounded linear operator $[\cdot, \cdot; F] : \mathcal{D} \times \mathcal{D} \subseteq \mathbb{R}^k \times \mathbb{R}^k \to L(\mathbb{R}^k)$, where $\mathcal{D}$ is a convex subset of $\mathbb{R}^k$. According to Ortega [263], this operator is defined by the relation

$$[Y, X; F](Y - X) = F(Y) - F(X), \quad \forall\ (X, Y) \in \mathcal{D} \times \mathcal{D}, \quad X \neq Y. \tag{1.7}$$

Potra and Pták [274] observed that for $k > 1$, the definition above does not uniquely specify the divided difference. When F is differentiable, an integral representation of the divided difference can be given as

$$[X + h, X; F] = \int_0^1 F'(X + th)\, dt, \quad \forall\ (X, X + h) \in \mathcal{D} \times \mathcal{D}, \tag{1.8}$$

where $h = Y - X$. In this formulation, the operator $[\cdot, \cdot; F]$ corresponds to a $k \times k$ matrix.

1.2.3 Order of convergence

Consider a sequence $\left\{X^{(n)}\right\}_{n=0}^{\infty}$ in $\mathbb{R}^k$ generated by an iterative method and converging to a solution X^* of the system $F(X) = 0$. The rate of convergence is classified as follows:

1. Linear convergence: The sequence converges linearly if there exists a constant $M \in (0, 1)$ and an integer $n_0 \geq 1$ such that for all $n \geq n_0$,

$$\|X^{(n+1)} - X^*\| \leq M \|X^{(n)} - X^*\|. \tag{1.9}$$

2. Convergence of order p: For $p > 1$, the sequence is said to converge with order p if there exists a constant $M > 0$ and an integer $n_0 \geq 1$ such that for all $n \geq n_0$,

$$\|X^{(n+1)} - X^*\| \leq M\|X^{(n)} - X^*\|^p. \tag{1.10}$$

Specific cases are termed as quadratic and cubic convergence when $p = 2$ and 3, respectively.

1.2.4 Error equation

Define the error at the n^{th} iteration by $\xi^{(n)} = X^{(n)} - X^*$, where $\{X^{(n)}\}$ is the sequence produced by the iterative method and X^* is the exact solution. Then, the error equation of an iterative method can be expressed as

$$\xi^{(n+1)} = L\Big(\xi^{(n)}\Big)^p + O\bigg(\Big(\xi^{(n)}\Big)^{p+1}\bigg), \tag{1.11}$$

where $L \in \mathcal{L}(\mathbb{R}^k \times \cdots \times \mathbb{R}^k, \mathbb{R}^k)$ is a p-linear operator, and p denotes the order of convergence. Recall that a p-linear function is linear separately in each of its p arguments.

1.2.5 Computational order of convergence

Let $X^{(n-1)}$, $X^{(n)}$, and $X^{(n+1)}$ be three consecutive iterates near the solution X^* of the system $F(X) = 0$. The computational order of convergence (COC) [363] can be approximated as

$$\rho \approx \frac{\log\Big(\|X^{(n+1)} - X^*\| \;/\; \|X^{(n)} - X^*\|\Big)}{\log\left(\|X^{(n)} - X^*\| \;/\; \|X^{(n-1)} - X^*\|\right)}, \quad n = 1, 2, \ldots. \tag{1.12}$$

The COC is used either to verify the actual convergence order of an iterative method or to measure how closely it matches the theoretical order in practice. A major limitation of this formula is its dependence on the exact solution X^*, which is generally unknown. To circumvent this, the following alternative approximation, proposed by Jay [176], is often employed:

$$\rho_c \approx \frac{\log\Big(\|F(X^{(n+1)})\| \;/\; \|F(X^{(n)})\|\Big)}{\log\left(\|F(X^{(n)})\| \;/\; \|F(X^{(n-1)})\|\right)}, \quad n = 1, 2, \ldots. \tag{1.13}$$

Here, $X^{(n+1)}$, $X^{(n)}$, and $X^{(n-1)}$ are three consecutive iterations close to the zero X^* of the system $F(X)$. Another way to avoid using the value of unknown zero X^* is with help of approximated computational order of convergence (ACOC), which was introduced by Cordero and Torregrosa

[101] and is given by

$$\hat{\rho} \approx \frac{\log\left(\|X^{(n+1)} - X^{(n)}\| \,/\, \|X^{(n)} - X^{(n-1)}\|\right)}{\log\left(\|X^{(n)} - X^{(n-1)}\| \,/\, \|X^{(n-1)} - X^{(n-2)}\|\right)}, \quad n = 2, \ldots. \tag{1.14}$$

This expression estimates the convergence order using only the computed iterates, avoiding any need for the exact solution.

1.2.6 Efficiency index

The computational efficiency of an iterative method (see [265]) is measured with the help of efficiency index

$$E = p^{\frac{1}{\theta}}, \tag{1.15}$$

where p is the order of convergence and θ is the number of functional evaluations per iteration, which includes evaluation of functions and its derivatives. To compare various iterative techniques, a tool called the computational efficiency index is used.

The definition (1.15) has been modified for systems of nonlinear equations because while dealing with nonlinear systems, the total operational or computational cost per iteration is the sum of the evaluations of functions and the derivatives involved per step. In 2011, Grau-Sánchez et al. [142] developed the idea of estimating the computational cost when solving a system of nonlinear equations. An iterative approach with convergence order ρ has the computational efficiency index (CEI) as given by

$$CEI = p^{1/C(\mu_0,\mu_1,k,l)}. \tag{1.16}$$

Here, the computational cost per iteration is given by

$$C(\mu_0, \mu_1, k, l) = P_0\mu_0 k + P_1\mu_1 k^2 + P(k, l), \tag{1.17}$$

where P_0 and P_1 depict the number of evaluations of $F(X)$ and $F'(X)$, respectively. For computation of F, one needs to evaluate k scalar functions, whereas the number of scalar evaluations is k^2 for any new derivative F'. Here, $P(k, l)$ is the number of products and quotients needed per iteration, l is the ratio between quotients and products, $\mu_0 > 0$ and $\mu_1 > 0$ are the ratios between products and function evaluations of F and F', respectively.

Additionally, we must take into account the computational cost associated with calculating a

matrix inverse. The LU decomposition entails $k(k-1)/2$ quotients and $k(1-2k)(1-k)/6$ products, prompting us to resolve a linear system rather than compute the inverse operator. Furthermore, solving two triangular linear systems requires k quotients and $(k-1)k$ products. We also consider k products when performing matrix-vector (or scalar) multiplication or when multiplying a vector by a scalar. Furthermore, we examine a quotient that is comparable to the l_q products.

However, in case of derivative-free iterative methods, the computing cost at each iteration for a k-dimensional nonlinear system is evaluated through the following expression:

$$\mathcal{C}(\mu, k, l_q) = \mu\mathcal{A}(k) + \mathcal{P}(k, l_q), \tag{1.18}$$

where $\mathcal{A}(k)$ denotes the count of scalar function evaluations (FEs) in F and $[x, y; F]$. However, $\mu > 0$ is the ratio between the products and FEs of F. The function $\mathcal{P}(k, l_q)$ denotes the count of products or quotients required at each iterative step. In order to examine (1.18), we should evaluate each possible factor that contributes to the total computational cost. For instance, to evaluate F in an iterative technique, we consider k scalar functions and consequently $2(k-1)k$ scalar functions for each DD, i.e., $[X, Y; F]$ (see [132]), where $F(X)$ and $F(Y)$ are computed separately. Moreover, we take into account the computational effort needed to reckon a matrix inverse. As a result, we solve a linear system instead of computing the inverse operator since the LU decomposition has $\frac{k(k-1)}{2}$ quotients and $\frac{k(k-1)(2k-1)}{6}$ products. Furthermore, k quotients and k^2 products are needed for resolving two triangular linear systems. Additionally, for multiplication of the matrix by a vector, k^2 products and vector by a scalar, k products are counted. We consider a quotient to be comparable to l_q products.

In order to estimate the values in each problem, we expressed the cost of every elementary function in product units in Table 1.1, and accordingly, the value of is approximately 2.6, which directly depends on the software, computer system, and the arithmetic used [127]. All the computational work has been performed in the programming package MATHEMATICA 11 or 13 [365] on a 64-bit machine with an Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz and Microsoft Windows 11 as the operating system.

Table 1.1: Time elapsed and evaluation of EFs computing cost, where $y = \sqrt{5}$ and $x = \sqrt{3} - 1$

EFs	$x * y$	x/y	$\sqrt{x}$	e^x	$\ln(x)$	$\sin(x)$	$\cos(x)$	$\cos^{-1}(x)$	$\tan^{-1}(x)$
CPU time	0.0125	0.0296	0.0109	0.3437	0.2813	0.4844	0.4687	0.3906	0.3594
Cost	1	2.375	0.875	27.5	22.5	38.75	37.5	31.25	28.75

Remark 1.2.1. *Throughout this work, the computational cost of iterative methods is evaluated following the approach of Grau-Sánchez et al. [142] (Case 1), which provides a systematic comparison.*

1.2.7 Fixed point iteration

A function $\psi : \mathcal{D} \subseteq \mathbb{R}^k \to \mathbb{R}^k$ is said to have a fixed point at $X^* \in \mathcal{D}$ if

$$\psi(X^*) = X^*. \tag{1.19}$$

The problem of solving a nonlinear system $F(X) = 0$ is connected to the fixed-point formulation. Indeed, one can construct a continuous mapping $G : \mathbb{R}^k \to \mathbb{R}^k$ defined on a closed set $\mathcal{D} \subseteq \mathbb{R}^k$ such that $G(\mathcal{D}) \subseteq \mathcal{D}$ and G possesses a fixed point X^* that coincides with a solution of $F(X) = 0$. One natural construction is $G(X) = X - F(X)$. It should be noted that this representation is not unique; many different G may yield the same solution X^*. Conversely, if G admits a fixed point X^*, then the function defined by $F(X) = X - G(X)$ has X^* as a zero. This equivalence provides the use of fixed-point iteration to solve nonlinear equations. The iterative procedure is given by

$$X^{(n+1)} = G\Big(X^{(n)}\Big), \qquad n \in \hat{\mathbb{N}}_0, \tag{1.20}$$

where G is referred to as the iteration function.

1.2.8 Classification of iterative methods

Traub [351] classified iterative methods into two main categories, namely one-point and multi-point methods, which can be further subdivided into two subclasses based on their formulation and computational features.

1. *One-point iterative method without memory*: In this class of methods, the next iterate $X^{(n+1)}$ depends only on the current approximation $X^{(n)}$, and no previously computed information is reused. Accordingly, the iteration can be written as

$$X^{(n+1)} = G\Big(X^{(n)}\Big), \quad n \in \hat{\mathbb{N}}_0, \tag{1.21}$$

where G is referred to as a one-point iteration function without memory.

One-point iterative method with memory: Here, the new approximation $X^{(n+1)}$ is obtained

using both the present iterate $X^{(n)}$ and a finite number of earlier iterates $X^{(n-1)}, \ldots, X^{(n-s)}$. The corresponding iterative process is given by

$$X^{(n+1)} = G\Big(X^{(n)}; X^{(n-1)}, \ldots, X^{(n-s)}\Big), \quad n \in \mathbb{N}, \tag{1.22}$$

where G denotes a one-point iteration function with memory.

2. *Multi-point iterative method without memory*: In this category, the next iterate $X^{(n+1)}$ is computed using only newly generated quantities based on the current approximation $X^{(n)}$, namely $w_1(X^{(n)}), \ldots, w_q(X^{(n)})$ with $q \geq 1$, and no previously stored information is utilized. The iterative process is expressed as

$$X^{(n+1)} = G\Big(X^{(n)}, w_1(X^{(n)}), \ldots, w_q(X^{(n)})\Big), \quad n \in \hat{\mathbb{N}}_0, \tag{1.23}$$

where G is referred to as a multi-point iteration function without memory.

Multi-point iterative method with memory: In this case, the new approximation $X^{(n+1)}$ is obtained from a set of newly generated values $Z^{(n)} = \{X^{(n)}, w_1(X^{(n)}), \ldots, w_q(X^{(n)})\}$ together with a finite number of previously computed sets $Z^{(n-1)}, \ldots, Z^{(n-s)}$. Accordingly, the iteration is given by

$$X^{(n+1)} = G\Big(Z^{(n)}; Z^{(n-1)}, \ldots, Z^{(n-s)}\Big), \quad n \in \mathbb{N}, \tag{1.24}$$

where G denotes a multi-point iteration function with memory.

1.2.9 Taylor's theorem

Now, we define the following hypothesis, based on Taylor's expansion of vector functions (see [263]), to examine the convergence analysis of an iterative scheme.

Lemma 1.2.1. *Suppose $F : \mathcal{D} \subseteq \mathbb{R}^k \to \mathbb{R}^k$ is a Fréchet differentiable function on a open convex subset $\mathcal{D}$ of $\mathbb{R}^k$. Then,*

$$F(X+\xi) = F(X) + F'(X)\xi + \frac{1}{2!}F''(X)\xi^2 + \cdots + \frac{1}{(r-1)!}F^{(r-1)}(X)\xi^{r-1} + \mathcal{C}_r, \tag{1.25}$$

for all $X, \xi \in \mathbb{R}^k$, where $||\mathcal{C}_r|| \leq \frac{1}{r!}\sup_{0<t<1} ||F^{(r)}(X+t\xi)||\ ||\xi||^r$, and $\xi^r = (\xi, \xi, \overset{r\text{-}times}{\cdots}, \xi)$.

We outline a few notations and findings (see [89]) that must be satisfied by each iterative

scheme. Let F be r-times Fréchet differentiable function with $r \geq 1$. The r-linear function $F^{(r)}(u) : \mathbb{R}^k \times \ldots \times \mathbb{R}^k \to \mathbb{R}^k$ denotes the r^{th} derivative of F at $u \in \mathbb{R}^k$. It is simple to see that

(i) $F^{(r)}(u)(h_1, \ldots, h_r, \cdot) \in L(\mathbb{R}^k)$,

(ii) $F^{(r)}(u)(h_{\alpha(1)}, \ldots, h_{\alpha(r)}) = F^{(r)}(u)(h_1, \ldots, h_r)$, $\forall\ \alpha \in \{1, 2, \ldots, r\}$.

Based on these properties, we define

(i) $F^{(r)}(u)(h_1, \ldots, h_r) = F^{(r)}(u)h_1, \ldots, h_r$,

(ii) $F^{(r)}(u)h^{r-1}F^{(p)}h^p = F^{(r)}(u)F^{(p)}(u)h^{r+p-1}$.

1.3 Fundamental concepts in the Banach spaces

We now present an overview of the concepts above and notions in the context of Banach spaces.

1.3.1 Banach space

A Banach space is a complete normed vector space. More precisely, it is a vector space $\mathcal{B}$ over the field of real or complex numbers, equipped with a norm $\|\cdot\| : \mathcal{B} \to \mathbb{R}$ satisfying the following properties for all $x, y \in \mathcal{B}$ and any scalar α:

1. Positive definiteness: $\|x\| \geq 0$, and $\|x\| = 0$ if and only if $x = 0$.
2. Homogeneity: $\|\alpha x\| = |\alpha|\,\|x\|$.
3. Triangle inequality: $\|x + y\| \leq \|x\| + \|y\|$.

Additionally, the space must be complete with respect to the metric induced by the norm, i.e., every Cauchy sequence in $\mathcal{B}$ converges to an element within $\mathcal{B}$.

1.3.2 Linear operator

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be Banach spaces defined over an arbitrary scalar field $\mathbb{S}$. A mapping $F : \mathcal{B}_1 \to \mathcal{B}_2$ is termed additive if it preserves vector addition, i.e.,

$$F(x_1 + x_2) = F(x_1) + F(x_2) \quad \text{for all } x_1, x_2 \in \mathcal{B}_1. \tag{1.26}$$

It is called homogeneous if it respects scalar multiplication

$$F(sx) = sF(x) \quad \text{for all } x \in \mathcal{B}_1 \text{ and } s \in \mathbb{F}. \tag{1.27}$$

A mapping that is both additive and homogeneous is known as a linear operator.

Furthermore, we say a linear operator is bounded when the image of every bounded subset of $\mathcal{B}_1$ is itself bounded in $\mathcal{B}_2$. This property is fundamental in functional analysis, as it ensures continuity and facilitates many central results, including the classical uniform boundedness principle and the theory of linear functionals.

1.3.3 Fréchet derivative

The Fréchet derivative extends the classical notion of differentiation from real-valued functions to vector-valued operators defined on Banach spaces. Let F be an operator mapping on a Banach space $\mathcal{B}_1$ into another Banach space $\mathcal{B}_2$. We say that F is Fréchet-differentiable at a point $x_0 \in \mathcal{B}_1$ if there exists a bounded linear operator $L : \mathcal{B}_1 \to \mathcal{B}_2$ such that

$$\lim_{\|h\|\to 0} \frac{\|F(x_0 + h) - F(x_0) - Lh\|}{\|h\|} = 0. \tag{1.28}$$

This limit must hold uniformly regardless of the direction in which h approaches zero. The operator L is uniquely determined and is denoted by $F'(x_0)$, referred to as the first Fréchet derivative of F at x_0.

For any $x \in \mathcal{B}_1$ and sufficiently small $\delta x \in \mathcal{B}_1$, the Fréchet differential $\delta F(x) = F'(x)\,\delta x$, provides a local linear approximation to the increment $F(x + \delta x) - F(x)$, making it a useful tool for analyzing functional variation in infinite-dimensional settings.

1.3.4 Invertible operator

The existence of inverse operators in Banach spaces is characterized by the following fundamental result, often referred to as the Banach lemma on invertible operators.

Theorem 1.3.1. *Let F be a bounded linear operator on a Banach space $\mathcal{B}_1$. Then, the inverse operator F^{-1} exists if and only if there exists a bounded linear operator L on $\mathcal{B}_1$ such that L^{-1} exists and*

$$\|I - LF\| < 1.$$

Moreover, if F^{-1} exists, it can be expressed via the Neumann series

$$F^{-1} = \sum_{n=0}^{\infty} (I - LF)^n L, \tag{1.29}$$

and its norm satisfies the estimate

$$\|F^{-1}\| \leq \frac{\|L\|}{1 - \|I - LF\|}. \tag{1.30}$$

This theorem provides a constructive criterion for invertibility and is important in the analysis of linear operators, particularly in perturbation theory and iterative methods in functional analysis.

1.3.5 Local and semilocal analyses

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the two Banach spaces. Here, we posit that $F : \mathbb{S} \subseteq \mathcal{B}_1 \to \mathcal{B}_2$ is a nonlinear operator mapping on a convex, open, non-empty, subset $\mathbb{S}$ of $\mathcal{B}_1$ space. The task of approximating locally unique solutions to equations of the form $F(x) = 0$ using iterative methods naturally gives rise to four fundamental problems in numerical analysis.

The first problem is to demonstrate the well-defined nature of iterates. For instance, if the algorithm involves evaluating F at each x_n, it must be ensured that the iterates remain within the domain of F. While it's generally challenging to identify the precise set of initial data for which a process is well defined.

The second one is to address the sequences convergence that is produced by an algorithm and ascertain if their limit points actually represent solutions to $F(x) = 0$. Mainly two types of convergence results exist. The first, termed a local convergence analysis (LCA), assumes the existence of a particular solution x^* and asserts the existence of a neighborhood $\mathcal{U}$ of x^* such that, for all initial vectors in $\mathcal{U}$, the iterates converge to x^*. The second type, a semi-local convergence analysis (SLCA), does not require knowledge of a solution but guarantees convergence to some nearby solutions x^* from initial vectors satisfying specific, often stringent conditions. These theorems usually provide computable estimates for the error $x_n - x^*$, unlike local convergence theorems.

The third is to address the efficiency of overall operations, precisely the convergence speed for a given sequence. Two approaches correspond to LCA and SLCAs. The latter often yields error estimates, which may be reinterpreted as estimates of the sequence's convergence rate. However, these estimates tend to be overly pessimistic. The second approach explores the behavior of the sequence $\{x_n\}$ as n becomes large, particularly when x_n is near solutions x^*. This behavior is determined, to a first approximation, by the properties of the iteration function near x^* and leads to asymptotic rates of convergence.

The last one is to determine the optimal selection of an algorithm or software program for

solving a particular problem, along with specifying the conditions under which a given algorithm or method proves successful or unsuccessful.

More specifically, the problems defines that if the algorithm evaluates F at every x_n, it must be ensured that the iterates locate within the F domain. While it's generally challenging to identify the precise initial data set for a iterative procedure is well-defined, we focus on providing conditions that guarantee the well-defined nature of an iteration sequence for specific initial guesses.

1.4 Fundamental concepts of matrix sign function

We now present an overview of the concepts above and notions in the context of matrices.

1.4.1 Matrix function via Cauchy integral form

The matrix function $\mathcal{F}(A)$ of any given matrix $A \in \mathbb{C}^{m\times m}$ is generally defined by the Cauchy integral formula as follows:

$$\mathcal{F}(A) = \frac{1}{2\pi i}\oint_{\sigma} \mathcal{F}(\zeta)(\zeta I - A)^{-1}d\zeta, \tag{1.31}$$

where $\mathcal{F} : \Theta \subseteq \mathbb{C} \to \mathbb{C}$ is a holomorphic function in a closed loop σ that envelopes each eigenvalue of A (or simply the domain of $\mathcal{F}$ should contain these eigenvalues analytically) and I is an identity matrix of order $m \times m$.

1.4.2 Matrix function via Jordan canonical form

Let $\mathcal{F}$ be a function defined on the spectrum of a matrix $A \in \mathbb{C}^{m\times m}$. Suppose A admits the Jordan canonical form $A = UJU^{-1}$, such that $U^{-1}AU = J = \mathrm{diag}(J_1, J_2, \ldots, J_p)$. Then,

$$\mathcal{F}(A) = U\mathcal{F}(J)U^{-1} = U\mathrm{diag}(\mathcal{F}(J_1), \mathcal{F}(J_2), \ldots, \mathcal{F}(J_p))U^{-1}, \tag{1.32}$$

where

$$\mathcal{F}(J_q) := \begin{bmatrix} \mathcal{F}(\lambda_q) & \mathcal{F}'(\lambda_q) & \cdots & \dfrac{\mathcal{F}^{(m_q-1)}(\lambda_q)}{(m_q-1)!} \\ & \mathcal{F}(\lambda_q) & \ddots & \vdots \\ & & \ddots & \mathcal{F}'(\lambda_q) \\ & & & \mathcal{F}(\lambda_q) \end{bmatrix},$$

U is non-singular, $J_q = J_q(\lambda_q) \in \mathbb{C}^{m_q \times m_q}$ is a Jordan block and $\sum_{q=1}^{p} m_q = m$. The Jordan canonical form J of a matrix is determined uniquely, except for possible reordering of its constituent Jordan blocks. However, the similarity transformation matrix U that satisfies $A = UJU^{-1}$ is not unique. For each distinct eigenvalue λ_q of A, we define its index m_q as the size of the largest Jordan block associated with λ_q.

1.4.3 Matrix function via Hermite interpolation

Let $A \in \mathbb{C}^{m\times m}$ and suppose $\mathcal{F}$ is defined on the spectrum of A. Denote ψ be the minimal polynomial of A. We define $\mathcal{F}(A) := \mathcal{P}(A)$, where $\mathcal{P}$ is a polynomial of degree less than $\deg \psi = \sum_{i=1}^{s} m_i$ that satisfies the Hermite interpolation conditions:

$$\mathcal{P}^{(j)}(\lambda_i) = \mathcal{F}^{(j)}(\lambda_i), \quad j = 0, 1, \ldots, m_i - 1, \quad i = 1, 2, \ldots, s. \tag{1.33}$$

The polynomial $\mathcal{P}$ satisfying these conditions exists and is unique; it is called the Hermite interpolating polynomial.

Remark 1.4.1. *The three definitions of the matrix function are equivalent, provided that in the Cauchy integral definition, $\mathcal{F}$ is analytic in a domain containing the spectrum of the matrix. For more details, readers can refer [154]. However, in this work we majorly use the definition of matrix function via JCF.*

Remark 1.4.2. *A wide class of matrix functions can be defined using the notions introduced above. However, the present study is restricted to the matrix sign function (MSF). Accordingly, all subsequent definitions are defined specifically with respect to the MSF.*

1.4.4 Matrix sign function

It is known that in scalar case, the sign function is represented for each $\zeta \in \mathbb{C}$ not located on imaginary axis by

$$\operatorname{sign}(\zeta) = \begin{cases} +1, & \Re(\zeta) > 0, \\ -1, & \Re(\zeta) < 0. \end{cases} \tag{1.34}$$

where $\Re$ represents the real part of ζ. This definition can be extended to matrices via matrix function definition 1.4.2. Suppose that nonsingular matrix $A \in \mathbb{C}^{m\times m}$ has no eigenvalues on the imaginary axis. Then, the matrix sign function for a matrix A having JCF, $A = UJU^{-1}$ is defined

by

$$\operatorname{sign}(A) = U \begin{bmatrix} -I_p & 0 \\ 0 & I_{m-p} \end{bmatrix} U^{-1}. \tag{1.35}$$

1.4.5 Properties of matrix sign function

Let $A \in \mathbb{C}^{m\times m}$ has no eigenvalues on the imaginary axis and let $S = \operatorname{sign}(A)$. Then

(i) $S^2 = I$, i.e., S is involutory,

(ii) S is diagonalizable with eigenvalues ± 1,

(iii) $SA = AS$,

(iv) if A is real, then S is real,

(v) $(I-S)/2$ and $(I+S)/2$ are projections onto invariant subspaces that correspond to eigenvalues in the left and right half-plane, respectively.

1.4.6 Order of convergence

The order of convergence of a matrix iteration for computing the matrix sign function S is defined analogously to the vector case. Let $\{\mathcal{X}_n\}_{n=0}^{\infty}$ be a sequence of matrix iterates. The iteration is said to have order p if there exists a constant $M > 0$ such that

$$\|\mathcal{X}_{n+1} - S\| \leq M \|\mathcal{X}_n - S\|^p. \tag{1.36}$$

1.4.7 Asymptotic stability

Let $\{\mathcal{X}_n\}_{n=0}^{\infty}$ be a sequence generated by a matrix iteration for computing the matrix sign function S. The iteration is said to be asymptotically stable if $\lim_{n\to\infty} \mathcal{X}_n = S$, and for any sufficiently small perturbation $\mathcal{E}_0$ in the initial matrix $\mathcal{X}_0$, the corresponding perturbed sequence $\{\widetilde{\mathcal{X}}_n\}_{n=0}^{\infty}$ satisfies

$$\lim_{n\to\infty} \|\widetilde{\mathcal{X}}_n - \mathcal{X}_n\| = 0. \tag{1.37}$$

Remark 1.4.3. *It is worth distinguishing between stability and asymptotic stability. Stability ensures that the iterates remain close to the solution under sufficiently small perturbations of the initial matrix, whereas asymptotic stability additionally guarantees that the iterates converge to the*

solution as the iteration index tends to infinity.

1.4.8 Matrix sign decomposition

Suppose that matrix $A \in \mathbb{C}^{m\times m}$ has no eigenvalues on the imaginary axis. Then, the matrix sign decomposition is given by (see [155, 210])

$$A = SN = A(A^2)^{-1/2}(A^2)^{1/2}, \tag{1.38}$$

whereas $S = A(A^2)^{-1/2}$ is the matrix sign function which generalizes the scalar formula $\text{sign}(\zeta) = \zeta/(\zeta^2)^{1/2}$ and $1/2$ denotes the principal matrix square root of a given matrix.

1.5 Basins of attraction

Different iterative methods are analyzed from various perspectives. The investigation into the basins of attraction of scalar iterative methods is a current area of research. This study facilitates the classification of different iterative methods based on their order of convergence and aids in analyzing how these formulas perform relative to the chosen initial approximation. This approach not only allows for the selection of family members with superior stability properties but also enables the categorization of iterative methods of the same order according to their dynamic behavior. A concise recall of some basic ideas concerned with the visual tool is provided in the research article [358].

In order to determine the complex zeros of a complex polynomial $p(z)$, Cayley [71] employed the well-known Newton's method [351], given by

$$z_{n+1} = z_n - \frac{p(z_n)}{p'(z_n)}. \tag{1.39}$$

In fact, Cayley [71] discovered that if we know a function's zeros, we can check the behavior of different initial guesses through the Newton's approach. Further, in 1918 Julia [180] demonstrated that the answer lies in the theory of basins of attraction. To understand the concept more appropriately, assume the rational map $\psi : \mathbb{C} \to \mathbb{C}$ on extended complex plane (the Riemann sphere). Let $z_0 \in \mathbb{C}$ be any point, and we define its orbit like

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

Usually, we call a point $z_0 \in \mathbb{C}$ as periodic point with period q (where q is the least positive integer) if $\psi^q(z_0) = z_0$. Accordingly, when $q = 1$, a fixed point is named as a periodic point. If $|\psi'(z_0)| < 1$ then a complex point z_0 is attracting, repelling if $|\psi'(z_0)| > 1$, neutral if $|\psi'(z_0)| = 1$ and super attracting if $|\psi'(z_0)| = 0$. Consider α^* be a attracting fixed point of the rational function ψ. Concerning each attracting fixed point α^*, the attraction basins (denoted by $D_1(\alpha^*)$) is defined as the collection of those initial points z_0 such that their orbits tend to α^* as n increases, given by

$$D_1(\alpha^*) = \{z_0 \in \mathbb{C} : \psi^n(z_0) \to \alpha^* \text{ as } n \to \infty\}. \tag{1.40}$$

The collection of all points whose orbits are approaching attracting fixed point α^* is called the Fatou set. Its complementary set is called the Julia set. The Julia set represents the closure of the set where fixed points are repelling, which confirms the lines between the basins of the respective zeros. Therefore, the basins of attraction technique permits us to choose those initial points that converge to the desired root when we employ an iterative scheme on a polynomial. Moreover, we can visualize those initial points which are not suitable candidates for an iterative method.

In each example, we take the grid points 512×512 and choose $z_0 \in \mathbb{C}$ as the initial point in D, where $D = [-3, 3] \times [-3, 3]$ represents the rectangular domain in $\mathbb{C}$ such that it contains the desired root of an equation $p(z) = 0$. In addition, if the root is not enclosed in the domain D, we can enlarge the domain accordingly. When we draw basins, there are two possibilities that a scheme starts with initial point $z_0 \in D$ may or may not converge to the zero of a function $p(z)$. To estimate the attraction basins, we use the value 0.001 in the convergence-stopping rule, which is limited to 25 iterations. If the tolerance is not attained in the expected number of iterations, the scheme will be discarded, including any findings that show non-convergence in the iteration formula beginning with z_0. We allow one color to each initial guess z_0 as we form the attraction basins. The basins with the designated color are generated if the numerical approach converges for the initial point. Otherwise, it will be shaded black as the scheme diverges in the expected number of iterations.

For instance, consider $p_1(z) = (z^2-1)^2$ and $p_2 = (z^3-z)^3$ be the polynomials with multiplicity two and three, respectively. The basins of attraction for these two polynomials are shown in the Figure 1.1 using modified Newton's method. For the polynomial $p_1(z)$, if the initial guesses are converging to the -1 and 1, than it will be colored yellow and red, respectively. However, if the initial guess for the polynomial $p_2(z)$ converges to the roots $-1, 0$ and 1, it will be colored yellow, red and green, respectively. The required roots are denoted by white point in the attraction basins.

The dynamic nature of a rational map in the context of an vectorial iterative approach is also

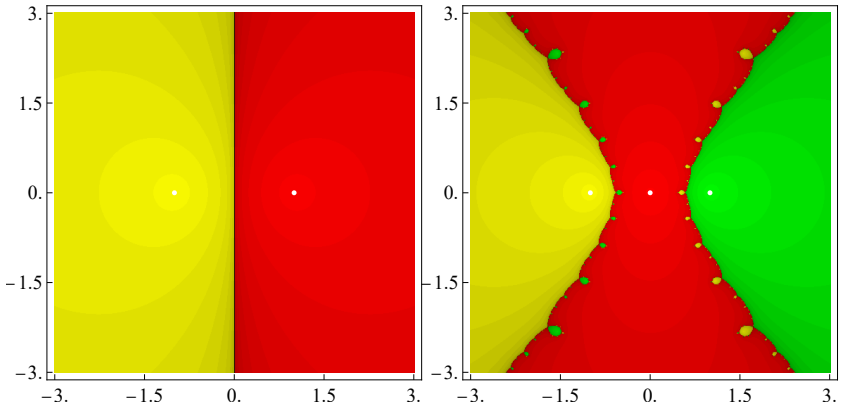


Figure 1.1: Basins of attraction of the polynomials $p_1(z)$ and $p_2(z)$.

important for understanding the convergence and stability of the method, which was first analyzed by Cordero et al. [103] who studied the behavior of several methods on particular polynomial systems. Let $\psi : \mathbb{R}^k \to \mathbb{R}^k$ denote the vector rational map. The orbit of a point $X^{(0)} \in \mathbb{R}^k$ represents the set containing successive images of $X^{(0)}$ via the map ψ, denoted as $\{X^{(0)}, \psi(X^{(0)}), \ldots, \psi^k(X^{(0)}), \ldots\}$. Further, the orbit can categorize the behavior of any point in $\mathbb{R}^k$ pivoting on its asymptotic behavior. A fixed point $X^* \in \mathbb{R}^k$ of ψ is a point such that $\psi(X^*) = X^*$. The stability of fixed points can be classified using the practical tool described in [85]. Consider X^* denotes the fixed point of ψ. Then, X^* is

(a) attracting fixed point (AFP), if $\left|\frac{\partial \psi_i(X^*)}{\partial X_j}\right| < \frac{1}{k}$,

(b) super attracting fixed point (SFP), if $\left|\frac{\partial \psi_i(X^*)}{\partial X_j}\right| = 0$,

(c) unstable or repelling fixed point (RFP) and belongs to the Julia set, if $\left|\frac{\partial \psi_i(X^*)}{\partial X_j}\right| > \frac{1}{k}$,

where $\psi_i(x)$ are the coordinate functions of the fixed point multivariate function ψ and $i, j \in \{1, 2, \ldots, k\}$. The following set of pre-images of arbitrary order

$$\mathcal{B}(X^*) = \{X^{(0)} \in \mathbb{R}^k : \psi^n(X^{(0)}) \to X^*,\ n \to \infty\}.$$

are referred to as the basin of attraction $\mathcal{B}(X^*)$, whereas $F(\psi)$ denotes the Fatou set containing points whose orbits are approaching the AFP, i.e., X^*. In contrast, the set consisting of the closure of RFPs, referred to as the Julia set, denoted by $J(\psi)$, apparently specifies the borders between the attraction basins. For instance,

Problem 1: Consider a nonlinear system

$$F_1(X_1, X_2) = \begin{cases} X_1^2 - 1 & = 0, \\ X_2^2 - 1 & = 0, \end{cases} \tag{1.41}$$

that has four exact solutions $X_1^* = (1, 1)^T$, $X_2^* = (1, -1)^T$, $X_3^* = (-1, 1)^T$, $X_4^* = (-1, -1)^T$, where yellow, green, blue, and red colors are allocated to each initial point that converges to the solutions X_1^*, X_2^*, X_3^* and X_4^*, respectively, as shown in Figure 1.2.

Problem 2: Let us suppose a nonlinear system

$$F_2(X_1, X_2) = \begin{cases} -X_1 + X_1^3 - 3X_1X_2^2 & = 0, \\ -X_2 + 3X_1^2X_2 - X_2^3 & = 0, \end{cases} \tag{1.42}$$

having three solutions $X_1^* = (-1, 0)^T$, $X_2^* = (0, 0)^T$, $X_3^* = (1, 0)^T$, where green, red, and yellow colors are allocated to each initial point that converges to the solutions X_1^*, X_2^*, and X_3^*, respectively, as shown in Figure 1.2.

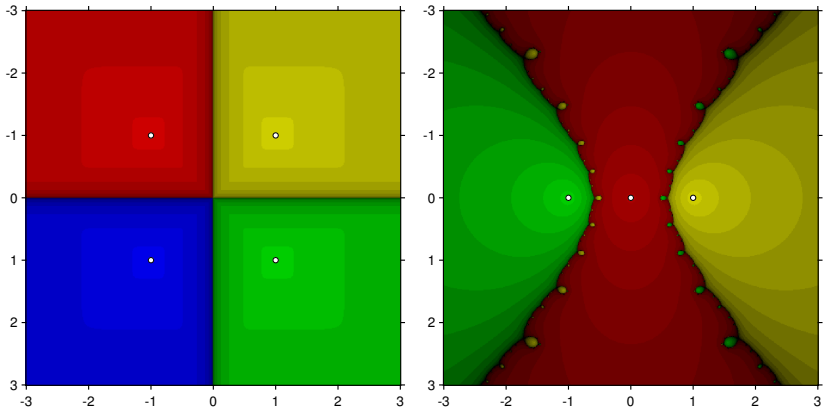


Figure 1.2: Basins of attraction of nonlinear systems $F_1(z)$ and $F_2(z)$ using Newton's method

1.6 Literature review

The study of complex real-world phenomena often depends on mathematical models expressed through nonlinear equations. The inherent nonlinearity of these models makes their analysis challenging, and finding their solutions is important for understanding the underlying physical processes. While analytical solutions are generally infeasible, iterative numerical methods provide a practical

alternative, yielding approximate solutions to nonlinear problems up to a desired accuracy.

The development of iterative methods for efficiently approximating the roots of three major classes of problems:

(i) simple or multiple roots of nonlinear equations,

(ii) solutions of system of nonlinear equations,

(iii) solutions of nonlinear matrix equations,

represents a rich and challenging area in numerical mathematics. For more than four centuries, these so-called root-finding or zero-finding problems have attracted considerable research interest, resulting in an extensive body of literature. Generally, two important classes of methods are used to solve such problems: direct and iterative methods. Direct methods yield the exact roots in a finite number of steps and often provide all roots simultaneously. However, their applicability is limited; for instance, no direct method exists for solving every polynomial equation of degree greater than four. Consequently, one usually goes to iterative methods.

The idea behind iterative techniques is to make successive approximations, starting with one or more initial guesses. A recurrence formula is used to make a series of approximations that, when the suitable conditions are met, converge to the desired root. The literature has proposed a vast number of such methods. Since exact solutions are rarely attainable, the goal of iterative methods is to produce an approximate root that lies within a prescribed tolerance. This practical requirement, together with the fact that no single iterative method works well for all types of problems, explains the considerable diversity of research in this field. For a comprehensive overview of the fundamental iterative schemes, the reader may consult standard books by Traub [351], Ostrowski [265], Ortega and Rheinboldt [263], Petković et al. [268], Argyros and Magreñán [22], and Higham [154].

The design of iterative schemes generally focuses on a few, such as good convergence properties, computational efficiency, numerical stability, robustness, and low algorithmic complexity. However, the convergence of these methods depends on factors such as the initial estimates, the behavior of the function and its derivatives near the root, and the presence of a bracketing interval for the root.

1.6.1 Literature review on the scalar nonlinear equations

We begin with the simplest problem, i.e., finding a simple root of a scalar nonlinear equation $f(x) = 0$. Classical methods for this task include the bisection method, the regula-falsi method, the secant method, Newton's method, and its variants. The bisection and regula-falsi methods

are globally convergent but only linearly convergent, with an efficiency index of 1. However, they cannot handle the roots of even multiplicity. The secant method converges superlinearly, while Newton's method converges quadratically.

Most iterative methods can be derived geometrically [16, 77, 203, 265, 351] by approximating the function near the root with a simple curve (e.g., a straight line, parabola, hyperbola, circle, or ellipse). Other common derivation techniques include compositional techniques [213, 214], weight-function approaches [133], inverse interpolation [268, 351], sampling methods [79, 174, 175, 258], Adomian decomposition [2, 3, 7, 8, 9, 78, 268], quadrature formulas [245], power-mean approaches [201], and functional approaches [204, 300]. For example, Newton's method [351] can be obtained by fitting the tangent line $y(x) - f(x_n) = f'(x_n)(x - x_n)$, and setting $y(x_{n+1}) = 0$, which yields the iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0. \tag{1.43}$$

This is perhaps the most classical root-finding algorithm. It requires two function evaluations per step (one of f and one of f') and has an efficiency index of $\sqrt{2} \approx 1.414$. Nevertheless, Newton's method has several well-known drawbacks, as it requires the evaluation of the first derivative at every step, which may be expensive or impractical. Its convergence depends critically on a correct initial guess since a poor choice may lead to divergence or slow convergence. The method fails if the derivative $f'(x_n)$ equals zero near the root.

To address the first problem, Steffensen [345] proposed a derivative-free variant

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.44}$$

which replaces $f'(x_n)$ by a finite difference. This scheme retains quadratic convergence and is often referred to as a Steffensen-type method. To alleviate the second difficulty, Wu [367] introduced the family

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) - \lambda_1 f(x_n)}, \quad \lambda_1 \in \mathbb{R}, \quad n \in \hat{\mathbb{N}}_0. \tag{1.45}$$

This method converges quadratically even when $f'(x_n) = 0$ at some points, as long as $f'(x_n) - \lambda_1 f(x_n) \neq 0$. The classical Newton method corresponds to $\lambda_1 = 0$.

Several other quadratic families have been derived from different conic sections, e.g., the ellipse method of Gupta et al. [145] does not fail when $f'(x_n)$ is small or zero. Higher-order one-point iterative methods, such as Chebyshev's method, Halley's method, super-Halley method, and Ostrowski's

method, achieve cubic convergence at the cost of evaluating second derivatives. These are usually derived by fitting a quadratic curve (parabola or hyperbola) and imposing tangency conditions up to the second order. The need to compute second derivatives is often cumbersome for complicated functions, which has motivated the development of multi-point methods. Multi-point schemes reuse information from previous steps or from several points within the same step, thereby overcoming certain theoretical limitations of one-point iterations. According to the Kung-Traub conjecture [221], a multi-point method without memory that uses d function evaluations per iteration has an optimal order of 2^{d-1}.

Traub [351] introduced the third-order midpoint Newton method, and many other non-optimal methods [80, 113, 114, 157, 158, 159, 231, 240, 259, 260, 270, 273, 269, 353] have since been derived by approximating the integral in Newton's theorem with different quadrature rules (rectangular, trapezoidal, Simpson's, etc.). Although these methods are free of second derivatives, they are not optimal because they require three to four evaluations per step. Optimal fourth-order methods using three evaluations were first discovered by Ostrowski [265] and later generalized by Jarratt [174] and King [212] using the approximation to

$$f'(y_n) = \frac{f'(x_n)(f(x_n) + \gamma f(y_n))}{f'(x_n)(f(x_n) + \beta f(y_n))}. \tag{1.46}$$

By substituting this approximation for $f'(y_n)$ with $\gamma = \beta - 2$ into the double Newton method formula, we obtain the well-known King's family of iterative methods, which converges quartically, as follows:

$$x_{n+1} = y_n - \frac{f(y_n)\Big(f(x_n) + \beta f(y_n)\Big)}{f'(x_n)\Big(f(x_n) + (\beta - 2) f(y_n)\Big)}, \quad n \in \hat{\mathbb{N}}_0. \tag{1.47}$$

In pursuit of this, a mean-based approach on a function's second derivative in the Chebyshev-Halley family, Kansal et al. [191] constructed iterative methods with optimal fourth-order convergence. More recently, optimal fourth-order, eighth-order, and sixteenth-order multi-point methods have been proposed by Argyros et al. [21], Chang et al. [72], Cordero et al. [94], Kansal et al. [193], Kaur et al. [206], and Kumar et al. [218], with efficiency indices $\sqrt[3]{4} \approx 1.587$, $\sqrt[4]{8} \approx 1.682$ and $\sqrt[5]{16} \approx 1.741$, respectively. All these methods, however, share the same locality drawbacks as Newton's method, converging only if the initial guess is sufficiently close to the root.

The problem of finding multiple roots (multiplicity $m > 1$) presents additional challenges. Bracketing methods fail for roots of even multiplicity because the function does not change sign.

The first classical approach is given by Rall's [277], i.e., the modified Newton method

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.48}$$

which converges quadratically for locating multiple roots of $f(x) = (x-\eta)^m g(x),\ g(\eta) \neq 0$ when m is known. This scheme (1.48) can be obtained by applying Newton's method to the function $\hat{f}(x) = \sqrt[m]{f(x)}$. With the same transformation, a third-order Laguerre family [268] of iterative methods has been obtained by Bodewig [62] for computing multiple roots. Then, the well-known Hansen-Patrick, Euler-Cauchy, Halley, and Ostrowski-like methods can be retrieved as a special case from the Laguerre family. But the presence of first and second-order derivatives makes it computationally expensive.

To get rid of derivatives, Traub-Steffensen [351] employed the following approximation:

$$f'(x_n) \simeq \frac{f(x_n + \gamma f(x_n)) - f(x_n)}{\gamma f(x_n)},\ \gamma \in \mathbb{R}\backslash\{0\}, \tag{1.49}$$

where the derivative is replaced with the first-order divided difference approximation (1.49) in the modified Newton method (1.48), and one arrives at

$$\begin{cases} \mu_n &= x_n + \gamma f(x_n), \\ x_{n+1} &= x_n - m\dfrac{f(x_n)}{f[\mu_n, x_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{cases} \tag{1.50}$$

known as the modified Traub-Steffensen method, where $f[\mu_n, x_n] = \frac{f(\mu_n)-f(x_n)}{\mu_n - x_n}$ is the first-order divided difference. By maintaining second-order convergence, the approach (1.50) improves upon the modified Newton's method significantly and is derivative-free. Further, Kansal et al. [186] proposed the generalized optimal derivative-free scheme, given by

$$x_{n+1} = x_n - m\frac{(1-a)f(\mu_n) + af(x_n)}{f[\mu_n, x_n]}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.51}$$

where $\mu_n = x_n + \gamma f(x_n)$ and γ is any real number. When $a = 1$, the above scheme reduces to scheme (1.48). Using the weight function approach, Arora et al. [24] have proposed another optimal one-point iterative family with quadratic order of convergence.

Subsequently, several variations of high-order schemes (1.48) have been proposed and examined. In research papers [262, 351], a third-order Chebyshev approach for multiple zeros was

proposed. For polynomial zeros, a thorough local convergence study of this approach has been provided in [172]. One can find further variations of Chebyshev's technique for multiple zeros at [259]. Furthermore, the modified Halley's approach for multiple zeros was proposed by Obreshkov in [262] and later by Hansen and Patrick in [148]. In addition to error estimates and an estimate of the asymptotic error constant of Halley's approach for multiple roots, the theoretical result of [172] provides explicit bounds on the convergence domain. The scheme for multiple roots was introduced by Ostrowski in [265], and subsequently Osada [264] and Chun and Neta in [80] proposed the third-order iterative methods.

Several versions of the modified Newton's approach have been developed and examined in the literature to approximate multiple zeros of nonlinear functions. Using derivatives, many researchers, such as Cordero et al. [94] and Zafar et al. [376], extended the modified Newton's technique (1.48) for multiple roots. Higher-order variants, such as Halley-type, Chebyshev-type [259], and Jaratt-type methods [260], have been developed, but they usually require second derivatives. Derivative-free two-point methods of third and fourth order have also been proposed by Kanwar et al. [202], Wang and Liu [362], Yun [374], though optimal fourth-order methods for multiple roots remain relatively scarce.

Strikingly, the first optimal fourth-order technique for finding multiple roots was not introduced until 2009 by Shengguo et al. [314]. This development occurred nearly fifty years after optimal two-point methods for simple roots were established. A significant advancement followed in 2011 when Zhou et al. [383] presented a generalized, optimal two-point scheme that utilized a weight function. This work gained wide recognition for its straightforward structure and for encompassing numerous existing methods as particular examples, including the earlier work of Shengguo et al. [314] and the fourth-order techniques of Sharma and Sharma [311]. More recently, Zhou et al. [384] constructed another family of fourth-order methods, with each iteration demanding two function evaluations and one derivative evaluation. Separately, Sharma and Sharma [311] also developed an optimal fourth-order adaptation of Jarratt's classical method [175], specifically designed for multiple roots. For more, see the articles [51, 124, 165, 173, 217, 219, 220] and their references therein.

Furthermore, recent and some derivative-free higher-order multi-point techniques have been discussed by Bala et al. [40], Kansal et al. [186, 187, 190, 195], Sharma et al. [298, 309], Soleymani et al. [334], etc. These methods belong to a class of iterative solvers that require knowledge of the multiplicity. However, the exact value of this multiplicity may not be available in practice. In such circumstances, a very close approximation of multiplicity can be calculated by using the subsequent

forms given by

1. Traub [351] approximation formula:

$$m \approx \frac{\log |f(x)|}{\log \left| \dfrac{f(x)}{f'(x)} \right|}, \tag{1.52}$$

   as x approaches the multiple root of f in close proximity;

2. Lagouanelle [223] approximate formula:

$$m \approx \frac{f'(x)^2}{f'(x)^2 - f(x)f''(x)}, \tag{1.53}$$

   as x approaches the multiple root of f in close proximity.

When the multiplicity m is unknown, one may apply the transformation

$$\hat{G}(x) = \begin{cases} \dfrac{f(x)}{f'(x)}, & f'(x) \neq 0, \\ 0, & f'(x) = 0, \end{cases} \tag{1.54}$$

which converts a multiple root into a simple zero of $\hat{G}$. Applying Newton's method to $\hat{G}$ yields Schröder's method [272, 275, 292]

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{\{f'(x_n)\}^2 - f(x_n)f''(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.55}$$

which converges quadratically without knowledge of m. The need to compute second derivatives, however, makes this approach costly. References to discussions of Schröder's method (1.55) are provided by various authors, including Traub [351], McNamee [241], Dahlquist and Björck [106], Amat et al. [17], Gilbert [136], Scavo and Thoo [290], Ypma [372], and others. Soleymani et al. [328, 329] further implemented this transformation for handling multiple roots in the absence of multiplicity through iterative techniques. Then, in 2021, Cordero et al. [98] introduced a memory-based iterative scheme

$$x_{n+1} = x_n - \frac{\hat{G}(x_n)}{\hat{G}[x_{n-2}, x_n] - \hat{G}[x_{n-2}, x_{n-1}] + \hat{G}[x_{n-1}, x_n]}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.56}$$

for computing roots of unknown multiplicity using three initial approximations x_{n-2}, x_{n-1}, x_n. This

approach improves the efficiency of Schröder's derivative-free counterpart and is considered a base for developing higher-order methods with memory.

In later work, Cordero et al. [88] adapted Kurchatov's scheme to obtain derivative and derivative-free iterative methods with memory for multiple roots. The iteration that uses derivatives is given by

$$x_{n+1} = x_n - \frac{\hat{G}(x_n)}{\hat{G}[2x_n - x_{n-1}, x_{n-1}]}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.57}$$

while the derivative-free form is expressed as

$$x_{n+1} = x_n - \frac{\hat{g}(x_n)}{\hat{g}[2x_n - x_{n-1}, x_{n-1}]}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.58}$$

where $\hat{g}(x) = \frac{f(x)}{f[x+f(x),x]}$. These modifications preserve second-order convergence and exhibit favorable dynamical properties.

Since many practical problems involve more than one equation, the next section provides the literature on nonlinear systems of equations.

1.6.2 Literature review on the systems of nonlinear equations

We are now addressing the second class of problems, that is solving vectorial or multivariable problems. Usually, these cases are viewed as extensions of scalar nonlinear problems. However, it is important to note that not every iterative method derived from the scalar form can be extended. The development of computationally efficient algorithms for solving vectorial problems has motivated extensive research employing a range of techniques.

A classical algorithm for approximating solutions to the nonlinear system (1.2) is the quadratically convergent Newton method [351]:

$$X^{(n+1)} = X^{(n)} - \left[F'\left(X^{(n)}\right)\right]^{-1} F\left(X^{(n)}\right), \quad n \in \hat{\mathbb{N}}_0, \tag{1.59}$$

where $X^{(0)}$ is an initial guess and $\left[F'\left(X^{(n)}\right)\right]^{-1}$ denotes the inverse of the Fréchet derivative. Newton's method and its variants converge provided the Fréchet derivative $F'(X)$ remains non-singular and well-conditioned near the solution. In 1916, Fine [126] significantly opened a new chamber of research in numerical analysis and introduced the SLCA of Newton's method. Further, L.V. Kantorovich [263] pioneered the generalization of Newton's method, and explored its semi-local convergence analysis (SLCA) for operator equations in $\mathcal{B}_1$-spaces. In 1939, Kantorovich [200] provided the

fundamental convergence results for Newton's method and in 1948, utilizing recurrence relations, he established a result for Newton's method in $\mathcal{B}_1$-space, known as the Newton-Kantorovich theorem [198]. This theorem covers the error estimate, initial point, and the existence and uniqueness of the solution, though not in the explicit form. However, the explicit formulations of these error estimates were independently obtained by Ostrowski [265]; Gragg and Tapia [139]. Furthermore, Kantorovich [199] provided an alternative proof of the Newton-Kantorovich theorem (NKT) using the principle of majorizing sequences.

Increasing the convergence order from quadratic to cubic has led to various one-point methods. A notable example is the Chebyshev method of cubic convergence, introduced by Gutiérrez and Hernández [146]:

$$X^{(n+1)} = X^{(n)} - \Big(I + \tfrac{1}{2}L_F\big(X^{(n)}\big)\Big)\Big[F'\big(X^{(n)}\big)\Big]^{-1}F\big(X^{(n)}\big), \quad n \in \hat{\mathbb{N}}_0, \tag{1.60}$$

where $L_F\big(X^{(n)}\big) = \Big[F'\big(X^{(n)}\big)\Big]^{-1}F''\big(X^{(n)}\big)\Big[F'\big(X^{(n)}\big)\Big]^{-1}F\big(X^{(n)}\big)$, and I is the identity operator. This scheme explicitly depends on the first and second Fréchet derivatives. For a system of k equations, evaluating F' requires k^2 operations, while F'' demands $\frac{k^2(k+1)}{2}$ operations. Hence, third-order methods such as Halley's and Chebyshev's, despite their higher convergence, are often considered computationally impractical due to the high cost of evaluating the second derivative.

To circumvent the computation of F'', the two-point Newton method with frozen derivative [351] is frequently employed:

$$\begin{aligned} Y^{(n)} &= X^{(n)} - \Big[F'\big(X^{(n)}\big)\Big]^{-1}F\big(X^{(n)}\big), \\ X^{(n+1)} &= Y^{(n)} - \Big[F'\big(X^{(n)}\big)\Big]^{-1}F\big(Y^{(n)}\big), \quad n \in \hat{\mathbb{N}}_0. \end{aligned} \tag{1.61}$$

This method achieves a higher order without requiring extra derivative evaluations or matrix inversions, which makes it more efficient than the basic Newton iteration. To find faster algorithms, researchers have suggested and studied several higher-order multi-point methods that do not use second-order derivatives [250, 305, 307, 312, 338]. For instance, Cordero et al. [90, 91, 92, 95] proposed a technique that integrates an iterative method of order p with a modified variant of Newton's method. However, these kinds of compositional methods often need to evaluate extra nonlinear functions and their Jacobian matrices, which makes the calculations more expensive. As a result, the algorithms that come out of this may not work well or computationally less efficient.

In 2014, Frontini and Sormani [129] proposed a third-order method based on quadrature for-

mulas of order at least one. Subsequently, Cordero et al. [101] derived variants of Newton's method using fifth-order quadrature formulas to achieve third-order convergence. Further, some fourth-order methods were later developed by Nedzhibov [257]. Additionally, various novel iterative techniques have been explored, including those based on the Adomian decomposition approach, functional approach, sampling approach, inverse interpolation approach, and weight function approach, as documented in [7, 26, 89, 104, 166, 174, 204, 253, 351, 371].

A common feature among many iterative schemes is the need to compute Jacobian matrices at one or more points during each iteration. This requirement presents an important computational challenge, particularly when the Jacobian is undefined or in high-dimensional cases where its evaluation becomes more expensive or impractical. To avoid the explicit computation of derivatives, Traub [351] introduced the derivative-free scheme

$$\begin{cases} Y^{(n)} & = X^{(n)} + \beta F(X^{(n)}),\ \beta \in \mathbb{R}\backslash\{0\}, \\ X^{(n+1)} & = X^{(n)} - \left[Y^{(n)}, X^{(n)}; F\right]^{-1} F\left(X^{(n)}\right), \quad n \in \hat{\mathbb{N}}_0, \end{cases} \tag{1.62}$$

where $\left[Y^{(n)}, X^{(n)}; F\right]^{-1}$ is the inverse of the first-order divided difference of F. For $\beta = 1$, this reduces to Steffensen's method as presented by Samanskii [287], a derivative-free alternative that retains quadratic convergence. While both the Steffensen and Newton methods achieve quadratic convergence, studies indicate that Steffensen's scheme exhibits weaker numerical stability relative to Newton's method, with its performance being more sensitive to the choice of the initial approximation. This behavior has been examined in detail by [75, 100], where it was demonstrated for scalar equations that derivative-free iterative schemes gain stability when the auxiliary point is chosen as $y = x + \beta f(x)$ for sufficiently small real parameters β.

Nevertheless, replacing the exact Jacobian with divided-difference approximations may reduce the convergence order of certain iterative methods. One of the example is the multidimensional extension of Ostrowski's fourth-order method (see [97, 265]):

$$\begin{aligned} Y^{(n)} &= X^{(n)} - \left[F'\left(X^{(n)}\right)\right]^{-1} F\left(X^{(n)}\right), \\ X^{(n+1)} &= Y^{(n)} - \left(2\left[Y^{(n)}, X^{(n)}; F\right] - \left[F'\left(X^{(n)}\right)\right]\right)^{-1} F\left(Y^{(n)}\right), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{1.63}$$

attains only cubic convergence when the Fréchet derivative $F'(X)$ is substituted with the divided

difference $[Y, X; F]$, as shown below:

$$\begin{aligned} Y^{(n)} &= X^{(n)} - \Big[X^{(n)} + F\Big(X^{(n)}\Big), X^{(n)}; F\Big]^{-1} F\Big(X^{(n)}\Big), \\ X^{(n+1)} &= Y^{(n)} - \Big(2\Big[Y^{(n)}, X^{(n)}; F\Big] - \Big[X^{(n)} + F\Big(X^{(n)}\Big), X^{(n)}; F\Big]\Big)^{-1} F\Big(Y^{(n)}\Big), \quad n \in \hat{\mathbb{N}}_0. \end{aligned} \tag{1.64}$$

It is important to note that a similar loss of convergence order occurs with other fourth-order schemes as well, such as Jarratt's method [174], Sharma's scheme [307], Montazeri's algorithm [250], the vectorial extension of Ostrowski's method [97, 265], and the fifth-order method of Sharma and Arora [301]. In each case, the derivative-free variant converges at a lower rate. However, Amiri et al. [18] showed that employing a structured divided-difference operator of the form $[X, X + G(X); F]$, where $G(X) = \Big(F_1(X)^\alpha, F_2(X)^\alpha, \ldots, F_k(X)^\alpha\Big)^T$, $\alpha \in \mathbb{N}$, as an approximation of the Jacobian matrix, for a suitably chosen exponent α, preserve the original fourth-order convergence of these methods.

Although certain derivative-free modifications lead to a decrease in convergence order, it should be emphasized that iterative schemes exist that can be reformulated to retain their original order even when the exact Jacobian is eliminated. One such example is the family of methods presented by Behl et al. [50], which merges the Traub-Steffensen approach with a second step based on divided-difference operators. Following Amiri et al.'s [18] methodology with parameter $\alpha = 2$, the Jacobian matrix is replaced by a structured divided-difference operator, and some derivative-free variants have been obtained from existing schemes proposed by authors [1, 97, 265, 304]. In 2023, Singh et al. [324] introduced a parametric family of Jacobian-based iterative methods, given by

$$\begin{aligned} W^{(n)} &= X^{(n)} - \Big[F'(X^{(n)})\Big]^{-1} F(X^{(n)}), \\ X^{(n+1)} &= W^{(n)} - \left(p_1 + p_2 \frac{F(W^{(n)})^T F(W^{(n)})}{F(X^{(n)})^T F(X^{(n)})}\right) \Big[F'(W^{(n)})\Big]^{-1} F(W^{(n)}), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{1.65}$$

where the quotient $\frac{F(W^{(n)})^T F(W^{(n)})}{F(X^{(n)})^T F(X^{(n)})}$ acts as a scalar accelerator. The real parameters p_1 and p_2 govern the convergence order; the scheme attains fifth-order when $p_1 = p_2 = 1$, fourth-order when $p_1 = 1$ (with p_2 arbitrary), and second-order when $p_1 \neq 1$ (with p_2 arbitrary). Replacing the Jacobian matrices in (1.65) with suitable divided-difference operators yields the derivative-free family proposed by Cordero et al. [96] using $F'(X^{(n)}) \approx \Big[U_X^{(n)}, X^{(n)}; F\Big]$, $F'(Y^{(n)}) \approx \Big[U_Y^{(n)}, Y^{(n)}; F\Big]$, with $U_X^{(n)} = X^{(n)} + \alpha F(X^{(n)})$ and $U_Y^{(n)} = Y^{(n)} + \alpha F(Y^{(n)})$, $\alpha \in \mathbb{R}$.

In the last decade, many other iterative methods for solving nonlinear systems have been developed. These include approaches for systems originating from real-world applications such as chemotaxis models (e.g., Keller–Segel models [179, 234]) and neurophysiology models in medicine [11, 18, 108, 276, 285]. More recently, a high-order iterative scheme called the asynchronous parsimonious method was introduced for model identification by Sun et al. [347]. High-order iterative solvers can be particularly useful in this context for achieving accurate parameter estimation and model calibration.

Most existing two-step methods offer at most fourth-order convergence, while three-step methods generally reach up to eighth-order. However, some two-step fifth-order methods have been proposed by Singh et al. [324] and Cordero et al. [92], and three-step fifth-order methods by Sharma et al. [303] and Singh et al. [321]. Comparative studies have also been conducted by Singh and Sharma [321, 322], who evaluated fifth and seventh-order methods using efficiency indices, further Chun and Neta [81] compared sixth-order methods based on computational order and iteration cost; and Zhanlav et al. [381] performed a broader comparison of more than twenty methods of orders six to eight, utilizing metrics such as average of iteration count, CPU time, and divergence rate. Additional comparisons of higher-order three-step methods can be found in [48, 84, 237, 323, 380].

Recently, Behl et al. [48] introduced a sixth-order two-step method. They emphasized the practical importance of developing new two-step sixth-order methods, as these are typically simpler than three-step methods of the same order. The primary motivation behind such developments is to design schemes that achieve high convergence order while maintaining low computational cost. A key advantage of the methods proposed here is their higher order of convergence coupled with the use of only one matrix inversion per iteration, making them computationally efficient. Readers can see more iterative methods in the articles by Bhavna and Bhatia [58], Guo et al. [144], Kansal et al. [188, 189], Sharma and Arora [306], Shang et al. [295], Wang et al. [361], Yu and Wang [373] and their references therein.

As we have already discussed, one of the primary goals of the development of optimal techniques is their high computational efficiency. It is generally known that Kung and Traub [221, 351] established the optimality criterion conjecture for scalar equations (1.1). Nevertheless, this conjecture is not relevant, especially when dealing with nonlinear equation systems. In that instance, the following conjecture was put up by Arroyo et al. [25]:

The order of convergence of any memory-free Newton-type iterative method for solving nonlinear systems cannot be greater than $2^{d_1+d_2-1}$*, where* d_1 *is the number of functional evaluations*

in the Jacobian matrix entries per iteration and d_2 is the number of functional evaluations of the nonlinear function, where $d_1 \leq d_2$. In the vectorial sense, the approach is considered optimal if it achieves this bound.

In the sense of the above conjecture, the optimal method for solving nonlinear systems is only Newton's method. Due to which, Cordero et al. [99] generalizes the Ostrowski's, King's, Chun's, and KLAM families by proposing a first novel optimal class of efficient two-parameter optimal iterative techniques in a multidimensional setting, given by

$$\begin{aligned}
\nu_n &= \frac{\|F(Y^{(n)})\|^2}{\|F(X^{(n)})\|^2} = \frac{F(Y^{(n)})^T F(Y^{(n)})}{F(X^{(n)})^T F(X^{(n)})}, \\
K_n &= \frac{1}{1+\lambda\nu_n}, \\
p_n &= K_n(1+\psi\nu_n),\ q_n = 2K_n\nu_n, \\
Y^{(n)} &= X^{(n)} - \left[F'(X^{(n)})\right]^{-1} F(X^{(n)}), \\
X^{(n+1)} &= Y^{(n)} - \left[F'(X^{(n)})\right]^{-1} \left[p_n F(Y^{(n)}) + q_n F(X^{(n)})\right], \quad n \in \hat{\mathbb{N}}_0.
\end{aligned} \tag{1.66}$$

Further, Cordero et al. [102] presented another iterative technique for solving nonlinear systems that achieves eighth-order convergence and demonstrates its optimality. Recently, the Newton scheme was expanded by Singh et al. [323] to a weighted-Newton iteration, and the suggested scheme exhibits the optimal fourth order of convergence. Nowadays, many researchers are focusing on developing higher-order optimal methods for solving vectorial problems. However, due to local convergence limitations, some scholars are exploring global or semilocal convergence characteristics. Quasi-Newton methods [263] offer global convergence, but at the cost of reducing the convergence order from quadratic to superlinear, thereby sacrificing one of the key advantages of iterative schemes. Another approach for improving convergence order and efficiency is the use of memorization. Iterative methods with memory can elevate the convergence order of their memory-free counterparts without requiring extra function or Jacobian evaluations. Such methods are broadly classified into two categories, i.e., Newton-type methods with memory [5, 299] and Steffensen-type methods with memory [183]. However, we will focus exclusively on methods without memory, so this discussion will be omitted.

In recent literature, various robust and efficient high-order iterative methods have been developed for solving nonlinear equations. However, a significant limitation of many such schemes is their inability to generalize effectively to multivariate problems. Because of this, researchers are

exploring additional ways to combine old methods with new algorithms designed for multivariable contexts. By addressing these problems, the field aims to enhance the usefulness of high-order methods for a wider range of mathematical problems.

1.6.3 Literature review on the matrix sign function

The first two classes of the nonlinear problems extend naturally to the matrix domain, where functions of matrices are important for solving higher-dimensional problems arising in systems theory, control, and scientific computing. Among these, the matrix sign function has emerged as a pivotal tool with deep theoretical connections and wide-ranging applications.

Matrix functions generalize scalar functions to the matrix setting. The MSF, in particular, has experienced substantial growth in interest due to advances in high-speed computing. In literature, the Schur method [154] is computationally intensive but numerically stable for matrix sign determination. But it is not recommended for practical use, as the Schur decomposition (SD) can often solve problems without explicitly computing the MSF. When a complex matrix $A \in \mathbb{C}^{m\times m}$ has an Schur decomposition, i.e., $A = QTQ^*$, then

$$\operatorname{sign}(A) = Q\operatorname{sign}(T)Q^*, \tag{1.67}$$

where T is an upper triangular and Q is a unitary matrix. This simplifies the problem to compute $\operatorname{sign}(T)$ as an upper triangular matrix whose diagonal elements are ± 1. The modern computational foundation of MSF was established by Roberts [283], who introduced Newton's iteration for computing the MSF:

$$\begin{cases} \mathcal{X}_0 = A, \\ \mathcal{X}_{n+1} = \dfrac{1}{2}(\mathcal{X}_n + \mathcal{X}_n^{-1}), \quad n \in \hat{\mathbb{N}}_0, \end{cases} \tag{1.68}$$

where A is a nonsingular matrix having no eigenvalues on the imaginary axis. This iteration is fundamental for model reduction and for solving algebraic matrix Riccati equations, as it decomposes a matrix into components whose spectra lie on opposite sides of the imaginary axis. Subsequently, Denman [110] and Benner [56] demonstrated the MSF's critical role in control theory, specifically for solving Lyapunov and Sylvester matrix equations. Its utility has been further cemented in various other applications, [226]. These problems include solving spectral factorizations, pole assignments, Lyapunov equations, coupled Riccati equations, MSRs, etc. More precisely, Gardiner and Laub

[131] proposed a generalized Newton sign iteration that has the form:

$$\mathcal{X}_{n+1} = \frac{1}{2}(\mathcal{X}_n + H\mathcal{X}_n^{-1}H), \quad n \in \hat{\mathbb{N}}_0, \tag{1.69}$$

with $\mathcal{X}_0 = A$ as the initial value. If H is nonsingular, it fundamentally performs the standard Newton iteration on $H^{-1}A$ and converges to $H\text{sign}(H^{-1}A)$. However, for singular H, the convergence is uncertain and may be linear. In 1994, Higham [155] pointed out the appeal of the concise representation of $\text{sign}(A) = A(A^2)^{-1/2}$, by employing extension to $\text{sign}(z) = z/(z^2)^{1/2}$ in scalar form. The computation of MSF involves an explicit matrix inverse, which is usually avoided through the Newton-Schultz iteration, i.e., replacing the inverse with the Schultz iteration [161]. Despite being more computationally expensive, this method only exhibits local convergence. In addition, by leveraging the storage format of the hierarchical $\mathcal{H}$-matrix, as described in [42, 140], the MSF can be modified to determine efficient solutions to low-rank problems. The MSF also possesses important theoretical links with other matrix operations, including polar decomposition [156], the matrix square root [154], and the matrix p^{th} root [168].

A significant application of the MSF is in spectral decomposition and the analysis of matrix pairs. Bai and Demmel [36, 37, 38] investigated perturbation theory, conditioning, and iterative refinement for the MSF, employing it to compute unitary matrices Z and Q that decouple a matrix pair (A, B) via a transformation of the form:

$$\begin{aligned} Q^H AZ &= \begin{pmatrix} Q_1^H \\ Q_2^H \end{pmatrix} A(Z_1, Z_2) = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}, \\ Q^H BZ &= \begin{pmatrix} Q_1^H \\ Q_2^H \end{pmatrix} B(Z_1, Z_2) = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}. \end{aligned} \tag{1.70}$$

This decoupling splits the eigenspectrum into smaller subproblems (A_{11}, B_{11}) and (A_{22}, B_{22}), which is essential for finding deflating subspaces in linear control systems [23, 55, 56, 59, 68, 109, 130, 131, 164, 232, 233, 246, 368]. While the recursive use of this technique provides partial spectral information, the QZ algorithm [248] remains the most stable method for complete eigenspectrum computation. However, the Newton iteration for the MSF is often computationally more efficient, relying on simpler building blocks, with high-performance software available for both serial and parallel architectures. Sun and Quintana-Qrtí [348] have proposed another generalized technique for determining the MSF.

To improve upon the quadratic convergence of Newton's method (1.68), families of higher-order iterations have been developed. A generic family was proposed by Kenney and Laub [210, 211] and later by Gomilko et al. [138], based on Padé approximants to the function $\mathcal{F}(\chi) = (1-\chi)^{-1/2}$, where $\chi = 1 - z^2$. Representing the sign function as $\operatorname{sign}(z) = z/(1-\chi)^{1/2}$, and letting $R_{s,l}(\chi)/T_{s,l}(\chi)$ denote the (s,l)-Padé approximant to $\mathcal{F}(\chi)$, the iteration

$$z_{n+1} = \frac{z_n R_{s,l}(1-z_n^2)}{T_{s,l}(1-z_n^2)} =: \psi_{2s+1,2l}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.71}$$

converges locally (when $s \geq l-1$) to ± 1 with order $l+s+1$. Iannazzo [168] later showed these iterations belong to the larger class known as the basic family [184] or König family [65]. Further improvements include the extension of Chebyshev-Halley type methods for global convergence by Cordero et al. [86]. Theoretical aspects, including partial fraction expansions [266] and the analysis of Padé recursions on triangular matrices [211], have also been thoroughly studied.

The MSF is deeply intertwined with other important matrix functions, forming a cohesive computational ecosystem. A key result by Higham [153] expresses the MSF of a block anti-diagonal matrix in terms of the matrix square root:

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}. \tag{1.72}$$

This relationship is part of a broader pattern. The unitary polar factor ($\operatorname{polar}(A)$), the geometric mean (GM) ($A\#B$) of two Hermitian positive definite (HPD) matrices, and MSF can all be expressed via MSR, like

$$\begin{cases} \operatorname{polar}(A) & = A(A^*A)^{-1/2}, \\ A\#B & = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}, \\ \operatorname{sign}(A) & = A(A^2)^{-1/2}, \end{cases} \tag{1.73}$$

respectively. Conversely, the MSF can compute these functions through elegant block matrix formulations:

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & \operatorname{polar}(A) \\ (\operatorname{polar}(A))^* & 0 \end{bmatrix}, \tag{1.74}$$

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ B^{-1} & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A\#B \\ (A\#B)^{-1} & 0 \end{bmatrix}. \tag{1.75}$$

These identities highlight the MSF's role as a unifying computational primitive.

The geometric mean of two HPD matrices, A and B, defined as $GM(A, B) := A(A^{-1}B)^{1/2}$ or equivalently $A\#B$, is a particularly important application. While the scalar arithmetic-geometric mean (AGM) iteration is simple, its direct matrix generalization is nontrivial. Bhatia [57] provided the standard definition $A\#B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}$, spurring significant research into its computation, often leveraging MSF or MSR iterations [137, 167, 330, 377]. Applications span adaptive filtering [366], surface wave problems [261], and other areas of numerical linear algebra [41, 119, 120, 235, 332, 378].

In 2011, Iannazzo and Meini [170] proved the uniqueness and existence of the $\mathcal{X}_*$ solution with non-unit spectral radius for the matrix equation $\varphi(\mathcal{X}) = 0$. They also showed that when $\rho(\mathcal{X}_*) < 1$, the matrix Laurent polynomial $L(z) = Q + Pz + Pz^{-1}$ has inversion within an annulus that encompasses the circle of unit radius. The inverse $\mathcal{H}(Z) = L^{-1}(z)$ can be expressed as an expansion of power series $\mathcal{H}(z) = H_0 + \sum_{i=1}^{\infty} H_i(z^{-i} + z^i)$. They demonstrated that for particular choices of the coefficients P and Q, namely $P = \frac{1}{4}(A - B)$ and $Q = \frac{1}{2}(A + B)$, H_0 is equal to one of the matrix functions, i.e., $A\#B$. So one can obtain MSF using the (1.75) form.

Soheili et al. [327] proposed a new iterative approach in matrix form for finding the MSF using six matrix-matrix products, given by the formula

$$\begin{cases} \mathcal{X}_0 & = A, \\ \mathcal{X}_{n+1} & = \mathcal{X}_n(10I + 98\mathcal{X}_n^2 + 126\mathcal{X}_n^4 + 22\mathcal{X}_n^6)[I + 42\mathcal{X}_n^2 + 140\mathcal{X}_n^4 + 70\mathcal{X}_n^6 + 3\mathcal{X}_n^8]^{-1}, \end{cases} \tag{1.76}$$

with applications to stochastic differential equations. The geometric mean of a and b (nonnegative numbers) is defined by iterating $a_{n+1} = \frac{1}{2}(a_n + b_n)$ and $b_{n+1} = \sqrt{a_n b_n}$ until $a_n = b_n$ to the desired precision. However, extending this iterative formula for general nonsingular matrices is not straightforward. In 2017, Cordero et al. [86] extended the Chebyshev–Halley type method into a matrix iteration for approximating the MSF and demonstrating global convergence. Further, in 2018, Kumar and Cardoso [215] have considered the iterative scheme as follows:

$$\mathcal{X}_{n+1} = \mathcal{X}_n^2(2\mathcal{X}_n - I)^{-1}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.77}$$

where the initial matrix $\mathcal{X}_0$ commutes with A and has no eigenvalues (say λ_i's) whose $\mathrm{Re}(\lambda_i) = 1/2$. This method works for both diagonalizable and non-diagonalizable matrices, and a thorough investigation of the generated sequence $\{\mathcal{X}_n\}_{n=0}^{\infty}$ through (1.77) is provided to analyze its numerical behavior. Thereafter, Song et al. [341] have considered again the notion of the geometric mean in a matrix case to highlight the application of MSF by developing matrix iteration

$$\mathcal{X}_{n+1} = 4\mathcal{X}_n(I + \mathcal{X}_n^2)[I + 6\mathcal{X}_n^2 + \mathcal{X}_n^4]^{-1}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.78}$$

In 2022, Dong and Wang [115] have proposed a generalized iterative method of arbitrary order p using the generalized formula

$$\frac{x_{n+1} - 1}{x_{n+1} + 1} = \frac{(x_n - 1)^p}{(x_n + 1)^p} \frac{(x_n - b)}{(-x_n - b)}, \; b \in \mathbb{R}, \quad n \in \hat{\mathbb{N}}_0, \tag{1.79}$$

such that many existing matrix iterations can be seen as special cases of this family. For instance, [249, 313, 327, 337].

In 2024, Kansal et al. [197] proposed an iterative method for computing generalized eigenvalues of regular matrix pencils in the context of MSF computation, expressed as

$$\mathcal{X}_{n+1} = \left[\mathcal{X}_n \left(21I + 50\mathcal{X}_n^2 + 9\mathcal{X}_n^4\right)\right]\left[4I + 45\mathcal{X}_n^2 + 30\mathcal{X}_n^4 + \mathcal{X}_n^6\right]^{-1}. \tag{1.80}$$

In addition, the Kansal et al. [196] used the MSF to determine invariant subspaces associated with eigenvalues lying within a chosen region of the complex plane for a square matrix. To achieve this, they extended a three-step iterative scheme into the following matrix iteration:

$$\mathcal{X}_{n+1} = \left[\mathcal{X}_n \left(25I + 30\mathcal{X}_n^2 + \mathcal{X}_n^4\right)\right]\left[6I + 40\mathcal{X}_n^2 + 10\mathcal{X}_n^4\right]^{-1}. \tag{1.81}$$

Recently, Jung et al. [181] extended the general weight-function approach to derive a multi-parameter family of three-point iterations for the matrix sign function. By imposing conditions for conformal conjugacy on quadratic polynomials, the resulting class generalizes the Padé family and yields several new methods. Using a similar approach, Wang et al. [359] have proposed sixth-order matrix iteration for computing S. For more information, one can see research articles [61, 70, 73, 115, 121, 122, 143, 147, 150, 160, 169, 182, 194, 227, 247, 286, 296, 331, 339].

The primary objective of this research is to extend scalar iterative methods to the matrix

setting for computing the matrix sign function S with global convergence properties. Accordingly, we propose new iterative schemes with convergence orders six and eight that do not belong to the Padé family, nor to any existing literature, or their reciprocals. The convergence behavior and stability of the proposed methods are established through analytical investigation.

In this thesis, we aim to develop new iterative methods that are computationally efficient, robust, and, wherever possible, free from higher-order derivatives. We shall construct families of multi-point methods for scalar equations involving multiple roots, nonlinear systems of equations, and nonlinear matrix equations involving the matrix sign function. While developing matrix iterations for the computation of the matrix sign function, particular attention is given to ensuring that the proposed methods are distinct from the Padé family [154], existing methods available in the literature, and their reciprocal forms. The performance of the proposed methods will be evaluated in terms of the number of iterations, computational order of convergence, and accuracy achieved for a fixed number of function evaluations.

1.7 Objectives and summary of the thesis

As high-performance computing becomes more common, it is even more important to make numerical methods better for solving nonlinear problems. Multi-point iterative methods are very effective here because they can converge faster than one-point methods. However, multi-point approaches sometimes converge better, but the result usually means that more function or derivative evaluations are needed, which can slow down computing, especially for complex systems. This trend has led to research on faster, more efficient multi-point algorithms. Also, a thorough study of their dynamic aspects is necessary to determine how stable and efficient these iterative processes are.

Taking all of these factors into account, this study aims to achieve the following main goals:

1. Construction and analysis of efficient iterative methods for solving nonlinear problems and its dynamics.
2. To develop and analyze iterative methods for nonlinear systems of equations and their applications.
3. To study and implement the globally convergent matrix iteration for computing matrix sign function along with its numerical stability.

This thesis comprises eight chapters, beginning with an introduction in Chapter One. The further contents of each chapter are summarized below:

In chapter 2, the authors have introduced a higher-order optimal family of Chebyshev–Halley type methods to solve a univariate nonlinear equation with multiple roots. The proposed scheme considers weight functions that are selected adequately to optimize the convergence order and demands only four functional evaluations at each iteration. An extensive convergence analysis is also provided that demonstrates the establishment of eighth-order convergence for the developed scheme. As a result, the efficiency index of the proposed scheme is optimal according to the Kung-Traub conjecture. Finally, the theoretical findings are verified through numerical experiments that include real-life and nonlinear academic problems. In addition, the dynamical study of iterative schemes reflects a comprehensive overview of their stability, convergence properties, and graphical aspects by drawing attraction basins in the complex plane.

In chapter 3, the authors aim to develop and analyze a new derivative-free class of higher-order iterative methods for locating multiple roots numerically. The scheme is generated by using King's type iterative method. By employing the Traub-Steffensen technique, a derivative-free family is proposed, which requires three functional evaluations to achieve optimal fourth-order convergence. Moreover, it can be observed that the theoretical convergence results of the family are symmetrical for different multiplicities of roots. This further motivates us to present the result in general, which confirms the convergence order of the proposed scheme. It is also worth mentioning that we can obtain already existing methods as particular cases of the proposed family for some suitable choice of free disposable parameters. Finally, we include a wide variety of nonlinear problems to confirm the applicability and effectiveness of the proposed methods.

In chapter 4, we develop multi-step vectorial iterative schemes for solving nonlinear systems, achieving fourth and sixth-order convergence. The proposed methods are designed to minimize computational costs by employing a single inverse operator and reducing the number of functional evaluations per iteration. Furthermore, we generalize the sixth-order three-step scheme into a $(q+1)$-step family, increasing the convergence order to $2q+2$. While standard local convergence analysis based on Taylor series expansion is common, but it has limitations, as it requires the use of higher-order derivatives. To overcome this limitation, we conduct the theoretical analysis in a Banach space setting that relies solely on first-order derivatives. The existence of a unique solution is guaranteed within a specific domain, whose radius of convergence is formally obtained using Lipschitz constants. Furthermore, we test the theoretical results on several numerical examples to check the performance and stability of the proposed methods in comparison to existing counterparts.

In chapter 5, the authors propose a third-order two-step and a fifth-order three-step iterative

family that utilizes a derivative-free approach, requiring two frozen divided differences and one matrix inversion per iteration. Additionally, we derive a general multi-point family using a fifth-order scheme as a predictor. The proposed families are analyzed both theoretically and numerically, and experimentation is conducted using a range of numerical problems to confirm the theoretical results on convergence order and computational efficiency. The dynamical properties of each iterative method are studied by means of basins of attraction.

In chapter 6, we propose a novel matrix iteration for numerically computing the matrix sign function. An extensive convergence analysis is carried out in the matrix case to achieve fourth-order convergence. The basins of attraction are illustrated to demonstrate the global convergence behavior of the proposed method. Analytically, it is shown that the presented scheme is asymptotically stable. Furthermore, theoretical developments are numerically verified by discussing the clustering of eigenvalues around ± 1. Moreover, a class of numerical examples for matrices of various dimensions is assessed to exhibit the efficacy of the proposed method.

In chapter 7, the primary objective is to develop two novel iterative schemes for computing the sign of a matrix that has no pure imaginary eigenvalues. Detailed convergence analysis and asymptotic stability of the proposed methods are discussed. It is shown that the new schemes converge globally by drawing attraction basins. In addition, we extend the obtained results to compute nontrivial solutions of the Yang-Baxter-like equation, provided the given matrix has no eigenvalues on the imaginary axis. To illustrate the effectiveness of theoretical developments, a class of numerical examples for matrices of various dimensions, including eigenvalue clustering, is worked out.

In chapter 8, we provide the summary and future aspects of the presented work.

Chapter 2

Optimal family of Chebyshev–Halley type methods for multiple roots

In this chapter, we introduce a higher-order optimal family of Chebyshev–Halley type methods to solve a univariate nonlinear equation with multiple roots. The proposed scheme considers weight functions that are selected adequately to optimize the convergence order and demands only four functional evaluations at each iteration. An extensive convergence analysis is also provided that demonstrates the establishment of eighth-order convergence for the developed scheme. As a result, the efficiency index of the proposed scheme is optimal according to the Kung–Traub conjecture. Finally, the theoretical findings are verified through numerical experiments that include real-life and nonlinear academic problems. In addition, the dynamical study of iterative schemes reflects a comprehensive overview of their stability, convergence properties, and graphical aspects by drawing attraction basins in the complex plane.

2.1 Introduction

One of the most well-known and straightforward method for calculating multiple roots is the modified Newton's technique [292], given by

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \tag{2.1}$$

The contents of this chapter are published in Mathematical Methods in the Applied Sciences, Vol. 48 No. 7, pp. 7037–7055, 2025 (SCIE, I.F.: 1.8)

such that the initial guess should be close enough to the required root. Moreover, the modified Newton's method achieves second-order convergence but requires the computation of the first derivative at each iteration. Further, to achieve even higher orders of convergence, various iterative schemes based on Newton's approach have been given and studied in the literature, see Victory and Neta [353], Soleymani et al. [329], Kansal et al. [190, 192], Ali et al. [14], and more. One such well-known scheme is the cubically convergent Chebyshev-Halley family [268], given as follows

$$x_{n+1} = x_n - \left[1 + \frac{1}{2}\left(\frac{L_f(x_n)}{1 - \alpha L_f(x_n)}\right)\right]\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \tag{2.2}$$

where $L_f(x_n) = \frac{f''(x_n)f(x_n)}{f'(x_n)^2}$, and $\alpha \in \mathbb{R}$. By giving certain values to the parameter α, the family (2.2) generates many iterative techniques, including the classical Chebyshev method for $(\alpha = 0)$, Halley method for $\left(\alpha = \frac{1}{2}\right)$, and super-Halley method for $(\alpha = 1)$. Although the family (2.2) attains a higher order of convergence than Newton's method, it depends on the second derivative of the function f, which restricts the efficiency and practical applicability. In general, one-point methods have theoretical limitations since they can not achieve the p-th convergence order without evaluating the first $p-1$ derivatives of f. On the other hand, multi-point methods can overcome the drawbacks of one-point methods in terms of convergence order and computational efficiency. This feature prompted several researchers to investigate higher-order multi-point Newton-like methods that are free from second- or higher-order derivative computations for approximating multiple zeros of nonlinear functions. For a detailed survey on different classes of such methods, one can consult the research articles given by Zafar et al. [375], Behl et al. [46, 52, 53], Alharbey at al. [13], Sharma and Kumar [308], and references therein.

It was previously challenging to generate a higher-order optimal multi-point scheme for multiple zeros of a given function with multiplicity m. Nowadays, highly efficient computers enable the design of higher-order optimal methods. Many scholars, including Shengguo et al. [314], Zhou et al. [383], Li et al. [230], Sharifi et al. [297], Sharma and Sharma [311], Soleymani and Babajee [328, 329], Thukral [349], Behl et al. [49], developed optimal fourth-order iterative methods. In addition, several researchers have attempted to develop higher-order optimal methods, but they can achieve, at most, sixth-order convergence for calculating multiple roots. For instance, in 2013, Thukral [350] presented a multi-point iterative method with sixth-order convergence, which is given by

$$y_n = x_n - m\frac{f(x_n)}{f'(x_n)},$$

$$
\begin{aligned}
z_n &= x_n - m\frac{f(x_n)}{f'(x_n)}\sum_{i=1}^{3}\left(i\frac{f(y_n)}{f(x_n)}\right)^{i/m},\\
x_{n+1} &= z_n - \frac{f(x_n)}{f'(x_n)}\left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}\left[\sum_{i=1}^{3}\left(i\frac{f(y_n)}{f(x_n)}\right)^{\frac{i}{m}}\right]^2, \quad n \in \hat{\mathbb{N}}_0.
\end{aligned} \tag{2.3}
$$

In 2015, Geum et al. [134] have given the following two-point sixth-order iterative scheme

$$
\begin{aligned}
y_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\\
x_{n+1} &= y_n - Q(p_n, s_n)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0,
\end{aligned} \tag{2.4}
$$

where $p_n = \sqrt[m]{\frac{f(y_n)}{f(x_n)}}$, $s_n = \sqrt[m-1]{\frac{f'(y_n)}{f'(x_n)}}$ and $Q : \mathbb{C}^2 \to \mathbb{C}$ is holomorphic function in the neighborhood of origin. Furthermore, Geum et al. [135] have proposed a new three-point iterative scheme of sixth-order convergence using weight functions for estimating multiple zeros, which can be seen in the following expression

$$
\begin{aligned}
y_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\\
w_n &= x_n - mG(p_n)\frac{f(x_n)}{f'(x_n)},\\
x_{n+1} &= x_n - mK(p_n, t_n)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0,
\end{aligned} \tag{2.5}
$$

where $p_n = \sqrt[m]{\frac{f(y_n)}{f(x_n)}}$, and $t_n = \sqrt[m]{\frac{f(w_n)}{f(x_n)}}$. Here $G : \mathbb{C} \to \mathbb{C}$ and $K : \mathbb{C}^2 \to \mathbb{C}$ are holomorphic functions in the neighborhood of origin. The above three schemes (2.3)–(2.5) require four function evaluations to attain sixth-order convergence. With the passage of time, many mathematicians have attempted to refine existing schemes for improved accuracy and higher convergence order. Several optimal methods are also proposed by Zafar et al. [375], Cordero et al. [94], Behl et al. [52], and Akram et al. [12]. For instance, in 2019, Behl et al. [52] proposed a three-point sixth-order iterative technique based on a Chebyshev-Halley family with four functional evaluations per iteration, given as follows

$$
\begin{aligned}
y_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\\
z_n &= x_n - m\left(1 + \frac{u_n}{1 - \alpha u_n}\right)\frac{f(x_n)}{f'(x_n)},\\
x_{n+1} &= z_n - H(u_n, v_n)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0,
\end{aligned} \tag{2.6}
$$

where $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $v_n = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}$, and $H : \mathbb{C}^2 \to \mathbb{C}$ is holomorphic function in the neighborhood of origin. The above family (2.6) requires four functional evaluations to achieve sixth-order convergence under specified constraints on the weight function $H(u_n, v_n)$. It attains eighth-order convergence for only a specific parametric value, i.e., $\alpha = 2$. Recently, Sharma and Sharma [308] proposed an optimal three-point iterative scheme using different multi-valued functions for computing multiple roots $m > 1$, requiring four functional evaluations to achieve at least the eighth order of convergence.

Motivated by the research going in this direction to generate more efficient optimal higher-order methods, this work focuses on the design, analysis, and implementation of multi-point iterations with a higher convergence order requiring minimum functional evaluations, which follows the Kung-Traub conjecture [221]. In the literature, the class of optimum three-point iterative methods of the Chebyshev-Halley type for estimating the multiple roots of nonlinear equations has yet to be discussed. Therefore, inspired by this, we have attempted to present a new three-point Chebyshev-Halley family with a weight function approach that is adequately selected to optimize the convergence order for distinct multi-valued functions. Furthermore, the applicability of the newly suggested methods is tested numerically on real-life problems that can be modeled as nonlinear equations with multiple roots. The basins of attraction will also be drawn to determine the convergence domains of the proposed methods.

2.2 Iterative scheme

In this section, the primary objective is to construct an iterative algorithm that utilizes the weight function approach to accelerate the convergence rate of the Chebyshev-Halley scheme (2.2) without relying on any second-order derivatives. The technique is designed to minimize computing cost while achieving the highest feasible convergence order. In this view, it is reasonable to analyze an iterative approach of the form:

$$\begin{aligned} w_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= w_n - m\left(1 + \frac{\mu_n}{1-\alpha\mu_n}\right) Q(\mu_n)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= z_n - m\vartheta_n G(\mu_n, \vartheta_n)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{2.7}$$

where $\mu_n = \left(\frac{f'(w_n)}{f'(x_n)}\right)^{\frac{1}{m-1}}, \vartheta_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$, and α is a free parameter. In addition, $Q(\mu_n)$, and $G(\mu_n, \vartheta_n)$ denote the weight functions in the neighborhood of 0 and $(0, 0)$, respectively. Note that the functions μ_n and ϑ_n are multi-valued. Therefore, we consider their principal analytic branches (see Ahlfors [10]). For instance, we consider the principal root of ϑ_n given by $\vartheta_n = \exp\left[\frac{1}{m}\log\left(\frac{f(z_n)}{f(x_n)}\right)\right]$, where $\log\left(\frac{f(z_n)}{f(x_n)}\right) = \log\left|\frac{f(z_n)}{f(x_n)}\right| + i\arg\left(\frac{f(z_n)}{f(x_n)}\right)$ for $-\pi < \arg\left(\frac{f(z_n)}{f(x_n)}\right) \leq \pi$.

Also, we can write $\vartheta_n = \left|\frac{f(z_n)}{f(x_n)}\right|^{\frac{1}{m}} \exp\left[\frac{1}{m}\arg\left(\frac{f(z_n)}{f(x_n)}\right)\right] = O(e_n)$. This convention of principal argument $\arg(z)$ for $z \in \mathbb{C}$ coincides with the built-in command of computer algebra system used in this work.

2.3 Convergence analysis

This section is to analyze the convergence order of the proposed scheme (2.7). In the following theorem, the Taylor series expansion is applied to functions, and we derive conditions for the weight functions to optimize the convergence order.

Theorem 2.3.1. *Assume that $\eta \in \mathbb{R}$ be a multiple zero with multiplicity $m \geq 2$ of the continuous function $f : \mathbb{D} \subseteq \mathbb{C} \to \mathbb{C}$ defined on the open interval $\mathbb{D}$, containing the required root. Then, the proposed scheme* (2.7) *has at least eighth-order convergence, provided,*

$$Q_0 = 0,\ Q_1 = 1,\ Q_2 = \frac{2(1+m)}{m-1},$$

$$Q_3 = \frac{3(4+4m-G_{20}m-4m^2+G_{20}m^2-2\alpha+4m\alpha-2m^2\alpha)}{(m-1)^2},$$

$$G_{00} = 1,\ G_{10} = 2,\ G_{01} = \frac{m-1}{m},\ G_{11} = \frac{2(2m^2-2m-1)}{m^2},$$

and

$$G_{20} = \frac{2(18-10m-28m^2+24m^3-6\alpha(1-2m+m^2)+3\alpha^2(1-3m+3m^2-m^3))}{3(2+m-6m^2+3m^3)},$$

where α is any real number.

Proof. Suppose that $e_n = x_n - \eta$ denotes the error in the n^{th} iteration. We assume Taylor's expansion for the functions $f(x_n)$ and $f'(x_n)$ around the zero $x = \eta$, given by

$$f(x_n) = \frac{f^m(\eta)}{m!}e_n^m\left(1 + c_1e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + c_8e_n^8 + O(e_n^9)\right), \tag{2.8}$$

and

$$f'(x_n) = \frac{f^m(\eta)}{m!}e_n^{m-1}\Big(m + (m+1)c_1e_n + (m+2)c_2e_n^2 + (m+3)c_3e_n^3 + (m+4)c_4e_n^4 + (m+5)c_5e_n^5 + (m+6)c_6e_n^6 + (m+7)c_7e_n^7 + O(e_n^8)\Big), \tag{2.9}$$

respectively. Here, $c_k = \frac{m!}{(m-k)!}\frac{f^{(m-k)}(\eta)}{f^{(m)}(\eta)}, \quad k = 1, 2, 3, 4, \ldots, 8.$

Substituting the expressions (2.8) and (2.9) in the first sub-step of scheme (2.7), one gets

$$w_n - \eta = \frac{c_1}{m}e_n^2 + \sum_{i=0}^{5}\kappa_i e_n^{i+3} + O(e_n^9), \tag{2.10}$$

where

$$\kappa_0 = \frac{-(1+m)c_1^2 + 2mc_2}{m^2},\ \kappa_1 = \frac{(1+m)^2c_1^3 - m(4+3m)c_1c_2 + 3m^2c_3}{m^3},$$

etc. For the sake of simplicity, we are not mentioning the terms $\kappa_2, \kappa_3, \kappa_4$, and κ_5 as these contain lengthy expressions. Now, we substitute (2.10) in (2.9) to obtain $f'(w_n)$ and by adopting (2.9), we can get

$$\mu_n = \left(\frac{f'(w_n)}{f'(x_n)}\right)^{\frac{1}{m-1}} = \frac{c_1e_n}{m} + \frac{\left(-\frac{(1+m)c_1^2}{(m-1)} + 2c_2\right)e_n^2}{m} + \sum_{i=0}^{5}\theta_i^{i+3} + O(e_n^9), \tag{2.11}$$

where

$$\theta_0 = \frac{((-2 - m + 2m^2 + 3m^2 + 2m^4)c_1^3 - 2m^2(-4 + m + 3m^2)c_1c_2 + 6(-1+m)^2m^2c_3)}{2(m-1)^2m^3},$$

etc. It is straightforward to say from the expression (2.11) that μ_n is of order e_n. So, we can expand the weight function $Q(\mu_n)$ in the neighborhood of the origin by Taylor series expansion up to third-order terms as follows

$$Q(\mu_n) \approx Q_0 + \frac{Q_1\mu_n}{1!} + \frac{Q_2\mu_n^2}{2!} + \frac{Q_3\mu_n^3}{3!}, \tag{2.12}$$

where $Q_j = Q^j(0)$ for $0 \le j \le 3$. By substituting the equations (2.8)–(2.12) in the second sub-step of proposed scheme (2.7), we get the following relation

$$z_n - \eta = -Q_0e_n - \frac{(-1+Q_1)c_1e_n^2}{m} + \frac{1}{2(-1+m)m^2}\Big((2 + 2m^2(-1 + Q_1) + Q_2 - m(-2Q_1 + Q_2 + 2Q_0(-2+\alpha)) + 2Q_0\alpha)c_1^2$$

$$+ 4m(m-1+Q_1-mQ_1)c_2\Big)e_n^3 + O(e_n^4). \tag{2.13}$$

In order to get at least fourth-order convergence, we can substitute the conditions $Q_0 = 0,\ Q_1 = 1,$ and $Q_2 = \dfrac{2(1+m)}{m-1}$. Then, (2.13) becomes

$$\begin{aligned}\tilde{e}_n = z_n - \eta =& \frac{1}{6(m-1)^2m^3}\Big((12+3m^3-Q_3-6\alpha-m^2(18+Q_3+6\alpha)\\ &+ m(3+2Q_3+12\alpha))c_1^3 - 6(m-1)m^2c_1c_2\Big)e_n^4 + O(e_n^5).\end{aligned} \tag{2.14}$$

Using (2.8) and Taylor's series expansion of $f(z_n)$, we obtain

$$\begin{aligned}\vartheta_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}} =& \frac{1}{6(m-1)^2m^3}\Big((12+3m^3-Q_3-6\alpha-m^2(-18+Q_3\\ &+ 6\alpha)+m(3+2Q_3+12\alpha))c_1^3 - 6(m-1)^2c_1c_2)\Big)e_n^3\\ &+ \sum_{i=0}^{5}\varphi_i^{i+3} + O(e_n^9),\end{aligned} \tag{2.15}$$

where $\varphi_i = \varphi_i(m, f^{(m)}(\eta), \alpha, c_1, c_2, c_3, c_4, c_5, c_6)$. On expanding the weight function using Taylor's series of $G(\mu_n, \vartheta_n)$ in the neighborhood of origin, we get the following

$$G(\mu_n, \vartheta_n) \approx G_{00} + G_{10}\mu_n + G_{01}\frac{\vartheta_n}{\mu_n} + \frac{1}{2}\left(\mu_n^2G_{20} + 2\vartheta_nG_{11} + \frac{\vartheta_n^2}{\mu_n^2}G_{02}\right), \tag{2.16}$$

where $G_{ij} = G^{i,j}(0,0)$ for $0 \le i, j \le 2$. After substituting (2.9)–(2.16) in the third sub-step of iterative scheme (2.7), it leads us to

$$\begin{aligned}e_{n+1} =& -\frac{(-1+G_{00})c_1}{6(m-1)^2m^3}\Big((12+3m^3-Q_3-6\alpha-m^2(-18+Q_3+6\alpha)+m(3\\ &+ 2Q_3+12\alpha))c_1^2 - 6(m-1)m^2c_2)\Big)e_n^4 + \sum_{i=0}^{5}\omega_i^{i+5} + O(e_n^9),\end{aligned} \tag{2.17}$$

where $\omega_i = \omega_i(m, f^{(m)}(\eta), \alpha, c_1, c_2, c_3, c_4, c_5, c_6)$.

Now, to get convergence order eight, the coefficients of $e_n^4, e_n^5, e_n^6, e_n^7$ are equated to zero, then we obtain the following conditions on the weight function

$$G_{00} = 1,\ G_{10} = 2,\ G_{01} = \frac{m-1}{m},\ G_{11} = \frac{2(2m^2-2m-1)}{m^2},$$

$$Q_3 = \frac{3(4+4m-G_{20}m-4m^2+G_{20}m^2-2\alpha+4m\alpha-2m^2\alpha)}{(m-1)^2},$$

$$G_{20} = \frac{2(18-10m-28m^2+24m^3-6\alpha(1-2m+m^2)+3\alpha^2(1-3m+3m^2-m^3))}{3(2+m-6m^2+3m^3)}.$$

Therefore, from (2.17), we obtain the following error equation

$$e_{n+1} = \left(\sum_{i=0}^{8}\Delta_i\right) e_n^8 + O(e_n^9). \tag{2.18}$$

Here $\Delta_i = \Delta_i(m, f^{(m)}(\eta), \alpha, c_1, c_2, c_3, c_4, c_5, c_6)$. For instance, the first few coefficients can be explicitly written as

$$\begin{aligned}
\Delta_0 = \tau((&21m^7 + m^2(44+156\alpha-78\alpha^2) + 18m^6(6+\alpha) + 12(3-2\alpha+\alpha^2) - 4m^5(35\\
&- 9\alpha + 3\alpha^2 + 2m(65-6\alpha^2) - 12m^4(27-\alpha+8\alpha^2) + 3m^3(15-60\alpha+56\alpha^2))c_1^4\\
&- 6(m-1)m(4+12m^5 - 2m(4-6\alpha+3\alpha^2) - 6m^3(13-2\alpha+3\alpha^2) + m^4(33\\
&+ 6\alpha^2) + 3m^2(-5-8\alpha+6\alpha^2))c_1^2c_2 - 12(m-1)^2m^3(2+3m-3m^2)(c_2^2+c_1c_3)),\\
\Delta_1 = \tau((&21m^7 + m^2(54+84\alpha-30\alpha^2) + m(94+24\alpha-24\alpha^2) + 18m^6(5+\alpha^2) + 12(3\\
&- 2\alpha + \alpha^2) - 4m^5(47-9\alpha+6\alpha^2)_4m^4(31+3\alpha+12\alpha^2) - 3m^3(21+36\alpha-\alpha^2))c_1^4\\
&- 6(m-1)m(4+12m^5 - 6m(2-2\alpha+\alpha^2) + 3m^4(9+2\alpha^2) - 6m^3(11-2\alpha\\
&+ 3\alpha^2) - m^2(17+24\alpha-18\alpha^2))c_1^2c_2 - 12(m-1)^2m^3(2+3m-3m^2)(c_2^2+c_1c_3)),
\end{aligned}$$

etc., where $\tau = \frac{\left((1-3m+2m^2)c_1^3-6(-1+m)mc_1c_2+6m^2c_3\right)}{9(-1+m)^3m^7(-2-3m+3m^2)}$. The equation (2.18) confirms that the proposed scheme (2.7) achieves eighth-order convergence for any value of the free parameter α. The proof is complete now. □

Now, the proposed class (2.7) can be written as

$$\begin{aligned}
w_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\\
z_n &= w_n - m\left(1+\frac{\mu_n}{1-\alpha\mu_n}\right)Q(\mu_n)\frac{f(x_n)}{f'(x_n)},\\
x_{n+1} &= z_n - m\vartheta_n G(\mu_n,\vartheta_n)\frac{f(x_n)}{f'(x_n)}, \quad n\in\hat{\mathbb{N}}_0,
\end{aligned} \tag{2.19}$$

such that $Q(\mu_n) = \mu_n + \frac{(1+m)\mu_n^2}{(m-1)} + \beta$, $G(\mu_n,\vartheta_n) = 1 + 2\mu_n + \frac{2(2m^2-2m-1)\vartheta_n}{m^2} + \frac{(m-1)\vartheta_n}{m\mu_n} + \frac{G_{02}\vartheta_n^2}{2\mu_n^2} + \gamma$,

$\beta = \frac{\mu_n^3(12(-2+\alpha)+6m(-4-3\alpha+\alpha^2)-6m^4(-2+3\alpha+\alpha^2)-2m^2(-2+9\alpha+9\alpha^2)+2m^3(8+21\alpha+9\alpha^2))}{6(m-1)^2(3m^2-3m-2)}$, and
$\gamma = -\frac{\mu_n^2(m^2(28+6\alpha-9\alpha^2)+3m^3(\alpha^2-8)-3(6-2\alpha+\alpha^2)+m(10-12\alpha+9\alpha^2))}{3(2+m-6m^2+3m^3)}$, where α and G_{02} are free parameters. It can be seen that the scheme (2.19) satisfies all the conditions stated in Theorem 2.3.1.

Remark 2.3.1. *To achieve eighth-order convergence for every value of the parameter, the proposed family* (2.19) *requires four functional evaluations per step, namely* $f(x_n)$, $f'(x_n)$, $f'(w_n)$, *and* $f(z_n)$. *The suggested class demonstrates that it has optimum order in terms of the Kung-Traub conjecture by requiring just four functional evaluations at each iteration. As a result, the efficiency index of the presented scheme* (2.19) *is* $8^{\frac{1}{4}} \approx 1.6817$.

Remark 2.3.2. *It is worth noting that the proposed approach achieves eighth-order convergence only for multiplicity* $m \geq 2$ *because when* $m = 1$, *the value of* μ_n *is not defined. Moreover, in the coefficient of* e_n^8, *the denominator of* τ *becomes* 0 *for* $m = 1$, *which is impossible. As a result, the primary focus of this research is solely on multiple roots.*

2.4 Numerical results

This section analyzes the convergence behavior of proposed methods LM_1 (for $\alpha = 0$), LM_2 (for $\alpha = 1$), and LM_3 (for $\alpha = 1.9$) on some well-known real-world applications from the scientific literature. First, we study the global CO_2 ocean chemistry and continuous stirred tank reactor (CSTR) models. Following that, we study the root clustering problem to determine the nature of approximate solutions when the roots are closer. An example of higher multiplicity ($m = 50$) is also discussed.

We have considered some previously existing Newton-like schemes from the literature, which have an optimal order of eight, to compare with the proposed methods. Firstly, we consider the scheme proposed by Zafar et al. [375] for computing multiple roots with known multiplicity, having two free parameters and three univariate weight functions. This technique requires four functional evaluations, i.e., $f(x_n), f'(x_n), f(y_n)$, and $f(z_n)$ to get optimal order eight as follows

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - mu_nH(u_n)\frac{f(x_n)}{f'(x_n)}, \\ x_n &= z_n - u_nv_n\left(A_2 + A_3u_n\right)P(v_n)G(w_n)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{2.20}$$

where $A_2,\ A_3 \in \mathbb{R}$ are free parameters and the weight functions $H : \mathbb{C} \to \mathbb{C}$, and $P : \mathbb{C} \to \mathbb{C}$ are analytic in the neighborhood of 0 with $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}},\ v_n = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}, w_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$. For comparison, the following weight functions are taken:

$$H(u_n) = 6u_n^3 - u_n^2 + 2u_n + 1,\ P(v_n) = v_n + 1,\ G(w_n) = \frac{2mw_n}{A_2P_0} + \frac{m}{A_2P_0}, \tag{2.21}$$

and

$$H(u_n) = \frac{1 - 5u_n^2 + 8u_n^3}{1 - 2u_n},\ P(v_n) = v_n + 1,\ G(w_n) = \frac{3w_n + 1}{A_2P_0}. \tag{2.22}$$

Thus, the optimal eighth-order scheme corresponding to weight functions (2.21) and (2.22) takes the form

1. Zafar et al. method ($\mathbf{ZM_1}$):

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - mu_n(6u_n^3 - u_n^2 + 2u_n + 1)\frac{f(x_n)}{f'(x_n)}, \\ x_n &= z_n - mu_nv_n(1 + 2u_n)(1 + v_n)\left(\frac{2w_n + 1}{A_2P_0}\right)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{2.23}$$

2. Zafar et al. method ($\mathbf{ZM_2}$):

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - mu_n\left(\frac{1 - 5u_n^2 + 8u_n^3}{1 - 2u_n}\right)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= z_n - mu_nv_n(1 + 2u_n)(1 + v_n)\left(\frac{3w_n + 1}{A_2P_0(1 + w_n)}\right)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{2.24}$$

where $A_2 = P_0 = 1$.

Next, we have considered an optimal eighth-order scheme given by Behl et al. [46], which requires the four functional evaluations $f(x_n),\ f'(x_n),\ f(y_n)$, and $f(z_n)$, given as follows:

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - u_nG_f(u_n)\frac{f(x_n)}{f'(x_n)}, \end{aligned} \tag{2.25}$$

$$x_{n+1} = z_n + \left(\frac{w_n u_n}{1-w_n}\right)\left(H_f(u_n) + K_f(v_n)\right)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0,$$

where the weight functions G_f, H_f, and $K_f : \mathbb{C} \to \mathbb{C}$ are analytic functions in a neighborhood of 0 with $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $v_n = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}$, and $w_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$. After substituting the values of respective weight functions as provided in [46], we obtain

3. Behl et al. method ($\mathbf{BM_1}$):

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - mu_n(1+2u_n)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= z_n + m\left(\frac{u_n w_n}{-w_n+1}\right)(4u_n^3 - u_n^2 - 2u_n - 2v_n - 1)\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{2.26}$$

4. Behl et al. method ($\mathbf{BM_2}$):

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - mu_n(1+2u_n)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= z_n + m\left(\frac{w_n u_n}{-w_n+1}\right)\frac{(9u_n^2 + 8u_n v_n + 2v_n + 6u_n + 1)}{(1+4u_n)}\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0. \end{aligned} \tag{2.27}$$

Furthermore, the method given by Behl et al. [52] is considered for comparison, whose iterative expression is

5. Behl et al. method ($\mathbf{OM_1}$):

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - m\left(1 + \frac{u_n}{1-\alpha u_n}\right)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= z_n - \frac{u_n v_n(\beta - (\alpha-2)^2 u_n^2(u_n+1) + v_n^2 + v_n^3)}{(v_n+1)(u_n+1)}\frac{f(x_n)}{f'(x_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{2.28}$$

where $\beta = m\left((\alpha(\alpha+2)+9)u_n^3 + u_n^2(\alpha(\alpha+3-6v_n-3)) + u_n(\alpha+8v_n+1) + 2v_n + 1\right)$.

In Tables 2.1–2.5, we have displayed the number of iterations (n), residual errors in the consecutive iterations ($|x_{n+1} - x_n|$), absolute error of the corresponding function ($|f(x_n)|$), the computational order of convergence (denoted by ρ), and CPU-time (execution of time in seconds), where

the expression for ρ (see [105]) is

$$\rho \approx \frac{\log|f(x_{n+1})/f(x_n)|}{\log|f(x_n)/f(x_{n-1})|},\ n \in \mathbb{N}. \tag{2.29}$$

In order to minimize round-off errors, we perform our calculations with a minimum of 3000 significant digits. We calculate the absolute residual error and functional residuals up to several significant digits; however, we display the value of $|x_{n+1} - x_n|$ and $|f(x_n)|$ up to first two significant digits with exponent power in Tables 2.1–2.5. An approximate value of x_n with significant digits up to 25 is also provided in Examples 2.4.1–2.4.5. Furthermore, the accuracy of the computational order of convergence is demonstrated to five significant digits in all tables. Further, the error representation of the type $a \times 10^{\pm b}$ is shown in the form $a(\pm b)$. Here, numerical tests are performed using $G_{02} = 2.5$ and the following examples are selected for testing:

2.4.1 Real-life examples

Example 2.4.1. ***Global*** CO_2 ***model in ocean chemistry [291]:***
This example is to study the global CO_2 model in ocean chemistry, where the impact of atmospheric CO_2 is complicated and modifies among the location. The problem is to find a numerical solution to a four-degree nonlinear function

$$f_1(x) = \left(x^4 - \frac{2309}{250}x^3 - \frac{65226608163}{500000}x^2 + \frac{425064009069}{25000}x - \frac{109548083684052 09}{62500000}\right)^5, \tag{2.30}$$

interpreted by Babajee [34] in the estimation of the pH of the ocean. The roots of the above equation are given by $x \approx 268.332, 140.771, 11.286, -411.152$. Here, the desired root is $\eta \approx -411.152$ having multiplicity five. After implementing the proposed and existing methods, the approximated root up to 25 digits is $-411.1521869660539592549395$. The computational results are displayed in Table 2.1 for an initial guess of $x_0 = -412$.

In Table 2.1, the results show that the newly proposed methods LM_1, LM_2, and LM_3 have a better performance in comparison to other compared methods in terms of absolute error, functional error, and CPU time. Moreover, the method BM_2 diverges for this example.

Example 2.4.2. ***Continuous stirred tank reactor problem [83]:***
Here, we consider the problem related to a continuous stirred tank reactor as shown in Figure 2.1. The reactors R and A are filled into the tank at the rates $\mathsf{q} - \mathsf{Q}$ and Q, respectively. The reaction

Table 2.1: Numerical comparisons on Example 2.4.1

Methods	n	$\lvert x_{n+1}-x_n\rvert$	$\lvert f_1(x_n)\rvert$	ρ	CPU-Time
	1	9.5(−15)	7.7(−30)		
ZM_1	2	2.5(−126)	1.1(−587)	8.0000	0.346
	3	6.7(−1019)	1.4(−5050)		
	1	1.3(−14)	3.2(−29)		
ZM_2	2	3.3(−125)	3.7(−582)	8.0000	0.406
	3	6.7(−1010)	1.3(−5005)		
	1	2.7(−15)	1.4(−32)		
BM_1	2	2.9(−131)	2.0(−612)	8.0000	0.390
	3	5.1(−1059)	3.6(−5251)		
	1				
BM_2	2	**	**	**	**
	3				
	1	1.9(−15)	2.3(−33)		
OM_1	2	2.1(−132)	3.2(−619)	8.0000	0.374
	3	2.5(−1070)	9.1(−5308)		
	1	5.1(−16)	3.6(−36)		
LM_1	2	8.9(−138)	5.5(−645)	8.0000	0.313
	3	7.1(−1112)	1.9(−5515)		
	1	9.3(−16)	6.9(−35)		
LM_2	2	1.9(−135)	2.6(−633)	8.0000	0.313
	3	6.4(−1093)	1.0(−5420)		
	1	1.7(−15)	1.2(−33)		
LM_3	2	3.4(−133)	4.7(−622)	8.0000	0.312
	3	1.1(−1074)	2.0(−5329)		

** indicates the non-convergence of respective method

schemes that originated in the tank are

$$\mathsf{A} + \mathsf{R} \rightarrow \mathsf{B},$$

$$\mathsf{B} + \mathsf{R} \rightarrow \mathsf{C},$$

$$\mathsf{C} + \mathsf{R} \rightarrow \mathsf{D},$$

$$\mathsf{D} + \mathsf{R} \rightarrow \mathsf{E}.$$

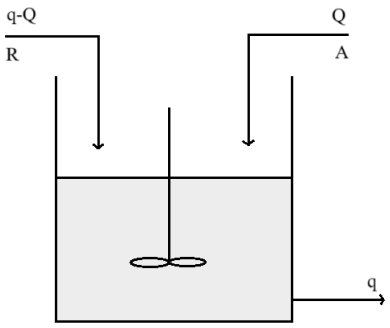


Figure 2.1: Continuous stirred tank reactor.

This problem is further converted by Douglas [117] into a mathematical form

$$K_c \frac{2.98(t+2.25)}{(t+2.85)^2(t+1.45)(t+4.35)} = -1, \tag{2.31}$$

where K_c represents the gain of the proportional controller. For any value of K_c, this control system has stability. So, if we put $K_c = 0$, we can obtain

$$f_2(x) = x^4 + 11.50x^3 + 41.49x^2 + 83.06325x + 51.23266875 = 0. \tag{2.32}$$

The above equation shows that $f_2(x)$ has three roots: -2.85, -1.45, and -4.35. Here, -2.85 is a multiple root with multiplicity $m = 2$. We test this problem for the initial guess $x_0 = -2.5$, and the corresponding results are displayed in Table 2.2. Also, the approximated root computed here is similar to the exact root.

In Table 2.2, the computational results show that the new methods LM_1, LM_2, and LM_3 possess better results due to lesser absolute residual error and functional error than the compared ones. However, the technique ZM_2 diverges for this initial guess. Regarding CPU time, the fastest iterative method is LM_3.

2.4.2 Academic examples

Example 2.4.3. ***Standard problem [268]:***
We consider the following standard academic problem:

$$f_3(x) = (\cos x - x)^5,$$

Table 2.2: Numerical comparisons on Example 2.4.2

Methods	n	$\lvert x_{n+1} - x_n \rvert$	$\lvert f_2(x_n) \rvert$	ρ	CPU-Time
ZM_1	1 2 3	**	**	**	**
ZM_2	1 2 3	9.5(−1) −6.2(0) 4.2(0)	3.4(−1) 5.1(−1) 9.6(2)	18.575	0.374
BM_1	1 2 3	2.0(−2) 7.5(−8) 1.9(−24)	8.6(−4) 1.2(−14) 7.8(−48)	3.0550	0.375
BM_2	1 2 3	2.0(−2) 7.5(−8) 2.0(−24)	8.6(−4) 1.2(−14) 8.0(−48)	3.0551	0.359
OM_1	1 2 3	7.6(−1) 8.9(−1) 2.1(0)	3.0(−1) 2.7(−1) 5.1(−1)	−5.3268	0.437
LM_1	1 2 3	3.4(−3) 5.2(−23) 1.6(−91)	2.5(−5) 5.6(−45) 5.7(−182)	3.4559	0.297
LM_2	1 2 3	4.2(−3) 2.9(−22) 1.6(−88)	3.8(−5) 1.7(−43) 5.1(−176)	3.4567	0.328
LM_3	1 2 3	5.5(−3) 2.1(−21) 4.6(−85)	6.3(−5) 9.4(−42) 4.5(−169)	3.4576	0.266

** indicates the nonconvergence of respective method

where f_3 has a only zero, i.e., $x = 0.739085133$ with multiplicity 5. For this function, we choose the initial guess $x_0 = 0.8$, and after implementing the iterative algorithms, the value of the approximated root up to 25 digits is 0.73908513321516064165531 21.

In Table 2.3, the computational results indicate that the newly suggested methods LM_1, LM_2, and LM_3 offer lesser absolute errors, functional errors, and CPU time than the compared methods. Moreover, the method LM_3 works exceptionally well for this example.

Table 2.3: Numerical comparisons on Example 2.4.3

Methods	n	$\lvert x_{n+1} - x_n \rvert$	$\lvert f_3(x_n) \rvert$	ρ	CPU-Time
	1	1.5(−12)	1.1(−58)		
ZM_1	2	3.5(−97)	6.7(−482)	8.0000	0.515
	3	2.6(−774)	1.5(−3867)		
	1	1.8(−12)	2.4(−58)		
ZM_2	2	1.5(−96)	1.0(−478)	8.0000	0.640
	3	2.6(−769)	1.0(−3841)		
	1	5.2(−13)	5.2(−61)		
BM_1	2	2.2(−101)	7.4(−503)	8.0000	0.563
	3	2.2(−808)	1.3(−4037)		
	1	2.4(−13)	1.0(−62)		
BM_2	2	1.4(−104)	7.4(−519)	8.0000	0.532
	3	2.1(−834)	5.9(−4168)		
	1	1.0(−12)	1.4(−59)		
OM_1	2	8.1(−99)	4.6(−490)	8.0000	0.500
	3	1.3(−787)	5.7(−3934)		
	1	1.3(−13)	5.8(−64)		
LM_1	2	9.0(−107)	7.8(−530)	8.0000	0.375
	3	3.7(−852)	9.2(−4257)		
	1	1.3(−13)	5.4(−64)		
LM_2	2	7.9(−107)	4.0(−530)	8.0000	0.422
	3	1.3(−852)	4.1(−4259)		
	1	9.7(−14)	1.1(−64)		
LM_3	2	3.8(−108)	1.1(−536)	8.0000	0.454
	3	2.1(−863)	5.4(−4313)		

Example 2.4.4. ***Root clustering problem [379]:***

In the fourth example, we consider a polynomial with clustered roots that Zeng [379] has analyzed, given by

$$f_4(x) = (x-1)^{20}(x-2)^{15}(x-3)^{10}(x-4)^5.$$

The function f_4 has four multiple zeros at $x = 1, 2, 3$, and 4 with multiplicities of twenty, fifteen, ten, and five, respectively. All multiple zeros are clustered closely together, but here, the desired

root is $x = 2$ with multiplicity 15. The numerical performance of methods is depicted in Table 2.4 using an initial guess $x_0 = 2.1$. The approximated root is 2, which implies the actual root.

In Table 2.4, the computational results illustrate that the new methods LM_1, LM_2, and LM_3

Table 2.4: Numerical comparisons on Example 2.4.4

Methods	n	$\lvert x_{n+1} - x_n \rvert$	$\lvert f_4(x_n) \rvert$	ρ	CPU-Time
	1	9.8(−9)	2.5(−119)		
ZM_1	2	7.4(−64)	3.5(−946)	8.0000	0.437
	3	7.6(−505)	5.0(−7561)		
	1	1.0(−8)	3.9(−119)		
ZM_2	2	1.0(−63)	6.0(−944)	8.0000	0.391
	3	1.3(−503)	1.8(−7542)		
	1	3.0(−9)	3.6(−127)		
BM_1	2	1.6(−68)	5.0(−1016)	8.0000	0.407
	3	1.4(−542)	7.7(−8127)		
	1	2.6(−9)	6.7(−128)		
BM_2	2	4.2(−69)	6.0(−1025)	8.0000	0.405
	3	1.6(−547)	2.5(−8201)		
	1	1.5(−8)	8.9(−117)		
OM_1	2	1.6(−62)	3.9(−926)	8.0000	0.344
	3	3.5(−494)	5.4(−7401)		
	1	1.9(−9)	6.3(−130)		
LM_1	2	4.1(−70)	4.2(−1040)	8.0000	0.375
	3	1.5(−555)	1.6(−8321)		
	1	1.9(−9)	4.2(−130)		
LM_2	2	3.4(−70)	3.6(−1041)	8.0000	0.344
	3	4.3(−556)	1.2(−8329)		
	1	1.7(−9)	1.4(−130)		
LM_3	2	2.1(−70)	2.2(−1044)	8.0000	0.374
	3	9.1(−558)	7.5(−8355)		

work exceptionally well in terms of error between the consecutive iterations, functional error, and CPU time.

Example 2.4.5. ***Higher multiplicity [268]:***
Now, we consider the following standard academic problem

$$f_5(x) = \left((x-1)^3 - 1\right)^{50},$$

where f_5 has a zero at $x = 2$ with multiplicity 50. The computational results are displayed in Table 2.5 with initial guess $x_0 = 2.1$. Moreover, the approximated root is converging exactly towards the actual root.

In Table 2.5, the computational results show that the newly suggested methods LM_1, LM_2, LM_3 and OM_1 perform better than the other compared schemes in terms of absolute residual error, functional error, and CPU time. Furthermore, in the resulting scheme, by choosing another parametric value like $\alpha = 1.9,\ G_{02} = 5$, we can generate a new iterative method that gives a minor functional error, i.e., $2.2(-22006)$, which is significantly less in comparison to all existing methods.

Therefore, the numerical results show that the proposed methods LM_1, LM_2, and LM_3 are comparable and competitive to the existing optimal eighth-order algorithms. It is evident that the new methods consistently outperform their counterparts in terms of residual errors. Although the considered parametric values are not specific, we can generate several new iterative methods by taking $(\alpha = 2, G_{02} = 0.1)$, $(\alpha = 1.8, G_{02} = 2.5)$, and $(\alpha = 2.1, G_{02} = 2.4)$. Consequently, the new methods can be a better alternative as they offer lesser absolute functional errors than the other compared methods.

2.5 Basins of attraction

In this section, we explain the convergence and stability of the iterative scheme by analyzing its basins of attraction. These basins are obtained by applying the approach to some complex polynomials $f(z)$. Basins of attraction are generally assumed to be like visual geometrical tools for comparing iterative techniques, which explains how a specific scheme functions for different initial points. A concise review of some basic ideas related to the visual tool is provided in the introduction 1.5 and also in the research article [358].

For the complex dynamics, five polynomials with multiple zeros are utilized. We quantify the suggested techniques in Tables 2.6–2.10, by examining different convergence measures per point. The first column (M/P) indicates the average number of iterations required by a method per point until the root is not found. Otherwise, the point is considered non-convergent. The second column

Table 2.5: Numerical comparisons on Example 2.4.5

Methods	n	$\lvert x_{n+1} - x_n \rvert$	$\lvert f_5(x_n) \rvert$	ρ	CPU-Time
	1	4.8(−7)	6.6(−293)		
ZM_1	2	5.7(−49)	3.4(−2389)	8.0000	0.343
	3	2.2(−384)	1.6(−19159)		
	1	6.5(−7)	2.3(−286)		
ZM_2	2	8.4(−48)	9.4(−2331)	8.0000	0.345
	3	6.6(−375)	7.1(−18686)		
	1	1.9(−7)	1.8(−313)		
BM_1	2	8.0(−53)	9.8(−2582)	8.0000	0.312
	3	9.6(−416)	8.8(−20728)		
	1	6.3(−8)	6.2(−337)		
BM_2	2	4.2(−57)	1.1(−2795)	8.0000	0.327
	3	5.9(−169)	3.4(−8388)		
	1	1.4(−7)	3.3(−319)		
OM_1	2	6.7(−54)	1.6(−2635)	8.0000	0.312
	3	1.7(−424)	6.1(−21166)		
	1	1.8(−7)	2.0(−314)		
LM_1	2	4.2(−53)	1.2(−2595)	8.0000	0.297
	3	4.3(−418)	2.1(−20845)		
	1	2.2(−7)	3.1(−310)		
LM_2	2	2.6(−52)	1.0(−2555)	8.0000	0.360
	3	1.4(−411)	1.1(−20519)		
	1	3.4(−7)	1.5(−300)		
LM_3	2	1.7(−50)	8.4(−2466)	8.0000	0.298
	3	6.0(−397)	8.5(−19788)		

contains the proportion of non-convergent points NC(%) that can be seen as black spots in Figures 2.2–2.6. Notably, the value of M/P is contingent on the non-convergent points, as these points have always contributed to the maximum number of 25 iterations permitted. In contrast, convergent points are typically obtained relatively fast because we use higher-order multi-point techniques. Finally, we include a column (M_C/C) that represents the mean number of iterations per convergent point to reduce rounding errors.

To draw the attraction basins, the following test problems (Examples 2.4.1–2.4.5) are chosen in the complex plane for testing the proposed scheme.

Test Problem 1: Let us consider a complex polynomial $f_1(z)$ in the region of the complex plane. Since the prescribed domain D does not enclose the roots, we enlarge the domain up to $[-500, 500] \times [-500, 500]$. Each zero of the polynomial $f_1(z)$ is associated with a specific color. Specifically, we assign the colors purple, yellow, orange, and cyan to the zeros -411.152, 11.286, 140.771, and 268.332, respectively. Also, we paint the points in orange if their orbit converges to the multiple root -411.152, and in black, if it diverges.

Table 2.6: Convergence measures of methods for the test function $f_1(z)$

Methods	ZM_1	ZM_2	BM_1	BM_2	OM_1	LM_1	LM_2	LM_3
M/P	14.3320	16.0899	7.40186	11.7237	10.6193	14.1716	14.0380	11.9634
NC(%)	24.9562	37.0849	7.23033	15.4148	8.67883	35.4286	34.1568	21.8115
M_C/C	10.7855	10.8379	6.03029	9.30427	9.25261	8.25238	8.35214	8.33259

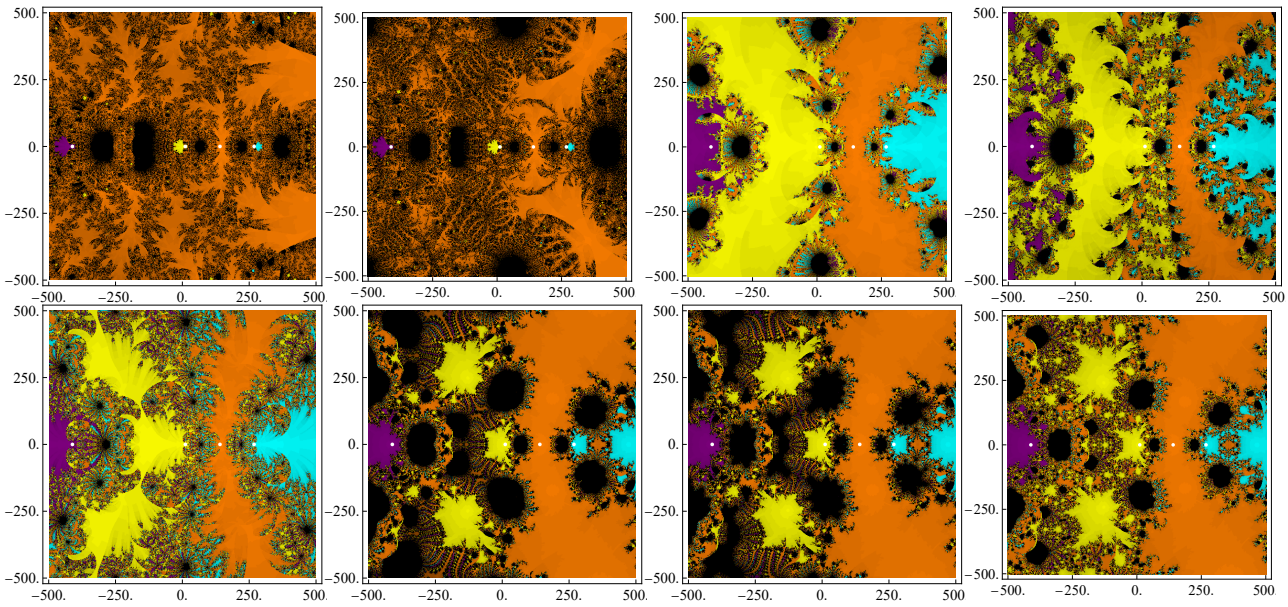


Figure 2.2: Convergence planes of ZM_1, ZM_2, BM_1, BM_2, OM_1, LM_1, LM_2 and LM_3 for test problem 1.

The convergence planes of all methods indicate that the NC(%) of BM_1, BM_2, OM_1 and LM_3 is minimal, therefore exhibiting its stable performance in comparison to other existing methods that can be seen in Figure 2.2 and Table 2.6.

Test Problem 2: Here, the polynomial $f_2(z)$ is analyzed for testing, and we have allotted orange

and yellow colors to the zeros at -2.85 and -1.45, respectively. The points whose orbit is convergent to the desired multiple root -2.85 are colored orange, while those whose orbit diverges are colored black.

Based on the convergence planes of all methods, the NC(%) of each technique except ZM_1

Table 2.7: Convergence measures of methods for the test function $f_2(z)$

Methods	ZM_1	ZM_2	BM_1	BM_2	OM_1	LM_1	LM_2	LM_3
M/P	10.0404	9.94492	6.66829	8.29554	6.92977	9.20234	9.05933	8.77740
NC(%)	14.0788	13.7026	3.31232	4.64188	0.69385	10.3747	9.15495	7.75433
M_C/C	7.58940	7.55463	6.04028	7.48248	6.08351	7.37366	7.45290	7.41370

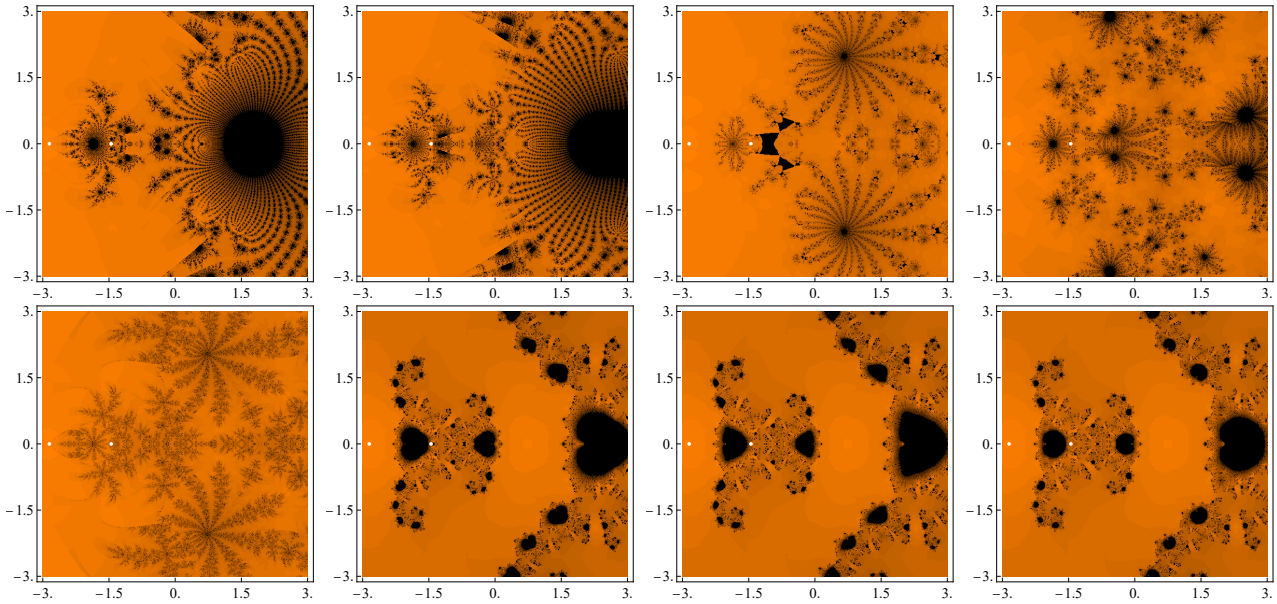


Figure 2.3: Convergence planes of ZM_1, ZM_2, BM_1, BM_2, OM_1, LM_1, LM_2 and LM_3 for test problem 2.

and ZM_2 is smaller. Moreover, the average number of iterations required per convergent point is less for BM_1, BM_2, and LM_1, therefore revealing their stability over the other compared methods that can be seen in Figure 2.3 and Table 2.7.

Test Problem 3: Here, the attraction basins are drawn for $f_3(z)$. Particularly, the points whose orbits converge to the multiple root 0.739085133 are colored orange, and the ones whose orbits converge to strange fixed points, cycles, etc., are colored black.

From the basins Figure 2.4 and Table 2.8, the convergence planes reveal that the LM_1, LM_2, and LM_3 methods require a lesser number of iterations for each mesh point, i.e., M/P. Also, ZM_1

Table 2.8: Convergence measures of methods for the test function $f_3(z)$

Methods	ZM_1	ZM_2	BM_1	BM_2	OM_1	LM_1	LM_2	LM_3
M/P	13.2060	15.1328	9.86413	12.1591	18.4858	9.72074	10.3118	10.1385
NC(%)	89.1245	83.8074	56.0290	67.8594	73.4342	61.0357	60.6462	57.2875
M_C/C	4.37651	4.96719	3.08720	4.01465	4.26228	4.83724	4.85489	5.03506

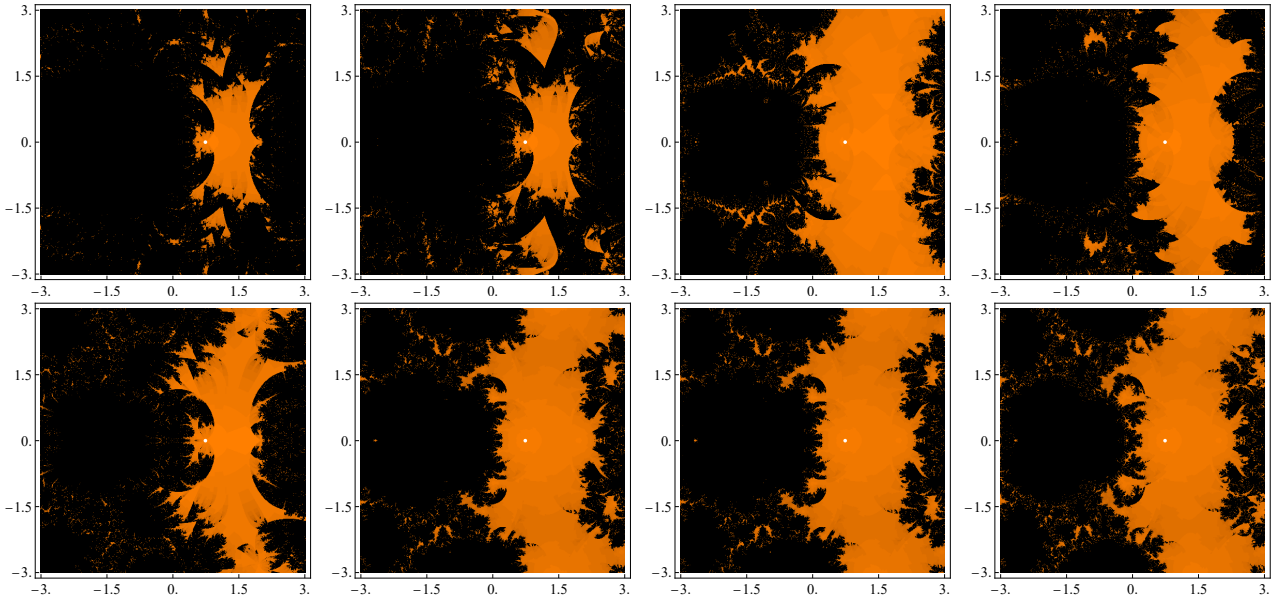


Figure 2.4: Convergence planes of ZM_1, ZM_2, BM_1, BM_2, OM_1, LM_1, LM_2 and LM_3 for test problem 3.

an ZM_2 exhibit more NC(%), while new ones has smaller diverging area. As a result, we can state that the proposed methods are more stable and efficient.

Test Problem 4: For testing, consider the polynomial $f_4(z)$, and its attraction basins are drawn. We assign the colors yellow and orange to zeros 2 and 1, respectively. We color the points gray if their orbit converges to the multiple root 1 and black if it diverges.

From the basins Figure 2.5 and Table 2.9, it indicates that the BM_1 and LM_3 require a lesser

Table 2.9: Convergence measures of methods for the test function $f_4(z)$

Methods	ZM_1	ZM_2	BM_1	BM_2	OM_1	LM_1	LM_2	LM_3
M/P	24.9478	24.9281	6.82426	10.2644	7.42974	7.75535	7.68308	6.86403
NC(%)	99.7067	99.5972	6.35219	5.00249	4.68558	9.44070	7.36865	5.23200
M_C/C	7.23705	7.14151	5.59138	9.48840	6.56599	5.95807	6.30555	5.86285

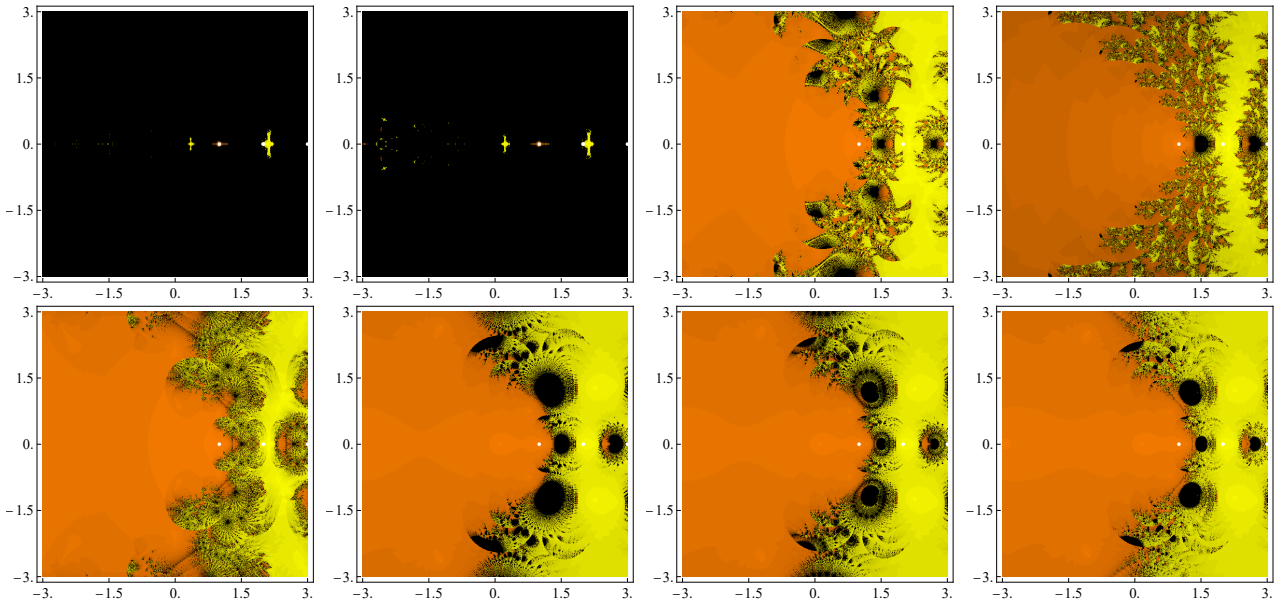


Figure 2.5: Convergence planes of ZM_1, ZM_2, BM_1, BM_2, OM_1, LM_1, LM_2 and LM_3 for test problem 4.

number of iterations, i.e., M/P. Moreover, their NC(%) is less than that of other compared schemes. Whereas ZM_1 and ZM_2 show almost 100% of the diverging area or simply instability. Therefore, we can say that the proposed methods are more stable and have better efficiency, especially LM_3.

Test Problem 5: In this problem, the attraction basins are drawn for $f_5(z)$, and we allot yellow, orange, and cyan colors to zeros $0.5 - 0.8660254037844386I, 0.5 + 0.8660254037844386I$ and 2, respectively. We paint the points whose orbit converges to the multiple root 2 in pink and the points whose orbit diverges in black.

In the basin Figure 2.6 and Table 2.10, it is clear that BM_1, LM_1, LM_2, and LM_3 methods

Table 2.10: Convergence measures of methods for the test function $f_5(z)$

Methods	ZM_1	ZM_2	BM_1	BM_2	OM_1	LM_1	LM_2	LM_3
M/P	24.9957	24.9975	15.7322	19.9901	17.6521	18.7253	18.6005	17.0400
NC(%)	99.9878	99.9886	42.3906	69.5078	43.6522	59.0221	56.2243	47.4002
M_C/C	3.06250	3.13333	8.91670	8.56977	11.9597	9.76522	10.3833	9.86921

require a lesser number of iterations per point (i.e., M/P) and has smaller NC% in the fractal images, whereas ZM_1 an ZM_2 has much larger divergence region indicating, instability. Consequently, the proposed method's stability is thus made apparent.

It is clear from the Figures 2.2–2.6 and convergence measures from Tables 2.6–2.10 that the

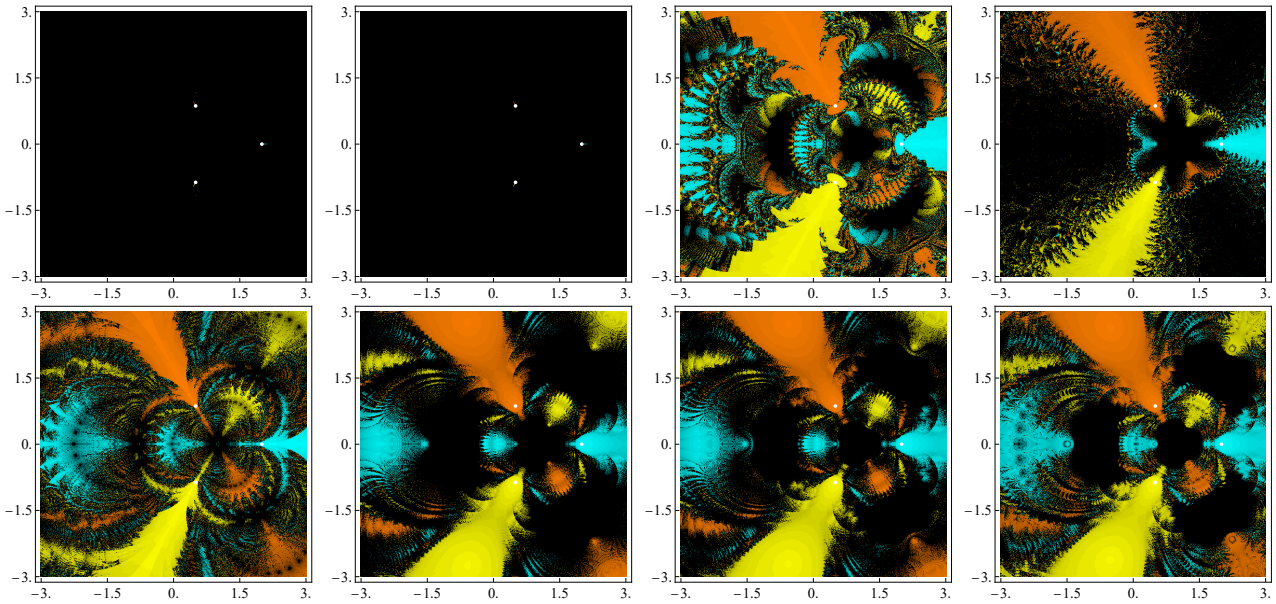


Figure 2.6: Convergence planes of ZM_1, ZM_2, BM_1, BM_2, OM_1, LM_1, LM_2 and LM_3 for test problem 5.

existing methods ZM_1 and ZM_2 have unstable behavior due to the presence of black shaded region in their fractal pictures. However, the basin figures constructed through the proposed algorithms possess brighter areas. They are almost free from black regions compared to others of a similar type, justifying the supremacy of the constructed algorithms over the others.

2.6 Concluding remarks

In this chapter, a three-step Chebyshev-Halley family is presented and analyzed to compute multiple roots of scalar nonlinear equations. The construction of the proposed technique is based on the weight function approach to accelerate the convergence order of the well-known Chebyshev-Halley family. With only four functional evaluations per iteration, the new methods are considered optimal according to the Kung-Traub conjecture. The extensive convergence analysis supports this assertion. Further, numerical performance is evaluated by running the algorithms using high-precision arithmetic to find the multiple zeros of nonlinear problems. Finally, the attraction basins in the complex plane ensure efficiency, accuracy, and stability. Moreover, after a comparative analysis, it was concluded that the proposed methods outperform existing ones in terms of convergence behavior, absolute residual error, functional error, and CPU time elapsed during process execution.

The next chapter expands the applicability of the presented algorithms by introducing a derivative-free iterative scheme for solving nonlinear equations. The method is specifically designed

to address problems where computing the derivative is complex, costly, or impractical.

Chapter 3

Optimal family of derivative-free King's type methods for multiple roots

In this chapter, the authors develop and analyze a new derivative-free class of higher-order iterative methods for locating multiple roots numerically. The scheme is generated by using King's type iterative method. By employing the Traub-Steffensen technique, a derivative-free family is proposed, which requires three functional evaluations to achieve optimal fourth-order convergence. Moreover, it can be observed that the theoretical convergence results of the family are symmetrical for different multiplicities of roots. This further motivates us to present the result in general, which confirms the convergence order of the proposed scheme. It is also worth mentioning that we can obtain already existing methods as particular cases of the proposed family for some suitable choice of free disposable parameters. Finally, we include a wide variety of nonlinear problems to confirm the applicability and effectiveness of the proposed methods.

3.1 Introduction

Many higher-order iterative schemes based on modified Newton's method have been presented and explored in the literature, see references Hansen and Patrick [148], Dong [114], Osada [264], Neta [259], Shengguo et al. [314], Li et al. [230], Sharma an Sharma [311], Zhou et al. [383], Sharifi et

The contents of this chapter are published in Engineering Computations, Vol. 39 No. 6, pp. 2367–2390, 2022 (SCIE, I.F.: 1.9)

al. [297], Soleymani et al. [329], Geum et al. [134], Kansal et al. [192], Behl et al. [53], Behl and Al-Hamdan [44]. These methods can be broadly classified into two categories:

(i) methods with derivatives

(ii) methods without derivatives

The first category of methods requires the evaluation of the first or second-order derivative at each iteration, for instance, modified Newton's method and the Chebyshev-Halley family. These iterative variants cannot be applied to non-smooth functions because they involve derivatives. This fact inspired researchers to develop iterative methods that do not require derivatives for locating multiple roots.

In order to circumvent the use of derivatives, Traub-Steffensen [351] used the following approximation

$$f'(x_n) \simeq \frac{f(x_n + \gamma f(x_n)) - f(x_n)}{\gamma f(x_n)},\ \gamma \in \mathbb{R}\backslash\{0\}. \tag{3.1}$$

In the replacement of derivative $f'(x)$ in the modified Newton method, one obtains the modified Traub-Steffensen method given by

$$x_{n+1} = x_n - m\frac{f(x_n)}{f[u_n, x_n]}, \quad n \in \hat{\mathbb{N}}_0, \tag{3.2}$$

where $u_n = x_n + \gamma f(x_n)$, and $f[u_n, x_n] = \frac{f(u_n)-f(x_n)}{u_n - x_n}$ denotes the first-order divided difference. The method (3.2) is derivative-free and a significant improvement of modified Newton's method because second-order convergence is retained.

Several variants of King's family have been developed and analyzed in the literature for approximating the simple zeros of nonlinear functions. Recently, Behl et al. [51] have extended the King's family using derivatives for multiple zeros of a function, given as follows

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n + (y_n - x_n)\left[\frac{1 + \beta\mu_n}{1 + (\beta - 2)\mu_n}\right] u_n Q(u_n), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.3}$$

where the parameter can take any real value, the weight function $Q(u_n) : \mathbb{C} \to \mathbb{C}$ is a holomorphic function in the neighborhood of origin and $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$ is multi-valued function.

Going a step further, Behl et al. [45] presented the following Ostrowski's method (a special case

of King's family) without using derivatives for the case of multiple roots ($m \geq 2$)

$$\begin{aligned} z_n &= x_n - m\frac{f(x_n)}{f[\mu_n, x_n]}, \\ x_{n+1} &= z_n + (z_n + x_n)\left[\frac{\alpha s_n + \beta t_n}{1 - 2s_n}\right], \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.4}$$

where $\mu_n = x_n + \gamma f(x_n)$, $f[\mu_n, x_n]$ denotes the first-order divided difference, and $\gamma \in \mathbb{R}\backslash\{0\}$. Also, in scheme (3.4), s_n and t_n are muti-valued functions. Furthermore, some derivative-free multi-point methods for computing multiple roots have also been analyzed by Cordero et al. [94], Behl et al. [47, 49], and Kansal et al. [186].

Motivated by the research going in this direction, we have made an attempt to propose a derivative-free King's family [212] that requires only three functional evaluations to attain optimal fourth-order convergence. Moreover, it is in accordance with the Kung–Traub conjecture [221]. The proposed family is analyzed numerically on a variety of problems. It is interesting to observe that the entire family in [45] is a special case of the proposed class. The proposed family is compared to its counterparts of a similar nature to demonstrate computational efficiency.

3.2 Iterative scheme

We consider the following derivative-free scheme

$$\begin{aligned} z_n &= x_n - m\frac{f(x_n)}{f[\mu_n, x_n]}, \\ x_{n+1} &= z_n - m\frac{f(x_n)}{f[\mu_n, x_n]}\left[\frac{1 + \beta s_n}{1 + (\beta - 2)s_n}\right]\frac{(s_n + v_n)}{2}(1 + \alpha(s_n - v_n)), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.5}$$

where $\mu_n = x_n + \gamma f(x_n)$, $s_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $v_n = \left(\frac{f(z_n)}{f(\mu_n)}\right)^{\frac{1}{m}}$. Here, α and β are free parameters, but $\gamma \in \mathbb{R}\backslash\{0\}$. Note that the functions s_n and v_n are multi-valued functions. Therefore, we consider their principal analytic branches [10]. For instance, we consider the principal root of s_n given by

$$s_n = \exp\left[\frac{1}{m}\log\left(\frac{f(z_n)}{f(x_n)}\right)\right],$$

where $\log\left(\frac{f(z_n)}{f(x_n)}\right) = \log\left|\frac{f(z_n)}{f(x_n)}\right| + i\arg\left(\frac{f(z_n)}{f(x_n)}\right)$ for $-\pi < \arg\left(\frac{f(z_n)}{f(x_n)}\right) \leq \pi$. Also, we can write $s_n = \left|\frac{f(z_n)}{f(x_n)}\right|^{\frac{1}{m}} \exp\left[\frac{1}{m}\arg\left(\frac{f(z_n)}{f(x_n)}\right)\right] = O(e_n)$.

3.3 Convergence analysis

In this section, we prove that the proposed scheme (3.5) attains an optimal fourth-order of convergence for all non-zero real values of γ. In the following Theorems 3.3.1–3.3.2, we will show that the theoretical convergence results of the scheme (3.5) are symmetrical for different cases depending on the multiplicity m.

Firstly, we prove the result for the case $m = 2$ in the following theorem.

Theorem 3.3.1. *Assume that $\eta \in \mathbb{R}$ be a multiple zero with multiplicity $m = 2$ of the function $f : \mathbb{D} \subseteq \mathbb{C} \to \mathbb{C}$ defined on the interval $\mathbb{D}$, containing the required root η. Then, the proposed scheme* (3.5) *has at least fourth-order convergence satisfying the following error equation*

$$e_{n+1} = \frac{1}{32}\Big((\sigma_1 + 2c_1)(\sigma_1^2(-1+\beta-\alpha) + \sigma_1(-3+4\beta-2\alpha)c_1 + (3+4\beta)c_1^2 - 4c_2\Big)\Big)e_n^4 + O(e_n^5),$$

where α, $\beta \in \mathbb{R}, \gamma \in \mathbb{R}\backslash\{0\}$, $\sigma_1 = \gamma f''(\eta)$, and $c_k = \frac{2!}{(2+k)!}\frac{f^{(2+k)}(\eta)}{f''(\eta)}$, $k = 1, 2$.

Proof. Suppose that $e_n = x_n - \eta$ denotes the error at the n^{th} iteration. We assume Taylor's expansion for the functions $f(x_n)$ and $f(\mu_n)$ around the zero $x = \eta$ such that $f(\eta) = f'(\eta) = 0$ and $f''(\eta) \neq 0$, which are given by

$$f(x_n) = \frac{f''(\eta)}{2!}e_n^2\left(1 + c_1e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right), \tag{3.6}$$

and

$$\begin{aligned} f(\mu_n) =& \frac{f''(\eta)}{2!}e_n^2\Bigg[1 + (\sigma_1 + c_1)e_n + \frac{1}{4}(\sigma_1^2 + 10\sigma_1c_1 + 4c_2)e_n^2 + \frac{1}{4}(5\sigma_1^2c_1 \\ &+ 6\sigma_1c_1^2 + 12\sigma_1c_2 + 4c_3)e_n^3 + \frac{1}{8}(\sigma_1^3c_1 + 14\sigma_1^2c_1^2 + 16\sigma_1^2c_2 \\ &+ 28\sigma_1c_1c_2 + 28\sigma_1c_3 + 8c_4)e_n^4 + O(e_n^5)\Bigg], \end{aligned} \tag{3.7}$$

respectively. Here, $c_k = \frac{2!}{(2+k)!}\frac{f^{(2+k)}(\eta)}{f''(\eta)}$, for $k = 1, 2, 3, 4, \ldots$.

Substituting the expressions (3.6) and (3.7) in the first sub-step of scheme (3.5), one gets

$$\begin{aligned} z_n - \eta =& \frac{1}{4}(\sigma_1 + 2c_1)e_n^2 - \frac{1}{16}\left[\sigma_1^2 - 8\sigma_1c_1 + 12c_1^2 - 16c_2\right]e_n^3 + \frac{1}{64}\Bigg[\sigma_1^3 - 10c_1(\sigma_1^2 \\ &+ 16c_2) - 20\sigma_1^2c_1^2 + 64\sigma_1c_2 + 72c_1^3 + 96c_3\Bigg]e_n^4 + O(e_n^5). \end{aligned} \tag{3.8}$$

Using equation (3.8) and expansion of Taylor series for the function $f(z_n)$, we obtain

$$\begin{aligned} f(z_n) =& \frac{f''(\eta)}{2!}e_n^2\Bigg[\frac{1}{16}(\sigma_1+2c_1)^2e_n^2 - \frac{1}{32}(\sigma_1+2c_1)(\sigma_1^2-8\sigma_1c_1+12c_1^2-16c_2)e_n^3 \\ &+ \frac{1}{256}\Big(3\sigma_1^4 - 4c_1\Big(7\sigma_1^3-48\sigma_1c_2-96c_3\Big)+96\sigma_1^2c_2+32c_1^2(\sigma_1^2-32c_2) \\ &- 80\sigma_1c_1^3+192\sigma_1c_3+464c_1^4+256c_2^2\Big)e_n^4+O(e_n^5)\Bigg]. \end{aligned} \tag{3.9}$$

By adopting the expressions (3.6), (3.7) and (3.9), we get

$$\begin{aligned} s_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{2}} =& \frac{1}{4}(\sigma_1+2c_1)e_n + \frac{1}{16}\left(-\sigma_1^2+6\sigma_1c_1-16c_1^2+16c_2\right)e_n^2 + \frac{1}{64}\left(\sigma_1^3\right. \\ &\left.-6\sigma_1^2c_1-22\sigma_1c_1^2+56\sigma_1c_2+116c_1^3-208c_1c_2+96c_3\right)e_n^3+O(e_n^4), \end{aligned} \tag{3.10}$$

and

$$\begin{aligned} v_n = \left(\frac{f(z_n)}{f(\mu_n)}\right)^{\frac{1}{2}} =& \frac{1}{4}\left(\sigma_1+2c_1\right)e_n + \frac{1}{16}\left(-3\sigma_1^2-4\sigma_1c_1-16c_1^2+16c_2\right)e_n^2 + \frac{1}{64}(7\sigma_1^3 \\ &- 22\sigma_1^2c_1-14\sigma_1c_1^2+24\sigma_1c_2+116c_1^3-208c_1c_2+96c_3)e_n^3+O(e_n^4). \end{aligned} \tag{3.11}$$

Now, substituting the equations (3.6)–(3.11) in the proposed scheme (3.5), we get the following relation

$$\begin{aligned} e_{n+1} = \frac{1}{32}\Big((\sigma_1+2c_1)(\sigma_1^2(-1+\beta-\alpha)+\sigma_1(-3+4\beta-2\alpha)c_1+(3+4\beta)c_1^2 \\ -4c_2\Big)\Big)e_n^4+O(e_n^5), \end{aligned} \tag{3.12}$$

where $\sigma_1 = \gamma f''(\eta)$. Hence, the family (3.5) achieves fourth-order convergence for multiplicity two. □

Now, we prove the result for $m = 3$.

Theorem 3.3.2. *Assuming the same hypothesis of Theorem 3.3.1, the family* (3.5) *achieves fourth-order convergence for* $m = 3$ *with error equation*

$$e_{n+1} = \frac{1}{108}\left(8(1+\beta)b_1^3-3b_1(\sigma_2+4b_2)\right)e_n^4+O(e_n^5),$$

where $\alpha,\ \beta \in \mathbb{R}$, $\gamma \in \mathbb{R}\backslash\{0\}$, $\sigma_2 = \gamma f'''(\eta)$, *and* $b_j = \dfrac{3!}{(3+j)!}\dfrac{f^{(3+j)}(\eta)}{f'''(\eta)}$, $j = 1, 2$.

Proof. Firstly, we suppose that $f(\eta) = f'(\eta) = f''(\eta) = 0$ and $f'''(\eta) \neq 0$. Then, we obtain the

following Taylor series expansions of the functions $f(x_n)$ and $f(\mu_n)$ around $x = \eta$ as

$$f(x_n) = \frac{f'''(\eta)}{3!}e_n^2(1 + b_1e_n + b_2e_n^2 + b_3e_n^3 + b_4e_n^4 + O(e_n^5)), \tag{3.13}$$

and

$$\begin{aligned} f(\mu_n) =& \frac{f'''(\eta)}{3!}e_n^3\Bigg[1 + b_1e_n + \frac{1}{2}(\sigma_2 + 2b_2)e_n^2 + \left(\frac{7}{6}\sigma_2b_1 + b_3\right)e_n^3 \\ &+ \frac{1}{12}\Big(\sigma_2^2 + 8\sigma_2b_1^2 + 16\sigma_2b_2 + 12b_4\Big)e_n^4 + O(e_n^5)\Bigg], \end{aligned} \tag{3.14}$$

where $b_j = \dfrac{3!}{(3+j)!}\dfrac{f^{(3+j)}(\eta)}{f^{(3)}(\eta)}$, $j = 1, 2, 3, 4$ are error constants. By using the expressions (3.13) and (3.14) in the first sub-step of scheme (3.5), we have

$$\begin{aligned} z_n - \eta =& \frac{b_1}{3}e_n^2 + \frac{1}{18}(3\sigma_2 - 8b_1^2 + 12b_2)e_n^3 + \left(\frac{1}{9}b_1(2\sigma_2 - 13b_2) + \frac{16b_1^3}{27} + b_3\right)e_n^4 \\ &+ O(e_n^5). \end{aligned} \tag{3.15}$$

Expression (3.15) and Taylor series expansion of $f(z_n)$ lead us to

$$f(z_n) = \frac{f'''(\eta)}{3!}e_n^3\left[\frac{b_1^3}{27}e_n^3 + \frac{1}{54}b_1^2(3\sigma_2 - 8b_1^2 + 12b_2)e_n^4 + O(e_n^5)\right]. \tag{3.16}$$

By adopting the expressions (3.13), (3.14) and (3.16), we obtain

$$\begin{aligned} s_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{3}} =& \frac{b_1}{3}e_n + \frac{1}{18}(3\sigma_2 - 10b_1^2 + 12b_2)e_n^2 + \frac{1}{54}\Big(9\sigma_2b_1 + 46b_1^3 \\ &- 96b_2b_1 + 54b_3\Big)e_n^3 + O(e_n^4), \end{aligned} \tag{3.17}$$

and

$$\begin{aligned} v_n = \left(\frac{f(z_n)}{f(\mu_n)}\right)^{\frac{1}{3}} =& \frac{b_1}{3}e_n + \frac{1}{18}(3\sigma_2 - 10b_1^2 + 12b_2)e_n^2 + \frac{1}{27}\Big(3\sigma_2b_1 + 23b_1^3 \\ &- 48b_2b_1 + 27b_3\Big)e_n^3 + O(e_n^4). \end{aligned} \tag{3.18}$$

On substituting the expressions (3.13)–(3.18) in the proposed scheme (3.5), one can get the following error equation

$$e_{n+1} = \frac{1}{108}\left(8(1+\beta)b_1^3 - 3b_1(\sigma_2 + 4b_2)\right)e_n^4 + O(e_n^5), \tag{3.19}$$

where $\sigma_2 = \gamma f'''(\eta)$. Hence, the presented scheme (3.5) attains fourth-order convergence for multi-

plicity three. □

3.3.1 General form of error equation

Now, we shall present the general form of the error equation of the derivative-free scheme (3.5) for multiplicity $m \geq 4$.

Theorem 3.3.3. *The proposed scheme* (3.5) *has convergence order four for multiplicity* $m \geq 4$ *for any real values of* α, β *and non-zero real values of* γ*. Moreover, error equation of the proposed scheme* (3.5) *is*

$$e_{n+1} = \frac{1}{2m^3}\left[((m+1)+4\beta)w_1^3 - 2mw_1w_2\right]e_n^4 + O(e_n^5),$$

where $\alpha,\ \beta \in \mathbb{R},\ \gamma \in \mathbb{R}\backslash\{0\}$*, and* $w_j = \dfrac{m!}{(m+j)!}\dfrac{f^{(m+j)}(\eta)}{f^{(m)}(\eta)},\ j = 1$ *and* 2*.*

Proof. Let us consider that $e_n = x_n - \eta$ and $w_j = \dfrac{m!}{(m+j)!}\dfrac{f^{(m+j)}(\eta)}{f^{(m)}(\eta)}$, $j = 1, 2, 3, 4$ denotes the error at n^{th} iteration and asymptotic error constants, respectively. Now, adopting the Taylor's expansion for the functions $f(x_n)$ and $f(\mu_n)$ about the point $x = \eta$ such that $f(\eta) = f'(\eta) = \cdots = f^{(m-1)}(\eta) = 0$ and $f^m(\eta) \neq 0$ are

$$f(x_n) = \frac{f^{(m)}(\eta)}{m!}e_n^m(1 + w_1e_n + w_2e_n^2 + w_3e_n^3 + w_4e_n^4) + O(e_n^5), \tag{3.20}$$

and

$$f(\mu_n) = \frac{f^{(m)}(\eta)}{m!}e_n^m\left[1 + \sum_{i=0}^{2}\Delta_i e_n^{i+1} + O(e_n^4)\right], \tag{3.21}$$

respectively. Here $\Delta_i = \Delta_i(m, f^{(m)}(\eta), \gamma, w_1, w_2, w_3, w_4)$. For instance, the first few coefficients can be explicitly written as $\Delta_0 = w_1, \Delta_1 = w_2$, and

$$\Delta_2 = \begin{cases} \dfrac{1}{6}\left(\gamma f^{(4)}(\eta) + 6w_3\right) & m = 4 \\ w_3 & m \geq 5. \end{cases}$$

Now, using the expressions (3.20) and (3.21) in the scheme (3.5), one can get

$$\begin{aligned} e_{z_n} = z_n - \eta =& \frac{w_1}{m}e_n^2 + \frac{1}{m^2}(2mw_2 - (1+m)w_1^2)e_n^3 + \frac{1}{m^3}\Big(\kappa_0 + 3m^2w_3 \\ &+ (1+m)^2w_1^3 - m(3m+4)w_2w_1\Big)e_n^4 + O(e_n^5), \end{aligned} \tag{3.22}$$

where

$$\kappa_0 = \begin{cases} \left(m\gamma f^{(4)}(\eta)\right) & m = 4 \\ 0 & m \geq 5. \end{cases}$$

The Taylor's expansion of the function $f(z_n)$ and expression (3.22) give

$$f(z_n) = \frac{f^{(m)}(\eta)}{m!} e_{z_n}^m (1 + w_1 e_{z_n} + w_2 e_{z_n}^2 + w_3 e_{z_n}^3 + w_4 e_{z_n}^4 + O(e_n^5)). \tag{3.23}$$

From the expressions (3.20), (3.21) and (3.23), we obtain

$$\begin{aligned} s_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}} = & \frac{w_1}{m} e_n + \frac{1}{m^2}\left(2mw_2 - (m+2)w_1^2\right) e_n^2 + \frac{1}{2m^3}\Big(\kappa_1 + (2m^2 + 7m \\ & + 7)w_1^3 - 2m(3m+7)w_1 w_2 + 6m^2 w_3\Big) e_n^3 + O(e_n^4), \end{aligned} \tag{3.24}$$

and

$$\begin{aligned} v_n = \left(\frac{f(z_n)}{f(\mu_n)}\right)^{\frac{1}{m}} = & \frac{w_1}{m} e_n + \frac{1}{m^2}(2mw_2 - (m+2)w_1^2) e_n^2 + \frac{1}{2m^3}\Big(\kappa_1 + (2m^2 + 7m \\ & + 7)w_1^3 - 2m(3m+7)w_1 w_2 + 6m^2 w_3\Big) e_n^3 + O(e_n^4), \end{aligned} \tag{3.25}$$

where

$$\kappa_1 = \begin{cases} \left(2m\gamma f^{(4)}(\eta)\right) & m = 4 \\ 0 & m \geq 5. \end{cases}$$

By using expressions (3.20)–(3.25) in the scheme (3.5), we have

$$e_{n+1} = \frac{1}{2m^3}\left[((m+1) + 4\beta)w_1^3 - 2mw_1 w_2\right] e_n^4 + O(e_n^5). \tag{3.26}$$

The Eq. (3.26) demonstrates at least fourth-order convergence of the presented family (3.5) for all non-zero real values of γ when $m \geq 4$. □

Remark 3.3.1. *Finally, the proposed class* (3.5) *can be written as follows*

$$\begin{aligned} z_n &= x_n - m\frac{f(x_n)}{f[\mu_n, x_n]}, \\ x_{n+1} &= z_n - m\frac{f(x_n)}{f[\mu_n, x_n]}\left[\frac{1 + \beta s_n}{1 + (\beta - 2)s_n}\right]\left(\frac{s_n + \nu_n + \alpha({s_n}^2 - {\nu_n}^2)}{2}\right), \end{aligned} \tag{3.27}$$

where α and β are free parameters. It is worth mentioning that the scheme (3.27) *has a simple body*

structure and satisfies all the conditions stated in Theorems 3.3.1–3.3.3 above.

Remark 3.3.2. *It can be seen that the parameters α and γ are not present in the final error equation* (3.26) *(for $m \geq 4$). Although they exist in the coefficient of e_n^5, we do not state it, as we have already attained the optimal convergence order of four. Moreover, the role of parameters α, β and γ can be seen in equations* (3.12) *and* (3.19) *for $m = 2$ and $m = 3$, respectively.*

Remark 3.3.3. *It can be observed that the iterative scheme of Behl et al. [45] can be retrieved as a particular case of the proposed family* (3.5) *for $\alpha = 0$ and $\beta = 0$.*

3.4 Numerical results

In this section, we check the numerical behavior of the derivative-free proposed scheme (3.5) on some problems. The first two real-life problems 3.4.1 and 3.4.2 are related to the case of multiple roots. Next, we choose an example 3.4.3, where roots are clustered. Furthermore, we consider two standard academic problems in Examples 3.4.4 and 3.4.5, which have multiple zeros of multiplicity $m = 100$ and $m = 77$, respectively. In the last two examples, we examine the behavior of the proposed scheme when the roots have higher multiplicity.

We consider several existing optimal Newton-like schemes for multiple roots having order four. The methods are expressed as follows:

1. **Shengguo et al. [314] method ($\mathbf{LC_1}$)**:

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - b_1\frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)}{b_2 f'(x_n) + b_3 f'(y_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.28}$$

where $b_1 = m - \frac{m^2}{2}$, $b_2 = -\frac{1}{m}$, and $b_3 = \frac{1}{m\left(\frac{m}{m+2}\right)^m}$.

2. **Soleymani et al. [329] method ($\mathbf{SB_1}$)**:

$$\begin{aligned} z_n &= x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \frac{f'(z_n)f(x_n)}{q_1(f'(z_n))^2 + q_2 f'(z_n)f'(x_n) + q_3(f'(x_n))^2}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.29}$$

where $q_1 = \frac{1}{16}m^{3-m}(m+2)^m$, $q_2 = \frac{8-m(m+2)(m^2-2)}{8m}$, and $q_3 = \frac{1}{16}(m-2)m^{m-1}(m+2)^{3-m}$.

3. **Sharma et al. [309] method (SM_1)**:

$$\begin{aligned} w_n &= u_n - m\frac{f(u_n)}{f[v_n,u_n]},\\ u_{n+1} &= w_n - \left(\frac{s_n - y_n + my_n - m^2 s_n y_n + 2ms_n y_n}{-ms_n + {s_n}^2 + 1}\right)\frac{f(u_n)}{f[v_n,u_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.30}$$

4. **Sharma et al. [309] method (SM_2)**:

$$\begin{aligned} w_n &= u_n - m\frac{f(u_n)}{f[v_n,u_n]},\\ u_{n+1} &= w_n + \left(\frac{s_n + m{s_n}^2 - (m-1)y_n(my_n - 1)}{my_n - 1}\right)\frac{f(u_n)}{f[v_n,u_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.31}$$

where $v_n = u_n + \beta f(u_n)$, $s_n = \left(\frac{f(w_n)}{f(u_n)}\right)^{\frac{1}{m}}$, $y_n = \left(\frac{f(w_n)}{f(v_n)}\right)^{\frac{1}{m}}$ and $\beta = 0.5$ in the schemes (3.30) and (3.31).

5. **Sharma et al. [310] method (MM_1)**:

$$\begin{aligned} z_n &= u_n - m\frac{f(u_n)}{f[v_n,u_n]},\\ u_{n+1} &= z_n - \frac{mh_n(1+3h_n)}{2}\left(\frac{1}{y_n} + 1\right)\frac{f(u_n)}{f[v_n,u_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.32}$$

6. **Sharma et al. [310] method (MM_2)**:

$$\begin{aligned} z_n &= u_n - m\frac{f(u_n)}{f[v_n,u_n]},\\ u_{n+1} &= z_n - \frac{mh_n}{2-6h_n}\left(\frac{1}{y_n} + 1\right)\frac{f(u_n)}{f[v_n,u_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.33}$$

where $v_n = u_n + \beta f(u_n)$, $x_n = \left(\frac{f(z_n)}{f(u_n)}\right)^{\frac{1}{m}}$, $y_n = \left(\frac{f(v_n)}{f(u_n)}\right)^{\frac{1}{m}}$, $h_n = \frac{x_n}{1+x_n}$, and $\beta = 0.5$ in the methods (3.32) and (3.33).

7. **Behl et al. [47] method (KM_1)**:

$$z_n = x_n - m\frac{f(x_n)}{f[\mu_n,x_n]},$$

$$x_{n+1} = z_n + (z_n - x_n)\frac{1+\beta s_n}{(1+(\beta-2)s_n)}\left(\frac{1}{2}y_n + \frac{s_n}{2} + \frac{Q'''(0)}{3!}{s_n}^3\right), \quad n \in \hat{\mathbb{N}}_0, \tag{3.34}$$

where $\mu_n = x_n + \gamma f(x_n)$, $y_n = \left(\frac{f(z_n)}{f(\mu_n)}\right)^{\frac{1}{m}}$, $s_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$, and $\gamma = 0.5$. Also, $\beta = \frac{1}{2}$ and $Q'''(0) = 2$.

8. **Behl et al. [49] method (AM_1)**:

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f[\mu_n, x_n]}, \\ x_{n+1} &= y_n + (y_n - x_n)\left(\frac{1}{2}v_n + 2{s_n}^2 + \frac{s}{2}\right), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.35}$$

where $\mu_n = x_n + \gamma f(x_n)$, $v_n = \left(\frac{f(y_n)}{f(\mu_n)}\right)^{\frac{1}{m}}$, $s_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, and $\gamma = 0.5$.

9. **Behl et al. [45] method (BM_1)**:

$$\begin{aligned} z_n &= x_n - m\frac{f(x_n)}{f[\mu_n, x_n]}, \\ x_{n+1} &= z_n + \frac{(z_n - x_n)(s_n + \nu_n)}{2(1-2s_n)}, \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{3.36}$$

where $\mu_n = x_n + \beta f(x_n)$, $s_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$ and $\nu_n = \left(\frac{f(z_n)}{f(\mu_n)}\right)^{\frac{1}{m}}$. Here $\beta = 0.5$.

Numerical tests are performed by using $\gamma = 1/2$ in the new methods with other parametric set of values

- LM_1: $\alpha = \frac{4}{17}, \beta = -\frac{3}{17}$,
- LM_2: $\alpha = \frac{5}{19}, \beta = -\frac{4}{19}$,
- LM_3: $\alpha = \frac{8}{9}, \beta = -\frac{2}{9}$,
- LM_4: $\alpha = 2, \beta = 0$.

In Tables 3.1–3.5, we have displayed the number of iterations (n), errors in the consecutive approximations $|x_{n+1} - x_n|$, residual errors $|f(x_n)|$, and the computational order of convergence (denoted by ρ). The expression for ρ [176] is given in (2.29).

To minimize the round-off errors, we have considered minimum 3000 significant digits. We demonstrate the value of $|x_{n+1} - x_n|$ and $|f(x_n)|$ up to first two significant digits with exponent

power in Tables 3.1–3.5. Furthermore, the computational order of convergence is shown up to first five significant digits. The following numerical examples were selected for testing:

3.4.1 Real-life examples

Example 3.4.1. ***Van der Waals equation [69]:***

We consider the following Van der Waals equation of the ideal gas [69]

$$\left(P_1 + \frac{a_1 n_1^2}{V_1}\right)(V_1 - n_1 b_1) = n_1 R_1 T_1,$$

which describes the behavior of a real gas having two constants a_1 and b_1. The volume V_1 can be obtained by solving the following nonlinear equation

$$f_1(x) = x^3 - 5.22x^2 + 9.0825x - 5.2675.$$

The multiple root is $\eta = 1.75$ of multiplicity $m = 2$. In Table 3.1, we include the computational results for several iterative schemes with $x_0 = 2$.

Table 3.1: Numerical comparisons on Example 3.4.1

Methods	n	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert f_1(x_3)\rvert$	ρ	CPU-Time
LC_1	3	2.8×10^{-2}	7.6×10^{-4}	4.3×10^{-9}	5.5×10^{-19}	3.0792	0.437
SB_1	3	2.8×10^{-2}	7.6×10^{-4}	4.3×10^{-9}	5.5×10^{-19}	3.0792	0.313
SM_1	3	2.7×10^{-2}	6.7×10^{-4}	2.1×10^{-9}	1.3×10^{-19}	3.1312	0.327
SM_2	3	2.4×10^{-2}	3.7×10^{-4}	1.2×10^{-10}	4.7×10^{-22}	3.3247	0.468
MM_1	3	3.7×10^{-2}	2.6×10^{-3}	1.3×10^{-6}	5.2×10^{-14}	2.4417	0.282
MM_2	3	2.3×10^{-2}	3.4×10^{-4}	8.6×10^{-11}	2.2×10^{-22}	3.3485	0.360
KM_1	3	2.6×10^{-2}	5.6×10^{-4}	9.1×10^{-10}	2.5×10^{-20}	3.1956	0.313
AM_1	3	3.2×10^{-2}	1.5×10^{-3}	9.0×10^{-8}	2.4×10^{-16}	2.7748	0.328
BM_1	3	2.3×10^{-2}	3.4×10^{-4}	8.6×10^{-11}	2.2×10^{-22}	3.3485	0.265
LM_1	3	2.2×10^{-2}	2.4×10^{-4}	1.6×10^{-11}	7.9×10^{-24}	3.4441	0.297
LM_2	3	2.1×10^{-2}	2.2×10^{-4}	1.1×10^{-11}	3.8×10^{-24}	3.4639	0.298
LM_3	3	2.1×10^{-2}	2.0×10^{-4}	8.0×10^{-12}	1.9×10^{-24}	3.4813	0.297
LM_4	3	2.2×10^{-2}	2.9×10^{-4}	5.0×10^{-11}	7.4×10^{-23}	3.3831	0.282

Example 3.4.2. ***Planck's radiation law [63]:***

Let us consider the problem related to Planck's radiation law [63], whose mathematical form is defined as follows

$$H(y) = \frac{8\pi chy^{-5}}{e^{\frac{ch}{ykT}} - 1},$$

where $y,\ T,\ k,\ c$, and h denote the wavelength of radiation, absolute temperature of the black-body, Boltzmann constant, speed of light, and Planck's constant, respectively.

In thermal equilibrium, this law provides the spectral distribution of radiation at a particular temperature from a black body. For this purpose, we solve the equation $H'(y) = 0$ for evaluating the wavelength y, corresponding to the maximum energy density of $H(y)$.

After algebraic calculations, we obtain the following equation

$$\frac{\left(\frac{ch}{ykT}\right) e^{\frac{ch}{ykT}}}{e^{\frac{ch}{ykT}} - 1} = 5.$$

Further, we can formulate it in the form of a nonlinear equation as follows

$$f_2(x) = \left(e^{-x} - 1 + \frac{x}{5}\right)^3,$$

where $x = \dfrac{ch}{ykT}$. The multiple root for this problem is $\eta = 4.96511423174428$ of multiplicity three. Now, one can compute the wavelength y using $x = \dfrac{ch}{ykT}$. We test this problem for the initial guess $x_0 = 5.4$ and the corresponding results are displayed in Table 3.2.

3.4.2 Academic examples

Example 3.4.3. ***Clustered root problem [379]:***

Now, we consider a clustered root problem given by

$$f_3(x) = (x-1)^{20}(x-2)^{15}(x-3)^{10}(x-4)^5.$$

The function f_3 has four multiple zeros $x = 1, 2, 3$ and 4 with multiplicities twenty, fifteen, ten, and five, respectively. All multiple zeros are clustered closely together. Our desired root is $x = 1$ with multiplicity 20. The numerical performance of methods is depicted in Table 3.3 using an initial guess $x_0 = 0.8$.

Table 3.2: Numerical comparisons for Example 3.4.2

Methods	n	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert f_2(x_3)\rvert$	ρ	CPU-Time
LC_1	3	2.0×10^{-5}	1.2×10^{-22}	1.5×10^{-91}	2.5×10^{-275}	3.9999	0.328
SB_1	3	2.0×10^{-5}	1.2×10^{-22}	1.5×10^{-91}	2.6×10^{-275}	3.9999	0.407
SM_1	3	1.9×10^{-6}	1.1×10^{-27}	1.5×10^{-112}	2.3×10^{-338}	4.0000	0.374
SM_2	3	1.8×10^{-6}	8.1×10^{-28}	3.6×10^{-113}	3.3×10^{-340}	4.0000	0.438
MM_1	3	2.8×10^{-6}	7.9×10^{-27}	5.2×10^{-109}	1.0×10^{-327}	4.0000	0.359
MM_2	3	1.7×10^{-6}	6.8×10^{-28}	1.8×10^{-113}	4.0×10^{-341}	4.0000	0.453
KM_1	3	1.8×10^{-6}	8.6×10^{-28}	4.7×10^{-113}	7.7×10^{-340}	4.0000	0.422
AM_1	3	2.2×10^{-6}	2.3×10^{-27}	3.1×10^{-111}	4.0×10^{-334}	4.0000	0.391
BM_1	3	1.7×10^{-6}	6.8×10^{-28}	1.8×10^{-113}	4.0×10^{-341}	4.0000	0.376
LM_1	3	1.7×10^{-6}	6.0×10^{-28}	1.0×10^{-113}	7.4×10^{-342}	4.0000	0.375
LM_2	3	1.6×10^{-6}	5.8×10^{-28}	9.0×10^{-114}	5.3×10^{-342}	4.0000	0.344
LM_3	3	1.6×10^{-6}	5.6×10^{-28}	8.1×10^{-114}	3.8×10^{-342}	4.0000	0.359
LM_4	3	1.7×10^{-6}	6.4×10^{-28}	1.4×10^{-113}	1.9×10^{-341}	4.0000	0.389

Table 3.3: Numerical comparisons on Example 3.4.3

Methods	n	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert f_3(x_3)\rvert$	ρ	CPU-Time
LC_1	3	2.6×10^{-3}	2.2×10^{-10}	1.1×10^{-38}	1.3×10^{-754}	3.9982	0.328
SB_1	3	3.4×10^{-3}	7.5×10^{-10}	1.9×10^{-36}	6.5×10^{-710}	3.9977	0.375
SM_1	3	1.2×10^{-2}	1.3×10^{-6}	3.4×10^{-12}	1.1×10^{-224}	1.3914	0.296
SM_2	3	1.3×10^{-2}	1.4×10^{-6}	4.1×10^{-12}	5.5×10^{-223}	1.3931	0.312
MM_1	3	4.7×10^{-3}	5.7×10^{-9}	1.3×10^{-32}	7.0×10^{-633}	3.9953	0.359
MM_2	3	9.0×10^{-4}	7.3×10^{-13}	3.2×10^{-49}	2.7×10^{-965}	3.9997	0.359
KM_1	3	1.4×10^{-3}	7.4×10^{-12}	5.9×10^{-45}	6.4×10^{-880}	3.9994	0.281
AM_1	3	3.0×10^{-3}	5.2×10^{-10}	4.7×10^{-37}	6.0×10^{-722}	3.9978	0.390
BM_1	3	9.0×10^{-4}	7.3×10^{-13}	3.2×10^{-49}	2.7×10^{-965}	3.9997	0.360
LM_1	3	6.3×10^{-4}	1.1×10^{-13}	9.0×10^{-53}	2.7×10^{-1036}	3.9998	0.359
LM_2	3	5.8×10^{-4}	6.6×10^{-14}	1.1×10^{-53}	1.6×10^{-1054}	3.9998	0.328
LM_3	3	5.6×10^{-4}	5.5×10^{-14}	5.0×10^{-54}	2.6×10^{-1061}	3.9998	0.328
LM_4	3	5.6×10^{-4}	5.5×10^{-14}	5.0×10^{-54}	2.6×10^{-1061}	3.9998	0.328

Example 3.4.4. ***Standard academic problem [44]:***

We consider a standard academic problem given by

$$f_4(x) = \Big((x-1)^3 - 1\Big)^{100},$$

where f_4 has a zero at $x = 2$ with multiplicity 100. In Table 3.4, we display the computational results by taking $x_0 = 2.1$.

Table 3.4: Numerical comparisons on Example 3.4.4

Methods	n	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert f_4(x_3)\rvert$	ρ	CPU-Time
LC_1	3	2.1×10^{-4}	6.1×10^{-15}	4.5×10^{-57}	8.8×10^{-5588}	3.9999	0.250
SB_1	3	2.9×10^{-4}	2.9×10^{-14}	3.0×10^{-54}	6.6×10^{-5305}	3.9999	0.282
SM_1	3	1.6×10^{-3}	1.6×10^{-9}	5.2×10^{-18}	1.4×10^{-1681}	1.4141	0.329
SM_2	3	1.6×10^{-3}	1.6×10^{-9}	5.4×10^{-18}	1.1×10^{-1679}	1.4144	0.266
MM_1	3	4.9×10^{-4}	5.6×10^{-13}	9.7×10^{-49}	2.9×10^{-4754}	3.9997	0.312
MM_2	3	5.3×10^{-5}	5.2×10^{-18}	5.0×10^{-70}	5.9×10^{-6883}	4.0000	0.328
KM_1	3	9.9×10^{-5}	1.3×10^{-16}	3.5×10^{-64}	1.7×10^{-6298}	4.0000	0.267
AM_1	3	2.8×10^{-4}	2.9×10^{-14}	3.3×10^{-54}	6.0×10^{-5301}	3.9999	0.343
BM_1	3	5.3×10^{-5}	5.2×10^{-18}	5.0×10^{-70}	5.9×10^{-6883}	4.0000	0.281
LM_1	3	2.9×10^{-5}	2.1×10^{-19}	6.0×10^{-76}	4.1×10^{-7475}	4.0000	0.282
LM_2	3	2.4×10^{-5}	7.9×10^{-20}	9.3×10^{-78}	5.0×10^{-7656}	4.0000	0.297
LM_3	3	2.2×10^{-5}	5.3×10^{-20}	1.8×10^{-78}	6.5×10^{-7728}	4.0000	0.312
LM_4	3	5.3×10^{-5}	5.2×10^{-18}	5.0×10^{-70}	5.9×10^{-6883}	4.0000	0.329

Example 3.4.5. ***Higher multiplicity problem [268]:***

Let us consider another problem related with higher multiplicity, which is given as follows:

$$f_5(x) = (x^5 - x^3 + x + 1)^{77}.$$

The above equation has a multiple zero at $\eta = -1$ with multiplicity 77. Using $x_0 = -1.1$, we include numerical results in the Table 3.5.

In Tables 3.1–3.5, we present the numerical results of different methods for computing the multiple roots of nonlinear functions f_1, f_2, f_3, f_4, and f_5, respectively. Therefore, the numerical results in Tables 3.1–3.5 reveal that the proposed methods LM_1, LM_2, LM_3, and LM_4 perform

Table 3.5: Numerical comparisons on Example 3.4.5

Methods	n	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	$\lvert f_5(x_3)\rvert$	ρ	CPU-Time
LC_1	3	1.5×10^{-3}	1.5×10^{-10}	1.6×10^{-38}	4.1×10^{-2874}	3.9984	0.328
SB_1	3	2.5×10^{-3}	1.5×10^{-9}	2.0×10^{-34}	1.1×10^{-2558}	3.9976	0.312
SM_1	–	Diverges	–	–	–	–	0.375
SM_2	–	Diverges	–	–	–	–	0.313
MM_1	3	3.6×10^{-3}	1.8×10^{-8}	1.3×10^{-29}	5.4×10^{-2188}	3.9929	0.374
MM_2	3	4.4×10^{-4}	2.1×10^{-13}	1.0×10^{-50}	3.8×10^{-3813}	3.9998	0.375
KM_1	3	8.9×10^{-4}	9.0×10^{-12}	9.5×10^{-44}	9.8×10^{-3277}	3.9993	0.250
AM_1	3	2.3×10^{-3}	1.5×10^{-9}	3.1×10^{-34}	2.8×10^{-2544}	3.9969	0.359
BM_1	3	4.4×10^{-4}	2.1×10^{-13}	1.0×10^{-50}	3.8×10^{-3813}	3.9998	0.297
LM_1	3	1.9×10^{-4}	1.5×10^{-15}	5.8×10^{-60}	2.1×10^{-4525}	4.0000	0.312
LM_2	3	1.4×10^{-4}	1.2×10^{-16}	7.6×10^{-65}	5.9×10^{-4900}	4.0002	0.266
LM_3	3	1.2×10^{-4}	1.1×10^{-17}	7.3×10^{-70}	1.1×10^{-5287}	4.0012	0.312
LM_4	3	4.4×10^{-4}	2.1×10^{-13}	1.0×10^{-50}	3.8×10^{-3813}	3.9998	0.281

better in comparison to the existing robust methods as they offer lesser absolute errors in the consecutive iterations and absolute residual errors than the other methods.

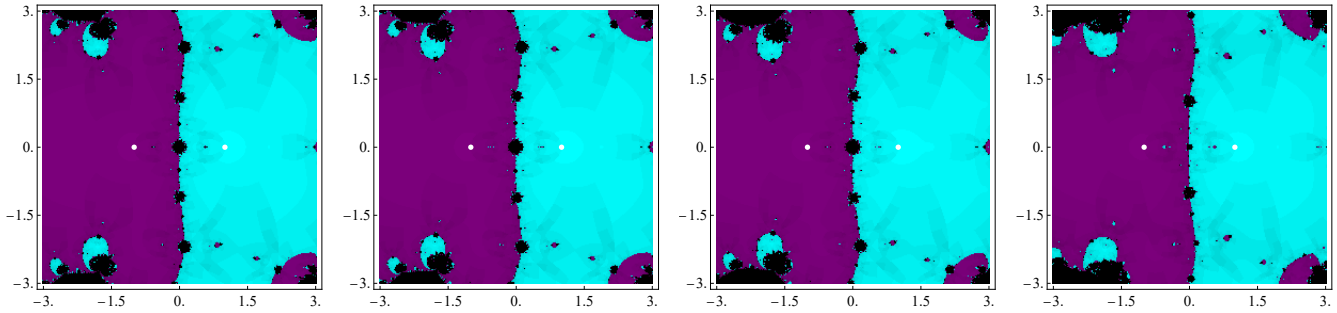
In addition to the parameters considered, we can generate some new methods from the proposed scheme (3.5) by considering the different parametric values like $\left\{\alpha = \frac{1}{8}, \beta = -\frac{1}{5}\right\}$, $\left\{\alpha = \frac{11}{12}, \beta = -\frac{5}{23}\right\}$, $\left\{\alpha = \frac{54}{10}, \beta = -\frac{3}{14}\right\}$, and $\left\{\alpha = \frac{49}{10}, \beta = -\frac{224}{100}\right\}$ to have even more better results.

3.5 Basins of attraction

Here, we illustrate the relevant information regarding the convergence and stability of the iterative scheme by analyzing their basins of attraction obtained by employing the approach to some complex polynomials $p(z)$. Basins of attraction are generally assumed to be visual geometrical tools for comparing iterative techniques, which explain how a specific scheme functions for different initial points. A concise review of some basic ideas related to the visual tool is provided in the introduction 1.5, while more information can be found in [352].

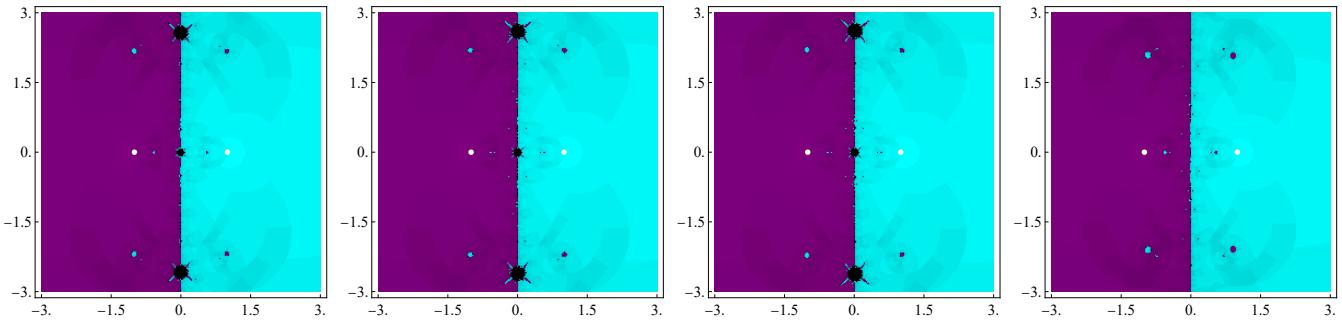
Three polynomials with multiple zeros are used for the complex dynamics of the new methods LM_1, LM_2, LM_3, and LM_4. The problems are given below.

Test Problem 1: For the testing, let us suppose the polynomial $p_1(z) = (z^2 - 1)^2$ carrying two zeros $\{-1, 1\}$ with multiplicity $m = 2$. Basins for the polynomial are displayed within Figures 3.1–3.3 corresponding to $\gamma = 0.01, 10^{-4}, 10^{-6}$, respectively. One color is allotted to each starting point in the basins of attraction. Particularly, we assign the colors purple and cyan to the zeros -1 and 1, respectively.



(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.1: Convergence planes of the proposed methods for $\gamma = 0.01$ in the test problem 1.

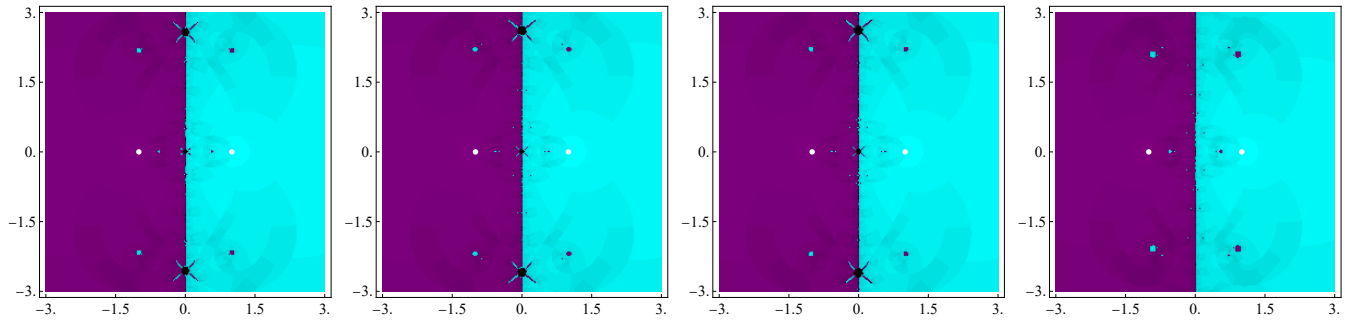


(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.2: Convergence planes of the proposed methods for $\gamma = 0.0001$ in the test problem 1.

Test Problem 2: Consider the polynomial $p_2(z) = (z^3 + \frac{1}{4}z)^3$ carrying three zeros $\left\{-\frac{i}{2}, \frac{i}{2}, 0\right\}$ with multiplicity $m = 3$. Basins of attractors are estimated by the proposed scheme for the polynomial, as depicted in Figures 3.4–3.6, corresponding to chosen values for the parameter γ, $0.01, 10^{-4}, 10^{-6}$. The corresponding basins of zeros are distinguished via a color allotted to them. Particularly, we assign the colors purple, cyan, and green to the values $-\frac{i}{2}, \frac{i}{2}$, and 0, respectively.

Test Problem 3: Let us examine the function $p_3(z) = (z^3 - z)^4$ having three zeros $\{-1, 1, 0\}$ with multiplicity $m = 4$. Basins corresponding to the zeros are depicted in Figures 3.7–3.9, for

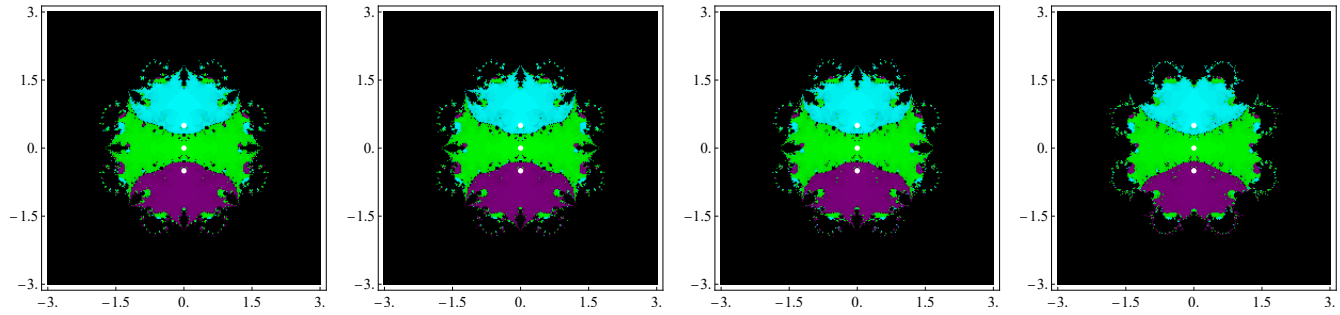


(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.3: Convergence planes of the proposed methods for $\gamma = 0.000001$ in the test problem 1.

Table 3.6: Convergence measures of methods for test function $p_1(z)$.

γ	Methods	LM_1	LM_2	LM_3	LM_4
10^{-2}	M/P	4.18814	4.21450	4.43860	4.56074
	NC(%)	4.53093	4.75094	5.79856	5.66549
	M_C/C	3.20042	3.17774	3.17295	3.3332
10^{-4}	M/P	3.22445	3.22109	3.21875	3.22129
	NC(%)	0.514498	0.567696	0.586695	0.043906
	M_C/C	3.11183	3.09675	3.0902	3.21172
10^{-6}	M/P	3.16681	3.15637	3.15320	3.22626
	NC(%)	0.184292	0.203671	0.215071	0.007570
	M_C/C	3.12649	3.11179	3.10611	3.22461

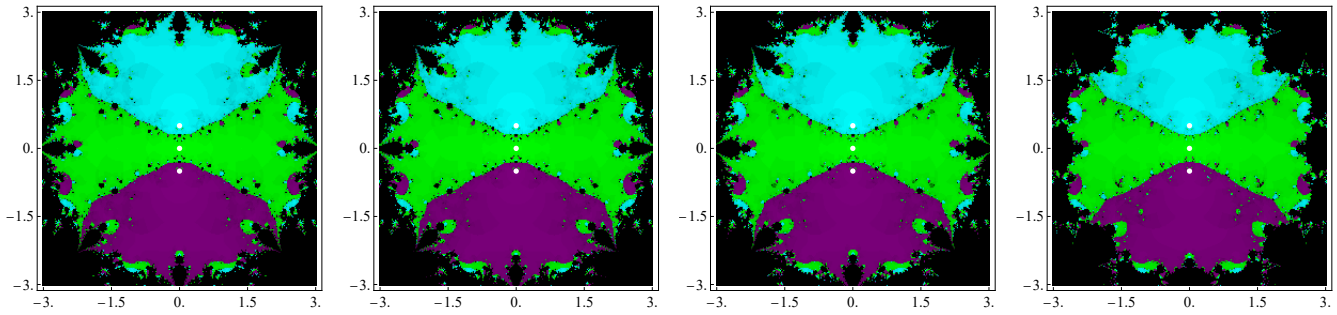


(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.4: Convergence planes of the proposed methods for $\gamma = 0.01$ in the test problem 2.

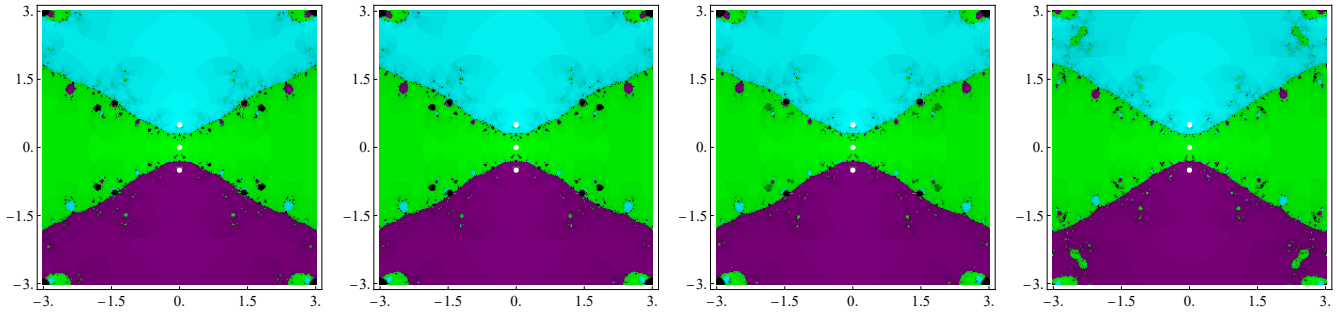
the particular values of $\gamma = 0.01, 0.0001, 0.000001$ respectively. One color is allotted to each initial point in the basins with zeros $-1, 1$, and 0. Particularly, we assign the colors purple, cyan, and green to the values $-1, 1, 0$, respectively.

In Table 3.6–3.8, we present convergence measures of the proposed methods by considering



(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.5: Convergence planes of the proposed methods for $\gamma = 0.0001$ in the test problem 2.



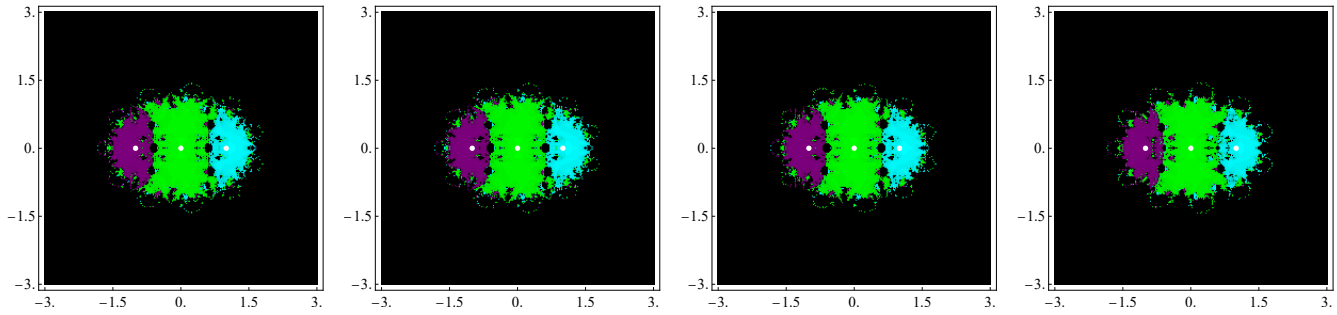
(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.6: Convergence planes of the proposed methods for $\gamma = 0.000001$ in the test problem 2.

Table 3.7: Convergence measures of methods for test function $p_2(z)$

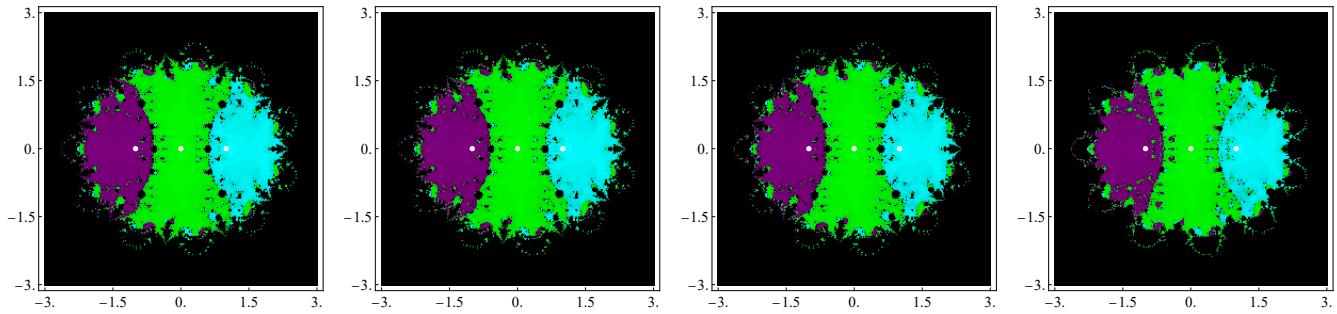
γ	Methods	LM_1	LM_2	LM_3	LM_4
10^{-2}	M/P	20.4717	20.4703	20.4913	20.6493
	NC(%)	78.5465	78.5594	78.5852	79.4471
	M_C/C	3.89238	3.87314	3.94567	3.81900
10^{-4}	M/P	11.2258	11.1851	11.2568	11.9144
	NC(%)	33.0183	32.8724	33.0328	36.0853
	M_C/C	4.43589	4.41994	4.47770	4.52650
10^{-6}	M/P	4.57397	4.53642	4.49333	4.81418
	NC(%)	1.05982	1.15672	0.678284	0.393647
	M_C/C	4.35517	4.29694	4.35329	4.73441

different values of the parameter γ. The first column (M/P) represents the mean value of iterations required by the algorithm per point to reach the desired root. Otherwise, that point comes out to be non-convergent. The second column NC(%) defines the percentage of non-convergent points. The black zones indicate non-convergent points that can be seen in the fractal images (Figs. 3.1–3.9)



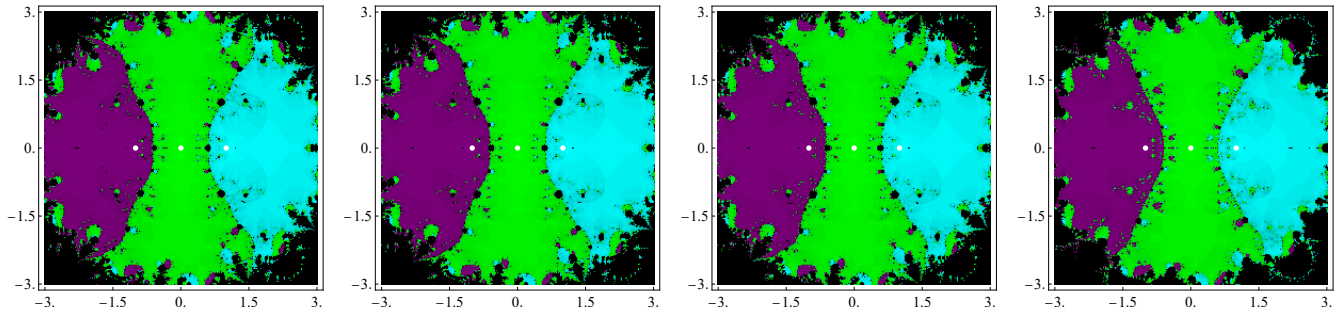
(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.7: Convergence planes of the proposed methods for $\gamma = 0.01$ in the test problem 3.



(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.8: Convergence planes of the proposed methods for $\gamma = 0.0001$ in the test problem 3.



(a) LM_1 (b) LM_2 (c) LM_3 (d) LM_4

Figure 3.9: Convergence planes of the proposed methods for $\gamma = 0.000001$ in the test problem 3.

for different test functions.

Note that the value of M/P is dependent on the non-convergent points since these points always contributed with the maximum number of 25 allowed iterations. Usually, the convergent points are reached very quickly while working with higher-order multi-point methods. Finally, we include another column (M_C/C) which depicts the mean value of iterations per convergent point to minimize the round-off errors.

Table 3.8: Convergence measures of methods for test function $p_3(z)$

γ	Methods	LM_1	LM_2	LM_3	LM_4
10^{-2}	M/P	21.9461	21.9571	21.9974	22.0097
	NC(%)	85.8923	85.9528	86.1572	86.2042
	M_C/C	3.35309	3.33810	3.30903	3.32474
10^{-4}	M/P	17.5385	17.5385	17.5993	17.6877
	NC(%)	64.9225	64.8758	65.2717	65.4620
	M_C/C	3.72863	3.72167	3.68965	3.82812
10^{-6}	M/P	8.25922	8.21239	8.32539	8.56264
	NC(%)	20.3637	20.2062	20.7603	21.4250
	M_C/C	3.98477	3.96501	3.96060	4.08688

It can be seen that the estimation of parameter γ plays a crucial role in the convergence behavior of the proposed family (3.5). Due to this reason, we have selected different values of γ to visualize the basins. The basin attractors indicate that they are becoming wider as the parameter γ decreases. On the contrary, the fractals shrink for large values of the parameter γ. Moreover, the black areas indicate the divergence regions, which are also diminishing as the value of γ decreases. Finally, we conclude that the convergence of the proposed methods is significantly improved for smaller values of the parameter γ.

3.6 Concluding remarks

We present derivative-free iterative methods for approximating multiple roots of nonlinear equations. The theoretical analysis is provided for $m = 2$ and $m = 3$. It is further generalized for $m \geq 4$, which confirms that the proposed scheme is optimal, having convergence order four. It is noticeable that some already existing methods are special cases of the proposed class. In order to verify stability, we have also displayed the basins of attraction of the newly proposed methods for various parameter values in the complex plane. The performance of the proposed methods is superior, as demonstrated by the numerical results. The numerical experiments further reveal that the presented derivative-free family works very well for higher multiplicity corresponding to the considered test functions, as compared to the other methods. Thus, we can conclude that the proposed class would be a useful alternative for numerically computing multiple zeros of nonlinear functions.

After developing a robust, derivative-free scheme for finding multiple roots of scalar nonlinear equations, the next important step is to extend these ideas to higher-dimensional cases. The

challenges multiply when transitioning from scalar to systems, requiring a focus on computational efficiency and a rigorous theoretical base that extends beyond the limitations of the Taylor series. In the next chapter, we address these challenges by developing a new family of multi-step iterative schemes for nonlinear systems. We shift our analysis to a Banach space setting, which enables us to provide a more general convergence theory that relies solely on first-order derivatives and to formally establish the existence and uniqueness of solutions.

Chapter 4

Family of multi-step vectorial iterative methods for solving nonlinear systems

In this chapter, we develop multi-step vectorial iterative schemes for solving nonlinear systems, achieving fourth and sixth-order convergence. The proposed methods are designed to minimize computational costs by employing a single inverse operator and reducing the number of functional evaluations per iteration. Furthermore, we generalize the sixth-order three-step scheme into a $(q+1)$-step family, increasing the convergence order to $2q+2$. While standard local convergence analysis based on Taylor series expansion is common, but it has limitations, as it requires the use of higher-order derivatives. To overcome this limitation, we conduct the theoretical analysis in a Banach space setting that relies solely on first-order derivatives. The existence of a unique solution is guaranteed within a specific domain, whose radius of convergence is formally obtained using Lipschitz constants. Furthermore, we test the theoretical results on several numerical examples to check the performance and stability of the proposed methods in comparison to existing counterparts.

4.1 Introduction

Nonlinear problems, which describe complex phenomena like turbulent fluid flow, heat transfer with sharp gradients, and reaction-diffusion processes in catalysts, are mathematically governed by boundary value problems (BVPs) [289], ordinary/partial differential equations (ODEs/PDEs) [93, 278, 279, 280, 282, 316, 317, 318, 355, 357], or integral equations. For instance, the concentration

The contents of this chapter are published in Journal of Mathematical Chemistry, Vol. 64, Article ID 8, 2026 (SCIE, I.F.: 2.0)

distribution of the diffusion of reactants in a spherical catalyst pellet is governed by the nonlinear ODE:

$$\begin{cases} \dfrac{\mathcal{D}_f}{r^2}\dfrac{d}{dr}\left(r^2\dfrac{dC}{dr}\right) = k_r R(C),\ 0 < r < R_p, \\ \dfrac{dC}{dr}(0) = 0,\ C(R_p) = C_s, \end{cases} \tag{4.1}$$

where C represents reactant concentration, $R(C)$ is the reaction rate function, $\mathcal{D}_f$ denotes effective diffusivity, k_r is the reaction rate constant, and r indicates radial position. Similarly, oscillatory systems like the Van der Pol oscillator [93] exhibit nonlinear dynamics, described by

$$\begin{cases} \dfrac{d^2u}{d\tau^2} - \mu(u^2-1)\dfrac{du}{d\tau} + u = 0,\ \mu > 0, \\ u(0) = 0,\ u(2) = 1. \end{cases} \tag{4.2}$$

Moving to PDEs, Burger's equation [289] serves as a fundamental model for nonlinear wave interactions and viscous flow, having a mathematical form

$$\begin{cases} \dfrac{\partial^2 u}{\partial l^2} - \dfrac{\partial u}{\partial r} + u\dfrac{\partial u}{\partial l} + R(l,r) = 0,\ (l,r) \in [0,1]\times[0,1], \\ u(0,r) = u(1,r) = 0, \\ u(l,0) = 10l(l-1), \quad u(l,1) = \dfrac{10l(l-1)}{e}, \end{cases} \tag{4.3}$$

where $R(l,r) = -10\left[10l(1-3l+2l^2) + e^r(2-l+l^2)\right]/e^{2r}$ is a source term, and the solution is constrained by mixed boundary conditions. Another example is the nonlinear heat conduction problem [289]:

$$\begin{cases} \dfrac{\partial^2 u}{\partial l^2} - \dfrac{\partial u}{\partial l} - \dfrac{\partial u}{\partial r} + u^2 = \hat{H}(l,r),\ l \in [0,1],\ r \geq 0, \\ u(l,0) = \sin(l\pi), \\ u(0,r) = 0,\ u(1,r) = 0, \end{cases} \tag{4.4}$$

with $\hat{H}(l,r) = -\left((\pi^2-2)\sin(\pi l) + \pi\cos(\pi l)\right)/e^r$ introducing additional complexity.

The challenges posed by nonlinear models have spurred significant research into numerical methods across scientific and engineering disciplines. Classical approaches for solving ODE-based BVPs include Runge–Kutta methods [67] and Adams–Bashforth multi-step methods [224], which offer efficiency through the reuse of previously computed values. While these methods are well-established for many problems, they can encounter certain difficulties with stiff systems or when applied as

shooting methods. This motivates alternative approaches, such as the iterative scheme developed in this work, which solves the discretized system iteratively.

A central problem involves determining a solution vector $X^* = (X_1^*, X_2^*, \ldots, X_k^*)$ such that it satisfies the nonlinear system (1.3). These systems arise naturally when discretizing BVPs, modeling equilibrium states in dynamical systems, or PDEs. An alternative approach reformulates (1.3) as a fixed-point problem, seeking a function $\Phi_F : \mathcal{D} \subseteq \mathbb{R}^k \to \mathbb{R}^k$ such that

$$X^{(n+1)} = \Phi_F(X^{(n)}),\ n = 0, 1, 2, \ldots, \tag{4.5}$$

where iterations converge to the solution or fixed point X^*. Recent studies [20, 189, 338] have developed efficient algorithms to approximate the solutions within specified tolerances. Among these, Newton's method [263] remains classical due to its quadratic convergence under standard conditions. To achieve higher-order convergence, modified Newton-type methods introduce additional computations, e.g., new functional evaluations, Jacobian updates, or inverse operators. For instance, Cordero et al. [89] proposed a fourth-order scheme, given by

$$\begin{aligned} Y^{(n)} &= \Phi_1^{(1)}(X^{(n)}) = X^{(n)} - \frac{2}{3}[F'(X^{(n)})]^{-1}F(X^{(n)}), \\ X^{(n+1)} &= \Phi_2^{(4)}(X^{(n)}, Y^{(n)}) \\ &= X^{(n)} - \frac{1}{2}[3F'(Y^{(n)}) - F'(X^{(n)})]^{-1}(3F'(Y^{(n)}) + F'(X^{(n)})) \\ &\quad \times [F'(X^{(n)})]^{-1}F(X^{(n)}), \end{aligned} \tag{4.6}$$

requiring two Jacobians and two inverse operators per iteration. Similarly, Babajee et al. [33] derived a fourth-order technique using weight functions. Further refinements include in the work of Narang et al. [253], who combined this approach with Nedzhibov's scheme [257] to preserve fourth-order convergence while reducing computational costs. They later extended it to sixth order by incorporating an additional function evaluation. However, due to two matrix inversions, there is an immediate increase in the computational cost and a decrease in computational efficiency. Recently, Dehghan and Shirilord [108] suggested another sixth-order iterative technique, which includes three functional evaluations, two Jacobians, and two inverse operators. Going further, Sharma and Kumar [303] presented a generalized scheme with $(q+1)$-steps using Newton-Jaratt composition requiring q functional evaluations, two Jacobians, and one matrix inverse operator to attain $2q+1,\ q \in \mathbb{N}$ order of convergence.

The primary aim when developing efficient iterative algorithms is to maximize the convergence order while minimizing the number of function evaluations. However, the most evident obstruction in solving nonlinear systems is the need for matrix inversion computation, which requires a substantial amount of computational work. Therefore, utilizing the minimum number of such inverse operators will be reasonable. Keeping this in mind, we attempted to propose a two-step iterative scheme using one functional evaluation, two Fréchet operators, and one matrix inverse operator to achieve at least fourth-order convergence. Further, the convergence order is increased to six by including one extra function evaluation. Due to certain limitations of the Taylor series expansion, we carry out an extensive local convergence analysis for each developed class in the Banach space setting. Furthermore, in pursuit of higher convergence order, the approach has been extended to incorporate $(q+1)$-steps, where the convergence order increases to $2q+2, q \in \mathbb{N}$. One noteworthy aspect of the proposed methods is that they enhance the convergence order by two while using only one auxiliary function per step. The proposed schemes have also been tested on some real-life models to determine their practicality.

4.2 Formulation of new schemes

This section presents the formulation and analysis of two new iterative families for solving nonlinear vector problems.

4.2.1 Fourth-order scheme

We propose a two-step iterative family that enhances the convergence rate of Newton's technique (1.59) while reducing the computational cost per iteration by minimizing function and Jacobian evaluations, as well as the use of inverse operators, given by

$$\begin{aligned} Y^{(n)} &= X^{(n)} - \frac{2}{3} v_0(X^{(n)}), \\ X^{(n+1)} &= \Phi_4^{(4)}(X^{(n)}, Y^{(n)}) \\ &= X^{(n)} - \left(\sigma I + \frac{\eta}{2\delta}\mathcal{H}(X^{(n)})\right) W(\mathcal{H}(X^{(n)})) v_0(X^{(n)}), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{4.7}$$

where $v_0(X^{(n)}) = [F'(X^{(n)})]^{-1}F(X^{(n)})$, $\mathcal{H}(X^{(n)}) = I - [F'(X^{(n)})]^{-1}F'(Y^{(n)})$, and $W(\mathcal{H}(X^{(n)}))$ is the weight function. Also, σ, δ are two non-zero free parameters, whereas η can take any value in $\mathbb{R}$. Here, I denotes an k-order identity matrix. The first step of the proposed scheme (4.7) is based on

the well-known the Jaratt technique; however, the next sub-step is primarily based on the weight function approach.

We now establish the convergence analysis of the proposed fourth-order family (4.7).

Theorem 4.2.1. *Suppose the function $F : \mathcal{D} \to \mathbb{R}^k$, defined on an open convex domain $\mathcal{D} \subseteq \mathbb{R}^k$, has a zero at X^* and is r-times Fréchet differentiable in a neighborhood of X^*. Furthermore, assume the derivative $F'(X)$ is continuous and non-singular at $X = X^*$. Then, the iteration (4.7) defines a sequence $\{X^{(n)}\}_{n=0}^{\infty}$ that converges to X^* with order at least four, provided the initial iterate $X^{(0)}$ is chosen sufficiently close to the root. The following equation describes the asymptotic error behavior*

$$\xi^{(n+1)} = \left(\frac{(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 - 8\lambda^3)}{27\sigma^3\delta^3}A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\right)(\xi^{(n)})^4 + O((\xi^{(n)})^5),$$

where $\sigma, \delta \in \mathbb{R}\backslash\{0\}$ and $\eta \in \mathbb{R}$.

Proof. Let $\xi^{(n)} = X^{(n)} - X^*$. By using Lemma (1.2.1), the function $F(X^{(n)})$ can be expanded in a neighborhood of zero X^*, one obtains

$$F(X^{(n)}) = F(X^*) + F'(X^*)\xi^{(n)} + \frac{1}{2!}F''(X^*)(\xi^{(n)})^2 + \frac{1}{3!}F'''(X^*)(\xi^{(n)})^3 + O((\xi^{(n)})^4). \tag{4.8}$$

For an actual root X^*, we have $F(X^*) = 0$ which further leads us to

$$F(X^{(n)}) = F'(X^*)[\xi^{(n)} + A_2(\xi^{(n)})^2 + A_3(\xi^{(n)})^3 + A_4(\xi^{(n)})^4 + O((\xi^{(n)})^5)], \tag{4.9}$$

where $A_i = \dfrac{1}{i!}\Gamma F^{(i)}(X^*) \in L_i(\mathbb{R}^k, \mathbb{R}^k)$ and $L_i(\mathbb{R}^k, \mathbb{R}^k)$ for $i = 2, 3, \ldots$ denotes the set of bounded i-linear functions. Also, $F^{(i)}(X^*) \in L_i(\mathbb{R}^k \times \overset{i\text{-times}}{\cdots} \times \mathbb{R}^k, \mathbb{R}^k), \Gamma = [F'(X^*)]^{-1} \in L(\mathbb{R}^k, \mathbb{R}^k)$, $(\xi^{(n)})^k = (\xi^{(n)}, \xi^{(n)}, \overset{k\text{-times}}{\cdots}, \xi^{(n)})$, $\quad \xi^{(n)} \in \mathbb{R}^k$ and $A_i(\xi^{(n)})^j \in \mathbb{R}^k$. Now, the Fréchet derivative function $F'(X^{(n)})$ which is determined from (4.9), is given by

$$F'(X^{(n)}) = F'(X^*)\left[I + \sum_{j=2}^{6} jA_j(\xi^{(n)})^{j-1} + O((\xi^{(n)})^6)\right]. \tag{4.10}$$

Taking inverse of Jacobian matrix $F'(X^{(n)})$ as follows

$$[F'(X^{(n)})]^{-1} = \left[I + \sum_{j=1}^{5} C_j(\xi^{(n)})^j + O((\xi^{(n)})^6)\right]\Gamma, \tag{4.11}$$

where

$$\begin{aligned}
C_1 =& -2A_2,\ C_2 = -3A_3 + 4A_2^2,\ C_3 = -4A_4 + 6A_3A_2 + 6A_2A_3 - 8A_2^3,\\
C_4 =& -5A_5 + 16A_2^4 - 12A_3A_2^2 - 12A_2A_3A_2 - 12A_2^2A_3 + 8A_4A_2 + 8A_2A_4 + 9A_3^2,\\
C_5 =& 10A_2A_5 + 12A_3A_4 + 12A_4A_3 + 10A_5A_2 - 16A_2^2A_4 - 18A_2A_3^2 - 16A_2A_4A_2 - 18A_3A_2A_3\\
&- 18A_3^2A_2 - 16A_4A_2^2 + 24A_2^3A_3 + 24A_2^2A_3A_2 + 24A_2A_3A_2^2 + 24A_3A_2^3 - 32A_5^5 - 6A_6.
\end{aligned}$$

Post-multiplying (4.11) by (4.9), one gets

$$v_0(X^{(n)}) = [F'(X^{(n)})]^{-1}F(X^{(n)}) = \xi^{(n)} + \sum_{j=1}^{5} B_j(\xi^{(n)})^{j+1} + O((\xi^{(n)})^7), \tag{4.12}$$

where

$$\begin{aligned}
B_1 =& -A_2,\ B_2 = 2(A_2^2 - A_3), B_3 = -4A_2^3 + 4A_2A_3 + 3A_3A_2 - 3A_4,\\
B_4 =& 6A_2A_4 + 6A_3^2 + 4A_4A_2 - 8A_2^2A_3 - 6A_2A_3A_2 - 6A_3A_2^2 + 8A_2^4 - 4A_5,\\
B_5 =& 8A_2A_5 + 9A_3A_4 + 8A_4A_3 + 5A_5A_2 - 12A_2^2A_4 - 12A_2A_3^2 - 8A_2A_4A_2 - 12A_3A_2A_3\\
&- 9A_3^2A_2 - 8A_4A_2^2 + 16A_2^3A_3 + 12A_2^2A_3A_2 + 12A_2A_3A_2^2 + 12A_3A_2^3 - 16A_5^5 - 5A_6.
\end{aligned}$$

Let $Y^{(n)} = \tilde{\xi}^{(n)} + X^*$ and employing (4.12) in the first step of (4.7), one gets

$$\begin{aligned}
\tilde{\xi}^{(n)} =& \frac{1}{3}\xi^{(n)} + \frac{2}{3}A_2(\xi^{(n)})^2 - \frac{4}{3}(A_2^2 - A_3)(\xi^{(n)})^3 - \frac{2}{3}(4A_2A_3 + 3A_3A_2 - 3A_4\\
&- 4A_2^3)(\xi^{(n)})^4 + O((\xi^{(n)})^5).
\end{aligned} \tag{4.13}$$

Further, utilizing the above Eq. (4.13) in the expansion of $F'(Y^{(n)})$ at X^*, we have

$$\begin{aligned}
F'(Y^{(n)}) =& F'(X^*)\left[I + \frac{2}{3}A_2\xi^{(n)} + \frac{1}{3}(A_3 + 4A_2^2)(\xi^{(n)})^2 + \frac{4}{27}(18A_2A_3 + 9A_3A_2\right.\\
&\left.-18A_2^3 + A_4)(\xi^{(n)})^3 + O((\xi^{(n)})^4)\right].
\end{aligned} \tag{4.14}$$

Pre-multiplying the Eq. (4.14) by (4.11), one gets

$$\begin{aligned}
[F'(X^{(n)})]^{-1}F'(Y^{(n)}) =& I - \frac{4}{3}A_2\xi^{(n)} + \frac{4}{3}(-2A_3 + 3A_2^2)(\xi^{(n)})^2 - \frac{8}{27}\left(36A_2^3 + 13A_4\right.\\
&\left.-27A_2A_3 - 18A_3A_2\right)(\xi^{(n)})^3 + O((\xi^{(n)})^4).
\end{aligned} \tag{4.15}$$

From Eq. (4.15), $\mathcal{H}(X^{(n)})$ can be obtained as

$$\begin{aligned}\mathcal{H}(X^{(n)}) = \frac{4}{3}A_2\xi^{(n)} - \frac{4}{3}(-2A_3 + 4A_2^2)(\xi^{(n)})^2 + \frac{8}{27}(36A_2^3 + 13A_4 - 27A_2A_3 \\ - 18A_3A_2)(\xi^{(n)})^3 + O((\xi^{(n)})^4).\end{aligned} \tag{4.16}$$

Consider the weight function $W(\mathcal{H}(X^{(n)}))$ up to two degree terms only as

$$W(\mathcal{H}(X^{(n)})) = W_0 I + W_1\frac{\mathcal{H}(X^{(n)})}{1!} + W_2\frac{\mathcal{H}(X^{(n)})^2}{2!}.$$

On substituting Eq. (4.16) in the above weight function, one gets

$$\begin{aligned}W(\mathcal{H}(X^{(n)})) = W_0 + \frac{4}{3}W_1A_2\xi^{(n)} + \left(\frac{4}{3}W_1(-3A_2^2 + 2A_3) + \frac{8}{9}W_2A_2^2\right)(\xi^{(n)})^2 \\ + \left(\frac{8}{27}W_1(36A_2^3 + 13A_4 - 27A_2A_3 - 8A_3A_2) + \frac{16}{9}W_2(6A_2^3 \right. \\ \left. - 2A_2A_3 - 2A_3A_2))\right)(\xi^{(n)})^3 + O((\xi^{(n)})^4).\end{aligned} \tag{4.17}$$

We now use Eq. (4.17) to expand the second sub-step in (4.7), which leads us to

$$\begin{aligned}\xi^{(n+1)} = (I - \sigma W_0)(\xi^{(n)}) - \left(\frac{4}{3\delta}\sigma W_1A_2 - \sigma W_0A_2 + \frac{2}{3}\eta W_0A_2\right)(\xi^{(n)})^2 \\ - \left(\frac{8}{9}\sigma W_2A_2^2 - 4\sigma W_1A_2^2 + \frac{8}{3}\sigma W_1A_3 - \frac{4}{3}\sigma W_1A_2^2 + 2\sigma W_0(A_2^2 - A_3)\right. \\ \left. + \frac{2\eta}{3\delta}\left(\frac{4}{3}W_1A_2^2 - W_0A_2^2\right) - \frac{\eta W_0}{\delta}(2A_2^2 - \frac{4}{3}A_3)\right)(\xi^{(n)})^3 + O((\xi^{(n)})^4).\end{aligned} \tag{4.18}$$

Moreover, one can achieve fourth-order convergence by adopting the values

$$W_0 = \frac{1}{\sigma},\ W_1 = \frac{3\sigma\delta - 2\eta}{4\sigma^2\delta},\ W_2 = \frac{9\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{4\sigma^3\delta^2}.$$

On substituting, the values mentioned above in (4.18), the error equation becomes

$$\begin{aligned}\xi^{(n+1)} = \left(\frac{(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 - 8\lambda^3)}{27\sigma^3\delta^3}A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2)\right. \\ \left. + \frac{A_4}{9}\right)(\xi^{(n)})^4 + O((\xi^{(n)})^5).\end{aligned} \tag{4.19}$$

Therefore, the proposed scheme (4.7) attains at least fourth-order convergence, under suitable smoothness assumptions. □

4.2.2 Sixth-order scheme

Here, we propose a three-step iterative scheme for solving vectorial problems consisting of the first two steps of the proposed family (4.7), given by

$$\begin{aligned} Z^{(n)} &= \Phi_4^{(4)}(X^{(n)}), \\ X^{(n+1)} &= \Phi_5^{(6)}(X^{(n)}, Y^{(n)}, Z^{(n)}) \\ &= Z^{(n)} - \Big(p_1 I + v_1(X^{(n)})(p_2 I + p_3 v_1(X^{(n)}))\Big)\,[F'(X^{(n)})]^{-1}F(Z^{(n)}), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{4.20}$$

where $v_1(X^{(n)}) = [F'(X^{(n)})]^{-1}F'(Y^{(n)})$. Here, $p_1,\ p_2$, and p_3 are constants that need to be determined.

Now, we analyze the convergence of the proposed scheme (4.20) in the following Theorem (4.2.2), requiring an additional function evaluation $F(Z^{(n)})$ at the third step.

Theorem 4.2.2. *If the conditions of Theorem 4.2.1 hold, then, for $\eta \in \mathbb{R}$ and $\sigma,\ \delta \in \mathbb{R}\backslash\{0\}$, the resulting error equation for* (4.20) *yields at least sixth-order convergence for the sequence $\{X^{(n)}\}_{n=0}^{\infty}$*

$$\begin{aligned} \xi^{(n+1)} = &-\frac{1}{9}\left(9A_3 + 2(-27+8p_3)A_2^2\right)\left(\frac{1}{27\sigma^3\delta^3}(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 - 8\lambda^3)A_2^3\right. \\ &\left. + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\right)(\xi^{(n)})^6 + O((\xi^{(n)})^7), \end{aligned}$$

provided $p_1 = \frac{5}{2} + p_3,\ p_2 = -\frac{1}{2}(3+4p_3)$, and $p_3 \in \mathbb{R}$.

Proof. Suppose $Z^{(n)} = s^{(n)} + X^*$, we can use the Eq. (4.19) established earlier for scheme (4.7) to obtain the following expression

$$\begin{aligned} s^{(n)} = &\left(\frac{(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 - 8\lambda^3)}{27\sigma^3\delta^3}A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\right)(\xi^{(n)})^4 \\ &+ O((\xi^{(n)})^5). \end{aligned} \tag{4.21}$$

Now, expand $F(Z^{(n)})$ about X^*, one acquires

$$F(Z^{(n)}) = F'(X^*)[s^{(n)} + A_2(s^{(n)})^2 + A_3(s^{(n)})^3 + O(s^{(n)})^4]. \tag{4.22}$$

Pre-multiplying Eq. (4.22) by Eq. (4.11) and simplifying, we get

$$[F'(X^{(n)})]^{-1}F(Z^{(n)}) = s^{(n)} - 2A_2(\xi^{(n)})s^{(n)} - (3A_3 - 4A_2^2)(\xi^{(n)})^2 s^{(n)} + (6A_2A_3 + 6A_3A_2 - 4A_4 - 8A_2^3)(\xi^{(n)})^3 s^{(n)} + A_2(s^{(n)})^2 + O(\xi^{(n)})^7. \tag{4.23}$$

By employing Eq. (4.15) in the following expression, one can obtain

$$p_1 I + v_1(X^{(n)})(p_2 I + p_3\ v_1(X^{(n)})) = D_0\ I + D_1\xi^{(n)} + D_2(\xi^{(n)})^2 + D_3(\xi^{(n)})^3 + O((\xi^{(n)})^4), \tag{4.24}$$

where

$$D_0 = p_1 + p_2 + p_3,\ D_1 = -\frac{4}{3}A_2\left(p_2 + 2p_3\right), D_2 = \frac{4}{9}\Big((9p_2 + 22p_3)\,A_2^2 - 6(p_2 + 2p_3)A_3\Big),$$
$$D_3 = \frac{8}{27}\Big[3(9p_2 + 22p_3)A_2A_3 + 6(3p_2 + 8p_3)A_3A_2 - 36(p_2 + 3p_3)A_2^3 - 13(p_2 + 2p_3)A_4\Big].$$

Substituting Eqs. (4.23)–(4.24) in the last sub-step of scheme (4.20), the error equation becomes

$$\begin{aligned}\xi^{(n+1)} =& (1 - p_1 - p_2 - p_3)s^{(n)} + \frac{2}{3}\bigg(3(p_1 + p_2 + p_3) + 2(p_2 + 2p_3)\bigg)A_2\xi^{(n)}s^{(n)} \\ &- \frac{1}{9}\bigg(2\Big(18(p_1 + p_2 + p_3) + 30(p_2 + 2p_3) + 8p_3\Big)A_2^2 - 3\Big(9(p_1 + p_2 + p_3) \\ &+ 8(p_2 + 2p_3)\Big)A_3\bigg)(\xi^{(n)})^2 s^{(n)} + O((\xi^{(n)})^7).\end{aligned} \tag{4.25}$$

Now, our objective is to get appropriate values of p_1, p_2, and p_3 in scheme (4.20) that maximize its convergence order. Therefore, if we put $p_1 = \frac{5}{2} + p_3,\ p_2 = -\frac{1}{2}(3 + 4p_3)$, we obtain our final error equation as

$$\begin{aligned}\xi^{(n+1)} =& -\frac{1}{9}\left(9A_3 + 2(-27 + 8p_3)A_2^2\right)\bigg(\frac{1}{27\sigma^3\delta^3}(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 \\ &- 8\lambda^3)A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\bigg)(\xi^{(n)})^6 + O((\xi^{(n)})^7),\end{aligned} \tag{4.26}$$

where $\eta \in \mathbb{R}$ and $\sigma,\ \delta \in \mathbb{R}\backslash\{0\}$, and p_3 is considered as a free parameter. Hence, it reveals that by employing just two functional evaluations, two Jacobians, and one inverse operator, the presented family (4.20) gives at least sixth-order convergence. □

We have conducted a local convergence analysis for the proposed schemes (4.7) and (4.20) on

the basis of Taylor series expansion to achieve at least fourth and sixth-order convergence, which in turn limited the functionality due to the presence of higher-order derivatives, particularly in schemes that only contain first-order derivatives. These results demonstrate that the sequence $\{X^{(n)}\}$ converges to the solution X^*, provided the initial guess $X^{(0)}$ is sufficiently near the root X^*. However, the question arises: how accurate should the initial guess $X^{(0)}$ be? To estimate the radius of the convergence region associated with the method, these results provide no information. Thus, we use only first-order derivative of F to increase the applicability of the suggested techniques. Furthermore, we employ Lipschitz parameters in place of Taylor expansion. In this manner, we may demonstrate the local convergence of these techniques without the need for higher-order derivatives.

4.2.3 Local convergence of the sixth-order scheme

Let us first consider that F be a nonlinear mapping on the subset $\mathcal{D} = \mathbb{S}$ of Banach space $\mathcal{B}_1$ to $\mathcal{B}_2$, then the proposed sixth-order scheme takes the form

$$\begin{aligned} y_n &= x_n - \frac{2}{3}v_0(x_n), \\ z_n &= x_n - \left(\sigma I + \frac{\eta}{2\delta}\mathcal{H}(x_n)\right) W(\mathcal{H}(x_n))v_0(x_n), \\ x_{n+1} &= z_n - (p_1 + v_1(x_n)(p_2 + p_3 v_1(x_n)))\,[F'(x_n)]^{-1}F(z_n), \quad n \in \hat{\mathbb{N}}_0, \end{aligned} \tag{4.27}$$

where $v_0(x_n) = [F'(x_n)]^{-1}F(x_n)$ and $v_1(x_n) = [F'(x_n)]^{-1}F'(y_n)$.

Using a few scalar functions and parameters, the local convergence of scheme (4.27) in the Banach space is initially shown to derive the error estimates for the required solution X^*.

Consider the parameters $\mathcal{L}_0 > 0$, $\mathcal{L} > 0$, $M > 0$ and define some scalar functions $\hat{h}_1$, $\hat{h}_2$, $\hat{h}_3$, g_2, u_1 and g_3 on the interval $[0, \frac{1}{L_0})$ by

$$\hat{h}_1(s) = \frac{\mathcal{L}s + \frac{2M}{3}}{2(1 - \mathcal{L}_0 s)},$$

$$\hat{h}_2(s) = \frac{\mathcal{L}s}{2\,(1 - \mathcal{L}_0 s)} + \frac{3M\mathcal{L}_0(1 + \hat{h}_1 s)}{4\,(1 - \mathcal{L}_0 s)^2}s + \beta_0 \frac{M\mathcal{L}_0^2(1 + \hat{h}_1 s)^2}{(1 - \mathcal{L}_0 s)^3}s^2 + \beta_1 \frac{M\mathcal{L}_0^3(1 + \hat{h}_1\,(s))^3}{(1 - \mathcal{L}_0 s)^4}s^3,$$

$$g_2(s) = \hat{h}_2(s) - 1,$$

$$u_1(s) = \left(1 + p_1 + \frac{p_2 M}{(1 - \mathcal{L}_0 s)} + \frac{p_3 M^2}{(1 - \mathcal{L}_0 s)^2}\right)\frac{M}{(1 - \mathcal{L}_0 s)},$$

$$\hat{h}_3(s) = u_1(s)\hat{h}_2(s),$$

$$g_3(s) = \hat{h}_3(s) - 1,$$

and some parameters r_1 and r_A defined by

$$r_1 = \frac{2\left(1 - \frac{M}{3}\right)}{\mathcal{L} + 2\mathcal{L}_0},\ r_A = \frac{2}{\mathcal{L} + 2\mathcal{L}_0},$$

and also, we have $\hat{h}_1(r_1) = 1$,

$$0 < r_1 < r_A, \tag{4.28}$$

and

$$0 \leq \hat{h}_1(s) < 1,$$

for each $s \in [0, r_1)$, where $\beta_0 = \left|\frac{18\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{8\sigma^2\delta^2}\right|$, and $\beta_1 = \frac{\eta}{2\delta}\left|\frac{9\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{4\sigma^2\delta^2}\right|$.

From the definitions of the above defined functions $\hat{h}_1, \hat{h}_2, \hat{h}_3, g_2, g_3$ and u_1, we get $g_i(0) = -1 < 0$ and $g_i(s) \to +\infty$ as we move 's' towards $\frac{1^-}{\mathcal{L}_0}$, which follows the condition of intermediate value theorem and hence, the equation $g_i(s) = 0$ have roots in the interval $\left(0, \frac{1}{\mathcal{L}_0}\right)$. Now, we define the minimal positive root r as

$$r = \min\{r_i\},\ i = 1, 2, 3. \tag{4.29}$$

Then, we arrive at

$$0 < r < r_A < \frac{1}{\mathcal{L}_0}, \tag{4.30}$$

$$0 \leq \hat{h}_i(s) < 1, \tag{4.31}$$

and

$$0 \leq u_1(s), \tag{4.32}$$

for each $s \in [0, \bar{r})$.

Let $B(a, \rho)$ and $\overline{B}(a, \rho)$ represent, respectively, the open and closed balls centered at a with radius ρ. The following theorem details the local convergence analysis for the iterative scheme (4.27).

Theorem 4.2.3. *Consider a nonlinear function $F : \mathcal{D} \subseteq \mathcal{B}_1 \to \mathcal{B}_2$ to be Fréchet differentiable and suppose that there exists $X^* \in \mathcal{D}$ and $\mathcal{L}_0 > 0$, such that $F(X^*) = 0,\ F'(X^*)^{-1} \in L(\mathcal{B}_2, \mathcal{B}_1)$, for each*

$x \in \mathcal{D}$,

$$\left\|F'(X^*)^{-1}\left(F'(x)-F'(X^*)\right)\right\| \leq \mathcal{L}_0 \left\|x-X^*\right\|. \tag{4.33}$$

Also, assume that there exists $\mathcal{L}>0$ *and* $M \in [1,3)$ *such that for every* $x, y \in \mathcal{D}_0 := \mathcal{D} \cap B\left(X^*, \frac{1}{\mathcal{L}_0}\right)$, *then*

$$\left\|F'(X^*)^{-1}\left(F'(x)-F'(y)\right)\right\| \leq \mathcal{L}\left\|x-y\right\|, \tag{4.34}$$

$$\left\|F'(X^*)^{-1}F'(x)\right\| \leq M, \tag{4.35}$$

and $\overline{B}(X^*, r) \in \mathcal{D}$, *where* r *in* (4.29) *defines the radius of convergence. Then, the sequence* $\{x_n\}_{n=0}^{\infty}$ *generated from the proposed scheme* (4.27) *is well defined and remain in the open ball* $B(X^*, r)$ *for an initial estimate* $x_0 \in B(X^*, r) - \{X^*\}$ *and converges to* X^* *and the following conditions hold on error estimates*

$$||y_n - X^*|| \leq \hat{h}_1\left(||x_n - X^*||\right)||x_n - X^*|| \leq ||x_n - X^*|| < r, \tag{4.36}$$

$$||z_n - X^*|| \leq \hat{h}_2\left(||x_n - X^*||\right)||x_n - X^*|| \leq ||x_n - X^*|| < r, \tag{4.37}$$

and

$$||x_{n+1} - X^*|| \leq \hat{h}_3\left(||x_n - X^*||\right)||x_n - X^*|| \leq ||x_n - X^*||, \tag{4.38}$$

where the '$\hat{h}_i$*' functions are defined earlier. For* $T_u \in \left[r, \frac{2}{\mathcal{L}_0}\right)$, *the limit* X^* *constitutes the unique zero of* $F(x)=0$ *within uniqueness domain* $\mathcal{D}_1 = \mathcal{D} \cap \overline{B}(X^*, T_u)$.

Proof. For any $x \in B(X^*, r)$, we will first demonstrate that $F'(X^*)^{-1} \in L(\mathcal{B}_2, \mathcal{B}_1)$.

Employing (4.30) and (4.33), we have

$$\left\|F'(X^*)^{-1}\left(F'(x)-F'(X^*)\right)\right\| \leq \mathcal{L}\left\|x-X^*\right\| < \mathcal{L}r < 1. \tag{4.39}$$

Using Banach's invertibility lemma and the inequality (4.39) imply that $F(x)^{-1} \in L(\mathcal{B}_2, \mathcal{B}_1)$, and therefore

$$\left\|F'(x)^{-1}F'(X^*)\right\| \leq \frac{1}{1-\mathcal{L}_0\left\|x-X^*\right\|}. \tag{4.40}$$

As of fact the initial estimate $x_0 \in B(X^*, r)$, therefore the estimate (4.40) for x_0, it takes the form

$$\left\|F'(x_0)^{-1}F'(X^*)\right\| \leq \frac{1}{1-\mathcal{L}_0\left\|x_0-X^*\right\|}. \tag{4.41}$$

For $n = 0$, the scheme (4.27) yields well-defined y_0, z_0, and x_1 iterates. Consequently, we obtain

$$F(x_0) = F(x_0) - F(X^*) = \int_0^1 F'\left(X^* + \theta(x_0 - X^*)\right)(x_0 - X^*)d\theta. \tag{4.42}$$

It is important to note that $X^*+\theta(x_0-X^*) \in B(X^*, r)$ as $||X^* + \theta(x_0 - X^*) - X^*|| = \theta\,||x_0 - X^*|| < r$. From the Eqs. (4.35) and (4.42), we obtain

$$\begin{aligned} ||F'(X^*)^{-1}F(x_0)|| &= \left|\left|\int_0^1 F'(X^*)^{-1}F'(X^* + \theta(x_0 - X^*))(x_0 - X^*)d\theta\right|\right| \\ &\leq M||x_0 - X^*||. \end{aligned} \tag{4.43}$$

By using the first substep of the method (4.27), (4.39), (4.41), and (4.43) for $n = 0$, we get

$$\begin{aligned} ||y_0 - X^*|| &\leq \left|\left|F'(x_0)^{-1}F'(X^*)\right|\right| \left|\left|F'(X^*)^{-1}\left(F'(X^* + \theta(x_0 - X^*)) - F'(x_0)\right)(x_0 - X^*)d\theta\right|\right| \\ &\quad + \frac{1}{3}\left|\left|F'(x_0)^{-1}F'(X^*)\right|\right| \left|\left|F'(X^*)^{-1}F(x_0)\right|\right| \\ &\leq \frac{\mathcal{L}\,||x_0 - X^*||^2}{2\left(1 - \mathcal{L}_0\,||x_0 - X^*||\right)} + \frac{M\,||x_0 - X^*||}{3\left(1 - \mathcal{L}_0\,||x_0 - X^*||\right)} \\ &= \hat{h}_1\left(||x_0 - X^*||\right)||x_0 - X^*|| \leq ||x_0 - X^*|| < r, \end{aligned}$$

which shows that (4.36) is true for $n = 0$ and $y_0 \in B(X^*, r)$.

For simplicity, let us consider $\alpha_0 = F'(x_0)^{-1}F'(y_0)$, and let $C_w = \left(\sigma I + \frac{\eta}{2\delta}\mathcal{H}(X^{(n)})\right)W(\mathcal{H}(X^{(n)}))$. Then, we obtain

$$\begin{aligned} C_w = I + \frac{3}{4}(I - \alpha_0) + \left(\frac{18\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{8\sigma^2\delta^2}\right)(I - \alpha_0)^2 \\ + \frac{\eta}{2\delta}\left(\frac{9\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{4\sigma^2\delta^2}\right)(I - \alpha_0)^3. \end{aligned} \tag{4.44}$$

Proceeding to the second sub-step of algorithm (4.27) for the initial index for $n = 0$ then allows us to state that

$$\begin{aligned} z_0 - X^* = (x_0 - X^*) - F'(x_0)^{-1}F(x_0) + \Big(\frac{3}{4}\Big(F'(x_0)^{-1}\left((F'(x_0) - F'(X^*)) + \right. \\ \left.\times(F'(X^*) - F'(y_0))\right)\Big) + \left(\frac{18\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{8\sigma^2\delta^2}\right)\Big(F'(x_0)^{-1}\left((F'(x_0)\right. \\ \left.-F'(X^*)) + (F'(X^*) - F'(y_0))\right)\Big)^2 + \frac{\eta}{2\delta}\left(\frac{9\sigma^2\delta^2 - 3\sigma\delta\eta + 2\eta^2}{4\sigma^2\delta^2}\right) \\ \times\Big(F'(x_0)^{-1}\left((F'(x_0) - F'(X^*)) + (F'(X^*) - F'(y_0))\right)\Big)^3\Big) \end{aligned} \tag{4.45}$$

$$\times \left(F'(x_0)^{-1}F'(X^*)\right)\left(F'(X^*)^{-1}F(x_0)\right),$$

and applying norms on both sides, one gets

$$\begin{aligned}
||z_0 - X^*|| \leq{}& \left|\left|x_0 - X^* + F'(x_0)^{-1}F(x_0)\right|\right| + \left(\frac{3}{4}\left|\left|F'(x_0)^{-1}F'(X^*)\right|\right|^2\right. \\
&\times \left(\left|\left|F'(X^*)^{-1}(F'(x_0) - F'(X^*))\right|\right| + \left|\left|F'(X^*)^{-1}(F'(X^*) - F'(y_0))\right|\right|\right) \\
&+ \beta_0\left|\left|F'(x_0)^{-1}F'(X^*)\right|\right|^3\left(\left|\left|F'(X^*)^{-1}(F'(x_0) - F'(X^*))\right|\right|\right. \\
&\left.+\left|\left|F'(X^*)^{-1}(F'(X^*) - F'(y_0))\right|\right|\right)^2 + \beta_1\left|\left|F'(x_0)^{-1}F'(X^*)\right|\right|^4 \\
&\times \left(\left|\left|F'(X^*)^{-1}(F'(x_0) - F'(X^*))\right|\right| + \left|\left|F'(X^*)^{-1}(F'(X^*)\right|\right|\right. \\
&\left.-F'(y_0))||^3\right)\left|\left|F'(X^*)^{-1}F(x_0)\right|\right| \\
\leq{}& \frac{\mathcal{L}\,||x_0 - X^*||^2}{2\left(1 - \mathcal{L}_0\,||x_0 - X^*||\right)} + \frac{3M\mathcal{L}_0(1 + \hat{h}_1\left(||x_0 - X^*||\right))}{4\left(1 - \mathcal{L}_0\,||x_0 - X^*||\right)^2}\,||x_0 - X^*||^2 \\
&+ \beta_0\frac{M\mathcal{L}_0^2(1 + \hat{h}_1\left(||x_0 - X^*||\right))^2}{\left(1 - \mathcal{L}_0\,||x_0 - X^*||\right)^3}\,||x_0 - X^*||^3 + \beta_1\frac{M\mathcal{L}_0^3(1 + \hat{h}_1\left(||x_0 - X^*||\right))^3}{\left(1 - \mathcal{L}_0\,||x_0 - X^*||\right)^4}\,||x_0 - X^*||^4 \\
={}& \hat{h}_2\left(||x_0 - X^*||\right)||x_0 - X^*|| \leq ||x_0 - X^*|| < r,
\end{aligned} \tag{4.46}$$

which implies (4.37) holds for $n = 0$ and $z_0 \in B(X^*, r)$.

Then, from the third step of the proposed scheme (4.27) for $n = 0$, we have

$$\begin{aligned}
x_1 - X^* = (z_0 - X^*) &- \left(p_1 I + p_2\left(F'(x_0)^{-1}F'(X^*)\right)\left(F'(X^*)^{-1}F'(y_0)\right)\right. \\
&+p_3\left(F'(x_0)^{-1}F'(X^*)\right)\left(F'(X^*)^{-1}F'(y_0)\right)\left(F'(x_0)^{-1}F'(X^*)\right) \\
&\left.\times\left(F'(X^*)^{-1}F'(y_0)\right)\right)\left(F'(x_0)^{-1}F'(X^*)\right)\left(F'(X^*)^{-1}F(z_0)\right).
\end{aligned} \tag{4.47}$$

Now taking norms on both sides of (4.47) and using (4.35), (4.39), (4.41), (4.43), and (4.46), in turn we obtain that

$$\begin{aligned}
||x_1 - X^*|| \leq{}& ||z_0 - X^*|| + \left(p_1 + p_2||F'(x_0)^{-1}F'(X^*)||||F'(X^*)^{-1}F'(y_0)||\right. \\
&\left.+p_3||F'(x_0)^{-1}F'(X^*)||^2||F'(X^*)^{-1}F'(y_0)||^2\right)||F'(x_0)^{-1}F'(X^*)|| \\
&\times ||F'(X^*)^{-1}F(z_0)|| \\
\leq{}& \left(1 + p_1 + \frac{p_2 M}{(1 - \mathcal{L}_0||x_0 - X^*||)} + \frac{p_3 M^2}{(1 - \mathcal{L}_0||x_0 - X^*||)^2}\right)
\end{aligned}$$

$$
\begin{aligned}
&\times \frac{M}{(1-\mathcal{L}_0||x_0-X^*||)}||z_0-X^*|| \\
&\leq u_1(||x_0-X^*||)\ \hat{h}_2(||x_0-X^*||)||x_0-X^*|| \\
&\leq \hat{h}_3(||x_0-X^*||)||x_0-X^*|| \\
&\leq ||x_0-X^*|| < r.
\end{aligned}
$$

For the next step, we replace x_1, z_0, y_0, x_0 by x_{n+1}, z_n, y_n, x_n, and employing the mathematical induction, we arrive at the estimates (4.36)–(4.38). Henceforth,

$$||x_{n+1}-X^*|| \leq \mathcal{C}\,||x_n-X^*||\,,$$

where $\mathcal{C} = \hat{h}_3(||x_n-X^*||) \in [0,1)$. So, $\lim_{n\to\infty} x_{n+1} = X^*$ and $x_{n+1} \in B(X^*,r)$. Finally, we show the uniqueness part, for that let us consider $\mathcal{Q} = \int_0^1 F(X^*+\theta(\mathcal{Y}^*-X^*))d\theta$, where $F(\mathcal{Y}^*)=0$ and $\mathcal{Y}^* \in \mathcal{D}_1$. Then, using

$$\Big|\Big|F'(X^*)^{-1}(\mathcal{Q}-F'(X^*))\Big|\Big| \leq \frac{\mathcal{L}_0}{2}\,||X^*-\mathcal{Y}^*|| < \frac{\mathcal{L}_0 T_u}{2} < 1, \tag{4.48}$$

which implies $\mathcal{Q} \in L(\mathcal{B}_2,\mathcal{B}_1)$. Hence, $\mathcal{Y}^* = X^*$. □

4.3 Generalized scheme

The generalized $(q+1)$-step scheme is proposed by taking three-step scheme (4.20) as a predictor and has the following structure

$$
\begin{aligned}
w_0^{(n)} &= X^{(n)} - \frac{2}{3}[F'(X^{(n)})]^{-1}F(X^{(n)}), \\
w_1^{(n)} &= X^{(n)} - \Big(\sigma I + \frac{\eta}{2\delta}\mathcal{H}(X^{(n)})\Big)\,W(\mathcal{H}(X^{(n)}))[F'(X^{(n)})]^{-1}F(X^{(n)}), \\
w_2^{(n)} &= w_1^{(n)} - Q_n\ [F'(X^{(n)})]^{-1}F(w_1^{(n)}), \\
w_3^{(n)} &= w_2^{(n)} - Q_n\ [F'(X^{(n)})]^{-1}F(w_2^{(n)}), \\
w_4^{(n)} &= w_3^{(n)} - Q_n\ [F'(X^{(n)})]^{-1}F(w_3^{(n)}), \\
&\vdots \qquad \vdots \qquad\qquad\qquad \vdots \\
w_{q-1}^{(n)} &= w_{q-2}^{(n)} - Q_n\ [F'(X^{(n)})]^{-1}F(w_{q-2}^{(n)}), \\
w_q^{(n)} &= w_{q-1}^{(n)} - Q_n\ [F'(X^{(n)})]^{-1}F(w_{q-1}^{(n)}), \quad n \in \hat{\mathbb{N}}_0,
\end{aligned}
\tag{4.49}
$$

where $q \in \mathbb{N}\backslash\{1\}$, $w_0^{(n)} = Y^{(n)}$, $w_1^{(n)} = Z^{(n)}$, $w_q^{(n)} = X^{(n+1)}$, $v_1(X^{(n)}) = [F'(X^{(n)})]^{-1}F'(Y^{(n)})$, $\mathcal{H}(X^{(n)}) = I - v_1(X^{(n)})$, $Q_n = \Big(p_1 I + v_1(X^{(n)})(p_2 I + p_3 v_1(X^{(n)}))\Big)$, $p_1 = \frac{5}{2} + p_3$, and $p_2 = -\frac{1}{2}(3+4p_3)$.

We establish the following theorem to examine the convergence order:

Theorem 4.3.1. *If the conditions of Theorem 4.2.2 hold, then, for $\eta \in \mathbb{R}$ and $\sigma,\ \delta \in \mathbb{R}\backslash\{0\}$, the resulting error equation for (4.49) yields at least $2q+2$, $q \in \mathbb{N}\backslash\{1\}$ convergence order for the sequence $\{X^{(n)}\}_{n=0}^{\infty}$ is*

$$\begin{aligned}\xi^{(n+1)} = &\Big(-\frac{1}{9}(9A_3 + 2(-27+8p_3)A_2^2)\Big)^{q-1}\left(\frac{1}{27\sigma^3\delta^3}(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 - 8\lambda^3)A_2^3\right.\\ &\left.+\frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\right)(\xi^{(n)})^{2q+2} + O((\xi^{(n)})^{2q+3}),\end{aligned}$$

where $\xi^{(n)} = X^{(n)} - X^*$, and $\xi_{w_{q-1}}^{(n)} = w_{q-1}^{(n)} - X^*$.

Proof. Let $\xi^{(n)} = X^{(n)} - X^*$, and $\xi_{w_{q-1}}^{(n)} = w_{q-1}^{(n)} - X^*$. By using Lemma (1.2.1), expanding $F(X^{(n)})$ in a neighborhood of X^*, one can obtain

$$\begin{aligned}F(w_{q-1}^{(n)}) = F'(X^*)[&\xi_{w_{q-1}}^{(n)} + A_2\Big(\xi_{w_{q-1}}^{(n)}\Big)^2 + A_3\Big(\xi_{w_{q-1}}^{(n)}\Big)^3 + A_4\Big(\xi_{w_{q-1}}^{(n)}\Big)^4\\ &+ O((\xi_{w_{q-1}}^{(n)})^5)],\end{aligned} \tag{4.50}$$

Post-multiplying (4.11) by (4.50), we get

$$\begin{aligned}[F'(X^{(n)})]^{-1}F(w_{q-1}^{(n)}) = \xi_{w_{q-1}}^{(n)} - \Big(&2A_2\xi^{(n)} - (4A_2^2 - 3A_3)(\xi^{(n)})^2 + (-8A_2^3\\ &+ 6A_2A_3 + 6A_3A_2 - 4A_4)(\xi^{(n)})^3 + O(\xi^{(n)})^4)\xi_{w_{q-1}}^{(n)}\\ &+ A_2(\xi_{w_{q-1}}^{(n)})^2 - 2A_2\xi^{(n)}(\xi_{w_{q-1}}^{(n)})^2 + O((\xi^{(n)}\xi_{w_{q-1}}^{(n)})^2).\end{aligned} \tag{4.51}$$

Using Eq. (4.24), we can write

$$\begin{aligned}Q_n = I + 2A_2\xi^{(n)} + \frac{4}{9}\Bigg(\Big(-\frac{27}{2} + 4p_3\Big)A_2^2 + 9A_3\Bigg)(\xi^{(n)})^2 + \frac{8}{27}\Bigg(3\Bigg(4p_3 - \frac{27}{2}\Bigg)A_2A_3\\ + 6\Big(-\frac{9}{2} + 2p_3\Big)A_3A_2 - 36\Big(-\frac{3}{2} + p_3\Big)A_2^3 + \frac{39}{2}A_4\Bigg)(\xi^{(n)})^3 + O((\xi^{(n)})^4).\end{aligned} \tag{4.52}$$

By employing Eq. (4.51) and Eq. (4.52), one obtains

$$Q_n[F'(X^{(n)})]^{-1}F(w_{q-1}^{(n)}) = \xi_{w_{q-1}}^{(n)} + \Bigg(-\frac{1}{9}\Bigg((-27+8p_3)A_2^2 + 9A_3\Bigg)(\xi^{(n)})^2 + O((\xi^{(n)})^3)\Bigg)\xi_{w_{q-1}}^{(n)} \tag{4.53}$$

$$+ O((\xi^{(n)}_{w_{q-1}})^2).$$

Then, using the Eq. (4.53) in the scheme's (4.49) final step, we have

$$w_q^{(n)} - X^* = \bigg(-\frac{1}{9}\bigg((-27+8p_3)A_2^2 + 9A_3\bigg)(\xi^{(n)})^2 + O((\xi^{(n)})^3)\bigg)\xi^{(n)}_{w_{q-1}} + O((\xi^{(n)}_{w_{q-1}})^2). \tag{4.54}$$

We obtain the relevant error equations for $q = 3, 4$ as

$$\begin{aligned} w_3^{(n)} - X^* &= \bigg(-\frac{1}{9}\bigg((-27+8p_3)A_2^2 + 9A_3\bigg)(\xi^{(n)})^2 + O((\xi^{(n)})^3)\bigg)\xi^{(n)}_{w_2} + O((\xi^{(n)}_{w_2})^2), \\ &= \frac{1}{81}\Big(9A_3 + 2(-27+8p_3)A_2^2)\Big)^2\bigg(\frac{1}{27\sigma^3\delta^3}(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 \\ &\quad - 8\lambda^3)A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\bigg)(\xi^{(n)})^8 + O((\xi^{(n)})^9), \end{aligned} \tag{4.55}$$

and

$$\begin{aligned} w_4^{(n)} - X^* &= \bigg(-\frac{1}{9}\bigg((-27+8p_3)A_2^2 + 9A_3\bigg)(\xi^{(n)})^2 + O((\xi^{(n)})^3)\bigg)\xi^{(n)}_{w_3} + O((\xi^{(n)}_{w_3})^2), \\ &= -\frac{1}{243}\Big(9A_3 + 2(-27+8p_3)A_2^2\Big)^3\bigg(\frac{1}{27\sigma^3\delta^3}(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 \\ &\quad - 8\lambda^3)A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\bigg)(\xi^{(n)})^{10} + O((\xi^{(n)})^{11}). \end{aligned} \tag{4.56}$$

On proceeding through induction, we obtain

$$\begin{aligned} \xi^{(n+1)} &= w_q^{(n)} - X^* \\ &= \bigg(-\frac{1}{9}\bigg((-27+8p_3)A_2^2 + 9A_3\bigg)(\xi^{(n)})^2 + O((\xi^{(n)})^3)\bigg)\xi^{(n)}_{w_{q-1}} + O((\xi^{(n)}_{w_{q-1}})^2), \\ &= \bigg(-\frac{1}{9}(9A_3 + 2(-27+8p_3)A_2^2)\bigg)^{q-1}\bigg(\frac{1}{27\sigma^3\delta^3}(135\sigma^3\delta^3 - 36\sigma^2\delta^2\eta + 12\sigma\delta\eta^2 \\ &\quad - 8\lambda^3)A_2^3 + \frac{1}{2}(A_2A_3 - 3A_3A_2) + \frac{A_4}{9}\bigg)(\xi^{(n)})^{2q+2} + O((\xi^{(n)})^{2q+3}). \end{aligned}$$

Hence, the result follows for $q \geq 2$, where p_3 is free parameter. □

Now, we are going to present some additional auxiliary functions and parameters for analyzing the local convergence of the novel generalized family (4.49) in the Banach space setting. Define

functions $\hat{h}_i$ on the interval $[0, \frac{1}{\mathcal{L}_0}\big), i = 3, \cdots, q+1$ by

$$\hat{h}_i(s) = u_1(s)\hat{h}_{i-1}(s) \tag{4.57}$$

and

$$g_i(s) = \hat{h}_i(s) - 1. \tag{4.58}$$

Then, again we have that $g_i(0) = -1 < 0$ and $g_i(s) \to +\infty$ as $s \to \frac{1^-}{\mathcal{L}_0}$. For each function g_i in $\big(0, \frac{1}{\mathcal{L}_0}\big)$, let r_i represent its smallest zero and define

$$\bar{r} = \min\{r_j\}, \quad j = 1, 2, \cdots, q+1. \tag{4.59}$$

Hence,

$$0 < \bar{r} < r_A < \frac{1}{\mathcal{L}_0}, \tag{4.60}$$

$$0 \leq \hat{h}_j(s) < 1, \tag{4.61}$$

and $0 \leq u_1(s)$, for each $s \in [0, \bar{r})$. Employing the previously mentioned notations and the demonstration of Theorem 4.2.3, we obtain the following result.

Theorem 4.3.2. *Consider a function $F : \mathcal{D} \subseteq \mathcal{B}_1 \to \mathcal{B}_2$ to be Fréchet differentiable and assume that there exists $X^* \in \mathcal{D}$ and $\mathcal{L}_0 > 0$ and $M \in [1, 3)$ satisfying $F(X^*) = 0$, $F'(X^*)^{-1} \in L(\mathcal{B}_2, \mathcal{B}_1)$, and (4.33)–(4.35) and $\bar{B}(X^*, \bar{r}) \subset \mathcal{D}$, where $\bar{r}$ denotes the the radius of convergence given in (4.59). Then, the sequence $\{w_n^j\}$ generated by the scheme (4.49) for each $x_0 \in \mathcal{B}(X^*, \bar{r}) - \{X^*\}$ is well-defined, remains in $\mathcal{B}(X^*, \bar{r})$ for $n = 0, 1, 2, \ldots$, and $j = 1, 2, \ldots, q$, the sequence converges to X^* and satisfies the error bounds:*

$$||w_n^j - X^*|| \leq \hat{h}_{j+1}(||x_n - X^*||)||x_n - X^*|| \leq ||x_n - X^*|| < \bar{r}, \ j = 0, 1, 2, \cdots, q-1, \tag{4.62}$$

and

$$||x_{n+1} - X^*|| = ||w_n^q - X^*|| \leq \hat{h}_{q+1}(||x_n - X^*||)||x_n - X^*|| \leq ||x_n - X^*||, \tag{4.63}$$

where the $\hat{h}$ functions are provided earlier. Furthermore, for $T \in [\bar{r}, \frac{2}{\mathcal{L}_0}\big)$, the limit X^* of the sequence is the unique solution of $F(x) = 0$ in $\mathcal{D}_1$.

Proof. This theorem can be proved by following similar steps as in the proof of Theorem 4.2.3, with appropriate modifications to account for the altered coefficients and corresponding error terms. □

4.4 Computational complexity

This section investigates the computational cost of an iterative solver. Here, we consider different sets of parametric values in our proposed schemes (4.7), and (4.20) as follows:

- $\mathrm{MM}_{4,1}$: $\sigma = \frac{1}{2}, \delta = \frac{5}{7}, \eta = 1$,
- $\mathrm{MM}_{4,2}$: $\sigma = \frac{3}{4}, \delta = \frac{4}{7}, \eta = 1$,
- $\mathrm{MM}_{4,3}$: $\sigma = \frac{1}{4}, \delta = \frac{11}{10}, \eta = \frac{9}{10}$,
- $\mathrm{MM}_{6,1}$: $\sigma = 1, \delta = \frac{1}{2}, \eta = 1$,
- $\mathrm{MM}_{6,2}$: $\sigma = \frac{1}{2}, \delta = \frac{5}{7}, \eta = 1$,
- $\mathrm{MM}_{6,3}$: $\sigma = 1, \delta = \frac{2}{5}, \eta = 1$.

The computational efficiency of the proposed fourth and sixth-order methods is compared with several existing schemes, i.e., the Nedzhibov method [257] ($\mathrm{NM}_{4,1}$), Sharma and Arora method [306] ($\mathrm{SM}_{4,1}$), Cordero et al. method [89] ($\mathrm{CM}_{6,1}$), Narang et al. method [253] ($\mathrm{NM}_{6,1}$), and the Dehghan and Shirilord method [108] ($\mathrm{DM}_{6,1}$). We define the computational cost and efficiency index of each solver as $\mathcal{C}_{\mathrm{M}_{p,k}}$ and $E_{\mathrm{M}_{p,k}}$, respectively. Based on the provided information, we compute the costs and efficiencies displayed in Table 4.1.

In the following subsection, we will discuss the efficiency comparisons of various approaches.

4.4.1 Efficiency comparisons

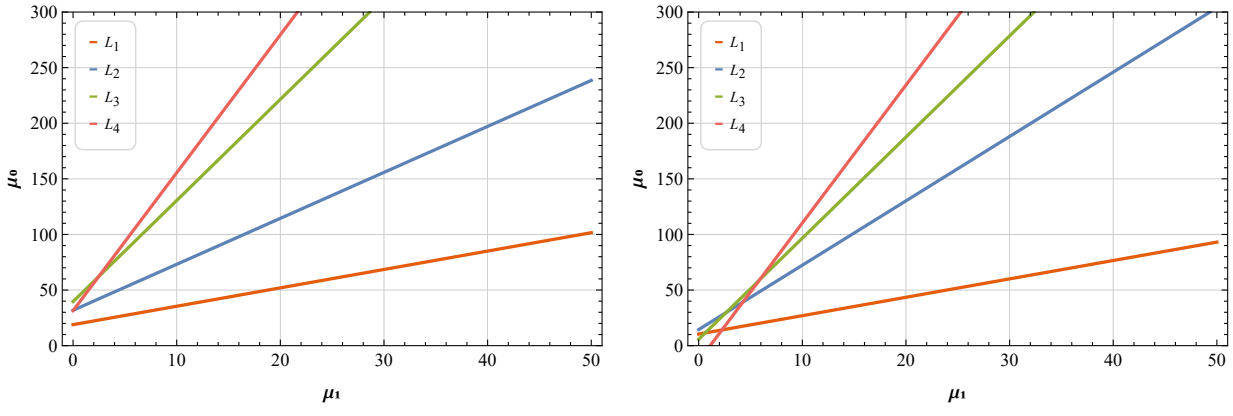
We evaluate the efficiency of two iterative methods, $\hat{M}_{p,i}$ and $\hat{M}_{q,j}$, through their ratio comparisons, which is mathematically defined by

$$R(\hat{M}_{p,i}, \hat{M}_{q,j}) = \frac{\log(E_{\hat{M}_{p,i}})}{\log(E_{\hat{M}_{q,j}})} = \frac{\mathcal{C}_{\hat{M}_{q,j}} \log(p)}{\mathcal{C}_{\hat{M}_{p,i}} \log(q)}. \tag{4.64}$$

It is evident that if $R(\hat{M}_{p,i}, \hat{M}_{q,j}) > 1$, then $\hat{M}_{p,i}$ is more efficient method than $\hat{M}_{q,j}$. The boundary between the two computational efficiencies occurs when $R(\hat{M}_{p,i}, \hat{M}_{q,j}) = 1$. This boundary can be described by an expression μ_0 which is a function of μ_1, k, and l. Here, (μ_1, μ_0) are in the range $(0, \infty) \times (0, \infty)$, where $k \geq 2$ and $l_q \geq 1$. Using the ratio (4.64), we compile our comparison results of the proposed solvers alongside existing ones, considering both the same and different convergence orders in the following theorem:

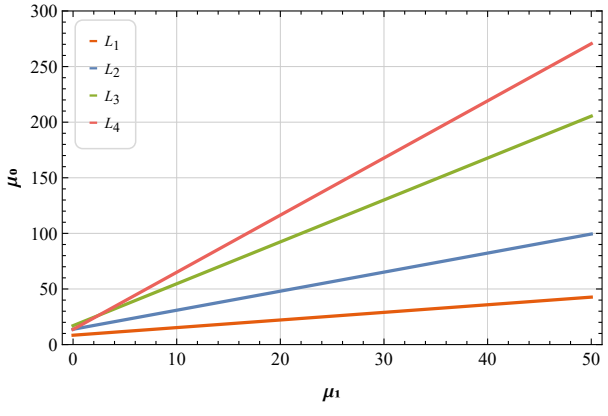
Theorem 4.4.1. *1. Given $l_q \geq 1$, μ_1, and $\mu_0 > 0$:*

(a) $E_{MM_{4,1}} > E_{NM_{4,1}} \iff MM_{4,1} > NM_{4,1}\ \forall\ k \geq 13$.

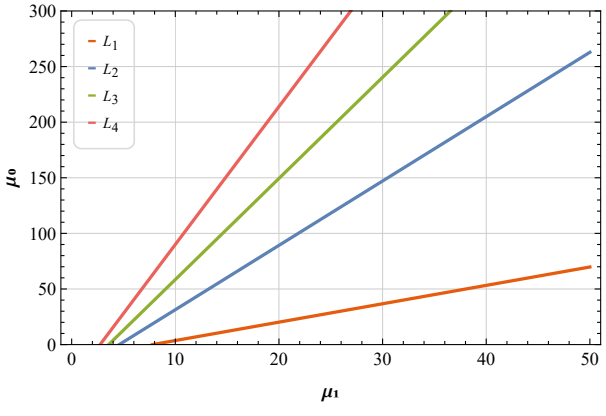


(a) $\mathrm{MM}_{4,1}$ versus $\mathrm{CM}_{6,1}$

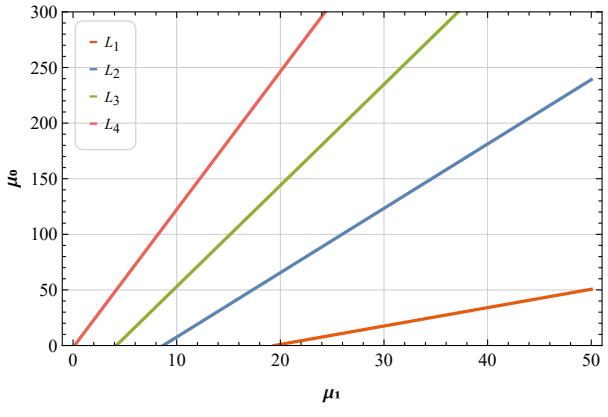
(b) $\mathrm{MM}_{4,1}$ versus $\mathrm{NM}_{6,1}$



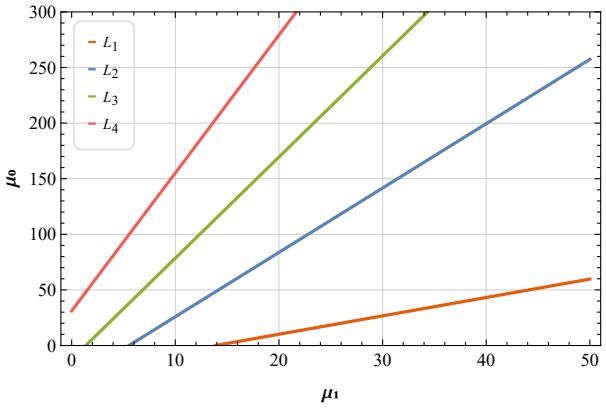
(c) $\mathrm{MM}_{4,1}$ versus $\mathrm{DM}_{6,1}$



(d) $\mathrm{MM}_{4,1}$ versus $\mathrm{MM}_{6,1}$



(e) $\mathrm{MM}_{6,1}$ versus $\mathrm{NM}_{4,1}$



(f) $\mathrm{MM}_{6,1}$ versus $\mathrm{SM}_{4,1}$

Figure 4.1: Boundary lines in the plane-(μ_0, μ_1).

(b) $E_{MM_{4,1}} > E_{SM_{4,1}} \iff MM_{4,1} > SM_{4,1}\ \forall\ k \geq 10.$

(c) $E_{MM_{6,1}} > E_{CM_{6,1}} \iff MM_{6,1} > CM_{6,1}\ \forall\ k \geq 25.$

(d) $E_{MM_{6,1}} > E_{NM_{6,1}} \iff MM_{6,1} > NM_{6,1}\ \forall\ k \geq 19.$

(e) $E_{MM_{6,1}} > E_{DM_{6,1}} \iff MM_{6,1} > DM_{6,1}\ \forall\ k \geq 25.$

2. *For all $k \geq 2$, we have:*
   (a) $E_{MM_{4,1}} > E_{CM_{6,1}} \iff MM_{4,1} > CM_{6,1} \ \forall \ \mu_0 \geq \gamma_2$.
   (b) $E_{MM_{4,1}} > E_{NM_{6,1}} \iff MM_{4,1} > NM_{6,1} \ \forall \ \mu_0 \geq \gamma_3$.
   (c) $E_{MM_{4,1}} > E_{DM_{6,1}} \iff MM_{4,1} > DM_{6,1} \ \forall \ \mu_0 \geq \gamma_4$.
   (d) $E_{MM_{4,1}} > E_{MM_{6,1}} \iff MM_{4,1} > MM_{6,1} \ \forall \ \mu_0 \geq \gamma_5$.
   (e) $E_{MM_{6,1}} > E_{NM_{4,1}} \iff MM_{6,1} > NM_{4,1} \ \forall \ \mu_0 \geq \gamma_6$.
   (f) $E_{MM_{6,1}} > E_{SM_{4,1}} \iff MM_{6,1} > SM_{4,1} \ \forall \ \mu_0 \geq \gamma_7$.

Table 4.1: Computational costs and efficiencies of different methods.

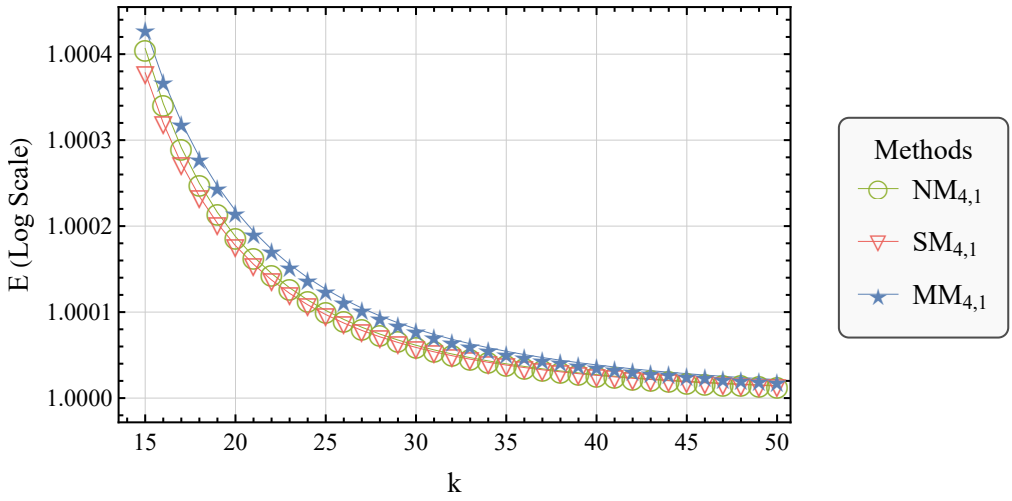
$\hat{M}_{p,i}$	$\mathcal{C}_{\hat{M}_{p,i}}$	$E_{\hat{M}_{p,i}}$
$\mathrm{NM}_{4,1}$	$k\mu_0 + 2k^2\mu_1 + \frac{k}{3}(1 + 2k^2 + 3l_q(1+k) + 6k)$,	$(4)^{1/C_{\mathrm{NM}_{4,1}}}$
$\mathrm{SM}_{4,1}$	$k\mu_0 + 2k^2\mu_1 + \frac{k}{3}(4 + 2k^2 + 3l_q(2+k) + 9k)$,	$(4)^{1/C_{\mathrm{SM}_{4,1}}}$
$\mathrm{MM}_{4,1}$	$k\mu_0 + 2k^2\mu_1 + \frac{k}{6}(1 + 2k^2 + 3l_q(7+k) + 39k)$,	$(4)^{1/C_{\mathrm{MM}_{4,1}}}$
$\mathrm{CM}_{6,1}$	$2k\mu_0 + 2k^2\mu_1 + \frac{k}{3}(1 + 2k^2 + 3l_q(2+k) + 9k)$,	$(6)^{1/C_{\mathrm{CM}_{6,1}}}$
$\mathrm{NM}_{6,1}$	$2k\mu_0 + 2k^2\mu_1 + \frac{k}{3}(1 + 2k^2 + 3l_q(3+k) + 15k)$,	$(6)^{1/C_{\mathrm{NM}_{6,1}}}$
$\mathrm{DM}_{6,1}$	$3k\mu_0 + 2k^2\mu_1 + \frac{k}{3}(-8 + 2k^2 + 3l_q(3+k) + 9k)$,	$(6)^{1/C_{\mathrm{DM}_{6,1}}}$
$\mathrm{MM}_{6,1}$	$2k\mu_0 + 2k^2\mu_1 + \frac{k}{6}(1 + 2k^2 + 3l_q(13+k) + 69k)$,	$(6)^{1/C_{\mathrm{MM}_{6,1}}}$

Where

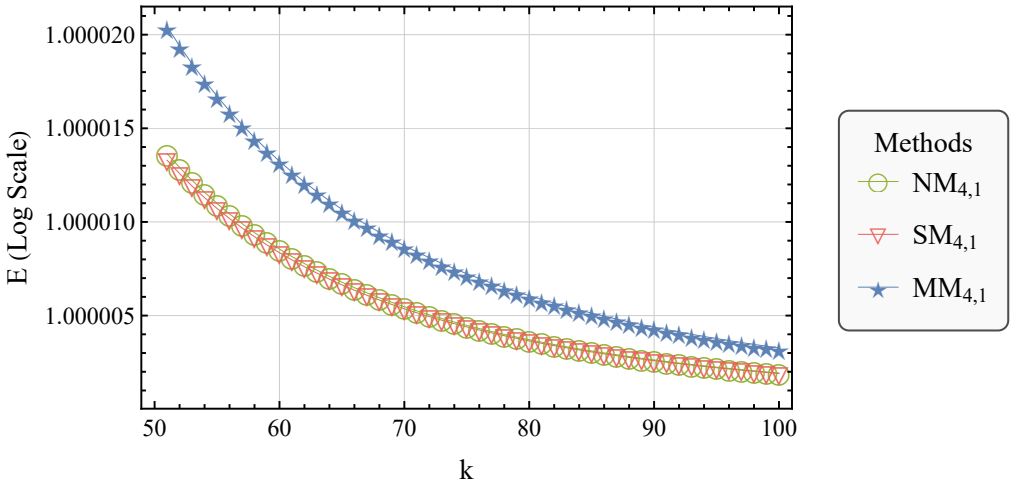
$$
\begin{aligned}
\gamma_2 &= -\frac{1}{6(2\hat{a}-\hat{b})}(\hat{a}(2 + 12l_q + k(18 + 6l_q + 4k + 12\mu_1)) - \delta_1),\\
\gamma_3 &= -\frac{1}{6(2\hat{a}-\hat{b})}(\hat{a}(2 + 18l_q + k(30 + 6l_q + 4k + 12\mu_1)) + \delta_1),\\
\gamma_4 &= \frac{1}{6(3\hat{a}-\hat{b})}(\hat{a}(16 - 18l_q - k(18 + 6l_q + 4k + 12\mu_1)) + \delta_1),\\
\gamma_5 &= -\frac{1}{6(2\hat{a}-\hat{b})}(\hat{a}(1 + 39l + k(69 + 3l_q + 2k + 12\mu_1)) - \delta_1),\\
\gamma_6 &= -\frac{1}{6(\hat{b}-2\hat{a})}(\hat{b}(2 + 6l_q + k(12 + 6l_q + 4k + 12\mu_1)) - \delta_2),\\
\gamma_7 &= -\frac{1}{6(\hat{b}-2\hat{a})}(\hat{b}(8 + 12l_q + k(18 + 6l_q + 4k + 12\mu_1)) - \delta_2).
\end{aligned}
$$

where $\hat{a} = \log(4)$, $\hat{b} = \log(6)$, $\delta_1 = \hat{b}(1 + 21l_q + k(39 + 3l_q + 2k + 12\mu_1))$, *and* $\delta_2 = \hat{a}(1 + 39l_q + k(69 + 3l_q + 2k + 12\mu_1))$.

Additionally, the boundary lines L_1, L_2, L_3, L_4 are delineated for $k = 2, 7, 11, 15$ and $l_q = 2$ in Figures 4.1a–4.1f to evaluate the efficacy of the proposed methods against established solvers. Within the area below the boundary lines, the proposed solvers demonstrate greater efficacy than the comparative methods, as illustrated in Figures 4.1a–4.1f, where the effective region of the proposed solvers expands with increasing values of k. Furthermore, as illustrated in Figures 4.2–4.5, the proposed methodologies exhibit an enhanced computational efficiency index with the increase of k. The theoretical results indicate that, for large-scale systems, the proposed solvers demonstrate superior efficiency compared to the alternative ones, as depicted in Figures 4.2–4.5.



(a) For $k = 15, 16, \cdots, 50$.



(b) For $k = 51, 52, \cdots, 100$.

Figure 4.2: Efficiency plots for fourth-order methods on small k-scale systems.

4.5 Numerical tests

This section considers many real-life applications to demonstrate the computational efficiency of the proposed schemes. For the first three problems, we perform comparative analyses between our proposed $\mathrm{MM}_{4,1}$ method and two well-established fourth-order ODE techniques, i.e., the Runge–Kutta method [67] (denoted by RKM_4) and the Adams–Bashforth method [224] (denoted by ABM_4). For the third problem, the Mathieu equation, we additionally include the method PFML [356]. These comparisons assess accuracy and convergence behavior across different problem classes. The results are displayed in tables by taking certain parameters, i.e., iterations (n) needed to converge from a particular initial guess to the approximate solution $\hat{X}^*$, the residual error for the consecutive iterations ($||X^{(n+1)} - X^{(n)}|| = ||\hat{\xi}_{n+1}||$), computational cost, computational efficiency, and approximated computational order of convergence (ACOC). The stopping criterion used here is $||\hat{\xi}_{n+1}|| + ||F(X^{(n)})|| < 10^{-800}$.

We use $p_3 = 27/8$ in the proposed sixth-order methods $\mathrm{MM}_{6,1}$, $\mathrm{MM}_{6,2}$, and $\mathrm{MM}_{6,3}$, and in Tables 4.10–4.8 the representation $A \times 10^{(\pm m)}$ is written in the form $A(\pm m)$. For testing, we have considered following numerical examples.

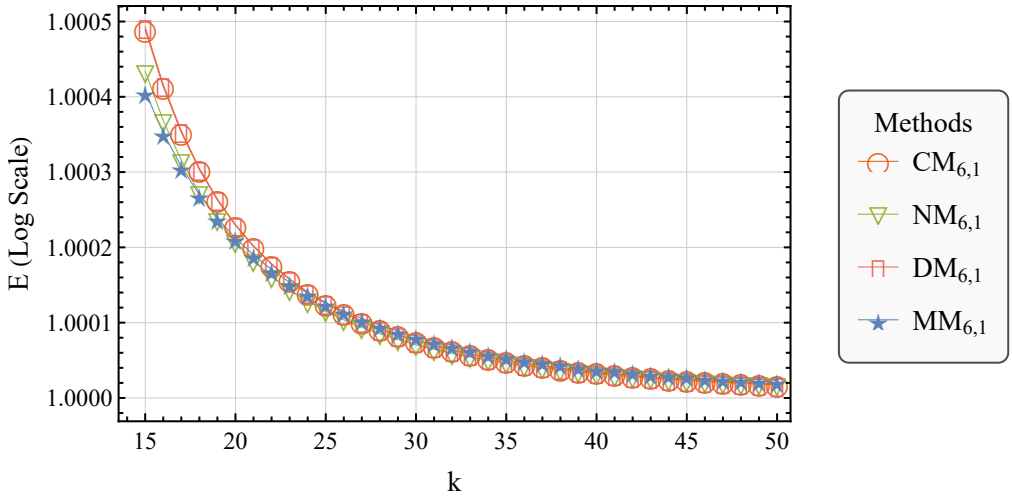
4.5.1 Modeling diffusion-reaction processes in catalyst pellets

Example 4.5.1. *A key challenge in catalytic engineering is analyzing the simultaneous diffusion and chemical reactions within porous catalyst particles. The central objective is to determine the global reaction rate exhibited by the catalyst pellet in Eq.* (4.1)*. Consider a practical example: A 2mm γ-alumina sphere with platinum (Pt) catalyst promoting cyclohexane dehydrogenation at 700K, where $k_r = 4s^{-1}$ and $\mathcal{D}_f = 0.05cm^2/s$. For a second-order reaction $R(C) = C^2$, we define dimensionless variables $\hat{C} = \frac{C}{C_s}$ and $\hat{\lambda} = \frac{r}{R_p}$. Assuming isothermal conditions, the dimensionless model becomes:*

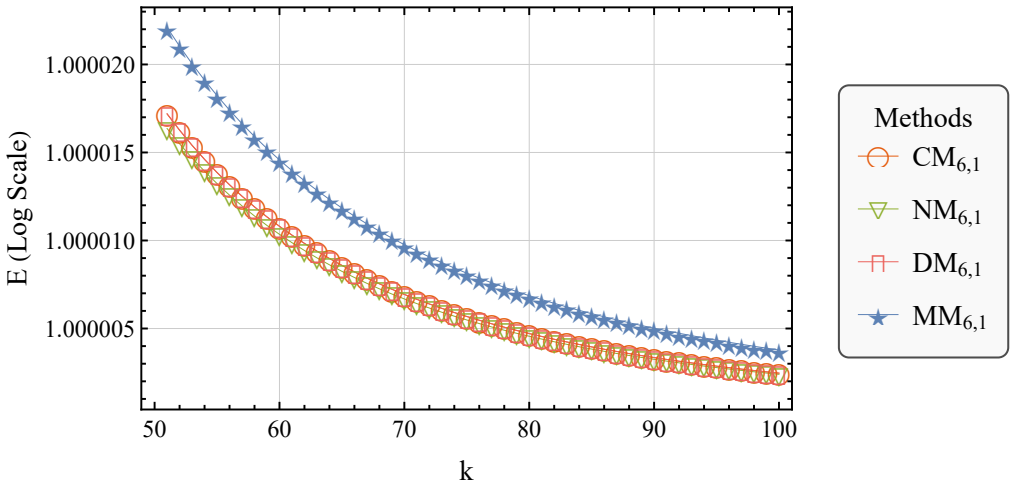
$$\begin{cases} \dfrac{d^2\hat{C}}{d\hat{\lambda}^2} + \dfrac{2}{\hat{\lambda}}\dfrac{d\hat{C}}{d\hat{\lambda}} = \phi^2\hat{C}^2, \ 0 < \hat{\lambda} < 1, \\ \dfrac{d\hat{C}}{d\hat{\lambda}}(0) = 0, \ \hat{C}(1) = 1, \end{cases} \tag{4.65}$$

where $\phi = R_p\sqrt{\frac{k_r}{\mathcal{D}_f}}$ is the Thiele modulus. The boundary value problem is discretized using central finite differences:

$$\frac{d^2\hat{C}}{d\hat{\lambda}^2} \approx \frac{X_{j+1} - 2X_j + X_{j-1}}{\hat{\hbar}^2}, \ \frac{d\hat{C}}{d\hat{\lambda}} \approx \frac{X_{j+1} - X_{j-1}}{2\hat{\hbar}}, \tag{4.66}$$



(a) For $k = 15, 16, \cdots, 50$.



(b) For $k = 51, 52, \cdots, 100$.

Figure 4.3: Efficiency plots for sixth-order methods on small k-scale systems.

where the step size is $\hat{\hbar} = 1/(k+1)$ *and* $X_s = \hat{C}(\hat{\lambda}_s)$, *and* $\hat{\lambda}_s = 0 + s\hat{\hbar}$, $0 \leq s \leq k+1$, *after substituting it gives the nonlinear system as*

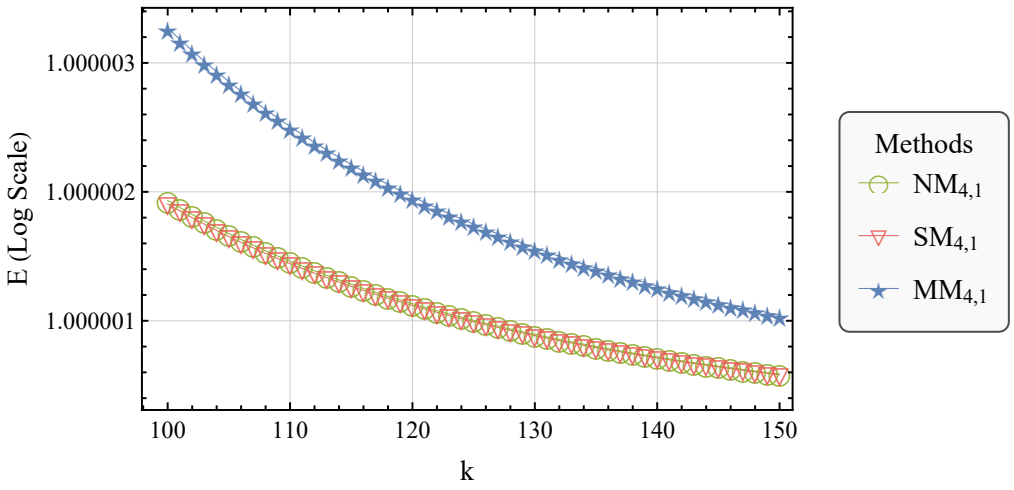
$$\left(1 - \frac{1}{s}\right) X_{s-1} + \left(1 + \frac{1}{s}\right) X_{s+1} - 2X_s = \hat{\hbar}^2 \phi^2 X_s^2, \ s = 1, 2, \cdots, k, \tag{4.67}$$

and $X_{k+1} = 1$. *At* $\hat{\lambda} = 0$, *the differential equation simplifies, as its second component becomes* $\lim_{\hat{\lambda}\to 0} \frac{\hat{C}'}{\hat{\lambda}} = \hat{C}''$. *As a result, it becomes* $3\hat{C}'' - \phi^2 \hat{C}^2 = 0$, *then we get* $X_1 - 2X_0 + X_{-1} - \frac{1}{3}\hat{\hbar}^2\phi^2 X_0 = 0$. *A central difference approximation is applied to the boundary condition* $\hat{C}'(0) = 0$, *leading to* $X_{-1} = X_1$, *hence our first nonlinear equation would be* $X_1 - X_0 - \frac{1}{6}\hat{\hbar}^2\phi^2 X_0^2 = 0$. *Thus, problem* (4.1) *is*

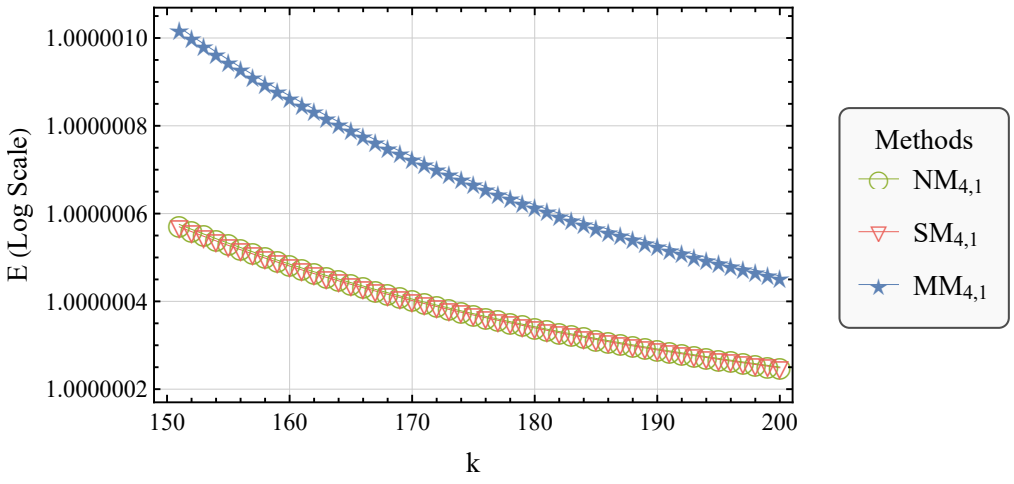
governed by the system $F(X) = 0$, *where* $F : \mathbb{R}^{k+1} \to \mathbb{R}^{k+1}$ *given by*

$$F(X) = \begin{cases} X_1 - X_0 - \dfrac{1}{6}\hat{h}^2\phi^2 X_0^2, \\ \left(1 - \dfrac{1}{s}\right) X_{s-1} + \left(1 + \dfrac{1}{s}\right) X_{s+1} - 2X_s - \frac{1}{6}\hat{h}^2\phi^2 X_s^2, \quad s = 1, 2, \cdots, k-1, \\ \left(1 - \dfrac{1}{k}\right) X_{k-1} + 1 + \dfrac{1}{k} - 2X_k. \end{cases} \tag{4.68}$$

Using an initial guess of $X^{(0)} = (1/10, \cdots, 1/10)^T$ *and* $k = 30$, *the proposed methods converged to*

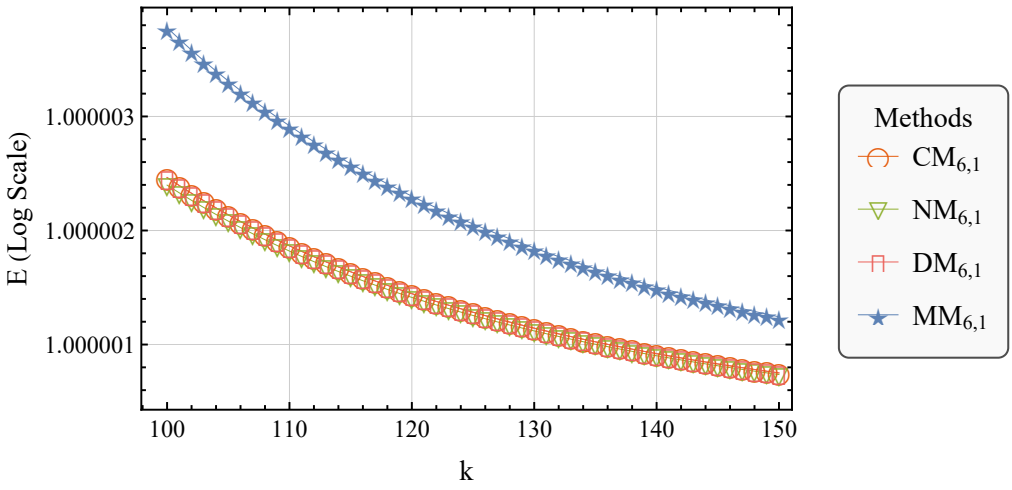


(a) For $k = 101, 102, \cdots, 150$.

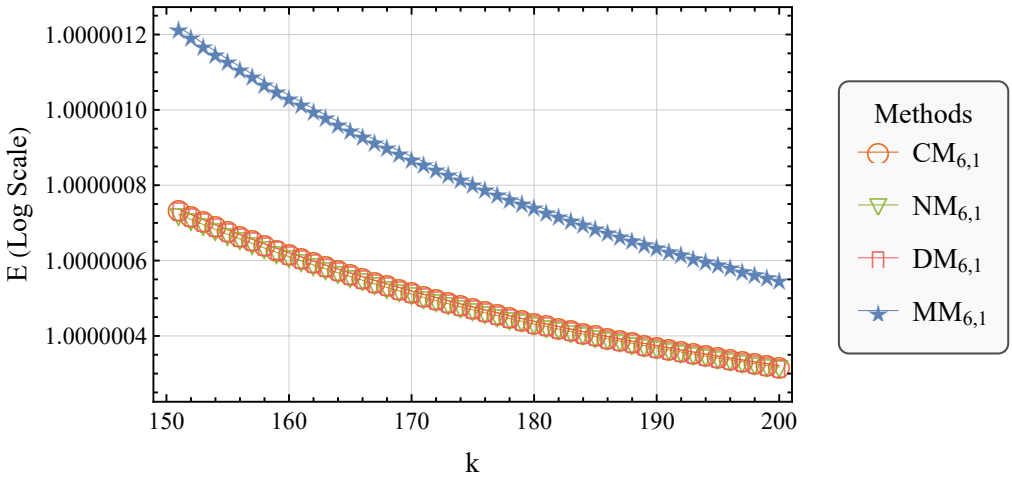


(b) For $k = 151, 152, \cdots, 200$.

Figure 4.4: Efficiency plots for fourth-order methods on large k-scale systems.



(a) For $k = 101, 102, \cdots, 150$.



(b) For $k = 151, 152, \cdots, 200$.

Figure 4.5: Efficiency plots for sixth-order methods on large k-scale systems.

the approximate solution:

$$
\begin{aligned}
X^* = (&0.03631659227845877\ldots,\ 0.03638978797043073\ldots,\ 0.03670815069254177\ldots,\\
&0.03720385359214455\ldots,\ 0.03786999171157227\ldots,\ 0.03871204043974698\ldots,\\
&0.03974123599837580\ldots,\ 0.04097330169422567\ldots,\ 0.04242848301604330\ldots,\\
&0.04413211895514183\ldots,\ 0.04611558388678215\ldots,\ 0.04841760397912291\ldots,\\
&0.05108602866466188\ldots,\ 0.05418019901879434\ldots,\ 0.05777412778747872\ldots,\\
&0.06196080812655120\ldots,\ 0.06685812114702312\ldots,\ 0.07261704879757132\ldots,
\end{aligned}
$$

$0.07943327307585394\ldots,\ 0.08756384999493383\ldots,\ 0.09735165744012810\ldots,$
$0.10926204524086780\ldots,\ 0.12393916715642040\ldots,\ 0.14229504912843550\ldots,$
$0.16565504019428740\ldots,\ 0.19600433870853440\ldots,\ 0.23642431664043280\ldots,$
$0.29190517591636820\ldots,\ 0.37095470033075400\ldots,\ 0.48902827574993210\ldots,$
$0.67654865292402530\ldots)^T,$

which is also depicted in Figure 4.6. Table 4.2 summarizes the corresponding computational results for the parameter set $(k, l_q, \mu_0, \mu_1) = (31, 2.375, 7.47177, 0.114074)$.

Table 4.2: Numerical comparisons on Example 4.5.1

Methods	n	$\|\|\hat{\mathcal{E}}_{n-1}\|\|$	$\|\|\hat{\mathcal{E}}_n\|\|$	$\mathcal{C}_{\hat{M}_{p,i}}$	$E_{\hat{M}_{p,i}}$	ACOC
$NM_{4,1}$	7	7.0(−568)	4.5(−2269)	25591.9	1.00005417078308	4.0000
$SM_{4,1}$	7	1.2(−426)	8.6(−1704)	26688.5	1.00005194485888	4.0001
$MM_{4,1}$	7	1.0(−550)	3.3(−2200)	18620.8	1.000074451671734	4.0001
$MM_{4,2}$	7	1.6(−494)	6.3(−1976)	18620.8	1.000074451671734	4.0000
$MM_{4,3}$	7	5.4(−763)	0	18620.8	1.000074451671734	4.0000
$CM_{6,1}$	5	6.4(−149)	4.0(−889)	26889.1	1.00006663731779	6.0007
$NM_{6,1}$	6	1.6(−738)	2.8(−4426)	28915.8	1.00006196674765	6.0001
$DM_{6,1}$	6	4.9(−469)	1.1(−2340)	27132.4	1.00006603987422	5.0028
$MM_{6,1}$	6	1.8(−794)	1.4(−4763)	23971.3	1.00007474897782	6.0011
$MM_{6,2}$	5	1.6(−156)	7.3(−937)	23971.3	1.00007474897782	6.0064
$MM_{6,3}$	5	4.8(−163)	5.1(−975)	23971.3	1.00007474897782	5.9978

Table 4.3: Comparison of iterative solvers for $\phi = 17.8885$, $\hat{\hbar} = 0.03226$

Method	**Concentration**		**Performance**		**Iterations**
	$\hat{C}(0)$	$\hat{C}(1)$	**Time (s)**	**Error**	
$MM_{4,1}$	0.0363166	1.000000	0.5906	–	7
RKM_4	0.0377591	1.000000	0.3109	1.47(−8)	–
ABM_4	0.0380469	0.999998	0.1575	2.05(−7)	–

Figure 4.6 shows the concentration profiles $\hat{C}(\hat{\lambda})$ *from all three methods. Table 4.3 reveals that ODE methods (RKM*$_4$ *and ABM*$_4$*) produce slightly higher center concentrations than MM*$_{4,1}$*. The*

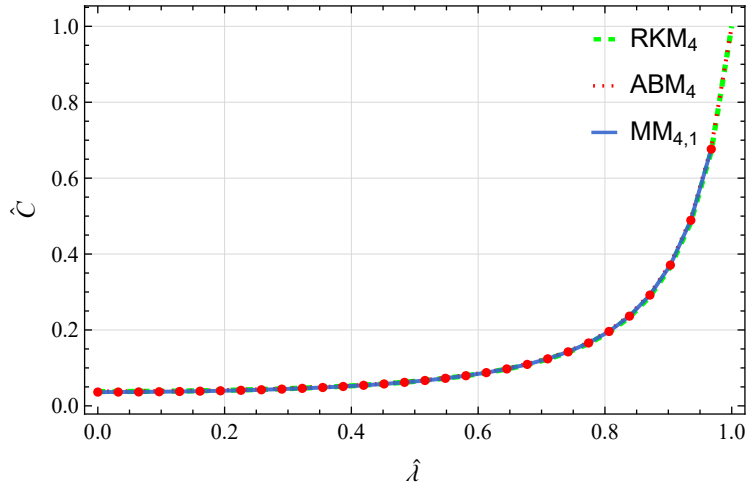


Figure 4.6: Approximate solution of catalyst pellet problem 4.5.1.

differences in $\hat{C}(1)$ values arise from the distinct numerical approaches rather than accuracy differences. Moreover, the $MM_{4,1}$ method converged in 7 iterations, whereas the ODE methods achieved comparable satisfaction of the boundary condition. The small differences in error between methods relative to our proposed method confirm the consistency of the solution. Although ABM_4 is the fastest method, $MM_{4,1}$ remains valuable for stiff problems where shooting methods may fail to converge, despite its higher computational cost from matrix operations.

4.5.2 Nonlinear reaction-diffusion equation with Van der Pol damping

Example 4.5.2. *We consider a Van der Pol equation [66] whose expression is given in Eq. (4.2). The dynamics of the Van der Pol oscillator have nonlinear damping, making it a non-conservative oscillator. It changes over time (τ) in accordance with Eq. (4.2), by which the current flow in a vacuum tube is governed. Here, u denotes the position component that depends on time τ, and μ denotes a scalar parameter that describes the damping nonlinearity and strength. Now, to solve the problem, we first partition the given interval $[0, 2]$ as*

$$\tau_0 = 0 < \tau_1 < \ldots < \tau_{n-1} < \tau_n, \ \tau_s = \tau_0 + sh, \ h = \frac{2}{n}.$$

For simplicity, we define

$$u_0 = u(\tau_0) = 0, \ u_1 = u(\tau_1), \ldots, u_{n-1} = u(\tau_{n-1}), u_n = u(\tau_n) = 1.$$

Further, we employ the mathematical formulas for first and second derivatives $u_\tau = \dfrac{X_{s+1} - X_{s-1}}{2h}$, $u_{\tau\tau} = \dfrac{X_{s-1} - 2X_s + X_{s+1}}{h^2}$, $s = 1, 2, \ldots, n-1$ *in the differential equation* (4.2). *After discretizing the Eq.* (4.2), *one gets a* $n-1$ *dimensional nonlinear system:*

$$2h^2 X_s + h\mu(X_s^2 - 1)(X_{s-1} - X_{s+1}) + 2(X_{s+1} + X_{s-1} - 2X_s) = 0, \ s = 1, 2, \ldots, n-1.$$

In particular, consider $\mu = 0.5$ *and* $n = 28$, *then, we obtain a nonlinear system of* $k = 27$ *equations in* 27 *unknowns and using initial estimate* $X^{(0)} = (0.4, 0.4, \ldots, 0.4)^T$, *the corresponding approximate solution is obtained:*

$$\begin{aligned} X^* = (&0.1016368421075541\ldots,\ 0.1992341283569330\ldots,\ 0.2925412250417608\ldots,\\ &0.3813814943256929\ldots,\ 0.4656345717973973\ldots,\ 0.5452206012476981\ldots,\\ &0.6200866994000466\ldots,\ 0.6901956779337255\ldots,\ 0.7555169051502446\ldots,\\ &0.8160191204841613\ldots,\ 0.8716649985988837\ldots,\ 0.9224072768658683\ldots,\\ &0.9681862958871847\ldots,\ 1.0089288465209730\ldots,\ 1.0445482605781440\ldots,\\ &1.0749457197638950\ldots,\ 1.1000127833275650\ldots,\ 1.1196351445584520\ldots,\\ &1.1336976154388730\ldots,\ 1.1420903039131950\ldots,\ 1.1447158873681130\ldots,\\ &1.1414977997468240\ldots,\ 1.1323890429341970\ldots,\ 1.1173812154511210\ldots,\\ &1.0965132383910720\ldots,\ 1.0698791698760060\ldots,\ 1.0376344571750050\ldots)^T, \end{aligned}$$

which is also shown in Figure 4.7 and the computational results are shown in the Table 4.4 by taking parametric values $(k, l_q, \mu_0, \mu_1) = (27, 2.375, 6.1759, 0.1958)$.

The solution profiles for the Van der Pol problem, obtained via the $MM_{4,1}$, RKM_4, *and* ABM_4 *methods, are presented in Figure 4.7 and Table 4.5. All methods satisfy the boundary condition* $u(2) \approx 1$, *but* RKM_4 *and* ABM_4 *show small boundary errors, confirming consistency across methods. The phase portrait in Figure 4.8 shows all three methods tracing nearly identical limit cycle trajectories, accurately capturing the nonlinear oscillatory dynamics. In terms of computational performance,* ABM_4 *is the fastest method but exhibits a larger boundary error. Also,* RKM_4 *offers a balanced performance in terms of speed and accuracy. Meanwhile,* $MM_{4,1}$, *although the slowest due to its matrix operations, provides a robust iteration-based solution without requiring shooting*

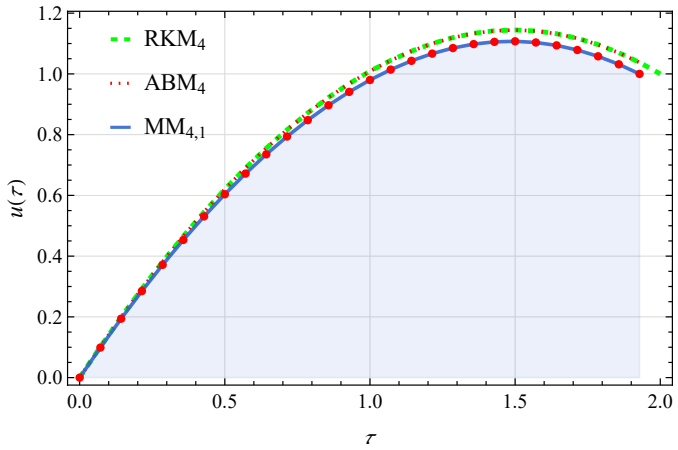


Figure 4.7: Approximate solution $u(\tau)$ for Van der Pol problem 4.5.2 comparing $\mathrm{MM}_{4,1}$, RKM_4, and ABM_4 methods.

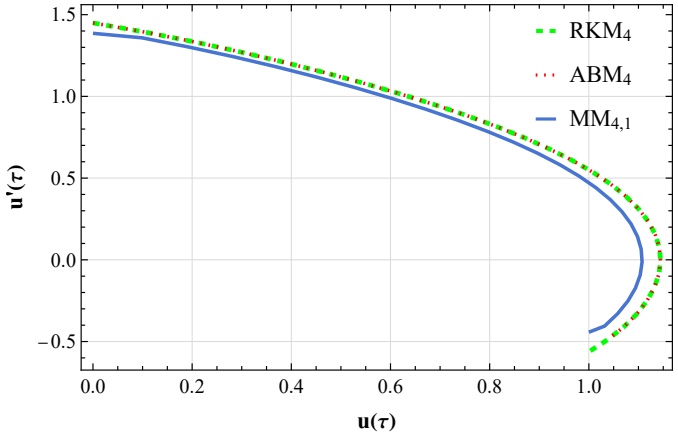


Figure 4.8: Phase portrait comparison Van der Pol problem 4.5.2.

parameter estimation.

4.5.3 Transition curves of the Mathieu equation

Example 4.5.3. *In applied mathematics, the angular Mathieu functions are special functions defined as solutions to Mathieu's differential equation:*

$$\frac{d^2y}{dx^2} + (\hat{\delta} + \varepsilon\cos(2x))y = 0, \quad x \in [0, L], \tag{4.69}$$

where $\hat{\delta}$ and ε are real-valued parameters. The behavior of solutions to equation (4.69) can be analyzed using Floquet theory [82]. The boundaries separating stable from unstable regions, referred to as transition curves, are defined by the periodic solutions of the equation. Here, we consider

Table 4.4: Numerical comparisons on Example 4.5.2

Methods	n	$\|\|\hat{\mathcal{E}}_{n-1}\|\|$	$\|\|\hat{\mathcal{E}}_n\|\|$	$\mathcal{C}_{\hat{M}_{p,i}}$	$E_{\hat{M}_{p,i}}$	ACOC
$NM_{4,1}$	5	2.3(−226)	2.1(−907)	16793.0	1.00008255539593	4.0014
$SM_{4,1}$	5	9.3(−217)	9.8(−868)	17613.1	1.00007871119091	3.9994
$MM_{4,1}$	5	9.4(−265)	1.1(−1060)	12802.6	1.00010828802436	3.9997
$MM_{4,2}$	5	7.0(−236)	1.8(−945)	12802.6	1.00010828802436	4.0006
$MM_{4,3}$	5	6.6(−212)	5.4(−849)	12802.6	1.00010828802436	3.9973
$CM_{6,1}$	4	8.9(−196)	2.3(−1177)	17752.9	1.00010093302665	6.0016
$NM_{6,1}$	4	8.8(−175)	7.9(−1052)	19275.0	1.00009296208276	6.0065
$DM_{6,1}$	5	7.9(−330)	4.2(−1650)	17902.7	1.00010008801474	4.9989
$MM_{6,1}$	4	2.1(−203)	1.8(−1223)	16806.7	1.00010661529333	6.0031
$MM_{6,2}$	4	7.9(−218)	3.6(−1310)	16806.7	1.00010661529333	5.9963
$MM_{6,3}$	4	1.9(−221)	5.2(−1331)	16806.7	1.00010661529333	5.9999

Table 4.5: Comparison of methods for Van der Pol BVP ($\mu = 0.5$, $\tau \in [0, 2]$, $h = 0.07143$)

Method	**Solution Characteristics**			**Performance**		**Iterations**
	$u'(0)$	$u(2)$	Error at $u(2)$	Time (s)	Max Error	
$MM_{4,1}$	1.38586	1.0000000	–	10.6820	–	7
RKM_4	1.44931	0.9999998	2.04(−9)	0.2154	3.85(−2)	–
ABM_4	1.44931	1.0374716	3.75(−2)	0.0057	3.84(−2)	–

following two cases (see [256]) for testing our method:

Case (i). Setting $\varepsilon = 0.001$*, the parameter* $\hat{\delta} = 1.000499968748047$*, with initial conditions* $y(0) = 0$ *and* $y'(0) = 1$*. This case lies directly on a transition curve for* (4.69)*.*

Case (ii). Again taking $\varepsilon = 0.001$*, we set* $\hat{\delta} = 0.9997918443656178$*. The initial conditions are* $y(0) = -1.557212993975872$ *and* $y'(0) = 1$*, which corresponds to a solution near to the transition curve for* (4.69)*.*

Let the domain $[0, L]$ *be divided into* $\hat{m}$ *equal intervals with step size* $h = L/\hat{m}$*. Define grid points* $x_i = ih$ *for* $i = 0, 1, \ldots, \hat{m}$*, and let* $y_i = y(x_i)$ *denote the approximate solution at* x_i*.*

Using the central difference approximation for the second derivative:

$$\frac{d^2y}{dx^2} \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}, \tag{4.70}$$

which simplifies to

$$y_{i+1} - 2y_i + y_{i-1} + h^2(\hat{\delta} + \varepsilon\cos(2x_i))y_i = 0. \tag{4.71}$$

The initial condition $y(0) = 0$ *gives* $y_0 = 0$. *For the initial derivative condition* $y'(0) = 1$, *we use a forward difference approximation* $y'(0) \approx \frac{y_1 - y_0}{h} = 1$, *The discretized equations form a nonlinear system* $F(X) = 0$, *where* $X = (y_1, \ldots, y_{\hat{n}-1})^T$. *For* $i = 0$, *we have* $y_0 = 0$, *and* $i = 1$*:*

$$F_1(X) = y_1 - h = 0. \tag{4.72}$$

For interior points $i = 2, 3, \ldots, \hat{n} - 1$ *(from* (4.71)*):*

$$\begin{aligned} F_i(X) &= y_{i+1} - 2y_i + y_{i-1} + h^2(\hat{\delta} + \varepsilon\cos(2x_i))y_i = 0, \\ F_{\hat{n}}(X) &= h^2 y_{\hat{n}+1}(\hat{\delta} + \varepsilon\cos(2h(\hat{n}+1))) + y_{\hat{n}-1} + y_{\hat{n}+1} - 2y_n = 0. \end{aligned} \tag{4.73}$$

To solve the resulting nonlinear system $F(X) = (F_1(X), F_2(X), \ldots, F_{\hat{n}}(X))^T = 0$, *we employ the* $MM_{4,1}$ *method by defining the solution vector* $X = (X_1, X_2, \ldots, X_{\hat{n}})^T$, *where* $X_i = y_{i+1}$ *for* $i = 1, 2, \ldots, \hat{n}$ *correspond to the interior grid points. Starting from an initial guess* $X^{(0)} = (0, 0, \ldots, 0)^T$ *in the method* $MM_{4,1}$ *for computing the solution. The iteration proceeds until convergence, typically requiring only 2–3 iterations to achieve* $\|F(X^{(\hat{n})})\| < 10^{-12}$. *The solution profiles obtained for the transition and near-transition cases are presented in Figures 4.9–4.10, respectively, alongside comparisons with* RKM_4, ABM_4, *and PFML methods. For* $\hat{m} = \hat{n} - 1$, $L = 10\pi$, $h = L/(\hat{n} - 1)$ *and* $\hat{n} = 1000$, *the results are obtained in the Table 4.6.*

Table 4.6: Comparative analysis of methods for the Mathieu equation solutions

Method	**Transition Case**		**Near-Transition Case**		**Iterations**
	Error	**Time (s)**	**Error**	**Time (s)**	
RKM_4	1.03(−5)	0.2083	1.79(−5)	0.2020	–
ABM_4	1.04(−5)	0.1775	1.81(−5)	0.1741	–
PFML	1.62	0.0664	3.30	0.0759	–
$MM_{4,1}$	1.00	109.41	1.89	109.84	2

Figure 4.11 illustrates the growth of absolute error relative to a high-precision reference solution for each method. Table 4.6 summarizes the maximum absolute errors at $x = 10\pi$. *The error growth results for* RKM_4 *and* ABM_4 *are consistent with those for classical stepwise solvers. In contrast,* $MM_{4,1}$ *shows a distinct error behavior due to its different solution approach, while PFML*

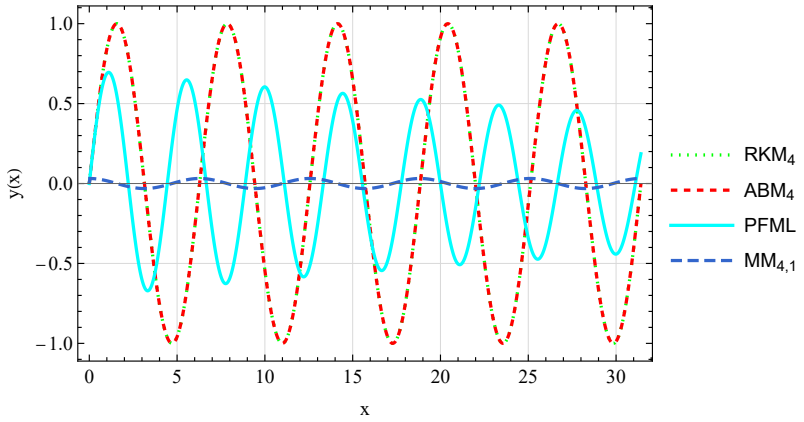


Figure 4.9: Mathieu equation on transition curve.

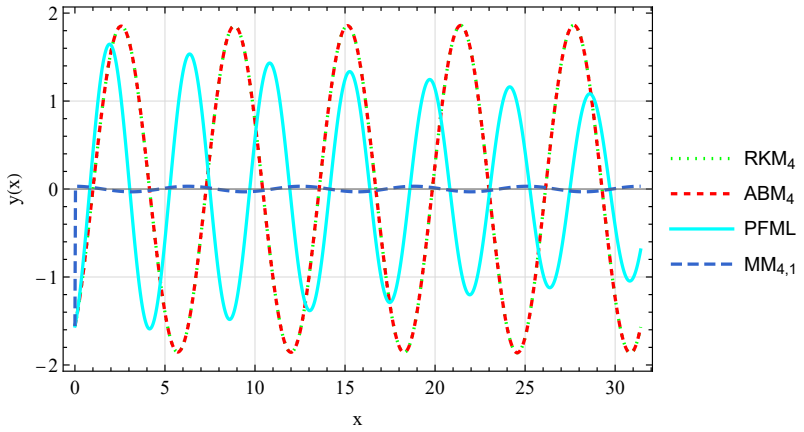


Figure 4.10: Mathieu equation on near transition curve.

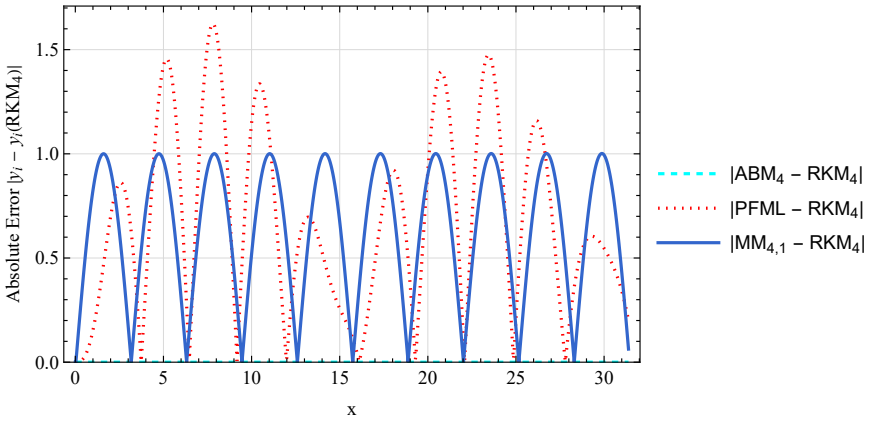


Figure 4.11: Absolute error relative error to RKM_4 on Mathieu equation.

demonstrates reduced error accumulation in oscillatory problems, aligning with its frequency-adapted design. The $MM_{4,1}$ method offers an alternative to classical ODE methods by solving the Mathieu equation iteratively rather than step-wise. While computationally more expensive, it provides robust

convergence.

4.5.4 Burger's equation

Example 4.5.4. *Here, we consider Burger's equation (see [289]), whose mathematical model is given in* (4.3)*. The function* $u = u(l, r)$ *is the required solution. Now, we partition the given domain* $[0, 1] \times [0, 1]$ *as follows*

$$\begin{aligned} &0 = l_0 < l_1 < l_2 < \cdots < l_{s-1} < l_s = 1, \qquad l_{i+1} = l_i + h, \\ &0 = r_0 < r_1 < r_2 < \cdots < r_{s-1} < r_s = 1, \quad r_{\nu+1} = r_\nu + h,\ h = 1/s, \end{aligned}$$

where $0 \leq i, \nu \leq s-1$*. We denote* $u_{i,\nu} = u(l_i, r_\nu)$ *and* $R_{i,\nu} = R(l_i, r_\nu)$ *for* $i, \nu = 0, 1, \cdots, s$*, then the boundary conditions takes the form:* $u_{0,\nu} = u(l_0, r_\nu) = 0,\ u_{s,\nu} = u(l_s, r_\nu) = 0,\ u_{i,0} = u(l_i, r_0) = 10l_i(l_i - 1)$*, and* $u_{i,s} = u(l_i, r_s) = -10(1 - l_i)l_i/e$*.*

In (4.3)*, the partial derivatives are approximated by*

$$\frac{\partial u}{\partial r} = \frac{u_{i,\nu+1} - u_{i,\nu-1}}{2h}, \frac{\partial u}{\partial l} = \frac{u_{i+1,\nu} - u_{i-1,\nu}}{2h},\ \text{and}\ \frac{\partial^2 u}{\partial l^2} = \frac{u_{i+1,\nu} - 2u_{i,\nu} + u_{i-1,\nu}}{h^2},$$

and we obtain the following nonlinear system of $(s-1) \times (s-1)$ *dimension:*

$$2h^2 R_{i,\nu} + u_{i-1,\nu}(2 - hu_{i,\nu}) + h(u_{i,\nu-1} - u_{i,\nu+1}) - u_{i,\nu}(4 - hu_{i+1,\nu}) + 2u_{i+1,\nu} = 0, \tag{4.74}$$

where $i, \nu = 1, 2, \cdots, s-1$*. Denoting* $u_{i,\nu} \to X_{\nu+(i-1)(s-1)}$ *for each* i *and* ν*. Specifically, we set* $s = 11$*, which corresponds to a 100-dimensional nonlinear system. Further, the obtained approximated solution is displayed in Figure 4.12 for the problem* (4.74) *by taking initial guess* $u_{i,\nu} = -\frac{14}{10}$ *for* $i, \nu = 1, 2, \cdots, 10$*.*

The estimated parametric values are $(k, l_q, \mu_0, \mu_1) = (100, 2.375, 4.37, 0.01)$*, for which the performance of the schemes is given in Table 4.7.*

4.5.5 Heat conduction problem

Example 4.5.5. *Now, we consider the heat conduction problem (see [289]) whose mathematical expression is given in Eq.* (4.4) *where* $u = u(l, r)$ *is the required solution, and* l *and* r *are spatial and temporal domains, respectively. We choose the partition of domain* $[0, 1] \times [0, 1]$ *with maximum*

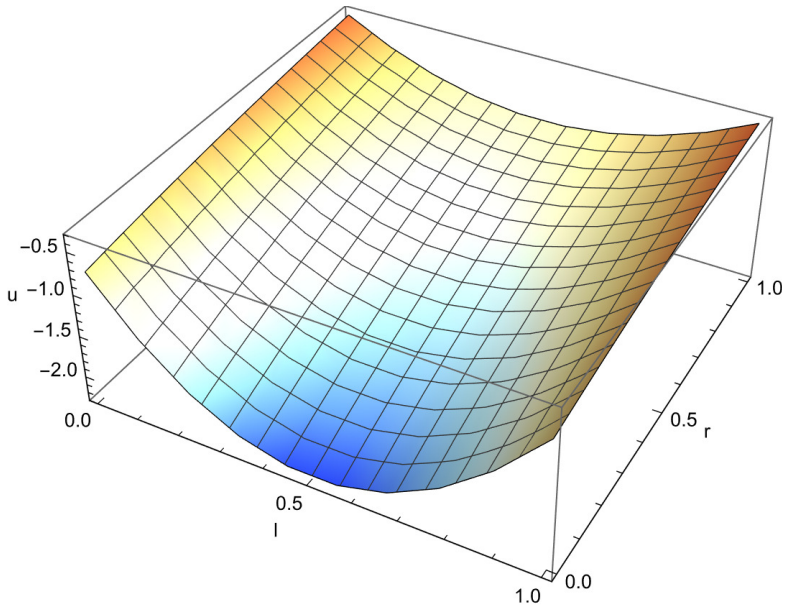


Figure 4.12: Approximate solution of Burger's equation.

Table 4.7: Numerical comparisons on Example 4.5.4

Methods	n	$\|\|\hat{\mathcal{E}}_{n-1}\|\|$	$\|\|\hat{\mathcal{E}}_{n}\|\|$	$\mathcal{C}_{\hat{M}_{p,i}}$	$E_{\hat{M}_{p,i}}$	ACOC
$NM_{4,1}$	6	1.5(−396)	6.7(−1590)	711325	1.00000194889352	4.0012
$SM_{4,1}$	6	8.1(−357)	1.1(−1430)	721662	1.000001920976430	4.0004
$MM_{4,1}$	6	4.6(−333)	6.8(−1336)	411693	1.00000336730489	3.999
$MM_{4,2}$	6	2.9(−378)	0	411693	1.00000336730489	4.0000
$MM_{4,3}$	6	2.6(−305)	7.7(−1225)	411693	1.00000336730489	4.0004
$CM_{6,1}$	5	5.9(−322)	2.3(−1937)	721999	1.00000248166783	5.9979
$NM_{6,1}$	5	8.1(−305)	1.8(−1834)	742237	1.00000241400367	6.0010
$DM_{6,1}$	6	1.8(−730)	1.1(−3653)	722374	1.00000248038126	4.9987
$MM_{6,1}$	5	2.3(−292)	1.9(−1759)	462843	1.00000387121314	5.9990
$MM_{6,2}$	5	7.4(−285)	5.4(−1716)	462843	1.00000387121314	6.0046
$MM_{6,3}$	5	3.0(−269)	1.1(−1621)	462843	1.00000387121314	6.0044

value of $r = 1$ *by*

$$l_i = 0 + ih,\ i = 0, 1, \cdots, n_l,\ h = 1/n_l,$$

$$r_\nu = 0 + \nu s,\ \nu = 0, 1, \cdots, n_r,\ s = 1/n_r,$$

where n_l and n_r denotes the number of sub-intervals in l and r domains, respectively.

Let us take $u_{i,\nu} = u(l_i, r_\nu)$ and $\hat{H}_{i,\nu} = \hat{H}(l_i, r_\nu)$ for $i = 0, 1, \cdots, n_l$, and $\nu = 0, 1, \cdots, n_r$. The model (4.4) has been discretized using the following approximation to partial derivatives

$$\frac{\partial u}{\partial l} = \frac{u_{i+1,\nu} - u_{i-1,\nu}}{2h}, \ \frac{\partial u}{\partial r} = \frac{u_{i,\nu} - u_{i,\nu-1}}{s}, \ \text{and} \ \frac{\partial^2 u}{\partial l^2} = \frac{u_{i-1,\nu} - 2u_{i,\nu} + u_{i+1,\nu}}{2h},$$

and correspondingly, we obtain the nonlinear system of $(n_l - 1) \times n_r$ equations as follows

$$(2-h)su_{i+1,\nu} - 2(2s + h^2)u_{i,\nu} + (2+h)su_{i-1,\nu} + 2sh^2u_{i,\nu}^2 + 2h^2u_{i,\nu-1} = 2sh^2\hat{H}_{i,\nu}, \tag{4.75}$$

where $i = 1, 2, \cdots, n_l - 1$, and $\nu = 1, 2, \cdots, n_r$. Denoting $u_{i,\nu} \to X_{\nu+(i-1)(n_l-1)}$ for each i and ν. In particular, we set $n_l = 11$ and $n_r = 10$, resulting in a nonlinear system with 10×10 equations. Further, the approximated solution of the Eq. (4.75) is presented in the Figure 4.13 with initial estimate of $u_{i,\nu} = \sin\left(\frac{\pi i}{11}\right)$ for each ν.

The estimated parametric values for this problem are $(k, l_q, \mu_0, \mu_1) = (100, 2.375, 6.26, 0.01)$, for

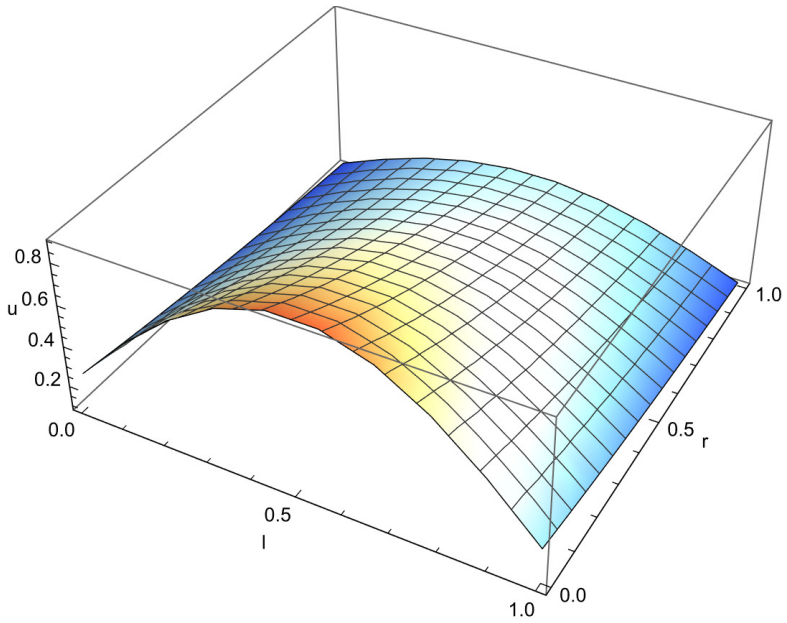


Figure 4.13: Approximate solution of heat conduction problem.

which the performance of the methods is displayed in Table 4.8.

Table 4.8: Numerical comparisons on Example 4.5.5

Methods	n	$\|\|\hat{\mathcal{E}}_{n-1}\|\|$	$\|\|\hat{\mathcal{E}}_n\|\|$	$\mathcal{C}_{\hat{M}_{p,i}}$	$E_{\hat{M}_{p,i}}$	ACOC
$NM_{4,1}$	5	2.6(−272)	3.0(−1092)	711514	1.00000194837584	3.9991
$SM_{4,1}$	5	3.7(−256)	1.8(−1027)	721851	1.00000192047346	3.9991
$MM_{4,1}$	5	6.1(−282)	1.6(−1130)	411882	1.00000336575974	4.0000
$MM_{4,2}$	5	7.3(−278)	1.0(−1114)	411882	1.00000336575974	3.9991
$MM_{4,3}$	5	1.5(−272)	8.5(−1093)	411882	1.00000336575974	4.0000
$CM_{6,1}$	4	4.5(−217)	0	722377	1.00000248036924	6.0000
$NM_{6,1}$	4	1.8(−208)	3.0(−1255)	742615	1.00000241277491	5.9972
$DM_{6,1}$	5	2.0(−608)	1.7(−3045)	722941	1.00000247843590	5.0009
$MM_{6,1}$	4	4.7(−221)	0	463221	1.00000386805413	6.0000
$MM_{6,2}$	4	2.7(−234)	0	463221	1.00000386805413	6.0000
$MM_{6,3}$	4	5.1(−232)	0	463221	1.00000386805413	6.0000

4.5.6 Academic problem

Example 4.5.6. *Now, we take the mixed Hammerstein integral equation [253]:*

$$X(r) = 1 + \frac{1}{5}\int_0^1 \mathcal{M}(r,l)X(l)^3 dl, \tag{4.76}$$

where $X \in C[0,1]$, $r, l \in [0,1]$ *and* $\mathcal{M}(r,l) = \begin{cases} (1-r)l, & l \leq r, \\ r(1-l), & r < l \end{cases}$ *is the kernel. We convert the equation* (4.76) *into a finite-dimensional case by employing the Gauss-Legendre quadrature formula as*

$$\int_0^1 g(l)dl \approx \sum_{j=1}^{m} w_j g(l_j),$$

where w_j *and* l_j *denotes the weights and abscissas, respectively, for* $m = 8$. *We approximate* $X(l_s)$ *with* X_s *for* $1 \leq s \leq 8$, *and therefore we get*

$$5X_s - \sum_{j=1}^{8} a_{sj} X_j^3 - 5 = 0, \ s = 1, 2, \ldots, 8,$$

Table 4.9: Abscissas and weights of Gauss-Legendre quadrature formula for $m = 8$

j	l_j	w_j
1	0.019855071751231884158219570	0.050614268145188129576265670
2	0.101666761293186630204223030	0.111190517226687235272178000
3	0.237233795041835507091130470	0.156853322938943643668981100
4	0.408282678752175097530261930	0.181341891689180991482575220
5	0.591717321247824902469738070	0.181341891689180991482575220
6	0.762766204958164492908869520	0.156853322938943643668981100
7	0.898333238706813369795776960	0.111190517226687235272178000
8	0.980144928248768115841780430	0.050614268145188129576265670

where $a_{sj} = \begin{cases} l_j X_j(1-l_s), & j \leq s, \\ l_s X_j(1-l_j), & s < j \end{cases}$. *The initial guess considered here is* $X^{(0)} = (1, 1, \ldots, 1)^T$ *and the corresponding approximate solution obtained here is*

$$X^* = (1.002096245031123, 1.009900316187357, 1.019726960993045, 1.026435743030670,$$
$$1.026435743030670, 1.019726960993045, 1.009900316187357, 1.002096245031123)^T.$$

This problem has parametric values $(k, l_q, \mu_0, \mu_1) = (8, 2.375, 11, 1.25)$, *and the computational results are shown in Table 4.10.*

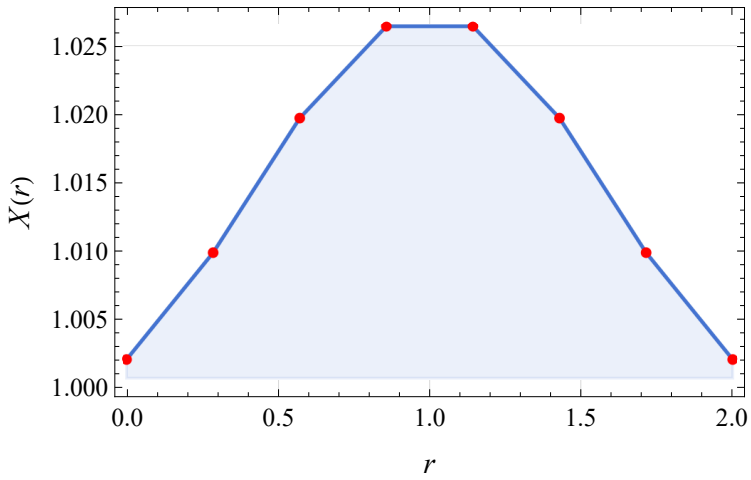


Figure 4.14: Approximate solution of Hammerstein integral problem.

Table 4.10: Numerical comparisons on Example 4.5.6

Methods	n	$\|\|\hat{\mathcal{E}}_{n-1}\|\|$	$\|\|\hat{\mathcal{E}}_{n}\|\|$	$\mathcal{C}_{\hat{M}_{p,i}}$	$E_{\hat{M}_{p,i}}$	ACOC
$NM_{4,1}$	5	1.3(−647)	8.0(−2592)	891.0	1.00155709694568	4.0000
$SM_{4,1}$	5	2.6(−640)	1.4(−2562)	982.0	1.00141270197670	4.0000
$MM_{4,1}$	5	1.8(−669)	1.4(−2679)	978.5	1.00141775865558	4.0000
$MM_{4,2}$	5	2.2(−650)	5.3(−2602)	978.5	1.00141775865558	4.0000
$MM_{4,3}$	5	1.4(−762)	1.4(−3053)	978.5	1.00141775865558	3.9923
$CM_{6,1}$	4	2.6(−532)	6.9(−3196)	1062.0	1.00168857985717	6.0000
$NM_{6,1}$	4	5.1(−524)	5.7(−3146)	1209.0	1.00148311648887	6.0000
$DM_{6,1}$	-	2.2(−345)	Diverges	1145.0	1.00156608045681	Diverges
$MM_{6,1}$	4	3.3(−536)	2.2(−3219)	1443.5	1.00124203113928	6.0000
$MM_{6,2}$	4	1.3(−542)	3.3(−3256)	1443.5	1.00124203113928	6.0000
$MM_{6,3}$	4	2.2(−549)	8.2(−3299)	1443.5	1.00124203113928	6.0000

The numerical results demonstrate the stability of the suggested approaches in Tables 4.10–4.8 in terms of all parameters like iteration count, residual error, computing cost, and computational efficiency. Moreover, the ACOC for each method strongly supports the theoretical convergence order.

4.6 Basins of attraction

Now, we study the stability of proposed techniques through 'basins of attraction' when applied to different nonlinear systems; for more details, see [352]. We look at the following three test problems to analyze the attraction basins related to the proposed variants:

Test problem 1: Consider a nonlinear system

$$S_1(X_1, X_2) = \begin{cases} X_1^2 - 1 & = 0, \\ X_2^2 - 1 & = 0, \end{cases} \tag{4.77}$$

that has four exact solutions $X_1^* = (1,1)^T$, $X_2^* = (1,-1)^T, X_3^* = (-1,1)^T$, $X_4^* = (-1,-1)^T$, where green, yellow, blue, and red colors are allocated to each initial point that converges to the solutions X_1^*, X_2^*, X_3^* and X_4^*, respectively.

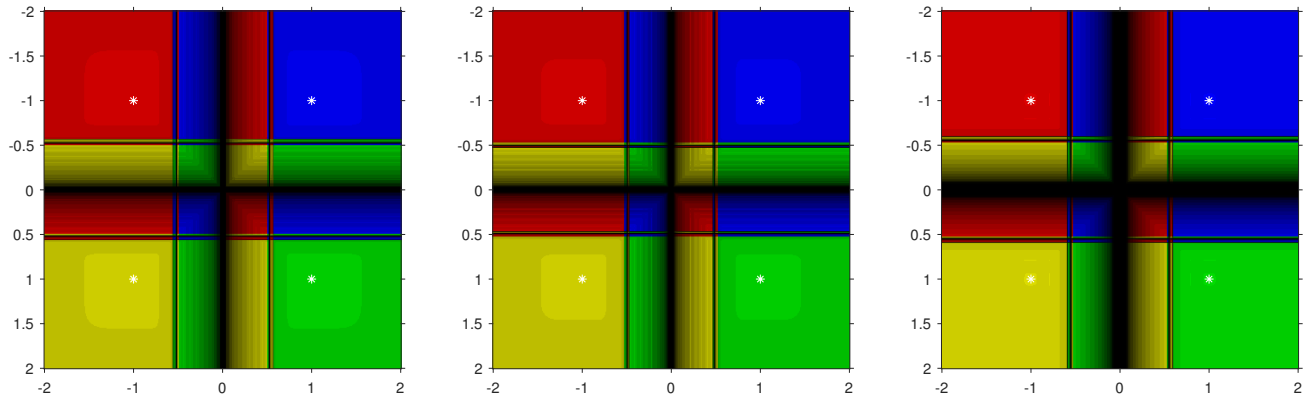


Figure 4.15: Attraction basins of $MM_{4,1}$, $MM_{4,2}$ and $MM_{4,3}$ methods, for the test problem 1.

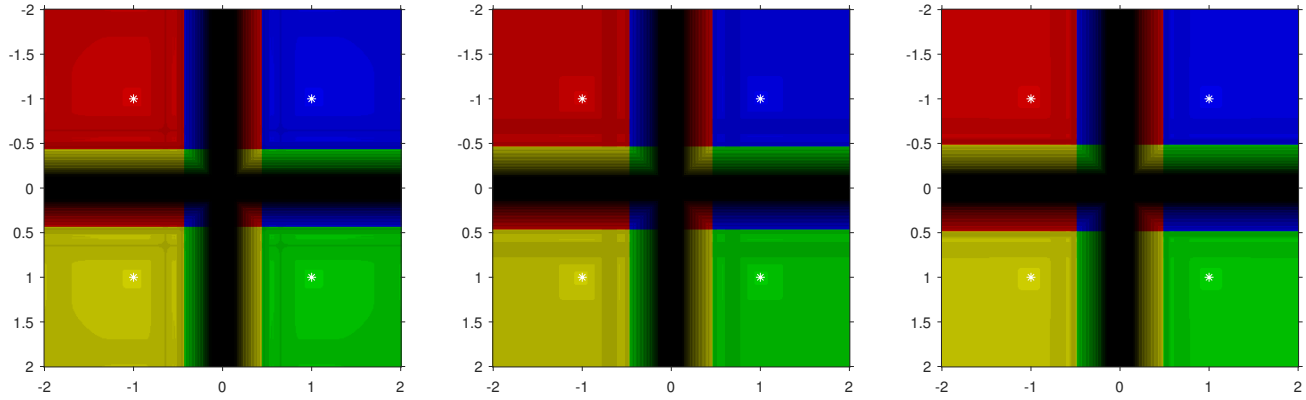


Figure 4.16: Attraction basins of $MM_{6,1}$, $MM_{6,2}$ and $MM_{6,3}$ methods, for the test problem 1.

Test problem 2: Let us suppose a nonlinear system

$$S_2(X_1, X_2) = \begin{cases} -X_1 + X_1^3 - 3X_1X_2^2 & = 0, \\ -X_2 + 3X_1^2X_2 - X_2^3 & = 0, \end{cases} \tag{4.78}$$

having three solutions $X_1^* = (-1, 0)^T$, $X_2^* = (0, 0)^T$, $X_3^* = (1, 0)^T$, where cyan, purple, and yellow colors are allocated to each initial point that converges to the solutions X_1^*, X_2^*, and X_3^*, respectively.

Test problem 3: Now consider

$$S_3(X_1, X_2) = \begin{cases} X_1 + X_1^5 - 10X_1^3X_2^2 + 5X_1X_2^4 & = 0, \\ X_2 + 5X_1^4X_2 - 10X_1^2X_2^3 + X_2^5 & = 0, \end{cases} \tag{4.79}$$

that has five solutions $X_1^* = (0, 0)^T$, $X_2^* = (-0.707107, -0.707107)^T$, $X_3^* = (-0.707107, 0.707107)^T$,

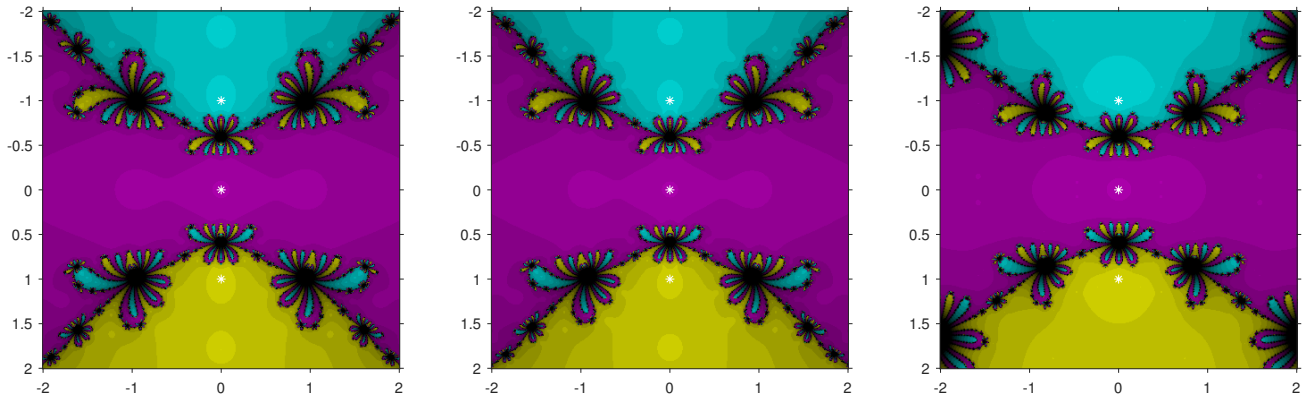


Figure 4.17: Attraction basins of $MM_{4,1}$, $MM_{4,2}$ and $MM_{4,2}$ methods, for the test problem 2.

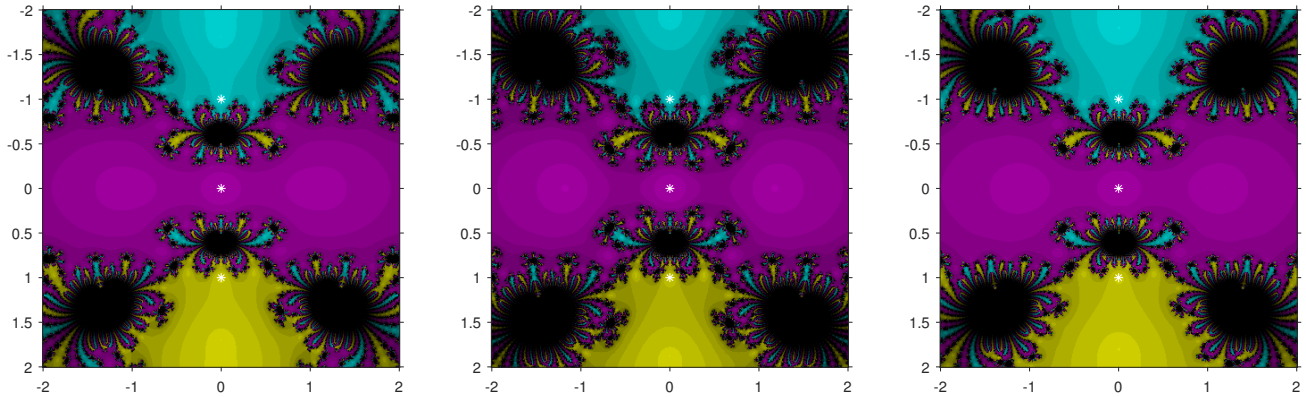


Figure 4.18: Attraction basins of $MM_{6,1}$, $MM_{6,2}$ and $MM_{6,3}$ methods, for the test problem 2.

$X_4^* = (0.707107, -0.707107)^T$, $X_5^* = (0.707107, 0.707107)^T$, where blue, green, red, pink, and yellow colors are allocated to each initial point that converges to the solutions X_1^*, X_2^*, X_3^*, X_4^* and X_5^*, respectively.

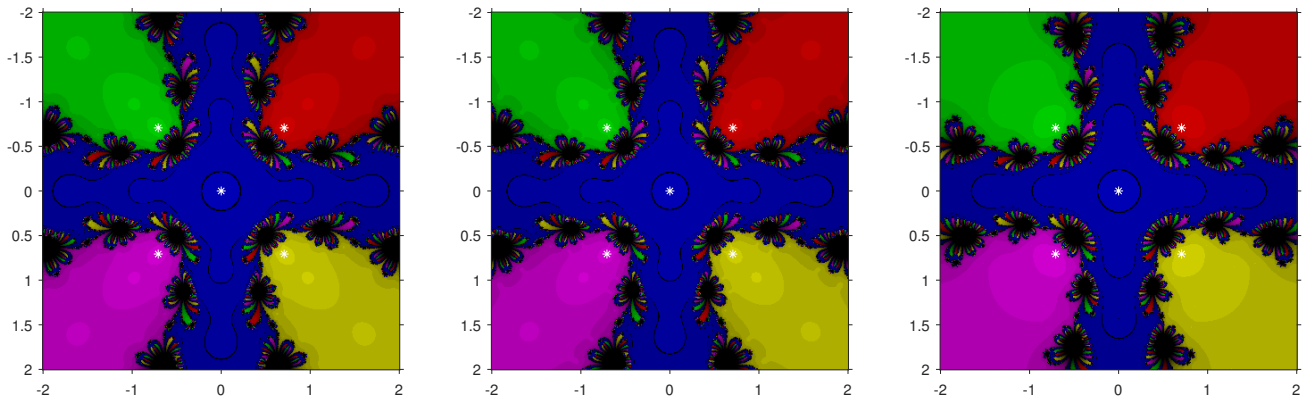


Figure 4.19: Attraction basins of $MM_{4,1}$, $MM_{4,2}$ and $MM_{4,3}$ methods, for the test problem 3.

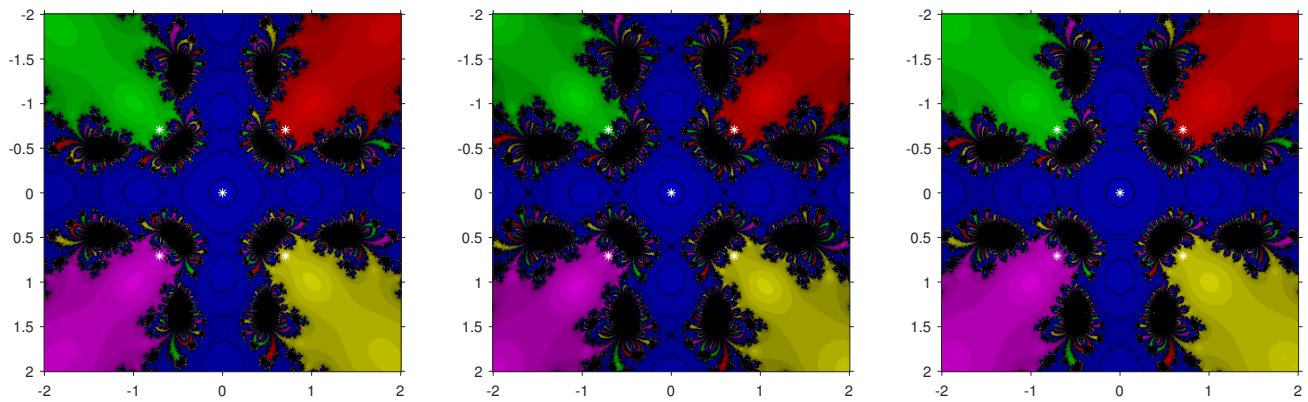


Figure 4.20: Attraction basins of $MM_{6,1}$, $MM_{6,2}$ and $MM_{6,3}$ methods, for the test problem 3.

We explore the attraction basins of the proposed techniques by analyzing a specified area $[-2, 2]\times[-2, 2]$ with a mesh of 401×401 points enclosing each desired solution. If, for the given initial point, the iterative algorithm converges to the desired solution X^*, then it will be colored according to the assigned color of X^*; else, the points are considered divergent initial points, hence colored black. Figures 4.15–4.20 display the attraction basins of the suggested algorithms, where the solutions are marked by white stars, notably in Figures 4.15–4.20, the basins show broader coverage and stable behavior when solving nonlinear systems $S_1 - S_3$ in test problems 1–3.

4.7 Concluding remarks

In this chapter, we have introduced two new multi-point iterative families with fourth- and sixth-order convergence for approximating solutions of nonlinear systems. A novel generalized $(q+1)$-step family with convergence order increased up to $2q + 2$, where $q \in \mathbb{N}$, has been further developed and investigated. Moreover, using a single inverse operator throughout each iteration makes the scheme computationally more efficient. Analyzing the efficiency indices of the proposed family and comparing them to the efficiencies of existing procedures supports this idea. The numerical testing of the new techniques is executed on academic problems and several real-life models, including boundary value problems, Hammerstein integral equations, Burger's equations, and heat conduction problems. A performance comparison of the new methods with the previous ones reveals that they perform noticeably better in terms of efficiency and convergence behavior. It is essential to highlight that the proposed techniques are generally applicable, but they are particularly effective for computing numerical solutions of large nonlinear systems. Finally, the dynamical planes of

different test problems are obtained using new schemes to demonstrate the stability.

The next chapter expands the applicability of the presented family by introducing a derivative-free iterative family for solving vectorial nonlinear equations. The methods are specifically designed to address problems where computing the derivative is computationally complex.

Chapter 5

Family of Steffensen-like iterative methods for solving nonlinear systems

In this chapter, we propose a third-order two-step and a fifth-order three-step iterative family that utilizes a derivative-free approach, requiring two frozen divided differences and one matrix inversion per iteration. Additionally, we derive a general multi-point family using a fifth-order scheme as a predictor. The proposed families are analyzed both theoretically and numerically, and experimentation is conducted using a range of numerical problems to confirm the theoretical results on convergence order and computational efficiency. The dynamical properties of each iterative method are studied by means of basins of attraction.

5.1 Introduction

In various fields, including chemistry [39], engineering [216], applied mathematics [107, 325, 326], and physics, computing an efficient solution of nonlinear models is crucial. For instance, the Frank-Kamenetskii theory [189] elucidates the thermal explosion phenomenon in a homogeneous mixture of reactants confined in a closed vessel with walls maintained at a constant temperature, and heat transfer within porous catalyst particles often leads to nonlinear systems

$$F(X) = 0, \tag{5.1}$$

The contents of this chapter are published in Journal of Mathematical Chemistry, Vol. 62, pp. 109–144, 2024 (SCIE, I.F.: 2.0)

where $X = (X_1, X_2, \cdots, X_k)^T$ is a vector in $\mathbb{R}^k$, $F : \mathcal{D} \subseteq \mathbb{R}^k \to \mathbb{R}^k$ is nonlinear mapping, and $\mathcal{D}$ denotes the convex subset of $\mathbb{R}^k$. Moreover, the traveling wave solutions that switch between equilibrium states, as well as the auto-catalytic chemical reactions, involve nonlinear reaction-diffusion terms [238, 289]. A study of hydrogen's electronic structure under strong magnetic fields can also be numerically addressed through iterative procedures (see Lograd o and Vianna [236]). Importantly, the numerical technique allows researchers to validate models of observed phenomena [177] by applying it to distinct physical and chemical problems.

Several mathematicians have developed different iterative algorithms to study this area of research. Among these, the classical Newton's method is widely used for solving (5.1) due to its quadratic rate of convergence, given as follows

$$X^{(n+1)} = X^{(n)} - F'(X^{(n)})^{-1} F(X^{(n)}), \quad n \in \hat{\mathbb{N}}_0. \tag{5.2}$$

This method requires $F(X)$ to be continuously differentiable, and the initial estimate $(X^{(0)})$ should be close enough to the root. Here, $F'(X) \in \mathcal{L}(\mathbb{R}^k, \mathbb{R}^k)$ denotes the linear operator or Fréchet derivative of the vector function $F(X)$. To achieve quadratic order of convergence, the scheme (5.2) requires one function evaluation (F), and just one matrix inverse $(F')^{-1}$ per iteration. Additionally, Traub-Steffensen [351] introduced the following method that significantly improves the scheme (5.2) by retaining quadratic convergence without the need for derivatives

$$X^{(n+1)} = X^{(n)} - [u^{(n)}, X^{(n)}; F]^{-1} F(X^{(n)}), \quad n \in \hat{\mathbb{N}}_0, \tag{5.3}$$

where $[u^{(n)}, X^{(n)}; F]$ represents the first-order divided difference (DD) and β is a nonzero constant that can be chosen arbitrarily. It is worth noting that when $\beta = 1$, the scheme (5.3) simplifies to the well-known Steffensen's method [346]. For further insights and comprehensive discussions on this topic, interested readers are encouraged to explore relevant articles by Abad et al. [1], Alqahtani et al. [15], Argyros and Magreñán [19], Chicharro et al. [74], Cordero et al. [84, 93], Kansal et al. [188], Kelley [207], Ortega and Rheinboldt [263] and their references therein.

Although multiple higher-order techniques involving derivatives are available in the literature, finding efficient iterative methods for solving nonlinear systems of derivative-free nature remains a challenging task, and only a few efficient approaches are currently available. However, to improve the efficiency of Traub-Steffensen's scheme, various researchers have suggested different higher-order methods over the years (see Zheng et al. [382], Kansal et al. [189], Grau-Sánchez et al. [141],

Sharma and Arora [301, 302]). However, these solvers converge to the solution at different rates and can achieve a solution by increasing the convergence order, decreasing the computational cost of the algorithm, or both.

In addition, some seventh-order derivative-free methods have been developed by Wang et al. [360], Sharma et al. [302], which require five DDs, some matrix inversions, and other functional evaluations per step. In 2015, Wang et al. [361] introduced a derivative-free seventh-order procedure that requires three DDs and one matrix inverse per step. These procedures demonstrate notable efficacy when a well-informed initial approximation is provided, rendering them appealing for practical applications. However, it is worth noting that the computational cost of these methods increases due to the substantial number of divided differences involved.

With this motivation, we introduce a novel family of derivative-free methods that exhibit a third-order convergence rate. Notably, each iteration of this family requires only one matrix inverse, two function evaluations, and two DDs. The convergence rate is further enhanced to fifth order by incorporating just one additional function evaluation. We modify the technique to include $q+1$ steps, achieving even higher convergence rates with an increasing convergence order of $2q+1$, where $q \in \mathbb{N}$. This modified technique offers the advantage of an increase in convergence order up to two at the cost of one additional function evaluation per step. Notably, the calculation of the inverse matrix remains constant throughout the technique, resulting in minimal computational cost and better computational efficiency.

5.2 Formulation of new Steffensen-like families

This section outlines the construction of third and fifth-order iterative algorithms, which serve as the fundamental procedures for the generalized technique presented in the forthcoming section.

5.2.1 Novel third-order family

The proposed approach in this study aims to enhance the convergence order of (5.3) by minimizing the cost associated with the matrix inverse operator and function evaluations. The iterative procedure is

$$\begin{aligned} y^{(n)} &= X^{(n)} - \frac{2}{3}\Delta^{-1}F(X^{(n)}), \\ X^{(n+1)} &= y^{(n)} - (a_1 I + a_2 Q)\Delta^{-1}F(y^{(n)}), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{5.4}$$

where $\Delta = [u^{(n)}, v^{(n)}; F], Q = \Delta^{-1}[y^{(n)}, X^{(n)}; F], u^{(n)} = X^{(n)} + \beta F(X^{(n)}), v^{(n)} = X^{(n)} - \beta F(X^{(n)})$, $\beta \in \mathbb{R}\backslash\{0\}$, a_1, a_2 denote the free parameters, and the identity matrix of order $k \times k$ is denoted by I.

For convergence analysis, we use Taylor's expansion of $F'(X + th)$ at point X, and then integrating, one obtains the following divided difference

$$[X + h, X; F] = \int_0^1 F'(X + th)dt = F'(X) + \frac{1}{2}F''(X)h + \frac{1}{6}F'''(X)h^2 + O(h^3), \tag{5.5}$$

where $h^k = \left(h, h, \overset{k-times}{\cdots}, h\right), h \in \mathbb{R}^k$. Let $\xi^{(n)} = X^{(n)} - X^*$. Furthermore, assume that $\Gamma = [F'(X^*)]^{-1}$ exists, then expanding $F(X^{(n)})$ in the neighborhood of X^*, one obtains

$$F(X^{(n)}) = F'(X^*)\left[\xi^{(n)} + C_2(\xi^{(n)})^2 + C_3(\xi^{(n)})^3 + C_4(\xi^{(n)})^4 + O((\xi^{(n)})^5)\right], \tag{5.6}$$

where $C_i = \frac{1}{i!}\Gamma F^{(i)}(X^*) \in L_i\left(\mathbb{R}^k, \mathbb{R}^k\right), i = 2, 3, \ldots$, and $\left(\xi^{(n)}\right)^i = \left(\xi^{(n)}, \xi^{(n)}, \overset{i-times}{\cdots}, \xi^{(n)}\right), \xi^{(n)} \in \mathbb{R}^k$.

We can now analyze the convergence behavior of the scheme (5.4) by proving the following theorem.

Theorem 5.2.1. *Consider $F : \mathcal{D} \to \mathbb{R}^k$ be a function that is Fréchet differentiable on an open convex set $\mathcal{D} \subseteq \mathbb{R}^k$ having zero X^*. Assume that at $X = X^*$, $F'(X)$ is continuous and non-singular. If the initial approximation $X^{(0)}$ is close enough to the value X^*. Then, the sequence $\{X^{(n)}\}_{n=0}^{\infty}$ obtained from the iterative procedure* (5.4) *converges to X^* with at least third-order convergence $\forall\ \beta \in \mathbb{R}\backslash\{0\}$.*

Proof. Let $\xi^{(n)} = X^{(n)} - X^*$, $\xi_1^{(n)} = u^{(n)} - X^* = \xi^{(n)} + \beta F(X^{(n)})$, and $\xi_2^{(n)} = v^{(n)} - X^* = \xi^{(n)} - \beta F(X^{(n)})$ be the errors at n^{th} iteration. Therefore, we can write

$$\xi_1^{(n)} = (I + \beta F'(X^*))\xi^{(n)} + \beta F'(X^*)(C_2(\xi^{(n)})^2 + C_3(\xi^{(n)})^3 + O((\xi^{(n)})^4)), \tag{5.7}$$

$$\xi_2^{(n)} = (I - \beta F'(X^*))\xi^{(n)} - \beta F'(X^*)(C_2(\xi^{(n)})^2 + C_3(\xi^{(n)})^3 + O((\xi^{(n)})^4)). \tag{5.8}$$

Substituting $X + h = u^{(n)},\ X = v^{(n)}, h = \xi_1^{(n)} - \xi_2^{(n)}$ in Eq. (5.6), one obtains the following operator $\Delta = [u^{(n)}, v^{(n)}; F]$, on simplification

$$\begin{aligned}\Delta = F'(X^*)\Big[I + C_2\left(\xi_1^{(n)} + \xi_2^{(n)}\right) + C_3\left((\xi_1^{(n)})^2 + \xi_1^{(n)}\xi_2^{(n)} + (\xi_2^{(n)})^2\right) + C_4\Big((\xi_1^{(n)})^3 \\ + (\xi_1^{(n)})^2\xi_2^{(n)} + \xi_1^{(n)}(\xi_2^{(n)})^2 + (\xi_2^{(n)})^3\Big) + C_5\Big((\xi_1^{(n)})^4 + (\xi_1^{(n)})^3(\xi_2^{(n)}) + (\xi_1^{(n)})^2(\xi_2^{(n)})^2 \\ + (\xi_2^{(n)})^4\Big) + O((\xi^{(n)})^5)\Big].\end{aligned} \tag{5.9}$$

Furthermore, the inverse of the same can be obtained as

$$\begin{aligned}\Delta^{-1} = \Big[I - C_2(\xi_1^{(n)} + \xi_2^{(n)}) + (C_2^2 - C_3)((\xi_1^{(n)})^2 + (\xi_2^{(n)})^2) + (2C_2^2 - C_3)\xi_1^{(n)}\xi_2^{(n)} \\ -(C_4 - C_2C_3 - C_3C_2 + C_2^3)((\xi_1^{(n)})^3 + (\xi_2^{(n)})^3) - (C_4 - 2C_2C_3 - 2C_3C_2 \\ +3C_2^3)((\xi_1^{(n)})^2\xi_2^{(n)} + \xi_1^{(n)}(\xi_2^{(n)})^2) + O((\xi^{(n)})^4)\Big]\,\Gamma.\end{aligned} \tag{5.10}$$

Now, pre-multiply Eq. (5.10) by (5.6), we get

$$\begin{aligned}\Delta^{-1}F(X^{(n)}) = \xi^{(n)} - C_2(\xi_1^{(n)})^2 - (2C_2^2 - C_3)(\xi^{(n)})^3 - (C_2^2 - C_3)((\xi_1^{(n)})^2(\xi_2^{(n)})^2)\xi^{(n)} \\ - (2C_2^2 - C_3)\xi_1^{(n)}\xi_2^{(n)}\xi^{(n)} + O((\xi^{(n)})^4).\end{aligned} \tag{5.11}$$

On substituting the above, in the first substep of the scheme (5.4) such that $\xi_y^{(n)} = y^{(n)} - X^*$, one obtains

$$\xi_y^{(n)} = \frac{1}{3}\xi^{(n)} + \frac{2}{3}C_2(\xi^{(n)})^2 + \frac{2}{3}(2C_2^2 - C_3)(\xi^{(n)})^3 - \frac{2}{3}(2C_2^2 - C_3)\xi_1^{(n)}\xi_2^{(n)}\xi^{(n)} + O((\xi^{(n)})^4). \tag{5.12}$$

Now, using the Taylor's expansion of $F(y^{(n)})$, one obtain the DD operator as

$$\begin{aligned}[y^{(n)}, X^{(n)}; F] = F'(X^*)\Big[I + \frac{4}{3}C_2\xi^{(n)} + \frac{1}{9}(13C_3 + 6C_2^2)(\xi^{(n)})^2 + \Big(\frac{4}{3}C_2^3 - \frac{2}{3}C_2C_3 \\ + \frac{10}{9}C_3C_2 + \frac{40}{27}C_4\Big)(\xi^{(n)})^3 + \Big(-\frac{2}{3}C_2^3 + \frac{2}{3}C_2C_3 - 4C_4C_2^2 + 4C_4C_3\Big) \\ \times((\xi_1^{(n)})^2 + (\xi_2^{(n)})^2)\xi^{(n)} + \Big(\frac{4}{3}C_2^3 - \frac{2}{3}C_2C_3 - 4C_3C_2^2 + 2C_3^2\Big)\xi_1^{(n)}\xi_2^{(n)}\xi^{(n)} \\ + O((\xi^{(n)})^4)\Big].\end{aligned} \tag{5.13}$$

Substituting the above in second substep of (5.4), one gets

$$\begin{aligned}\xi^{(n+1)} = (1 - a_1 - a_2)\xi_y^{(n)} + \Big(2a_1 + \frac{8}{3}a_2\Big)C_2\xi_y^{(n)} - (a_1 + a_2)(C_2^2 - C_3)((\xi_1^{(n)})^2 \\ + (\xi_2^{(n)})^2)\xi_y^{(n)} - (a_1 + a_2)(2C_2^2 - C_3)\xi_1^{(n)}\xi_2^{(n)}\xi_y^{(n)} - \frac{1}{9}(13C_3 - 6C_2^2)a_2(\xi^{(n)})^2\xi_y^{(n)} \\ - a_2(2C_2^2 - C_3)\xi_1^{(n)}\xi_2^{(n)}\xi_y^{(n)} - (a_1 + a_2)C_2(\xi_y^{(n)})^2 + \Big(2a_1 + \frac{8}{3}a_2\Big)C_2^2(\xi_y^{(n)})^2\xi^{(n)} \\ - (a_1 + a_2)(\xi_y^{(n)})^3 + O((\xi^{(n)})^4).\end{aligned} \tag{5.14}$$

The proposed scheme (5.4) achieves at least the third order of convergence, provided $a_1 = \frac{7}{2}$, and $a_2 =$

$-\frac{5}{2}$. Then, the error equation becomes

$$\xi^{(n+1)} = \frac{1}{54}(68C_2^2 - 9C_3(3\gamma^2+2))(\xi^{(n)})^3 + O((\xi^{(n)})^4), \tag{5.15}$$

where $\gamma = \beta F'(X^*)$ and $\beta \in \mathbb{R}\backslash\{0\}$. □

5.2.2 Novel fifth-order family

Now, we further enhance the convergence order of family (5.4) by adding just one functional evaluation at the third step and obtain the following valuation at the third step. Thus, we have the following iterative family

$$\begin{aligned} y^{(n)} &= X^{(n)} - \frac{2}{3}\Delta^{-1}F(X^{(n)}), \\ z^{(n)} &= y^{(n)} - \frac{1}{2}(7I - 5Q)\Delta^{-1}F(y^{(n)}), \\ X^{(n+1)} &= z^{(n)} - (b_1 I + b_2 Q)\Delta^{-1}F(z^{(n)}), \quad n = 0, 1, 2, \ldots, \end{aligned} \tag{5.16}$$

where $\Delta = [u^{(n)}, v^{(n)}; F], Q = \Delta^{-1}[y^{(n)}, X^{(n)}; F], u^{(n)} = X^{(n)} + \beta F(X^{(n)}), v^{(n)} = X^{(n)} - \beta F(X^{(n)})$, $\beta \in \mathbb{R}\backslash\{0\}$ and b_1, b_2 denote the free parameters.

Theorem 5.2.2. *Assume the same hypothesis as in Theorem* (5.2.1). *Then,* $\forall\ \beta \in \mathbb{R}\backslash\{0\}$ *and* $\gamma = \beta F'(X^*)$, *the iterative procedure* (5.16) *yields at least local convergence order of five for the sequence* $\{X^{(n)}\}_{n=0}^{\infty}$ *if* $b_1 = 4$, *and* $b_2 = -3$. *The error equation is*

$$\begin{aligned} \xi^{(n+1)} = \frac{1}{486}\Big(&3672C_2^4 - 1458\gamma^2 C_2^2 C_3 - 972C_2^2 C_3 - 1224\gamma^2 C_3 C_2^2 - 1020C_3 C_2^2 \\ &+ \left(486\gamma^4 + 729\gamma^2 + 270\right) C_3^2\Big)(\xi^{(n)})^5 + O((\xi^{(n)})^6), \end{aligned} \tag{5.17}$$

Proof. Suppose $z^{(n)} = s^{(n)} + X^*$ in Eq. (5.15) established earlier for scheme (5.4) such that

$$\begin{aligned} s^{(n)} = &\frac{1}{54}\left(68C_2^2 - \left(27\gamma^2 + 18\right)C_3\right)(\xi^{(n)})^3 + \frac{1}{162}\Big(-618C_2^3 + \left(207\gamma^2 + 392\right)C_2 C_3 \\ &+452C_3 C_2 - \left(324\gamma^2 + 126\right)C_4\Big)(\xi^{(n)})^4 + \frac{1}{486}\Big(2700C_2^4 - \left(927\gamma^2 + 2892\right)C_2^2 C_3 \\ &- \left(972\gamma^2 + 4254\right)C_2 C_3 C_2 + \left(2484\gamma^2 + 1778\right)C_2 C_4 + \left(1458\gamma^2 - 1716\right)C_3 C_2^2 \\ &+ \left(162\gamma^4 + 1287\gamma^2 + 2414\right)C_3^2 + 1982C_4 C_2 - \left(243\gamma^4 + 2430\gamma^2 + 612\right)C_5\Big)(\xi^{(n)})^5 \\ &+ O((\xi^{(n)})^6). \end{aligned} \tag{5.18}$$

Now, expand $F(z^{(n)})$ about X^*, one acquires

$$F(z^{(n)}) = F'(X^*)\left(s^{(n)} + C_2(s^{(n)})^2 + O((s^{(n)})^3)\right). \tag{5.19}$$

Pre-multiplying Eq. (5.19) by Eq. (5.10) and simplifying, we get

$$\begin{aligned}\Delta^{-1}F(z^{(n)}) = {} & \frac{1}{54}\left(68C_2^2 - 27\gamma^2C_3 - 18C_3\right)(\xi^{(n)})^3 + \frac{1}{162}\left(-1026C_2^3 + \left(369\gamma^2 + 500\gamma\right.\right.\\ & \left.+452\right)C_2C_3 - \left(324\gamma^2 + 126\right)C_4\Big)(\xi^{(n)})^4 + \frac{1}{486}\left(8856C_2^4 - \left(3141\gamma^2\right.\right.\\ & \left.+5892\right)C_2^2C_3 - \left(972\gamma^2 + 6966\right)C_2C_3C_2 + \left(4428\gamma^2 + 2534\right)C_2C_4\\ & + \left(846\gamma^2 - 3552\right)C_3C_2^2 + \left(405\gamma^4 + 2178\gamma^2\right)C_3^2 + 2900C_3^2 + 1982C_4C_2\\ & - \left(243\gamma^4 + 2430\gamma^2 + 612\right)C_5\Big)(\xi^{(n)})^5 + O((\xi^{(n)})^6).\end{aligned} \tag{5.20}$$

Substituting Eq. (5.20) in the last sub-step of scheme (5.16), the error equation becomes

$$\begin{aligned}\xi^{(n+1)} = {} & \frac{1}{54}\Big(-68b_1C_2^2 + 27\gamma^2b_1C_3 + 18b_1C_3 - 68b_2C_2^2 + 27\gamma^2b_2C_3 + 18b_2C_3 + 68C_2^2\\ & - 27\gamma^2C_3 - 18C_3\Big)(\xi^{(n)})^3 + \frac{1}{162}\Big(1026b_1C_2^3 - 369\gamma^2b_1C_2C_3 - 500b_1C_2C_3\\ & - 452b_1C_3C_2 + 324\gamma^2b_1C_4 + 126b_1C_4 + 1162b_2C_2^3 - 423\gamma^2b_2C_2C_3 - 536b_2C_2C_3\\ & - 452b_2C_3C_2 + 324\gamma^2b_2C_4 + 126b_2C_4 - 618C_2^3 + 207\gamma^2C_2C_3 + 392C_2C_3\\ & + 452C_3C_2 - 324\gamma^2C_4 - 126C_4\Big)(\xi^{(n)})^4 + O((\xi^{(n)})^5).\end{aligned} \tag{5.21}$$

Now, our objective is to get appropriate values of b_1 and b_2 in scheme (5.16) that maximize its convergence order. Therefore, if we put

$$\begin{cases} b_1 = 1 - b_2, \\ b_2 = -3, \end{cases}$$

we obtain our final error equation as

$$\begin{aligned}\xi^{(n+1)} = {} & \frac{1}{486}\Big(3672C_2^4 - 1458\gamma^2C_2^2C_3 - 972C_2^2C_3 - 1224\gamma^2C_3C_2^2 - 1020C_3C_2^2\\ & + 486\gamma^4C_3^2 + 729\gamma^2C_3^2 + 270C_3^2\Big)(\xi^{(n)})^5 + O((\xi^{(n)})^6),\end{aligned} \tag{5.22}$$

where $\beta \in \mathbb{R}$. Hence, the presented family (5.16) gives at least fifth-order convergence. □

5.3 Generalized multi-point Steffensen-like family

Here, we propose the simple generalized version of procedure (5.16) in $(q+1)$-steps, as given by

$$\begin{aligned}
w_0^{(n)} &= X^{(n)} - \frac{2}{3}\Delta^{-1}F(X^{(n)}),\\
w_1^{(n)} &= w_0^{(n)} - \frac{1}{2}(7I-5Q)\Delta^{-1}F(w_0^{(n)}),\\
w_2^{(n)} &= w_1^{(n)} - (4I-3Q)\Delta^{-1}F(w_1^{(n)}),\\
w_3^{(n)} &= w_2^{(n)} - (4I-3Q)\Delta^{-1}F(w_2^{(n)}),\\
&\vdots \qquad \vdots \qquad \vdots \qquad \vdots\\
w_q^{(n+1)} &= w_{q-1}^{(n)} - (4I-3Q)\Delta^{-1}F(w_{q-1}^{(n)}), \quad n=0,1,2,\ldots,
\end{aligned} \tag{5.23}$$

where $\Delta = [u^{(n)}, v^{(n)}; F], Q = \Delta^{-1}[y^{(n)}, X^{(n)}; F], u^{(n)} = X^{(n)} + \beta F(X^{(n)}), v^{(n)} = X^{(n)} - \beta F(X^{(n)})$, $\beta \in \mathbb{R}\backslash\{0\}, w_0^{(n)} = y^{(n)}$ and $w_1^{(n)} = z^{(n)}$.

Theorem 5.3.1. *Assume the same hypothesis as in Theorem* (5.2.2)*. Then,* $\forall\ \beta \in \mathbb{R}\backslash\{0\}$*, the following error equation of proposed technique* (5.23) *yields at least local convergence order of* $2q+1,\ q \geq 3$ *for the sequence* $\{X^{(n)}\}_{n=0}^{\infty}$

$$\begin{aligned}
\xi_q^{(n+1)} = &\frac{1}{54(3)^{q-1}}(18C_2^2 - (5+6\gamma^2)C_3)^{q-1}(68C_2^2 - 9C_3(3\gamma^2+2))(\xi^{(n)})^{2q+1}\\
&+ O((\xi^{(n)})^{2q+2}),
\end{aligned} \tag{5.24}$$

where $\gamma = \beta F'(X^*)$*, and* $\xi_q^{(n+1)}$ *is the error at the* $(q+1)^{th}$ *step.*

Proof. The proof of convergence for the iterative scheme (5.16) follows a similar approach as that used for the proof of Theorem (5.2.1). It is hence omitted. □

Different iterative families can be generated from the proposed family (5.23) by assigning various values to q. However, our focus will be on the numerical and stability behavior of the family denoted as $LM_{2q+1,i}$, where $2q+1$ indicates the order of convergence and i refers to the index associated with different parametric values of β. We will specifically examine cases for $q = 1, 2$, and 3 only. The proposed generalized scheme requires only one matrix inverse per step, used only once throughout the iteration. This feature of the proposed scheme is referred to as a 'method with frozen operator'.

5.4 Computational complexity

This section studies the computational cost of an iterative solver. Here, we have considered different set of values to parameter β, specifically, we take $\beta = 0.01, 0.9,$ and 1.5 in the ρ-order proposed methods $\mathrm{LM}_{\rho,1}$, $\mathrm{LM}_{\rho,2}$, and $\mathrm{LM}_{\rho,3}$, where $\rho = 3, 5,$ and 7. Further, the computational efficiency of the proposed methods is compared with several existing schemes, i.e., Zheng et al. [382] method of fourth-order (denoted by $\mathrm{ZM}_{4,1}$), Sharma et al. [301] method of sixth-order (denoted by $\mathrm{SA}_{6,1}$), Sharma et al. [302] method of seventh-order (denoted by $\mathrm{SA}_{7,1}$), and Wang et al. [361] method of seventh-order (denoted by $\mathrm{WZ}_{7,1}$).

We denote the computing cost and efficiency index of the solver $M_{\rho,i}$ ($\rho = 3, 4, 5, 6, 7$ and $i = 1, 2, 3$) as $\mathcal{C}_{M_{\rho,i}}$ and $E_{M_{\rho,i}}$, respectively. Using the above information, we obtain the computing cost and efficiencies as displayed in Table 5.1.

Table 5.1: Computational costs and efficiencies of different methods

$M_{\rho,i}$	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$
$\mathrm{LM}_{3,1}$	$\frac{k}{6}\left(15l_q(1+k)+2k^2+24k\mu+21k+1\right)$,	$(3)^{1/\mathcal{C}_{\mathrm{LM}_{3,1}}}$
$\mathrm{ZM}_{4,1}$	$\frac{k}{2}\left(l_q(3+9k)+2k^2+6(-1+2k)\mu+5k-5\right)$,	$(4)^{1/\mathcal{C}_{\mathrm{ZM}_{4,1}}}$
$\mathrm{LM}_{5,1}$	$\frac{k}{6}\left(3l_q(9+5k)+2k^2+6(1+4k)\mu+39k+1\right)$,	$(5)^{1/\mathcal{C}_{\mathrm{LM}_{5,1}}}$
$\mathrm{SA}_{6,1}$	$\frac{k}{6}\left(l_q(27+15k)+2k^2+6(1+4k)\mu+39k-5\right)$,	$(6)^{1/\mathcal{C}_{\mathrm{SA}_{6,1}}}$
$\mathrm{SA}_{7,1}$	$\frac{k}{3}\left(l_q(12+18k)+2k^2+6(-3+5k)\mu+18k-11\right)$,	$(7)^{1/\mathcal{C}_{\mathrm{SA}_{7,1}}}$
$\mathrm{WZ}_{7,1}$	$\frac{k}{6}\left(l_q(3+21k)+2k^2+6(-1+6k)\mu+51k-5\right)$,	$(7)^{1/\mathcal{C}_{\mathrm{WZ}_{7,1}}}$
$\mathrm{LM}_{7,1}$	$\frac{k}{6}\left(3l_q(13+5k)+2k^2+6(2+4k)\mu+57k+1\right)$,	$(7)^{1/\mathcal{C}_{\mathrm{LM}_{7,1}}}$

Now, the comparison between the efficiencies of different methods is examined in the following subsection.

5.4.1 Efficiency comparisons

For the comparison of computational efficiencies (CE's), we have considered the ratio as follows

$$R_{\hat{M}_A}^{\hat{M}_B} = \frac{\log E_{\hat{M}_A}}{\log E_{\hat{M}_B}} = \frac{\mathcal{C}_{\hat{M}_B}\log(\rho_A)}{\mathcal{C}_{\hat{M}_A}\log(\rho_B)}, \tag{5.25}$$

where ρ_A and ρ_B denote the convergence order of the solvers $\hat{M}_A$ and $\hat{M}_B$, respectively. More appropriately, if $R^{\hat{M}_B}_{\hat{M}_A} > 1$, then we can say iterative solver $\hat{M}_A$ is more efficient than $\hat{M}_B$.

The boundary lines between the CE's are when $R^{\hat{M}_B}_{\hat{M}_A} = 1$, and the respective expression μ is a function of k and l ($k \geq 2$, $\mu > 0$ and $l_q \geq 1$).

$LM_{5,1}$ and $ZM_{4,1}$ comparison: The boundary lines can be obtained if $R^{\mathrm{ZM}_{4,1}}_{\mathrm{LM}_{5,1}} = 1$, and correspondingly μ can be written as

$$\mu = \frac{((15 - 9l_q - 15k - 27l_qk - 6k^2)\eta_1 + (1 + 27l_q + 39k + 15l_qk + 2k^2)\eta_2)}{6(-3\eta_1 + 6k\eta_1 - \eta_2 - 4k\eta_2)}, \tag{5.26}$$

where $\eta_1 = \log(5)$ and $\eta_2 = \log(4)$. In the case of $k = 2.4896$, μ has a vertical asymptote. This represents that the iterative solver $\mathrm{LM}_{5,1}$ is more efficient than $\mathrm{ZM}_{4,1}$ $\forall$ $k \geq 3$, and $l_q \geq 1$. The graphical manifestation has also been shown in Figure 5.1.

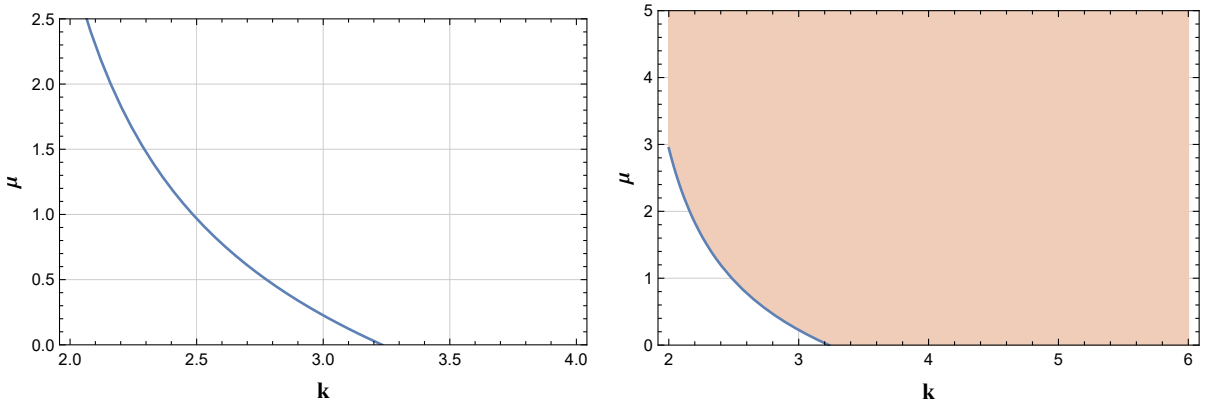


Figure 5.1: **a** Boundary between $\mathrm{LM}_{5,1}$ and $\mathrm{ZM}_{4,1}$. **b** Region shaded if $R^{\mathrm{ZM}_{4,1}}_{\mathrm{LM}_{5,1}} > 1$.

$LM_{5,1}$ and $LM_{3,1}$ comparison: The boundary lines can be obtained in similar manner, as $R^{\mathrm{LM}_{3,1}}_{\mathrm{LM}_{5,1}} = 1$, and

$$\mu = \frac{((-1 - 15l_q - 21k - 15l_qk - 2k^2)\eta_1 + (1 + 27l_q + 39k + 15l_qk + 2k^2)\eta_2)}{6(4k\eta_1 - \eta_2 - 4k\eta_2)}, \tag{5.27}$$

where $\eta_1 = \log(5)$ and $\eta_2 = \log(3)$. In the case of $k = 0.97717$, μ has a vertical asymptote. This represents that the iterative solver $\mathrm{LM}_{5,1}$ is more efficient than $\mathrm{LM}_{3,1}$ $\forall$ $k \geq 2$, and $l_q \geq 1$. The graphical manifestation has also been shown in Figure 5.2.

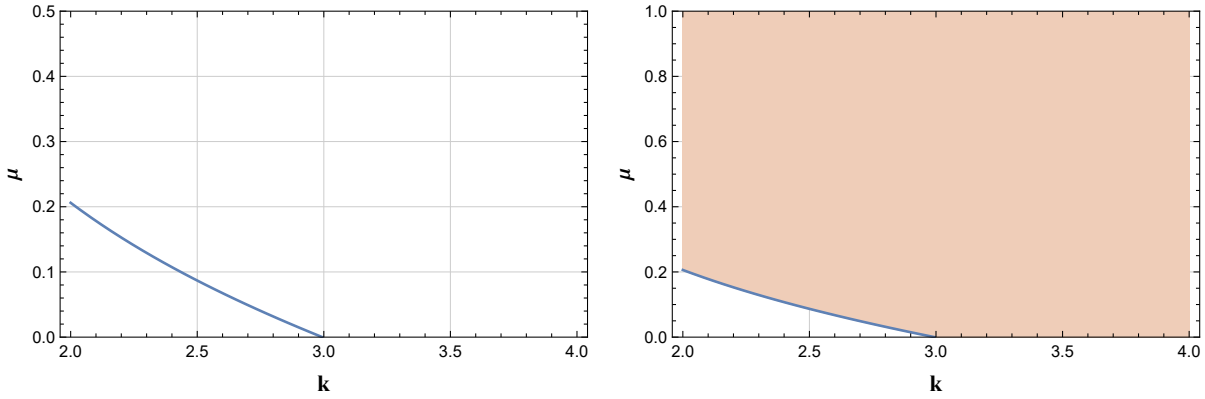


Figure 5.2: **a** Boundary between $\mathrm{LM}_{5,1}$ and $\mathrm{LM}_{3,1}$. **b** Region shaded if $R^{\mathrm{LM}_{3,1}}_{\mathrm{LM}_{5,1}} > 1$.

$\mathrm{LM}_{7,1}$ and $\mathrm{ZM}_{4,1}$ comparison: Here, we obtain the boundary lines if $R^{\mathrm{ZM}_{4,1}}_{\mathrm{LM}_{7,1}} = 1$, and

$$\mu = \frac{((15 - 9l_q - 15k - 27l_q k - 6k^2)\eta_1 + (1 + 27l_q + 39k + 15l_q k + 2k^2)\eta_2)}{6(-3\eta_1 + 6k\eta_1 - \eta_2 - 4k\eta_2)}, \tag{5.28}$$

where $\eta_1 = \log(7)$ and $\eta_2 = \log(4)$. In the case of $k = 2.75067$, μ has a vertical asymptote. This represents that the iterative solver $\mathrm{LM}_{7,1}$ is more efficient than $\mathrm{ZM}_{4,1} \forall\ k \geq 3$, and $l_q \geq 1$. The graphical manifestation has also shown in Figure 5.3.

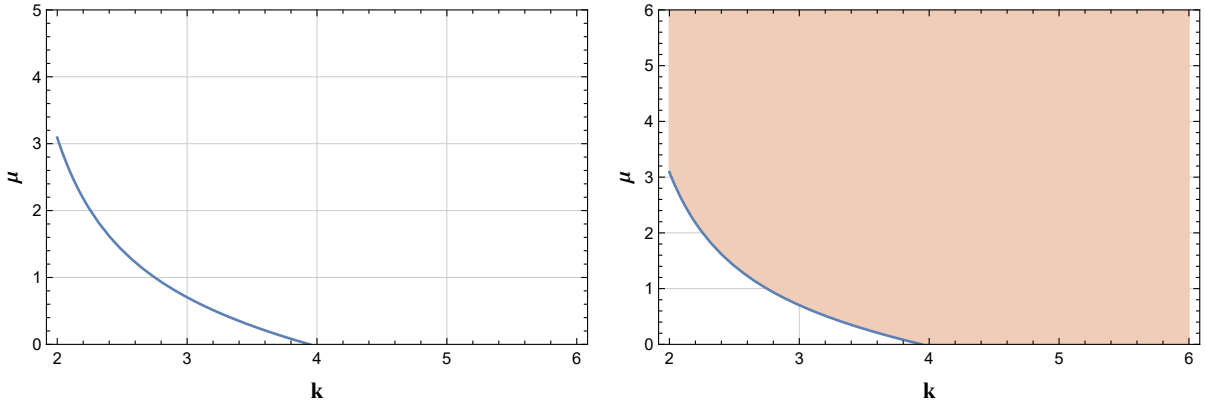


Figure 5.3: **a** Boundary between $\mathrm{LM}_{7,1}$ and $\mathrm{ZM}_{4,1}$. **b** Region shaded if $R^{\mathrm{ZM}_{4,1}}_{\mathrm{LM}_{7,1}} > 1$.

$\mathrm{LM}_{7,1}$ and $\mathrm{SA}_{6,1}$ comparison: The boundary condition is $R^{\mathrm{SA}_{6,1}}_{\mathrm{LM}_{7,1}} = 1$, and μ is given by

$$\mu = \frac{((5 - 27l_q - 39k - 15l_q k - 2k^2)\eta_1 + (1 + 39l_q + 57k + 15l_q k + 2k^2)\eta_2)}{6(\eta_1 + 4k\eta_1 - 2\eta_2 - 4k\eta_2)}, \tag{5.29}$$

where $\eta_1 = \log(7)$ and $\eta_2 = \log(6)$. In the case of $k = 67.47068$, μ has a vertical asymptote. This represents that the iterative solver $\mathrm{LM}_{7,1}$ is more efficient than $\mathrm{SA}_{6,1}\ \forall\ k \geq 68$, and $l_q \geq 1$. The

graphical manifestation has also been done in Figure 5.4.

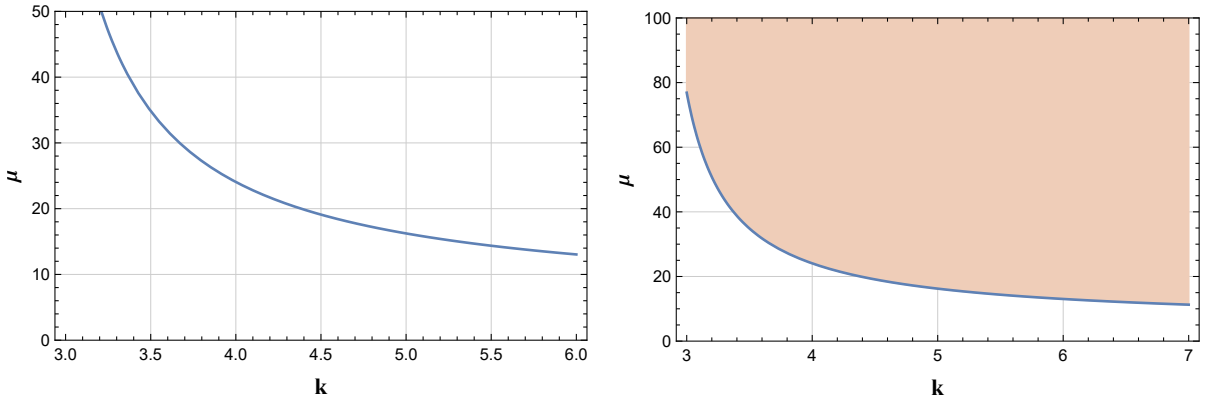
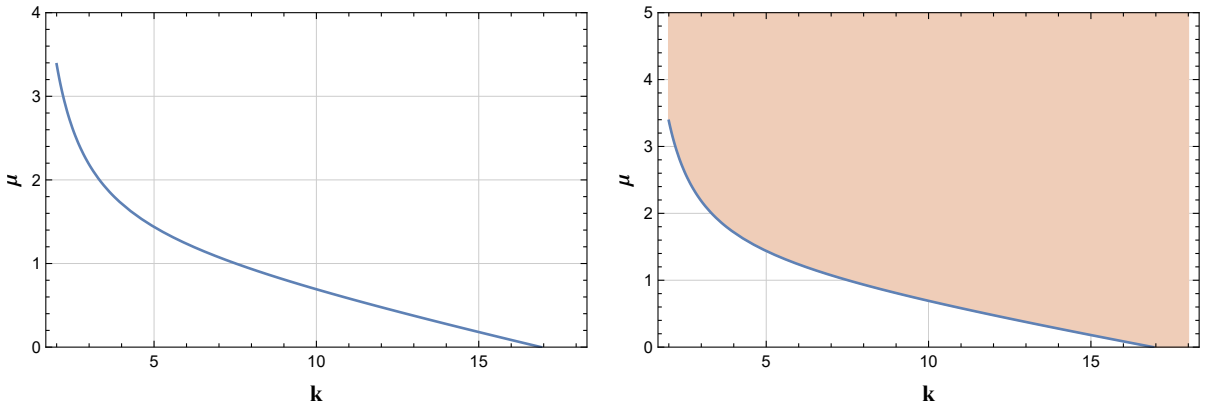


Figure 5.4: **a** Boundary between $\mathrm{LM}_{7,1}$ and $\mathrm{SA}_{6,1}$. **b** Region shaded if $R^{\mathrm{SA}_{6,1}}_{\mathrm{LM}_{7,1}} > 1$.

$\mathrm{LM}_{7,1}$ and $\mathrm{LM}_{5,1}$ comparison: Here,

$$\mu = \frac{((-1 - 27l_q - 39k - 15l_qk - 2k^2)\eta_1 + (1 + 39l_q + 57k + 15l_qk + 2k^2)\eta_2)}{6(\eta_1 + 4k\eta_1 - 2\eta_2 - 4k\eta_2)}, \tag{5.30}$$

where $\eta_1 = \log(7)$ and $\eta_2 = \log(5)$. In the case of $k = 7.51552$, μ has a vertical asymptote. This represents that the iterative solver $\mathrm{LM}_{7,1}$ is more efficient than $\mathrm{LM}_{5,1}$ $\forall$ $k \geq 8$, and $l_q \geq 1$. The graphical manifestation has also been done in Figure 5.5.



(a) (b)

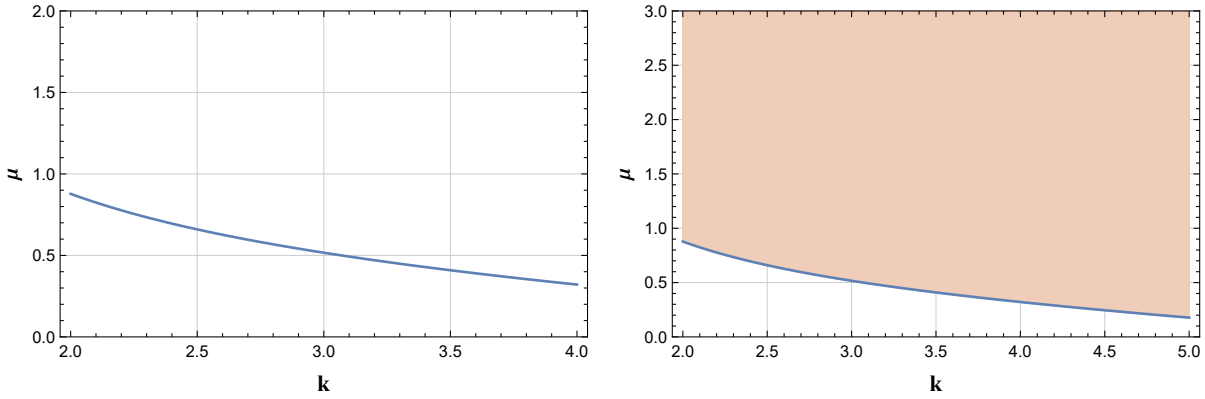
Figure 5.5: **a** Boundary between $\mathrm{LM}_{7,1}$ and $\mathrm{LM}_{5,1}$. **b** Region shaded if $R^{\mathrm{LM}_{5,1}}_{\mathrm{LM}_{7,1}} > 1$.

$\mathrm{LM}_{7,1}$ and $\mathrm{LM}_{3,1}$ comparison: The boundary lines can be obtained in similar manner as $R^{\mathrm{LM}_{3,1}}_{\mathrm{LM}_{7,1}} =$

1, and

$$\mu = \frac{((-1 - 15l_q - 21k - 15l_qk - 2k^2)\eta_1 + (1 + 39l_q + 57k + 15l_qk + 2k^2)\eta_2)}{12(2k\eta_1 - \eta_2 - 2k\eta_2)}, \tag{5.31}$$

where $\eta_1 = \log(7)$ and $\eta_2 = \log(3)$. In the case of $k = 1.81061$, μ has a vertical asymptote. This represents that the iterative solver $\mathrm{LM}_{7,1}$ is more efficient than $\mathrm{LM}_{3,1}$ $\forall$ $k \geq 2$, and $l_q \geq 1$. The graphical manifestation has also been done in Figure 5.6.



(a) (b)

Figure 5.6: **a** Boundary between $\mathrm{LM}_{7,1}$ and $\mathrm{LM}_{3,1}$. **b** Region shaded if $R^{\mathrm{LM}_{3,1}}_{\mathrm{LM}_{7,1}} > 1$.

$\mathrm{LM}_{7,1}$ and $\mathrm{SA}_{7,1}$ comparison: The ratio (5.25) can be written as

$$R^{\mathrm{SA}_{7,1}}_{\mathrm{LM}_{7,1}} = \frac{2\left(-11 + 2k^2 + 6l_q(2 + 3k) - 18\mu + 6k(3 + 5\mu)\right)}{\left(-1 + 2k^2 + 3l_q(13 + 5k) + 12\mu + 3k(19 + 8\mu)\right)}, \tag{5.32}$$

and the corresponding boundary equation is $R^{\mathrm{SA}_{7,1}}_{\mathrm{LM}_{7,1}} = 1$. Moreover, we obtain the following

$$\mu = \frac{23 + 15l_q + 21k - 21l_qk - 2k^2}{12(3k - 4)}. \tag{5.33}$$

Here, $R^{\mathrm{SA}_{7,1}}_{\mathrm{LM}_{7,1}} > 1,\ \forall\ l_q \geq 1,\ \mu > 0$, and $k \geq 3$.

$\mathrm{LM}_{7,1}$ and $\mathrm{WZ}_{7,1}$ comparison: On a similar basis, the ratio (5.25) becomes

$$R^{\mathrm{WZ}_{7,1}}_{\mathrm{LM}_{7,1}} = \frac{\left(-5 + 2k^2 + 3l_q(11 + 7k) - 6\mu + k(51 + 36\mu)\right)}{\left(1 + 2k^2 + 3l_q(13 + 5k) + 12\mu + 3k(19 + 8\mu)\right)}, \tag{5.34}$$

and for boundary lines $R^{\mathrm{WZ}_{7,1}}_{\mathrm{LM}_{7,1}} = 1$, it takes the form

$$\mu = \frac{1 + l_q + k - l_q k}{2k - 3}. \tag{5.35}$$

Here, $R^{\mathrm{WZ}_{7,1}}_{\mathrm{LM}_{7,1}} > 1, \forall\ l_q \geq 1,\ \mu > 0$, and $k \geq 3$.

The efficiency plots of different iterative methods, including $\mathrm{SA}_{7,1}$, $\mathrm{WZ}_{7,1}$, and $\mathrm{LM}_{7,1}$, are presented in Figure 5.7. These graphs confirm the findings obtained in the two previous cases for specific values of $\mu = 1$ and $l_q = 1$. It is clear from the plot that the efficiency of $\mathrm{LM}_{7,1}$ improves compared to $\mathrm{SA}_{7,1}$ and $\mathrm{WZ}_{7,1}$ as k increases. Although the efficiencies of the compared methods appear similar for $k \geq 31$, they are actually quite different. Therefore, a subplot is included in Figure 5.7 to illustrate the more efficient nature of the proposed scheme $\mathrm{LM}_{7,1}$ compared to the other methods. The following theorem provides a summary of the above findings:

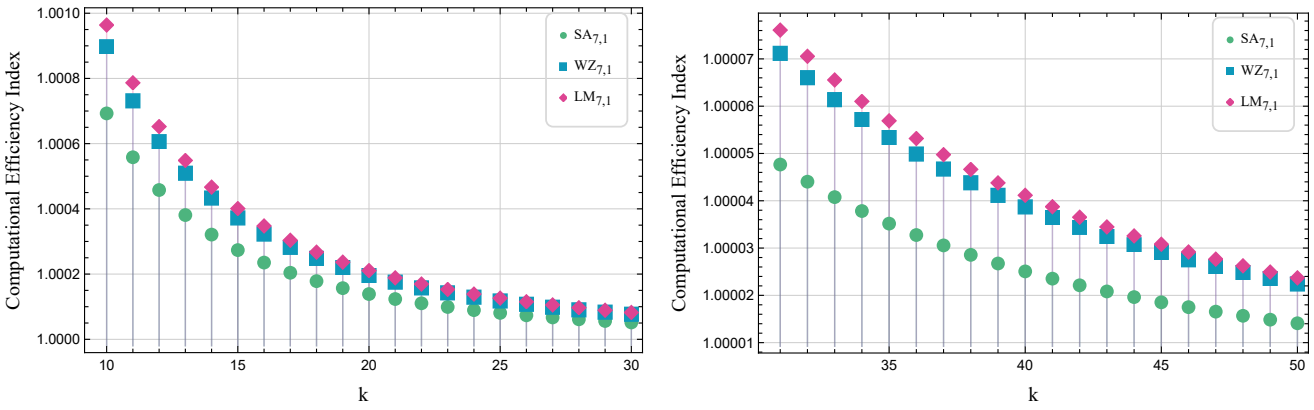


Figure 5.7: Comparison of efficiencies between $\mathrm{SA}_{7,1}$, $\mathrm{WZ}_{7,1}$, and $\mathrm{LM}_{7,1}$ for $k = 10, \cdots, 50$.

Theorem 5.4.1. *For $l_q \geq 1$ and $\mu > 0$, we obtain*

(a). $E_{LM_{5,1}} > E_{ZM_{4,1}}\ \forall\ k \geq 3$,

(b). $E_{LM_{5,1}} > E_{LM_{3,1}}\ \forall\ k \geq 2$,

(c). $E_{LM_{7,1}} > E_{ZM_{4,1}}\ \forall\ k \geq 3$,

(d). $E_{LM_{7,1}} > E_{SA_{6,1}}\ \forall\ k \geq 68$,

(e). $E_{LM_{7,1}} > E_{LM_{5,1}}\ \forall\ k \geq 8$,

(f). $E_{LM_{7,1}} > E_{LM_{3,1}}\ \forall\ k \geq 2$,

(g). $E_{LM_{7,1}} > E_{SA_{7,1}}\ \forall\ k \geq 3$,

(h). $E_{LM_{7,1}} > E_{WZ_{7,1}}\ \forall\ k \geq 3$.

5.5 Numerical tests

This section presents various numerical problems to demonstrate the theoretical results, such as the convergence order and efficiency of the proposed iterative solvers. The new solvers with specific values of the parameter β are compared to existing solvers, namely $ZM_{4,1}$ (for $\beta = 0.01$), $SA_{6,1}$ (for $\beta = 0.01$), $SA_{7,1}$ (for $\beta = 1.5$), $SA_{7,2}$ (for $\beta = 0.9$), $SA_{7,3}$ (for $\beta = 0.7$), and $WZ_{7,1}$. In each problem, the iteration number is computed with the stopping criterion of $\left\|X^{(n+1)} - X^{(n)}\right\|_\infty + \left\|F\left(X^{(n+1)}\right)\right\|_\infty < 10^{-300}$. Additionally, we compute the computational order of convergence (ρ_{COC}) [363] to verify the theoretical convergence order. We validate the theoretical results through several numerical problems, and the corresponding outcomes are presented in Table 5.3–5.9. In these tables, ρ denotes the convergence order of an iterative solver, and $\xi_{n+1} = \left\|X^{(n+1)} - X^{(n)}\right\|$ denotes the residual error norm. The values $\mathcal{C}_{M_{\rho,i}}$ and $E_{M_{\rho,i}}$ represent the computational cost and efficiency of an iterative technique $M_{\rho,i}$ in terms of the products and quotients required, which are computed using the expressions provided in Table 5.1. The CPU time (e-Time) is also reported in the last column of each table.

In order to estimate the values of l_q and μ in each problem, we expressed the cost of every elementary function in product units in Table 5.2, and accordingly, the value of l_q is approximately 2.6., which directly depends on the software, computer system, and the arithmetic used [127].

Table 5.2: Estimation of CPU time and computational cost for elementary functions, where $\hat{x} = \sqrt{3} - 1$ and $\hat{y} = \sqrt{5}$

Functions	$\hat{x} * \hat{y}$	$\hat{x}/\hat{y}$	$\sqrt{\hat{x}}$	$e^{\hat{x}}$	$\ln(\hat{x})$	$\sin(\hat{x})$	$\cos(\hat{x})$	$\cos^{-1}(\hat{x})$	$\tan^{-1}(\hat{x})$
CPU time	0.0390625	0.103125	0.034375	1.03125	0.90625	1.39063	1.39063	1.1875	1.04688
Cost	1	2.64	0.88	26.4	23.2	35.6	35.6	30.4	26.8

In the Table 5.3–5.9, the expression $\mathcal{M}(\pm e)$ represents the form $\mathcal{M} \times 10^{(\pm e)}$. For numerical experimentation the following examples have been considered.

5.5.1 Academic problems

Example 5.5.1. *Consider the nonlinear system of two equations:*

$$X_1 + e^{X_2} - \cos(X_2) = 0,$$
$$3X_1 - X_2 - \sin(X_1) = 0.$$

The initial guess here is $X^{(0)} = (0.1, 0.1)^T$ *in the iterative solvers to get the solution* $X^* = (0, 0)^T$. *Also, the parametric values,* $(k, \mu) = (2, 49.3)$ *are used to evaluate the computing costs and efficiency. The final outcomes are shown in the Table 5.3.*

Table 5.3: Comparison results on Example 5.5.1

Methods	n	ξ_2	ξ_3	ξ_4	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$\text{LM}_{3,1}$	6	5.61(−11)	7.43(−32)	1.73(−94)	844.8	1.00130128662225	3	3.0000	0.436
$\text{LM}_{3,2}$	6	6.99(−11)	1.48(−31)	1.50(−93)	844.8	1.00130128662225	3	3.0000	1.484
$\text{LM}_{3,3}$	6	5.17(−11)	5.85(−32)	1.02(−94)	844.8	1.00130128662225	3	2.9961	0.828
$\text{ZM}_{4,1}$	4	9.27(−39)	1.49(−154)	9.30(−618)	955	1.00145267123820	4	4.0002	1.048
$\text{LM}_{5,1}$	4	7.67(−26)	2.24(−126)	4.82(−629)	965.8	1.00166781907776	5	5.0000	0.469
$\text{LM}_{5,2}$	4	8.81(−26)	5.18(−126)	3.63(−627)	965.8	1.00166781907776	5	5.0000	0.906
$\text{LM}_{5,3}$	4	2.41(−26)	7.35(−129)	2.39(−641)	965.8	1.00166781907776	5	5.0000	0.889
$\text{SA}_{6,1}$	4	1.89(−34)	9.36(−203)	1.41(−1212)	963.8	1.00186078646357	6	6.0000	0.437
$\text{SA}_{7,1}$	4	6.33(−43)	1.40(−296)	3.63(−2072)	1485.6	1.00131070619877	7	6.9999	2.952
$\text{SA}_{7,2}$	3	1.01(−45)	3.88(−316)	4.79(−2209)	1485.6	1.00131070619877	7	7.0000	3.000
$\text{SA}_{7,3}$	3	7.42(−47)	4.55(−324)	1.48(−2264)	1485.6	1.00131070619877	7	7.0000	2.142
$\text{WZ}_{7,1}$	3	2.41(−46)	1.34(−319)	2.28(−2232)	1184.6	1.00164402268390	7	7.0000	0.610
$\text{LM}_{7,1}$	3	1.60(−47)	4.50(−328)	6.35(−2292)	1086.8	1.00179209906202	7	7.0000	0.516
$\text{LM}_{7,2}$	3	1.81(−47)	1.32(−327)	1.48(−2288)	1086.8	1.00179209906202	7	7.0000	0.859
$\text{LM}_{7,3}$	3	3.57(−48)	1.78(−332)	1.48(−2322)	1086.8	1.00179209906202	7	6.9999	0.906

Example 5.5.2. *Here, we take the integral equation* (4.76) *of mixed Hammerstein (for reference, [252]) such that after discretization, we obtain the following nonlinear system:*

$$5X_s - \sum_{j=1}^{8} a_{sj} X_j^3 - 5 = 0, \ \ s = 1, 2, \ldots, 8,$$

where $a_{sj} = \begin{cases} l_j \Delta_j (1 - l_k), & j \leq s, \\ l_s \Delta_j (1 - l_j), & s < j. \end{cases}$ *For solving this system, we take an initial guess* $X^{(0)} =$

$(0.9, 0.9, \ldots, 0.9)^T$, *and the correspondingly, we obtain the following approximate solution*

$$X^* = (1.002096..., 1.009900..., 1.019727..., 1.026435..., 1.026435..., 1.019727...,$$
$$1.009900..., 1.002096...)^T.$$

This problem considers parametric values of $(k, \mu) = (8, 11)$*, which we use to evaluate the computing cost and efficiency. The computational outcomes are shown in the Table 5.4.*

Table 5.4: Numerical comparison results on Example 5.5.2

Methods	n	ξ_2	ξ_3	ξ_4	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$LM_{3,1}$	5	3.93(−16)	2.10(−49)	3.24(−149)	3680	1.00029858051430	3	2.9999	12.985
$LM_{3,2}$	5	2.91(−11)	1.37(−33)	1.47(−100)	3680	1.00029858051430	3	2.9997	13.062
$LM_{3,3}$	6	5.56(−5)	2.53(−14)	2.44(−42)	3680	1.00029858051430	3	2.9986	25.938
$ZM_{4,1}$	4	7.27(−29)	6.37(−117)	3.78(−469)	5392	1.00025713512351	4	4.0000	15.406
$LM_{5,1}$	4	1.04(−41)	7.20(−210)	1.16(−1050)	4001.6	1.00040227949137	5	5.0000	17.375
$LM_{5,2}$	4	1.40(−27)	6.97(−137)	2.17(−683)	4001.6	1.00040227949137	5	4.9999	19.438
$LM_{5,3}$	4	19.30(−22)	3.44(−110)	4.46(−549)	4001.6	1.00040227949137	5	4.9999	19.297
$SA_{6,1}$	3	2.58(−60)	1.57(−363)	8.11(−2183)	3993.6	1.00044875838158	6	6.0000	20.063
$SA_{7,1}$	–	Diverges	–	–	–	–	-	–	–
$SA_{7,2}$	–	Diverges	–	–	–	–	-	–	–
$SA_{7,3}$	3	1.79(−71)	9.84(−501)	1.50(−3505)	8289.6	1.00023476869324	7	7.0000	25.109
$WZ_{7,1}$	–	Diverges	–	–	–	–	-	–	–
$LM_{7,1}$	3	1.96(−79)	1.11(−557)	2.17(−3905)	4323.2	1.00045021006439	7	7.0000	17.594
$LM_{7,2}$	3	1.69(−51)	1.18(−358)	9.55(−2509)	4323.2	1.00045021006439	7	7.0000	25.609
$LM_{7,3}$	4	3.87(−41)	7.25(−285)	6.01(−1991)	4323.2	1.00045021006439	7	6.9999	25.250

Example 5.5.3. *Next, the nonlinear system of 25 equations has been considered:*

$$X_i^2 X_{i+1} - 1 = 0, \ 1 \leq i \leq 24,$$
$$X_i^2 X_1 - 1 = 0, \qquad i = 25.$$

The initial estimate here is $X^{(0)} = (15/10, 15/10, ..., 15/10)^T$ *with resulting approximate corrected value of* $X^* = (1, 1, ..., 1)^T$*. The concrete parametric values in this example are* $(k, \mu) = (25, 2)$*, which we use to calculate the computing cost and efficiency. These computational outcomes are*

shown in the Table 5.5.

Table 5.5: Numerical comparison results on Example 5.5.3

Methods	n	ξ_2	ξ_3	ξ_4	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$LM_{3,1}$	7	9.56(−4)	4.02(−11)	2.97(−33)	16625	1.00006608412564	3	3.0000	6.874
$LM_{3,2}$	6	3.49(−6)	1.14(−16)	3.95(−51)	16625	1.00006608412564	3	3.0000	7.327
$LM_{3,3}$	7	1.53(−2)	3.20(−7)	2.91(−21)	16625	1.00006608412564	3	3.0008	5.562
$ZM_{4,1}$	5	5.09(−9)	1.73(−36)	2.30(−146)	31885	1.00004347889284	4	4.0000	5.11
$LM_{5,1}$	5	5.76(−8)	6.37(−39)	1.05(−193)	18680	1.00008616205823	5	5.0000	9.922
$LM_{5,2}$	5	1.08(−9)	9.03(−50)	3.75(−250)	18680	1.00008616205823	5	5.0000	9.875
$LM_{5,3}$	5	7.89(−5)	8.78(−23)	1.50(−112)	18680	1.00008616205823	5	5.0000	12.656
$SA_{6,1}$	4	6.33(−10)	5.45(−58)	2.21(−346)	18655	1.00009605175656	6	6.0000	9.686
$SA_{7,1}$	5	5.21(−4)	5.68(−24)	1.04(−163)	36285	1.00005362993877	7	7.0000	10.282
$SA_{7,2}$	5	1.29(−5)	5.49(−36)	1.42(−248)	36285	1.00005362993877	7	7.0000	12.641
$SA_{7,3}$	5	1.40(−6)	4.71(−43)	2.24(−298)	36285	1.00005362993877	7	7.0000	11.297
$WZ_{7,1}$	4	3.22(−8)	4.83(−56)	8.31(−391)	23995	1.00008109977305	7	7.0000	11.187
$LM_{7,1}$	4	6.52(−14)	1.09(−95)	3.95(−668)	20735	1.00009385104705	7	7.0000	12.594
$LM_{7,2}$	4	3.43(−16)	8.12(−115)	3.40(−805)	20735	1.00009385104705	7	7.0000	14.594
$LM_{7,3}$	4	3.61(−8)	7.34(−55)	1.06(−381)	20735	1.00009385104705	7	7.0000	14.204

Example 5.5.4. *Now, consider the Frank-Kamenetskii problem [128] described by the differential equation*

$$\begin{cases} \tau X'' + X' + \tau e^X = 0, \\ X'(0) = X(1) = 0. \end{cases} \tag{5.36}$$

Here, the following finite difference discretization is used

$$X'' = \frac{X_{j-1} - 2X_j + X_{j+1}}{h^2},\ X' = \frac{X_{j+1} - X_j}{h},$$

and we can convert the BVP (5.36) *into systems of* 50 *nonlinear equations in* 50 *unknown variables with step size* $h = \frac{1}{51}$*, given by*

$$jh^3 e^{X_j} + jh(X_{j-1} - 2X_j + X_{j+1}) + h(X_{j+1} - X_j) = 0,\ j = 1,\ 2,\ \ldots,\ 50.$$

The initial estimate used here is $X^{(0)} = (1/10, 1/10, ..., 1/10)^T$ *with resulting approximate solution*

of

$$X^* = (0.474290765\ldots, 0.473878907\ldots, 0.473055394\ldots, 0.471820648\ldots, 0.470175306\ldots,$$
$$0.468120222\ldots, 0.465656465\ldots, 0.462785314\ldots, 0.459508258\ldots, 0.455826994\ldots,$$
$$0.451743424\ldots, 0.447259647\ldots, 0.442377962\ldots, 0.437100858\ldots, 0.431431015\ldots,$$
$$0.425371293\ldots, 0.418924732\ldots, 0.412094545\ldots, 0.404884113\ldots, 0.397296979\ldots,$$
$$0.389336840\ldots, 0.381007544\ldots, 0.372313084\ldots, 0.363257589\ldots, 0.353845317\ldots,$$
$$0.344080652\ldots, 0.333968096\ldots, 0.323512259\ldots, 0.312717856\ldots, 0.301589700\ldots,$$
$$0.290132692\ldots, 0.278351818\ldots, 0.266252139\ldots, 0.253838787\ldots, 0.241116956\ldots,$$
$$0.228091898\ldots, 0.214768913\ldots, 0.201153346\ldots, 0.187250579\ldots, 0.173066025\ldots,$$
$$0.158605121\ldots, 0.143873326\ldots, 0.128876110\ldots, 0.113618952\ldots, 0.098107334\ldots,$$
$$0.082346735\ldots, 0.066342626\ldots, 0.050100467\ldots, 0.033625700\ldots, 0.016923746\ldots)^T.$$

The concrete parametric values in this case are $(k, \mu) = (50, 36.4)$ *which we use to evaluate the computing cost and efficiency. The computational outcomes are presented in the Table 5.6.*

For illustration, we have graphed efficiency log plots for the above four examples using histograms. Figure 5.8 reveals that the proposed methods have reasonably better efficiencies compared to existing ones.

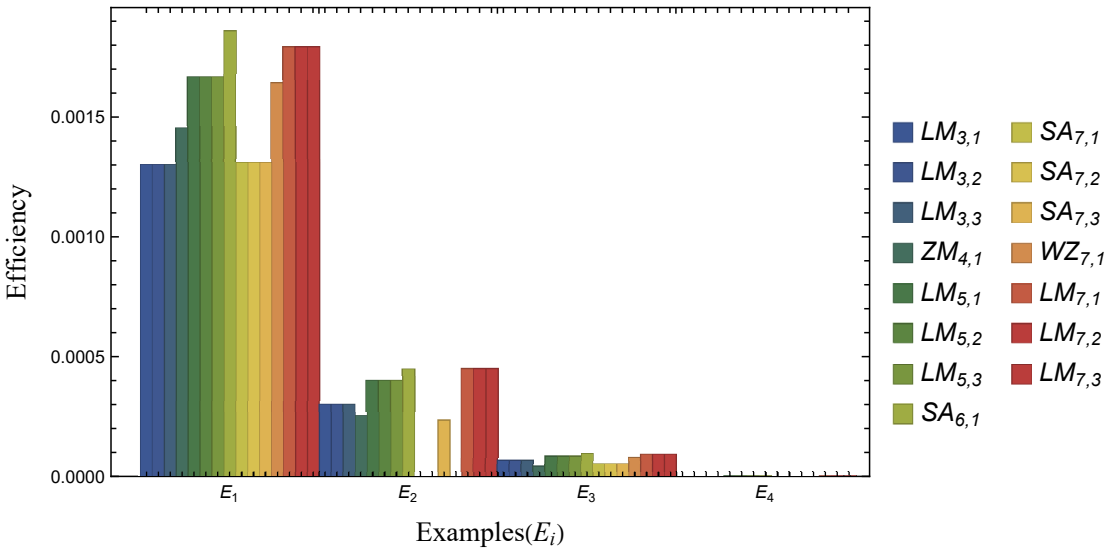


Figure 5.8: Efficiency log plot of Examples (E_i) 5.5.1–5.5.4 w.r.t different methods.

Table 5.6: Numerical comparison results on Example 5.5.4

Methods	n	ξ_2	ξ_3	ξ_4	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$LM_{3,1}$	6	7.02(−09)	1.23(−27)	6.61(−84)	431000	1.00000254898768	3	3.0000	91.859
$LM_{3,2}$	6	7.02(−09)	1.23(−27)	6.60(−84)	431000	1.00000254898768	3	3.0000	91.359
$LM_{3,3}$	6	7.01(−09)	1.23(−27)	6.59(−84)	431000	1.00000254898768	3	2.9961	92.125
$ZM_{4,1}$	5	5.04(−17)	7.98(−70)	5.02(−281)	701110	1.000001977287061	4	4.0000	60.859
$LM_{5,1}$	4	5.35(−22)	2.69(−111)	8.62(−558)	440580	1.00000365300479	5	5.0000	145.593
$LM_{5,2}$	4	5.35(−22)	2.69(−111)	8.57(−558)	440580	1.00000365300479	5	5.0000	145.187
$LM_{5,3}$	4	5.35(−22)	2.68(−111)	8.48(−558)	440580	1.000003653004795	5	5.0000	145.203
$SA_{6,1}$	4	1.16(−30)	2.27(−185)	1.29(−1113)	440530	1.00000406728966	6	6.0000	83.313
$SA_{7,1}$	-	Diverges	–	–	–	–	-	–	–
$SA_{7,2}$	-	Diverges	–	–	–	–	-	–	–
$SA_{7,3}$	-	Diverges	–	–	–	–	-	–	–
$WZ_{7,1}$	-	Diverges	–	–	–	–	-	–	–
$LM_{7,1}$	4	4.18(−41)	2.36(−289)	4.25(−2027)	450160	1.00000432271716	7	7.0000	191.562
$LM_{7,2}$	4	4.18(−41)	2.35(−289)	4.20(−2027)	450160	1.00000432271716	7	7.0000	195.235
$LM_{7,3}$	4	4.18(−41)	2.35(−289)	4.10(−2027)	450160	1.00000432271716	7	6.9999	193.923

5.5.2 Application to chemical engineering

Example 5.5.5. *In chemical engineering, a significant challenge involves predicting the diffusion and reaction processes within a porous catalyst pellet. The primary aim is to determine the overall reaction rate of the catalyst pellet. The preservation of mass within a spherical domain yields a mathematical model, as shown in Example 4.5.1. We employ central divided differences to solve this problem, which transforms this into a nonlinear system. Subsequently, our problem* (4.1) *results in system* $F(X) = 0$*, where* $F : \mathbb{R}^{k+1} \to \mathbb{R}^{k+1}$ *given by*

$$F(X) = \begin{cases} X_1 - X_0 - \dfrac{1}{6}\hat{\hbar}^2\phi^2 X_0^2, \\ \left(1 - \dfrac{1}{s}\right) X_{s-1} + \left(1 + \dfrac{1}{s}\right) X_{s+1} - 2X_s - \frac{1}{6}\hat{\hbar}^2\phi^2 X_s^2, \ s = 1, 2, \cdots, k-1, \\ \left(1 - \dfrac{1}{k}\right) X_{k-1} + 1 + \dfrac{1}{k} - 2X_k. \end{cases} \tag{5.37}$$

For the initial guess $X^{(0)} = (0.1, \cdots, 0.1)^T$ *and* $k = 10$*, the proposed solvers converge to the approximate solution*

$$\begin{aligned} X^* =& (0.0361320888367\ldots, 0.0367075281067\ldots, 0.0392749976060\ldots, 0.0436182869414\ldots, \\ & 0.0502495007795\ldots, 0.0602350754674\ldots, 0.0755922657555\ldots, 0.1003330789655\ldots, \\ & 0.1432405989194\ldots, 0.2264025349114\ldots, 0.4176792391196\ldots)^T. \end{aligned}$$

We have presented the computational results in the Table 5.7 for parametric values $(k+1, \mu) = (11, 7.8)$*.*

Table 5.7: Numerical comparison results on Example 5.5.5

Methods	n	ξ_2	ξ_3	ξ_4	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$LM_{3,1}$	6	2.7(−05)	1.7(−13)	4.3(−38)	5511	1.00019936886219	3	3.0037	0.594
$LM_{3,2}$	6	2.7(−05)	1.7(−13)	4.3(−38)	5511	1.00019936886219	3	3.0037	0.984
$LM_{3,3}$	6	2.7(−05)	1.7(−13)	4.3(−38)	5511	1.00019936886219	3	3.0037	0.843
$ZM_{4,1}$	5	3.4(−9)	9.9(−35)	2.0(−136)	8482.6	1.00016344135606	4	3.9820	0.578
$LM_{5,1}$	5	5.7(−10)	2.2(−44)	1.4(−216)	6017.2	1.00026750867050	5	5.0030	0.563
$LM_{5,2}$	5	5.7(−10)	2.2(−44)	1.4(−216)	6017.2	1.00026750867050	5	5.0030	0.359
$LM_{5,3}$	5	5.7(−10)	2.2(−44)	1.4(−216)	6017.2	1.00026750867050	5	5.0030	0.265
$SA_{6,1}$	6	4.7(−2)	1.9(−6)	2.1(−31)	6006.2	1.00029836281728	5	5.6787	0.343
$SA_{7,1}$	–	*Diverges*	−−	−−	−−	−−	-	–	–
$SA_{7,2}$	–	*Diverges*	−−	−−	−−	−−	-	–	–
$SA_{7,3}$	–	*Diverges*	−−	−−	−−	−−	-	–	–
$WZ_{7,1}$	4	5.3(+01)	3.7(+02)	5.8(+02)	8311.4	1.00023415284553	7	1.3370	1.438
$LM_{7,1}$	4	4.0(−16)	4.5(−106)	4.9(−734)	6523.4	1.00029834141819	7	6.9816	0.500
$LM_{7,2}$	4	4.2(−09)	2.2(−56)	1.7(−387)	6523.4	1.00029834141819	7	6.9860	0.531
$LM_{7,3}$	4	4.0(−16)	4.5(−106)	4.9(−734)	6523.4	1.00029834141819	7	6.9816	0.625

5.5.3 Applications to partial differential equations

This section focuses on numerical computations for two well-known partial differential equations using the proposed methods and various existing solvers. The following problems are considered:

Example 5.5.6. *Theoretical physics makes considerable use of an elliptic partial differential equa-*

tion known as Poisson's equation. It is often used to calculate the potential field produced by a specific distribution of electric charge or mass density. It is possible to calculate the associated electrostatic or gravitational force field once the potential field has been identified. In order to obtain such a force, we consider its mathematical model in the domain $(x, y) \in [0, 1] \times [0, 1]$, *given as follows*

$$\Upsilon_{xx} + \Upsilon_{yy} = \Upsilon^2, \tag{5.38}$$

having boundary constraints

$$\begin{aligned} \Upsilon(x, 0) &= 1 - x + 2x^2, \quad \Upsilon(x, 1) &&= 2, \\ \Upsilon(0, y) &= 1 - y + 2y^2, \quad \Upsilon(1, y) &&= 2. \end{aligned} \tag{5.39}$$

Let $\Upsilon_{i,j} = \Upsilon(x_i, y_j)$ *denote the estimated solution at the mesh grid points, where* $x_0 = y_0 = 0$ *and*

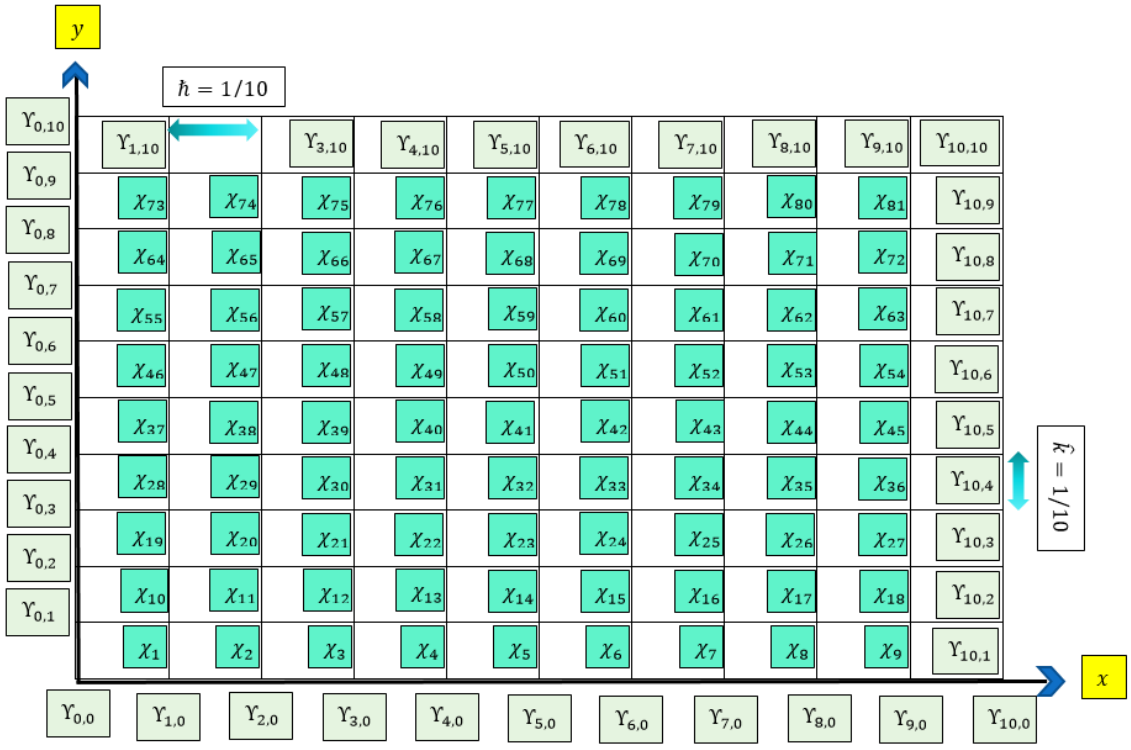


Figure 5.9: Discretization of the domain $0 \leq x, y \leq 1$ with the step sizes $\hbar, \hat{k} = 1/10$.

$x_{\hat{m}} = y_{\hat{n}} = 1$. *Henceforth, we denote* $\Upsilon(x_i, y_j) = \Upsilon_{i,j} \to X_{j+(i-1)\hat{m}}$ *for each* i *and* j. *Then, the domain of* x *and* y *can be discretized with step sizes* $\hbar$ *and* $\hat{k}$ *as shown in Figure 5.9 in which the boundary and initial points are colored green with values* $\Upsilon_{0,j} = 2(i\hbar)^2 - i\hbar + 1$, $\Upsilon_{\hat{m},j} = 2$, *and*

$\Upsilon_{i,0} = 2(j\hat{k})^2 - j\hat{k} + 1,\ \Upsilon_{i,\hat{n}} = 2$, *and also the unknown variables* t_i *are colored black. Assume* $\hat{m}$ *and* $\hat{n}$ *defines the step count in* x *and* y *directions, respectively, and* $\hbar$ *and* $\hat{k}$ *be their respective step sizes* ($\hbar = 1/\hat{m},\ \hat{k} = 1/\hat{n}$). *Now, the problem is been discretized by utilizing the central divided differences, i.e.,*

$$\Upsilon_{xx} = \frac{(\Upsilon_{i-1,j} - 2\Upsilon_{i,j} + \Upsilon_{i+1,j})}{\hbar^2}, \quad \Upsilon_{yy} = \frac{(\Upsilon_{i,j+1} - 2\Upsilon_{i,j} + \Upsilon_{i,j-1})}{\hat{k}^2},$$

we obtain the following systems of nonlinear equations:

$$\Upsilon_{i+1,j} - 4\Upsilon_{i,j} + \Upsilon_{i-1,j} + \Upsilon_{i,j+1} + \Upsilon_{i,j-1} - \hbar^2\Upsilon_{i,j}^2 = 0, \tag{5.40}$$

where $i = 1, 2, \ldots, \hat{m} - 1,\ j = 1, 2, \ldots, \hat{n} - 1$. *We have considered* $\hat{m} = 10$ *and* $\hat{n} = 10$, *and thereby solve the resulting nonlinear system of* 81 *nonlinear equations in* 81 *unknowns by using initial vector* $X^{(0)} = (2, \cdots, 2)^T$, *to get an approximate solution:*

$$\begin{aligned}
X^* = (&0.925427\ldots, 0.935136\ldots, 0.963300\ldots, 1.016549\ldots, 1.098466\ldots, 1.211243\ldots, 1.356417\ldots,\\
&1.535275\ldots, 1.749184\ldots, 0.935136\ldots, 0.9805620\ldots, 1.030795\ldots, 1.094764\ldots, 1.178137\ldots,\\
&1.284762\ldots, 1.417550\ldots, 1.579069\ldots, 1.772058\ldots, 0.963300\ldots, 1.030795\ldots, 1.095180\ldots,\\
&1.165562\ldots, 1.248435\ldots, 1.348623\ldots, 1.470045\ldots, 1.616329\ldots, 1.791382\ldots, 1.016549\ldots,\\
&1.094764\ldots, 1.165562\ldots, 1.237454\ldots, 1.317002\ldots, 1.409440\ldots, 1.519289\ldots, 1.650943\ldots,\\
&1.809232\ldots, 1.098466\ldots, 1.178137\ldots, 1.248435\ldots, 1.317002\ldots, 1.390025\ldots, 1.472710\ldots,\\
&1.569810\ldots, 1.686181\ldots, 1.827334\ldots, 1.211243\ldots, 1.284762\ldots, 1.348623\ldots, 1.409440\ldots,\\
&1.472710\ldots, 1.543252\ldots, 1.625703\ldots, 1.725066\ldots, 1.847317\ldots, 1.356417\ldots, 1.417550\ldots,\\
&1.470045\ldots, 1.519289\ldots, 1.569810\ldots, 1.625703\ldots, 1.691114\ldots, 1.770824\ldots, 1.870992\ldots,\\
&1.535275\ldots, 1.579069\ldots, 1.616329\ldots, 1.650943\ldots, 1.686181\ldots, 1.725066\ldots, 1.770824\ldots,\\
&1.827480\ldots, 1.900835\ldots, 1.749184\ldots, 1.772058\ldots, 1.791382\ldots, 1.809232\ldots, 1.827334\ldots,\\
&1.847317\ldots, 1.870992\ldots, 1.900835\ldots, 1.940999)^T,
\end{aligned}$$

which is also shown in Figure 5.10. In this problem, we use $(k, \mu) = (81, 3)$ *to calculate computational costs and efficiency indices. The computational results are displayed in the Table 5.8.*

Example 5.5.7. *Let us now examine the reaction-diffusion system, also known as Fisher's equation,*

whose mathematical expression is given as follows

$$\upsilon_t = D_f \upsilon_{xx} + \upsilon(1-\upsilon), \tag{5.41}$$

with homogeneous Neumann's boundary conditions,

$$\left.\begin{aligned} &\upsilon(x,0) = 0.5\cos(\pi x) + 1.5, 0 \le x \le 1, \\ &\upsilon_x(0,t) = 0,\ \upsilon_x(1,t) = 0, \forall\ t \ge 0. \end{aligned}\right\}$$

The equation features a semi-linear reaction-diffusion term with inhomogeneous terms, which can

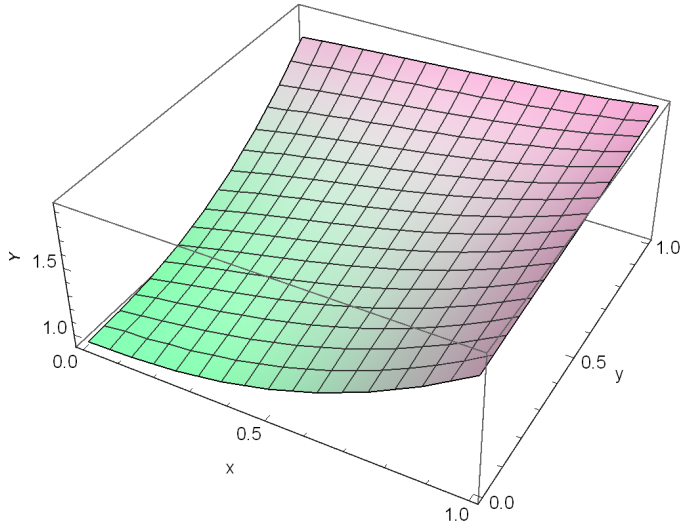


Figure 5.10: Approximated solution of Poisson's equation for $x, y \in [0,1]$ by the method $\mathrm{LM}_{7,1}$.

exhibit traveling wave solutions that switch between equilibrium states.

Let this diffusive equation have an exact solution denoted by $\upsilon = \upsilon(x,t)$. *Suppose* $w_{i,j} \simeq \upsilon(x_i, t_j) \to X_{j+(i-1)\hat{m}}$ *is its approximate solution at the grid points of the mesh. Let* $\hbar = \frac{1}{\hat{m}}$ *and* $\hat{k} = \frac{1}{\hat{n}}$ *be the step sizes, where* $\hat{m}$ *and* $\hat{n}$ *are the step count of space and time, respectively. To solve the above partial differential equation, we can discretize it by applying backward differences to* υ_t *and central differences to other terms, i.e.,*

$$\upsilon_x(x_i,t_j) \simeq \frac{(w_{i+1,j} - w_{i,j})}{\hbar}, \quad \upsilon_t(x_i,t_j) \simeq \frac{(w_{i,j} - w_{i,j-1})}{\hat{k}},$$
$$\upsilon_{xx}(x_i,t_j) \simeq \frac{(w_{i+1,j} - 2w_{i,j} + w_{i-1,j})}{\hbar^2}.$$

This gives rise to a nonlinear system of $(\hat{m}-1)\times\hat{n}$ *equations. Here, we assume* $D_f = 1$ *with initial*

Table 5.8: Numerical comparison results on Example 5.5.6

Methods	n	ξ_2	ξ_3	ξ_4	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$LM_{3,1}$	6	5.5(+0)	5.6(−12)	1.0(−38)	322029	1.00000341153798	3	2.9996	30.975
$LM_{3,2}$	6	5.5(+0)	5.6(−12)	1.0(−38)	322029	1.00000341153798	3	2.9996	32.9953
$LM_{3,3}$	6	5.5(+0)	5.6(−12)	1.0(−38)	322029	1.00000341153798	3	2.9996	40.4843
$ZM_{4,1}$	7	5.5(+0)	1.0(−6)	1.5(−28)	742090	1.00000186809740	4	3.9889	24.2438
$LM_{5,1}$	5	2.1(−6)	5.7(−32)	9.5(−165)	342376	1.00000470079899	5	4.9979	34.3296
$LM_{5,2}$	5	2.1(−6)	5.7(−32)	9.5(−165)	342376	1.00000470079899	5	4.9979	33.9688
$LM_{5,3}$	5	2.1(−6)	5.7(−32)	9.5(−165)	342376	1.00000470079899	5	4.9979	39.2469
$SA_{6,1}$	5	2.2(−6)	3.7(−44)	1.2(−270)	342295	1.00000523455824	6	5.9983	42.886
$SA_{7,1}$	4	5.9(−9)	4.6(−71)	8.7(−506)	691929	1.00000281230139	7	6.9997	283.03
$SA_{7,2}$	4	1.1(−8)	6.5(−69)	2.0(−590)	691929	1.00000281230139	7	6.9995	275.488
$SA_{7,3}$	4	1.3(−8)	3.0(−68)	1.0(−485)	691929	1.00000281230139	7	6.9994	249.867
$WZ_{7,1}$	4	1.7(−7)	3.1(−53)	1.3(−326)	411566	1.00000472807000	7	5.9781	57.902
$LM_{7,1}$	4	9.2(−8)	2.1(−61)	7.9(−437)	362723	1.00000536473624	7	6.9986	50.2141
$LM_{7,2}$	4	9.2(−8)	2.1(−61)	7.9(−437)	362723	1.00000536473624	7	6.9986	45.5375
$LM_{7,3}$	4	9.2(−8)	2.1(−61)	7.9(−437)	362723	1.00000536473624	7	6.9986	43.3017

estimate $X^{(0)} = (X_1, X_2, \ldots, X_{196}) = (0.5, \cdots, 0.5)^T$ *for* $\hat{m} = 15$ *and* $\hat{n} = 14$ *of size* 196*, and the approximate solution is evaluated using different techniques, given by*

$$
\begin{aligned}
X^* = (&1.698102\ldots, 1.536494\ldots, 1.438474\ldots, 1.374222\ldots, 1.328692\ldots, 1.294034\ldots, 1.266064\ldots,\\
&1.242497\ldots, 1.222043\ldots, 1.203943\ldots, 1.187722\ldots, 1.173065\ldots, 1.159750\ldots, 1.147607\ldots,\\
&1.685266\ldots, 1.530102\ldots, 1.435178\ldots, 1.372510\ldots, 1.327800\ldots, 1.293569\ldots, 1.265821\ldots,\\
&1.242370\ldots, 1.221976\ldots, 1.203908\ldots, 1.187703\ldots, 1.173055\ldots, 1.159745\ldots, 1.147604\ldots,\\
&1.660669\ldots, 1.517660\ldots, 1.428752\ldots, 1.369171\ldots, 1.326061\ldots, 1.292661\ldots, 1.265346\ldots,\\
&1.242121\ldots, 1.221846\ldots, 1.203839\ldots, 1.187667\ldots, 1.173037\ldots, 1.159735\ldots, 1.147599\ldots,\\
&1.625776\ldots, 1.499812\ldots, 1.419516\ldots, 1.364371\ldots, 1.323561\ldots, 1.291356\ldots, 1.264665\ldots,\\
&1.241765\ldots, 1.221660\ldots, 1.203741\ldots, 1.187616\ldots, 1.173010\ldots, 1.159721\ldots, 1.147592\ldots,\\
&1.582413\ldots, 1.477457\ldots, 1.407931\ldots, 1.358349\ldots, 1.320424\ldots, 1.289720\ldots, 1.263810\ldots,\\
&1.241317\ldots, 1.221425\ldots, 1.203618\ldots, 1.187551\ldots, 1.172976\ldots, 1.159703\ldots, 1.147582\ldots,
\end{aligned}
$$

$$
\begin{aligned}
&1.532718\ldots, 1.451708\ldots, 1.394572\ldots, 1.351406\ldots, 1.316809\ldots, 1.287835\ldots, 1.262825\ldots,\\
&1.240802\ldots, 1.221155\ldots, 1.203477\ldots, 1.187477\ldots, 1.172936\ldots, 1.159682\ldots, 1.147571\ldots,\\
&1.479075\ldots, 1.423832\ldots, 1.380104\ldots, 1.343887\ldots, 1.312895\ldots, 1.285793\ldots, 1.261759\ldots,\\
&1.240244\ldots, 1.220863\ldots, 1.203324\ldots, 1.187396\ldots, 1.172894\ldots, 1.159660\ldots, 1.147560\ldots,\\
&1.424026\ldots, 1.395200\ldots, 1.365246\ldots, 1.336169\ldots, 1.308878\ldots, 1.283699\ldots, 1.260665\ldots,\\
&1.239672\ldots, 1.220563\ldots, 1.203166\ldots, 1.187314\ldots, 1.172851\ldots, 1.159637\ldots, 1.147548\ldots,\\
&1.370186\ldots, 1.367226\ldots, 1.350741\ldots, 1.328638\ldots, 1.304960\ldots, 1.281657\ldots, 1.259598\ldots,\\
&1.239114\ldots, 1.220271\ldots, 1.203013\ldots, 1.187233\ldots, 1.172808\ldots, 1.159615\ldots, 1.147536\ldots,\\
&1.320137\ldots, 1.341299\ldots, 1.337315\ldots, 1.321672\ldots, 1.301337\ldots, 1.279769\ldots, 1.258612\ldots,\\
&1.238598\ldots, 1.220001\ldots, 1.202871\ldots, 1.187159\ldots, 1.172769\ldots, 1.159594\ldots, 1.147525\ldots,\\
&1.276330\ldots, 1.318723\ldots, 1.325647\ldots, 1.315622\ldots, 1.298192\ldots, 1.278130\ldots, 1.257757\ldots,\\
&1.238151\ldots, 1.219766\ldots, 1.202748\ldots, 1.187094\ldots, 1.172735\ldots, 1.159576\ldots, 1.147516\ldots,\\
&1.240991\ldots, 1.300653\ldots, 1.316328\ldots, 1.310794\ldots, 1.295683\ldots, 1.276823\ldots, 1.257074\ldots,\\
&1.237794\ldots, 1.219579\ldots, 1.202650\ldots, 1.187043\ldots, 1.172708\ldots, 1.159562\ldots, 1.147508\ldots,\\
&1.216035\ldots, 1.288034\ldots, 1.309835\ldots, 1.307433\ldots, 1.293937\ldots, 1.275913\ldots, 1.256599\ldots,\\
&1.237546\ldots, 1.219449\ldots, 1.202582\ldots, 1.187007\ldots, 1.172689\ldots, 1.159552\ldots, 1.147503\ldots,\\
&1.202998\ldots, 1.281543\ldots, 1.306502\ldots, 1.305708\ldots, 1.293041\ldots, 1.275446\ldots, 1.256356\ldots,\\
&1.237418\ldots, 1.219383\ldots, 1.202547\ldots, 1.186989\ldots, 1.172680\ldots, 1.159547\ldots, 1.147500\ldots)^T,
\end{aligned}
$$

also displayed in Figure 5.11. Also, the computing cost and efficiency for parametric values $(k, \mu) = (196, 9)$ *are provided in Table 5.9 for several procedures.*

The results obtained from numerical experiments imply the stable behavior of the proposed techniques in Tables 5.3–5.9, where the ρ_{COC} of each solver is in good agreement with the theoretical convergence order. The elapsed time (in seconds) for program execution indicates that the proposed solvers are more efficient than other existing techniques. It is generally considered that a method that consumes less computing time is highly efficient, and vice versa. Moreover, the presented procedures exhibit better efficiency than the compared solvers, as illustrated in Figure 5.8 and Figure 5.12. Therefore, the numerical experiments performed on several examples from different types provide strong evidence for the above findings.

The following section will discuss the stability of the methods using the basins of attraction.

Table 5.9: Numerical comparison results on Example 5.5.7

Methods	n	ξ_1	ξ_2	ξ_3	$\mathcal{C}_{M_{\rho,i}}$	$E_{M_{\rho,i}}$	ρ	ρ_{COC}	e-Time
$LM_{3,1}$	6	1.3(+0)	1.4(−3)	1.1(−12)	4.27829(+6)	1.00000025678786	3	3.0641	216.605
$LM_{3,2}$	6	1.3(+0)	1.4(−3)	1.1(−12)	4.27829(+6)	1.00000025678786	3	3.0641	197.739
$LM_{3,3}$	6	1.3(+0)	1.4(−3)	1.1(−12)	4.27829(+6)	1.00000025678786	3	3.0641	162.189
$ZM_{4,1}$	5	2.0(+0)	3.7(−9)	2.2(−40)	1.01445(+7)	1.00000013665492	4	4.0359	161.417
$LM_{5,1}$	5	1.2(+0)	1.8(−6)	1.3(−35)	4.39632(+6)	1.00000036608766	5	5.0161	291.900
$LM_{5,2}$	5	1.2(+0)	1.8(−6)	1.3(−35)	4.39632(+6)	1.00000036608766	5	5.0161	297.836
$LM_{5,3}$	5	1.2(+0)	1.8(−6)	1.3(−35)	4.39632(+6)	1.00000036608766	5	5.0161	354.691
$SA_{6,1}$	5	2.3(+0)	4.0(−6)	1.9(−40)	4.39612(+6)	1.00000040757726	5	5.9514	425.302
$SA_{7,1}$	–	*Diverges*	−−	−−	−−	−−	-	–	–
$SA_{7,2}$	–	*Diverges*	−−	−−	−−	−−	-	–	–
$SA_{7,3}$	–	*Diverges*	−−	−−	−−	−−	-	–	–
$WZ_{7,1}$	4	1.1(+0)	1.7(−9)	6.8(−64)	5.26131(+6)	1.00000036985311	7	6.1787	478.663
$LM_{7,1}$	4	6.1(−1)	3.0(−11)	2.9(−83)	4.51435(+6)	1.00000043104996	7	6.9860	520.431
$LM_{7,2}$	4	6.1(−1)	3.0(−11)	2.9(−83)	4.51435(+6)	1.00000043104996	7	6.9860	451.728
$LM_{7,3}$	4	6.1(−1)	3.0(−11)	2.9(−83)	4.51435(+6)	1.00000043104996	7	6.9860	413.016

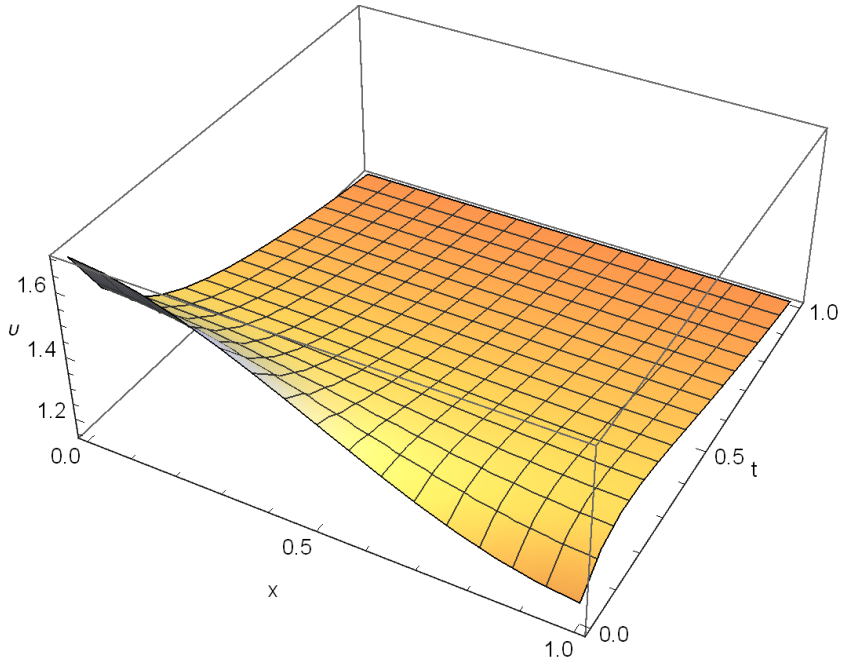


Figure 5.11: Approximated solution of Fisher's equation for $t \in [0, 1]$ by the method $LM_{7,1}$.

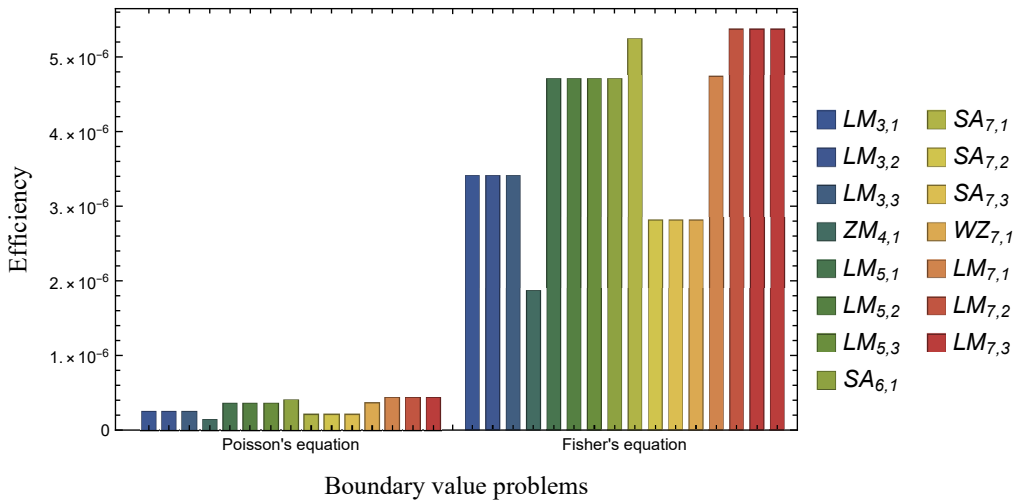


Figure 5.12: Efficiency log plot of Poisson's and Fisher's equation w.r.t different methods.

5.6 Basins of attraction

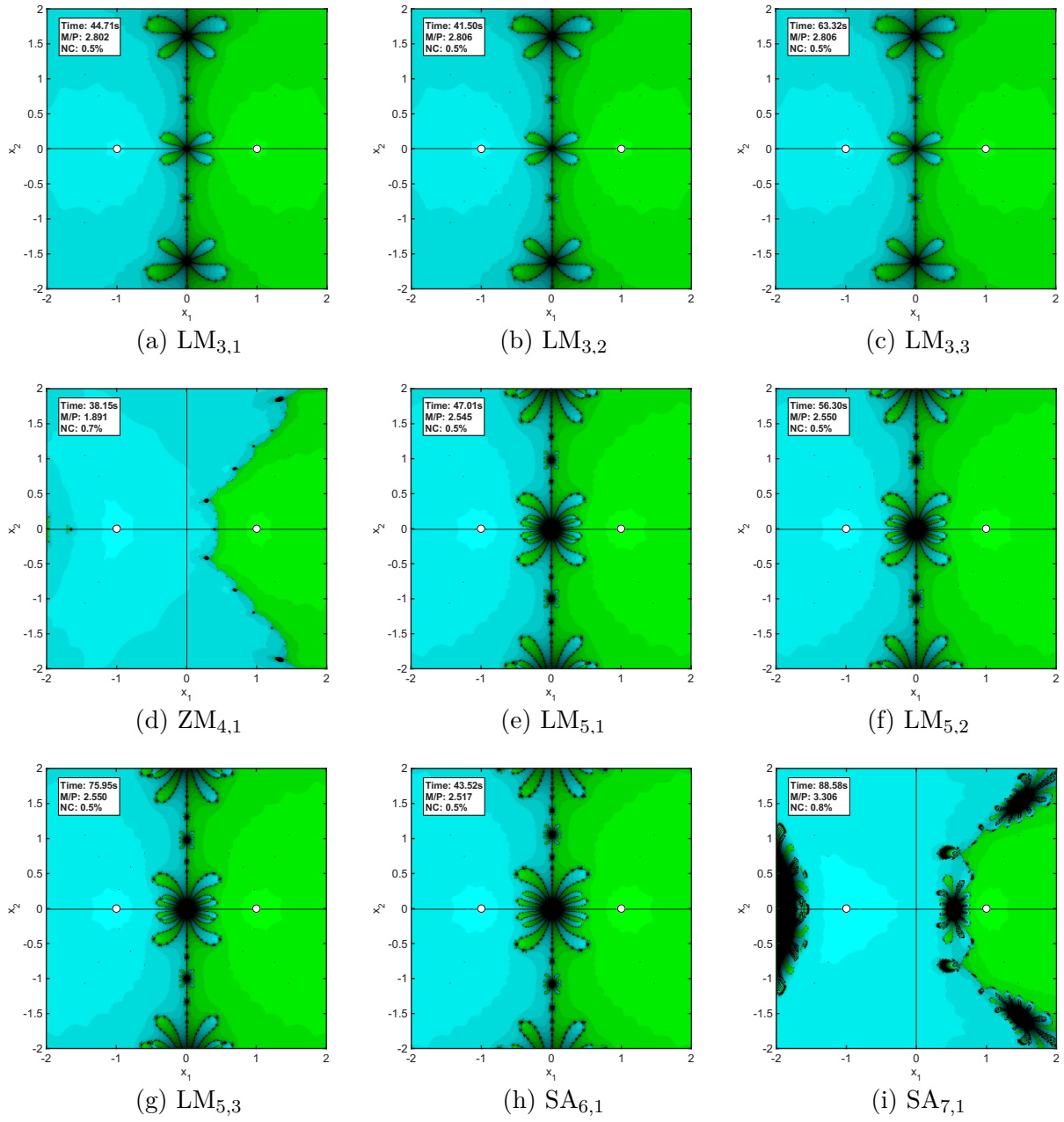
To investigate the dynamic behavior of the solvers presented in the previous sections, we analyze their basins of attraction on the polynomial system

$$\begin{cases} X_1^2 - X_2^2 - 1 &= 0, \\ X_1 X_2 &= 0, \end{cases} \tag{5.42}$$

which has solutions $\left\{ r_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, r_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$. We employ an iterative procedure starting at each point in a square $[-2, 2] \times [-2, 2]$ with 401×401 points, which encompasses all the roots of the system, to create these basins of attraction connected to the roots of the nonlinear systems. Each point receives a color based on the root to which the appropriate orbit of the iterative process, starting at the point, converges. In particular, we assign cyan to the r_1 root and green to the r_2 root. We designate those points with a black color if the associated orbit, with a tolerance of 10^{-3} in a maximum of 50 iterations, does not converge to any root of the polynomial.

For the above test case, we can see in Figures 5.13–5.14 that all of the polynomial system's roots have a unique basin of attraction, each of which is colored differently. Additionally, the Julia set can be viewed as a set of chaotic black lines. From Figures 5.13–5.14, it can be seen that the fractals of $\mathrm{SA}_{7,1}$, $\mathrm{SA}_{7,2}$, and $\mathrm{SA}_{7,3}$ have more black shaded regions, while the rest have a more stable form. Thus, we conclude that the proposed methods sustain stability in a reasonable manner compared

to existing counterparts.



(a) $LM_{3,1}$ (b) $LM_{3,2}$ (c) $LM_{3,3}$

(d) $ZM_{4,1}$ (e) $LM_{5,1}$ (f) $LM_{5,2}$

(g) $LM_{5,3}$ (h) $SA_{6,1}$ (i) $SA_{7,1}$

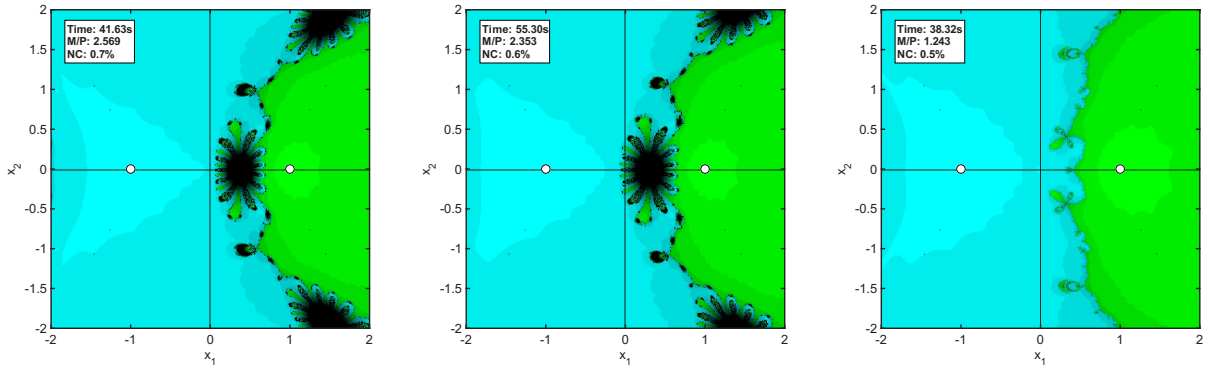
Figure 5.13: Basins of attraction for iterative solvers.

5.7 Concluding remarks

We have developed new families of multi-point iterative procedures that can accurately approximate solutions to nonlinear systems with third and fifth-order convergence. Additionally, we introduce a

generalized version of the method that utilizes $q+1$ steps with an increasing convergence order of $2q+1$, where q is a natural number. The key innovation of these families is that the convergence order is increased by two at the expense of an auxiliary function per step, resulting in a computationally efficient scheme that utilizes only one inverse matrix operator per step. We compared the suggested solvers to existing solvers by analyzing their computational efficiency using various methods. Our results show that the presented methods outperform existing robust techniques, particularly for large-scale systems. Numerical analyses confirmed that the proposed solvers are effective, efficient, and superior to the compared solvers in terms of iteration counter, residual error, computational efficiency, stability, and program execution time.

We now extend this methodology to the theory of matrix functions. The next chapter presents



(a) $SA_{7,2}$ (b) $SA_{7,3}$ (c) $WZ_{7,1}$

(d) $LM_{7,1}$ (e) $LM_{7,2}$ (f) $LM_{7,3}$

Figure 5.14: Basins of attraction for different iterative solvers.

a novel, fourth-order matrix iteration for the numerical computation of the matrix sign function, an important tool in control theory and other applications.

Chapter 6

Globally convergent iterative scheme for computing matrix sign function

In this chapter, we propose a novel matrix iteration for numerically computing the matrix sign function. An extensive convergence analysis is carried out in the matrix case to achieve fourth-order convergence. The basins of attraction are illustrated to demonstrate the global convergence behavior of the proposed method. Analytically, it is shown that the presented scheme is asymptotically stable. Furthermore, theoretical developments are numerically verified by discussing the clustering of eigenvalues around ± 1. Moreover, a class of numerical examples for matrices of various dimensions is assessed to exhibit the efficacy of the proposed method.

6.1 Introduction

The matrix function $\mathcal{F}(A)$ of any given matrix $A \in \mathbb{C}^{m\times m}$ is generally defined by the Cauchy integral formula as follows:

$$\mathcal{F}(A) = \frac{1}{2\pi i}\oint_{\sigma} \mathcal{F}(\zeta)(\zeta I - A)^{-1} d\zeta, \tag{6.1}$$

where $\mathcal{F} : \Theta \subseteq \mathbb{C} \to \mathbb{C}$ is a holomorphic function in a closed loop σ that envelopes each eigenvalue of A (or simply the domain of $\mathcal{F}$ should contain these eigenvalues analytically), and I is an identity matrix of order $m \times m$. The integral of any arbitrary matrix $\mathcal{Q}$ can be thought of as a matrix in which elements are the integrals of $\mathcal{Q}$'s entries. This attractive formula is difficult to understand

The contents of this chapter are published in Mathematical Methods in the Applied Sciences, Vol. 48, pp. 7580–7594, 2025 (SCIE, I.F.: 1.8)

when it comes to discovering the function of matrices and requires studying complex analysis. As a result, numerous techniques for determining the matrix function have been developed and researched, including the Jordan canonical form, polar decomposition, and others, which can be found in the literature by Howland [162], Filbir [125], Higham [154], and Cordero et al. [86, 87]. Among the wide applications of a matrix function in the domain of mathematics, we direct the readers to explore references [225, 333, 340].

Let us recall the definition of a sign function. It is known that in the scalar case, the sign function is represented for each $\zeta \in \mathbb{C}$ not located on the imaginary axis by

$$\operatorname{sign}(\zeta) = \begin{cases} +1, & \Re(\zeta) > 0, \\ -1, & \Re(\zeta) < 0, \end{cases}$$

where $\Re$ represents the real part of ζ. In 1980, Roberts [283] developed the matrix extension of this function as a tool to obtain an explicit solution of the Riccati and Lyapunov algebraic matrix equations. He derived the integral formulation of $\operatorname{sign}(A)$ by defining the sign function as a Cauchy integral:

$$\operatorname{sign}(A) = S = \frac{2}{\pi} A \int_0^\infty (t^2 I + A^2)^{-1} dt. \tag{6.2}$$

The sign of the matrix holds the fundamental theoretical and computational relationships with the polar decomposition [156], the square root of a matrix, and the p^{th} roots of a matrix. For example, corresponding iterations for the matrix sign function can yield a large class of matrix square root iterations that require the discussion and design of a novel iterative method to find the sign of a matrix.

On the imaginary axis, we assume that the set containing eigenvalues of non-singular matrix $A \in \mathbb{C}^{m\times m}$ is empty. Thus, the matrix sign function S can be expressed uniquely. In research articles by Higham [155] and Kenney and Laub [210, 211], the simplest formulation is the matrix sign decomposition, provided by

$$A = SN = A(A^2)^{-1/2}(A^2)^{1/2}, \tag{6.3}$$

whereas $S = A(A^2)^{-1/2}$ is the matrix sign function and $1/2$ denotes the principal square root of a given matrix.

In fact, similar to the matrix sign function, the matrix disk function can be utilized to obtain

invariant subspaces. Other applications of the matrix sign function can be explored in references [246, 315]. This matrix sign function has a number of characteristics, and Kenny and Laub [210] provide a few of them, as mentioned in the introduction 1.4.5. The reader is directed to study the reference [171] for more information on the matrix sign function.

Remember that the matrix sign function is indeed a primary matrix function with a non-primary flavor, i.e., for a given matrix $A \in \mathbb{C}^{m\times m}$ it is (broadly) a non-primary square root of I depending upon A. Iteration functions of the following type, as cited by Higham [154], can be used to compute a variety of matrix functions $\mathcal{F}(A)$:

$$\mathcal{X}_{n+1} = G(\mathcal{X}_n),\ n = 0, 1, 2, \ldots, \tag{6.4}$$

where $\mathcal{X}_0$ is not arbitrary, but G is a fixed iterative function of A for the iterations used in practice. When the computational cost is taken into consideration, it is clear that G is a polynomial or rational function. The linear system solutions with several right-side expressions, or even an explicit matrix inverse, would be required for a rational G. It is noticeable that the results and concepts from nonlinear iterations in the scalar case do not always apply to matrix cases. Standard convergence measures at a fixed point of G stated in terms of derivatives of Frechét or higher-order kind do not simply convert into similar matrix environments.

In literature, Newton's method [283] (NM) is the frequent and well-known means of determining the sign of a non-singular square matrix with a quadratic convergence rate, as given as

$$\mathcal{X}_{n+1} = \frac{1}{2}(\mathcal{X}_n + \mathcal{X}_n^{-1}),\ n = 0, 1, 2 \ldots, \tag{6.5}$$

with $\mathcal{X}_0 = A$ as an initial matrix. Moreover, the order of convergence of the NM method (6.5) can be accelerated via the norm scaling parameter at the amount of just two l_2-matrix norms, and it is called the accelerated Newton's method (ANM), represented by

$$\begin{cases} \mathcal{X}_0 = A, \\ \nu_n = \sqrt{\dfrac{||\mathcal{X}_n^{-1}||}{||\mathcal{X}_n||}}, \\ \mathcal{X}_{n+1} = \dfrac{1}{2}\left(\nu_n \mathcal{X}_n + {\nu_n}^{-1}\mathcal{X}_n^{-1}\right),\ n = 0, 1, 2 \ldots. \end{cases} \tag{6.6}$$

Several authors have attempted to improve the convergence rate of matrix iteration (6.5) by scaling;

in fact, this is a quite efficient approach. In order to achieve this goal, Gomilko [138] and Kenney [209] used Padé approximants to $\mathcal{F}(\xi) = (1-\xi)^{-1/2}$ and developed a general family of matrix iterative techniques for computing S. Here, consider the (r,s)-Padé approximant to $\mathcal{F}(\xi)$ by $\frac{P_{r,s}}{Q_{r,s}}$, where $r+s \geq 1$. Then, for any $r \geq s-1$, the general scalar iterative expression is

$$x_{n+1} = \frac{x_n P_{r,s}(1-x_n^2)}{Q_{r,s}(1-x_n^2)} := \psi_{2r+1,2s},\ n = 0,1,2\ldots, \tag{6.7}$$

having convergence order $r+s+1$ and converging towards ± 1.

It is significant to note that there are some well-known matrix schemes for determining S, including the Newton-Schultz method [293] (NSM):

$$\mathcal{X}_{n+1} = \frac{1}{2}\mathcal{X}_n(3I - \mathcal{X}_n^2),\ n = 0,1,2\ldots, \tag{6.8}$$

and Halley's method [209] (HM):

$$\mathcal{X}_{n+1} = [I + 3\mathcal{X}_n^2][\mathcal{X}_n(3I + \mathcal{X}_n^2)]^{-1},\ n = 0,1,2\ldots, \tag{6.9}$$

which are members of the Padé family or its reciprocals.

Recently, Soleymani [337] proposed an iterative method for computing the matrix sign function S with a convergence rate of four as follows:

$$\mathcal{X}_{n+1} = (3I + 22\mathcal{X}_n^2 + 7\mathcal{X}_n^4)[\mathcal{X}_n(13I + 18\mathcal{X}_n^2 + \mathcal{X}_n^4)]^{-1},\ n = 0,1,2\ldots. \tag{6.10}$$

We denote the above method (6.10) by SM. One may refer to [155, 156, 154, 296] and the references therein for more information regarding such matrix iterations.

The primary goal of this research is to propose a numerically stable iterative scheme that is different from the Padé family or its reciprocals. Moreover, the matrix iteration should have global convergence behavior for computing the matrix sign function S. Therefore, we present a new iterative scheme in the upcoming section.

6.2 Iterative scheme

We construct and propose a new matrix iterative scheme to solve the matrix problem $\mathcal{X}^2 = I$. We have two obstacles to overcome in order to achieve this goal. The first one is the uniqueness of the

obtained matrix iteration, i.e., it should not coincide with any of the members of the general family by Padé (6.7) or their reciprocals. The second one is its global convergence behavior.

To address such issues, we use the three-step new iterative scheme in the scalar case outlined below:

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \left(\dfrac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right)\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \dfrac{f(z_n)}{f[y_n,z_n]+f'(x_n)}, \quad n=0,1,2,\ldots, \end{cases} \tag{6.11}$$

to solve $x^2 = 1$ in the scalar case, then naively extend the iteration to a matrix environment. Where $f[y_n, z_n] = \frac{f(y_n)-f(z_n)}{y_n-z_n}$ denotes the first-order finite-difference. Although the root solver (6.11) is not optimal in the context of the Kung-Traub conjecture, it is a perfect structure for extending to finding the matrix sign function.

Now, we provide the error equation of the proposed scheme in the scalar case, as follows:

Theorem 6.2.1. *For sufficiently differentiable function $f : \mathbb{D} \subseteq \mathbb{C} \to \mathbb{C}$ having simple zero $\eta \in \mathbb{D}$ with initial approximation x_0, in the open interval. The error equation of* (6.11) *can be obtained as*

$$e_{n+1} = \frac{1}{2}\left(c_2^3 - c_2c_3\right)e_n^4 + O(e_n^5),$$

where $c_i = \dfrac{1}{i!}\dfrac{f^i(\eta)}{f'(\eta)}$, $i \geq 2$ *and* $e_n = x_n - \eta$.

Proof. The steps for proving the order of convergence of this iterative method are straightforward via Taylor's expansion (e.g., refer to [268]). It is hence omitted. □

We are now ready to deduce the matrix iteration function for the proposed scheme (6.11). In a matrix environment, the iterative scheme (or their reciprocals) of the achieved iteration would take the form as follows:

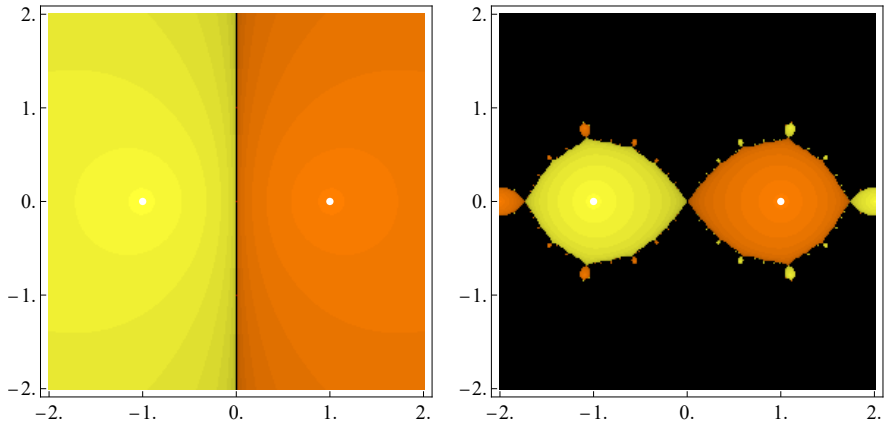
$$\mathcal{X}_{n+1} = [I + 19\mathcal{X}_n^2 + 39\mathcal{X}_n^4 + 5\mathcal{X}_n^6][6\mathcal{X}_n + 36\mathcal{X}_n^3 + 22\mathcal{X}_n^5]^{-1}, \ n = 0, 1, 2\ldots, \tag{6.12}$$

where $\mathcal{X}_0 = A$. The iterative approach (6.12) can be expressed in a more cost-effective manner as follows:

$$\mathcal{X}_{n+1} = [I + 19\mathcal{X}_n^2 + 39\mathcal{X}_n^4 + 5\mathcal{X}_n^6][\mathcal{X}_n(6I + 36\mathcal{X}_n^2 + 22\mathcal{X}_n^4)]^{-1}, \ n = 0, 1, 2\ldots, \tag{6.13}$$

requiring five products of matrix-matrix type in contrast to (6.12), which uses six matrix-matrix products per iteration. Eventually, we came up with a new matrix iterative scheme that is not related to (6.7) or its reciprocal family of iterations. This indicates that the proposed scheme (denoted by LM) has adequate ability and originality to support deeper study.

Before proceeding, it is important to note that the convergence should be global. To achieve this goal, we draw attraction basins of the proposed method (6.13) in order to get the solution of the scalar nonlinear equation $f(x) = x^2 - 1 = 0$ (see Iannazzo [171] for more details). We color each point in a rectangle $\mathbb{D} = [-2, 2] \times [-2, 2] \subset \mathbb{C}$ based on the simple zero at which the iterative technique starting at $z_0 \in \mathbb{D}$ converges, or mark the point as black if the procedure diverges. The stopping criterion used in this section for the method to converge is $|f(x_n)| \leq 10^{-2}$ with maximum iterations 25. As a result, the colors of the attraction basins for different methods can be distinguished. Two white points in the basins also indicate the precise location of the zeros.



(a) NM (b) NSM

(c) HM (d) LM

Figure 6.1: The attraction basins of different methods for solving $f(x) = x^2 - 1 = 0$.

Figure 6.1 depicts the attraction basins for the scheme (6.5), (6.8), (6.9), and (6.13), revealing its global convergence except for the method NSM, since it has a black shaded region. The higher convergence order of scheme (6.11) results in broader and lighter attraction basins.

The motivation for developing an iterative method is to achieve a higher-order globally convergent matrix iteration for determining the matrix sign function with the least computational effort, as well as to apply it to problems that frequently arise in computational and applied mathematics. It is essential to note that constructing a matrix iteration from a multi-point iterative scheme, such as (6.11), is not a straightforward task, as not every scalar iterative method can be transformed into a matrix case. Henceforth, we analyze certain crucial factors of matrix iteration for determining the matrix sign function, including the convergence rate, numerical stability, and global convergence behavior. These are described in the next two sections.

6.3 Convergence analysis

The aim of this section is to show analytically that the proposed iteration (6.13) is convergent for matrix cases under some assumptions. This is done in the following theorem.

Theorem 6.3.1. *Consider a set containing imaginary eigenvalues of the initial matrix $\mathcal{X}_0 = A \in \mathbb{C}^{m\times m}$ as empty. Then, the sequence of matrix iterations $\{\mathcal{X}_n\}_{n=0}^{\infty}$ from* (6.13) *converges to the sign of matrix, i.e., S.*

Proof. Suppose T is a rational function with which (6.13) is associated. As we know, for any complex matrix $\mathcal{X} \in \mathbb{C}^{m\times m}$, the Jordan canonical form can be written if there exists an invertible matrix V with $\mathcal{X} = VJV^{-1}$. Then, $T(\mathcal{X}) = VT(J)V^{-1}$. Using Eq. (6.13), the eigenvalue β of $\mathcal{X}_n$ is mapped into an eigenvalue $T(\beta)$ of $\mathcal{X}_{n+1}$. Because of the eigenvalue scalar relationship, we investigate how the complex plane is mapped into itself by $T(\beta)$. The rational function T, to be precise, must satisfy the following properties:

(i) Preservation of sign, i.e., $\text{sign}(T(x)) = \text{sign}(x),\ \forall\ x \in \mathbb{C}$, &

(ii) Convergence should be global, i.e, the sequence defined by $x_{n+1} = T(x_n)$, with initial guess $x_0 = x$ not located on the imaginary axis, converges to $\text{sign}(x)$.

To prove this, consider A has the following Jordan canonical form (for reference, see [154]):

$$V^{-1}AV = \Lambda = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}, \tag{6.14}$$

where V is a non-singular matrix and $C,\ N$ are Jordan blocks equivalent to eigenvalues in $\mathbb{C}^-$ and $\mathbb{C}^+$, respectively. The values located on the principal diagonal of blocks C and N are denoted by $\beta_1,\ \beta_2, \ldots, \beta_p$ and $\beta_{p+1}, \ldots, \beta_m$, respectively.

Adopting Eq. (6.14), we have

$$\operatorname{sign}(A) = V \begin{bmatrix} -I_p & 0 \\ 0 & I_{m-p} \end{bmatrix} V^{-1}. \tag{6.15}$$

Therefore, one can get

$$\begin{aligned} \operatorname{sign}(\Lambda) &= \operatorname{sign}(V^{-1}AV) \\ &= V^{-1}\operatorname{sign}(A)V, \\ &= \begin{pmatrix} \operatorname{sign}(\beta_1) & & & & & & \\ & \ddots & & & & & \\ & & \operatorname{sign}(\beta_p) & & & & \\ & & & \operatorname{sign}(\beta_{p+1}) & & & \\ & & & & \ddots & & \\ & & & & & \operatorname{sign}(\beta_m) \end{pmatrix}. \end{aligned} \tag{6.16}$$

From $\mathfrak{D}_0 = V^{-1}AV$, we define $\mathfrak{D}_n = V^{-1}\mathcal{X}_n V,\ n = 0, 1, 2, \ldots$, such that the sequence converges to $\operatorname{sign}(\Lambda)$. Then, from the proposed scheme (6.12), one can obtain

$$\mathfrak{D}_{n+1} = [(I + 19\mathfrak{D}_n^2 + 39\mathfrak{D}_n^4 + 5\mathfrak{D}_n^6)][6\mathfrak{D}_n + 36\mathfrak{D}_n^3 + 22\mathfrak{D}_n^5]^{-1}. \tag{6.17}$$

If $\mathfrak{D}_0$ has entries on the main diagonal, then all subsequent $\mathfrak{D}_n$ are diagonal as well, based on mathematical induction. It is important to note that the case where $\mathfrak{D}_0$ is not a diagonal matrix can be treated in the same way, as it is addressed in the proof later. To solve $f(x) = x^2 - 1 = 0$, express Eq. (6.17) in the form of an m uncoupled scalar iterative technique as follows:

$$d_{n+1}^i = \frac{1 + 19{d_n^i}^2 + 39{d_n^i}^4 + 5{d_n^i}^6}{6d_n^i + 36{d_n^i}^3 + 22{d_n^i}^5}, \tag{6.18}$$

where $d_n^i = (\mathfrak{D}_n)_{i,i}$ and $1 \leq i \leq m$. From Eqs. (6.17) and (6.18), we analyze the convergence behavior of $\{d_n^i\}$ to $\operatorname{sign}(\beta_i),\ \forall\ 1 \leq i \leq m$.

From Eq. (6.18) and using the assumption on eigenvalues, we have $\text{sign}(\beta_i) = s_i = \pm 1$. Thus,

$$\frac{d_{n+1}^i - s_i}{d_{n+1}^i + s_i} = \left(\frac{-s_i + d_n^i}{s_i + d_n^i}\right)^4 \frac{1 - 2s_i d_n^i + 5(d_n^i)^2}{1 + 2s_i d_n^i + 5(d_n^i)^2}. \tag{6.19}$$

It can be seen that the right-side expression of Eq. (6.19) is bounded for $i = 1, 2, \ldots, m$.

By adopting an appropriate initial matrix $\mathcal{X}_0 = A$, and as the factor $\left|\frac{-s_i + d_0^i}{s_i + d_0^i}\right| < 1$, we have

$$\lim_{n\to\infty} \left|\frac{-s_i + d_{n+1}^i}{s_i + d_{n+1}^i}\right| = 0, \tag{6.20}$$

and therefore, $\lim_{n\to\infty}(d_n^i) = s_i = \text{sign}(\beta_i)$. Hence, we conclude that $\lim_{n\to\infty} \mathfrak{D}_n = S = \text{sign}(\Lambda)$.

As briefly presented at the start of proof, we should study the scalar relationship among eigenvalues of the iterates for (6.13), if $\mathfrak{D}_0$ is not diagonal. The eigenvalues of $\mathcal{X}_n$ are mapped from the iterate n to $n+1$ by the following relation:

$$\beta_{n+1}^i = [(I + 19{\beta_n^i}^2 + 39{\beta_n^i}^4 + 5{\beta_n^i}^6)][6\beta_n^i + 36{\beta_n^i}^3 + 22{\beta_n^i}^5]^{-1}, \quad 1 \le i \le m. \tag{6.21}$$

Eq. (6.21) demonstrates that in the usual case, the eigenvalues are convergent to $s_i = \pm 1$, but in other cases, we employ the same approach as before, and we get

$$\lim_{n\to\infty} \left|\frac{\beta_{n+1}^i - s_i}{\beta_{n+1}^i + s_i}\right| = 0. \tag{6.22}$$

Finally, it is simple to conclude that

$$\lim_{n\to\infty} \mathcal{X}_n = V(\lim_{n\to\infty} \mathfrak{D}_n)V^{-1} = V\text{sign}(\Lambda)V^{-1} = \text{sign}(A). \tag{6.23}$$

This finishes the proof. □

We now look at the rate of convergence of (6.11), despite the fact that the preceding theorem dealt with convergence analysis.

6.3.1 Rate of convergence

In Theorem 6.3.1, we have studied the convergence of the proposed matrix iteration (6.13) to the matrix sign function S. We will now discuss the convergence rate in the following theorem.

Theorem 6.3.2. *Consider the same hypotheses as in Theorem 6.3.1. Then, the proposed method* (6.13) *achieves fourth-order convergence to* S.

Proof. This is clear that $\mathcal{X}_n$ is a rational operator of A for every $n \geq 0$ and hence commutes with S, just like A. In addition, the following relationships are available in the research article [335]:

$$S^2 = I,\ S^{-1} = S,\ S^{2j} = I \text{ and } S^{2j+1} = S,\ j \geq 1. \tag{6.24}$$

Applying the substitution $C_n = (6\mathcal{X}_n + 36\mathcal{X}_n^3 + 22\mathcal{X}_n^5)$, we obtain

$$\begin{aligned}
\mathcal{X}_{n+1} - S &= (I + 19\mathcal{X}_n^2 + 39\mathcal{X}_n^4 + 5\mathcal{X}_n^6)C_n^{-1} - S,\\
&= (I + 19\mathcal{X}_n^2 + 39\mathcal{X}_n^4 + 5\mathcal{X}_n^6 - SC_n)C_n^{-1},\\
&= (I + 19\mathcal{X}_n^2 + 39\mathcal{X}_n^4 + 5\mathcal{X}_n^6 - 6S\mathcal{X}_n - 36S\mathcal{X}_n^3 - 22S\mathcal{X}_n^5)C_n^{-1},\\
&= (I - 6S\mathcal{X}_n + 19\mathcal{X}_n^2 - 36S\mathcal{X}_n^3 + 39\mathcal{X}_n^4 - 22S\mathcal{X}_n^5 + 5\mathcal{X}_n^6)C_n^{-1},\\
&= (S^6 - 6S^5\mathcal{X}_n + 19S^4\mathcal{X}_n^2 - 36S^3\mathcal{X}_n^3 + 39S^2\mathcal{X}_n^4 - 22S\mathcal{X}_n^5 + 5\mathcal{X}_n^6)C_n^{-1},\\
&= ((S - \mathcal{X}_n)^6 + 4\mathcal{X}_n^2(S^4 - 4S^3\mathcal{X}_n + 6S^2\mathcal{X}_n^2 - 4S\mathcal{X}_n^3 + \mathcal{X}_n^4))C_n^{-1},\\
&= (\mathcal{X}_n - S)^4(I - 2S\mathcal{X}_n + 5\mathcal{X}_n^2)C_n^{-1}.
\end{aligned} \tag{6.25}$$

Now, applying the matrix norm on both sides of Eq. (6.25), we obtain

$$||\mathcal{X}_{n+1} - S|| \leq ||C_n^{-1}||\ ||I - 2S\mathcal{X}_n + 5\mathcal{X}_n^2||\ ||\mathcal{X}_n - S||^4. \tag{6.26}$$

This demonstrates that the new method (6.13) has a fourth-order convergence. This step finishes the proof. □

6.4 Stability analysis

The stability of (6.13) for finding S in a neighborhood of the solution is explored in this section. We study how a small perturbation at the n^{th} iteration is exaggerated or dissipated as the iterations progress.

Theorem 6.4.1. *The sequence* $\{\mathcal{X}_n\}_{n=0}^{n=\infty}$ *formed by* (6.13) *is asymptotically stable under the identical assumptions as in Theorem 6.3.2.*

Proof. If the initial condition is $\mathcal{X}_0 = A$, then from (6.13) the obtained iterate is also a function of

A and hence satisfies the commutative property on A. Consider δ_n to be the numerical perturbation generated at the n^{th} iterate in (6.13), we have

$$\tilde{\mathcal{X}}_n = \mathcal{X}_n + \delta_n. \tag{6.27}$$

We now undertake error analysis of first-order here, which means we use approximations explicitly $(\delta_n)^l \approx 0$, as $(\delta_n)^l,\ l \geq 2$ is closer to zero (matrix). If $(\delta_n)^l$ is small enough, this usual approach is appropriate. For large enough n, we have (assuming $\mathcal{X}_n \approx \text{sign}(A) = S$).

$$\begin{aligned}
\tilde{\mathcal{X}}_{n+1} &= (I + 19\tilde{\mathcal{X}}_n^2 + 39\tilde{\mathcal{X}}_n^4 + 5\tilde{\mathcal{X}}_n^6)[6\tilde{\mathcal{X}}_n + 36\tilde{\mathcal{X}}_n^3 + 22\tilde{\mathcal{X}}_n^5]^{-1}, \\
&= (I + 19(\mathcal{X}_n + \delta_n)^2 + 39(\mathcal{X}_n + \delta_n)^4 + 5(\mathcal{X}_n + \delta_n)^6)[6(\mathcal{X}_n + \delta_n) + 36(\mathcal{X}_n + \delta_n)^3 \\
&\quad + 22(\mathcal{X}_n + \delta_n)^5]^{-1}, \\
&\approx (64I + 112\delta_n S + 112S\delta_n)(64S + 144\delta_n + 80S\delta_n S)^{-1}, \\
&\approx \left(I + \frac{7}{4}\delta_n S + \frac{7}{4}S\delta_n\right)\left(S + \frac{9}{4}\delta_n - \frac{5}{4}S\delta_n S\right), \\
&\approx \frac{1}{2}(2S - \delta_n + S\delta_n S),
\end{aligned} \tag{6.28}$$

wherein the following identity was used (for any nonsingular matrix $\mathcal{B}$ and the matrix $\mathcal{C}$, one can refer [151])

$$(\mathcal{B} + \mathcal{C})^{-1} = \mathcal{B}^{-1} - \mathcal{B}^{-1}\mathcal{C}\mathcal{B}^{-1} + O(\mathcal{C}^2).$$

Note that after some algebraic manipulations and using $\delta_{n+1} = \tilde{\mathcal{X}}_{n+1} - \mathcal{X}_{n+1}$, we have

$$\delta_{n+1} \approx \frac{1}{2}\delta_n - \frac{1}{2}S\delta_n S. \tag{6.29}$$

We used the facts on the matrix sign function, i.e., $S^2 = I$ and $S^{-1} = S$. We can now conclude that the perturbation at the $(n+1)^{\text{th}}$ iterate is bounded, i.e., $\delta_{n+1} \approx \frac{1}{2^{n+1}}(\delta_0 - S\delta_0 S)$. Therefore, the sequence $\{\mathcal{X}_n\}_{n=0}^{n=\infty}$ generated by (6.13) is asymptotically stable. This ends the proof. □

6.5 Efficiency challenge

The computational efficiency index (EI) of an iteration method is estimated according to Soleymani et al. [335], i.e,

$$EI_M = p^{\frac{1}{C}}. \tag{6.30}$$

In this context, p denotes the convergence order, and C represents the total computational cost per iteration required to execute the iterative method for locating S.

To fairly compare the EI of various iterative schemes, we consider that one matrix-matrix multiplication costs $m^{2373/1000}$ and one matrix inverse costs $\frac{2m^3}{3}$. Noting that matrix multiplications in recent programming packages cost on the order of 2.373 (refer to article [228]). Furthermore, we ignore the cost of matrix addition and subtraction for each iteration.

We have the following efficiency index for the methods NM, ANM, HM, and LM (while the cost of adopting norm computation per iteration is being ignored):

$$\begin{aligned} EI_{\mathrm{NM}} &= 2^{\frac{1}{m^{2373/1000}+\frac{2m^3}{3}}}, \quad EI_{\mathrm{ANM}} = (\hat{\eta})^{\frac{1}{m^{2373/1000}+\frac{2m^3}{3}}}, \\ EI_{\mathrm{HM}} &= 3^{\frac{1}{3m^{2373/1000}+\frac{2m^3}{3}}}, \quad EI_{\mathrm{LM}} = 4^{\frac{1}{5m^{2373/1000}+\frac{2m^3}{3}}}, \end{aligned} \tag{6.31}$$

where m is the matrix size and $\hat{\eta} = \sqrt{3}+1$. It is worth mentioning that NM and ANM only

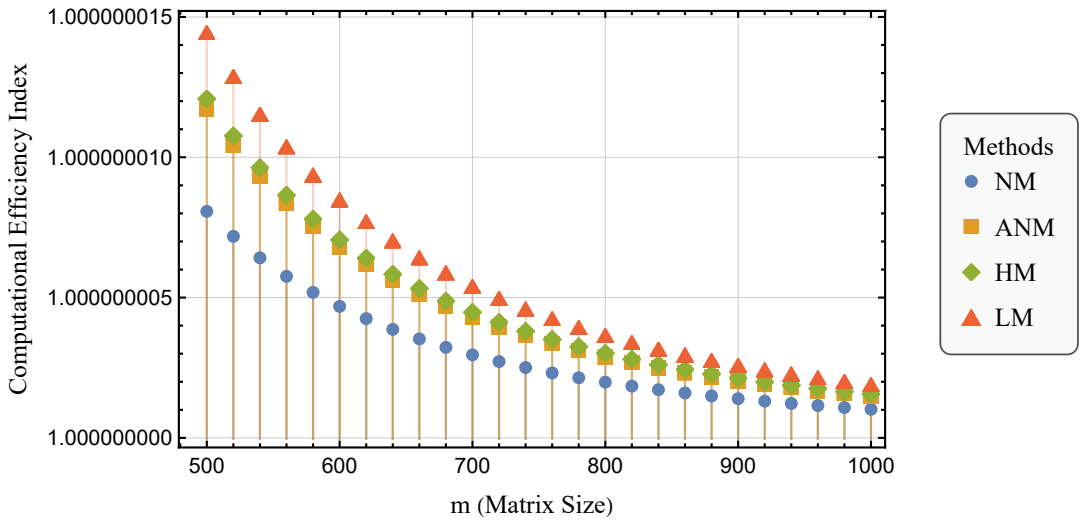


Figure 6.2: The efficiency indices comparison of NM, ANM, HM, and LM.

require one matrix multiplication per iteration to achieve a safer stopping criterion, as numerical evidence will subsequently demonstrate. The comparison results (assuming a unit cost per flop) are presented in Figure 6.2, which shows that the proposed scheme is more efficient in terms of the efficiency index.

6.6 Numerical tests

For various matrix iterations, the comparison results based on iteration number and residual norm are presented here. If the iteration $\mathcal{X}_n$ has a largest eigenvalue, i.e., if $||\mathcal{X}_n|| >> 1$, the convergence may be slow. As a result, we can accelerate the convergence of the proposed iteration by scaling (for reference, see [211]). The scaling parameter ν_n is introduced for this purpose as follows:

$$\nu_n = \begin{cases} \sqrt{\dfrac{||\mathcal{X}_n^{-1}||}{||\mathcal{X}_n||}}, & \text{(Norm scaling)} \\ \sqrt{\dfrac{\rho(\mathcal{X}_n^{-1})}{\rho(\mathcal{X}_n)}}, & \text{(Spectral scaling)} \\ |\det(\mathcal{X}_n)|^{-\frac{1}{m}}, & \text{(Determinant scaling)} \end{cases} \tag{6.32}$$

with scaling factor, the scheme can be written as

$$\begin{cases} \mathcal{X}_0 & = A, \\ \nu_n & = \text{the scaling parameter computed by (6.32)}, \\ \mathcal{X}_{n+1} & = [I + 19{\nu_n}^2\mathcal{X}_n^2 + 39{\nu_n}^4\mathcal{X}_n^4 + 5{\nu_n}^6\mathcal{X}_n^6][\nu_n\mathcal{X}_n(6I + 36{\nu_n}^2\mathcal{X}_n^2 + 22{\nu_n}^4\mathcal{X}_n^4)]^{-1}, \end{cases} \tag{6.33}$$

where $\lim_{n\to\infty} \nu_n = 1$ and $\lim_{n\to\infty} \mathcal{X}_n = S$. However, due to high cost in some cases, the computation of scaling parameter ν_n for any iteration method is not studied in depth. The stopping termination considered for this work is

$$||\mathcal{X}_n^2 - I||_* \leq \epsilon, \tag{6.34}$$

where ϵ and $||.||_*$ denote matrices (see reference [327] for tolerance and suitable matrix norm, respectively). The l_∞ and l_2 are considered for real and complex input matrices.

The iteration number and computational e-time (CPU) for various methods are compared. We only employ approaches that have global convergence behavior for the sake of comparison. The methods being compared are NM (6.5), ANM (6.6), HM (6.9), LM (6.13), and ALM (spectral scaling (6.33)). Additionally, comparisons with methods exhibiting local convergence behavior (such as NSM) or methods from different categories (like computing the Cauchy integral (6.2)) are excluded.

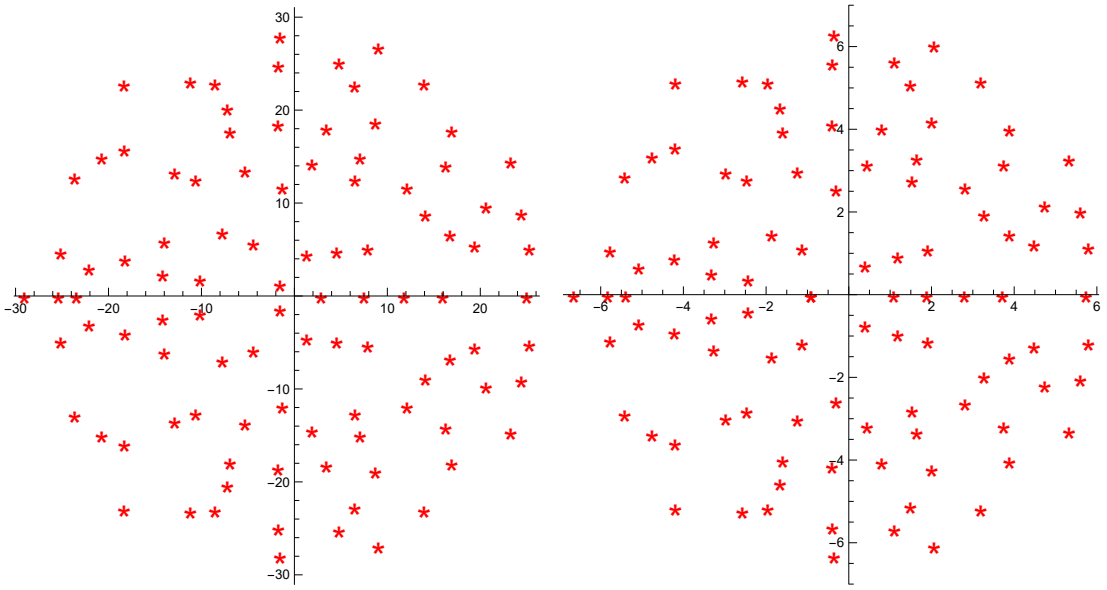
The following examples are tested to illustrate the performance of the proposed method:

Experiment 6.6.1. *In this example, we discuss below the study of eigenvalue clustering using a*

randomly generated real matrix of size 100×100*:*

```
SeedRandom[1]; A = RandomReal[{-5, 5}, {100, 100}]; tolerance = 10^-3;
n = 0; max = 50;
```

The results are presented in Figures 6.3–6.4, by applying the infinity norm with tolerance $\epsilon = 10^{-3}$ *in double precision. It is observed that, at the initial stage, the eigenvalues of* $\mathcal{X}_0$ *are widely scattered. As the proposed scheme* (6.13) *is applied, the eigenvalues of* $\mathcal{X}_i$*,* $i = 2, \ldots, 6$*, progressively cluster around* -1 *or* 1*. This behavior indicates that the matrix sequence* $\{\mathcal{X}_n\}_{n=0}^{\infty}$ *converges to* $\operatorname{sign}(A)$*. Consequently, the theoretical results of Theorem 6.3.1 are verified.*



(a) $\mathcal{X}_0$ (Initial Iteration) (b) $\mathcal{X}_1$ (First Iteration)

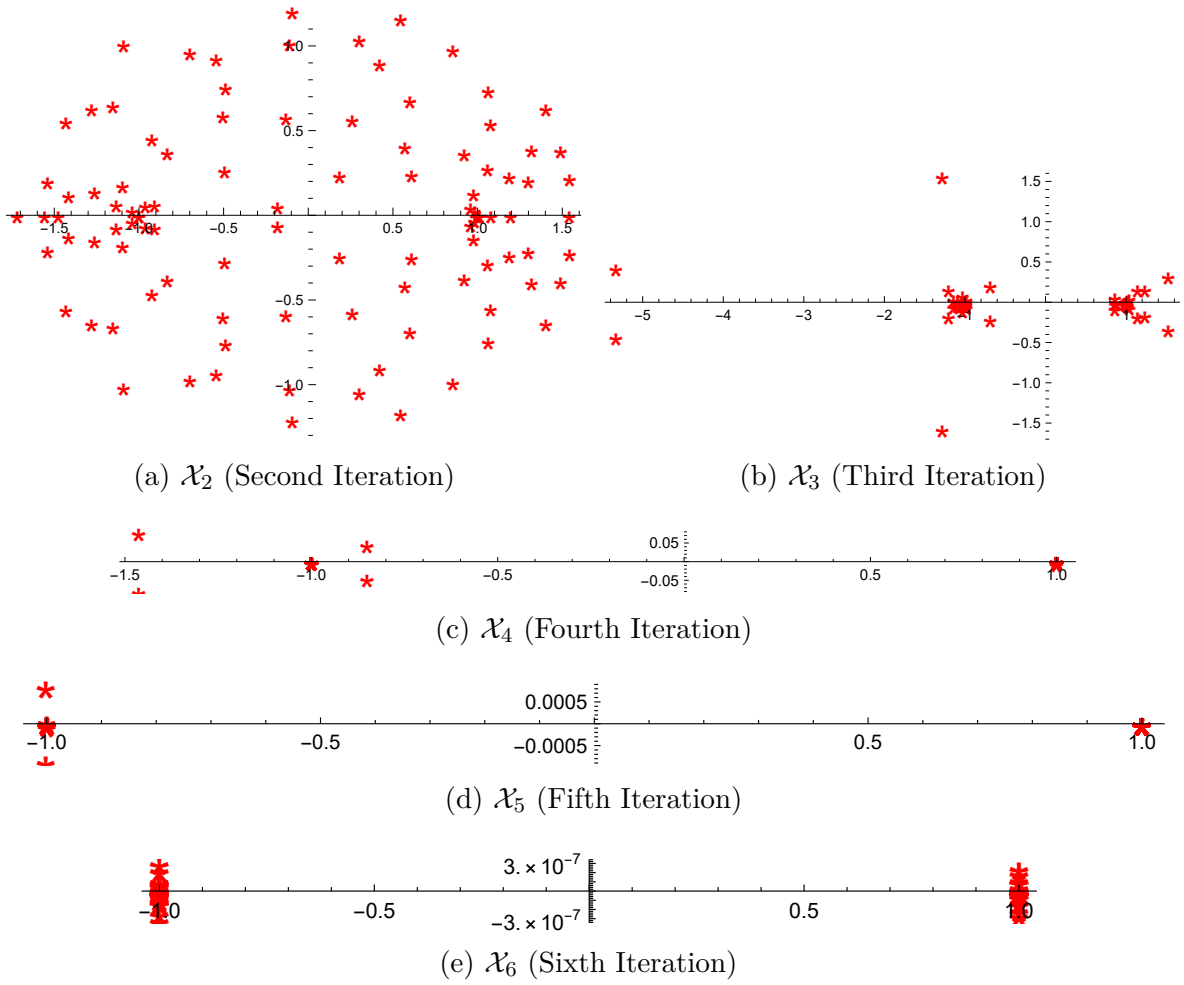
Figure 6.3: Clustering of eigenvalues of real matrix 100×100 in Experiment 6.6.1.

Experiment 6.6.2. *In this example, we discuss the study of the matrix sign function using a random complex matrix* 1000×1000 *for different iterative methods by employing the following input*

```
SeedRandom[12];
A = RandomComplex[{-10 - 10*Sqrt[-1], 10 + 10*Sqrt[-1]}, {1000, 1000}];
tolerance = 10^-4; n = 0; max = 50;
```

using $\epsilon = 10^{-4}$ *with* l_2 *norm, since the input is complex matrix with* $\mathcal{X}_0 = A$ *as an initial condition.*

Figure 6.5 presents the comparison results in terms of iteration number and absolute error, illustrating the steady and consistent behavior of LM in computing the matrix sign function.



(a) $\mathcal{X}_2$ (Second Iteration)

(b) $\mathcal{X}_3$ (Third Iteration)

(c) $\mathcal{X}_4$ (Fourth Iteration)

(d) $\mathcal{X}_5$ (Fifth Iteration)

(e) $\mathcal{X}_6$ (Sixth Iteration)

Figure 6.4: Clustering of eigenvalues of real matrix 100×100 in Experiment 6.6.1

Experiment 6.6.3. *Next, we discuss the study of matrix sign function using a random complex matrix* 1500×1500 *for different iterative methods by employing the following input*

```
SeedRandom[125];
A = RandomComplex[{-15 - 15*Sqrt[-1], 15 + 15*Sqrt[-1]}, {1500, 1500}];
tolerance = 10^-4; n = 0; max = 50;
```

using $\epsilon = 10^{-4}$ *with* l_2 *norm, since the input is a complex matrix with initial condition* $\mathcal{X}_0 = A$.

Figure 6.6 presents the comparison results in terms of iteration number and absolute error, illustrating the steady and consistent behavior of LM in computing the sign of a matrix.

Experiment 6.6.4. *In this experiment, we study the convergence history of different methods for a randomly* 2000×2000 *real matrix as follows:*

```
SeedRandom[979];
```

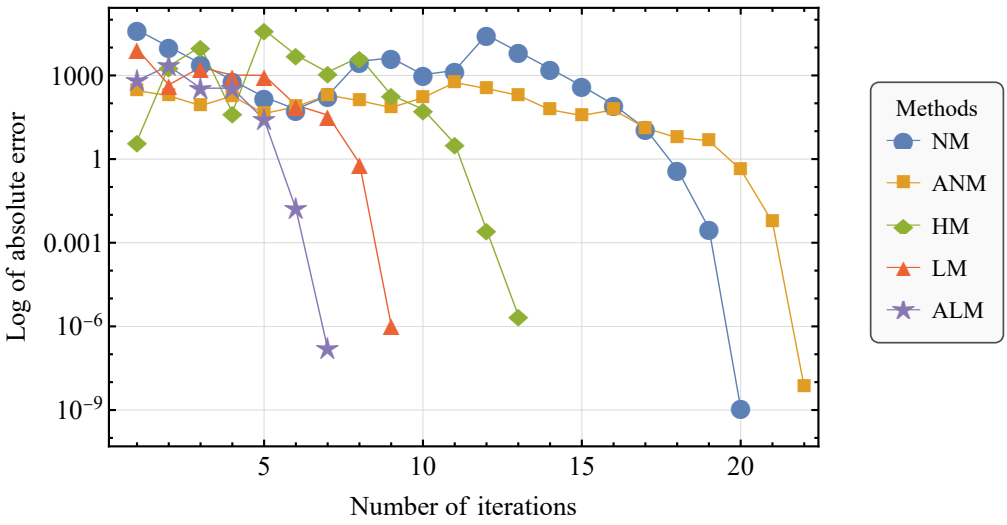


Figure 6.5: The convergence history of different methods in Experiment 6.6.2.

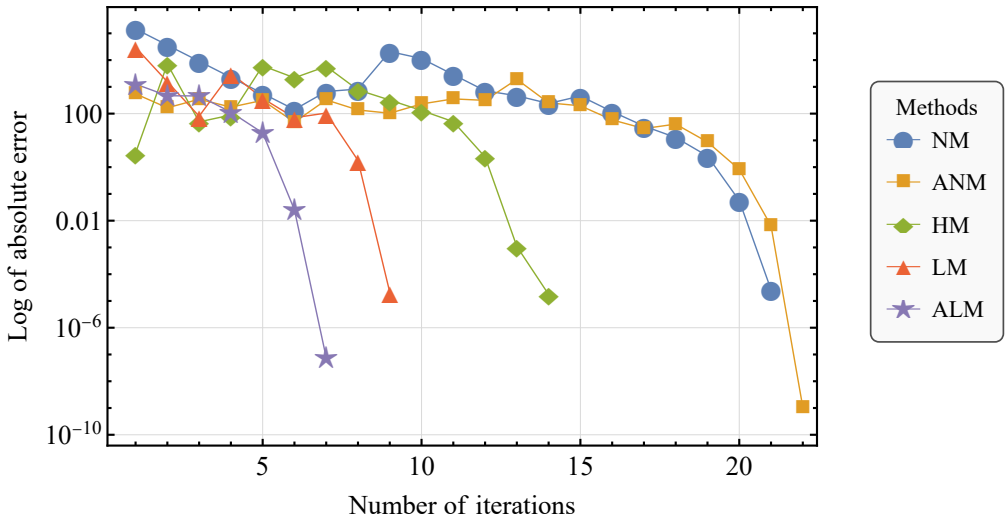


Figure 6.6: The convergence history of different methods in Experiment 6.6.3.

```
A = RandomReal[{-6, 6}, {2000, 2000}]; tolerance = 10^-4; n = 0; max = 50;
```

using $\epsilon = 10^{-4}$ *with* l_∞ *norm, since the input is a real matrix with initial condition* $\mathcal{X}_0 = A$.

Figure 6.7 shows the comparison results regarding the iteration number and absolute error, which demonstrate the steady and consistent behavior of LM in computing the matrix sign function.

In the next two experiments, we introduce a new method, SM, for comparison with the proposed methods, which have the same convergence order as given in equation (6.10).

Experiment 6.6.5. *Here, we compare the convergence histories of several methods for a set of ten*

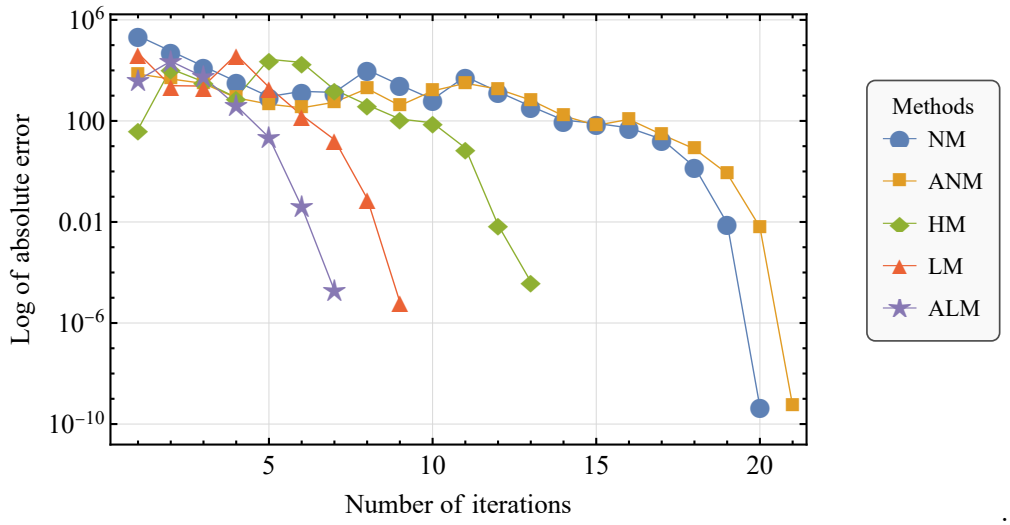


Figure 6.7: The convergence history of different methods in Experiment 6.6.4.

randomly chosen complex matrices

```
SeedRandom[15]; number = 10;
Table[A[l] = RandomComplex[{-5-5Sqrt[-1],5+5Sqrt[-1]}, {50*l, 50*l}];,
{l, number}]; tol = 10^{-4}; n = 0; max = 50;
```

The numerical results are given in Tables 6.1–6.2 for input matrices of different sizes depending on the required iteration number and the e-time (CPU-time) using l_2-norm, since the matrices are complex. The results show that the average iteration number and CPU time improve when the new methods are employed. Table 6.2 has units in seconds. As a result, the proposed iterative methods exhibit robust and consistent behavior while computing the matrix sign function.

Experiment 6.6.6. *Lastly, we compare the convergence histories of several methods for a set of ten randomly chosen real matrices:*

```
SeedRandom[579]; number = 10;
Table[A[l] = RandomReal[{-9, 9}, {50*l, 50*l}];, {l, number}]; tol = 10^-5;
n = 0; max = 50;
```

The numerical results are reported in Tables 6.3–6.4 for input matrices of various sizes, in terms of the required number of iterations and the elapsed CPU time (e-time). The ℓ_∞-norm is used since the matrices are real. The results indicate that both the average iteration count and the CPU time are reduced when the proposed methods are applied, as shown in the last row of each table. The e-time in Table 6.4 is measured in seconds. The proposed iterative solver for the matrix sign function has

Table 6.1: Comparisons in terms of iteration number using various random matrices of Experiment 6.6.5

Matrix	NM	ANM	HM	SM	LM	ALM
$A_{50\times50}$	13	12	9	6	6	5
$A_{100\times100}$	16	16	10	7	7	5
$A_{150\times150}$	15	13	10	7	7	5
$A_{200\times200}$	15	14	10	7	7	6
$A_{250\times250}$	14	15	9	7	7	6
$A_{300\times300}$	15	14	10	7	7	5
$A_{350\times350}$	16	16	10	7	7	7
$A_{400\times400}$	20	17	13	9	9	6
$A_{450\times450}$	16	17	10	7	7	5
$A_{500\times500}$	18	17	12	8	8	6
Mean	15.8	15.1	10.3	7.2	7.2	**5.6**

Table 6.2: Comparisons in terms of CPU-time using various random matrices of Experiment 6.6.5

Matrix	NM	ANM	HM	SM	LM	ALM
$A_{50\times50}$	0.07	0.07	0.04	0.03	0.03	0.04
$A_{100\times100}$	0.18	0.30	0.16	0.13	0.11	0.15
$A_{150\times150}$	0.35	0.51	0.36	0.37	0.31	0.37
$A_{200\times200}$	0.77	1.07	0.67	0.63	0.62	0.75
$A_{250\times250}$	0.95	1.54	0.98	0.91	0.96	1.47
$A_{300\times300}$	1.55	2.79	2.35	1.46	1.40	1.68
$A_{350\times350}$	2.18	3.66	2.14	2.11	1.90	3.62
$A_{400\times400}$	3.44	5.18	4.42	3.87	3.65	3.85
$A_{450\times450}$	3.68	7.22	4.61	4.35	3.72	4.53
$A_{500\times500}$	8.12	8.91	7.16	6.51	6.31	8.09
Mean	2.13	3.13	2.29	2.04	**1.90**	2.46

been computed.

Numerical evidence indicates that the proposed scheme provides a reasonable understanding and effective approach for iteratively calculating the matrix sign function.

Table 6.3: Comparisons in terms of iteration number using a various random matrices of Experiment 6.6.6

Matrix	NM	ANM	HM	SM	LM	ALM
$A_{50\times50}$	16	12	10	8	7	6
$A_{100\times100}$	15	13	10	7	7	5
$A_{150\times150}$	14	13	9	6	6	5
$A_{200\times200}$	15	14	10	7	7	6
$A_{250\times250}$	16	14	10	7	8	6
$A_{300\times300}$	16	17	11	7	7	5
$A_{350\times350}$	16	15	11	8	8	5
$A_{400\times400}$	17	17	11	8	8	6
$A_{450\times450}$	18	18	12	8	8	5
$A_{500\times500}$	20	19	13	7	8	5
Mean	16.3	15.2	10.7	7.3	7.4	**5.4**

Table 6.4: Comparisons in terms of CPU-time using various random matrices of Experiment 6.6.6

Matrix	NM	ANM	HM	SM	LM	ALM
$A_{50\times50}$	0.06	0.07	0.05	0.04	0.02	0.03
$A_{100\times100}$	0.14	0.18	0.11	0.09	0.06	0.09
$A_{150\times150}$	0.50	0.35	0.20	0.14	0.13	0.19
$A_{200\times200}$	0.58	0.63	0.40	0.37	0.27	0.42
$A_{250\times250}$	0.77	1.07	0.67	0.54	0.55	0.67
$A_{300\times300}$	1.24	1.92	1.22	1.27	0.75	0.79
$A_{350\times350}$	1.73	2.31	1.54	1.85	1.12	1.36
$A_{400\times400}$	2.20	3.39	2.56	1.94	1.87	2.04
$A_{450\times450}$	3.06	4.82	3.73	2.14	2.53	2.32
$A_{500\times500}$	3.97	6.49	5.39	2.55	3.48	3.07
Mean	1.43	2.12	1.59	1.09	**1.08**	1.10

6.7 Concluding remarks

The matrix function theory and computation are vital in many aspects of numerical linear algebra and scientific computing. It serves as an efficient technology in modern computational algebra, enabling the solution of current challenges in control theory. A matrix function can be computed

from a variety of approaches, but the three most common are polynomial interpolation, the Jordan canonical form, and the Cauchy integral. We focus on a new iterative approach for such an objective in this study. Therefore, a higher-order iterative solver for scalar nonlinear equations was used to develop a new method for determining the matrix sign with no pure imaginary eigenvalues. The proposed matrix iteration converges globally in the complex plane, as proven by attraction basins with a fourth-order convergence rate. Finally, numerical tests were used to demonstrate the superiority and applicability of the presented method. Furthermore, future work can focus on extending the matrix sign function to other related matrix functions and calculating the polar decomposition based on the application of the new schemes.

In the next chapter, we propose two novel higher-order iterative schemes for computing the matrix sign function. Furthermore, we extend their application to compute non-trivial solutions of the Yang-Baxter-like equation.

Chapter 7

Numerically stable higher-order matrix iterations for computing matrix sign function

In this chapter, the primary objective is to develop two novel iterative schemes for computing the sign of a matrix that has no purely imaginary eigenvalues. A detailed convergence analysis and the asymptotic stability of the proposed methods are discussed. It is shown that the new schemes converge globally, as demonstrated through attraction basins. In addition, the obtained results are extended to compute nontrivial solutions of the Yang–Baxter-like equation, provided that the given matrix has no eigenvalues on the imaginary axis. To illustrate the effectiveness of the theoretical developments, a class of numerical examples for matrices of various dimensions, including cases with eigenvalue clustering, is presented.

7.1 Introduction

The quadratic matrix equation

$$A\mathcal{X}A = \mathcal{X}A\mathcal{X}, \tag{7.1}$$

is known as a Yang–Baxter-like equation (or YBM equation), where A and $\mathcal{X}$ are two square matrices of the same order; see Ding and Rhee [112] and Dong and Ding [116] for reference. The YBM equation is named after the classical Yang–Baxter equation, which was first introduced by Yang [369] for the delta-function Fermi gas in 1967 and later by Baxter [43] for the eight-vertex model in 1972. They independently showed that certain rational matrix functions satisfy nonlinear

The contents of this chapter are published in Mathematical Methods in the Applied Sciences, Vol. 46, pp. 8596–8617, 2023 (SCIE, I.F.: 1.8)

matrix equations arising in their respective research problems. Since then, physicists [152] have constructed additional solutions to the Yang–Baxter equation in various forms.

Several approaches have been proposed for solving the YBM equation, such as the set-theoretic method [35], which is based on the idea [118] that the vector-space problem can be reduced to its basis as a finite set. The YBM equation has numerous applications, including quantum group theory and knot theory (see [370]). Motivated by these applications, many researchers [178] have sought to develop efficient methods for computing its solutions.

It should be emphasized that obtaining all solutions of equation (7.1) is a challenging task. One explanation can be drawn from algebraic geometry, which suggests that solving the equation for a generic matrix A of order $m \times m$ is equivalent to solving a polynomial system consisting of m^2 quadratic equations in m^2 unknowns. The first attempt to solve equation (7.1) from the perspective of matrix theory was made in 2012 by Ding and Rhee [111], where the Brouwer's fixed point theorem was employed to compute a nontrivial solution for the case in which the input matrix A is invertible.

There are two trivial solutions of the YBM equation, namely, $\mathcal{X} = 0$ and $\mathcal{X} = A$. Direct approaches, such as those presented in [112], can be used to obtain nontrivial solutions. However, these methods generally require substantial computational time and memory. Consequently, numerical iterative methods are preferable for computing such solutions. If an efficient iterative approach can be constructed for solving the equation (7.1) and applied to a larger class of matrices A, it would be highly effective. With this motivation, the primary objective is to develop a new iterative method for solving equation (7.1).

This study focuses on a particular case generated by using a MSF (see [162]). It is known that, in the case of a scalar, the sign function is defined for each $\zeta \in \mathbb{C}$ that does not lie on the imaginary axis by Eq. (1.34). The matrix sign function holds the primary theoretical and computational connections with the polar decomposition (see [156]), the square root of a matrix, and, more generally, the p^{th} roots of a matrix. For instance, a large class of iterative methods for computing the matrix square root can be derived from the corresponding iterations for the matrix sign function, which motivates the study and development of new iterative schemes for evaluating the matrix sign function.

We assume that the spectrum of a matrix $A \in \mathbb{C}^{m\times m}$ does not intersect the imaginary axis. Under this assumption, the matrix sign function is uniquely defined (with A being non-singular). A concise formulation of the matrix sign decomposition, due to Higham [155] and Kenney [210, 211], is given by $A = SN = A(A^2)^{-1/2}(A^2)^{1/2}$, where $S = A(A^2)^{-1/2}$ denotes the matrix sign function and

$(\cdot)^{1/2}$ represents the principal matrix square root. Since A has no eigenvalues on the negative real axis, there exists a unique primary square root $\bar{\mathcal{X}}$ of A whose eigenvalues lie in the open right-half plane (see [154] for details). This square root is referred to as the principal square root of A and is denoted by $\bar{\mathcal{X}} = A^{1/2}$. Further applications and properties of the matrix sign function can be found in [154, 155, 171, 209, 246, 315].

Recently, Soleymani and Kumar [337] derived a new fourth-order matrix iterative scheme for computing the matrix sign function, expressed as

$$\mathcal{X}_{n+1} = (3I + 22\mathcal{X}_n^2 + 7\mathcal{X}_n^4)\Big[\mathcal{X}_n(13I + 18\mathcal{X}_n^2 + \mathcal{X}_n^4)\Big]^{-1}, \quad n \in \hat{\mathbb{N}}_0, \tag{7.2}$$

provided that the initial matrix has no purely imaginary eigenvalues. This iteration requires four matrix–matrix multiplications (MMM) and one matrix inversion per step.

Furthermore, Delshad and Lotfi [27] presented a constructive approach for computing the matrix sign function based on a stable variant of the Kung–Traub method with fourth-order convergence, given by

$$\mathcal{X}_{n+1} = (I + 3\mathcal{X}_n^2 + 23\mathcal{X}_n^4 + 5\mathcal{X}_n^6)\Big[2\mathcal{X}_n + 12\mathcal{X}_n^3 + 18\mathcal{X}_n^5\Big]^{-1}, \quad n \in \hat{\mathbb{N}}_0, \tag{7.3}$$

which requires five matrix–matrix multiplications and one matrix inversion per iteration. More recently, Zaka Ullah and Alaslani [378] investigated a higher-order iterative scheme for computing the matrix square root and further analyzed it through the framework of the matrix sign function.

Very recently, Soleymani et al. [336] proposed an iterative scheme for computing the matrix sign function S with sixth-order convergence, given by

$$\begin{aligned}\mathcal{X}_{n+1} = \mathcal{X}_n\Big(&(3 - 20a + 28a^2)I + 4(-13 + 40a + 8a^2)\mathcal{X}_n^2 + 2(93 + 108a - 68a^2)\mathcal{X}_n^4 \\ &+ 4(87 - 80a + 16a^2)\mathcal{X}_n^6 + 3(3 - 2a)^2\mathcal{X}_n^8\Big)\Big[(1 - 2a)^2 I + 4(-3 + 16a^2)\mathcal{X}_n^2 \\ &- 2(9 - 172a + 44a^2)\mathcal{X}_n^4 - 4(-97 + 40a + 8a^2)\mathcal{X}_n^6 + (-3 + 2a)(-51 + 26a)\mathcal{X}_n^8\Big]^{-1},\end{aligned} \tag{7.4}$$

where a is a free parameter and $n \in \hat{\mathbb{N}}_0$. In particular, for $a = \frac{3}{4}$, the method reduces to the following matrix iterative scheme (denoted by SM):

$$\begin{aligned}\mathcal{X}_{n+1} = \mathcal{X}_n(I + 3\mathcal{X}_n^2)(5I + 3\mathcal{X}_n^2)(3I + 58\mathcal{X}_n^2 + 3\mathcal{X}_n^4)\Big[(I + 7\mathcal{X}_n^2)(I + 89\mathcal{X}_n^2 + 139\mathcal{X}_n^4 \\ + 27\mathcal{X}_n^6)\Big]^{-1}.\end{aligned} \tag{7.5}$$

For further discussion of matrix iterative schemes of this type, see [60, 154, 155, 156, 296] and the references therein.

The primary objective of this research is to extend scalar iterative methods to the matrix setting for computing the matrix sign function with global convergence properties. Accordingly, we propose new iterative schemes with convergence orders six and eight that do not belong to the Padé family, nor to the iterative family introduced by Dong and Wang [115], or their reciprocals. The convergence behavior and stability of the proposed methods are established through analytical study.

7.2 New iterative methods

In this section, we construct new iterative schemes for computing the matrix sign function of A, where A has no eigenvalues on the imaginary axis. The first objective is to derive matrix iterations that do not coincide with any member of the Padé family (6.7) or their reciprocals. The second one is to ensure their global convergence behavior. Moreover, in designing efficient iterative methods for the matrix sign function, the overarching goal is to achieve higher orders of convergence while maintaining minimal computational cost.

In order to derive a novel and efficient family of iterative methods for solving the nonlinear matrix equation $\mathcal{X}^2 = I$, we propose the following scalar nonlinear solvers:

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \left(\dfrac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right)\dfrac{f(x_n)}{f'(x_n)}, \\ w_n = z_n - \dfrac{f(z_n)}{f[y_n,z_n]+f'(x_n)}, \\ x_{n+1} = w_n - \dfrac{f(w_n)}{f[w_n,y_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{cases} \tag{7.6}$$

and

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \left(\dfrac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right)\dfrac{f(x_n)}{f'(x_n)}, \\ w_n = z_n - \dfrac{f(z_n)}{f[y_n,z_n]+f'(x_n)}, \\ x_{n+1} = w_n - \dfrac{f(w_n)}{f[w_n,z_n]}, \quad n \in \hat{\mathbb{N}}_0, \end{cases} \tag{7.7}$$

where $f[y_n, z_n] = \frac{f(y_n)-f(z_n)}{y_n-z_n}$ and $f[w_n, z_n] = \frac{f(w_n)-f(z_n)}{w_n-z_n}$ denote the first-order finite differences.

Theorem 7.2.1. *Let $f : \mathbb{D} \subseteq \mathbb{C} \to \mathbb{C}$ be a sufficiently differentiable function having a simple zero η in the open interval $\mathbb{D}$. If the initial guess x_0 is close enough to η, then the convergence order of the scheme* (7.6) *is six and satisfies the following error equation:*

$$e_{n+1} = \frac{1}{2}\left(c_2^5 - c_2^3 c_3\right) e_n^6 + O(e_n^7), \tag{7.8}$$

where $c_k = \dfrac{1}{k!}\dfrac{f^{(k)}(\eta)}{f'(\eta)},\ k \geq 2.$

Proof. Let $e_n = x_n - \eta$ denote the error at the n^{th} iteration. We can write the Taylor's expansion of $f(x_n)$, and $f'(x_n)$ about exact zero η at n^{th} iteration as follows:

$$f(x_n) = f'(\eta)(e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + c_6 e_n^6 + c_7 e_n^7 + c_8 e_n^8 + O(e_n^9)), \tag{7.9}$$

and

$$f'(x_n) = f'(\eta)(1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + 6c_5 e_n^5 + 7c_6 e_n^6 + 8c_8 e_n^7 + O(e_n^8)). \tag{7.10}$$

Now, we divide equation (7.9) by (7.10) to get the following expansion of iteration function:

$$\begin{aligned}\frac{f(x_n)}{f'(x_n)} = {} & e_n - c_2 e_n^2 + 2(c_2^2 - c_3)e_n^3 + (7c_2c_3 - 4c_2^3 - 3c_4)e_n^4 + (8c_2^4 - 20c_2^2c_3 + 6c_3^2 \\ & + 10c_2c_4 - 4c_5)e_n^5 + (-16c_2^5 + 52c_2^3c_3 - 28c_2^2c_4 + 17c_3c_4 + c_2(-33c_3^2 \\ & + 13c_5) - 5c_6)e_n^6 + O(e_n^7).\end{aligned} \tag{7.11}$$

On substituting Eq. (7.11) in first sub-step of the proposed method (7.6), we obtain

$$\begin{aligned}e_{y_n} &= y_n - \eta \\ &= c_2 e_n^2 - 2(c_2^2 - c_3)e_n^3 - (7c_2c_3 - 4c_2^3 - 3c_4)e_n^4 - (8c_2^4 - 20c_2^2c_3 + 6c_3^2 + 10c_2c_4 - 4c_5)e_n^5 \\ &\quad - (-16c_2^5 + 52c_2^3c_3 - 28c_2^2c_4 + 17c_3c_4 + c_2(-33c_3^2 + 13c_5) - 5c_6)e_n^6 + O(e_n^7).\end{aligned} \tag{7.12}$$

Using Eq. (7.12), it is possible to obtain the value of error at the second sub-step as

$$e_{z_n} = z_n - \eta$$

$$= (c_2^3 - c_2c_3)e_n^4 - 2(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4)e_n^5 + (10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 - 7c_3c_4 + 3c_2(6c_3^2 - c_5))e_n^6 + O(e_n^7). \tag{7.13}$$

Similarly, using Eq. (7.13), Taylor's series expansion of $f(z_n)$ can be derived. Substituting this expansion into the third sub-step of the proposed scheme yields

$$\begin{aligned} e_{w_n} &= w_n - \eta \\ &= \frac{1}{2}\left(c_2^3 - c_2^3c_3\right)e_n^4 + \frac{1}{2}(-3c_2^4 + 7_2^2c_3 - 2c_3^2 - 2c_2c_4)e_n^5 + \frac{1}{4}(11c_2^5 - 40c_2^3c_3 + 20c_2^2c_4 \\ &\quad - 14c_3c_4 + c_2(29c_3^2 - 6c_5))e_n^6 + O(e_n^7). \end{aligned} \tag{7.14}$$

Finally, by substituting Eq. (7.14) into the last sub-step of (7.6), we obtain the final error equation as

$$e_{n+1} = \frac{1}{2}\left(c_2^5 - c_2^3c_3\right)e_n^6 + O(e_n^7). \tag{7.15}$$

This finishes the proof. □

Theorem 7.2.2. *Let $f : \mathbb{D} \subseteq \mathbb{C} \to \mathbb{C}$ be a sufficiently differentiable function having a simple zero η in the open interval $\mathbb{D}$. If the initial guess x_0 is close enough to η, then the convergence order of the scheme* (7.7) *is eight and satisfies the following error equation:*

$$e_{n+1} = \frac{1}{2}c_2^3\left(c_2^2 - c_3\right)^2 e_n^8 + O(e_n^9), \tag{7.16}$$

where $c_k = \frac{1}{k!}\frac{f^{(k)}(\eta)}{f'(\eta)}$, $k \geq 2$.

Proof. The proof is similar to the proof of Theorem 7.2.1. Hence, it is omitted. □

Finally, we can obtain the following matrix iterations by imposing the proposed schemes (7.6) and (7.7) on the matrix equation $\mathcal{X}^2 = I$:

$$\mathcal{X}_{n+1} = [I + 32\mathcal{X}_n^2 + 130\mathcal{X}_n^4 + 88\mathcal{X}_n^6 + 5\mathcal{X}_n^8][8(\mathcal{X}_n + 10\mathcal{X}_n^3 + 17\mathcal{X}_n^5 + 4\mathcal{X}_n^7)]^{-1}, \tag{7.17}$$

and

$$\begin{aligned} \mathcal{X}_{n+1} = [I + 49\mathcal{X}_n^2 + 322\mathcal{X}_n^4 + 490\mathcal{X}_n^6 + 157\mathcal{X}_n^8 + 5\mathcal{X}_n^{10}][10\mathcal{X}_n + 152\mathcal{X}_n^3 + 476\mathcal{X}_n^5 \\ + 344\mathcal{X}_n^7 + 42\mathcal{X}_n^9]^{-1}, \end{aligned} \tag{7.18}$$

where $\mathcal{X}_0 = A$. The iterative approaches (7.17) and (7.18) require eight and ten matrix-matrix products, respectively. In order to lessen the computational work, the iterative approaches (7.17) and (7.18) can be expressed more cost-effectively as

$$\mathcal{X}_{n+1} = [I + 32\mathcal{X}_n^2 + 130\mathcal{X}_n^4 + 88\mathcal{X}_n^6 + 5\mathcal{X}_n^8][8\mathcal{X}_n(I + 10\mathcal{X}_n^2 + 17\mathcal{X}_n^4 + 4\mathcal{X}_n^6)]^{-1}, \tag{7.19}$$

and

$$\begin{aligned}\mathcal{X}_{n+1} = [I + 49\mathcal{X}_n^2 + 322\mathcal{X}_n^4 + 490\mathcal{X}_n^6 + 157\mathcal{X}_n^8 + 5\mathcal{X}_n^{10}][\mathcal{X}_n(10I + 152\mathcal{X}_n^2 + 476\mathcal{X}_n^4 \\ + 344\mathcal{X}_n^6 + 42\mathcal{X}_n^8)]^{-1}.\end{aligned} \tag{7.20}$$

We denote the above iterations (7.19) and (7.20) by LM_1 and LM_2, respectively.

Remark 7.2.1. *It is worth mentioning that the matrix iterations for computing the sign function S must be globally convergent, and it should not be a member of the Padé family* (6.7) *or any existing family of iterative methods or their reciprocals. It is interesting to note that the proposed iterations* (7.17) *and* (7.18) *are novel since they are not members of the Padé family.*

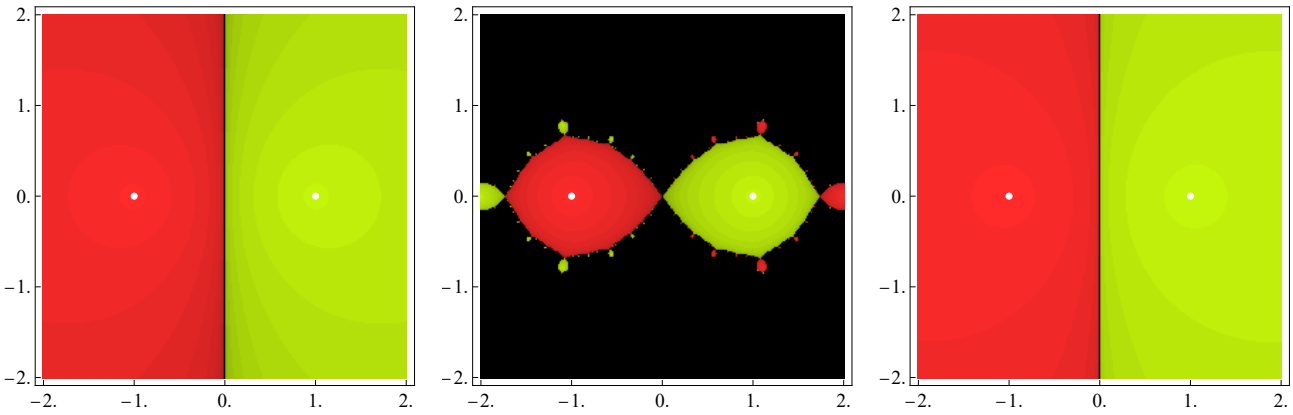
This demonstrates that the proposed schemes possess the requisite ability and originality to facilitate a deeper investigation.

7.3 Attraction basins to ensure global convergence

Now, our second goal is to achieve global convergence for the proposed methods (7.6) and (7.7) for solving equation $f(x) = x^2 - 1 = 0$ (see [171] for more information). The global convergence of the methods can be observed by drawing the basins of attraction.

For this purpose, we consider a rectangular domain $\mathbb{D} = [-2, 2] \times [-2, 2] \subset \mathbb{C}$ with a 256×256 mesh, containing two simple zeros at -1 and 1. Each point is colored according to the simple zero to which the iterative method, starting from $z_0 \in \mathbb{D}$, converges, or is marked in black if the iteration diverges. The stopping criterion adopted in this section is $|f(x_n)| \leq 10^{-2}$, with a maximum of 25 iterations. Consequently, the attraction basins corresponding to different methods can be clearly distinguished by their colors. The white points located within the basins also indicate the precise positions of the zeros.

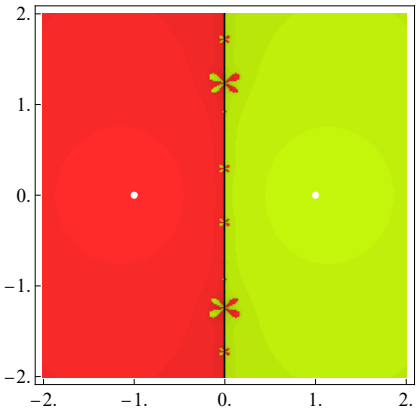
Figures 7.1–7.2 depict the basins of attraction for the schemes NM (6.5), NSM (6.8), HM (6.9), $\mathrm{PM}_{r,s}$ (6.7), LM_1 (7.19), and LM_2 (7.20). This indicates that the methods NSM, $\mathrm{PM}_{3,2}$, $\mathrm{PM}_{4,1}$,



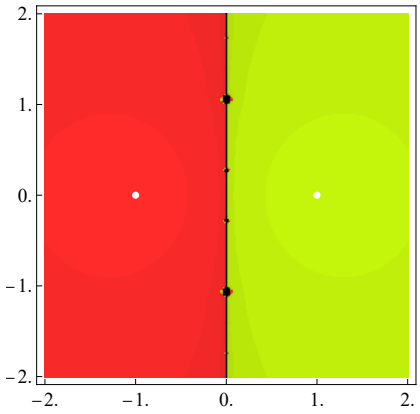
(a) NM

(b) NSM

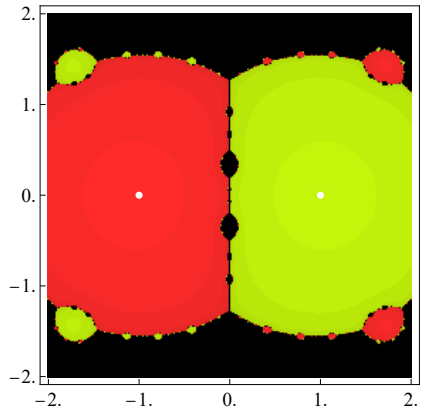
(c) HM



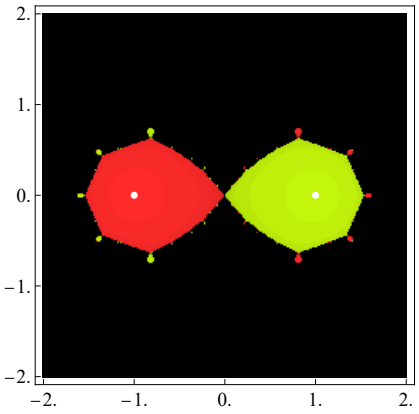
(d) $PM_{1,4}$



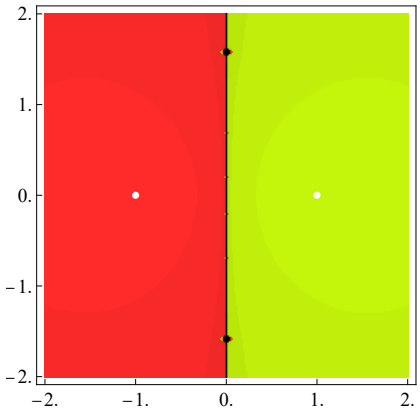
(e) $PM_{3,2}$



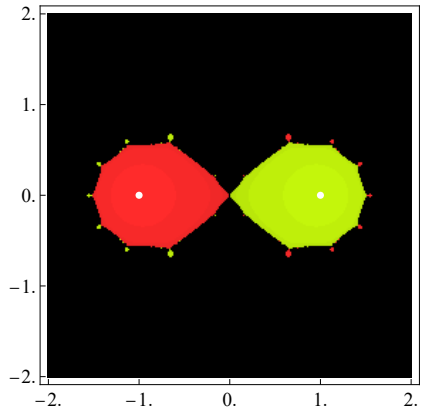
(f) $PM_{4,1}$



(g) $PM_{5,0}$



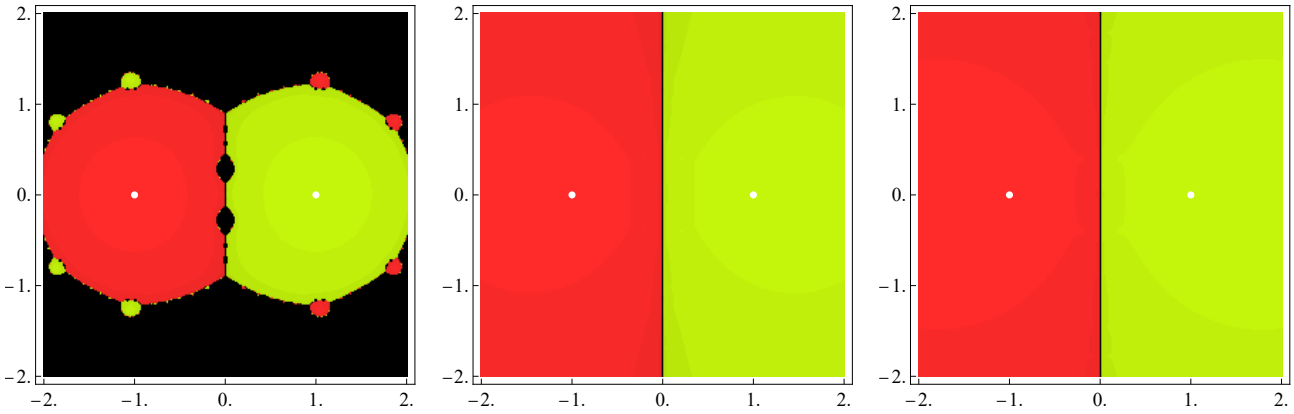
(h) $PM_{4,3}$



(i) $PM_{7,0}$

Figure 7.1: The basins of attraction of distinct methods for solving equation $f(x) = x^2 - 1 = 0$.

$PM_{5,0}$, $PM_{4,3}$, $PM_{7,0}$, $PM_{6,1}$, or simply Padé variants of (6.7) having sixth or eighth-order convergence are converging locally, while the other methods converge globally. Moreover, the higher



(a) $PM_{6,1}$ (b) LM_1 (c) LM_2

Figure 7.2: The basins of attraction of distinct methods for solving equation $f(x) = x^2 - 1 = 0$.

convergence order of $PM_{1,4}$, LM_1, and LM_2 resulted in broader and lighter attraction basins.

In the upcoming sections, we will analyze several crucial factors related to matrix iterations for determining the matrix sign function, including convergence rate and numerical stability.

7.4 Convergence analysis

Here, we present the theoretical behavior of matrix iterative expressions (7.19) and (7.20) in relation to the sign of a matrix.

Theorem 7.4.1. *Consider a set containing imaginary eigenvalues of the initial matrix $\mathcal{X}_0 = A \in \mathbb{C}^{m\times m}$ as empty. Then, the sequence of matrix iterations $\{\mathcal{X}_n\}_{n=0}^{\infty}$ obtained from (7.19) converges to the matrix sign, i.e., S.*

Proof. Suppose R is a rational function with which (7.19) is associated. As we know, for any complex matrix $\mathcal{X} \in \mathbb{C}^{m\times m}$, the Jordan canonical form can be written if there exists an invertible matrix U such that $\mathcal{X} = UJU^{-1}$. Then,

$$R(\mathcal{X}) = UR(J)U^{-1}.$$

Using Eq. (7.19), an eigenvalue β of $\mathcal{X}_n$ is mapped to an eigenvalue $R(\beta)$ of $\mathcal{X}_{n+1}$. Owing to this scalar eigenvalue relation, it is necessary to examine how the complex plane is mapped onto itself by $R(\beta)$. More precisely, the rational function R must satisfy the following two properties:

(i) Preservation of sign, i.e., $\text{sign}(R(x)) = \text{sign}(x),\ \forall\ x \in \mathbb{C}$.

(ii) Convergence should be global, i.e, the sequence defined by $x_{n+1} = R(x_n)$, with initial guess

$x_0 = x$ not located on the imaginary axis, converges to sign(x).

To prove this, consider A has a Jordan canonical form that can be written as in [154]:

$$U^{-1}AU = \Lambda = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}, \tag{7.21}$$

where U is a non-singular matrix and $C,\ N$ are Jordan blocks equivalent to eigenvalues in $\mathbb{C}^-$ and $\mathbb{C}^+$, respectively. The values located on the principal diagonal of blocks C and N are denoted by $\beta_1,\ \beta_2, \ldots, \beta_p$, and $\beta_{p+1}, \ldots, \beta_m$, respectively. Using Eq. (7.21), we have

$$\text{sign}(A) = U \begin{bmatrix} -I_p & 0 \\ 0 & I_{m-p} \end{bmatrix} U^{-1}. \tag{7.22}$$

Therefore, one can get

$$\begin{aligned} \text{sign}(\Lambda) &= \text{sign}(U^{-1}AU) = U^{-1}\text{sign}(A)U, \\ &= \begin{pmatrix} \text{sign}(\beta_1) & & & & & & \\ & \ddots & & & & & \\ & & \text{sign}(\beta_p) & & & & \\ & & & \text{sign}(\beta_{p+1}) & & & \\ & & & & \ddots & & \\ & & & & & \text{sign}(\beta_m) \end{pmatrix}. \end{aligned} \tag{7.23}$$

From $\mathfrak{D}_0 = U^{-1}AU$, we define $\mathfrak{D}_n = U^{-1}\mathcal{X}_n U,\ n = 1, 2, \ldots,$ such that the sequence converges to sign(Λ). Then, from the proposed scheme (7.19), one can obtain

$$\mathfrak{D}_{n+1} = [I + 32\mathfrak{D}_n^2 + 130\mathfrak{D}_n^4 + 88\mathfrak{D}_n^6 + 5\mathfrak{D}_n^8][8\mathfrak{D}_n(I + 10\mathfrak{D}_n^2 + 17\mathfrak{D}_n^4 + 4\mathfrak{D}_n^6)]^{-1}. \tag{7.24}$$

If $\mathfrak{D}_0$ has entries on the main diagonal, then all subsequent $\mathfrak{D}_n$ are diagonal, based on mathematical induction. It is important to note that the case where $\mathfrak{D}_0$ is not a diagonal matrix cannot be treated in the same way, as addressed in the proof later. To solve $f(x) = x^2 - 1 = 0$, express Eq. (7.24) in the form of an m uncoupled scalar iterative technique as follows:

$$d_{n+1}^i = \frac{1 + 32{d_n^i}^2 + 130{d_n^i}^4 + 88{d_n^i}^6 + 5{d_n^i}^8}{8d_n^i + 80{d_n^i}^3 + 126{d_n^i}^5 + 32{d_n^i}^7}, \tag{7.25}$$

where $d_n^i = (\mathfrak{D}_n)_{i,i}$ and $1 \leq i \leq m$. From Eqs. (7.24) and (7.25), we must analyze the convergence behavior of $\{d_n^i\}$ to $\text{sign}(\beta_i)$, $\forall\ 1 \leq i \leq m$.

From Eq. (7.25) and with the assumption on eigenvalues, we have $\text{sign}(\beta_i) = s_i = \pm 1$. Thus,

$$\frac{d_{n+1}^i - s_i}{d_{n+1}^i + s_i} = \left(\frac{-s_i + d_n^i}{s_i + d_n^i}\right)^6 \frac{1 - 2s_i d_n^i + 5(d_n^i)^2}{1 + 2s_i d_n^i + 5(d_n^i)^2}. \tag{7.26}$$

It can be seen that the right-side expression in Eq. (7.26) is bounded for $i = 1, 2, \ldots, m$.

By adopting an appropriate initial matrix $\mathcal{X}_0 = A$, $\left|\frac{-s_i + d_0^i}{s_i + d_0^i}\right| < 1$, we have

$$\lim_{n\to\infty} \left|\frac{-s_i + d_{n+1}^i}{s_i + d_{n+1}^i}\right| = 0, \tag{7.27}$$

and therefore, $\lim_{n\to\infty}(d_n^i) = s_i = \text{sign}(\beta_i)$. Hence, we conclude that $\lim_{n\to\infty}(\mathfrak{D}_n) = \text{sign}(\Lambda)$.

As briefly presented at the start of the proof, we should study the scalar relationship among eigenvalues of the iterates for (7.19), if $\mathfrak{D}_0$ is not diagonal. The eigenvalues of $\mathcal{X}_n$ are mapped from the iterate n to $n+1$ by the following relation:

$$\beta_{n+1}^i = [I + 32{\beta_n^i}^2 + 130{\beta_n^i}^4 + 88{\beta_n^i}^6 + 5{\beta_n^i}^8][8\beta_n^i(I + 10{\beta_n^i}^2 + 17{\beta_n^i}^4 + 4{\beta_n^i}^6)]^{-1}. \tag{7.28}$$

Eq. (7.28) demonstrates that in the usual case the eigenvalues are convergent to $s_i = \pm 1$, using the same methodology as before

$$\lim_{n\to\infty} \left|\frac{\beta_{n+1}^i - s_i}{\beta_{n+1}^i + s_i}\right| = 0. \tag{7.29}$$

Finally, it is simple to conclude that

$$\lim_{n\to\infty} \mathcal{X}_n = U(\lim_{n\to\infty} \mathfrak{D}_n)U^{-1} = U\text{sign}(\Lambda)U^{-1} = \text{sign}(A). \tag{7.30}$$

This finishes the proof. □

Analogously to the previous theorem, we can conclude the following result.

Theorem 7.4.2. *Consider a set containing imaginary eigenvalues of the initial matrix $\mathcal{X}_0 = A \in \mathbb{C}^{m\times m}$ as empty. Then, the sequence of matrix iterations $\{\mathcal{X}_n\}_{n=0}^{\infty}$ obtained from* (7.20) *converges to the sign of matrix, i.e., S.*

We now examine the convergence rate of (7.19) and (7.20), despite the preceding theorem focusing on convergence analysis.

7.4.1 Convergence rate

In Theorems 7.4.1–7.4.2, we have confirmed the convergence of the proposed iterations towards the matrix sign function S. Now, it is important to further analyze the rate of convergence of (7.19) and (7.20). This is the aim of the next theorem.

Theorem 7.4.3. *Assume the identical hypotheses as in Theorem 7.4.1. Then, the proposed method* (7.19) *has a convergence rate of six to* S.

Proof. This is clear that $\mathcal{X}_n$ is a rational operator of A for every $n \geq 0$ and hence commutes with S, just like A. In addition, the following relationships are available in [335]:

$$S^2 = I,\ S^{-1} = S,\ S^{2j} = I \text{ and } S^{2j+1} = S,\ j \geq 1. \tag{7.31}$$

Applying the following substitution $C_n = (8\mathcal{X}_n(I + 10\mathcal{X}_n^2 + 17\mathcal{X}_n^4 + 4\mathcal{X}_n^6))$, we obtain

$$\begin{aligned}
\mathcal{X}_{n+1} - S &= (I + 32\mathcal{X}_n^2 + 130\mathcal{X}_n^4 + 88\mathcal{X}_n^6 + 5\mathcal{X}_n^8)C_n^{-1} - S \\
&= (I + 32\mathcal{X}_n^2 + 130\mathcal{X}_n^4 + 88\mathcal{X}_n^6 + 5\mathcal{X}_n^8 - SC_n)C_n^{-1} \\
&= (I + 32\mathcal{X}_n^2 + 130\mathcal{X}_n^4 + 88\mathcal{X}_n^6 + 5\mathcal{X}_n^8 - 8S\mathcal{X}_n - 80S\mathcal{X}_n^3 - 136S\mathcal{X}_n^5 - 32S\mathcal{X}_n^7)C_n^{-1} \\
&= (S^8 - 8S^7\mathcal{X}_n + 32S^6\mathcal{X}_n^2 - 80S^5\mathcal{X}_n^3 + 130S^4\mathcal{X}_n^4 - 136S^3\mathcal{X}_n^5 + 88S^2\mathcal{X}_n^6 - 32S\mathcal{X}_n^7 + 5\mathcal{X}_n^8)C_n^{-1} \\
&= ((S - \mathcal{X}_n)^8 + 4\mathcal{X}_n^2(S^6 - 6S^5\mathcal{X}_n + 15S^4\mathcal{X}_n^2 - 20S^3\mathcal{X}_n^3 + 15S^2\mathcal{X}_n^4 - 6S\mathcal{X}_n^5 + \mathcal{X}_n^6))C_n^{-1} \\
&= (\mathcal{X}_n - S)^6(I - 2S\mathcal{X}_n + 5\mathcal{X}_n^2)C_n^{-1}.
\end{aligned} \tag{7.32}$$

Now, applying the matrix norm on both sides of Eq. (7.32), we obtain

$$||\mathcal{X}_{n+1} - S|| \leq ||C_n^{-1}||\ ||I - 2S\mathcal{X}_n + 5\mathcal{X}_n^2||\ ||\mathcal{X}_n - S||^6. \tag{7.33}$$

This demonstrates that the new method (7.19) has a sixth-order of convergence. This finishes the proof. □

Theorem 7.4.4. *Assume the identical hypotheses as in Theorem 7.4.2. Then, the proposed method* (7.20) *has a convergence rate of eight to* S.

Proof. The proof of this theorem is similar to the proof of Theorem 7.4.3. It is hence omitted. □

7.5 Application in solving Yang–Baxter equation

We can recall the existence result [337] for computing the solutions of the YBM equation (7.1) corresponding to the matrix A with no pure imaginary eigenvalues for determining non-trivial solutions of (7.1).

Theorem 7.5.1. *Let S be the sign function of a square matrix A. Then, $\mathcal{X} = A\frac{(I-S)}{2}$ and $\mathcal{X} = A\frac{(I+S)}{2}$ are both solutions of* (7.1).

Proof. Based on properties of the matrix sign function, we can derive the following

$$A\left(A\frac{(I+S)}{2}\right)A = A^3\frac{(I+S)}{2}, \tag{7.34}$$

and

$$\left(A\frac{(I+S)}{2}\right)A\left(A\frac{(I+S)}{2}\right) = A^3\left(\frac{(I+S)}{2}\right)^2 = A^3\frac{(I+S)}{2}. \tag{7.35}$$

Therefore, $\mathcal{X} = \left(A\frac{(I+S)}{2}\right)$ is a solution of equation (7.1). On similar basis, $\mathcal{X} = \left(A\frac{(I-S)}{2}\right)$ is also a solution of equation (7.1). □

Theorem 7.5.2. *For a given square matrix A and for $\gamma \in \mathbb{C}$. If*

(a) S^* be the sign function of a matrix $H = A + \gamma I$. Then, $\mathcal{X} = A\frac{(I-S^*)}{2}$ and $\mathcal{X} = A\frac{(I+S^*)}{2}$ are both solutions of (7.1).

(b) S^{**} be the sign function of a matrix $H = \gamma A$. Then, $\mathcal{X} = A\frac{(I-S^{**})}{2}$ and $\mathcal{X} = A\frac{(I+S^{**})}{2}$ are both solutions of (7.1).

Proof. Using $(S^*)^2 = I$ and $S^*(A + \gamma I) = (A + \gamma I)S^*$, we have $S^*A = AS^*$. It is obvious how the rest of the proof works. Similarly, proof in the second case can be done. □

7.6 Stability analysis

The stability of (7.19) and (7.20) for locating S in a neighborhood of the solution is explored in this section. We study how a small perturbation at the n^{th} iteration is exaggerated or dispersed as the iterations progress.

Theorem 7.6.1. *The sequence $\{\mathcal{X}_n\}_{n=0}^{\infty}$ formed by* (7.19) *is asymptotically stable under the identical assumptions as in Theorem 7.4.3.*

Proof. If the initial condition is $\mathcal{X}_0 = A$, then from (7.19) the obtained iterate is also a function of A

and hence satisfies the commutative property on A. Consider $\tilde{\mathcal{X}}_n$ to be the numerical perturbation generated at the n^{th} iterate in (7.19). We have

$$\tilde{\mathcal{X}}_n = \mathcal{X}_n + \delta_n. \tag{7.36}$$

We conduct first-order error analysis here, which means we explicitly use approximations where $(\delta_n)^l \approx 0$, since $(\delta_n)^l$ for $l \geq 2$ is closer to zero (matrix). This usual approach is appropriate if $(\delta_n)^l$ is small enough. For sufficiently large values of n, we have the assumption that $\mathcal{X}_n \approx \text{sign}(A) = S$.

$$\begin{aligned}
\tilde{\mathcal{X}}_{n+1} &= (I + 32\tilde{\mathcal{X}}_n^2 + 130\tilde{\mathcal{X}}_n^4 + 88\tilde{\mathcal{X}}_n^6 + 5\tilde{\mathcal{X}}_n^8)[8\tilde{\mathcal{X}}_n + 80\tilde{\mathcal{X}}_n^3 + 136\tilde{\mathcal{X}}_n^5 + 32\tilde{\mathcal{X}}_n^7]^{-1}, \\
&= (I + 32(\mathcal{X}_n + \delta_n)^2 + 130(\mathcal{X}_n + \delta_n)^4 + 88(\mathcal{X}_n + \delta_n)^6 + 5(\mathcal{X}_n + \delta_n)^8)[8(\mathcal{X}_n + \delta_n) \\
&\quad + 80(\mathcal{X}_n + \delta_n)^3 + 136(\mathcal{X}_n + \delta_n)^5 + 32(\mathcal{X}_n + \delta_n)^7]^{-1}, \\
&\approx (256I + 576\delta_n S + 576S\delta_n)(256S + 704\delta_n + 448S\delta_n S)^{-1}, \\
&\approx \left(I + \frac{9}{4}\delta_n S + \frac{9}{4}S\delta_n\right)\left(S - \frac{11}{4}\delta_n - \frac{7}{4}S\delta_n S\right), \\
&\approx S - \frac{1}{2}\delta_n + \frac{1}{2}S\delta_n S,
\end{aligned} \tag{7.37}$$

wherein using the identity $(\mathcal{B}+\mathcal{C})^{-1} = \mathcal{B}^{-1} - \mathcal{B}^{-1}\mathcal{C}\mathcal{B}^{-1} + O(\mathcal{C}^2)$, and $\delta_{n+1} = \tilde{\mathcal{X}}_{n+1} - \mathcal{X}_{n+1}$, and after some algebraic manipulations, one gets

$$\delta_{n+1} \approx \frac{1}{2}\delta_n - \frac{1}{2}S\delta_n S. \tag{7.38}$$

We have used the facts about the matrix sign function, i.e., $S^2 = I$ and $S^{-1} = S$. We can now conclude that the perturbation at iterate $n+1$ is bounded, i.e.,

$$\delta_{n+1} \approx \frac{1}{2^{n+1}}(\delta_0 - S\delta_0 S).$$

Therefore, the sequence $\{\mathcal{X}_n\}_{n=0}^{\infty}$ generated by (7.19) is asymptotically stable. This ends the proof. □

Theorem 7.6.2. *The sequence $\{\mathcal{X}_n\}_{n=0}^{\infty}$ formed by* (7.20) *is asymptotically stable under the identical assumptions as in Theorem 7.4.4.*

Proof. This theorem can be demonstrated in a manner analogous to Theorem 7.6.1. So, it is omitted. □

7.7 Efficiency challenge

As discussed in the previous chapter, now we have also computed the efficiency index of the methods SM, LM_1, $PM_{6,1}$, and LM_2 as

$$CEI_{\mathrm{SM}} = 6^{\frac{1}{8m^{2373/1000}+\frac{2m^3}{3}}}, CEI_{\mathrm{LM_1}} = 6^{\frac{1}{6m^{2373/1000}+\frac{2m^3}{3}}},$$
$$CEI_{\mathrm{PM_{6,1}}} = 8^{\frac{1}{8m^{2373/1000}+\frac{2m^3}{3}}}, CEI_{\mathrm{LM_2}} = 8^{\frac{1}{7m^{2373/1000}+\frac{2m^3}{3}}}.$$

where m is the matrix size. Meanwhile, we have plotted the efficiencies of several methods in Figure 7.3, showing that the proposed schemes have better efficiency indices.

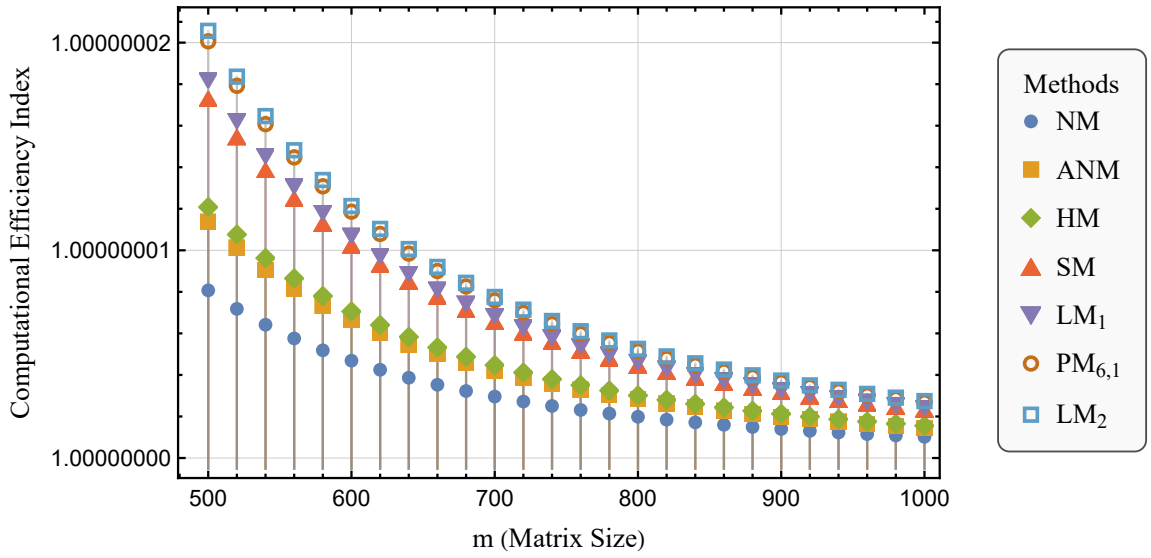


Figure 7.3: The efficiency indices comparison of NM, ANM, HM, SM, LM_1, $PM_{6,1}$ and LM_2 with the increase in m.

Using the above computational efficiencies, we can measure the time required for each iterative method to complete.

7.8 Numerical tests

This section presents several examples to verify the behavior of the methods. The computational results are based on the iteration number and the residual norm for various matrix iterations. If

the iteration $\mathcal{X}_n$ has a largest eigenvalue, i.e., if $||\mathcal{X}_n|| >> 1$, the convergence may be slow. As a result, we can accelerate the convergence of the proposed iteration via scaling (for reference, see [211]). The scaling parameter ν_n is introduced in equation (6.32). With scaling factor, the schemes (7.19) and (7.20) can be written as:

$$\begin{cases} \mathcal{X}_0 & = A, \\ \nu_n & = \text{the scaling parameter computed by } (\mathbf{??}), \\ \mathcal{X}_{n+1} & = [I + 32{\nu_n}^2\mathcal{X}_n^2 + 130{\nu_n}^4\mathcal{X}_n^4 + 88{\nu_n}^6\mathcal{X}_n^6 + 5{\nu_n}^8\mathcal{X}_n^8]A_n^{-1}, \quad n \in \hat{\mathbb{N}}_0, \end{cases} \tag{7.39}$$

and

$$\begin{cases} \mathcal{X}_0 & = A, \\ \nu_n & = \text{the scaling parameter computed by } (\mathbf{??}), \\ \mathcal{X}_{n+1} & = [I + 49{\nu_n}^2\mathcal{X}_n^2 + 322{\nu_n}^4\mathcal{X}_n^4 + 490{\nu_n}^6\mathcal{X}_n^6 + 157{\nu_n}^8\mathcal{X}_n^8 + 5{\nu_n}^{10}\mathcal{X}_n^{10}]B_n^{-1}, \ n \in \hat{\mathbb{N}}_0, \end{cases} \tag{7.40}$$

where $A_n = 8\nu_n\mathcal{X}_n(I + 10{\nu_n}^2\mathcal{X}_n^2 + 17{\nu_n}^4\mathcal{X}_n^4 + 4{\nu_n}^6\mathcal{X}_n^6)$, $B_n = \nu_n\mathcal{X}_n(10I + 152{\nu_n}^2\mathcal{X}_n^2 + 476{\nu_n}^4\mathcal{X}_n^4 + 344{\nu_n}^6\mathcal{X}_n^6 + 42{\nu_n}^8\mathcal{X}_n^8)$, $\lim_{n\to\infty} \nu_n = 1$, and $\lim_{n\to\infty} \mathcal{X}_n = S$. However, due to high cost in some cases, the study of scaling parameter ν_n is not studied in depth; for instance, when evaluating the determinantal scaling factor, some care is needed to avoid unnecessary overflow and underflow, especially when m is large. The quantity ν_n should be within the range of the floating-point arithmetic since its reciprocal has magnitude of the geometric mean of the eigenvalues of $\mathcal{X}_n$ and hence lies between the moduli of the smallest and largest eigenvalues. Determinantal scaling can be ineffective when a small group of outlying eigenvalues exists, and the rest are nearly converged. Moreover, spectral scaling can be ineffective when the eigenvalues of A are clustered close to the imaginary axis. In general, all of them work well, but there are some specific advantages and disadvantages. All three scaling parameters converge to 1 as $\mathcal{X}_n \to S$, so scaling does not destroy the convergence, nor does it bring any benefit.

The stopping criterion adopted for the numerical examples in this chapter is

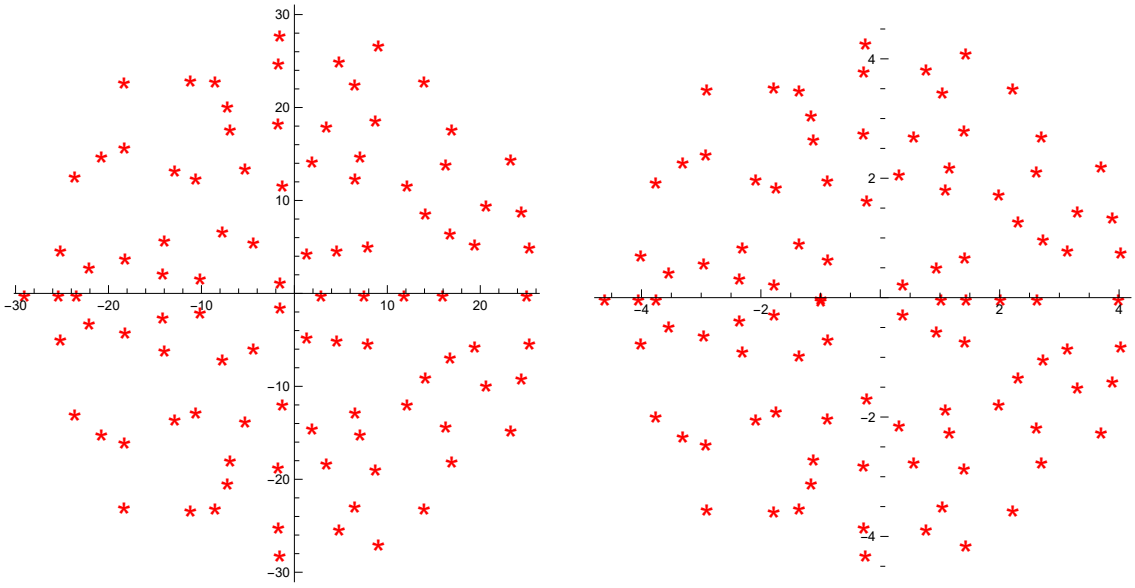
$$||\mathcal{X}_n^2 - I||_* \leq \epsilon, \tag{7.41}$$

where ϵ and $||.||_*$ denote tolerance and suitable matrix norm, respectively. The l_∞ and l_2 are considered for real and complex input matrices (see [327]).

The iteration number and computational e-time (CPU) for various methods are compared. We only employ approaches that have global convergence behavior for the sake of comparison. The methods being compared are NM (6.5), ANM (6.6), HM (6.9), SM_1 [$a = -1/2$ in (7.4)], SM_2 [$a = -5/4$ in (7.4)], $\text{PM}_{1,4}$ (6.7), LM_1 (7.19), LM_2 (7.20), ALM_1 (7.39), and ALM_2 (7.40). We have used spectral scaling in methods ALM_1 and ALM_2. Additionally, comparisons with methods exhibiting local convergence behavior (such as NSM (6.8)) or methods from different categories (like computing the Cauchy integral (6.1)) are excluded. The simulations are carried out using Mathematica [364] with system specifications "Windows 11 Pro Insider Intel(R) Core(TM) i7-7600U CPU running at 2.90 GHz processor having 8.00 GB RAM (64-bit operating system)."

Experiment 7.8.1. *In the first example, we study the eigenvalue clustering using a randomly generated real matrix of size* 100×100*, as follows:*

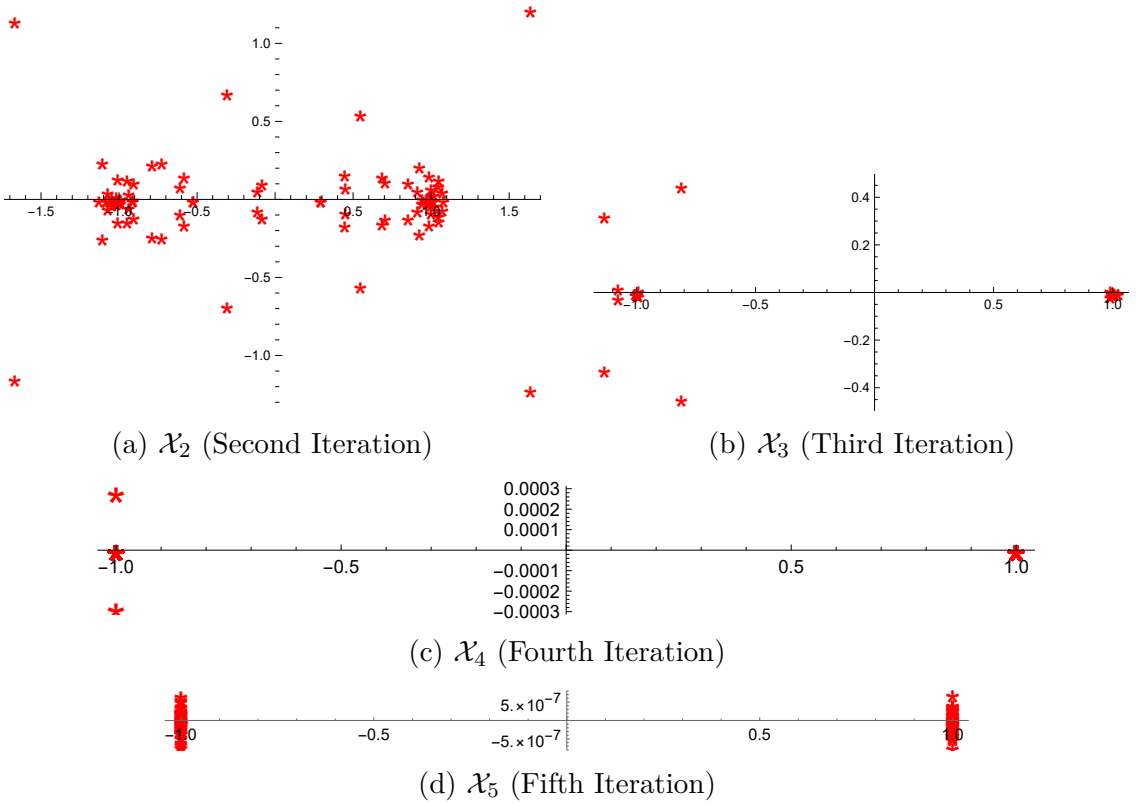
```
SeedRandom[1]; A = RandomReal[{-5, 5},{100, 100}];
```



(a) $\mathcal{X}_0$ (Initial Iteration) (b) $\mathcal{X}_1$ (First Iteration)

Figure 7.4: Clustering of eigenvalues of real matrix 100×100 in Experiment 7.8.1.

The results are reported in Figures 7.4–7.5, by applying the infinity norm with $\epsilon = 10^{-3}$ *in double precision* (7.41)*. We observe that at the initial stage, the eigenvalues of* $\mathcal{X}_0$ *are scattered. However, when using the proposed scheme* (7.19) *for the next approximation, the eigenvalues of* $\mathcal{X}_i$*,* $i = 2, \ldots, 5$*, accumulate either around* -1 *or* 1*. This reveals that the matrix sequence* $\{\mathcal{X}_n\}_{n=0}^{\infty}$ *is converging towards sign*(A)*. Hence, the theoretical results of Theorem 7.4.1 are verified.*



(a) $\mathcal{X}_2$ (Second Iteration)

(b) $\mathcal{X}_3$ (Third Iteration)

(c) $\mathcal{X}_4$ (Fourth Iteration)

(d) $\mathcal{X}_5$ (Fifth Iteration)

Figure 7.5: Clustering of eigenvalues of real matrix 100×100 in Experiment 7.8.1.

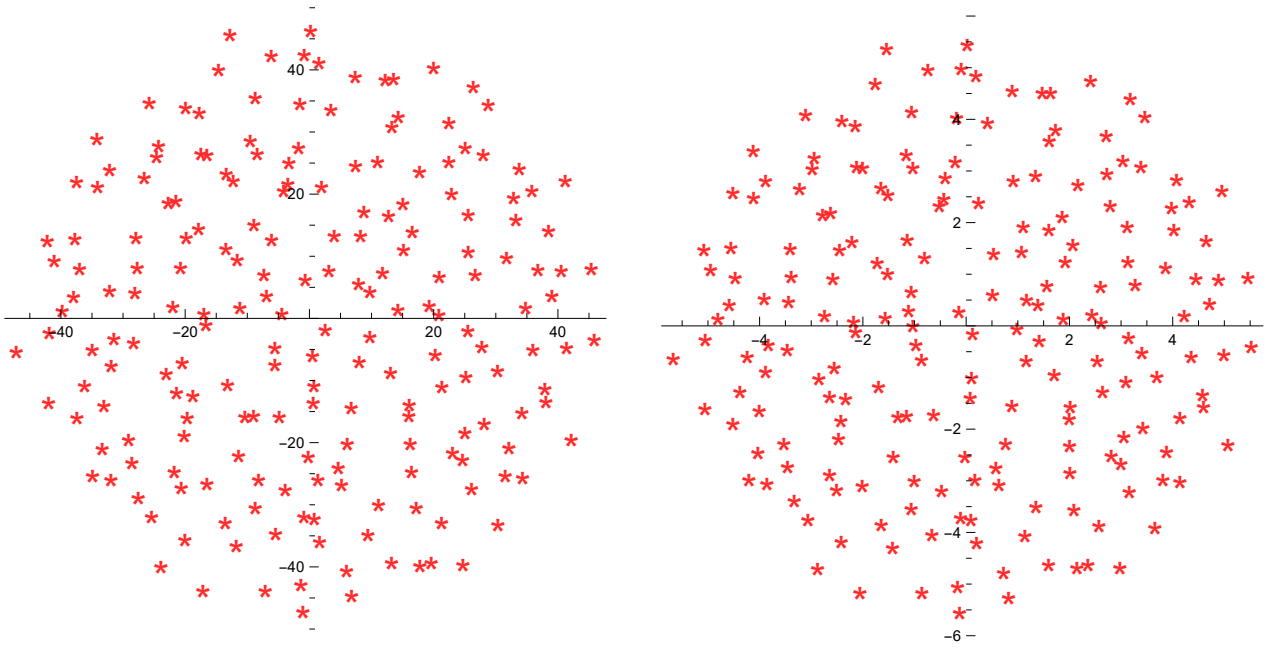
Experiment 7.8.2. *Here, we consider the clustering of eigenvalues using a randomly generated complex matrix of size* 200×200 *as follows:*

```
SeedRandom[5];
A = RandomComplex[{-4 - 4 I, 4 + 4 I}, {200, 200}];
```

Figures 7.6–7.7 show the results using $\epsilon = 10^{-3}$ *in double precision and* l_2 *norm. We can observe that the eigenvalues of* $\mathcal{X}_0$ *are scattered at first. Still, as we progress to the next approximation using the proposed method* (7.20)*, the eigenvalues of* $\mathcal{X}_i, i = 2, 3, \ldots, 5$ *are accumulating either around* -1 *or* 1*. This shows that the matrix sequence* $\{\mathcal{X}_n\}_{n=0}^{\infty}$ *is convergent towards sign*(A)*. As a result, the theoretical results of Theorem 7.4.2 are confirmed.*

Experiment 7.8.3. *In this example, we discuss the study of the matrix sign function using a random complex matrix of size* 1000×1000*, applying different iterative methods with the following input:*

```
SeedRandom[11];
A = RandomComplex[{-5 - 7*Sqrt[-1], 5 + 7*Sqrt[-1]}, {1000, 1000}];
```



(a) $\mathcal{X}_0$ (Initial Iteration) (b) $\mathcal{X}_1$ (First Iteration)

Figure 7.6: Clustering of eigenvalues of complex matrix 200×200 in Experiment 7.8.2.

using $\epsilon = 10^{-4}$ *with* l_2 *norm, since the input is a complex matrix with initial condition* $\mathcal{X}_0 = A$.

Figure 7.8 presents the comparison results in terms of iteration number and absolute error, illustrating a stable behavior for the new methods LM_1, LM_2, ALM_1, *and* ALM_2 *in computing the matrix sign function.*

Experiment 7.8.4. *Next, we discuss the study of the matrix sign function using a random complex matrix of* 1500×1500 *for different iterative methods by employing the following input:*

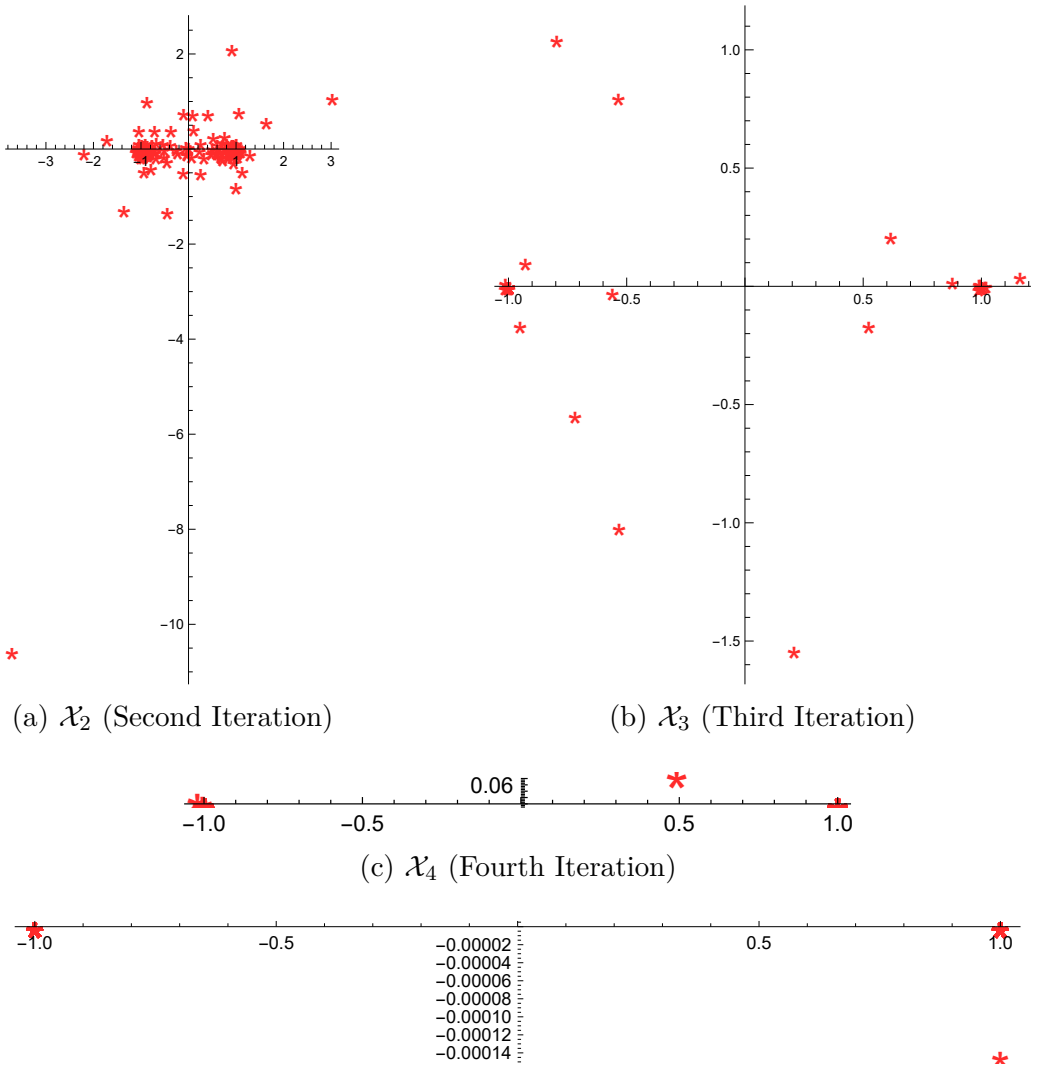
```
SeedRandom[15];
A = RandomComplex[{-1 - 1*Sqrt[-1], 1 + 1*Sqrt[-1]}, {1500, 1500}];
```

using $\epsilon = 10^{-4}$ *with* l_2 *norm, since the input is a complex matrix with initial condition* $\mathcal{X}_0 = A$.

Figure 7.9 depicts the comparison results in terms of the iteration number and absolute error, demonstrating a steady and consistent behavior of LM_1, LM_2, ALM_1, *and* ALM_2 *in computing the matrix sign function.*

Experiment 7.8.5. *In this experiment, we study the convergence history of different methods for a randomly* 2000×2000 *real matrix as follows:*

```
SeedRandom[111];
A = RandomReal[{-2, 2}, {2000, 2000}];
```



(a) $\mathcal{X}_2$ (Second Iteration)

(b) $\mathcal{X}_3$ (Third Iteration)

(c) $\mathcal{X}_4$ (Fourth Iteration)

(d) $\mathcal{X}_5$ (Fifth Iteration)

Figure 7.7: Clustering of eigenvalues of complex matrix 200×200 in Experiment 7.8.2.

using $\epsilon = 10^{-3}$ *with* l_∞ *norm, since the input is a real matrix with initial condition* $\mathcal{X}_0 = A$.

Figure 7.10 illustrates that the newly proposed methods outperform the existing ones and converge towards the matrix sign function. Additionally, the comparison results show that the new methods require fewer iterations and absolute values, further revealing the steady and consistent behavior of the suggested methods in computing the matrix sign function.

Experiment 7.8.6. *Now, we compare the convergence histories of several methods for a set of ten complex matrices chosen at random:*

```
SeedRandom[50]; number = 10; tolerance = 10^-5; n = 0; max = 50;
```

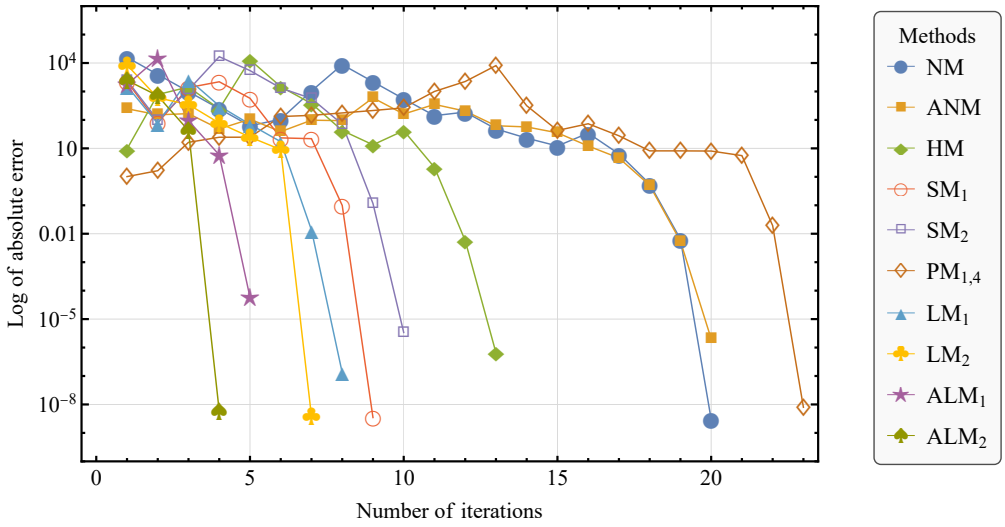


Figure 7.8: The convergence history of different methods in Experiment 7.8.3.

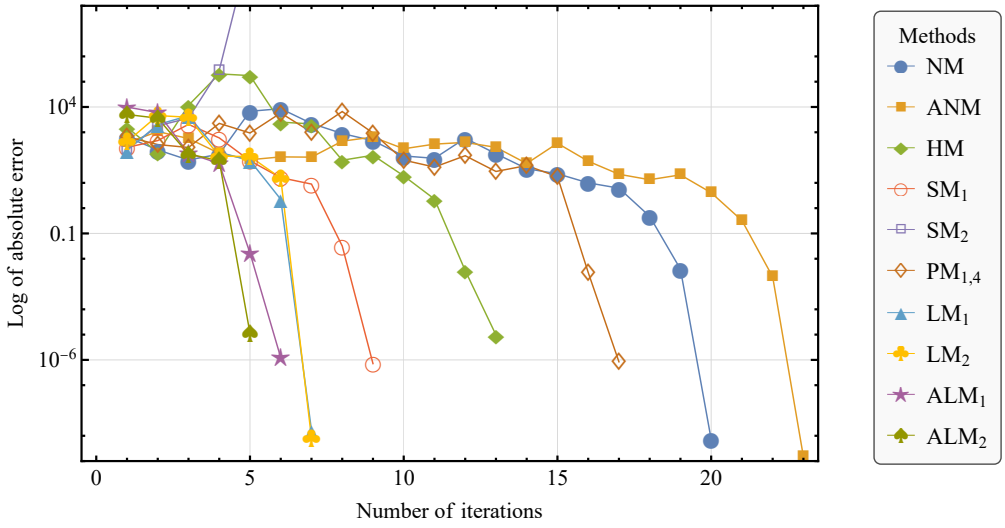


Figure 7.9: The convergence history of different methods in Experiment 7.8.4.

```
Table[A[l] = RandomComplex[{-1.5 - 1.5 I, 1.5 + 1.5 I}, {50*l, 50*l}];,
{l, number}];
```

The numerical results are given in Tables 7.1–7.2 for input matrices of different sizes, depending on the required iteration number and the e-time (CPU-time) using l_2-norm, since the matrices are complex. They demonstrate that the average iteration number and CPU time improve when the new methods are employed, as indicated in the last row of each table. Table 7.2 has units in seconds. As a result, the proposed iterative methods exhibit robust and consistent behavior while computing the matrix sign function.

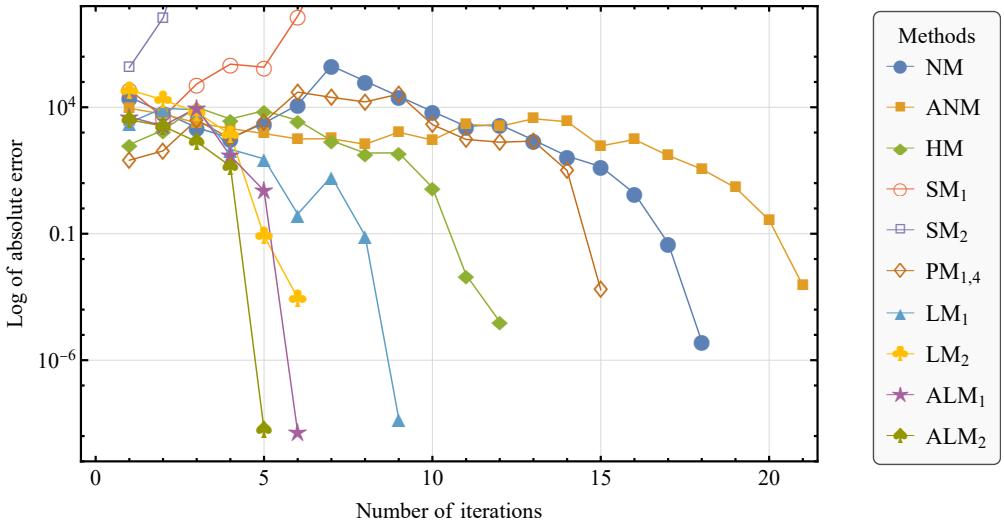


Figure 7.10: The convergence history of different methods in Experiment 7.8.5.

Table 7.1: Comparisons in terms of iteration number using various random matrices of Experiment 7.8.6

Matrix	NM	ANM	HM	SM_1	SM_2	$PM_{1,4}$	LM_1	LM_2	ALM_1	ALM_2
$A_{50\times50}$	9	10	6	4	4	8	4	3	3	3
$A_{100\times100}$	14	16	9	6	7	9	6	5	5	4
$A_{150\times150}$	16	17	11	7	7	13	6	5	5	5
$A_{200\times200}$	15	15	10	7	7	12	6	5	5	5
$A_{250\times250}$	16	16	10	7	7	15	6	6	5	5
$A_{300\times300}$	14	14	9	6	6	13	5	5	5	4
$A_{350\times350}$	15	16	9	7	7	12	6	5	5	4
$A_{400\times400}$	18	20	11	7	8	15	6	6	6	5
$A_{450\times450}$	19	21	12	8	8	14	7	7	5	5
$A_{500\times500}$	17	23	11	8	8	16	6	6	6	5
Mean	15.3	16.8	9.8	6.7	6.9	12.7	5.8	5.3	5	**4.5**

Numerical evidence indicates that the proposed scheme provides an effective approach for the iterative computation of the matrix sign function. Moreover, in each of the numerical examples presented above, the proposed matrix iteration significantly outperforms the existing methods in terms of convergence rate and numerical stability. We next employ the proposed matrix scheme to iteratively solve a Yang–Baxter matrix-like equation, which plays an important role in mathematical

Table 7.2: Comparisons in terms of CPU-time using various random matrices of Experiment 7.8.6

Matrix	NM	ANM	HM	SM_1	SM_2	$PM_{1,4}$	LM_1	LM_2	ALM_1	ALM_2
$A_{50\times50}$	0.06599	0.05477	0.03017	0.05627	0.04039	0.05823	0.01699	0.01441	0.02730	0.02395
$A_{100\times100}$	0.15771	0.36404	0.12112	0.14158	0.18226	0.22412	0.10187	0.09558	0.17089	0.12574
$A_{150\times150}$	0.37413	0.70010	0.45177	0.40362	0.40420	0.86365	0.24403	0.24210	0.42209	0.40058
$A_{200\times200}$	0.60484	0.99517	0.89464	0.86781	0.89233	1.55394	0.49616	0.50044	0.75972	0.84227
$A_{250\times250}$	1.18620	1.70152	1.03599	1.46507	1.40482	2.82938	0.79874	0.96901	1.34486	1.31423
$A_{300\times300}$	1.50399	2.18757	1.30565	1.79286	1.91045	3.96273	1.05604	1.33592	1.79213	1.51671
$A_{350\times350}$	2.08273	3.10377	2.02420	3.38013	3.27436	5.18537	2.11036	1.81117	2.80751	2.52464
$A_{400\times400}$	3.25383	5.38425	3.18201	4.29881	5.32141	9.21980	2.84087	3.54879	5.14803	4.33090
$A_{450\times450}$	4.07554	10.07840	4.83051	7.55678	7.12863	11.93657	4.93791	5.81708	5.22150	6.39332
$A_{500\times500}$	4.89676	11.15954	5.99030	9.85189	10.03518	19.73707	5.52582	7.37126	9.46699	8.40012
Mean	1.82017	3.57291	1.96964	2.98148	3.05940	5.55709	**1.81288**	2.17058	2.7161	2.58725

physics and quantum theory.

Experiment 7.8.7. *In this test problem, we demonstrate the computational results of the YBM equation. In this case, we apply newly proposed methods with stopping criteria* $||\mathcal{X}_n^2 - I||_2 \leq 10^{-8}$, *for computing the sign function of* H_j *corresponding to* A_j, *to solve several Y-B-like equations while*

$$\mathcal{F}_j(X) = A_j X A_j - X A_j X,\ 1 \leq j \leq 5,$$

with the following test matrices [337]:

$$A_1 = \begin{bmatrix} 10 & 7 & -8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -27+18i & -36+54i & -45+54i & -99+81i \\ -47+7i & -62+51i & -82+42i & -148+63i \\ 32-4i & 20-24i & 34-15i & 82-36i \\ 14-4i & 38-24i & 943-24i & 64-27i \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 34-133i & 34-288i & 51-339i & 85-318i \\ -80-65i & -80+10i & -120-45i & -200-90i \\ 74+62i & 74+82i & 111+146i & 185+102i \\ -26+62i & -26+82i & -39+96i & -65+152i \end{bmatrix},$$

$$A_4 = \begin{bmatrix} -87-79i & -45-169i & -315-248i & -360-428i \\ 47+40i & 24+82i & 147+122i & 168+206i \\ 54+50i & 27+104i & 192+154i & 216+262i \\ -30-29i & -15-59i & -105-88i & -117-148i \end{bmatrix},$$

$$A_5 = \begin{bmatrix} 1+3i & 9-2i & 2+3i & 7-5i \\ 5+3i & 6+7i & 11-17i & 14+3i \\ 3+4i & 9-1i & 18+5i & 6+19i \\ 7-9i & 1-8i & 10-3i & 9+8i \end{bmatrix}.$$

The residual norms $R_{j_n} = \left\| F_j\left(A_j \frac{(I+\mathcal{X}_n)}{2}\right)\right\|$ *for NM, AM (see reference [337]),* LM_1, LM_2, ALM_1,

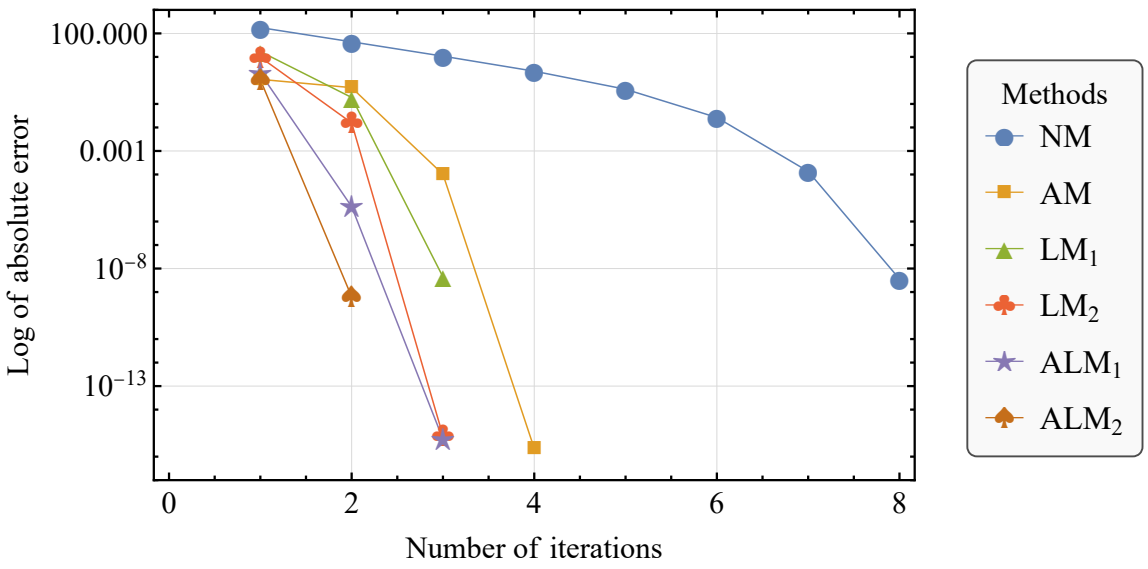


Figure 7.11: Comparison results of iteration number vs residual errors for the methods NM, AM, LM_1, LM_2, ALM_1, and ALM_2 using initial matrix H_1.

and ALM_2 *with chosen* H_j *for each YBM equation in* A_j *are shown in Table 7.3 with respect to the number of iterations required per algorithm. This illustrates how newly proposed methods are put into practice. As a result, new methods are not only a great alternative for obtaining* S *but also for*

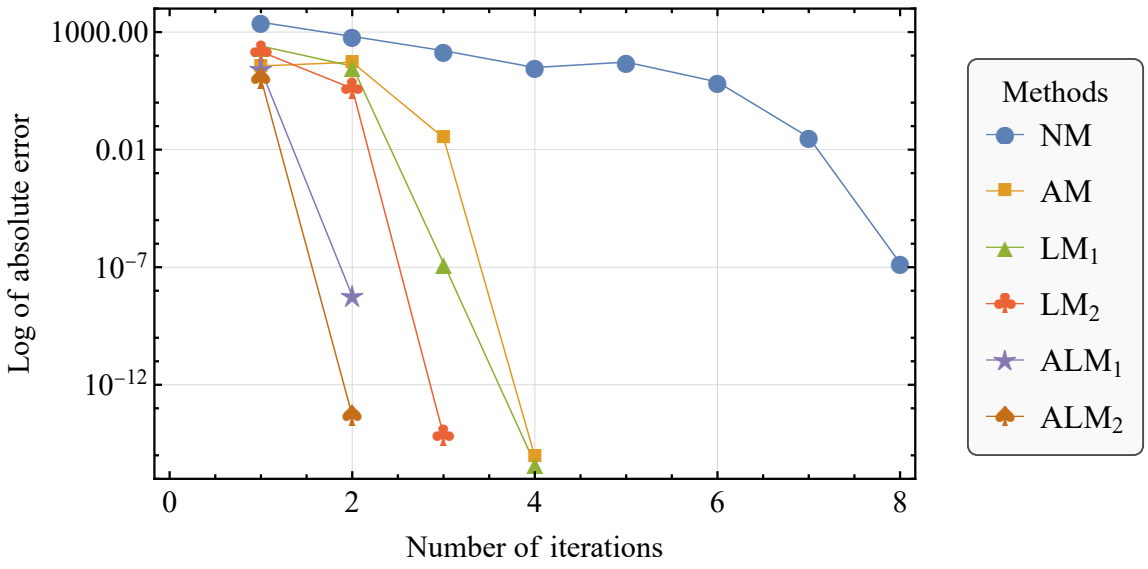


Figure 7.12: Comparison results of iteration number vs residual errors for the methods NM, AM, LM_1, LM_2, ALM_1, and ALM_2 using initial matrix H_2.

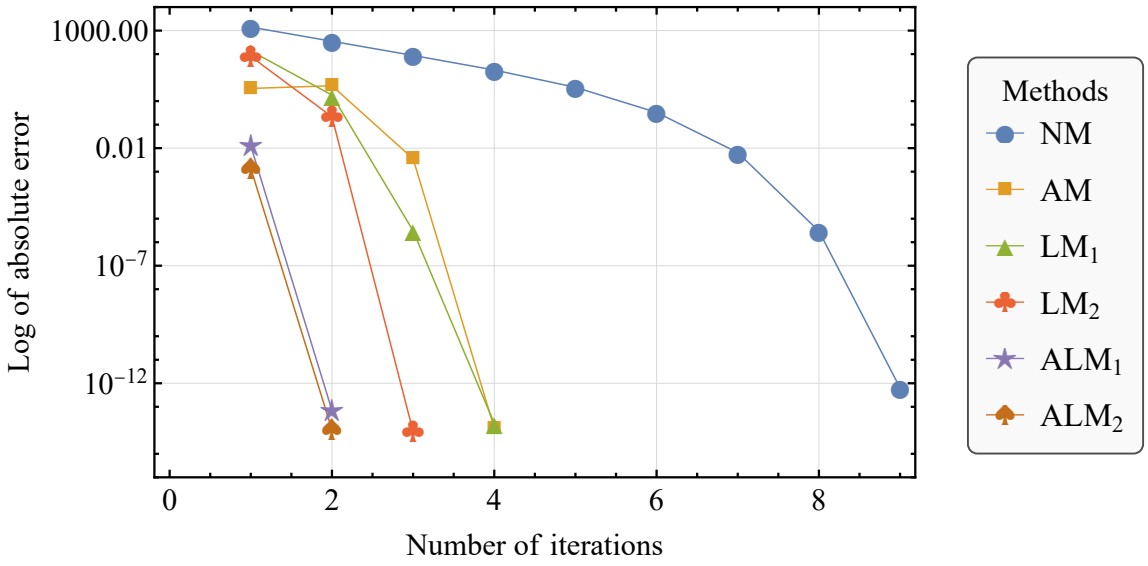


Figure 7.13: Comparison results of iteration number vs residual errors for the methods NM, AM, LM_1, LM_2, ALM_1, and ALM_2 using initial matrix H_3.

addressing YBM equation.

It should be noticed that the values of $R_{j_n} = \left\|\mathcal{F}_j\left(A_j\frac{(I-\mathcal{X}_n)}{2}\right)\right\|$ *are quite similar to those of* R_{j_n}*, thus we do not include them in Table 7.3. Also, the Figures 7.11–7.16 show numerical reports and evidence in terms of iteration number and CPU time, clearly demonstrating that NM takes longer than the other approaches.*

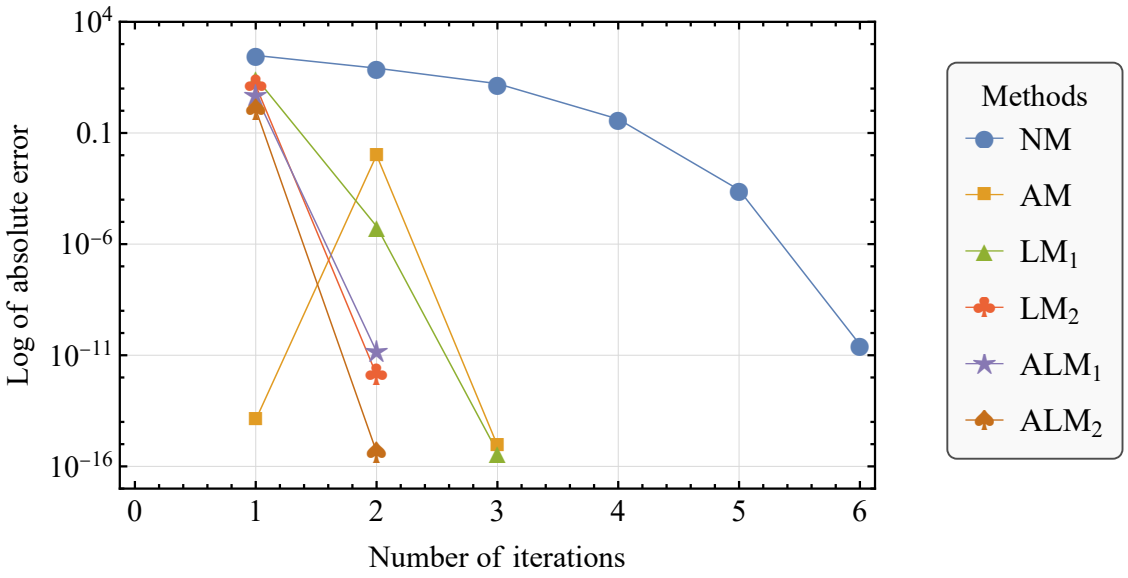


Figure 7.14: Comparison results of iteration number vs residual errors for the methods NM, AM, LM_1, LM_2, ALM_1, and ALM_2 using initial matrix H_4.

Table 7.3: Comparisons in terms of $R_{j_{n+1}}$ using H_j matrices for Experiment 7.8.7

j	H_j	NM	AM	LM_1	LM_2	ALM_1	ALM_2
1	A_j	1.29751×10^{-5}	5.14446×10^{-7}	1.13118×10^{-1}	3.42357×10^{0}	1.73675×10^{-6}	4.46391×10^{-3}
		$n = 8$	$n = 4$	$n = 3$	$n = 3$	$n = 3$	$n = 2$
2	$A_j + 7I$	0	5.14446×10^{-7}	2.08128×10^{-10}	6.59817×10^{-10}	1.34470×10^{-6}	6.16468×10^{-9}
		$n = 9$	$n = 4$	$n = 4$	$n = 3$	$n = 2$	$n = 2$
3	$\frac{1}{3j}(A_j)$	1.59968×10^{-7}	1.57423×10^{-6}	3.11998×10^{-5}	1.77969×10^{-3}	9.00670×10^{-3}	9.58247×10^{-6}
		$n = 9$	$n = 4$	$n = 4$	$n = 3$	$n = 2$	$n = 2$
4	$\frac{1}{3}(A_j + 5I) - I$	1.64076×10^{-8}	2.38788×10^{-9}	0	9.16434×10^{-9}	1.73090×10^{-8}	9.61863×10^{-9}
		$n = 6$	$n = 3$	$n = 3$	$n = 2$	$n = 2$	$n = 2$
5	$\frac{1}{5}(A_j + 11I) - I$	1.66475×10^{-10}	2.34792×10^{-7}	1.54877×10^{-6}	9.16434×10^{-9}	3.19238×10^{-8}	1.20947×10^{-6}
		$n = 9$	$n = 4$	$n = 4$	$n = 3$	$n = 3$	$n = 3$

7.9 Concluding remarks

This chapter presents novel matrix iterations for computing a sign matrix that does not have any pure imaginary eigenvalues. Convergence and stability analysis of the proposed schemes are thoroughly discussed. The suggested schemes are unique in the sense that they are not members of the Padé family or reciprocals. Furthermore, as illustrated by attraction basins, the matrix iterations have global convergence. The results are further extended to solve the nonlinear Yang–Baxter-like matrix equation. Finally, several numerical tests show that the proposed schemes are

more efficient than existing methods in terms of residual error, the number of iterations required by an algorithm, and computational time.

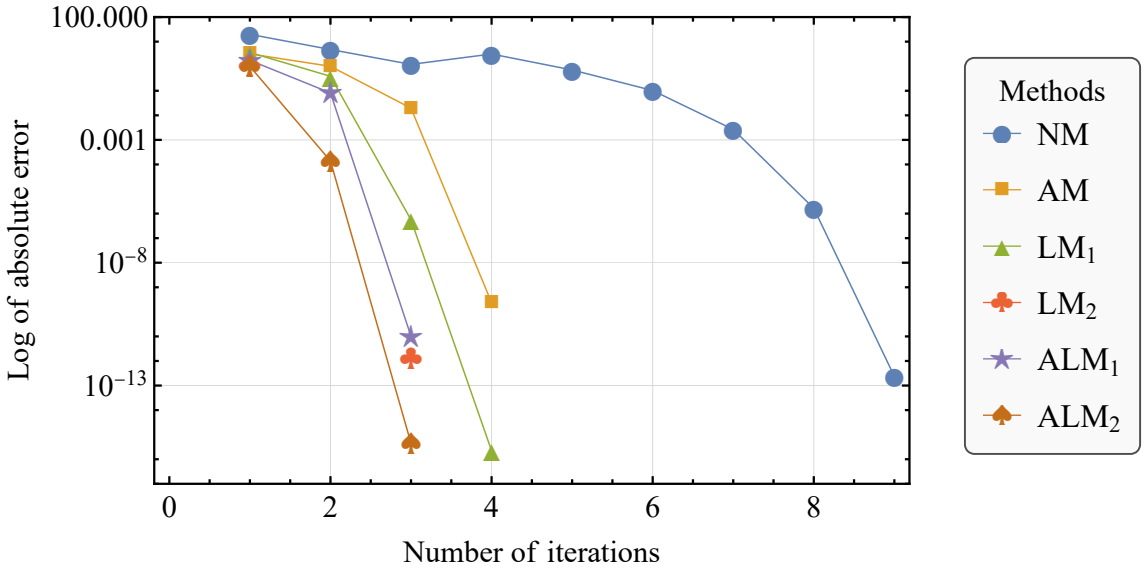


Figure 7.15: Comparison results of iteration number vs residual errors for the methods NM, AM, LM_1, LM_2, ALM_1, and ALM_2 using initial matrix H_5.

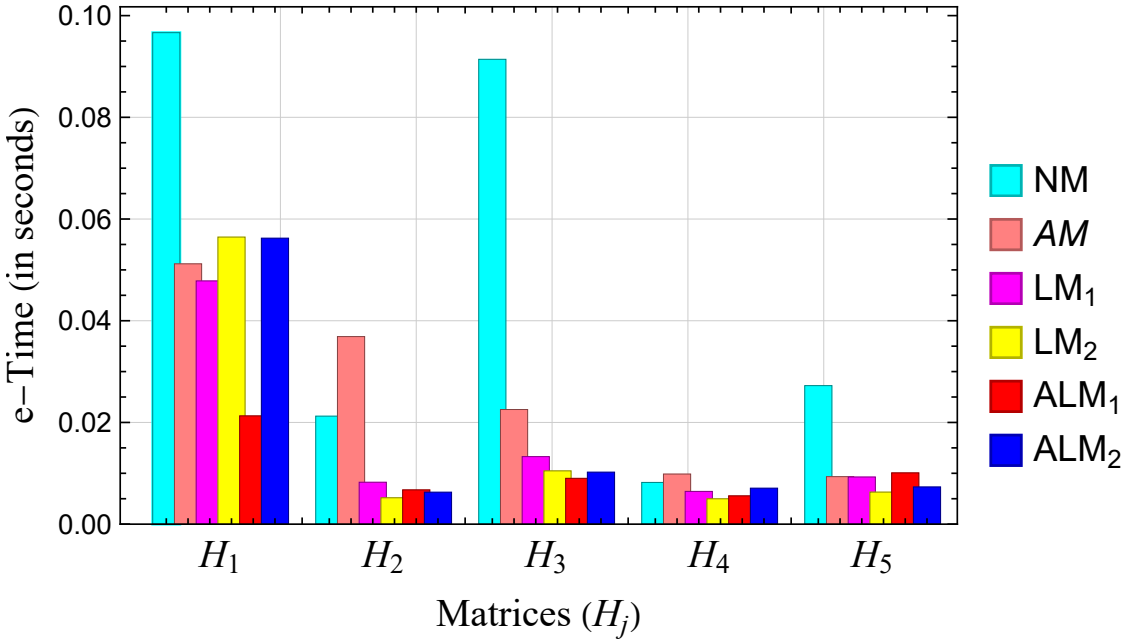


Figure 7.16: Time comparison results of the methods NM, AM, LM_1, LM_2, ALM_1, and ALM_2 using initial matrix H_j, $j = 1, 2, 3, 4, 5$.

Chapter 8

Conclusion and future scope

8.1 Conclusion

This research has methodically progressed the formulation and examination of efficient iterative techniques for resolving nonlinear equations, transitioning from scalar cases to complex matrix functions. The thesis began by illustrating the necessity for novel numerical solvers in areas where analytical solutions are impractical.

At first, the thesis introduces new derivative-free multi-point iterative families for finding multiple roots of scalar nonlinear equations. This is because classical methods have some problems. These methods use weight-function strategies to get higher-order convergence with as few function evaluations as possible. We use numerical benchmarks and dynamical analysis in the complex plane through basins of attraction to make sure that they work better than other methods in terms of accuracy, stability, and computational efficiency.

Based on these outcomes, the research then moved on to nonlinear systems. New multi-step iterative schemes were developed within a Banach space setting, facilitating a stringent convergence theory studied exclusively on first-order derivatives. These approaches use only one inverse operator at each step, which helps keep costs down while still achieving high-order convergence. When applied to academic and practical problems, such as boundary value problems, Hammerstein integral equations, BVPs, and PDEs, the proposed solvers exhibit much better convergence rates and residual errors, especially for large-scale systems. The final section of the study utilized similar iterative principles on the matrix functions, with an emphasis on the matrix sign function and related matrix equations. We constructed globally convergent matrix iterations that give higher-order convergence without employing Padé approximants.

In short, this thesis presents a set of iterative techniques that improve both the theory and practice of solving nonlinear problems at different levels, from scalar roots to systems of equations and matrix functions. The proposed methods are not only theoretically sound and computationally efficient, but they are also suitable for a wide range of problems in scientific computing and engineering.

8.2 Future scope

The findings of this thesis naturally set a basis for various directions for future research. The following points will be explored: The derivative-based or derivative-free scalar iterative families established here provide a way for several advancements:

- Developing with memory iterative methods that compute multiple roots, possibly using different divided-difference techniques to make them more efficient without adding more function evaluations.
- A thorough convergence analysis of higher-order Newton-like methods in Banach spaces via recurrence relations would strengthen their theoretical part.
- Using tools from complex dynamics to do a detailed dynamical analysis to find the best parameter ranges that make stability and basin of attraction characteristics as high as possible.

The multi-step iterative schemes for nonlinear systems can be extended in different ways:

- Incorporating memory-based techniques to increase the convergence order while maintaining computational efficiency.
- Conduct local and semi-local convergence analyses of the developed classes based on various assumptions in Banach spaces.
- The goal of developing iterative algorithms for differential equations is to make them more efficient than traditional Runge-Kutta-like methods and easier to scale up for use in high-dimensional settings.

The iterative methods developed for the matrix sign function have new extensions for applications in numerical linear algebra:

- Extend the proposed schemes to calculate associated matrix functions, including the matrix square root, matrix pth root, or polar decomposition, utilizing parallel computing software.
- Investigating global convergence characteristics for inverse-free iterations utilized on an expanded set of matrix equations.

- Study into how matrix functions can make stochastic or time-dependent matrix problems more robust for real-world simulations.

Each of these directions widens the theoretical concept while also making it more useful in real-world scientific computing and engineering applications.

Bibliography

[1] M. Abad, A. Cordero and J. R. Torregrosa, A family of seventh-order schemes for solving nonlinear systems, Bull. Math. Soc. Sci. Math. Roumanie 57(105) (2014), no. 2, 133–145.

[2] K. Abbaoui and Y. Cherruault, Convergence of Adomian's method applied to nonlinear equations, Math. Comput. Model. 20 (1994), no. 9, 69–73.

[3] S. Abbasbandy, Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method, Appl. Math. Comput. 145 (2003), no. 2-3, 887–893.

[4] S. Abdullah, N. Choubey and S. Dara, An efficient two-point iterative method with memory for solving non-linear equations and its dynamics, J. Appl. Math. Comput. 70 (2024), no. 1, 285–315.

[5] S. Abdullah, N. Choubey, S. Dara, M. Kansal and M. U. D. Junjua, On memory-based higher-order iterative schemes for nonlinear equations: Complex dynamics and applications, Eng. Comput. (2025), 1–38.

[6] N. H. Abel, Beweis der unmöglichkeit, algebraische gleichungen von höheren graden als dem vierten allgemein aufzulösen, J. Reine Angew Math. 1 (1826), no. 1, 65–84.

[7] G. Adomian and R. Rach, On the solution of algebraic equations by the decomposition method, J. Comput. Appl. Math. 105 (1985), no. 1, 141–166.

[8] G. Adomian, Nonlinear Stochastic Systems Theory and Applications to Physics, Vol. 46, Springer Science & Business Media, 1988.

[9] G. Adomian, Solving Frontier Problems of Physics: The Decomposition Method, Vol. 60, Springer Science & Business Media, 2013.

[10] L. V. Ahlfors, Complex Analysis, McGraw-Hill, Inc., New York, USA, 1979.

[11] F. Ahmad, F. Soleymani, F. K. Haghani and S. Serra-Capizzano, Higher order derivative-free iterative methods with and without memory for systems of nonlinear equations, Appl. Math. Comput. 314 (2017), 199–211.

[12] S. Akram, F. Zafar and N. Yasmin, An optimal eighth-order family of iterative methods for multiple roots, Mathematics 7 (2019), no. 8, Article ID 672.

[13] R. A. Alharbey, M. Kansal, R. Behl and J. A. T. Machado, Efficient three-step class of eighth-order multiple root solvers and their dynamics, Symmetry 11 (2019), no. 7, Article ID 837.

[14] F. Ali, W. Aslam, I. Khalid and A. Nadeem, Iteration methods with an auxiliary function for nonlinear equations, J. Math. 2020 (2020), no. 1, Article ID 7356408.

[15] H. F. Alqahtani, R. Behl and M. Kansal, Higher-order iteration schemes for solving nonlinear systems of equations, Mathematics 7 (2019), no. 10, Article ID 937.

[16] S. Amat, S. Busquier and J. M. Gutiérrez, Geometric constructions of iterative functions to solve nonlinear equations, J. Comput. Appl. Math. 157 (2003), no. 1, 197–205.

[17] S. Amat, S. Busquier and S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, Scientia 10 (2004), no. 3, Article ID 35.

[18] A. R. Amiri, A. Cordero, M. T. Darvishi and J. R. Torregrosa, Preserving the order of convergence: Low-complexity Jacobian-free iterative schemes for solving nonlinear systems, Comp. Appl. Math. 337 (2018), 87–97.

[19] I. K. Argyros, M. Kansal and V. Kanwar, Ball convergence of a stable fourth-order family for solving nonlinear systems under weak conditions, Stud. Univ. Babeș-Bolyai, Math. 62 (2017), no. 1, 127–135.

[20] I. K. Argyros, M. Kansal and V. Kanwar, Local convergence for multipoint methods using only the first derivative, SeMA J. 73 (2016), 369–378.

[21] I. K. Argyros, M. Kansal, V. Kanwar and S. Bajaj, Higher-order derivative-free families of Chebyshev–Halley type methods with or without memory for solving nonlinear equations, Appl. Math. Comput. 315 (2017), 224–245.

[22] I. K. Argyros and Á. A. Magreñán, Iterative Methods and their Dynamics with Applications: A Contemporary Study, CRC Press, New York, USA, 2017.

[23] W. F. Arnold and A. J. Laub, Generalized eigenproblem algorithms and software for algebraic Riccati equations, Proceedings of the IEEE 72 (1984), no. 12, 1746–1754.

[24] H. Arora, A. Cordero and J. R. Torregrosa, On generalized one-step derivative-free iterative family for evaluating multiple roots, Iran. J. Numer. Anal. Optim. 14 (2024), no. 1, 291–314.

[25] V. Arroyo, A. Cordero and J. R. Torregrosa, Approximation of artificial satellites' preliminary orbits: The efficiency challenge, Math. Comput. Model. 54 (2011), no. 7-8, 1802–1807.

[26] S. Artidiello, A. Cordero, J. R. Torregrosa and M. P. Vassileva, Design of high-order iterative methods for nonlinear systems by using weight function procedure, In: Abstr. Appl. Anal., Vol. 2015, Wiley Online Library, 2015, Article ID 289029.

[27] P. Ataei Delshad and T. Lotfi, Developing a stable method for computing the matrix sign function with applications to algebraic Riccati and Sylvester equations, Int. J. Ind. Math. 14 (2022), no. 2, 227–237.

[28] K. Atkinson, D. Chien and O. Hansen, Multivariate quadrature of a singular integrand, Adv. Comput. Math. 47 (2021), no. 3, 44.

[29] K. Atkinson and W. Han, Theoretical Numerical Analysis: A Functional Analysis Framework, Vol. 15, Springer, New York, 2005.

[30] K. Atkinson, W. Han and D. E. Stewart, Numerical Solution of Ordinary Differential Equations, John Wiley & Sons, USA, 2009.

[31] K. Atkinson and O. Hansen, Solving the nonlinear Poisson equation on the unit disk, J. Integral Equ. Appl. (2005), 223–241.

[32] K. Atkinson, O. Hansen and D. Chien, A spectral method for parabolic differential equations, Numer. Algorithms 63 (2013), no. 2, 213–237.

[33] D. K. R. Babajee, A. Cordero, F. Soleymani and J. R. Torregrosa, On a novel fourth-order algorithm for solving systems of nonlinear equations, J. Appl. Math. 2012 (2012), no. 1, Article ID 165452.

[34] D. K. R. Babajee, Analysis of higher order variants of Newton's method and their applications to differential and integral equations and in ocean acidification, Ph.D. thesis, University of Mauritius (2010).

[35] D. Bachiller and F. Cedó, A family of solutions of the Yang–Baxter equation, J. Algebra 412 (2014), 218–229.

[36] Z. Bai and J. Demmel, Using the matrix sign function to compute invariant subspaces, SIAM J. Matrix Anal. Appl. 19 (1998), no. 1, 205–225.

[37] Z. Bai and J. W. Demmel, Design of a Parallel Nonsymmetric Eigenroutine Toolbox, University of Kentucky, Department of Mathematics, United States, 1992.

[38] Z. Bai, J. Demmel and M. Gu, An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems, Numer. Math. 76 (1997), no. 3, 279–308.

[39] A. Baid, K. Ola, S. Bhoi, J. Kumar and E. Tsotsas, A weighted Monte Carlo method for modeling aggregation breakage phenomena: Application to Al2O3 nanoparticle aggregates, Powder Technol. 471 (2026), Article ID 122110.

[40] R. Bala, M. Kansal and V. Kanwar, An optimal class of fourth-order multiple-root finders of Chebyshev-Halley type and their basins of attraction, Int. J. Comput. Sci. Math. 14 (2021), no. 1, 17–35.

[41] J. Banks, J. Garza-Vargas, A. Kulkarni and N. Srivastava, Pseudospectral shattering, the sign function, and diagonalization in nearly matrix multiplication time, Found. Comput. Math. 23 (2023), no. 6, 1959–2047.

[42] U. Baur and P. Benner, Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic, Computing 78 (2006), 211–234.

[43] R. J. Baxter, Partition function of the eight-vertex lattice model, Ann. Phys. 70 (1972), no. 1, 193–228.

[44] R. Behl and W. M. Al-Hamdan, A 4th-order optimal extension of Ostrowski's method for multiple zeros of univariate nonlinear functions, Mathematics 7 (2019), no. 9, Article ID 803.

[45] R. Behl, S. K. Alharbi, F. O. Mallawi and M. Salimi, An optimal derivative-free Ostrowski's scheme for multiple roots of nonlinear equations, Mathematics 8 (2020), no. 10, Article ID 1809.

[46] R. Behl, A. S. Alshomrani and S. S. Motsa, An optimal scheme for multiple roots of nonlinear equations with eighth-order convergence, J. Math. Chem. 56 (2018), 2069–2084.

[47] R. Behl, S. Bhalla, E. Martínez and M. A. Alsulami, Derivative-free King's scheme for multiple zeros of nonlinear functions, Mathematics 9 (2021), no. 11, Article ID 1242.

[48] R. Behl, A. Cordero and J. R. Torregrosa, High order family of multivariate iterative methods: Convergence and stability, J. Comput. Appl. Math. 405 (2022), Article ID 113053.

[49] R. Behl, A. Cordero and J. R. Torregrosa, A new higher-order optimal derivative free scheme for multiple roots, J. Comput. Appl. Math. 404 (2022), Article ID 113773.

[50] R. Behl, A. Cordero, J. R. Torregrosa and S. Bhalla, A new high-order Jacobian-free iterative method with memory for solving nonlinear systems, Mathematics 9 (2021), no. 17, Article ID 2122.

[51] R. Behl, M. Kansal and M. Salimi, Modified King's family for multiple zeros of scalar nonlinear functions, Mathematics 8 (2020), no. 5, Article ID 827.

[52] R. Behl, E. Martínez, F. Cevallos and D. Alarcón, A higher order Chebyshev-Halley-type family of iterative methods for multiple roots, Mathematics 7 (2019), no. 4, Article ID 339.

[53] R. Behl, F. Zafar, A. S. Alshormani, M. U. D. Junjua and N. Yasmin, An optimal eighth-order scheme for multiple zeros of univariate functions, Int. J. Comput. Methods 16 (2019), no. 04, Article ID 1843002.

[54] A. Ben-Israel and T. N. E. Greville, Generalized Inverses: Theory and Applications, Vol. 15, Springer Science & Business Media, New York, 2003.

[55] P. Benner, Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems, Ph.D. thesis, Logos-Verlag Berlin (1997).

[56] P. Benner and E. S. Quintana-Ortí, Solving stable generalized Lyapunov equations with the matrix sign function, Numer. Algorithms 20 (1999), 75–100.

[57] R. Bhatia, Positive Definite Matrices, Princeton Series in Applied Mathematics, Princeton University Press, Princeton, New Jersey, United Kingdom (2007).

[58] Bhavna and S. Bhatia, Derivative free King's-type family for solving system of nonlinear equations, Int. J. Comput. Methods 22 (2025), no. 1, Article ID 2450040.

[59] G. J. Bierman, Computational aspects of the matrix sign function solution to the ARE, In: The 23rd IEEE Conference on Decision and Control, IEEE, 1984, 514–519.

[60] H. Bin Jebreen, Constructing a high-order globally convergent iterative method for calculating the matrix sign function, Math. Probl. Eng. 2018 (2018), no. 1, Article ID 8973867.

[61] D. Bini and B. Iannazzo, On the numerical solution of the matrix equation $\sum_{i=1}^{k} \log(XA^{-1}) = 0$, In: 16-th ILAS Conference, Pisa, June, 2010, 21–25.

[62] E. Bodewig, Sur la méthode laguerre pour l'approximation des racines de certaines équations algébriques et sur la critique d'hermite, Indag. Math. 8 (1946), 570–580.

[63] B. Bradie, A Friendly Introduction to Numerical Analysis, Pearson Education, New Delhi, India, 2006.

[64] Z. Bubnicki, Modern Control Theory, Springer-Verlag, Berlin Heidelberg, Germany, 2005.

[65] X. Buff and C. Henriksen, On König's root-finding algorithms, Nonlinearity 16 (2003), no. 3, Article ID 989.

[66] R. L. Burden and J. D. Faires, Numerical Analysis, PWS Publishing Company, Boston, 2001.

[67] J. C. Butcher, A history of Runge-Kutta methods, Appl. Numer. Math. 20 (1996), no. 3, 247–260.

[68] R. Byers, Solving the algebraic Riccati equation with the matrix sign function, Linear Algebra Appl. 85 (1987), 267–279.

[69] R. P. Canale and S. C. Chapra, Numerical Methods for Engineers, Mc Graw Hill Publications, New York, USA, 2014.

[70] Y. Cao and X. Wang, A Newton-type iterative method for computing matrix sign function, In: 2022 International Conference on Cloud Computing, Big Data and Internet of Things (3CBIT), IEEE, 2022, 359–361.

[71] P. Cayley, The Newton-Fourier imaginary problem, Amer. J. Math. 2 (1879), no. 1, 97–97.

[72] C. W. Chang, S. Qureshi, I. K. Argyros, F. I. Chicharro and A. Soomro, A modified two-step optimal iterative method for solving nonlinear models in one and higher dimensions, Math. Comput. Simul. 229 (2025), 448–467.

[73] S. H. Cheng, N. J. Higham, C. S. Kenney and A. J. Laub, Approximating the logarithm of a matrix to specified accuracy, SIAM J. Matrix Anal. Appl. 22 (2001), no. 4, 1112–1125.

[74] F. I. Chicharro, A. Cordero, N. Garrido and J. R. Torregrosa, On the improvement of the order of convergence of iterative methods for solving nonlinear systems by means of memory, Appl. Math. Lett. 104 (2020), Article ID 106277.

[75] F. Chicharro, A. Cordero, J. M. Gutiérrez and J. R. Torregrosa, Complex dynamics of derivative-free methods for nonlinear equations, Appl. Math. Comput. 219 (2013), no. 12, 7023–7035.

[76] G. D. Christian, P. K. Dasgupta and K. A. Schug, Analytical Chemistry, John Wiley & Sons, USA, 2013.

[77] C. Chun, A geometric construction of iterative functions of order three to solve nonlinear equations, Comput. Math. Appl. 53 (2007), no. 6, 972–976.

[78] C. Chun, Iterative methods improving Newton's method by the decomposition method, Comput. Math. Appl. 50 (2005), no. 10-12, 1559–1568.

[79] C. Chun and B. Neta, A new sixth-order scheme for nonlinear equations, Appl. Math. Lett. 25 (2012), no. 2, 185–189.

[80] C. Chun and B. Neta, A third-order modification of Newton's method for multiple roots, Appl. Math. Comput. 211 (2009), no. 2, 474–479.

[81] C. Chun and B. Neta, Developing high order methods for the solution of systems of nonlinear equations, Appl. Math. Comput. 342 (2019), 178–190.

[82] E. A. Coddington, N. Levinson and T. Teichmann, Theory of Ordinary Differential Equations, McGraw–Hill, New York, 1955.

[83] A. Constantinides and N. Mostoufi, Numerical Methods for Chemical Engineers with Matlab Applications with Cdrom, Prentice Hall PTR, 1999.

[84] A. Cordero, E. Gómez and J. R. Torregrosa, Efficient high-order iterative methods for solving nonlinear systems and their application on heat conduction problems, Complexity 2017 (2017), Article ID 6457532.

[85] A. Cordero, F. Soleymani and J. R. Torregrosa, Dynamical analysis of iterative methods for nonlinear systems or how to deal with the dimension?, Appl. Math. Comput. 244 (2014), 398–412.

[86] A. Cordero, F. Soleymani, J. R. Torregrosa and M. Z. Ullah, Numerically stable improved Chebyshev–Halley type schemes for matrix sign function, J. Comput. Appl. Math. 318 (2017), 189–198.

[87] A. Cordero, F. Soleymani, J. R. Torregrosa and M. Zaka Ullah, Approximating the matrix sign function by means of Chebyshev–Halley type method, In: Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering, 2016, 396–405.

[88] A. Cordero, N. Garrido, J. R. Torregrosa and P. Triguero-Navarro, Modifying Kurchatov's method to find multiple roots of nonlinear equations, Appl. Numer. Math. 198 (2024), 11–21.

[89] A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa, A modified Newton-Jarratt's composition, Numer. Algorithms 55 (2010), 87–99.

[90] A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa, Accelerated methods of order 2p for systems of nonlinear equations, J. Comput. Appl. Math. 233 (2010), no. 10, 2696–2702.

[91] A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa, Increasing the convergence order of an iterative method for nonlinear systems, Appl. Math. Lett. 25 (2012), no. 12, 2369–2374.

[92] A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa, Efficient high-order methods based on golden ratio for nonlinear systems, Appl. Math. Comput. 217 (2011), no. 9, 4548–4556.

[93] A. Cordero, J. R. Torregrosa and M. P. Vassileva, A family of parametric schemes of arbitrary even order for solving nonlinear models: CMMSE2016, J. Math. Chem. 55 (2017), 1443–1460.

[94] A. Cordero, M. Kansal, V. Kanwar and J. R. Torregrosa, A stable class of improved second-derivative free Chebyshev-Halley type methods with optimal eighth order convergence, Numer. Algorithms 72 (2016), no. 4, 937–958.

[95] A. Cordero, M. A. Leonardo-Sepúlveda, J. R. Torregrosa Sánchez and M. P. Vassileva, Increasing in three units the order of convergence of iterative methods for solving nonlinear systems, Math. Comput. Simul. 223 (2024), 509–522.

[96] A. Cordero, J. G. Maimó, A. Rodríguez-Cabral and J. R. Torregrosa, Two-step fifth-order efficient Jacobian-free iterative method for solving nonlinear systems, Mathematics 12 (2024), no. 21, Article ID 3341.

[97] A. Cordero, J. G. Maimó, J. R. Torregrosa and M. P. Vassileva, Solving nonlinear problems by Ostrowski–Chun type parametric families, J. Math. Chem. 53 (2015), no. 1, 430–449.

[98] A. Cordero, B. Neta and J. R. Torregrosa, Memorizing Schröder's method as an efficient strategy for estimating roots of unknown multiplicity, Mathematics 9 (2021), no. 20, Article ID 2570.

[99] A. Cordero, R. V. Rojas-Hiciano, J. R. Torregrosa and M. P. Vassileva, A highly efficient class of optimal fourth-order methods for solving nonlinear systems, Numer. Algorithms 95 (2024), no. 4, 1879–1904.

[100] A. Cordero, F. Soleymani, J. R. Torregrosa and S. Shateyi, Basins of attraction for various Steffensen-type methods, J. Appl. Math. 2014 (2014), no. 1, Article ID 539707.

[101] A. Cordero and J. R. Torregrosa, Variants of Newton's method using fifth-order quadrature formulas, Appl. Math. Comput. 190 (2007), no. 1, 686–698.

[102] A. Cordero, J. R. Torregrosa and P. Triguero-Navarro, First optimal vectorial eighth-order iterative scheme for solving non-linear systems, Appl. Math. Comput. 498 (2025), Article ID 129401.

[103] A. Cordero, J. R. Torregrosa and M. P. Vassileva, Increasing the order of convergence of iterative schemes for solving nonlinear systems, J. Comput. Appl. Math. 252 (2013), 86–94.

[104] A. Cordero and J. R. Torregrosa, On interpolation variants of Newton's method for functions of several variables, J. Comput. Appl. Math. 234 (2010), no. 1, 34–43.

[105] E. Cătinaş, A survey on the high convergence orders and computational convergence orders of sequences, Appl. Math. Comput. 343 (2019), 1–20.

[106] G. Dahlquist and Å. Björck, Numerical Methods in Scientific Computing, Volume I, SIAM, Philadelphia, 2008.

[107] N. Das, R. Singh, A. M. Wazwaz and J. Kumar, An algorithm based on the variational iteration technique for the Bratu-type and the Lane–Emden problems, J. Math. Chem. 54 (2016), no. 2, 527–551.

[108] M. Dehghan and A. Shirilord, Three-step iterative methods for numerical solution of systems of nonlinear equations, Eng. Comput. 38 (2022), no. 2, 1015–1028.

[109] P. A. Delshad and T. Lotfi, Developing a stable method for computing the matrix sign function with applications to algebraic Riccati and Sylvester equations, Int. J. Ind. Math. 14 (2022), no. 2, DOI: 10.30495/ijim.2022.19413.

[110] E. D. Denman and A. N. Beavers Jr, The matrix sign function and computations in systems, Appl. Math. Comput. 2 (1976), no. 1, 63–94.

[111] J. Ding and N. H. Rhee, A nontrivial solution to a stochastic matrix equation, East Asian J. Appl. Math. 2 (2012), no. 4, 277–284.

[112] J. Ding and N. H. Rhee, Computing solutions of the Yang-Baxter-like matrix equation for diagonalisable matrices, East Asian J. Appl. Math. 5 (2015), no. 1, 75–84.

[113] C. Dong, A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation, Math. Numerica Sinica 11 (1982), 445–450.

[114] C. Dong, A family of multiopoint iterative functions for finding multiple roots of equations, Int. J. Comput. Methods 21 (1987), no. 3-4, 363–367.

[115] J. Dong and X. Wang, A globally convergent nth-order iterative method for computing the matrix sign function, In: 2022 International Conference on Cloud Computing, Big Data and Internet of Things (3CBIT), IEEE, 2022, 15–18.

[116] Q. Dong and J. Ding, Complete commuting solutions of the Yang–Baxter-like matrix equation for diagonalizable matrices, Comput. Math. Appl. 72 (2016), no. 1, 194–201.

[117] J. M. Douglas, Process Dynamics and Control, Prentice Hall, Englewood Cliffs, NJ (1972), .

[118] V. G. Drinfeld, On some unsolved problems in quantum group theory, In: Quantum Groups: Proceedings of Workshops held in the Euler International Mathematical Institute, Leningrad, Fall 1990, Springer, 2006, 1–8.

[119] R. Erfanifar and M. Hajarian, On sign function of tensors with Einstein product and its application in solving Yang–Baxter tensor equation, Comp. Appl. Math. 43 (2024), no. 6, Article ID 373.

[120] R. Erfanifar, K. Sayevand and M. Hajarian, Convergence theory of efficient parametric iterative methods for solving the Yang-Baxter-like matrix equation, Eng. Comput. 42 (2025), no. 1, 255–276.

[121] M. Fasi, N. J. Higham and B. Iannazzo, An algorithm for the matrix Lambert W function, SIAM J. Matrix Anal. Appl. 36 (2015), no. 2, 669–685.

[122] M. Fasi and B. Iannazzo, Computing the weighted geometric mean of two large-scale matrices and its inverse times a vector, SIAM J. Matrix Anal. Appl. 39 (2018), no. 1, 178–203.

[123] H. Fatoorehchi and S. Djilali, Stability analysis of linear time-invariant dynamic systems using the matrix sign function and the Adomian decomposition method, Int. J. Dyn. Control 11 (2023), no. 2, 593–604.

[124] M. Ferrara, S. Sharifi and M. Salimi, Computing multiple zeros by using a parameter in Newton–Secant method, SeMA J. 74 (2017), no. 4, 361–369.

[125] F. D. Filbir, Computation of the structured stability radius via matrix sign function, Syst. Control Lett. 22 (1994), no. 5, 341–349.

[126] H. B. Fine, On Newton's method of approximation, Proceedings of the National Academy of Sciences 2 (1916), no. 9, 546–552.

[127] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier and P. Zimmermann, MPFR: A multiple-precision binary floating-point library with correct rounding, ACM Trans. Math. Softw. 33 (2007), no. 2, Article ID 13.

[128] D. A. Frank-Kamenetskii, Diffusion and Heat Exchange in Chemical Kinetics, Princeton University Press, London, 2015.

[129] M. Frontini and E. Sormani, Third-order methods from quadrature formulae for solving systems of nonlinear equations, Appl. Math. Comput. 149 (2004), no. 3, 771–782.

[130] J. D. Gardiner, A stabilized matrix sign function algorithm for solving algebraic Riccati equations, SIAM J. Sci. Comput. 18 (1997), no. 5, 1393–1411.

[131] J. D. Gardiner and A. J. Laub, A generalization of the matrix-sign-function solution for algebraic Riccati equations, Int. J. Control 44 (1986), no. 3, 823–832.

[132] A. Genocchi, Relation entre la différence et la dérivée d'un même ordre quelconque, Archives Math. Phys. (I) 49 (1869), 342–345.

[133] Y. H. Geum and Y. I. Kim, A family of optimal sixteenth-order multipoint methods with a linear fraction plus a trivariate polynomial as the fourth-step weighting function, Comput. Math. Appl. 61 (2011), no. 11, 3278–3287.

[134] Y. H. Geum, Y. I. Kim and B. Neta, A class of two-point sixth-order multiple-zero finders of modified double-Newton type and their dynamics, Appl. Math. Comput. 270 (2015), 387–400.

[135] Y. H. Geum, Y. I. Kim and B. Neta, A sixth-order family of three-point modified Newton-like multiple-root finders and the dynamics behind their extraneous fixed points, Appl. Math. Comput. 283 (2016), 120–140.

[136] W. J. Gilbert, The complex dynamics of Newton's method for multiple roots, Comput. Graph. 18 (1994), 227–229.

[137] J. Golzarpoor, D. Ahmed and S. Shateyi, Constructing a matrix mid-point iterative method for matrix square roots and applications, Mathematics 10 (2022), no. 13, Article ID 2200.

[138] O. Gomilko, F. Greco and K. Zietak, A Padé family of iterations for the matrix sign function and related problems, Numer. Linear Algebra Appl. 19 (2012), no. 3, 585–605.

[139] W. B. Gragg and R. A. Tapia, Optimal error bounds for the Newton–Kantorovich theorem, SIAM J. Numer. Anal. 11 (1974), no. 1, 10–13.

[140] L. Grasedyck, W. Hackbusch and B. N. Khoromskij, Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices, Computing 70 (2003), 121–165.

[141] M. Grau-Sánchez, M. Noguera and S. Amat, On the approximation of derivatives using divided difference operators preserving the local convergence order of iterative methods, J. Comput. Appl. Math. 237 (2013), no. 1, 363–372.

[142] M. Grau-Sánchez, À. Grau and M. Noguera, On the computational efficiency index and some iterative methods for solving systems of nonlinear equations, J. Comput. Appl. Math. 236 (2011), no. 6, 1259–1266.

[143] J. Guerra, M. Yagoubi and P. Chevrel, A matrix sign function based solution of parameter dependent Sylvester equations, In: 2014 European Control Conference (ECC), IEEE, 2014, 400–405.

[144] Y. Guo, Q. Wu and X. Wang, An extension of high-order Kou's method for solving nonlinear systems and its stability analysis, Electron. Res. Arch. 33 (2025), no. 3, 1566–1588.

[145] K. C. Gupta, V. Kanwar and S. Kumar, A family of ellipse methods for solving non-linear equations, Int. J. Math. Educ. Sci. Technol. 40 (2009), no. 4, 571–575.

[146] J. M. Gutierrez and M. A. Hernández, A family of Chebyshev-halley type methods in Banach spaces, Bull. Aust. Math. Soc. 55 (1997), no. 1, 113–130.

[147] F. K. Haghani, A generalized Steffensen's method for matrix sign function, Appl. Math. Comput. 260 (2015), 249–256.

[148] E. Hansen and M. Patrick, A family of root finding methods, Numer. Math. 27 (1976), no. 3, 257–269.

[149] D. C. Harris, Quantitative Chemical Analysis, Macmillan, 2010.

[150] M. A. Hasan, A. A. Hasan and K. B. Ejaz, Computation of matrix nth roots and the matrix sector function, In: Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228), Vol. 5, IEEE, 2001, 4057–4062.

[151] H. V. Henderson and S. R. Searle, On deriving the inverse of a sum of matrices, SIAM Rev. 23 (1981), no. 1, 53–60.

[152] J. Hietarinta, Solving the two-dimensional constant quantum Yang–Baxter equation, J. Math. Phys. 34 (1993), no. 5, 1725–1756.

[153] N. J. Higham, D. S. Mackey, N. Mackey and F. Tisseur, Functions preserving matrix groups and iterations for the matrix square root, SIAM J. Matrix Anal. Appl. 26 (2005), no. 3, 849–877.

[154] N. J. Higham, Functions of Matrices: Theory and Computation, SIAM, Philadelphia, 2008.

[155] N. J. Higham, The matrix sign decomposition and its relation to the polar decomposition, Linear Algebra Appl. 212 (1994), 3–20.

[156] N. J. Higham, D. S. Mackey, N. Mackey and F. Tisseur, Computing the polar decomposition and the matrix sign decomposition in matrix groups, SIAM J. Matrix Anal. Appl. 25 (2004), no. 4, 1178–1192.

[157] H. H. H. Homeier, A modified Newton method for rootfinding with cubic convergence, J. Comput. Appl. Math. 157 (2003), no. 1, 227–230.

[158] H. H. H. Homeier, On Newton-type methods with cubic convergence, J. Comput. Appl. Math. 176 (2005), no. 2, 425–432.

[159] H. H. H. Homeier, On Newton-type methods for multiple roots with cubic convergence, J. Comput. Appl. Math. 231 (2009), no. 1, 249–254.

[160] W. D. Hoskins and D. J. Walton, A faster, more stable method for computing the pth roots of positive definite matrices, Linear Algebra Appl. 26 (1979), 139–163.

[161] H. Hotelling, Some new methods in matrix calculation, Ann. Math. Statist. 14 (1943), no. 1, 1–34.

[162] J. L. Howland, The sign matrix and the separation of matrix eigenvalues, Linear Algebra Appl. 49 (1983), 221–232.

[163] C. J. Huang and L. S. Shieh, Modelling large dynamical systems with industrial specifications, Int. J. Syst. Sci. 7 (1976), no. 3, 241–256.

[164] C. C. Huang, J. S. H. Tsai, S. M. Guo, Y. J. Sun and L. S. Shieh, Solving algebraic Riccati equation for singular system based on matrix sign function, Int. J. Innov. Comput. Inf. Control 9 (2013), 2771–2788.

[165] J. L. Hueso, E. Martínez and C. Teruel, Determination of multiple roots of nonlinear equations and applications, J. Math. Chem. 53 (2015), no. 3, 880–892.

[166] J. L. Hueso, E. Martínez and C. Teruel, Convergence, efficiency and dynamics of new fourth and sixth order families of iterative methods for nonlinear systems, J. Comput. Appl. Math. 275 (2015), 412–420.

[167] B. Iannazzo, The geometric mean of two matrices from a computational viewpoint, Numer. Linear Algebra Appl. 23 (2016), no. 2, 208–229.

[168] B. Iannazzo, A family of rational iterations and its application to the computation of the matrix Pth root, SIAM J. Matrix Anal. Appl. 30 (2009), no. 4, 1445–1462.

[169] B. Iannazzo, A note on computing the matrix square root, Calcolo 40 (2003), no. 4, 273–283.

[170] B. Iannazzo and B. Meini, Palindromic matrix polynomials, matrix functions and integral representations, Linear Algebra Appl. 434 (2011), no. 1, 174–184.

[171] B. Iannazzo, Numerical Solution of Certain Nonlinear Matrix Equations, Ph.D. thesis, Dipartimento di Matematica. Università di Pisa, Pisa, 2007.

[172] S. Ivanov, On the convergence of Chebyshev's method for multiple polynomial zeros, Result. Math. 69 (2016), no. 1, 93–103.

[173] N. A. A. Jamaludin, N. M. A. Nik Long, M. Salimi and S. Sharifi, Review of some iterative methods for solving nonlinear equations with multiple zeros, Afr. Mat. 30 (2019), no. 3, 355–369.

[174] P. Jarratt, Some fourth order multipoint iterative methods for solving equations, Math. Comput. 20 (1966), 434–437.

[175] P. Jarratt, Some efficient fourth order multipoint methods for solving equations, BIT Numer. Math. 9 (1969), no. 2, 119–124.

[176] L. O. Jay, A note on q-order of convergence, BIT Numer. Math. 41 (2001), 422–429.

[177] C. G. Jesudason, I. Numerical nonlinear analysis: Differential methods and optimization applied to chemical reaction rate determination, J. Math. Chem. 49 (2011), no. 7, 1384–1415.

[178] W. Jiang, C. L. Lin, V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović and T. E. Simos, Zeroing neural network approaches based on direct and indirect methods for solving the Yang–Baxter-like matrix equation, Mathematics 10 (2022), no. 11, Article ID 1950.

[179] H. Y. Jin and Z. A. Wang, Asymptotic dynamics of the one-dimensional attraction–repulsion Keller–Segel model, Math. Methods Appl. Sci. 38 (2015), no. 3, 444–457.

[180] G. Julia, Mémoire sur l'itération des fonctions rationnelles, J. Math. Pures Appl. 1 (1918), 47–245.

[181] D. Jung and C. Chun, A general approach for improving the Padé iterations for the matrix sign function, J. Comput. Appl. Math. 436 (2024), Article ID 115348.

[182] D. Jung, C. Chun and X. Wang, Construction of stable and globally convergent schemes for the matrix sign function, Linear Algebra Appl. 580 (2019), 14–36.

[183] M. U. D. Junjua, S. Abdullah, M. Kansal and S. Ahmad, On Traub–Steffensen-type iteration schemes with and without memory: Fractal analysis using basins of attraction, Fractal and Fract. 8 (2024), no. 12, Article ID 698.

[184] B. Kalantari, I. Kalantari and R. Zaare-Nahandi, A basic family of iteration functions for polynomial root finding and its characterizations, J. Comput. Appl. Math. 80 (1997), no. 2, 209–226.

[185] R. Kamra and S. C. S. Rao, A stable parallel algorithm for block tridiagonal toeplitz–block–toeplitz linear systems, Math. Comput. Simulation 190 (2021), 1415–1440.

[186] M. Kansal, A. S. Alshomrani, S. Bhalla, R. Behl and M. Salimi, One parameter optimal derivative-free family to find the multiple roots of algebraic nonlinear equations, Mathematics 8 (2020), no. 12, Article ID 2223.

[187] M. Kansal, R. Behl, M. A. A. Mahnashi and F. O. Mallawi, Modified optimal class of Newton-like fourth-order methods for multiple roots, Symmetry 11 (2019), no. 4, Article ID 526.

[188] M. Kansal, A. Cordero, S. Bhalla and J. R. Torregrosa, Memory in a new variant of King's family for solving nonlinear systems, Mathematics 8 (2020), no. 8, Article ID 1251.

[189] M. Kansal, A. Cordero, S. Bhalla and J. R. Torregrosa, New fourth-and sixth-order classes of iterative methods for solving systems of nonlinear equations and their stability analysis, Numer. Algorithms 87 (2021), 1017–1060.

[190] M. Kansal, A. Cordero, J. R. Torregrosa and S. Bhalla, A stable class of modified Newton-like methods for multiple roots and their dynamics, Int. J. Nonlinear Sci. Numer. Simul. 21 (2020), no. 6, 603–621.

[191] M. Kansal, V. Kanwar and S. Bhatia, Optimized mean based second derivative-free families of Chebyshev–Halley type methods, Numer. Anal. Appl. 9 (2016), no. 2, 129–140.

[192] M. Kansal, V. Kanwar and S. Bhatia, On some optimal multiple root-finding methods and their dynamics, Appl. Appl. Math. (AAM) 10 (2015), no. 1, Article ID 22.

[193] M. Kansal, V. Kanwar and S. Bhatia, New modifications of Hansen–Patrick's family with optimal fourth and eighth orders of convergence, Appl. Math. Comput. 269 (2015), 507–519.

[194] M. Kansal, V. N. Katsikis and P. Sharma, An iterative algorithm for approximating the sign function of tensors with the Einstein product, Quaest. Math. In press (2026), 1–24.

[195] M. Kansal, V. Sharma, L. Rani and L. Jäntschi, A novel class of fourth-order derivative-free iterative methods to obtain multiple zeros and their basins of attraction, AIMS Math. 9 (2024), no. 12, Article ID 35823.

[196] M. Kansal, V. Sharma and P. Sharma, Computation of invariant subspaces associated with certain eigenvalues using an approach based on matrix sign function, J. Comput. Appl. Math. 472 (2026), Article ID 116800.

[197] M. Kansal, V. Sharma, P. Sharma and L. Jäntschi, A globally convergent iterative method for matrix sign function and its application for determining the eigenvalues of a matrix pencil, Symmetry 16 (2024), no. 4, Article ID 481.

[198] L. V. Kantorovich, On Newton's method for functional equations, In: Dokl. Akad. Nauk SSSR, Vol. 59, 1948, 1237–1240.

[199] L. V. Kantorovich, The majorant principle and Newton's method, In: Dokl. Akad. Nauk SSSR, Vol. 76, 1951, 17–20.

[200] L. V. Kantorovitch, The method of successive approximation for functional equations, Acta Math. 71 (1939), no. 1, 63–97.

[201] V. Kanwar, K. K. Sharma and R. Behl, A new family of Schröder's method and its variants based on power means for multiple roots of nonlinear equations, Int. J. Math. Educ. Sci. Technol. 41 (2010), no. 4, 558–565.

[202] V. Kanwar, S. Bhatia and M. Kansal, New optimal class of higher-order methods for multiple roots, permitting $f'(x_n) = 0$, Appl. Math. Comput. 222 (2013), 564–574.

[203] V. Kanwar, A. Cordero, J. R. Torregrosa, M. Rajput and R. Behl, A new third-order family of multiple root-findings based on exponential fitted curve, Algorithms 16 (2023), no. 3, Article ID 156.

[204] V. Kanwar, S. Singh, R. K. Guha and others, On method of osculating circle for solving nonlinear equations, Appl. Math. Comput. 176 (2006), no. 1, 379–382.

[205] V. N. Katsikis, P. S. Stanimirović, S. D. Mourtas, L. Xiao, D. Stanujkić and D. Karabašević, Zeroing neural network based on neutrosophic logic for calculating minimal-norm least-squares solutions to time-varying linear systems, Neural Process. Lett. 55 (2023), no. 7, 8731–8753.

[206] M. Kaur, S. Kumar and M. Kansal, New derivative-free iterative family having optimal convergence order sixteen and its applications, Eng. Comput. 39 (2022), no. 3, 965–992.

[207] C. T. Kelley, Solving Nonlinear Equations with Newton's Method, SIAM, Philadelphia, 2003.

[208] T. A. Kendi and F. J. Doyle, Nonlinear control of a fluidized bed reactor using approximate feedback linearization, Ind. Eng. Chem. Res. 35 (1996), no. 3, 746–757.

[209] C. S. Kenney and A. J. Laub, The matrix sign function, IEEE Trans. Autom. Control 40 (1995), no. 8, 1330–1348.

[210] C. Kenney and A. J. Laub, Rational iterative methods for the matrix sign function, SIAM J. Matrix Anal. Appl. 12 (1991), no. 2, 273–291.

[211] C. Kenney and A. J. Laub, On scaling Newton's method for polar decomposition and the matrix sign function, SIAM J. Matrix Anal. Appl. 13 (1992), no. 3, 688–706.

[212] R. F. King, A family of fourth order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973), no. 5, 876–879.

[213] J. Kou, Y. Li and X. Wang, A family of fifth-order iterations composed of Newton and third-order methods, Appl. Math. Comput. 186 (2007), no. 2, 1258–1262.

[214] J. Kou, Y. Li and X. Wang, A composite fourth-order iterative method for solving non-linear equations, Appl. Math. Comput. 184 (2007), no. 2, 471–475.

[215] A. Kumar and J. R. Cardoso, Iterative methods for finding commuting solutions of the Yang–Baxter-like matrix equation, Appl. Math. Comput. 333 (2018), 246–253.

[216] J. Kumar, M. Peglow, G. Warnecke, S. Heinrich and L. Mörl, Improved accuracy and convergence of discretized population balance for aggregation: The cell average technique, Chem. Eng. Sci. 61 (2006), no. 10, 3327–3342.

[217] S. Kumar and R. Ali, A robust family of optimal fourth-order iteration schemes for multiple roots with applications, J. Comput. Appl. Math. 471 (2026), Article ID 116758.

[218] S. Kumar, R. Ishwariya, M. U. D. Junjua and S. Akram, A novel optimal fourth-order iteration scheme for solving nonlinear problems in applied sciences, J. Appl. Math. Comput. 71 (2025), no. 1, 643–667.

[219] S. Kumar, J. R. Sharma and L. Jäntschi, Optimal fourth-order methods for multiple zeros: Design, convergence analysis and applications, Axioms 13 (2024), no. 3, Article ID 143.

[220] S. Kumar, J. R. Sharma and L. Jäntschi, An optimal family of eighth-order methods for multiple-roots and their complex dynamics, Symmetry 16 (2024), no. 8, Article ID 1045.

[221] H. T. Kung and J. F. Traub, Optimal order of one-point and multipoint iteration, J. ACM. 21 (1974), no. 4, 643–651.

[222] I. Kyrchei, D. Mosić and P. Stanimirović, The right–left WG inverse solutions to quaternion matrix equations, Symmetry 17 (2024), no. 1, 38.

[223] J. L. Lagouanelle, Sur une mtode de calcul de l'ordre de multiplicit des zros d'un polynme, C. R. Math. Acad. Sci. Paris A 262 (1966), no. 11, 626–627.

[224] J. D. Lambert, Numerical Methods for Ordinary Differential Systems, Vol. 146, Wiley, New York, 1991.

[225] P. Lancaster and L. Rodman, Algebraic Riccati Equations, Oxford University Press, New York, 1995.

[226] A. J. Laub, Invariant subspace methods for the numerical solution of Riccati equations, The Riccati Equation, Communications and Control Engineering Series, Springer-Verlag, Berlin, Heidelberg, 1991.

[227] J. D. Lawson and Y. Lim, The geometric mean, matrices, metrics, and more, Am. Math. Mon. 108 (2001), no. 9, 797–812.

[228] F. Le Gall, Powers of tensors and fast matrix multiplication, In: Proceedings of the 39th international symposium on symbolic and algebraic computation, 2014, 296–303.

[229] Y. Levin and A. Ben-Israel, An inverse-free directional Newton method for solving systems of nonlinear equations, In: Progress in Analysis: (In 2 Volumes), World Scientific, 2003, 1447–1457.

[230] S. G. Li, L. Z. Cheng and B. Neta, Some fourth-order nonlinear solvers with closed formulae for multiple roots, Comput. Math. Appl. 59 (2010), no. 1, 126–135.

[231] S. Li, H. Li and L. Cheng, Some second-derivative-free variants of Halley's method for multiple roots, Appl. Math. Comput. 215 (2009), no. 6, 2192–2198.

[232] M. M. Lin and C. Y. Chiang, An iterative method for solving the stable subspace of a matrix pencil and its application, Linear Multilinear A. 66 (2018), no. 7, 1279–1298.

[233] L. Linzhang and C. E. M. Pearce, On the matrix-sign-function method for solving algebraic Riccati equations, Appl. Math. Comput. 86 (1997), no. 2-3, 157–170.

[234] P. Liu, J. Shi and Z. A. Wang, Pattern formation of the attraction-repulsion Keller-Segel system, Discrete Contin. Dyn. Syst. Ser. B. 18 (2013), no. 10, 2597–2625.

[235] T. Liu, M. Z. Ullah, K. M. A. Alshahrani and S. Shateyi, From fractal behavior of iteration methods to an efficient solver for the sign of a matrix, Fractal and Fract. 7 (2023), no. 1, Article ID 32.

[236] P. G. Lograd o and J. D. M. Vianna, Partitioning technique procedure revisited: Formalism and first application to atomic problems, J. Math. Chem. 22 (1997), no. 2-4, 107–116.

[237] K. Madhu, D. K. R. Babajee and J. Jayaraman, An improvement to double-step Newton method and its multi-step version for solving system of nonlinear equations and its applications, Numer. Algorithms 74 (2017), no. 2, 593–607.

[238] M. Mahalakshmi, G. Hariharan and K. Kannan, The wavelet methods to linear and nonlinear reaction–diffusion model arising in mathematical chemistry, J. Math. Chem. 51 (2013), 2361–2385.

[239] S. Maji and S. Natesan, Analytical and numerical solutions of time-fractional advection-diffusion-reaction equation, Appl. Numer. Math. 185 (2023), 549–570.

[240] J. M. McNamee and V. Y. Pan, Methods involving second or higher derivatives, In: Studies in Computational Mathematics, Vol. 16, Elsevier, 2013, 215–379.

[241] J. M. McNamee, Numerical Methods for Roots of Polynomials, Part I, Elsevier, Amsterdam, 2007.

[242] J. M. McNamee and V. Y. Pan, Efficient polynomial root-refiners: A survey and new record efficiency estimates, Comput. Math. Appl. 63 (2012), no. 1, 239–254.

[243] J. M. McNamee, A comparison of methods for accelerating convergence of Newton's method for multiple polynomial roots, ACM SIGNUM Newsletter 33 (1998), no. 2, 17–22.

[244] J. M. McNamee, A bibliography on roots of polynomials, J. Comput. Appl. Math. 47 (1993), no. 3, 391–394.

[245] N. A. Mir and T. Zaman, Some quadrature based three-step iterative methods for non-linear equations, Appl. Math. Comput. 193 (2007), no. 2, 366–373.

[246] M. S. Misrikhanov and V. N. Ryabchenko, Matrix sign function in the problems of analysis and design of the linear systems, ARC 69 (2008), no. 2, 198–222.

[247] M. Moakher, A differential geometric approach to the geometric mean of symmetric positive-definite matrices, SIAM J. Matrix Anal. Appl. 26 (2005), no. 3, 735–747.

[248] C. B. Moler and G. W. Stewart, An algorithm for generalized matrix eigenvalue problems, SIAM J. Numer. Anal. 10 (1973), no. 2, 241–256.

[249] M. Momenzadeh and T. Lotfi, New iterative methods for finding matrix sign function: Derivation and application, Comp. Appl. Math. 38 (2019), no. 2, Article ID 53.

[250] H. Montazeri, F. Soleymani, S. Shateyi and S. S. Motsa, On a new method for computing the numerical solution of systems of nonlinear equations, J. Appl. Math. 2012 (2012), no. 1, Article ID 751975.

[251] C. D. Murray and S. F. Dermott, Solar System Dynamics, Cambridge University Press, New York, USA, 1999.

[252] M. Narang, S. Bhatia and V. Kanwar, New efficient derivative free family of seventh-order methods for solving systems of nonlinear equations, Numer. Algorithms 76 (2017), 283–307.

[253] M. Narang, S. Bhatia and V. Kanwar, New two-parameter Chebyshev–Halley-like family of fourth and sixth-order methods for systems of nonlinear equations, Appl. Math. Comput. 275 (2016), 394–403.

[254] S. Natesan and N. Ramanujam, A computational method for solving singularly perturbed turning point problems exhibiting twin boundary layers, Appl. Math. Comput. 93 (1998), no. 2-3, 259–275.

[255] S. Natesan and M. K. Singh, Robust computational method for singularly perturbed system of parabolic convection-diffusion problems with interior layers, Comput. Math. Meth. 3 (2021), no. 6, Article ID 1146.

[256] A. H. Nayfeh, Perturbation Methods, John Wiley & Sons, New York, 1973.

[257] G. H. Nedzhibov, A family of multi-point iterative methods for solving systems of nonlinear equations, J. Comput. Appl. Math. 222 (2008), no. 2, 244–250.

[258] B. Neta, A sixth-order family of methods for nonlinear equations, Int. J. Comput. Methods 7 (1979), no. 1997, 157–161.

[259] B. Neta, New third order nonlinear solvers for multiple roots, Appl. Math. Comput. 202 (2008), no. 1, 162–170.

[260] B. Neta and A. N. Johnson, High-order nonlinear solver for multiple roots, Comput. Math. Appl. 55 (2008), no. 9, 2012–2017.

[261] A. N. Norris, A. L. Shuvalov and A. A. Kutsenko, The matrix sign function for solving surface wave problems in homogeneous and laterally periodic elastic half-spaces, Wave Motion 50 (2013), no. 8, 1239–1250.

[262] N. Obreshkov, On the numerical solution of equations, Annuaire Univ. Sofia, Fac. Phys. Math. 56 (1963), 73–83.

[263] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[264] N. Osada, An optimal multiple root-finding method of order three, J. Comput. Appl. Math. 51 (1994), no. 1, 131–133.

[265] A. M. Ostrowski, Solutions of Equations and System of Equations, Academic Press, New York, 1960.

[266] P. Pandey, C. Kenney and A. J. Laub, A parallel algorithm for the matrix sign function, Int. J. High Perform. Comput. Appl. 2 (1990), no. 02, 181–191.

[267] A. K. Patra, A. K. Mishra and P. K. Rout, Backstepping model predictive controller for blood glucose regulation in type-I diabetes patient, IETE J. Res. 66 (2020), no. 3, 326–340.

[268] M. S. Petkovic, B. Neta, L. D. Petkovic and J. Dzunic, Multipoint Methods for Solving Nonlinear Equations, Academic Press, Cambridge, MA, USA, 2012.

[269] L. D. Petković, M. S. Petković and B. Neta, On optimal parameter of Laguerre's family of zero-finding methods, Int. J. Comput. Math. 96 (2019), no. 4, 692–707.

[270] M. S. Petković, B. Neta, L. D. Petković and J. Džunić, Multipoint methods for solving nonlinear equations: A survey, Appl. Math. Comput. 226 (2014), 635–660.

[271] M. S. Petković and L. D. Petković, Construction and efficiency of multipoint root-ratio methods for finding multiple zeros, J. Comput. Appl. Math. 351 (2019), 54–65.

[272] M. S. Petković and L. D. Petković, Dynamic study of Schröder's families of first and second kind, Numer. Algorithms 78 (2018), no. 3, 847–865.

[273] M. S. Petković, L. D. Petković and B. Neta, On generalized halley-like methods for solving nonlinear equations, Appl. Anal. Discrete Math. 13 (2019), no. 2, 399–422.

[274] F. A. Potra and V. Pták, Nondiscrete Induction and Iterative Processes, Pitman Publishing, Boston, 1984.

[275] P. D. Proinov and S. I. Ivanov, Convergence of Schröder's method for polynomial zeros of unknown multiplicity, C. R. Acad. Bulg. Sci. 66 (2013), no. 8, 1073–1080.

[276] S. Qureshi, A. Soomro, A. A. Shaikh, E. Hincal and N. Gokbulut, A novel multistep iterative technique for models in medical sciences with complex dynamics, Comput. Math. Methods Med. 2022 (2022), no. 1, Article ID 7656451.

[277] L. B. Rall, Convergence of the Newton process to multiple solutions, Numer. Math. 9 (1966), no. 1, 23–37.

[278] H. Ramos, S. Mehta and J. Vigo-Aguiar, A unified approach for the development of k-step block Falkner-type methods for solving general second-order initial-value problems in ODEs, J. Comput. Appl. Math. 318 (2017), 550–564.

[279] S. C. S. Rao and A. K. Chaturvedi, Parameter-uniform numerical method for a two-dimensional singularly perturbed convection–reaction–diffusion problem with interior and boundary layers, Math. Comput. Simulation 187 (2021), 656–686.

[280] S. C. S. Rao and S. Chawla, Parameter-uniform convergence of a numerical method for a coupled system of singularly perturbed semilinear reaction–diffusion equations with boundary and interior layers, J. Comput. Appl. Math. 352 (2019), 223–239.

[281] S. C. S. Rao and R. Kamra, A hybrid parallel algorithm for large sparse linear systems, Numer. Linear Algebra Appl. 25 (2018), no. 6, Article ID 2210.

[282] S. C. S. Rao, S. Kumar and J. Singh, A discrete schwarz waveform relaxation method of higher order for singularly perturbed parabolic reaction–diffusion problems, J. Math. Chem. 58 (2020), no. 3, 574–594.

[283] J. D. Roberts, Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, Int. J. Control 32 (1980), no. 4, 677–687.

[284] L. P. Russo and B. W. Bequette, State-space versus input/output representations for cascade control of unstable systems, Ind. Eng. Chem. Res. 36 (1997), no. 6, 2271–2278.

[285] B. Saheya, G. Q. Chen, Y. K. Sui and C. Y. Wu, A new Newton-like method for solving nonlinear equations, SpringerPlus 5 (2016), no. 1, Article ID 1269.

[286] S. Salehi and T. Lotfi, A globally convergent numerical method with fourth order for computing the matrix sign, Math. Methods Appl. Sci. 48 (2025), no. 7, 7460–7468.

[287] V. Samanskii, On a modification of the Newton method, Ukr. Math. J. 19 (1967), 133–138.

[288] B. Satpati, C. Koley and S. Datta, Sensor-less predictive drying control of pneumatic conveying batch dryers, IEEE Access 5 (2017), 3547–3568.

[289] T. Sauer, Numerical Analysis, Pearson, Hoboken, USA, 2012.

[290] T. R. Scavo and J. B. Thoo, On the geometry of Halley's method, Am. Math. Mon. 102 (1995), no. 5, 417–426.

[291] W. E. Schiesser, A. J. Mchugh and G. W. Griffiths, Introductory Global CO_2 Model, An (With Companion Media Pack), World Scientific Publishing Company, 2015.

[292] E. Schröder, Über unendlich viele algorithmen zur auflösung der gleichungen, Math. Ann. 2 (1870), no. 2, 317–365.

[293] G. Schulz, Iterative berechung der reziproken matrix, Z. Angew. Math. Mech. 13 (1933), no. 1, 57–59.

[294] A. Seal and S. Natesan, An accurate finite-difference scheme for the numerical solution of a fractional differential equation, Comput. Methods Differ. Equ. 14 (2026), no. 1, 165–187.

[295] N. Shang, X. Wang and Y. Wang, Local convergence analysis of a novel Kurchatov-type derivative-free methods with and without memory for solving nonlinear systems, Int. J. Comput. Methods 23 (2026), no. 1, Article ID 2550033.

[296] M. Sharifi, S. Karimi Vanani, F. Khaksar Haghani, M. Arab and S. Shateyi, On a cubically convergent iterative method for matrix sign, Sci. World J. 2015 (2015), no. 1, Article ID 964257.

[297] M. Sharifi, D. K. R. Babajee and F. Soleymani, Finding the solution of nonlinear equations by a class of optimal methods, Comput. Math. Appl. 63 (2012), no. 4, 764–774.

[298] H. Sharma, R. Behl, M. Kansal and H. Ramos, A robust iterative family for multiple roots of nonlinear equations: Enhancing accuracy and handling critical points, J. Comput. Appl. Math. 444 (2024), Article ID 115795.

[299] H. Sharma and M. Kansal, Stability analysis and dynamical behavior of optimal mean-based iterative methods, J. Math. Chem. 63 (2025), no. 2, 383–405.

[300] J. R. Sharma, A family of third-order methods to solve nonlinear equations by quadratic curves approximation, Appl. Math. Comput. 184 (2007), no. 2, 210–215.

[301] J. R. Sharma and H. Arora, Efficient derivative-free numerical methods for solving systems of nonlinear equations, Comp. Appl. Math. 35 (2016), 269–284.

[302] J. R. Sharma and H. Arora, A novel derivative free algorithm with seventh order convergence for solving systems of nonlinear equations, Numer. Algorithms 67 (2014), 917–933.

[303] J. R. Sharma and S. Kumar, A class of accurate Newton–Jarratt-like methods with applications to nonlinear models, Comp. Appl. Math. 41 (2022), no. 1, Article ID 46.

[304] J. R. Sharma and H. Arora, On efficient weighted-Newton methods for solving systems of nonlinear equations, Appl. Math. Comput. 227 (2013), 497–506.

[305] J. R. Sharma and H. Arora, Efficient Jarratt-like methods for solving systems of nonlinear equations, Calcolo. 51 (2014), no. 1, 193–210.

[306] J. R. Sharma and H. Arora, An efficient derivative free iterative method for solving systems of nonlinear equations, Appl. Anal. Discrete Math. (2013), 390–403.

[307] J. R. Sharma, R. K. Guha and R. Sharma, An efficient fourth order weighted-Newton method for systems of nonlinear equations, Numer. Algorithms 62 (2013), no. 2, 307–323.

[308] J. R. Sharma and S. Kumar, An excellent numerical technique for multiple roots, Math. Comput. Simul. 182 (2021), 316–324.

[309] J. R. Sharma, S. Kumar and L. Jäntschi, On a class of optimal fourth order multiple root solvers without using derivatives, Symmetry 11 (2019), no. 12, Article ID 1452.

[310] J. R. Sharma, S. Kumar and L. Jäntschi, On derivative free multiple-root finders with optimal fourth order convergence, Mathematics 8 (2020), no. 7, Article ID 1091.

[311] J. R. Sharma and R. Sharma, Modified Jarratt method for computing multiple roots, Appl. Math. Comput. 217 (2010), no. 2, 878–881.

[312] J. R. Sharma, R. Sharma and A. Bahl, An improved Newton–Traub composition for solving systems of nonlinear equations, Appl. Math. Comput. 290 (2016), 98–110.

[313] P. Sharma and M. Kansal, An iterative formulation to compute the matrix sign function and its application in evaluating the geometric mean of Hermitian positive definite matrices, Med. J. Math. 22 (2025), no. 2, Article ID 39.

[314] L. Shengguo, L. Xiangke and C. Lizhi, A new fourth-order iterative method for finding multiple roots of nonlinear equations, Appl. Math. Comput. 215 (2009), no. 3, 1288–1292.

[315] L. S. Shieh, Y. T. Tsay and C. T. Wang, Matrix sector functions and their applications to systems theory, In: IEE Proc. Control Theory Appl., Vol. 131, IET, 1984, 171–181.

[316] T. E. Simos and J. V. Aguiar, Symmetric eighth algebraic order methods with minimal phase-lag for the numerical solution of the Schrödinger equation, J. Math. Chem. 31 (2002), no. 2, 135–144.

[317] T. E. Simos and J. V. Aguiar, A symmetric high order method with minimal phase-lag for the numerical solution of the Schrödinger equation, Int. J. Mod. Phys. C. 12 (2001), no. 07, 1035–1042.

[318] T. E. Simos and J. V. Aguiar, A new modified Runge–Kutta–Nyström method with phase-lag of order infinity for the numerical solution of the Schrödinger equation and related problems, Int. J. Mod. Phys. C. 11 (2000), no. 06, 1195–1208.

[319] T. E. Simos, V. N. Katsikis, S. D. Mourtas and P. S. Stanimirović, Unique non-negative definite solution of the time-varying algebraic Riccati equations with applications to stabilization of LTV systems, Math. Comput. Simulation 202 (2022), 164–180.

[320] T. E. Simos, V. N. Katsikis, S. D. Mourtas and P. S. Stanimirović, Finite-time convergent zeroing neural network for solving time-varying algebraic Riccati equations, Journal of the Franklin Institute 359 (2022), no. 18, 10867–10883.

[321] H. Singh and J. R. Sharma, Simple yet highly efficient numerical techniques for systems of nonlinear equations, Comp. Appl. Math. 42 (2023), no. 1, Article ID 22.

[322] H. Singh and J. R. Sharma, Simple and efficient fifth order solvers for systems of nonlinear problems, Math. Model. Anal. 28 (2023), no. 1, 1–22.

[323] H. Singh and J. R. Sharma, A two-point Newton-like method of optimal fourth order convergence for systems of nonlinear equations, J. Complex. 86 (2025), Article ID 101907.

[324] H. Singh, J. R. Sharma and S. Kumar, A simple yet efficient two-step fifth-order weighted-Newton method for nonlinear models, Numer. Algorithms 93 (2023), no. 1, 203–225.

[325] R. Singh, N. Das and J. Kumar, The optimal modified variational iteration method for the Lane-Emden equations with Neumann and Robin boundary conditions, Eur. Phys. J. Plus 132 (2017), no. 6, Article ID 251.

[326] R. Singh and J. Kumar, An efficient numerical technique for the solution of nonlinear singular boundary value problems, Comput. Phys. Commun. 185 (2014), no. 4, 1282–1289.

[327] A. R. Soheili, F. Toutounian and F. Soleymani, A fast convergent numerical method for matrix sign function with application in SDEs, J. Comput. Appl. Math. 282 (2015), 167–178.

[328] F. Soleymani and D. K. R. Babajee, Computing multiple zeros using a class of quartically convergent methods, Alex. Eng. J. 52 (2013), no. 3, 531–541.

[329] F. Soleymani, D. K. R. Babajee and T. Lotfi, On a numerical technique for finding multiple zeros and its dynamic, J. Egypt. Math. Soc. 21 (2013), no. 3, 346–353.

[330] F. Soleymani, M. Sharifi, S. Shateyi and F. Khaksar Haghani, An algorithm for computing geometric mean of two Hermitian positive definite matrices via matrix sign, In: Abstr. Appl. Anal., Vol. 2014, Hindawi, 2014, Article ID 978629.

[331] F. Soleymani, P. S. Stanimirović, S. Shateyi and F. K. Haghani, Approximating the matrix sign function using a novel iterative method, In: Abstr. Appl. Anal., Vol. 2014, Wiley Online Library, 2014, 9, Article ID 105301.

[332] F. Soleymani, P. S. Stanimirović and I. Stojanović, A novel iterative method for polar decomposition and matrix sign function, Discrete Dyn. Nat. Soc. 2015 (2015), Article ID 649423.

[333] F. Soleymani, M. Barfeie and F. K. Haghani, Inverse multi-quadric RBF for computing the weights of FD method: Application to american options, Commun. Nonlinear Sci. Numer. Simul. 64 (2018), 74–88.

[334] F. Soleymani, S. Karimi Vanani and M. Jamali Paghaleh, A class of three-step derivative-free root solvers with optimal convergence order, J. Appl. Math. 2012 (2012), no. 1, Article ID 568740.

[335] F. Soleymani, F. Khaksar Haghani and S. Shateyi, Several numerical methods for computing unitary polar factor of a matrix, Adv. Differ. Equ. 2016 (2016), 1–11.

[336] F. Soleymani, F. W. Khdhr, R. K. Saeed and J. Golzarpoor, A family of high order iterations for calculating the sign of a matrix, Math. Methods Appl. Sci. 43 (2020), no. 14, 8192–8203.

[337] F. Soleymani and A. Kumar, A fourth-order method for computing the sign function of a matrix with application in the Yang–Baxter-like matrix equation, Comp. Appl. Math. 38 (2019), 1–12.

[338] F. Soleymani, T. Lotfi and P. Bakhtiari, A multi-step class of iterative methods for nonlinear systems, Optim. Lett. 8 (2014), no. 3, 1001–1015.

[339] F. Soleymani, E. Tohidi, S. Shateyi and F. K. Haghani, Some matrix iterations for computing matrix sign function, J. Appl. Math. 2014 (2014), no. 1, Article ID 425654.

[340] F. Soleymani and M. Zaka Ullah, A multiquadric RBF–FD scheme for simulating the financial HHW equation utilizing exponential integrator, Calcolo. 55 (2018), no. 4, Article ID 51.

[341] Y. Song, F. Soleymani and A. Kumar, Finding the geometric mean for two Hermitian matrices by an efficient iteration method, Math. Methods Appl. Sci. 48 (2025), no. 1–2, 7188–7196.

[342] P. S. Stanimirović, V. N. Katsikis, S. Srivastava and D. Pappas, A class of quadratically convergent iterative methods, Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. 113 (2019), no. 4, 3125–3146.

[343] P. S. Stanimirović, A. Kumar and V. N. Katsikis, Further efficient hyperpower iterative methods for the computation of generalized inverses $A^{(2)}_{T,S}$, RACSAM 113 (2019), no. 4, 3323–3339.

[344] P. S. Stanimirović, M. Ćirić, S. D. Mourtas, G. V. Milovanović and M. J. Petrović, Simultaneous method for solving certain systems of matrix equations with two unknowns, Axioms 13 (2024), no. 12, Article ID 838.

[345] J. F. Steffensen, Further remarks on iteration, Scand. Actuar. J. 1945 (1945), no. 1-2, 44–55.

[346] J. F. Steffensen, Remarks on iteration, Scand. Actuar. J. 1933 (1933), no. 1, 64–72.

[347] L. Sun, J. Hou, C. Xing and Z. Fang, A robust Hammerstein-Wiener model identification method for highly nonlinear systems., Processes 10 (2022), no. 12, Article ID 2664.

[348] X. Sun and E. S. Quintana-Ortí, The generalized Newton iteration for the matrix sign function, SIAM J. Sci. Comput. 24 (2002), no. 2, 669–683.

[349] R. Thukral, A new family of fourth-order iterative methods for solving nonlinear equations with multiple roots, J. Numer. Math. Stoch. 6 (2014), no. 1, 37–44.

[350] R. Thukral, Introduction to higher-order iterative methods for finding multiple roots of nonlinear equations, J. Math. 2013 (2013), no. 1, Article ID 404635.

[351] J. F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall:Englewood Cliffs, New Jersey, 1964.

[352] J. L. Varona, Graphic and numerical comparison between iterative methods, Math. Intell. 24 (2002), no. 1, 37–47.

[353] H. D. Victory Jr and B. Neta, A higher order method for multiple zeros of nonlinear functions, Int. J. Comput. Methods 12 (1983), no. 3-4, 329–335.

[354] J. Vigo-Aguiar and S. Natesan, An efficient numerical method for singular perturbation problems, J. Comput. Appl. Math. 192 (2006), no. 1, 132–141.

[355] J. Vigo-Aguiar and T. E. Simos, Family of twelve steps exponential fitting symmetric multistep methods for the numerical solution of the Schrödinger equation, J. Math. Chem. 32 (2002), no. 3, 257–270.

[356] J. Vigo-Aguiar and J. M. Ferrandiz, A general procedure for the adaptation of multistep algorithms to the integration of oscillatory problems, SIAM J. Numer. Anal. 35 (1998), no. 4, 1684–1708.

[357] J. Vigo-Aguiar and T. E. Simos, Review of multistep methods for the numerical solution of the radial Schrödinger equation, Int. J. Quantum Chem. 103 (2005), no. 3, 278–290.

[358] E. R. Vrscay and W. J. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and König rational iteration functions, Numer. Math. 52 (1987), 1–16.

[359] S. Wang, Z. Wang, W. Xie, Y. Qi and T. Liu, An accelerated sixth-order procedure to determine the matrix sign function computationally, Mathematics 13 (2025), no. 7, Article ID 1080.

[360] X. Wang and T. Zhang, A family of Steffensen type methods with seventh-order convergence, Numer. Algorithms 62 (2013), 429–444.

[361] X. Wang, T. Zhang, W. Qian and M. Teng, Seventh-order derivative-free iterative method for solving nonlinear systems, Numer. Algorithms 70 (2015), 545–558.

[362] X. Wang and L. Liu, Modified Ostrowski's method with eighth-order convergence and high efficiency index, Appl. Math. Lett. 23 (2010), no. 5, 549–554.

[363] S. Weerakoon and T. G. I. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett. 13 (2000), no. 8, 87–93.

[364] P. R. Wellin, R. J. Gaylord and S. N. Kamin, An Introduction to Programming with Mathematica®, Cambridge University Press, 2005.

[365] S. Wolfram, The Mathematica Book, Wolfram Media, Champaign, USA, 2003.

[366] F. Wu, L. Zhu, Y. Zhao and K. Zhang, A filtering technique for the matrix power series being near-sparse, ArXiv preprint (2022), https://api.semanticscholar.org/CorpusID:251493209.

[367] X. Y. Wu, A new continuation Newton-like method and its deformation, Appl. Math. Comput. 112 (2000), no. 1, 75–78.

[368] C. Xiaoshan, Solving the (generalized) periodic Sylvester equation with the matrix sign function, Mathematica Numerica Sinica 34 (2012), no. 2, 153–162.

[369] C. N. Yang, Some exact results for the many-body problem in one dimension with repulsive delta-function interaction, Phys. Rev. Lett. 19 (1967), no. 23, Article ID 1312.

[370] C. N. Yang and M. L. Ge, Braid Group, Knot Theory, and Statistical Mechanics II, World Scientific, Singapore, 1994.

[371] S. Yaseen and F. Zafar, A new sixth-order Jarratt-type iterative method for systems of nonlinear equations, Arab. J. Math. 11 (2022), no. 3, 585–599.

[372] T. J. Ypma, Finding a multiple zero by transformations and Newton-like methods, SIAM Rev. 25 (1983), no. 3, 365–378.

[373] J. Yu and X. Wang, A single parameter fourth-order Jarratt type iterative method for solving nonlinear systems, AIMS Math. 10 (2025), 7847–7863.

[374] B. I. Yun, A derivative free iterative method for finding multiple roots of nonlinear equations, Appl. Math. Lett. 22 (2009), no. 12, 1859–1863.

[375] F. Zafar, A. Cordero, R. Quratulain and J. R. Torregrosa, Optimal iterative methods for finding multiple roots of nonlinear equations using free parameters, J. Math. Chem. 56 (2018), no. 7, 1884–1901.

[376] F. Zafar, A. Cordero and J. R. Torregrosa, Stability analysis of a family of optimal fourth-order methods for multiple roots, Numer. Algorithms 81 (2019), no. 3, 947–981.

[377] N. Zainali and T. Lotfi, A globally convergent variant of mid-point method for finding the matrix sign, Comp. Appl. Math. 37 (2018), 5795–5806.

[378] M. Zaka Ullah and S. M. Alaslani, Exploiting a higher-order scheme for matrix square root and its inverse simultaneously, Math. Methods Appl. Sci. 48 (2025), 7078–7088.

[379] Z. Zeng, Computing multiple roots of inexact polynomials, Math. Comput. 74 (2005), no. 250, 869–903.

[380] T. Zhanlav, C. Chun, K. Otgondorj and V. Ulziibayar, High-order iterations for systems of nonlinear equations, Int. J. Comput. Methods 97 (2020), no. 8, 1704–1724.

[381] T. Zhanlav, R. Mijiddorj and O. Khuder, A family of Newton-type methods with seventh and eighth-order of convergence for solving systems of nonlinear equations, Hacet. J. Math. Stat. 52 (2023), no. 4, 1006–1021.

[382] Q. Zheng, P. Zhao and F. Huang, A family of fourth-order Steffensen-type methods with the applications on solving nonlinear ODEs, Appl. Math. Comput. 217 (2011), no. 21, 8196–8203.

[383] X. Zhou, X. Chen and Y. Song, Constructing higher-order methods for obtaining the multiple roots of nonlinear equations, J. Comput. Appl. Math. 235 (2011), no. 14, 4199–4206.

[384] X. Zhou, X. Chen and Y. Song, Families of third and fourth order methods for multiple roots of nonlinear equations, Appl. Math. Comput. 219 (2013), no. 11, 6030–6038.

Turnitin Originality Report

Processed on: 19-May-2026 11:41 IST
ID: 2706737331
Word Count: 81442
Submitted: 17

thesisDOM By Munish Kansal

Munish Kansal 19-5-26

Similarity Index	Similarity by Source	
11%	Internet Sources:	7%
	Publications:	9%
	Student Papers:	2%

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

1% match ()
Ahmad, Fayyaz. "Equations and systems of nonlinear equations: from high order numerical methods to fast Eigensolvers for structured matrices and applications", Italy, 2018

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

1% match (Internet from 05-Mar-2026)
https://tudr.thapar.edu/server/api/core/bitstreams/27e104d5-9e20-46cf-a953-67ca91099ff4/content

1% match (Internet from 21-Feb-2020)
https://res.mdpi.com/bookfiles/book/1729/Symmetry_with_Operator_Theory_and_Equations.pdf?v=1582296128

1% match (Internet from 03-Feb-2020)
https://repository.nwu.ac.za/bitstream/handle/10394/17990/CMMSE-Proceedings-2016-1525.pdf?isAllowed=y&sequence=1

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 21-Nov-2022)
http://uilis.unsyiah.ac.id/oer/files/original/5448af69cbcd2ab945486aabfab7c945.pdf

< 1% match (Internet from 20-Feb-2020)
https://res.mdpi.com/bookfiles/book/1877/Iterative_Methods_for_Solving_Nonlinear_Equations_and_Systems.pdf?v=1582215761

< 1% match (Abdullah Khamis Hassan Alzahrani, Ramandeep Behl, Ali Saleh Alshomrani. "Some higher-order iteration functions for solving nonlinear models", Applied Mathematics and Computation, 2018)
Abdullah Khamis Hassan Alzahrani, Ramandeep Behl, Ali Saleh Alshomrani. "Some higher-order iteration functions for solving nonlinear models", Applied Mathematics and Computation, 2018

< 1% match (publications)
Leslie Hogben, Richard Brualdi, Anne Greenbaum, Roy Mathias. "Handbook of Linear Algebra", CRC Press, 2006

< 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 (Internet from 25-Nov-2022)
https://downloads.hindawi.com/journals/specialissues/127152.pdf

< 1% match (publications)
Ioannis K. Argyros, Yeol J. Cho, Saïd Hilout. "Numerical Methods for Equations and its Applications", CRC Press, 2019

< 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 (Janak Raj Sharma, Deepak Kumar. "ON A CLASS OF EFFICIENT HIGHER ORDER NEWTON-LIKE METHODS", Mathematical Modelling and Analysis, 2019)
Janak Raj Sharma, Deepak Kumar. "ON A CLASS OF EFFICIENT HIGHER ORDER NEWTON-LIKE METHODS", Mathematical Modelling and Analysis, 2019

< 1% 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

< 1% match (publications)
Victor S. Ryaben'kii, Semyon V. Tsynkov. "A Theoretical Introduction to Numerical Analysis", CRC Press, 2006

< 1% match (Marlon Ernesto Moscoso Martínez. "Efficient Numerical Methods for Solving Nonlinear Problems", Universitat Politecnica de Valencia, 2024)