Study of some higher-order methods for multiple roots of nonlinear equations

Thesis Submitted in partial fulfillment of the requirements for

The award of the degree of

Masters of Science

In

Mathematics and Computing

Submitted by

Pavittarjit Kaur

Roll no.- 301203011

Under

the guidance of

Dr. Ramandeep Behl

July, 2014.

School of Mathematics and Computer Applications

Thapar University

Patiala- 147004(PUNJAB)

INDIA

To the love of

GOD

and

MY PARENTS

CERTIFICATE

I hereby certify that the work which is bring presented in the thesis entitled " **Study of some higher-order methods for multiple roots of nonlinear equations**" *in partial fulfillment of the requirements for the award of degree of Master of Science, School of Mathematics and Computer Applications, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of Dr. Ramandeep Behl.*

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Pavittarjit Kaur

(Pavittarjit Kaur)
Reg. No. 301203011

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Raman Behl

(Dr. Ramandeep Behl)
Lecturer (Adhoc)
SMCA, Thapar University
Patiala.

Countersigned by:

Dr. Rajesh Kumar

(Associate Professor & Head)
School of Mathematics & Computer Applications
Thapar University, Patiala.

Dr. S.K. Mohapatrara

Dean of Academic Affairs
Thapar University
Patiala.

ACKNOWLEDGMENT

I would like to express my greatest gratitude to the people who have helped and supported me throughout my thesis.

I am grateful to my supervisor, **Dr. Ramandeep Behl**, Lecturer, School of Mathematics and Computer Application, Thapar University, Patiala, for continuous support, advices, guidance for the thesis work. Moreover, for mentioning me when I needed it the most. I am fortunate that I got an opportunity to work under his supervision.

I would like to express my sincere thanks to **Dr. Rajesh Kumar**, Head of Department, School of Mathematics and Computer Application, Thapar University, Patiala, for providing the necessary facilities needed for this work and directly or indirectly encouraged me to do the work.

A special thanks to my family. Words cannot express how grateful I am to my mother, father and sister for all of the support, patience and understanding during my thesis work.

I would also thank all the academic and administrative staff of School of Mathematics and Computer Application, Thapar University, Patiala.

Finally, I express my deepest thanks with gratitude towards, my friends who has always supported, encouraged and contributed a lot in accomplishing this piece of work.

Pavittarjit Kaur

Place: Thapar University, Patiala
Date: July, 2014

(Pavittarjit Kaur)

ABSTRACT

One of the most basic and earliest problem of numerical analysis concerns with finding efficiency and accurately the approximate solution of the nonlinear equation of the form:

$$f(x) = 0,$$

where $f : I \subseteq \mathbb{R} \to \mathbb{R}$ is a nonlinear sufficiently differentiable function in an interval I.

Analytic methods for obtaining exact solutions of such problems are almost nonexistence. Therefore, one has to find the approximate solutions by relying on numerical methods which are based on iterative procedures. There are several one-point as well as multi-point iterative methods are available in the literature to solve these equations. Therefore, the construction of iterative methods for solving nonlinear equations is practically important and interesting task, which has attracted the attention of many researchers around the world.

Therefore, the main goal and motivation in the development of new equally competitive methods is to achieve highest computational efficiency with a fixed number of function evaluations per iteration. In our thesis, we have proposed several new one-point as well as multi-point families of methods for obtaining multiple roots of nonlinear equations. Further, we also intends to construct fourth-order optimal families of Jarratt's method. In majority of the tested examples, numerical results have shown that all the proposed methods can compete with their classical counterparts. For the comparison of iterative methods, one should take into account their convergence orders, the numerical stability, computational costs, asymptotic error constants, the dependence of convergence on the choice of initial guesses, the simple body structures, basins of attraction which provide their dynamic behavior and so on. The study of some of the mentioned features is often complicated so for better comparison, we have taken four section i.e. number of iterations, Computational order of convergence (COC), absolute error is calculated by using same total number of function evaluations (TNFE=12) and basins of attraction.

Many computer algebra software systems are available such as Mathematica, Matlab and Mapple etc. We use computer algebra software namely, Wolfram Mathem atica-9 in multi precision in the computation of nonlinear scalar equations throughout this work. We accept an approximate solution up to any specific degree rather than the exact root. Therefore, the following stopping criteria is used for computer program:

$$(i)\ |x_{n+1} - x_n| < \epsilon, \qquad (ii)\ |f(x_{n+1})| < \epsilon. \tag{0.0.1}$$

When the above stopping criterion is satisfied, x_{n+1} is consider as the required root.

Glossary of symbols

$\mathbb{R}$	is a set of real numbers
I	is an open interval
$\{x_n\}$	is a sequence
p	is the order of convergence of iterative methods
x_0	is an initial guess/approximation
ξ	is a simple root
ξ_m	is a multiple root
e_n	$e_n = x_n - \xi_m$ is the error at n^{th} iteration for multiple root
m	is the multiplicity of the required multiple root ξ_m
c_k	$c_k = \dfrac{m!}{m+k!}\dfrac{f^{(m+k)}(\xi_m)}{f^{(m)}(\xi_m)}$, $k = 1, 2, 3, \ldots$, is asymptotic error constant in the case of multiple root
$\lvert x_{n+1} - x_n \rvert$	difference between two consective iteration
$\lvert f(x_n) \rvert$	absolute value of the function at point x_n
I. M.	Iterative methods

Contents

Chapter 1

Introduction and Preliminaries

1.1 General Introduction

Numerical analysis is the area of mathematics and computer science that creates, analyzes, implement algorithms for solving numerically the problems of mathematics and applied sciences. One topic which has always been of paramount importance in numerical analysis is that of approximating efficiently and accurately the approximate multiple roots of nonlinear equation of the form

$$f(x) = 0. \tag{1.1.1}$$

where $f : I \subseteq \mathbb{R} \to \mathbb{R}$ is a nonlinear sufficiently differentiable function in an interval I.

With the large number of real world applications of the four major disciplines of engineering: chemical, civil, mechanical and electrical give rise to million of such types of problems. For example, the problem of investigating coarse-grained dynamical properties of neuronal networks in kinetic theory [RCT07], the problem of tracing the nonlinear equilibrium path of structure in mechanical engineering [SMM12], the problem of accurately estimating the molal volume (v) of the both carbon dioxide and oxygen for a number of different temperature and pressure combinations in chemical engineering [CC00a], many problems in control and optimization theory and some other mathematical problems namely, the problem of finding the eigen values of a square matrix etc. (for detail explanation one can see the literature) can be formulated as a nonlinear equation or a system of nonlinear equations. These equations may be higher order algebraic equations and possibly they may involve exponential, trigonometric and hyperbolic terms or completely be transcendental equations.

In all the above mentioned problems, we observe that analytical methods for obtaining exact solutions of such problems are almost non-existence. Therefore, one has to find the approximate solutions by relying on numerical methods which are based on iterative procedures. There are several one-point as well as multi-point iterative methods are available in the literature to solve these equations. The construction of iterative methods for solving nonlinear equations is practically important and interesting task, which has attracted the attention of many researchers around the world. Therefore, the development and analysis of some higher-order numerical methods for finding multiple roots of nonlinear equations for the present context are the main focus of present thesis.

1.2 Some basic definitions

In this section, we will provide some basic definitions and properties related to the iterative methods. All the basic concepts are given in this introductory chapter in a systematic fashion. Most symbols and notations are introduced in this chapter are global and maintain their meaning for the entire thesis.

1.2.1 Zero of a function

A zero of function $f(x)$ is a number $x = \xi$ such that $f(\xi) = 0$. Geometrically, a zero of $f(x)$ is the value of x (say $x = \xi$) at which the graph of $y = f(x)$ intersects the $x - axis$. A solution ξ of $f(x) = 0$ is called a multiple zero of $f(x)$ with multiplicity $m \geq 1$ and m is always taken as a natural number, if we can write $f(x)$ in the following way:

$$f(x) = (x - \xi)^m \ h(x), \tag{1.2.1}$$

where $h(x)$ is bounded and nonzero at $x = \xi$. For m=1, the number ξ is said to be a simple root.

Another definition of multiple root:

If $f(x)$ has at least m continuous derivatives then $x = \xi$ is called a multiple root of order m iff

$$f(\xi) = f'(\xi) = f''(\xi) = \cdots = f^{(m-1)}(\xi) = 0 \quad \text{and} \quad f^{(m)}(\xi) \neq 0.$$

1.2.2 Intermediate value property

If $f(x)$ is a continuous function on the finite interval $[a, b]$ and $f(a)f(b) < 0$, then the equation $f(x) = 0$, has at least one real root ξ in the open interval $(a,\ b)$.

1.2.3 Osculating curve

A curve $y(x)$ is osculating to curve $f(x)$ at point x_0 if it has same tangent and curvature at x_0. i.e. satisfying the following conditions.

$$\begin{aligned} y(x_0) &= f(x_0), \\ y'(x_0) &= f'(x_0), \\ y''(x_0) &= f''(x_0). \end{aligned} \tag{1.2.2}$$

1.2.4 Order of convergence

According to Atkinson [Atk89, pp. 56], a sequence of iterates $\{x_n | n \geq 0\}$ is said to converge with order $p \geq 1$ to a point ξ if

$$|\xi - x_{n+1}| \leq c|\xi - x_n|^p, \quad n \geq 0, \tag{1.2.3}$$

for some $c > 0$. If $p = 1$, the sequence $\{x_n\}$ is said to convergence linearly to ξ. In that case, we require $c < 1$; the constant c is called the rate of linear convergence of x_n to ξ. For $p = 2$ and $p = 3$, the sequence is said to converge quadratically and cubically, respectively.

This definition of order of convergence is not always a convenient one for some linearly convergent iterative methods. Using induction on (1.2.3) with $p = 1$, one can obtain

$$|\xi - x_{n+1}| \leq c^n |\xi - x_0|, \quad n \geq 0. \tag{1.2.4}$$

This shows directly the convergence of x_n to ξ. For some iterative methods one can show (1.2.4) directly, whereas (1.2.3) may not be true for any $c < 1$. In such a case, the method will still be said to converge linearly with a rate of c.

According to Schröder [Sch70], an iteration function $\phi(x)$ of the one-point iterative scheme defined below:

$$x_{n+1} = \phi(x_n), \quad n = 0, 1, 2, \ldots \tag{1.2.5}$$

is said to be of the p^{th} order if

$$\phi'(\xi) = \phi''(\xi) = \ldots \phi^{p-1}(\xi) = 0, \quad \phi^p(\xi) \neq 0, \tag{1.2.6}$$

where ξ is the solution of an equation: $x = \phi(x)$.

1.2.5 Efficiency of iterative methods

Let d be the number of new functional evaluations require per iteration and p is the order of convergence of an iterative method. Then efficiency of an iterative method, according to Traub (see [Tra64, pp. 11]) is always proportional to the order of convergence p of iterative method and inversely proportional to the number of function evaluation d per iteration. Thus

$$EFF = \frac{p}{d}, \tag{1.2.7}$$

According to Ostrowski (see [Ost60, pp. 19]) an alternative definition of an efficiency index is given by

$$E = p^{\frac{1}{d}}. \tag{1.2.8}$$

1.2.6 Computational order of convergence (COC)

Let $x_{n-1},\ x_n$ and x_{n+1} are three consecutive iterations closer to the required root $x = \xi$ and $e_n = x_n - \xi$. Then, the computational order of convergence (COC) (see

[WF00]) can be approximated by using the following formula

$$\rho \approx \frac{\ln|e_{n+1}/e_n|}{\ln|e_n/e_{n-1}|}. \tag{1.2.9}$$

The major difficulty in the application of this COC is that it requires the exact root ξ. But, there are many practical situations in which there is no prior knowledge of exact root. In order to overcome this problem, Grau-Sánchez et al. [GSNG10], proposed the new definition of COC as follows:

$$\hat{\rho} \approx \frac{\ln|\check{e}_{n+1}/\check{e}_n|}{\ln|\check{e}_n/\check{e}_{n-1}|}, \tag{1.2.10}$$

where $\check{e}_n = x_n - x_{n-1}$.

1.2.7 Fixed-point iteration

A point say s is called a fixed point if it satisfies the equation

$$x = g(x) \tag{1.2.11}$$

The equation $f(x) = 0$ converted algebraically into the form $x = g(x)$ and using the iterative scheme with the recursive relation

$$x_{i+1} = g(x_i) \tag{1.2.12}$$

with some initial guess x_0 is called the fixed point iterative scheme.

Lemma 1.2.1 *There is a interval $I = [a, b]$ such that, for all $x \in I$, $g(x)$ is defined and $g(x) \in I$, that is the function $g(x)$ maps I into itself.*

Lemma 1.2.2 *The iteration function $g(x)$ is continuous on $I = [a, b]$.*

Lemma 1.2.3 *The iteration function is differentiable on $I = [a, b]$. Further, there exists a non negative constant $k < 1$ such that for all $x \in I$, $|g'(x)| \leq k$*

Theorem 1.2.4 *Let $g(x)$ be an iterative function satisfying Lemma (1.2.1)- (1.2.3). Then $g(x)$ has exactly one fixed point ξ in I, and starting with any point $x_0 \in I$, the sequence $x_1,\ x_2,\ \ldots,\ x_n$, generated by fixed point iteration method converges to ξ.*

1.3 Classification of iterative methods

There are many types of classification of iterative methods. But we have divide them into the following two categories namely,

(A) Based on convergence

(B) Based on memory

1.3.1 Based on Convergence

A root-finding algorithm is a numerical method or algorithm, for obtaining a value $x = \xi$ such that $f(x) = 0$. Such point $x = \xi$ is called a zero of the function $f(x)$. It further divides into two categories namely, first one is the bracketing methods and second one is the open methods.

(a) Bracketing Methods: In bracketing methods, these methods exploit the fact that a function typically changes sign in the vicinity of a root. They started with two initial guesses that bracket the root and then systematically reduce the width of the bracket until the solution to a desired accuracy is achieved. Some of the well-known examples of the bracketing methods are

(i) Bisection method.

(ii) False position or regula-falsi method.

(b) Open Methods: These methods require one or more than one initial approximation for finding the required root of nonlinear equations. But, they do not necessarily bracketing the root. Therefore sometimes these methods diverge or overshoot i.e move away from the required root as the computation progresses. However, whenever they converge, they usually do so much more quickly than the bracketing methods. Some of the well-known examples of the open methods are

(i) Secant method.

(ii) Newton-Raphson method.

1.3.2 I.M. Based on Memory

In 1964, Traub had further classified iterative methods on the basis of the information they require into the following four categories:

(i) One-point I. M. without memory: If the estimation of root can be determined only by using new information at one-point and there is no need of old information, then method is called one-point iterative method. Thus

$$x_{k+1} = \phi(x_k), \tag{1.3.1}$$

where x_{k+1} is determined by new information at x_k, ϕ is one -point iterative function without memory. Newton's method, Chebyshev's method, Halley's method, super-Halley method etc. are well-known examples of such types of method.

(ii) One-point I. M. with memory: In these methods estimation to root is determined by using new information at one point and old information is used at one or more points. In these methods old information is used. Therefore these methods called one-point method with memory. Thus

$$x_{k+1} = \phi(x_k;\ x_{k-1},\ \ldots,\ x_{k-n}). \tag{1.3.2}$$

However, only x_k is new information, while $x_{k-1},\ \ldots,\ x_{k-n}$ are reused information, which is indicated by the inserted semicolon. Secant method is well-known examples of one-point iterative methods with memory.

(iii) Multi-point I. M. without memory: An iterative method is constructed by introducing the expression, $w_1(x_k), w_2(x_k), \ldots, w_n(x_k)$, where x_k is the common argument. Then, iterative method ϕ, defined as

$$x_{k+1} = \phi\big(x_k, w_1(x_k), w_2(x_k), \ldots, w_n(x_k)\big), \tag{1.3.3}$$

is known as multi-point I. M. without memory. Newton-Secant method is the well-known example of such methods.

(iv) Multi-point I. M. with memory: If an estimation to a root is determined by new information at number of points with reusing the old information at some other points, then iterative method is called a multi-point iterative method with memory. Mathematically, let z_j represents the $k+1$ quantities $x_j,\ w_1(x_j),\ w_2(x_j),\ \ldots,\ w_k(x_j)$, $k \geq 1$. Thus

$$x_{k+1} = \phi(z_n; z_{n-1}, \ldots, z_{n-s}). \tag{1.3.4}$$

Then ϕ is called a multi-point iterative function with memory. The semicolon separates in (1.3.4) the points at which new data are used from the points at which old data is reused. There is no well-known example of multi-point iterative method with memory.

1.4 Literature survey

One of the most basic and earliest problem of numerical analysis concerns with finding efficiency and accurately the approximate solution of the nonlinear equation (1.1.1). There are vast literature for numerical solution of these problems such types of problems are also called root-finding problems. There are two types of methods to solving such problems, as follows:

(i) Direct Methods

(ii) Iterative Methods

Direct Methods: These methods provides the solution of nonlinear equations in a finite number of steps. These methods also determines all the solutions at the same time. Many of times these methods not able to solve the problem. For instance, the roots of the quadratic equation by Sidharathchariya method. We have methods available for 3 rd degree Cardan's method, for 4^{th} degree Ferrari's method. There is no direct method available in the literature that can solve every polynomial equation of degree greater than four. So, we hope for approximate solutions. For this we turn towards the iterative procedures .

Iterative Methods: These methods are based on idea of successive approximations. In these methods we begin with one or more initial approximations to the root and determine a sequence of approximations by repeating a fixed sequence of steps till we obtain the solution of the nonlinear equations with desired accuracy. In these methods round off errors are negligible in comparison to Direct Methods and we can develop algorithm which can handle class of similar problems. Reliability of these methods on the some characteristics of method such as convergence, efficiency, stability for finding the solution of nonlinear equations.

Generally, there are two major difficulties in the application of the iterative methods. Firstly, they do not provide the exact value of the required root. Therefore, researchers always keen to provide a method or scheme that will yield an approximate root even though it may be differ from the exact root but with in a specific tolerance limit. Secondly, there exists no well-known iterative method can be applied to all types of problems. Therefore, in our point of view these are two main reasons for huge variety of research papers on iterative methods. Therefore, one can refer to some of excellent text books such as Traub [Tra64], Ostrowski [Ost60, Ost73], Atkinson [Atk89], Chapra and Canale [CC00b], McNamee [McN07], Petković et al. [PNPD12] and McNamee and Pan [MP13] for a good review of these important methods.

With the advancement of computer algebra, many researchers paid a great attention to propose methods or schemes for obtaining multiple roots of the nonlinear equations. For the case of multiple roots, there are two cases, namely

(i) when the multiplicity m of the multiple root is known in advance.

(ii) when the multiplicity m of the multiple root is not known in advance.

First of all, we consider the case when the multiplicity $m \geq 1$ of the required root is known in advance. The best known and most widely used algorithm for solving such problems is the quadratically convergent Rall's method,[Ral66]is given by

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}, n = 0, 1, 2, ... \tag{1.4.1}$$

In order to improve the local order of convergence of modified Newton's method (also known as Rall's method), many researchers proposed and analyzed many higher order methods [Tra64, HP77, Net10, Osa08, Osa94, Osa06, CN09, CBN09, BG10, SS11, KKS12] etc., with the assumption that multiplicity m of the root is known in advance.

All the previously mentioned methods are one-point and they have some drawback regarding their convergence and efficiency. Therefore, researchers turn towards to multi-point methods and it is one of the most important class of iterative methods. During the recent and past years, several multi-point methods have been proposed and analyzed by Dong [Don82, Don87], Victory and Neta [VN83], Neta [Net10], Basu [Bas08], Homeier [Hom09], Li et al. [LLC09], Chun and Neta [CN09], Chun et al. [CBN09], Kim and Lee [KL10], Parida and Gupta [PG10], Heydari et al. [HHL10], Kim and Geum [KG11], Kumar et al. [KKS12] and Zhou et al. [ZCS13] etc., with prior knowledge of multiplicity m. These methods have been proven to be competitive to Newton's method in their performance and efficiency. There are, however, not yet so many third and fourth-order two-point methods known in open literature that can handle the case of multiple roots.

In the recent years, some fourth-order multi-point methods have been proposed and analyzed by Neta and Johnson [NJ08], Mir and Rafiq [MR07b, MR07a] and Neta [Net10], etc. for finding multiple roots. However, all these third and fourth-order multi-point methods are not optimal according to the Kung-Traub conjecture [KT74]. Therefore, obtaining new optimal methods of order four not requiring the computation of a second-order derivative, is very important and interesting task from a practical point of view, because their corresponding efficiency index is very competitive.

Motivated and inspired by the recent activities in this direction, researchers proposed some fourth-order optimal multi-point methods Li et al. [SXL09], Sharma and Sharma [SS10a], Li et al. [LCN10] (proposed six fourth-order two-point methods with closed formulas but only two methods namely, method 69 and 75 are optimal), Zhou et al. [ZCS11, ZCS13] and Soleymani et al. [SBL13].

Now, we consider the second case when the multiplicity of the multiple root ξ_m is not known in advance. In this case, there exits a well-known method of transformation by which multiple zeros of $f(x)$ is transformed to a simple zero and is given as follows:

$$\hat{G}(x) = \begin{cases} \frac{f(x)}{f'(x)}, & for \quad f'(x) \neq 0, \\ 0, & for \quad f'(x) = 0. \end{cases} \tag{1.4.2}$$

For instance, if we apply this transformation on $\hat{G}$ instead of function f in a well-known quadratically convergent Newton's method for simple root, then we can easily obtain the following scheme:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{\{f'(x_n)\}^2 - f(x_n)f''(x_n)}. \tag{1.4.3}$$

This is a well-known quadratically convergent Schröder's method [Tra64] for simple root as well as multiple root. The beauty of this method is that it does not require multiplicity m to be known in advance. But, by applying the transformation (1.4.2), one has to calculate the value of certain order derivatives of a function as well as function at every iteration. Therefore, this approach might become problematic due to higher computational costs. In order to overcome of this problem, researcher developed some schemes which are based upon the use of interpolation, quadrature formulas, weight functions, finite differences or a combination of such approaches. For detailed study of such approaches one can refer to some excellent text books such as Traub [Tra64], McNamee [McN07] and Petković et al. [PNPD12].

In this thesis, we will present many new computationally effective root-finding methods and several well-known methods are obtained as special cases. The main goal and motivation in the development of new equally competitive methods is to achieve highest computational efficiency with a fixed number of function evaluations per iteration. In our thesis, we have proposed several new one-point as well as multi-point families of methods for obtaining multiple roots of nonlinear equations. Further, we also intends to construct fourth-order optimal families of Jarratt's method. In majority of the tested examples, numerical results have shown that all the proposed methods can compete with their classical counterparts. For the comparison of iterative methods, one should take into account their convergence orders, the numerical stability, computational costs, asymptotic error constants, the dependence of convergence on the choice of initial guesses, the simple body structures, basins of attraction [SNC11, NSC12] which provide their dynamic behavior and so on. The study of some of the mentioned features is often complicated so for better comparison, we have taken four section i.e. number of iterations, Computational order of convergence (COC), absolute error is calculated by using same total number of function evaluations (TNFE=12) and basins of attraction.

Many computer algebra software systems are available such as Mathematica, Matlab and Mapple etc. We use computer algebra software namely, Wolfram Mathem atica-9 in multi precision in the computation of nonlinear scalar equations (multiple roots) throughout this work. We accept an approximate solution up to any specific degree rather than the exact root. Therefore, the following stopping criteria is used for computer program:

$$(i)\ |x_{n+1} - x_n| < \epsilon, \qquad (ii)\ |f(x_{n+1})| < \epsilon. \tag{1.4.4}$$

When the above stopping criterion is satisfied, x_{n+1} is consider as the required root.

Chapter 2

Third-order variants of Chebyshev's method

2.1 Introduction

In this chapter, we proposed a new one-parameter family of Chebyshev type methods for finding multiple roots with cubic convergence when the multiplicity of the required root is known in advance. Further, we proposed many new multi-point families of third-order methods for multiple roots which are free from second-order derivative. In the majority of tested examples our methods are equally competent and better to other well-known methods for multiple roots.

In the past and recent years, many one-point and multi-point iterative methods have been proposed and analyzed by Hansen and Patrick [HP77], Dong [Don82, Don87], Victory and Neta [VN83], Osada [Osa94], Neta and Johnson [NJ08], Neta [Net08], Chun and Neta [CN09], Chun et al. [CBN09], Li et al. [LLC09], kumara et al. [Kum12] and the references cited therein.

Finding higher-order iterative methods, not requiring the computations of second-order derivative for multiple roots is very important and interesting task for the practical point of view. Motivated by research going in this direction, we proposed a general families of one-point as well as multi-point iterative methods, with cubic convergence for finding the multiple roots of non-linear equations numerically. Further the classical Chebyshev method for multiple roots is obtained as particular case of our proposed scheme. The numerical tests show their equal performance in the case of algebraic as well as non-algebraic equations.

2.2 Family of Chebyshev's Method

Let ξ_m be the required multiple root of equation (1.1.1) and $x = x_0$ be the initial guess known for the required root. Assume

$$x_1 = x_0 + h, \qquad |h| << 1, \tag{2.2.1}$$

be the first approximation to the root. Therefore

$$f(x_1) = 0. \tag{2.2.2}$$

Expanding the function $f(x_1)$ by the Taylor's Series expansion about x_0 and retaining the terms up to $O(h^2)$, we get

$$f(x_0) + hf'(x_0) + \frac{h^2}{2} f''(x_0) = 0. \tag{2.2.3}$$

Further, we can rewrite the equation (2.2.3) in the following way

$$h = -\frac{f(x_0)}{f'(x_0)} - \frac{h^2 f''(x_0)}{2f'(x_0)}. \tag{2.2.4}$$

Here, the function $\left(-\frac{f(x_0)}{f'(x_0)} - \frac{h^2 f''(x_0)}{2f'(x_0)}\right)$, which occurs on the right-hand side of equation (2.2.4) can not be computed till h is known. To overcome this problem, we can substitute the value of $h = -\frac{f(x_0)f'(x_0)}{\{f'(x_0)\}^2 - af(x_0)f''(x_0)}$, $a \in \mathbb{R}$ (free disposable parameter) from the correction factor of Schröder's method (1.4.3) in the formula (2.2.4), we obtain

Approximating h on the right-hand side of equation(2.2.4) by the correction term we obtain

$$h = -\frac{f(x_0)}{f'(x_0)} - \frac{1}{2}\frac{\{f(x_0)\}^2 f'(x_0) f''(x_0)}{\left[\{f'(x_0)\}^2 - af(x_0)f''(x_0)\right]^2}. \tag{2.2.5}$$

Thus the first approximation to the required root is given by

$$x_1 = x_0 - \theta_2 \frac{f(x_0)}{f'(x_0)} - \frac{1}{2}\frac{\theta_1 \ \{f(x_0)\}^2 f'(x_0) f''(x_0)}{\left[\{f'(x_0)\}^2 - a \ f(x_0)f''(x_0)\right]^2}, \tag{2.2.6}$$

where θ_1 and θ_2 are two free disposable parameters such that the convergence of this scheme reaches at third-order without using any more functional evaluations. Therefore, the general formula for successive approximation can be written as

$$x_{n+1} = x_n - \theta_2 \frac{f(x_n)}{f'(x_n)} - \frac{1}{2} \frac{\theta_1 \; \{f(x_n)\}^2 f'(x_n) f''(x_n)}{\left[\{f'(x_n)\}^2 - a \; f(x_n) f''(x_n)\right]^2}. \tag{2.2.7}$$

The following theorem (2.2.1) indicates that under what conditions on the disposable parameters namely, θ_1 and θ_2, the convergence will reach at third-order.

2.2.1 Convergence Analysis

Theorem 2.2.1 *Let $f : D \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function defined on an open interval D, enclosing a multiple zero of $f(x)$, say $x = \xi_m$ with multiplicity $m \geq 1$ and $x = x_0$ be the sufficiently close initial guess. Then the family of iterative methods defined by* (2.2.7) *has cubic convergence,*

$$\begin{cases} \theta_1 = -\dfrac{(a(m-1)-m)^3}{a(m-1)+m}, \\ \theta_2 = \dfrac{m\left(-(m-3)m + a\left(m^2-1\right)\right)}{2(a(m-1)+m)}, \end{cases} \tag{2.2.8}$$

such that satisfies the following error equation:

$$e_{n+1} = \left((2-4a)c_1^2 - c_2\right) e_n^3 + O(e_n^4) \tag{2.2.9}$$

Proof Let $x = \xi_m$ be a multiple zero of $f(x)$. Expanding $f(x_n)$, $f'(x_n)$ and $f''(x_n)$ about $x = \xi_m$ by the Taylor's series expansion (with the help of computer algebra software Mathematica 9), we get

$$f(x_n) = \frac{f^{(m)}(\xi_m)}{m!} e_n^m \left(1 + c_1 e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4\right) + O(e_n^5), \tag{2.2.10}$$

$$f'(x_n) = \frac{f^m(\xi_m)}{m!} e_n^{m-1} \left(m + (1+m)c_1 e_n + (2+m)c_2 e_n^2 + (3+m)c_3 e_n^3\right) + O(e_n^4), \tag{2.2.11}$$

and

$$\begin{aligned} f''(x_n) =& \frac{f^m(\xi_m)}{m!} e_n^{m-2} \left(-m + m^2 + m(1+m)c_1 e_n + \left(2 + 3m + m^2\right) c_2 e_n^2 \right. \\ & \left. + (6c_3 + 5mc_3 + m^2 c_3) e_n^3\right) + O(e_n^4), \end{aligned} \tag{2.2.12}$$

respectively.

From equations (2.2.10) and (2.2.11), we have

$$\frac{f(x_n)}{f'(x_n)} = \frac{1}{m}e_n - \frac{c_1}{m^2}e_n^2 + \left(\frac{(m+1)c_1^2 - 2mc_2}{m^3}\right)e_n^3 + O(e_n^4). \tag{2.2.13}$$

Further, by using equations (2.2.10), (2.2.11) and (2.2.12), we obtain

$$\begin{aligned}\frac{\theta_1\ \{f(x_n)\}^2 f'(x_n) f''(x_n)}{[\{f'(x_n)\}^2 - a\ f(x_n)f''(x_n)]^2} &= \frac{\theta_1(-1+m)e_n}{(a+m-am)^2} + \frac{\theta_1\left(a+(-3+m)m-am^2\right)c_1e_n^2}{(a(-1+m)-m)^3 m} + \\ &\frac{1}{m(a+m-am)^4}\theta_1\Big(\left(a^2(-1+m)(1+m)^2 + m\left(-6-3m+m^2\right) + 2a\left(3+m+m^2-m^3\right)\right) \\ &c_1^2 - 2\left(-2a(-1+m)^2m + (-4+m)m^2 + a^2\left(2-3m+m^3\right)\right)c_2\Big)e_n^3 + O(e_n^4).\end{aligned} \tag{2.2.14}$$

By using equations (2.2.13) and (2.2.14) in (2.2.7), we can have following error equation

$$\begin{aligned}e_{n+1} = &\left(1 - \frac{\theta_1(-1+m)}{2(a+m-am)^2} - \frac{\theta_2}{m}\right)e_n + \frac{c_1e_n^2}{2m^2}\left(-\frac{\theta_1 m\left(a+(-3+m)m-am^2\right)}{(a(-1+m)-m)^3} + 2\theta_2\right) \\ &+ \Bigg(-\frac{\theta_2\left((1+m)c_1^2 - 2mc_2\right)}{m^3} - \frac{1}{2m(a+m-am)^4}\theta_1\Big(\big(a^2(-1+m)(1+m)^2 \\ &+ m\left(-6-3m+m^2\right) + 2a\left(3+m+m^2-m^3\right)\big)c_1^2 - 2\big(-2a(-1+m)^2m \\ &+ (-4+m)m^2 + a^2\left(2-3m+m^3\right)\big)c_2\Big)\Bigg)e_n^3 + O(e_n^4).\end{aligned} \tag{2.2.15}$$

For obtaining an iterative method of order three, the coefficients of e_n and e_n^2 in the error equation (2.2.15) must be zero simultaneously. After simplifying the equation (2.2.15), we have the following values of θ_1 and θ_2 involving one free disposable parameter a

$$\begin{cases} \theta_1 = -\dfrac{(a(m-1)-m)^3}{a(m-1)+m}, \\ \theta_2 = \dfrac{m\left(-(m-3)m + a\left(m^2-1\right)\right)}{2(a(m-1)+m)}. \end{cases} \tag{2.2.16}$$

Using equations (2.2.13), (2.2.14) and (2.2.16) in (2.2.7), we get

$$e_{n+1} = \left((2-4a)c_1^2 - c_2\right)e_n^3 + O(e_n^4). \tag{2.2.17}$$

This completes the proof of the Theorem 2.1. □

2.2.2 Special cases

Finally, a family of Chebyshev type methods is obtain by inserting the values of free disposable parameters namely θ_1 and θ_2 in equation (2.2.7). Therefore the family of Chebyshev-type method is given by equation (2.2.18)

$$x_{n+1} = x_n - \frac{f(x)\left(-\frac{f''(x)\{f'(x)\}^2 f(x)(a(-1+m)-m)^3}{(\{f'(x)\}^2 - af''(x)f(x))^2} + m\left(-(-3+m)m + a\left(-1+m^2\right)\right)\right)}{2f'(x)(a(-1+m)+m)}. \tag{2.2.18}$$

Now for different specific values of the disposable parameter a in proposed scheme (2.2.18), we gets various one-point families of iterative methods for multiple roots as follow:

(i) For $a = 0$, we get the following formula

$$x_{n+1} = x_n - \frac{m(m-3)f(x_n)}{2f'(x_n)} - \frac{m^2\{f(x_n)\}^2 f''(x_n)}{2\{f'(x_n)\}^3}. \tag{2.2.19}$$

This is a well known Chebyshev's method for multiple roots.

(ii) For $a = 1$, one obtains the following formula

$$x_{n+1} = x_n - \frac{f(x_n)}{2f'(x_n)(-1+2m)}\left[\frac{f''(x_n)f'(x_n)^2 f(x_n)}{((f'(x_n))^2 - f''(x_n)f(x_n))^2} + m(3m-1)\right]. \tag{2.2.20}$$

This is a new cubically convergent method for multiple roots.

(iii) For $a = 1/2$, one obtains the following formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)(3m-1)}\left[\frac{f''(x_n)\{f'(x_n)\}^2 f(x_n)(3m+1)^3}{8\left(\{f'(x_n)\}^2 - \frac{f''(x_n)f(x_n)}{2}\right)^2} - \frac{m}{2}\left(m^2 - 6m + 1\right)\right]. \tag{2.2.21}$$

This is another new cubically convergent method for multiple roots.

2.3 Third-order multi-point families of iterative methods

The major difficulty in the application of recently developed methods is that they require the lengthy computation of second-order derivative at each step. Generally,

there are many practical situations in which the calculation of second-order derivative is expensive and/or it requires a great deal of time for them to be given or calculated. Therefore finding iterative methods with cubic convergence for multiple root which are free from second-order derivative is an interesting task in numerical mathematics. During the recent and past years, many papers devoted to multi-point methods for solving nonlinear equations have appeared in different journals, see [Net08] and references cited therein [Net08, Don87, Don82, LLC09, Kum12, CN09, CBN09] and the references cited therein. In terms of functional evaluation per iteration, all these methods may be broadly categorized into two classes:
(i) two functions and one first-order derivative,
(ii) one function and two first-order derivatives.

Here, author also intends to develop new third-order multi-point methods free from second-order derivative. The proposed methods require either two f and one f' or one f and two f' per iteration. The main idea of proposed methods lies in the discretization of second-order derivative involved in the recently mentioned methods.

2.3.1 Multi-point I. M. with two f and one f'

For obtaining third-order multi-point methods with two evaluations of $f(x)$ and one evaluation of $f'(x)$, one can expand the function $f\left(x_n - \theta\frac{f(x_n)}{f'(x_n)}\right)$ where $\theta \neq 0 \in \mathbb{R}$, about the point $x = x_n$ with $f'(x_n) \neq 0$, by Taylor's series expansion as follows:

$$f(y_n) = f(x_n) - \theta\left(\frac{f(x_n)}{f'(x_n)}\right) f'(x_n) + \frac{\theta^2}{2}\left(\frac{f(x_n)}{f'(x_n)}\right)^2 f''(x_n) + O\left(\theta\frac{f(x_n)}{f'(x_n)}\right)^3,$$

where

$$y_n = x_n - \theta\frac{f(x_n)}{f'(x_n)}. \tag{2.3.1}$$

Therefore, one obtains

$$f''(x_n) \approx \frac{2\{f(y_n) - f(x_n) + \theta f(x_n)\}\{f'(x_n)\}^2}{\theta^2\{f(x_n)\}^2}. \tag{2.3.2}$$

Using this approximate value of $f''(x_n)$ and (2.3.1) into the recently proposed formula (2.2.7), one gets different multi-point iterative methods free from second-order derivative as

$$x_{n+1} = x_n - \frac{1}{2}\left[\frac{\theta_3\ f(x_n)m(-1+3m)}{f'(x_n)(-2+4m)} + \frac{\theta_4\ (f'(x_n)-f'(y_n))f(x_n)\theta}{2(f'(y_n)+f'(x_n)(-1+\theta))^2(-1+2m)}\right], \tag{2.3.3}$$

where θ_3 and θ_4 are two new free disposable parameters such that the order of convergence reaches at three without using any more functional evaluations. Theorem 2.3.1 indicates that under what choices on the disposable parameters in (2.3.3), the order of convergence will reach at three.

2.3.2 Convergence analysis

Theorem 2.3.1 *Let $f : D \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function defined on an open interval D, enclosing a multiple zero of $f(x)$, say $x = \xi_m$ with multiplicity $m \geq 1$ and $x = x_0$ be the sufficiently close initial guess. Then the family of iterative methods defined by* (2.2.7) *has cubic convergence, when*

$$\begin{cases} \theta_3 = -\dfrac{\left(-1+\frac{(-1+\theta+\mu)\left(-2+2\theta-\theta^2+2\mu\right)\mu^{-1}(\theta-m)}{\theta^2(-2+2\theta+\theta^2+2\mu)}\right)(2-4m)}{1-3m}, \\ \theta_4 = \dfrac{\left(-2+2\theta-\theta^2+2\mu\right)^3\mu^{-1}(\theta-m)m(-1+2m)}{\theta^4\left(-2+2\theta+\theta^2+2\mu\right)}. \end{cases} \tag{2.3.4}$$

It satisfies the following error equation:

$$\begin{aligned} e_{n+1} =&\Bigg(\bigg(\Big(-2\theta^5(1+m)-4\left(-1+\mu\right)^2m(3+m)+\theta^3\left(8-8\mu+8m\right)+ \\ &\theta^4\left(-8\mu+3m+m^2\right)-4\theta^2\left(4-6\mu+2u^2+7m-4\mu m+m^2\right) \\ &+8\theta\Big(\left(-1+\mu\right)^2+\left(4-5\mu+\mu^2\right)m-\left(-1+\mu\right)m^2\Big)\Big)c_1^2 \\ &+2\Big(4\theta^2-\theta^4+8\theta\left(-1+\mu\right)+4\left(-1+\mu\right)^2\Big)(\theta-m)^2c_2\bigg)e_n^3\Bigg) \\ &\left(2\left(-2+2\theta-\theta^2+2\mu\right)\left(-2+2\theta+\theta^2+2\mu\right)(\theta-m)m^2\right)+O(e_n^4). \end{aligned} \tag{2.3.5}$$

where $\theta \neq 0,\ m$ *and* $\mu = \left(1-\frac{\theta}{m}\right)^m$.

Proof Using equation (2.2.12) and expanding $f\left(x_n - \theta\frac{f(x_n)}{f'(x_n)}\right)$ about $x = \xi_m$ by the Taylor's series expansion (with the help of computer algebra software Mathematica

9), one obtains

$$f\left(x_n-\theta\frac{f(x_n)}{f'(x_n)}\right)=e_n^m\frac{\left(1-\frac{\theta}{m}\right)^m}{m!}\left[1+\frac{(m^2-m\theta+\theta^2)\,c_1e_n}{m(m-\theta)}+\theta_5e_n^2+\theta_6e_n^3\right]+O(e_n^4), \tag{2.3.6}$$

where

$$\begin{aligned}\theta_5=&\frac{1}{2m^2(m-\theta)^2}\{\theta^2(-m^2+2\theta+m(-3+2\theta))c_1^2+2(m^4-2m^3\theta+4m^2\theta^2-4m\theta^3+\theta^4)c_2\},\\ \theta_6=&\frac{1}{6m^3(m-\theta)^3}\{\theta^2(3m^4+m^3(15-11\theta)+6\theta^2+m\theta(-16+15\theta)+3m^2(4-11\theta+3\theta^2))c_1^3\\ &-6\theta^2(2m^4+m^3(8-7\theta)-m(-10+\theta)\theta^2-2\theta^3+2m^2\theta(-8+3\theta))c_1c_2+6(m-\theta)^2(m^4\\ &-m^3\theta+6m^2\theta^2-4m\theta^3+\theta^4)c_3\}.\end{aligned} \tag{2.3.7}$$

Using equations (2.2.10), (2.2.11) and (2.3.6), in scheme (2.3.3) we can have

$$\begin{aligned}e_{n+1}=&\left(1-\frac{\frac{\theta_3(1-3m)m}{2-4m}+\frac{\theta_4\theta^2(-1+\theta+\mu)}{(2-2\theta+\theta^2-2\mu)^2(-1+2m)}}{m}\right)e_n\\ &+\frac{\left(\frac{\theta_3(1-3m)m}{2-4m}+\frac{\theta_4\theta^2(-1+\theta+\mu)}{(2-2\theta+\theta^2-2\mu)^2(-1+2m)}+\frac{\theta_4\theta^4\left(-2+2\theta+\theta^2+2\mu\right)\mu}{(2-2\theta+\theta^2-2\mu)^3(\theta-m)(-1+2m)}\right)c_1e_n^2}{m^2}\\ &-\frac{1}{2m^3}\Bigg(-\frac{2\theta_4\theta^4\left(-2+2\theta+\theta^2+2\mu\right)\mu c_1^2}{\left(-2+2\theta-\theta^2+2\mu\right)^3(\theta-m)(-1+2m)}\\ &+2\left(\frac{\theta_3(1-3m)m}{2-4m}+\frac{\theta_4\theta^2\left(-1+\theta+\mu\right)}{\left(2-2\theta+\theta^2-2\mu\right)^2\left(-1+2m\right)}\right)\left((1+m)c_1^2-2mc_2\right)\\ &+\Big(\theta_4\theta^4\mu\Big(\Big(8\theta^3\left(-1+\mu-2m\right)+12\left(-1+\mu\right)^2m(1+m)+\theta^5(2+4m)\\ &+\theta^4\left(8\mu-3m-3m^2\right)+4\theta^2\left(4-6\mu+2\mu^2+11m-8\mu m+3m^2\right)\\ &-8\theta\left((-1+\mu)^2+\left(5-7\mu+2\mu^2\right)m-3\left(-1+\mu\right)m^2\right)\Big)c_1^2\\ &+2\left(-4\theta^2+\theta^4-8\theta\left(-1+\mu\right)-4\left(-1+\mu\right)^2\right)\left(\theta^2-4\theta m+3m^2\right)c_2\Big)\Big)/\\ &\Big(\left(2-2\theta+\theta^2-2\mu\right)^4(\theta-m)^2(-1+2m)\Big)\Bigg)e_n^3+O(e_n^4)\end{aligned} \tag{2.3.8}$$

where $\theta\neq 0,\ m$.

For obtaining a multi-point family of third-order methods, the coefficients of e_n and

e_n^2 in above error equation must be zero simultaneously for θ_3 and θ_4, we obtain

$$\begin{cases} \theta_3 = -\dfrac{\left(-1 + \frac{(-1+\theta+\mu)\left(-2+2\theta-\theta^2+2\mu\right)\mu^{-1}(\theta-m)}{\theta^2(-2+2\theta+\theta^2+2\mu)}\right)(2-4m)}{1-3m}, \\ \theta_4 = \dfrac{\left(-2+2\theta-\theta^2+2\mu\right)^3\mu^{-1}(\theta-m)m(-1+2m)}{\theta^4\left(-2+2\theta+\theta^2+2\mu\right)}. \end{cases} \tag{2.3.9}$$

By using these values of disposable parameters in equation (2.3.3), we get

$$\begin{aligned} e_{n+1} = &\Bigg(\Bigg(\Bigg(-2\theta^5(1+m) - 4\left(-1+\mu\right)^2 m(3+m) + \theta^3\left(8-8\mu+8m\right) + \\ &\theta^4\left(-8\mu+3m+m^2\right) - 4\theta^2\left(4-6\mu+2\mu^2+7m-4\mu m+m^2\right) \\ &+ 8\theta\Bigg(\left(-1+\mu\right)^2 + \left(4-5\mu+\mu^2\right)m - \left(-1+\mu\right)m^2\Bigg)\Bigg)c_1^2 \\ &+ 2\Bigg(4\theta^2-\theta^4+8\theta\left(-1+\mu\right)+4\left(-1+\mu\right)^2\Bigg)(\theta-m)^2c_2\Bigg)e_n^3\Bigg) \\ &\left(2\left(-2+2\theta-\theta^2+2\mu\right)\left(-2+2\theta+\theta^2+2\mu\right)(\theta-m)m^2\right) + O(e_n^4). \end{aligned} \tag{2.3.10}$$

This completes the proof of Theorem 2.3.1. □

2.3.3 Multi-point I. M. with one f and two f'

Now for obtaining third-order multi-point methods with two evaluations of $f(x)$ and one evaluation of $f'(x)$, one can expand the function $f'\left(x_n - \theta\frac{f(x_n)}{f'(x_n)}\right) = f'(y_n)$ where $\theta \neq 0, m \in \mathbb{R}$, about the point $x = x_n$ with $f'(x_n) \neq 0$, by Taylor's series expansion as follows:

$$f'(y_n) = f'(x_n) - \theta\left(\frac{f(x_n)}{f'(x_n)}\right)f''(x_n). \tag{2.3.11}$$

Therefore, one obtains

$$f''(x_n) \approx \frac{\{f'(x_n) - f'(y_n)\}f'(x_n)}{\theta f(x_n)}. \tag{2.3.12}$$

Using this approximate value of $f''(x_n)$ into the recently proposed formula (2.2.7), we get different multi-point iterative methods free from second-order derivative as

$$x_{n+1} = x_n - \frac{1}{2}\left[\frac{\theta_7\, f(x_n)m(-1+3m)}{f'(x_n)(-2+4m)} + \frac{\theta_8\,(f'(x_n)-f'(y_n))f(x_n)\theta}{2(f'(y_n)+f'(x_n)(-1+\theta))^2(-1+2m)}\right], \tag{2.3.13}$$

where θ_7 and θ_8 are two new free disposable parameters such that the order of convergence reaches at three without using any more functional evaluations .

2.3.4 Convergence analysis

Theorem 2.3.2 *Let $f : D \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function defined on an open interval D, enclosing a multiple zero of $f(x)$, say $x = \xi_m$ with multiplicity $m \geq 1$ and $x = x_0$ be the sufficiently close initial guess. Then the family of iterative methods defined by* (2.3.13) *has cubic convergence, when*

$$\begin{cases} \theta_7 = \dfrac{(-2+4m)\left(-1+\frac{\mu^{-1}(\theta-m)(\theta+(-1+\mu)m)\left(-\theta^2+(-1+\mu)m+\theta(1+m)\right)}{\theta(\theta-2m+\theta m)(\theta+\theta^2-\theta m+(-1+\mu)m)}\right)}{1-3m}, \\ \theta_8 = \dfrac{2\mu^{-1}m(-1+2m)\left(-\theta^2+(-1+\mu)\,m+\theta(1+m)\right)^3}{\theta^2(\theta-2m+\theta m)\left(\theta+\theta^2-\theta m+(-1+\mu)\,m\right)}. \end{cases} \tag{2.3.14}$$

and satisfies the following error equation:

$$\begin{aligned} e_{n+1} =& \Bigg(\bigg(\Big(2\theta^6(1+m)^2 - 2\,(-1+\mu)^2\,m^4(3+m) + \theta\,(-1+\mu)\,m^3 \\ & \left(2\,(-9+7\mu)+(-13+9\mu)\,m+(-1+\mu)\,m^2\right) \\ & + \theta^5\left(4\mu+2\,(-5+4\mu)\,m+(-17+4\mu)\,m^2-5m^3\right) \\ & + 2\theta^2m^2\left(-10+16\mu-5\mu^2-3\left(4-3\mu+2u^{2m}\right)m+\left(1+3\mu-\mu^2\right)m^2+m^3\right) \\ & + 2\theta^4(1+m)\left(-1+\mu-(1+9\mu)\,m-2\,(-5+\mu)\,m^2+2m^3\right) \\ & - \theta^3m\big(-2\left(5-7\mu+\mu^2\right)-\left(17+12\mu+4\mu^2\right)m+\left(13-10\mu-2\mu^2\right)m^2 \\ & + 13m^3+m^4\big)\Big)c_1^2 + 2(\theta-m)^2(-2m+\theta(2+m))\big(\theta^4-2\theta^3m-2\theta\,(-1+\mu)\,m \\ & - (-1+\mu)^2\,m^2+\theta^2\left(-1+m^2\right)\big)c_2\bigg)e^3\Bigg)/\big(2(\theta-m)m^2(\theta-2m+\theta m) \\ & \left(\theta+\theta^2-\theta m+(-1+\mu)\,m\right)\left(\theta^2+m-\mu m-\theta(1+m)\right)\big)+O(e_n^4). \end{aligned} \tag{2.3.15}$$

Proof The proof of this theorem is similar to Theorem 2.3.1, so omitted here.

2.4 Numerical experiments

In this section, we will check the consistency and effectiveness of newly proposed one-point as well as multi-point methods namely, method (2.2.20) (OM_3^1), (2.2.21) (OM_3^2), method (2.3.3) for ($\theta = 1$) (OM_3^3), for ($\theta = 1/4$) (OM_3^4) and method (2.3.13) for ($\theta = 1$) (OM_3^5), for ($\theta = 1/4$) (OM_3^6), respectively to solve the nonlinear equations with roots of known multiplicity m, given in Table 2.1. These proposed methods were compared with the existing methods namely, super-Halley method (SH_3), Halley's Method (HM_3), Ostrowski's square-root method (OSM_3), Kumara et al. method (KM_1), method (KM_2) [Kum12], Neta's method (NM_3) [Net10], Li et al. method (M54), Dong's method (DM_3^1) [Don87], Dong's method (DM_3^2) [Don82],method (LM_3) [LLC09] respectively. The test functions and root ξ_m correct up to 35 decimal places are displayed in Table 2.1. All computations have been performed using the programming package $Mathematica$ 9 with multi-precision arithmetic.

Table 2.3 and Table 2.6 show the comparison of different iterative methods of order three with respect to the number of iterations respectively. Here, the stopping criterion is described as the distance between two consecutive approximations for the required root is less than the precision of 10^{-34}. In the implementation of the iterative methods, the good choice of initial guess is very important because all the methods are locally convergent. A badly chosen initial guess produces a bad predictor and consequently, destroys the rapid convergence. Thus, one has to choose the initial guess enough close to the required roots.

In order to verify theoretical order of convergence, Table 2.4 and Table 2.7 displayed the values of computational order of convergence ($\hat{\rho}$) calculated by using formula (1.2.10), taking into consideration the last three consecutive approximations or the best three consecutive approximations in the iterative process.

Table 2.2 and Table 2.5, display absolute values of the function ($|f(x_n)|$) based on the same total number of functional evaluations ($TNFE = 12$) required by each method. Also, it can be observed from Table 2.2 and Table 2.5 that in majority of the problems tested here, the proposed methods are equally efficient to other existing methods when the accuracy is tested in multi-precision digits.

Table 2.1: (Test problems)

$f(x)$	m	ξ_m
$f_1(x) = \left(\sin(x) - \frac{1}{\sqrt{2}}\right)^2 (x+1)$	2	0.785398163397448309615660845819875721
$f_2(x) = x^2 \sin 4x$	3	0.000000000000000000000000000000000000
$f_3(x) = (\cos x - x)^2$	2	0.739085133215160641655312087673873404
$f_4(x) = \left(5\tan^{-1} x - 4x\right)^8$	8	0.949134611288828951372581521479848875
$f_5(x) = (e^{-x} + \sin x)^3$	3	3.183063011933363591939186995636394 6

Table 2.2: (Comparison of different third-order one-point methods with the same number of total functional evaluations (TNFE=12))

$f(x)$	x_0	SH_3	HM_3	OSM_3	KM_3^1	KM_3^2	OM_3^1 ($a=1$)	OM_3^2 ($a=1/2$)
1.	0.2	1.5e−83	1.1e−71	*D*	1.7e−87	2.0e−79	4.4e−110	7.2e−75
	0.5	6.6e−127	4.4e−123	*D*	3.6e−130	9.5e−132	1.1e−129	2.0e−125
	1.0	4.3e−150	1.9e−134	3.5e−141	5.6e−154	3.0e−140	1.8e−285	1.5e−137
	1.2	3.1e−125	7.4e−79	2.1e−92	1.1e−134	4.8e−85	5.2e−91	3.0e−89
2.	−0.3	1.0e−146	1.6e−115	1.6e−127	8.6e−148	3.6e−117	2.8e−186	1.4e−113
	−0.2	4.6e−180	5.2e−168	1.6e−173	6.0e−181	1.2e−169	2.1e−200	8.6e−167
	0.2	4.6e−180	5.2e−168	1.6e−173	6.0e−181	1.2e−169	2.1e−200	8.6e−167
	0.3	1.0e−146	1.6e−115	1.6e−127	8.6e−148	3.6e−117	2.8e−186	1.4e−113
3.	0.2	2.2e−186	5.8e−111	*D*	1.4e−165	3.8e−145	1.7e−146	3.0e−120
	0.5	3.5e−203	1.5e−172	*D*	1.0e−238	6.1e−201	9.2e−236	2.3e−177
	1.0	1.1e−184	5.4e−173	3.0e−178	1.3e−202	4.3e−200	6.2e−234	6.7e−175
	1.5	1.9e−108	1.3e−102	3.7e−105	1.2e−124	1.0e−134	1.0e−135	1.5e−103
4.	0.7	9.9e−447	9.0e−269	2.6e−52	9.4e−447	2.7e−269	3.0e−297	8.2e−143
	1.1	2.1e−686	9.9e−551	2.2e−601	3.6e−687	3.1e−551	9.1e−739	8.3e−510
	1.3	8.6e−468	1.2e−366	3.3e−405	1.3e−468	3.0e−367	1.4e−540	3.6e−335
5.	2.8	9.3e−204	2.3e−199	*D*	4.3e−209	2.0e−210	1.6e−209	7.5e−199
	3.5	7.9e−223	2.1e−214	2.4e−3	3.7e−228	1.0e−224	1.7e−235	1.8e−213
	4.2	2.4e−142	5.4e−69	1.7e−24	1.8e−132	4.1e−84	8.4e−74	3.1e−72

D: stands for divergent.

Table 2.3: (Comparison of different third-order one-point methods with respect to number of iterations)

$f(x)$	x_0	SH_3	HM_3	OSM_3	KM_3^1	KM_3^2	OM_3^1 $(a=1)$	OM_3^2 $(a=1/2)$
1.	0.2	5	5	D	5	5	5	5
	0.5	5	5	D	5	5	5	5
	1.0	5	5	5	5	5	4	5
	1.2	5	5	5	5	5	5	5
2.	-0.3	5	5	5	5	5	5	5
	-0.2	5	5	5	5	5	5	5
	0.2	5	5	5	5	5	5	5
	0.3	5	5	5	5	5	5	5
3.	0.2	5	5	D	5	5	5	5
	0.5	4	5	D	4	4	4	5
	1.0	5	5	5	4	5	4	5
	1.5	5	5	5	5	5	5	5
4.	0.7	5	5	7	5	5	5	5
	1.1	5	5	5	5	5	5	5
	1.7	5	5	5	5	5	5	5
5.	2.8	5	6	D	5	6	6	6
	3.5	5	5	8	5	5	5	5
	4.2	5	6	7	5	5	5	5

Table 2.4: (Computational order of convergence of different third-order one-point methods)

$f(x)$	x_0	SH_3	HM_3	OSM_3	KM_3^1	KM_3^2	OM_3^1 $(a=1)$	OM_3^2 $(a=1/2)$
1.	0.2	3.0000	3.0000	nd	3.0000	3.0000	3.0000	3.0000
	0.5	3.0000	3.0000	nd	3.0000	3.0000	3.0000	3.0000
	1.0	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
2.	-0.3	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	-0.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	0.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	0.3	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
3.	0.2	3.0000	3.0000	nd	3.0000	3.0000	3.0000	3.0000
	0.5	3.0000	3.0000	nd	3.0000	3.0000	3.0000	3.0000
	1.0	3.0000	3.0000	3.0020	3.0000	3.0000	3.0001	3.0000
	1.5	3.0000	3.0000	3.0000	3.0000	3.0000	2.9999	3.0000
4.	0.7	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.1	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.3	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
5.	2.8	3.0000	3.0000	nd	3.0000	3.0000	3.0000	3.0000
	3.5	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	4.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000

nd: not defined in the case of divergent.

Table 2.5: (Comparison of different third-order multi-point methods with the same number of total functional evaluations (TNFE=12))

$f1(x)$	x_0	DM_3^1	NM_3	OM_3^3	OM_3^4	DM_3^2	LM_3	OM_3^5	OM_3^6
				$(\theta = 1)$	$(\theta = 1/4)$			$(\theta = 1)$	$(\theta = 1/4)$
1.	0.2	7.8e−97	9.4e−152	1.2e−108	1.8e−106	9.5e−119	2.8e−57	2.6e−148	7.1e−107
	0.5	4.0e−148	1.7e−188	4.6e−150	6.8e−138	3.9e−295	1.2e−107	1.2e−305	1.2e−144
	1.0	8.9e−157	1.3e−199	1.1e−175	2.9e−196	4.6e−178	5.7e−121	2.1e−227	9.0e−188
	1.2	1.2e−104	2.0e−88	5.9e−149	6.4e−108	2.5e−108	1.3e−70	6.0e−129	3.0e−123
2.	−0.3	1.9e−139	3.3e−176	1.2e−147	4.7e−219	9.5e−224	8.9e−101	5.3e−188	2.6e−185
	−0.2	4.6e−190	2.5e−301	7.0e−193	2.0e−194	2.8e−252	1.7e−150	4.2e−227	7.4e−193
	0.2	4.6e−190	2.5e−301	7.0e−193	2.0e−194	2.8e−252	1.7e−150	4.2e−227	7.4e−193
	0.3	1.9e−139	3.3e−176	1.2e−147	4.7e−219	9.5e−224	8.9e−101	5.3e−188	2.6e−185
3.	0.2	5.0e−129	3.7e−133	2.1e−188	3.7e−144	1.2e−121	3.0e−110	8.2e−165	4.2e−157
	0.5	7.7e−190	7.7e−238	4.0e−225	5.2e−246	5.6e−193	2.5e−165	4.5e−281	1.8e−252
	1.0	6.6e−191	7.9e−244	4.1e−205	7.8e−214	7.0e−218	4.1e−160	1.4e−253	1.2e−210
	1.5	6.9e−122	9.6e−168	2.3e−129	4.3e−127	2.6e−216	4.2e−88	4.7e−187	1.1e−127
4.	0.7	4.0e−287	1.2e−240	1.9e−184	8.4e−424	8.2e−385	3.3e−285	6.4e−155	3.3e−455
	1.1	1.1e−560	6.0e−634	6.5e−524	3.2e−634	7.6e−594	5.5e−551	1.8e−517	6.9e−586
	1.3	1.8e−375	1.6e−652	5.4e−347	1.6e−429	2.6e−402	2.9e−365	7.5e−343	1.2e−393
5.	2.8	1.8e−220	3.4e−281	2.1e−221	5.7e−211	2.8e−263	4.3e−181	2.5e−256	2.8e−212
	3.5	1.9e−235	6.9e−314	2.1e−237	1.7e−233	1.0e−286	6.2e−197	2.9e−269	2.4e−233
	4.2	1.6e−97	9.0e−163	1.1e−141	1.9e−83	6.7e−92	7.8e−71	3.0e−101	3.9e−91

Table 2.6: (Comparison of different third-order multi-point methods with respect to number of iterations)

$f1(x)$	x_0	DM_3^1	NM_3	OM_3^3 ($\theta = 1$)	OM_3^4 ($\theta = 1/4$)	DM_3^2	LM_3	OM_3^5 ($\theta = 1$)	OM_3^6 ($\theta = 1/4$)
1.	0.2	5	5	5	5	5	6	5	5
	0.5	5	5	5	5	5	5	4	5
	1.0	5	5	5	5	5	5	4	5
	1.2	5	5	5	5	5	6	5	5
2.	−0.3	5	5	5	5	5	5	5	5
	−0.2	5	4	5	5	5	5	5	5
	0.2	5	5	5	5	5	5	5	5
	0.3	5	5	5	5	5	5	5	5
3.	0.2	5	5	5	5	5	5	5	5
	0.5	5	4	4	4	5	5	4	4
	1.0	5	4	4	4	4	5	4	4
	1.5	5	5	5	5	4	5	5	5
4.	0.7	5	6	6	5	5	5	6	5
	1.1	5	5	5	5	5	5	5	5
	1.3	5	5	5	5	5	5	5	5
5.	2.8	5	5	5	5	5	5	5	5
	3.5	5	4	5	5	5	5	5	5
	4.2	6	5	5	6	6	6	5	6

Table 2.7: (Computational order of convergence of different third-order multi-point methods)

$f1(x)$	x_0	DM_3^1	NM_3	OM_3^3 ($\theta = 1$)	OM_3^4 ($\theta = 1/4$)	DM_3^2	LM_3	OM_3^5 ($\theta = 1$)	OM_3^6 ($\theta = 1/4$)
1.	0.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	0.5	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.0	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
2.	−0.3	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	−0.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	0.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	0.3	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
3.	0.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	0.5	3.0000	3.0000	3.0020	3.0000	3.0000	3.0000	3.0000	3.0000
	1.0	3.0000	3.0020	3.0000	3.0000	3.0001	3.0000	3.0003	3.0004
	1.5	3.0000	3.0000	3.0000	3.0000	3.0000	2.9999	3.0000	3.0000
4.	0.7	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.1	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	1.3	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
5.	2.8	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	3.5	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	4.2	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000

2.5 Attractor basins in the complex plane

Here, we investigate the comparison of the attained multiple root finders in the complex plane using basins of attraction. It is known that the corresponding fractal of an iterative root-finding method is a boundary set in the complex plane, which is characterized by the iterative method applied to a fixed polynomial $p(z) \in \mathbb{C}$, see e.g. [SNC11, NSC12]. The aim herein is to use basin of attraction as another way for comparing the iteration algorithms.

From the dynamical point of view, we consider a rectangle $D = [-3,\ 3] \times [-3,\ 3] \in \mathbb{C}$ with a 400×400 grid, and we assign a color to each point $z_0 \in D$ according to the multiple root at which the corresponding iterative method starting from z_0 converges, and we mark the point as black if the method does not converge. In this section, we consider the stopping criterion for convergence to be less than 10^{-4} wherein the maximum number of full cycles for each method is considered to be 200. In this way, we distinguish the attraction basins by their colors for different methods.

Here, we have compared the methods Dong's method (DM_3^1) [Don87], Dong's method (DM_3^2) [Don82], Neta's Method (NM_3) [Net08], method (LM_3) [LLC09] with our proposed methods namely, method (2.3.3) for $(\theta = 1)$ (OM_3^3), for $(\theta = 1/4)$ (OM_3^4) and method (2.3.13) for $(\theta = 1)$ (OM_3^5), for $(\theta = 1/4)$ (OM_3^6) respectively. The test functions and root ξ_m correct up to 35 decimal places are displayed in Table 2.1. All computations have been performed using the programming package *Mathematica* 9 with multi-precision arithmetic respectively for some complex polynomials having multiple zeros with known multiplicity.

For the first three test problem, we have taken the polynomial functions:

Test Problem 1. Let $p_1(z) = (z^4 - 1)^3$, having roots $\{-1.,\ 0. - 1.i,\ 0. + 1.i,\ 1.\}$ with multiplicity three. It is straight forward to see from the Fig. 2.1 and Fig. 2.2 that our methods (OM_3^3), (OM_4^3), (OM_5^3) and (OM_6^3) have almost same basins of attraction as compared to LM_3, NM_3, DM_3^1 and DM_3^2. Also, note that method NM_3 shows chaotic behaviour. Our methods (OM_3^3), (OM_3^4), (OM_5^3) and (OM_6^3)

contain lower number of divergent points in comparison to the methods NM_3, LM_3 and DM_3^1 respectively.

Test problem 2. Let $p_2(z) = (z^5+2z-1)^3$, having roots $\{ -0.945068-0.854518i, -0.945068+0.854518i, 0.486389, 0.701874-0.879697i, 0.701874+0.879697i\}$ of multiplicity three. It is straight forward to see from Fig. 2.3 and Fig. 2.4. that our methods (OM_3^3) and (OM_4^5) has less chaotic behaviour than DM_3^1 and DM_3^2.

Test Problem 3. Let $p(z) = (z^4+z)^2$, having roots $\{-1., 0., 0.5-0.866025i, 0.5+0.866025i\}$ of multiplicity two. It is straight forward to see from Fig. 2.5 and Fig. 2.6. that our methods (OM_3^3) and (OM_3^5) are equally competent with DM_3^1 and has no divergent point as compared to NM_3.

For the last test problem, we have taken a non-polynomial function:

Test Problem 4. Let $p(z) = (z^6 + 1/z)^4$, having roots $\{ -0.62349+0.781831i, -0.62349-0.781831i, -1, 0.900969+0.433884i, 0.900969-0.433884i, 0.222521+0.974928i, 0.222521-0.974928i\}$ of multiplicity four. It is straight forward to see from Fig. 2.7 and Fig. 2.8 that our methods (OM_3^3) and (OM_3^5) are equally competent with DM_3^1 and DM_3^2. Further, our methods (OM_3^4) and (OM_3^6) contain lower number of divergent points in comparison to the methods NM_3 and LM_3.

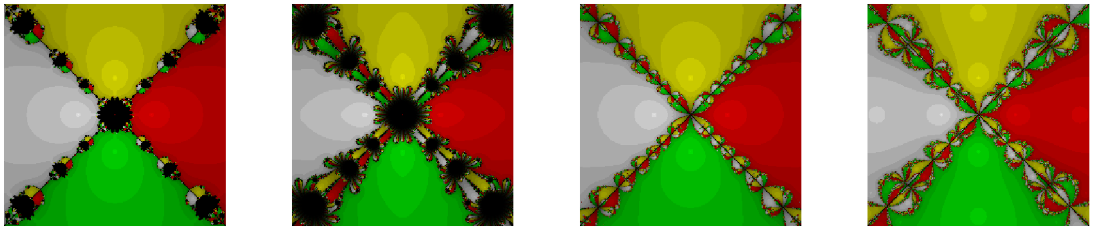

Figure 2.1: The basins of attraction for DM_3^1, NM_3, OM_3^3 and OM_3^4 respectively in test problem 1.

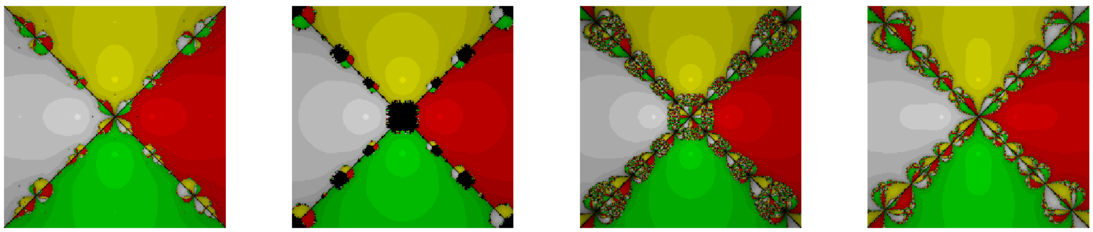

Figure 2.2: The basins of attraction for DM_3^2, LM_3, OM_3^5 and OM_3^6 respectively in test problem 1.

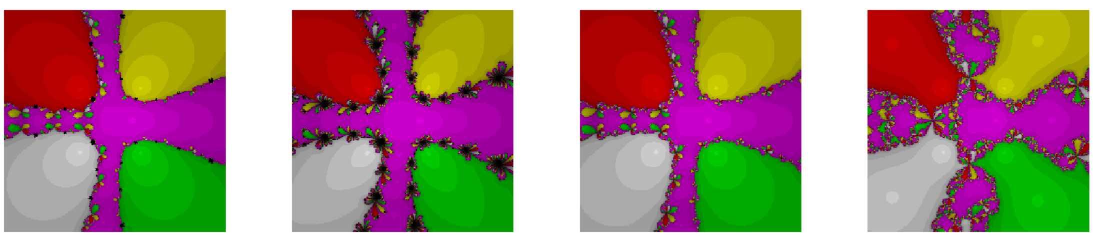

Figure 2.3: The basins of attraction for DM_3^1, NM_3, OM_3^3 and OM_3^4 respectively in test problem 2.

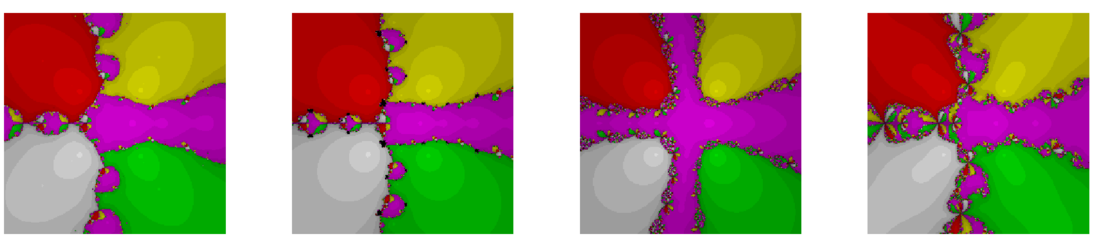

Figure 2.4: The basins of attraction for DM_3^2, LM_3, OM_3^5 and OM_3^6 respectively in test problem 2.

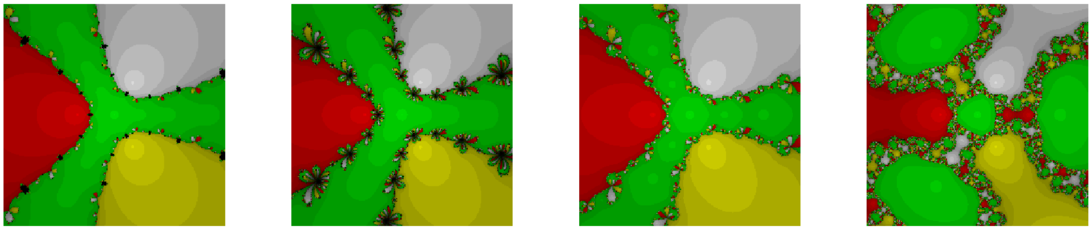

Figure 2.5: The basins of attraction for DM_3^1, NM_3, OM_3^3 and OM_3^4 respectively in test problem 3.

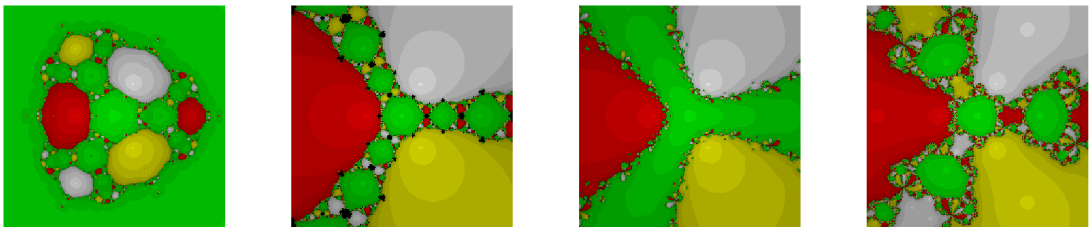

Figure 2.6: The basins of attraction for DM_3^2, LM_3, OM_3^5 and OM_3^6 respectively in test problem 3.

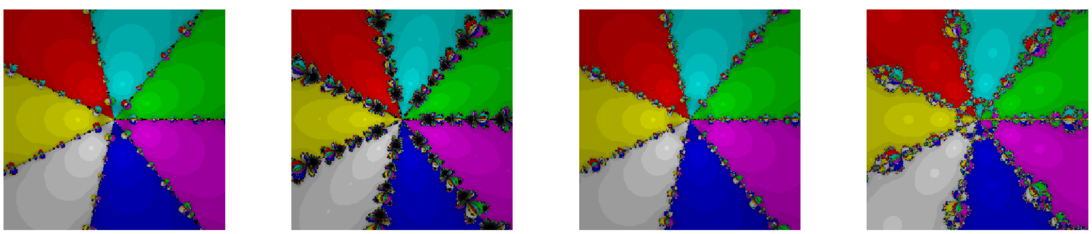

Figure 2.7: The basins of attraction for DM_3^1, NM_3, OM_3^3 and OM_3^4 respectively in test problem 4.

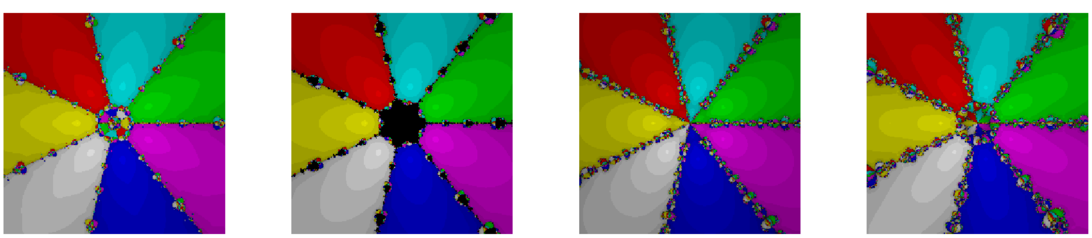

Figure 2.8: The basins of attraction for DM_3^2, LM_3, OM_3^5 and OM_3^6 respectively in test problem 4.

2.6 Concluding remarks

In this chapter, we have proposed a new simple and elegant root-finding third-order family of Chebyshev type method for obtaining multiple roots of nonlinear equations. Chebyshev method is obtained as a special case of our proposed scheme. Furthermore, we have also proposed many new third-order multi-point families of iterative methods which are free from second-order derivative. These methods require either two function and one derivative of first-order function and two derivative of first-order per iteration. A sufficiently close initial guess is required for the necessary convergence of these methods as they are the variants of Newton's method. Finally, from the numerical experiments it is observed that our proposed methods are efficient and perform equally competent to the existing methods available in the literature. Based on the Figures 2.1-2.8, we conclude that larger basins of attraction belong to our methods namely $OM_3^3$and OM_3^5. Although the other methods are slow and has darker basins while some of the methods are too sensitive upon the choice of initial guess.

Chapter 3

Two optimal general classes of Jarratt's method

3.1 Introduction

The main goal and motivation of this chapter is to present a simple and elegant technique to achieve as high as possible convergence order consuming as small as possible functional evaluations. Our proposed iterative methods are compared in their efficiency and performance to various other multi-point methods, and it is observed that our proposed methods are efficient and equally competent as compared to existing methods available in the literature.

Multi-point iterative methods are of great practical importance since they overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. Further, they are also capable to generate root approximations of very high accuracy. In the case of these multi-point methods, Kung and Traub [KT74] conjectured that the order of convergence of any multi-point method without memory, consuming n function evaluations per iteration, can not exceed the bound 2^{n-1} (called optimal order). Thus, the optimal order for a method with three functional evaluations per step would be four.

In the recent years, some fourth-order optimal modifications of Newton's method for multiple roots have been proposed and analyzed by Li et al. [LCN10], Sharma and Sharma [SS10b], Zhou et al. [ZCS11], Soleymani et al. [SBL13] and the references cited therein. All the mentioned methods requiring one-function and two first order-derivative evaluations per iteration. There are, however, not yet so many fourth or higher-order methods known that can handle the case of multiple roots.

The research of finding iterative methods with optimal fourth-order convergence, not requiring the computation of second-order derivative for multiple roots is very important and interesting task from the practical point of view. The presented approach in this chapter is based on the arithmetic mean of well known methods namely Schröder's method [Sch70], Halley's method [Tra64] and Chebyshev's method [Tra64] with five disposable parameters. First optimal general class of Jarratt's method is developed by taken arithmetic mean of Schröder's method [Sch70] and Halley's method [Tra64]. Second class is developed by taken arithmetic mean of Schröder's method [Sch70] and Chebyshev's method [Tra64]. All the proposed methods considered here are found to be more effective and comparable to the existing classical and recently proposed methods available in literature.

3.2 First optimal general class of Jarratt's method

The well-known Halley's method [Tra64] for simple zero and Schröder's method [Sch70] for simple as well as multiple zero are given by

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2\{f'(x_n)\}^2 - f(x_n)f''(x_n)}, \tag{3.2.1}$$

and

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{\{f'(x_n)\}^2 - f(x_n)f''(x_n)}, \tag{3.2.2}$$

respectively.

We now intend to develop new optimal families of Jarratt's method [Jar66]. For this, we take the arithmetic mean of (3.2.1) and (3.2.2) to get

$$x_{n+1} = x_n - \frac{1}{2}\left[\frac{2f(x_n)f'(x_n)}{2\{f'(x_n)\}^2 - f(x_n)f''(x_n)} + \frac{f(x_n)f'(x_n)}{\{f'(x_n)\}^2 - f(x_n)f''(x_n)}\right]. \tag{3.2.3}$$

Now consider a Newton-type iterative method

$$y_n = x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)}, \tag{3.2.4}$$

where $m \geq 1$ is the multiplicity of the multiple root $x = \xi_m$.
Now expanding the function $f'(y_n) = f'\left(x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)}\right)$ about the point $x = x_n$ by Taylor's series expansion, we have

$$f'\left(x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)}\right) \cong f'(x_n) - \frac{2m}{m+2}\frac{f(x_n)f''(x_n)}{f'(x_n)},$$

therefore, we obtain

$$f''(x_n) \cong \frac{(m+2)f'(x_n)[f'(x_n) - f'(y_n)]}{2mf(x_n)}. \tag{3.2.5}$$

Using this approximate value of $f''(x_n)$ in the method (3.2.3) and after some simplification, we get

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \left[\dfrac{2mf(x_n)}{(3m-2)f'(x_n) + (m+2)f'(y_n)} + \dfrac{mf(x_n)}{(m-2)f'(x_n) + (m+2)f'(y_n)}\right]. \end{cases} \tag{3.2.6}$$

This method has quadratic convergence and satisfies the following error equation

$$e_{n+1} = -\frac{c_2}{2}e_n^2 + \frac{1}{4}(6c_2^2 + (9a - 10)c_3)e_n^3 + O(e_n^4). \tag{3.2.7}$$

But according to the Kung-Traub conjecture [KT74], the above method (3.2.6) is not an optimal method because it has second-order convergence and requires three evaluations of function per full iteration. Therefore, to build our optimal family of Jarratt's method, we shall take five free disposable parameters. Therefore, we consider

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \left[\dfrac{2ma_1f(x_n)}{(3m-2)f'(x_n) + (m+2)a_2f'(y_n)} + \dfrac{ma_3f(x_n)}{(m-2)a_4f'(x_n) + (m+2)a_5f'(y_n)}\right], \end{cases} \tag{3.2.8}$$

where a_1, a_2, a_3, a_4, a_5 are disposable parameters such that the order of convergence reaches at the optimal level four without using any more functional evaluations. Theorem 2.1 indicates that under what choices on the disposable parameters in (3.2.8), the order of convergence will reach at the optimal level four.

3.2.1 Convergence analysis

Theorem 3.2.1 *Let $f : D \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function defined on an open interval D, enclosing a multiple zero of $f(x)$, say $x = \xi_m$ with multiplicity $m \geq 1$. Then the family of iterative methods defined by* (3.2.8) *has fourth-order convergence when*

$$\begin{cases} a_1 = -\dfrac{(3m-2)(m^2(m-2)a_4 + u(m^3+4m^2-8)a_5)^3}{2(m^3(m-2)a_4 + a_6a_5((m-2)^2m^3a_4^2 + 2m^3u(m^2-4)a_4a_5 + u^2(m+2)^2a_7a_5^2)}, \\ a_2 = -\dfrac{m^2\left(\frac{m}{m+2}\right)^{1-m}(3m-2)((m-2)a_4 + (m+2)ua_5)}{m^3(m-2)a_4 + a_6a_5}, \\ a_3 = -\dfrac{(m-2)(m(m-2)a_4 + u(m+2)^2a_5)^3}{2((m-2)^2m^3a_4^2 + 2m^3u(m^2-4)a_4a_5 + u^2(m+2)^2a_7a_5^2)}, \end{cases} \tag{3.2.9}$$

where $u = \left(\frac{m}{m+2}\right)^m$, $a_6 = u(m^4+2m^3-4m^2+16)$, $a_7 = (m^3-4m+8)$ and a_4 and a_5 are two free disposable parameters. The family (3.2.8) satisfies the following error equation

$$e_{n+1} = \left(\frac{\alpha_1}{\alpha_2}c_1^3 - \frac{1}{m}c_1c_2 + \frac{m}{(m+2)^2}c_3\right)e_n^4 + O(e_n^5), \tag{3.2.10}$$

where

$\alpha_1 = (u^2(m+2)^2(m^6+6m^5+10m^4-2m^3-24m^2+8m-32)a_5^2 + 2u(m^8+4m^7-18m^5-24m^4+24m^3+8m^2-64m+96)a_4a_5 + m^3(m-2)^2(m^3+2m^2+2m-2)a_4^2)$,

$\alpha_2 = 3m^4(m+2)^2(u(m+2)^2a_5 + m(m-2)a_4)(m^2(m-2)a_4 + u(m^3+4m^2-8))$ and

$c_k = \dfrac{m!}{m+k!}\dfrac{f^{(m+k)}(r_m)}{f^{(m)}(r_m)}$, $k = 1, 2, 3, \ldots$

Proof Let $x = \xi_m$ be a multiple zero of $f(x)$. Expanding $f(x_n)$ and $f'(x_n)$ about $x = \xi_m$ by the Taylor's series expansion (with the help of computer algebra software Mathematica), we have

$$f(x_n) = \frac{f^{(m)}(\xi_m)}{m!}e_n^m\left(1 + c_1e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4\right) + O(e_n^5), \tag{3.2.11}$$

and

$$f'(x_n) = \frac{f^{(m-1)}(\xi_m)}{(m-1)!}e_n^{m-1}\left(1 + \frac{m+1}{m}c_1e_n + \frac{m+2}{m}c_2e_n^2 + \frac{m+3}{m}c_3e_n^3 + \frac{(m+4)}{m}c_4e_n^4\right) + O(e_n^5), \tag{3.2.12}$$

respectively.

From equations (3.2.11) and (3.2.12), we have

$$\frac{f(x_n)}{f'(x_n)} = \frac{1}{m}e_n - \frac{c_1}{m^2}e_n^2 + \left(\frac{(m+1)c_1^2 - 2mc_2}{m^3}\right)e_n^3 + O(e_n^4). \tag{3.2.13}$$

Furthermore, we have

$$\frac{2ma_1f(x_n)}{(3m-2)f'(x_n) + (m+2)a_2f'(y_n)} = \left(\frac{2a_1}{3m-2+a_2m^{m-1}(m+2)^{2-m}}\right)e_n - \left(\frac{2a_1(m+2)^m((3m-2)m^2(m+2)^m + a_2m^m(m^3+4m^2-8))c_1}{m[a_2m^m(m+2)^2 + m(m+2)^m(3m-2)]^2}\right)e_n^2 + Ae_n^3 + O(e_n^4), \tag{3.2.14}$$

where

$$A = \frac{2a_1}{m^2[u(m+2)^2a_2 + m(3m-2)]^3}[(m^3(3m-2)^2(m+1) + 2u(3m^6 + 13m^5 + 8m^4 - 18m^3 - 20m^2 - 8m + 16)a_2 + u^2(m+2)^2(m^4 + 5m^3 + 4m^2 - 8m - 16)a_2^2)c_1^2 - 2m((3m-2)^2m^3 + 2mu(3m^4 + 10m^3 - 2m^2 - 16m + 8)a_2 + u^2(m+2)^3 (m^2 + 2m - 4)a_2^2)c_2].$$

Similarly, we can also get

$$\frac{ma_3f(x_n)}{(m-2)a_4f'(x_n) + (m+2)a_5f'(y_n)} = \left(\frac{ma_3}{m(m-2)a_4 + (m+2)^2ua_5}\right)e_n - \left(\frac{a_3(m^2(m-2)a_4 + u(m^3+4m^2-8)a_5)c_1}{m(m(m-2)a_4 + (m+2)^2ua_5)^2}\right)e_n^2 + Be_n^3 + O(e_n^4), \tag{3.2.15}$$

where

$$B = \frac{a_3}{m^2(m(m-2)a_4 + (m+2)^2ua_5)^3}[(m-2)^2m^3a_4^2((m+1)c_1^2 - 2mc_2) + 2u(m^2-4) a_4a_5((m^4 + 3m^3 - 2m - 4)c_1^2 - 2m^2(m^2 + 2m - 2)c_2) + u^2(m+2)^2a_5^2 ((m^4 + 5m^3 + 4m^2 - 8m - 16)c_1^2 - 2m(m^3 + 4m^2 - 8m)c_2)].$$

Using equations (3.2.14) and (3.2.15) in scheme (3.2.8), we get the following error equation

$$\begin{aligned}
e_{n+1} &= e_n - \left[\frac{2ma_1 f(x_n)}{(3m-2)f'(x_n)+(m+2)a_2 f'(y_n)} + \frac{ma_3 f(x_n)}{(m-2)a_4 f'(x_n)+(m+2)a_5 f'(y_n)}\right],\\
&= \left(1 - \frac{2a_1}{3m-2+a_2 m^{m-1}(m+2)^{2-m}} - \frac{ma_3}{m(m-2)a_4+(m+2)^2 u a_5}\right) e_n\\
&+ \left(\frac{2a_1(m+2)^m((3m-2)m^2(m+2)^m + a_2 m^m(m^3+4m^2-8))c_1}{m[a_2 m^m (m+2)^2 + m(m+2)^m(3m-2)]^2}\right.\\
&+ \left.\frac{a_3(m^2(m-2)a_4 + u(m^3+4m^2-8)a_5)c_1}{m(m(m-2)a_4+(m+2)^2 u a_5)^2}\right) c_1 e_n^2\\
&- (A+B)e_n^3 + O(e_n^4).
\end{aligned} \tag{3.2.16}$$

For obtaining an iterative method of order four, the coefficients of e_n, e_n^2 and e_n^3 in the error equation (3.2.16) must be zero simultaneously. After simplifying the equation (3.2.16), we have the following values of a_1, a_2 and a_3 involving two free disposable parameters a_4 and a_5

$$\begin{cases}
a_1 = -\dfrac{(3m-2)(m^2(m-2)a_4 + u(m^3+4m^2-8)a_5)^3}{2(m^3(m-2)a_4+a_6 a_5)((m-2)^2 m^3 a_4^2 + 2m^3 u(m^2-4)a_4 a_5 + u^2(m+2)^2 a_7 a_5^2)},\\
a_2 = -\dfrac{m^2\left(\frac{m}{m+2}\right)^{1-m}(3m-2)((m-2)a_4+(m+2)ua_5)}{m^3(m-2)a_4 + a_6 a_5},\\
a_3 = -\dfrac{(m-2)(m(m-2)a_4+u(m+2)^2 a_5)^3}{2((m-2)^2 m^3 a_4^2 + 2m^3 u(m^2-4)a_4 a_5 + u^2(m+2)^2(m^3-4m+8)a_5^2)},
\end{cases} \tag{3.2.17}$$

where $u = \left(\frac{m}{m+2}\right)^m$ and $a_7 = (m^3-4m+8)$.

The family (3.2.8) satisfies the following error equation

$$e_{n+1} = \left(\frac{\alpha_1}{\alpha_2}c_1^3 - \frac{1}{m}c_1 c_2 + \frac{m}{(m+2)^2}c_3\right) e_n^4 + O(e_n^5), \tag{3.2.18}$$

where

$\alpha_1 = (u^2(m+2)^2(m^6+6m^5+10m^4-2m^3-24m^2+8m-32)a_5^2 + 2u(m^8+4m^7-18m^5 - 24m^4+24m^3+8m^2-64m+96)a_4 a_5 + m^3(m-2)^2(m^3+2m^2+2m-2)a_4^2)$,

$\alpha_2 = 3m^4(m+2)^2(u(m+2)^2 a_5 + m(m-2)a_4)(m^2(m-2)a_4 + u(m^3+4m^2-8))$.

This reveals that the general two-step class of methods (3.2.8) reaches the optimal order of convergence four by using only three functional evaluations per full iteration. This completes the proof of the Theorem 2.1. □

3.2.2 Some special cases of first general class

Finally, we get first new optimal general class of methods from the scheme (3.2.8) by using the values of disposable parameters defined in (3.2.9), which is given by

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)\alpha_3}{2((m-2)a_4f'(x_n)+(m+2)a_5f'(y_n))\alpha_4}, \end{cases} \tag{3.2.19}$$

where

$\alpha_3 = [m^3(m-2)^2((m-2)f'(y_n) - uf'(x_n))a_4^2 - 2u(m^2-4)(u(m^4+2m^3-2m^2-4m-8)f'(x_n) - m^2(m^2-6)f'(y_n))a_4a_5 - u^2(m+2)^2(u(m-2)(m+2)^2f'(x_n) - (m^4+2m^3-8m^2-16m+16)f'(y_n))a_5^2]$, and $\alpha_4 = (m^3(2-m)(f'(y_n) - uf'(x_n))a_4 + u(m+2)((m^3-4m+8)f'(x_n) - m^3f'(y_n))a_5)$.

It is straight forward to see from the above mentioned general class that for different specific values of a_4 and a_5, the following various optimal families of methods can be derived by fixing one of the free disposable parameters. Some of the important families of methods are given below as:

(i) For $a_5 = -1$, family (3.2.19) reads as

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)\alpha_5}{2((m+2)f'(y_n)-(m-2)a_4f'(x_n))\alpha_6}, \end{cases} \tag{3.2.20}$$

where

$\alpha_5 = 2u(m^2-4)(u(m^4+2m^3-2m^2-4m-8)f'(x_n) - m^2(m^2-6)f'(y_n))a_4 + m^3(m-2)^2((m-2)f'(y_n) - muf'(x_n))a_4^2) - u^2(m+2)^2(u(m-2)(m+2)^3f'(x_n) - (m^4+2m^3-8m^2-16m-16)f'(y_n))$, and $\alpha_6 = (u(m+2)(u(m^3-4m+8)f'(x_n) - m^3f'(y_n)) + m^3(m-2)(f'(y_n) - uf'(x_n))a_4^2)$.

This is a new optimal family of fourth-order methods.

Sub cases of above family (3.2.20)

(a) For $a_4 = 0$, family (3.2.20) reads as

$$\begin{cases} y_n = & x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = & x_n - \dfrac{muf(x_n)}{2f'(y_n)}\left[\dfrac{u(m-2)(m+2)^3 f'(x_n) - (m^4+2m^3-8m^2-16m+16)f'(y_n)}{m^3 f'(y_n) - u(m^3-4m+8)f'(x_n)}\right]. \end{cases} \tag{3.2.21}$$

This is a well-known fourth-order method proposed by Li et al. [LCN10].

(b) For $a_4 = -\frac{u(m+2)^2}{m^2}$, family (3.2.20) reads as

$$\begin{cases} y_n = & x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = & x_n - \dfrac{muf(x_n)[m(m^5-9m^3+20m-8)f'(y_n) - u(m^6+2m^5-5m^4-10m^3+8m+16)f'(x_n)]}{\big(m^2 f'(y_n) + u(m^2-4)f'(x_n)\big)\big(u(m^3-4m+4)f'(x_n) - m(m^2-2)f'(y_n)\big)}. \end{cases} \tag{3.2.22}$$

This is a new optimal method of fourth-order.

(c) For $a_4 = -u(m+2)^2$, family (3.2.20) reads as

$$\begin{cases} y_n = & x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = & x_n - \dfrac{f(x_n)mu\left[f'(y_n)h_1 + f'(x_n)u(48+16m-20m^3+5m^4+4m^5-6m^6+m^8)\right]}{2\left(f'(y_n)+f'(x_n)u\left(-4+m^2\right)\right)\left(-f'(y_n)m^3\left(-3+m^2\right)+f'(x_n)u\left(8-4m-3m^3+m^5\right)\right)}, \end{cases} \tag{3.2.23}$$

where $h_1 = (-16+16m-40m^2+30m^3+3m^4-16m^5+6m^6+2m^7-m^8)$.

This is a new optimal method of fourth-order.

(d) For $a_4 = -\frac{2\sqrt{\left(\frac{m}{2+m}\right)^{2m}(2+m)^2(16+16m+12m^2-4m^3-3m^4)} - \left(\frac{m}{2+m}\right)^m\left(-16-16m-8m^2+2m^3+4m^4+m^5\right)}{(-2+m)m^4}$,

$$\begin{cases} y_n = & x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = & x_n - \dfrac{16 f'(y_n) f(x_n)}{\{f'(x_n)\}^2(-2+m)m^2\left(\frac{m}{2+m}\right)^{-3+m} + \{f'(y_n)\}^2 m^3\left(\frac{m}{2+m}\right)^{-m} + 2h_1 f'(x_n)f'(y_n)}, \end{cases} \tag{3.2.24}$$

where $h_1 = \left(4+\frac{8}{m}+2m-2m^2-m^3\right)$.

This is a well-known fourth-order method proposed by Soleymani et al. [SBL13].

(ii) For $a_5 = 0$ and $a_4 \neq 0$, general class (3.2.19) reads as

$$\begin{cases} y_n = & x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = & x_n - \dfrac{mf(x_n)}{2f'(x_n)}\left[\dfrac{(m-2)f'(y_n) - muf'(x_n)}{uf'(x_n) - f'(y_n)}\right]. \end{cases} \tag{3.2.25}$$

This is a well-known fourth-order method proposed by Li et al. [SXL09].

Remark 1. The family (3.2.19) can produce several new optimal Jarratt-type methods for multiple zeros by choosing different values of the parameter a_5.

Remark 2. Li et al. [LCN10, SXL09] methods are special cases of our proposed general class of methods (3.2.19).

Remark 3. The general class of methods (3.2.19) can produce several new optimal families of Jarratt's method for multiple roots by fixing one of the free disposable parameters namely either a_4 or a_5 respectively.

3.3 Second optimal general class of Jarratt's method

The well-known third-order Chebyshev's method [Tra64] for simple zero is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{\{f(x_n)\}^2 f''(x_n)}{2\{f'(x_n)\}^3}. \tag{3.3.1}$$

We now intend to develop another new optimal general class of Jarratt's method. For this, we take the arithmetic mean of (3.3.1) and (3.2.1) to get

$$x_{n+1} = x_n - \frac{1}{2}\left[\frac{f(x_n)}{f'(x_n)} - \frac{\{f(x_n)\}^2 f''(x_n)}{2\{f'(x_n)\}^3} + \frac{2f(x_n)f'(x_n)}{2\{f'(x_n)\}^2 - f(x_n)f''(x_n)}\right]. \tag{3.3.2}$$

Now using the approximate value of $f''(x_n)$ as mentioned in (3.2.5) in the above scheme (3.3.2) and after some simplification, we get

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{1}{2}\Bigg[\dfrac{((5m+2)f'(x_n) - (m+2)f'(y_n))f(x_n)}{4m\{f'(x_n)\}^2} + \\ \qquad \dfrac{2f(x_n)}{(m-2)f'(x_n) + (m+2)f'(y_n)}\Bigg]. \end{cases} \tag{3.3.3}$$

This method has linear convergence and satisfies the following error equation

$$e_{n+1} = \left(1 - \frac{m^2}{m^2(m-2) + m(m+2)^2 u} - \frac{m(5m+2) - u(m+2)^2}{4m^3}\right) e_n + O(e_n^2).$$

But again according to the Kung-Traub conjecture [KT74], the above method (3.3.3) is not an optimal method because it has linear-order convergence and requires three evaluations of function per full iteration. Therefore, to build our optimal family of Jarratt's method, we again take five free disposable parameters. Therefore, we consider

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{1}{2}\left[\dfrac{f(x_n)((5m+2)b_1 f'(x_n) - (m+2)b_2 f'(y_n))}{4m\{f'(x_n)\}^2} + \dfrac{2b_3 f(x_n)}{(m-2)f'(x_n)b_4 + (m+2)f'(y_n)b_5}\right], \end{cases} \tag{3.3.4}$$

where b_1, b_2, b_3, b_4, b_5 are free disposable parameters such that the order of convergence reaches at the optimal level four without using any more functional evaluations. Theorem 4.1 indicates that under what choices on the disposable parameters in (3.3.4), the order of convergence will reach at the optimal level four.

3.3.1 Convergence analysis

Theorem 3.3.1 *Let $f : D \subseteq \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function defined on an open interval D, enclosing a multiple zero of $f(x)$, say $x = \xi_m$ with multiplicity $m \geq 1$. Then the family of iterative methods defined by* (3.3.4) *has fourth-order convergence when*

$$\begin{cases} b_1 = -\dfrac{m^2((m-2)^2 m^3 b_4^2 + 3u(m-2)m^2(m+2)^2 b_4 b_5 + 2u^2(m+2)^2(m^3+3m^2+2m-4)b_5^2)}{u^2(m+2)^2(5m+2)b_5^2}, \\ b_2 = -\dfrac{m^5((m-2)b_4 + u(m+2)b_5)}{u^2(m+2)^2 b_5}, \\ b_3 = \dfrac{((m^2-2m)b_4 + u(m+2)^2 b_5)^3}{8(u(m+2)b_5)^2}, \end{cases} \tag{3.3.5}$$

where $u = \left(\frac{m}{m+2}\right)^m$ and b_4 and b_5 ($b_5 \neq 0$) are two free disposable parameters. The family (3.3.4) *satisfies the following error equation*

$$e_{n+1} = \left(\frac{u(m+2)^2 b_5 \beta_1 + (m-2)b_4\beta_2}{3m^4(m+2)^2(m(m-2)b_4 + u(m+2)^2 b_5)}\right)e_n^4 + O(e_n^5), \tag{3.3.6}$$

where

$$\beta_1 = (m^5 + 6m^4 + 14m^3 + 14m^2 + 12m + 16)c_1^3 - 3m^3(m+2)^2 c_1 c_2 + 3m^5 c_3),$$

$$\beta_2 = (m+2)^2(m^4+2m^3+2m^2-2m+12)c_1^3 - 3m^4(m+2)^2 c_1 c_2 + 3m^6 c_3,$$

and $c_k = \dfrac{m!}{m+k!}\dfrac{f^{(m+k)}(r_m)}{f^{(m)}(r_m)}, \quad k = 1,\ 2,\ 3,\ \ldots$

Proof The proof of this theorem is similar to the proof of Theorem 2.1. Hence, it is omitted here.

3.3.2 Some special cases of second general class

Finally, we get second new optimal general two-step class of methods from the scheme (3.3.4) by using the values of disposable parameters defined in (3.3.5), which is given by

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)}{8u^2 f'(x_n)}\left[\dfrac{m^3((m-2)b_4 + u(m+2)b_5\{f'(y_n)\}^2 + A_1\{f'(x_n)\}^2 - B_1 f'(x_n)f'(y_n)}{(m-2)b_4\{f'(x_n)\}^2 + (m+2)b_5 f'(x_n)f'(y_n)}\right]. \end{cases} \tag{3.3.7}$$

where

$A_1 = u^2((m^4+4m^3-4m^2-8m-16)b_4 + u(m+2)^4 b_5)$ and $B_1 = 2u(m^2(m^2+m-6)b_4 + u(m^4+5m^3+8m^2-8)b_5)$.

It is straight forward to see from the above mentioned optimal general class of fourth-order methods that for different specific values of b_4 and b_5, the following various optimal families of methods can be derived by fixing one the free disposable parameters.

(i) For $b_5 = 1$, general class (3.3.7) reads as

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)}{8u^2 f'(x_n)}\left[\dfrac{m^3((m-2)b_4 + u(m+2))\{f'(y_n)\}^2 + A_2\{f'(x_n)\}^2 - B_2 f'(x_n)f'(y_n)}{(m-2)b_4\{f'(x_n)\}^2 + (m+2)f'(x_n)f'(y_n)}\right]. \end{cases} \tag{3.3.8}$$

where

$A_2 = u^2((m^4+4m^3-4m^2-8m-16)b_4 + u(m+2)^4)$ and $B_2 = 2u(m^2(m^2+m-6)b_4 + u(m^4+5m^3+8m^2-8))$.

This is a new optimal family of fourth-order methods.

Sub cases of family (3.3.8)

(i) For $b_4 = 0$, family (3.3.8) reads as

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)}{f'(x_n)}\left[\dfrac{m^3\{f'(y_n)\}^2 + u^2(m+2)^3\{f'(x_n)\}^2 - 2u(m^3+3m^2+2m-4)f'(x_n)f'(y_n)}{8uf'(x_n)f'(y_n)}\right]. \end{cases} \tag{3.3.9}$$

This is a well-known method proposed by Zhou et al. [ZCS11].

(ii) For $b_4 = -\frac{u(m+2)^4}{m^4+4m^3-4m^2-8m-16}$, the family (3.3.8) reads as

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)f'(y_n)}{2u\{f'(x_n)\}^2}\left[\dfrac{m^4f'(y_n) - u(m^4+2m^3+4m^2+8m+16)f'(x_n)}{u(m+2)^3f'(x_n) - (m^3+6m^2+8m+8)f'(y_n)}\right]. \end{cases} \tag{3.3.10}$$

This is a new fourth-order optimal method.

(iii) For $b_4 = -\frac{u(m^4+5m^3+8m^2-8)}{m^2(m^2+m-6)}$, the family (3.3.8) reads as

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)}{4uf'(x_n)}\left[\dfrac{m^3(m-2)\{f'(y_n)\}^2 - u^2(m^4+2m^3+4m^2+8m+16)\{f'(x_n)\}^2}{u(m^3+3m^2+2m-4)\{f'(x_n)\}^2 - m^2(m+3)f'(x_n)f'(y_n)}\right]. \end{cases} \tag{3.3.11}$$

This is again a new fourth-order optimal method.

(iv) For $b_4 = -\frac{u(m+2)}{m-2}$, the family (3.3.8) gives the well-known fourth-order method (3.2.25) proposed by Li et al. [SXL09].

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{mf(x_n)}{2f'(x_n)}\left[\dfrac{(m-2)f'(y_n) - muf'(x_n)}{uf'(x_n) - f'(y_n)}\right]. \end{cases} \tag{3.3.12}$$

This is a well-known fourth-order method proposed by li et. al .

It is straightforward to see that per step these iterative methods require three functional evaluations per step, viz. one evaluation of $f(x)$ and two of $f'(x)$. In order to obtain an assessment of the efficiency of our methods, we shall make use of the efficiency index defined by equation (1.2.8). For our proposed iteration schemes, we find $p = 4$ and $d = 3$, yielding $E = \sqrt[3]{4} \cong 1.587$, which is better than those of most third-order methods $E \cong 1.442$ and modified Newton's method $E \cong 1.414$.

Remark 1. Zhou et al. [ZCS11] method (11) is a special case of our proposed scheme (3.3.4).

Remark 2. Here, we should note that one can easily develop several new optimal families of Jarratt's method for multiple roots from the scheme (3.3.4) by fixing one of the free disposable parameters namely b_4 and b_5.

3.4 Numerical results

In this section, we shall check the effectiveness of the new optimal methods. We employ the present methods namely, method (3.2.22), (3.2.23), (3.3.10) and (3.3.11) denoted by OM_4^1, OM_4^2, OM_4^4 and OM_4^4respectively to solve the following nonlinear equations. We compare them with the method of Zhou et al. [ZCS11] namely method (11) (ZM_4), Sharma and Sharma method [SS10b] (SSM_4), Li et al. [LCN10] method (75) (LM_4) and Soleymani et al. [SBL13] method (18) denoted by SM_4 respectively. For better comparisons of our proposed methods, we have given two comparison tables in each example: one is corresponding to absolute error value of given nonlinear functions (with the same total number of functional evaluations =12) and other is with respect to number of iterations taken by each method to obtain the root correct up to 35 significant digits. All computations have been performed using the programming package *Mathematica* 9 with multiple precision arithmetic. We use $\epsilon = 10^{-35}$ as a tolerance error. The following stopping criteria are used for computer programs:

(i) $|x_{n+1} - x_n| < \epsilon$, (ii) $|f(x_{n+1})| < \epsilon$.

Example 3.4.1 Consider the following 5×5 matrix

$$A = \begin{bmatrix} 29 & 14 & 2 & 6 & -9 \\ -47 & -22 & -1 & -11 & 13 \\ 19 & 10 & 5 & 4 & -8 \\ -19 & -10 & -3 & -2 & 8 \\ 7 & 4 & 3 & 1 & -3 \end{bmatrix}.$$

The corresponding characteristic polynomial of this matrix is as follows:

$$f_1(x) = (x-2)^4(x+1). \tag{3.4.1}$$

It's characteristic equation has one multiple root at $x = 2.0000000000000000000000000000000000$ of multiplicity four.

$f(x)$	x_0	ZM_4	SSM_4	LM_4	SM_4	OM_4^1	OM_4^2	OM_4^3	OM_4^4
Comparison of different fourth-order optimal methods with the same total number of functional evaluations (TNFE=12)									
$f_1(x)$	1.0	1.2e−614	2.8e−615	1.6e−616	5.2e−613	3.0e−616	1.8e−616	4.7e−617	9.6e−617
	1.5	5.6e−1017	7.1e−1018	1.2e−1019	1.6e−1014	2.9e−1019	1.4e−1019	2.0e−1020	5.8e−1020
	2.5	1.1e−1150	1.3e−1151	1.6e−1153	9.2e−1148	4.2e−1153	1.9e−1153	2.3e−1154	7.5e−1154
	2.9	7.3e−931	9.2e−932	1.3e−933	6.2e−928	3.3e−933	1.5e−933	2.0e−934	6.1e−934
Comparison of different fourth-order optimal methods with respect to number of iterations									
$f_1(x)$	1.0	4	4	4	4	4	4	4	4
	1.5	4	4	4	4	4	4	4	4
	2.5	4	4	4	4	4	4	4	4
	2.9	4	4	4	4	4	4	4	4

Example 3.4.2 Consider the following 8×8 matrix

$$B = \begin{bmatrix} 5 & 8 & 0 & 2 & 6 & -6 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 18 & -1 & 1 & 13 & -9 & 0 & 0 \\ 3 & 6 & 0 & 4 & 6 & -6 & 0 & 0 \\ 4 & -14 & -2 & 0 & 11 & -6 & 0 & 0 \\ 6 & 18 & -2 & 1 & 13 & -8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -19 \end{bmatrix}.$$

The corresponding characteristic polynomial of this matrix is as follows:

$$f_2(x) = x(x-4)(x-3)(x-2)(x-1)^3(x+19). \tag{3.4.2}$$

It's characteristic equation has one multiple root at $x = 1.00000000000000000000$ 0000000000000 of multiplicity three.

$f(x)$	x_0	ZM_4	SSM_4	LM_4	SM_4	OM_4^1	OM_4^2	OM_4^3	OM_4^4
Comparison of different fourth-order optimal methods with the same total number of functional evaluations (TNFE=12)									
$f_2(x)$	0.5	8.9e−287	4.9e−287	2.6e−287	6.1e−287	2.5e−287	2.2e−287	1.2e−287	1.8e−287
	0.8	9.1e−875	2.8e−875	7.8e−876	4.4e−875	7.0e−876	5.2e−876	1.7e−876	3.3e−876
	1.2	2.4e−603	1.4e−604	8.2e−606	3.0e−604	6.4e−606	3.4e−606	2.8e−607	1.2e−606
	1.4	1.8e−260	1.1e−260	5.9e−261	1.3e−260	5.6e−261	4.9e−261	2.9e−261	3.9e−261
Comparison of different fourth-order optimal methods with respect to number of iterations									
$f_2(x)$	0.5	5	5	5	5	5	5	5	5
	0.8	4	4	4	4	4	4	4	4
	1.2	4	4	4	4	4	4	4	4
	1.4	5	5	5	5	5	5	5	5

Example 3.4.3 $f_3(x) = (5\tan^{-1} x - 4x)^8$.

This equation has an finite number of roots with multiplicity eight but our desired root is $x = 0.94913461128828951372581521479848875$.

$f(x)$	x_0	ZM_4	SSM_4	LM_4	SM_4	OM_4^1	OM_4^2	OM_4^3	OM_4^4
Comparison of different fourth-order optimal methods with the same total number of functional evaluations (TNFE=12)									
$f_3(x)$	0.7	1.6e−248	1.6e−248	1.8e−248	4.7e−252	1.8e−248	1.8e−248	1.9e−248	1.8e−248
	1.0	2.2e−2297	5.4e−2298	5.5e−2300	4.3e−2264	3.6e−2300	1.8e−2300	4.4e−2301	9.0e−2301
	1.2	2.0e−1136	6.1e−1137	1.2e−1138	5.1e−1105	7.4e−1139	4.1e−1139	1.2e−1139	2.2e−1139
Comparison of different fourth-order optimal methods with respect to number of iterations									
$f_3(x)$	0.7	6	6	6	6	6	6	6	6
	1.0	4	4	4	4	4	4	4	4
	1.2	4	4	4	4	4	4	4	4

Example 3.4.4 $f_4(x) = \big((x-1)^3 - 1\big)^{50}$.

This equation has an finite number of roots with multiplicity but our desired root is $x = 2.00000000000000000000000000000000$.

$f(x)$	x_0	ZM_4	SSM_4	LM_4	SM_4	OM_4^1	OM_4^2	OM_4^3	OM_4^4
Comparison of different fourth-order optimal methods with the same total number of functional evaluations (TNFE=12)									
$f_4(x)$	1.7	4.7e−2269	4.6e−2269	2.6e−2269	1.8e−2122	2.6e−2269	2.6e−2269	2.5e−2269	2.5e−2269
	2.5	2.4e−4261	2.2e−4261	2.6e−4262	1.8e−3645	2.5e−4262	2.4e−4262	2.2e−4262	2.2e−4262
Comparison of different fourth-order optimal methods with respect to number of iterations									
$f_4(x)$	1.7	5	5	5	5	5	5	5	5
	2.5	5	5	5	5	5	5	5	5

Example 3.4.5 $f_5(x) = (x^2 - e^x - 3x + 2)^3$.

This equation has an finite number of roots with multiplicity three but our desired root $x = 0.25753028543986076045536730493724178$.

$f(x)$	x_0	ZM_4	SSM_4	LM_4	SM_4	OM_4^1	OM_4^2	OM_4^3	OM_4^4
Comparison of different fourth-order optimal methods with the same total number of functional evaluations (TNFE=12)									
$f_5(x)$	−0.5	3.6e−662	6.8e−657	5.4e−649	8.1e−660	6.4e−651	1.1e−649	5.0e−645	9.0e−648
	1.0	6.0e−726	8.0e−726	1.2e−725	7.2e−726	1.1e−725	1.2e−725	1.6e−725	1.3e−725
Comparison of different fourth-order optimal methods with respect to number of iterations									
$f_5(x)$	−0.5	4	4	4	4	4	4	4	4
	1.0	4	4	4	4	4	4	4	4

3.5 Attractor basins in the complex plane

We here investigate the comparison of the attained multiple root finders in the complex plane using basins of attraction. It is known that the corresponding fractal of an iterative root-finding method is a boundary set in the complex plane, which is characterized by the iterative method applied to a fixed polynomial $p(z) \in \mathbb{C}$, see e.g. [SNC11, NSC12]. The aim herein is to use basin of attraction as another way for comparing the iteration algorithms.

From the dynamical point of view, we consider a rectangle $D = [-3,\ 3] \times [-3,\ 3] \in \mathbb{C}$ with a 400×400 grid, and we assign a color to each point $z_0 \in D$ according to the multiple root at which the corresponding iterative method starting from z_0 converges, and we mark the point as black if the method does not converge. In this section, we consider the stopping criterion for convergence to be less than 10^{-4} wherein the maximum number of full cycles for each method is considered to be 200. In this way, we distinguish the attraction basins by their colors for different methods.

We have compared here the method of Zhou et al. [ZCS11], namely method (11) (ZM_4), Sharma and Sharma method [SS10b] (SSM_4), Li et al. [LCN10] method (75) (LM_4) and Soleymani et al. [SBL13] method (18) denoted by (SM_4) respectively for some complex polynomials having multiple zeros with known multiplicity. For the first three test problem, we have taken the polynomial functions

Test Problem 1. Let $p_1(z) = (z^6 - 1)^2$, having roots $\{1,\ -0.5+0.866025i,\ -0.5-0.866025i,\ 0.5+0.866025i,\ 0.5-0.866025i,\ -1\}$ with multiplicity two. It is straight forward to see from the Fig. 3.1 and Fig. 3.2 that our methods (OM_4^1), (OM_4^2),(OM_4^3) and (OM_4^4) have almost same basins of attraction as compared to SSM_4, LM_4^1 and SM_4. Also, note that method ZM_4 shows chaotic behaviour.

Test problem 2. Let $p_2(z) = (z^3 + 2z - 1)^3$, having roots $\{-0.226699 + 1.46771i,\ -0.226699 - 1.46771i,\ 0.453398\}$ of multiplicity three. It is straight forward to see from Fig. 3.3 and Fig. 3.4 that our methods (OM_4^3) and (OM_4^4) contain lower number of divergent points in comparison to the methods ZM_4, LM_4^1 and SM_4 respectively. But, our methods (OM_4^3) and (OM_4^4) have almost same basins of attraction as compared to SSM_4.

Test Problem 3. Let $p(z) = (z^4 + z)^3$, having roots $\{-1,\ 0.5 + 0.866025i,\ 0.5 - 0.866025i,\ 0\}$ of multiplicity three. It is straight forward to see from Fig. 3.5 and Fig. 3.6 that our methods (OM_4^1), (OM_4^2), (OM_4^3) and (OM_4^4) performed better as compared to the other methods namely, ZM_4, LM_4^1 and SM_4, respectively. But, our methods (OM_4^3) and (OM_4^4) have almost same basins of attraction as compared to SSM_4.

The last test problem is a non polynomial function as follows:

Test Problem 4. Let $p(z) = (z^3+1/z)^4$, having roots $\{0.707107+707107i, 0.707107-707107i, -0.707107+707107i, -0.707107-707107i\}$ of multiplicity four. It is straight forward to see from Fig.3. 7 and Fig. 3.8 that our methods (OM_4^3) and (OM_4^4) contain lower number of divergent points in comparison to the methods ZM_4, LM_4^1 and SM_4 respectively. But, they have same basins of attraction as compared to SSM_4.

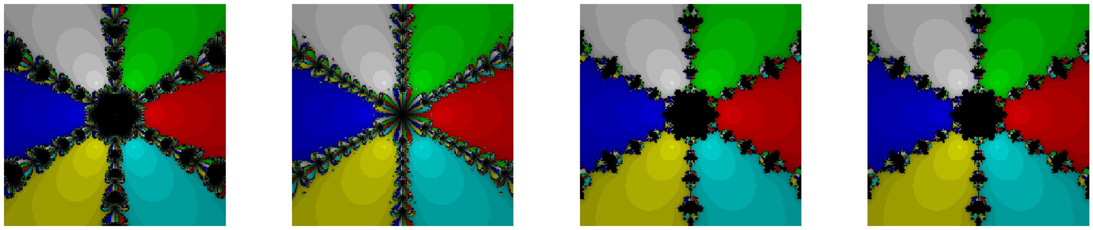

Figure 3.1: The basins of attraction for ZM_4, LM_4, SSM_4 and SM_4 respectively in test problem 1.

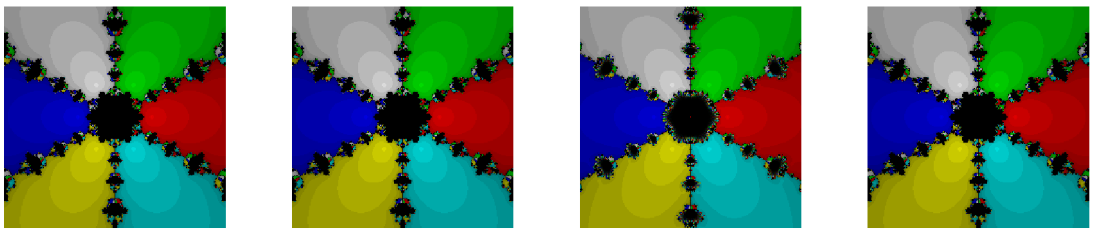

Figure 3.2: The basins of attraction for OM_4^1, OM_4^2, OM_4^3 and OM_4^4 respectively in test problem 1.

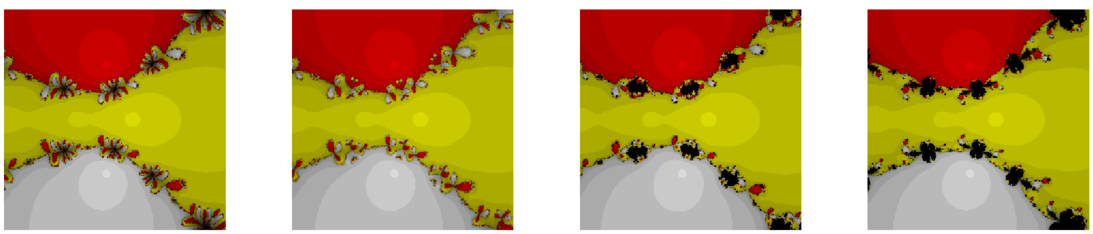

Figure 3.3: The basins of attraction for ZM_4, LM_4, SSM_4 and SM_4 respectively in test problem 2.

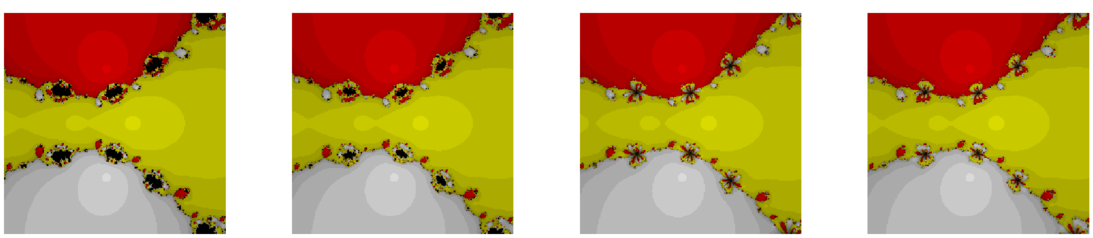

Figure 3.4: The basins of attraction for OM_4^1, OM_4^2, OM_4^3 and OM_4^4 respectively in test problem 2.

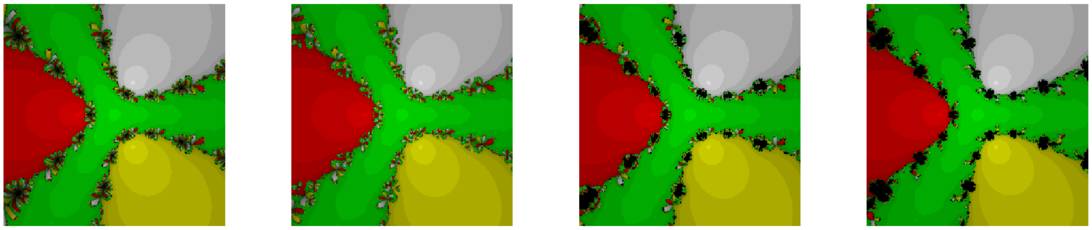

Figure 3.5: The basins of attraction for ZM_4, LM_4, SSM_4 and SM_4 respectively in test problem 3.

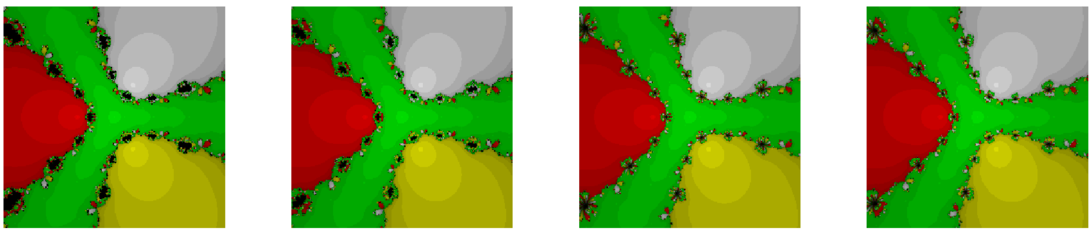

Figure 3.6: The basins of attraction for OM_4^1, OM_4^2, OM_4^3 and OM_4^4 respectively in test problem 3.

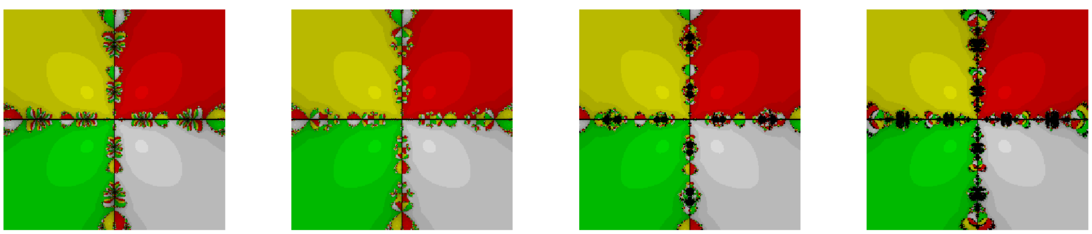

Figure 3.7: The basins of attraction for ZM_4, LM_4, SSM_4 and SM_4 respectively in test problem 4.

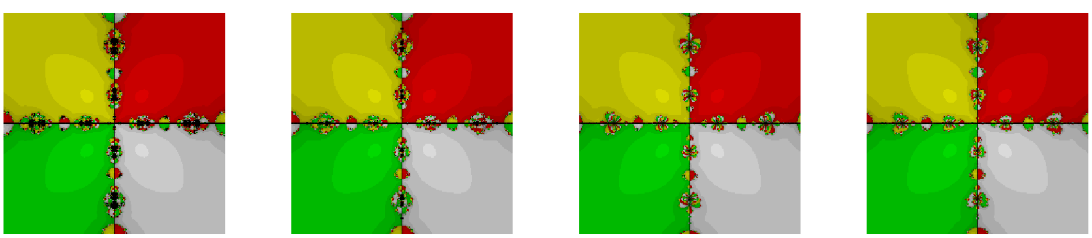

Figure 3.8: The basins of attraction for OM_4^1, OM_4^2, OM_4^3 and OM_4^4 respectively in test problem 4.

3.6 Concluding remarks

In this chapter, we find two optimal general classes of Jarratt's methods to find the multiple roots of nonlinear equations of higher order of convergence. Both the classes of methods requires one function and two of its first-order derivative evaluations per iteration step. Therefore, according to Kung-Traub conjecture all the proposed methods are optimal. Further, one can easily generate many new optimal families and some existing methods by fixing one of the free disposable parameters in our proposed schemes (3.2.19) and (3.3.7) respectively. Li et al. [LCN10, SXL09] methods, Soleymani et al. [SBL13] and Zhou et al. [ZCS11] method (11) are shown as special cases of our proposed schemes. From the numerical experiments, we conclude that newly proposed methods are equally competent with other similar robust methods available in literature, when the comparison of these methods is done with respect to number of iterations and accuracy is tested in multi-precision digits. Further, It is also noted that larger basins of attraction belong to our methods namely, (OM_4^3) and (OM_4^4) although the others methods are slow and has darker basins while some of the method are too sensitive upon the choice of the initial value.

Bibliography

[Atk89] Kendall E. Atkinson, *An introduction to numerical analysis*, second ed., John Wiley & Sons Inc., New York, 1989.

[Bas08] Dhiman Basu, *Family of cubically convergent schemes using inverse newton function for multiple roots*, Int. J. Comput. Methods Eng. Sci. Mech. **9** (2008), no. 3, 189–200.

[BG10] Jafar Biazar and Behzad Ghanbari, *A new third-order family of nonlinear solvers for multiple roots*, Comput. Math. Appl. **59** (2010), no. 10, 3315–3319.

[CBN09] Changbum Chun, Hwa ju Bae, and Beny Neta, *New families of nonlinear third-order solvers for finding multiple roots*, Comput. Math. Appl. **57** (2009), no. 9, 1574–1582.

[CC00a] Steven C. Chapra and Raymond P. Canale, *Numerical methods for engineers with programing and software applications*, Tata McGraw-Hill publishing company limited, New Delhi, 2000.

[CC00b] ——— , *Numerical methods for engineers with programing and software applications*, Tata McGraw-Hill publishing company limited, New Delhi, 2000.

[CN09] Changbum Chun and Beny Neta, *A third-order modification of Newton's method for multiple roots*, Appl. Math. Comput. **211** (2009), no. 2, 474–479.

[Don82] C. Dong, *A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation*, Math. Numer. Sinica **11** (1982), 445–450.

[Don87] ______, *A family of multipoint iterative functions for finding multiple roots of equations*, Int. J. Comput. Math. **21** (1987), no. 3-4, 363–367.

[GSNG10] Miquel Grau-Sánchez, Miquel Noguera, and José Manuel Gutiérrez, *On some computational orders of convergence*, Appl. Math. Lett. **23** (2010), no. 4, 472–478.

[HHL10] M. Heydari, S.M. Horseini, and G.M. Loghmani, *Convergence of family of third-order methods free from second-order derivatives for finding the multiple roots of nonlinear equations,*, World Appl. Sci. J. **11** (2010), no. 5, 507–512.

[Hom09] H. H. H. Homeier, *On Newton-type methods for multiple roots with cubic convergence*, J. Comput. Appl. Math. **231** (2009), no. 1, 249–254.

[HP77] Eldon Hansen and Merrell Patrick, *A family of root finding methods*, Numer. Math. **27** (1976/77), no. 3, 257–269.

[Jar66] P. Jarratt, *Multipoint iterative methods for solving certain equations*, Comput. J. **8** (1965/1966), 398–400.

[KG11] Young Ik Kim and Young Hee Geum, *A cubic-order variant of Newton's method for finding multiple roots of nonlinear equations*, Comput. Math. Appl. **62** (2011), no. 4, 1634–1640.

[KKS12] Sanjeev Kumar, V. Kanwar, and Sukhjit Singh, *On some modified families of multipoint iterative methods for multiple roots of nonlinear equations*, Appl. Math. Comput. **218** (2012), no. 14, 7382–7394.

[KL10] Y.I. Kim and S.D. Lee, *A third-order variant of newton-secant method finding multiple zero*, J. Chug. Math. Soc. **23** (2010), no. 4, 845–852.

[KT74] H. T. Kung and J. F. Traub, *Optimal order of one-point and multipoint iteration*, J. Assoc. Comput. Mach. **21** (1974), 643–651.

[Kum12] S. Kumar, *Development and analysis of some new iterative methods for numerical solutions of nonlinear equations*, Ph.D. thesis, Department of Mathematics, Sant Longowal Institute of Engineering and Technology Longowal (Deemed to be University), Punjab, India, 2012.

[LCN10] 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.

[LLC09] Shengguo Li, Housen Li, and Lizhi Cheng, *Some second-derivative-free variants of Halley's method for multiple roots*, Appl. Math. Comput. **215** (2009), no. 6, 2192–2198.

[McN07] J. M. McNamee, *Numerical methods for roots of polynomials. Part I*, Studies in Computational Mathematics, vol. 14, Elsevier B. V., Amsterdam, 2007.

[MP13] J. M. McNamee and V. Y. Pan, *Numerical methods for roots of polynomials. Part II*, Studies in Computational Mathematics, vol. 16, Elsevier B. V., Amsterdam, 2013.

[MR07a] N. A. Mir and Naila Rafiq, *Fourth-order two-step iterative methods for determining multiple zeros of non-linear equations*, Int. J. Comput. Math. **84** (2007), no. 7, 971–977.

[MR07b] ———, *Third-order and fourth-order iterative methods for finding multiple and distinct zeros of non-linear equations*, Appl. Math. Comput. **190** (2007), no. 1, 432–440.

[Net08] B. Neta, *New third order nonlinear solvers for multiple roots*, Appl. Math. Comput. **202** (2008), no. 1, 162–170.

[Net10] ———, *Extension of Murakami's high-order non-linear solver to multiple roots*, Int. J. Comput. Math. **87** (2010), no. 5, 1023–1031.

[NJ08] B. Neta and Anthony N. Johnson, *High-order nonlinear solver for multiple roots*, Comput. Math. Appl. **55** (2008), no. 9, 2012–2017.

[NSC12] Beny Neta, Melvin Scott, and Changbum Chun, *Basins of attraction for several methods to find simple roots of nonlinear equations*, Appl. Math. Comput. **218** (2012), no. 21, 10548–10556.

[Osa94] Naoki Osada, *An optimal multiple root-finding method of order three*, J. Comput. Appl. Math. **51** (1994), no. 1, 131–133.

[Osa06] ———, *Asymptotic error constants of cubically convergent zero finding methods*, J. Comput. Appl. Math. **196** (2006), no. 2, 347–357.

[Osa08] ———, *Chebyshev-Halley methods for analytic functions*, J. Comput. Appl. Math. **216** (2008), no. 2, 585–599.

[Ost60] A. M. Ostrowski, *Solution of equations and systems of equations*, Pure and Applied Mathematics, Vol. IX. Academic Press, New York-London, 1960.

[Ost73] ———, *Solution of equations in Euclidean and Banach spaces*, Pure and Applied Mathematics, Vol. 9, Academic Press, New York-London, 1973.

[PG10] P. K. Parida and D. K. Gupta, *A cubically convergent iteration method for multiple roots of* $f(x) = 0$, Int. J. Comput. Math. **87** (2010), no. 4, 877–884.

[PNPD12] M. S. Petković, B. Neta, L. D. Petković, and J. Džunić, *Multipoint methods for solving nonlinear equations*, Academic Press, 2012.

[Ral66] L. B. Rall, *Convergence of the Newton process to multiple solutions*, Numer. Math. **9** (1966), 23–37. MR 0210316 (35 #1209)

[RCT07] Aaditya V. Rangan, David Cai, and Louis Tao, *Numerical methods for solving moment equations in kinetic theory of neuronal network dynamics*, J. Comput. Phys. **221** (2007), no. 2, 781–798.

[SBL13] F. Soleymani, D. K. R. Babajee, and Taher Lotfi, *On a numerical technique for finding multiple zeros and its dynamic*, J. Egyptian Math. Soc. **21** (2013), no. 3, 346–353.

[Sch70] E. Schröder, *Ueber unendlich viele Algorithmen zur Auflösung der Gleichungen*, Math. Ann. **2** (1870), no. 2, 317–365.

[SMM12] Hamed Saffari, Nadia M. Mirzai, and Iman Mansouri, *An accelerated incremental algorithm to trace the nonlinear equilibrium path of structures*, J. Comput. Phys. **9** (2012), 425–442.

[SNC11] Melvin Scott, Beny Neta, and Changbum Chun, *Basin attractors for various methods*, Appl. Math. Comput. **218** (2011), no. 6, 2584–2599.

[SS10a] Janak Raj Sharma and Rajni Sharma, *Modified Jarratt method for computing multiple roots*, Appl. Math. Comput. **217** (2010), no. 2, 878–881.

[SS10b] ———, *Modified Jarratt method for computing multiple roots*, Appl. Math. Comput. **217** (2010), no. 2, 878–881.

[SS11] ———, *New third and fourth order nonlinear solvers for computing multiple roots*, Appl. Math. Comput. **217** (2011), no. 23, 9756–9764.

[SXL09] Li Shengguo, Liao Xiangke, and Cheng Lizhi, *A new fourth-order iterative method for finding multiple roots of nonlinear equations*, Appl. Math. Comput. **215** (2009), no. 3, 1288–1292.

[Tra64] J. F. Traub, *Iterative methods for the solution of equations*, Prentice-Hall, Englewood Cliffs, 1964.

[VN83] H. D. Victory, Jr. and B. Neta, *A higher order method for multiple zeros of nonlinear functions*, Internat. J. Comput. Math. **12** (1982/83), no. 3-4, 329–335.

[WF00] 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.

[ZCS11] Xiaojian Zhou, Xin Chen, and Yongzhong Song, *Constructing higher-order methods for obtaining the multiple roots of nonlinear equations*, J. Comput. Appl. Math. **235** (2011), no. 14, 4199–4206.

[ZCS13] ______, *Families of third and fourth order methods for multiple roots of nonlinear equations*, Appl. Math. Comput. **219** (2013), no. 11, 6030–6038.