INVESTIGATION OF DISTINCT ROOTS OF NONLINEAR EQUATIONS USING MODIFIED ROOT FINDERS

By

Aimen Safeer

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

June, 2023

Investigation of Distinct Roots of Nonlinear Equations Using Modified Root Finders

By

AIMEN SAFEER

A THESIS SUBMITED IN PARTIAL FULLFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE

In **Mathematics**

To

FACULTY OF ENGINEERING & COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

NATIONAL UNIUVERSITY OF MODERN LANGUAGES FACULTY OF ENGINEERIING & COMPUTER SCIENCE

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Title: Investigation of Distinct Roots of Nonlinear Equations Using Modified Root Finders

Submitted By: Aimen Safeer **Registration #:** 12/MS/Math/S20

Master of Science in Mathematics (MS Math)
Title of the Degree

Mathematics
Name of Discipline

Dr. Naila Rafiq
Name of Research Supervisor

Signature of Research Supervisor

Dr. Sadia Riaz
Name of HOD (MATH)

Signature of HOD (MATH)

Dr. Muhammad Noman Malik
Name of Dean (FE&CS)

Signature of Dean (FE&CS)

Date: 26 June, 2023

AUTHOR'S DECLARATION

I Aimen Safeer

Daughter of Muhammad Safeer Khan Abbasi

Registration #12/MS/MATH/S20

Discipline Mathematics

Candidate of **Master of Science in Mathematics** at National University of Modern Languages do here by declare that the thesis **Investigation of Distinct Roots of Nonlinear Equations Using Modified Root Finders** submitted by me in fractional fulfillment of **MS Mathematics** degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be cancelled and degree revoked.

Signature of Candidate ………………..

Name of Candidate…Aimen Safeer……

Date: 26 June, 2023

DEDICATION

I dedicate my thesis to my parents and teachers for their endless support and encouragement throughout my pursuit for education. I hope this achievement will fulfill the dream they envisioned for me.

Acknowledgements

I begin in the name of almighty Allah who has instilled in me the strength and spirit to complete the compulsory requirement of this dissertation. I offer my most humble words of thanks to the Holy Prophet Muhammad (Peace be upon him) who is a torch for mankind forever and who exhorts his followers to seek knowledge from both the cradle and the grave.

I experienced so much during this process, not only from the academic aspect but also from the personality aspect. I am grateful to **Dr. Naila Rafiq**, for providing access to facilities that ensured the successful completion of my work.

I heartily thank my respectable and honorable teacher and research supervisor, always smiling, Dr. Naila Rafiq , Assistant Professor, Department of Mathematics, NUML University Islamabad, for her proper guidance, encouragement, suggestions, inspiration, and a grateful keen interest in my work, which helped me a lot to accomplish this task with success. My deepest gratitude goes to my family members. Without support from them, it would not be possible to complete this thesis.

I would like to thank my dearest father, **M. Safeer khan Abbasi**, my mother, **Nazmeen Akhter** and my siblings. I am also thankful from the core of my heart to all my colleagues and friends for their motivation, encouragement, prayers, and sincere help during my studies. May Allah Almighty shower His countless blessings upon all of the above and give them a healthy and successful life.

Aimen Safeer

Abstract

In this research, derivative free and with memory iterative methods involving self-accelerating parameters have proposed for the solution of distinct roots of single variable non-linear equations. The technique of obtaining self-accelerating parameters is based on forth order iterative methods developed by Wang and Fan [16]. A very simple strategy has been used to construct two iterative methods using self-accelerating parameters which improve the convergence order from four to six. Numerical test examples show that the newly proposed methods are efficient, more accurate and robust in computation.

Contents

Chapter 1

Introduction

One of the most interesting problem in engineering, scientific computing and applied mathematics, in general, is the problem of solving a nonlinear equation

$$q(r) = 0. \tag{1.1}$$

In most of the cases, whenever real problems are faced, governing equations of the problems such as weather forecasting, accurate positioning of satellite systems in the desired orbit, measurement of earthquake magnitudes and other high-level engineering problems, cannot possibly be solved exactly. In such cases, approximate solutions may be worked out. Newton Raphson's method is the most familiar method for obtaining the solutions of nonlinear equation. It has quadratic convergence order and is an optimal method with two function evaluations per iterative step by the definition of Kung Traub conjecture.

In last few decades, mathematicians have done lot of work to get more improved approximate methods not only with higher convergence order but with better efficiency index also. As the order of convergence increases, so does the number of function evaluations per step. Hence, a new index to determine the efficiency called the efficiency index is introduced in order to measure the balance between these quantities.

The use of a variable self-accelerating parameter is suggested as a new technique with memory. In order to solve nonlinear equations, a modified Newton technique without memory with an invariant parameter is first built by replacing the Newton method's invariant parameter. Using a variable self-accelerating parameter without memory, a new Newton technique with memory is created. The novel Newton method with memory has a convergence order of $1+\sqrt{2}$ without doing any additional function evaluations. The main advancement is that the self-accelerating parameter is built in a straightforward manner.

The introduction and convergence study of a new two-parameter family of fourth-order iterative algorithms for numerically solving nonlinear equations (1.1) were studied. In the sense of the Kung-Traub conjecture, the new techniques of optimal order are the best. In many circumstances, the new family of fourth-order optimal techniques is superior to previous well-known and recently published fourth-order optimal methods, according to numerical experiments that are conducted to incorporate the theoretical results.

To solve nonlinear equations, iterative methods with memory have been developed often in recent years. Some of these techniques employ some variable parameters in the iteration schemes. These variable parameters, which are computed recursively while the iteration process proceeds, are known as the self-accelerating parameters. Data from both the previous and most recent iterations can be used to produce the self-accelerating parameter, which doesn't increase the processing costs for iterative algorithms. As a result, iterative algorithms with memory have extremely high levels of computational efficiency.

1.1 Significance of Study

The nonlinear equation (1.1) presents a challenge in applied science and engineering. Using the Colebrook equation to calculate the friction factor, or figuring out the critical values of a nonlinear function. Another example of how the initial eigenvalue of the Helmholtz equation can be found by decreasing a function .

A concrete range of real-world problems are considered, such as the path taken by an electron in the space created by two parallel plates, chemical engineering problem, the Van der Waal's equation, which incorporates the behavior of a real gas into the ideal gas equations, and fractional conversion in a chemical reactor in order to evaluate the viability, applicability and effectiveness of suggested approaches. Additionally, the numerical outcomes demonstrate that our suggested procedures beat those now in use of the same order when the accuracy is assessed in the multi precision digits.

Finding methods to solve nonlinear equations that provide an accurate and effective approximation of a nonlinear equation of the type has always been of almost importance in the field of numerical analysis. This topic's relevance to applied science and the four main human life domains is one of the primary causes for its prominence. Chemical, electrical, civil, and mechanical engineering are the four main engineering specialties. For instance, determining where the external points of a function describing a certain system that involves critical routes also call for the solution of algebraic equations, such as the zeros of the derivatives of that function, like figuring out all the ray pathways. The use of analytical techniques to solve such issues is essentially nonexistence. Therefore, we must use an iterative approach that can deliver a rough solution adjusted to a given level of accuracy.

A nonlinear equation that needs to be solved in engineering and applied sciences is the Colebrook equation, which calculates the friction factor. Finding the critical values of a nonlinear function is another illustration. Another example of how the first eigenvalue of the Helmholtz issue is found by lowering a functional is given by Ricceri. The majority of numerical solution techniques are based on Newton's scheme, which starts with an initial guess r_0 for the root r_t and builds a sequence from there using

$$r_{n+1} = r_n - \frac{q(r)}{q'(r)}. \tag{1.2}$$

1.2 Technical Approach

Assuming that $q : I \subset R \to R$ is sufficiently differentiable, we investigate the problem of locating simple roots of nonlinear equations (1.1). Let r_o be a first approximation to the simple root of $q(r) = 0$, with r_t being a simple zero. One of the aims is to build an iterative sequence of approximations $\{r_n\}_o^\infty$, that converges to, $\lim_{n\to\infty} r_n = r_t$, with n iterations. Newton's method, often known as the Newton-Raphson method, is one of the most popular and well-known iterative procedure. In the event of convergence, it converges linearly to multiple roots and quadratically to simple roots. The Newton's method and some of its variations are used in literature to either obtain higher order iterative approaches or to accelerate convergence. Since order four optimal methods combine high order of convergence and low computing cost, their development is crucial. For the iterative solution of nonlinear equations of the type $q(r) = 0$, In this studies, a new family of fourth-order convergent optimal methods utilizing a variation of Newton's method. The minimal number of new function evaluations each iteration for fourth-order convergent optimal techniques is three, which is what the new family necessitates.

In this study, two fast convergence iterative strategies for the solution of a single nonlinear equation $q(r) = 0$ with $q(r)$ being a non-differentiable function of r, which are derivative-free and do not require $q(r)$ for the differential of $q(r)$ are derived.

1.3 Comparison of Techniques

Studies on this topic subjects are included, Iterative techniques for approximating various generalized inverses, simple or multiple roots that can be found with or without derivatives, and real or complex dynamics linked to the rational functions that are produced when an iterative technique is used on a polynomial function are all examples of such techniques. Additionally, several adequate criteria for local, semi local, or global convergence have been used to analyses the convergence of the suggested approaches. Iterative methods are related to other areas of science and engineering, as shown by a number of manuscripts on signal processing, nonlinear integral equations, partial differential equations, or convex programming in addition to the theoretical works.

Compared to approaches that locate one root at a time, methods for simultaneously locating the roots of non-linear equations are fairly well known has further information on these methods, including their convergence analysis, computational effectiveness, and parallel implementation. Weierstrass correction is modified appropriately to permit convergence order eight with minimal computational cost and functional evaluation in each cycle, leading to a very high computational efficiency. Derivative-free methods are based on Steffensen's method as the first step since multistep methods are typically based on Newton's phases. Steffensen's approach for simple and multiple roots is the foundation for a number of derivative-free procedures. For a family of such multiple root approaches and for simple roots. In a recent article, Neta [1] demonstrated that, although not being second order, there is a superior option for a first step. Traub's method [2] is given (2.34), is of order 1.839, and compared to other derivative free approaches, it runs more quickly and has superior dynamics. Obviously, this is not how to obtain optimal approaches Kung and Traub hypothesized that multipoint approaches using the function that may be used without memory. The order of evaluations could not be more than $2d_1$. The definition of the efficiency index I as $p^{\frac{1}{d}}$. The efficiency index of an ideal order 8 method is therefore $I = 8^{1/4} = 1.6817$. as well as having an efficiency index of $I = 4^{1/3} = 1.5874$, which is ideal for a technique of order 4 superior to Newton's technique, when $I = p$. The ideal efficiency index is $I = \sqrt{2} = 1.4142$.

These techniques in particular, useful when evaluating the derivative will cost a lot of money and, of course, when the function is not distinguishable. Here, based on Traub's method, we create a derivative-free method with memory the derivative with the derivative as the first step, and the derivative as the basis for steps two and three of Newton interpolating a degree 3 polynomial. Researchers are now taking derivative free simultaneous approaches into consideration for more information. The major objective of this study is to provide a derivative-free method for identifying distinct roots of nonlinear equation.

1.4 Historical Background

It is a difficult undertaking involving many branches of science and technology to solve nonlinear equations in any Banach space, including real or complex nonlinear equations, nonlinear systems, and nonlinear matrix equations. Usually, the solution is out of reach and calls for an iterative algorithmic technique. Over the past few years, research in this field has expanded dramatically. The design, analysis of convergence, and stability of new iterative strategies for addressing nonlinear problems, as well as their application to real-world issues, are the key topics of this research.

In order to build the self-accelerating parameter, Dzunic et al. [3] presented various efficient self-accelerating type methods. The amount of self-accelerating parameters or the use of high-degree interpolation polynomials to create self-accelerating parameters can both significantly improve the convergence order and computational efficiency of self-accelerating type algorithms. The use of higher order self-accelerating techniques to solve nonlinear equations has increased recently useful tri-parametric iterative method produced a self-accelerating parameter utilizing a Newton interpolation polynomial. Using four self-accelerating parameters, Lotfi and Assari [4] achieved a derivative-free iterative technique with an efficiency index close to 2. In addition, an effective iterative approach with n self-accelerating parameters is presented which may be seen as a particular instance of some well-known methods .Cordero *et al.* [5] and Campos *et al.* [6] created several novel self-accelerating type approaches to investigate the stability of their methods . In order to evaluate the stability of their methods. Cordero *et al.* [5] and Campos *et al.* [6] developed a number of innovative self-accelerating type approaches. Utilizing a Newton interpolation polynomial, Zaka *et al.*[7] helpful tri-parametric iterative approach created a self-accelerating parameter and provided a powerful iterative method with n self-accelerating parameters, which may be viewed as a specific application of some well-known techniques. First, a modified optimal fourth-order technique is suggested for solving nonlinear equations based on Ren's method [8]. Then, using a self-accelerating parameter, the modified iterative technique is expanded into a new self-accelerating type method. The self-accelerating parameter is built using the interpolation approach, and the new method's convergence order is 4.2361. An approach is used to build the self-accelerating parameter modify this method with maximum convergence order 4.4495. Numerical examples are provided to support the theoretical findings.

It is common knowledge that several problems in various branches of research and engineering call for the solution of the nonlinear equation (1.1) where $q : I \rightarrow D$ is a scalar function for the intervals $I \subset R$ and $D \subset R$. The most popular procedures are iterative ones, including Newton's method, Halley's method, Cauchy's method, and so forth. Thus, one of the most crucial parts in contemporary researches is the development of iterative algorithms based on these iterative methods for locating the roots of nonlinear equations. To solve a single nonlinear equation, some iterative approaches with high-order convergence have recently been developed. These iterative methods can be built using a variety of approaches, including Taylor series, quadrature formulas, decomposition techniques, continuous fraction, Padé approximation, homotopy methods, Hermite interpolation, and clipping techniques. For instance, there are many ways of introducing Newton's method. Among these ways, using Taylor polynomials to derive Newton's method is probably the most widely known technique. Weerakoon and Fernando [9] use the trapezoidal quadrature formulae to derive an implicit iterative scheme with cubic convergence , where as Cordero and Torregrosa [10] create several adaptations of Newton's method using fifth order quadrature principles. Both groups take into account various quadrature formulas to compute the integral. Chun [11] presented a set of iterative techniques in 2006 that enhanced Newton's approach to solving nonlinear equations by employing the Adomian decomposition technique. A fourth-order convergent iterative technique based on Thiele's continuous portion of the function of the Halley's approach from the Padé approximation of the function use the divided differences to approximate the derivatives and leads to a few changes with third-order convergence. For the purpose of resolving nonlinear algebraic equations Abbas bandy *et al.* [12] presented a powerful numerical method based on the Newton-Raphson method and the homotopy analysis method. Noor *et al.* [13] presented and investigated a novel family of iterative techniques using the homotopy perturbation approach. In 2015, Wang *et al.* [14] developed a large family of n-point Newton type iterative techniques for solving nonlinear equations using direct Hermite interpolation. Additionally, certain effective univariate root-finding methods exist for a specific class of functions. For example, if q is a polynomial, to compute all solutions to the polynomial equation. A degree reduction-based technique for finding all univariate polynomial roots is presented in the literature by Barto *et al.* [15] and has a higher convergence rate than Newton's method. The most well-known and frequently applied iterative approach for root-finding issues is arguably Newton's method. Let's quickly review how the Newton iterative technique is derived by using Taylor's formula for the function $q(r)$.

The oldest challenge in mathematics and engineering in general is to solve the nonlinear equation (1.1). In numerous branches of science and engineering, nonlinear equations have a wide range of applications. Author considered an iterative strategies to approximate both the single root and all of the roots in order to identify the roots and examined both varieties of iterative systems in this research. There are numerous iterative techniques with various orders of convergence that can be used in the literature to approximate the roots of Ostrowski determined the efficiency index I of these iterative approaches as $I = \frac{k}{u}$ where k is the number of function evaluations per iteration and u is the order of convergence. The solution of a nonlinear algebraic problem is one of the subjects that arise frequently in computational mathematics. The problem can be expressed as a system of nonlinear algebraic equations or as the scalar case, $q(r) = 0$. Analytical methods cannot be used to find the solutions (assuming there are any). Analytical methods occasionally only provide the true result; the complicated zeros should be identified and reported. Therefore, numerical methods are a good option for resolving such nonlinear issues. Each of the computational methods currently in use has a distinct range of applicability as well as advantages and disadvantages.

Depending on the application being dealt with, two groups of methods those using derivatives and those not using them are recognized to be useful . In comparison to derivative-free methods, which have a smaller area for selecting initial approximations and require the use of a divided difference operator matrix, or in simpler terms, a dense matrix, to extend to higher dimensional problems, derivative-involved methods have a larger attraction basin and easier coding efforts. Two kinds of methods those employing derivatives and those without using them are recognized as useful depending on the application under consideration. Derivative-involved methods have a larger attraction basin and require less coding work compared to derivative-free methods, which have a smaller area for choosing initial approximations and need to use a dense matrix or to put it another way, a divided difference operator matrix, to extend to higher dimensional problem.

Take into account the issue of a real simple zero of a function $q : D \subset R \to R$. There are several areas of science and engineering where this problem is well suited for application. Iterative approaches for the solution of non-linear equations have been the subject of numerous articles [2], [9].

Two of the oldest iterative techniques for estimating a real or complex function's zero are Newton's and Halley's methods. The Newton-Raphson method, also referred to as Newton's method, iteratively approaches a straightforward real root of the real equation $q(r) = 0$. The well-known recurrence relation results in the generation of a sequence r_n that follows Newton's method (1.2).

In a region near, the sequence "r_n" converges quadratically. Newton's approach is without a doubt one of the most effective iterative techniques for non-linear equations.

The most well-known and often used approach is arguably Newton's method, which converges to the root quadratically by approximating the root of a nonlinear equation in one variable using the value of the function and its derivative. To put it another way, after a certain number of iterations, the process doubles the amount of valid decimal places or significant digits at each iteration.

The Newton's method, which approximates the root of a nonlinear equation in one variable using the value of the function and its derivative, is undoubtedly the most well-known and often applied method. It converges to the root quadratically. Or, to put it another way, the method doubles the number of significant digits or valid decimal places at each iteration after a specific number of iterations.

It is demonstrated that the proposed technique converges to the root and that the order of convergence is at least three in the vicinity of the root whenever the function's first and higher order derivatives exist in the vicinity of the root; in other words, the method roughly triples the number of following a few iterations, of significant digits. The results of computations strongly support this theory, and for some functions, the computational order of convergence is much higher than three.

1.5 Self-Accelerating Technique

The self-accelerating type technique is a kind of efficient iterative method with memory that chooses some variable parameters as self-accelerating parameters in the iteration processes used to solve nonlinear equations. The computing cost of the iterative method is not increased by using data from both previous and current iterations to find the self-accelerating parameter. As a result, self-accelerating algorithms have very high processing efficiency.

The self-accelerating parameter is crucial to the self-accelerating type technique since it can significantly affect how effective the iterative process is. The secant approach and the interpolation method are the two techniques for constructing the self-accelerating parameter.

1.6 Objectives of Study

The primary objective in this study is to provide some novel families of Kung's Technique approach for solving nonlinear equations of eighth and sixteenth-orders, which are anticipated to be superior than the existing schemes. The traditional Kung-Traub theory, developed in 1974, is supported by the presented families of optimal order. Some essential theorems describing the order of convergence of the suggested families, and a detailed study of the convergence properties are also given.

- In chapter 2, some useful iterative techniques are studied that are given in literature.

- In chapter 3, some basic definitions and related concepts are provided to facilitate the reader.
- In chapter 4, two with memory iterative methods are constructed for finding distinct roots of nonlinear equation based on the Wang and Fan technique [16].
- In chapter 5, numerical discussion of the newly proposed method are presented.
- In chapter 6, some conclusions and future research study in the direction of construction of more derivative free with memory iterative methods involving accelerating parameters with better convergence order for finding the distinct root of non linear equations.

Chapter 2

Literature Review

In the existing literature related to our topic of study, considerable efforts have been made to construct higher order methods. Several higher-order techniques have been constructed in order to improve the order of convergence of Newton's method. During literature survey, following articles have been studied:

- In 2010, Noor, M. A. *et al.* [17] used the following three-step iterative approach to solve nonlinear equations:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ t_n &= s_n - \frac{4q(s_n)}{q'(r_n) + 2q'((r_n+s_n)/2) + q'(s_n)}, \\ r_{n+1} &= t_n - \frac{4q(t_n)}{q'(r_n) + 2q'((r_n+t_n)/2) + q'(t_n)}, n = 0, 1, 2, \ldots \end{aligned} \tag{2.1}$$

  order of the convergence is 4. Error equation is given by

$$er_{n+1} = c_2^3 er_n^4 + O(er_n^5).$$

- In 2010, Wang, X. *et al.* [18] obtained the following three-step Newton's method to obtain a higher convergence order and a higher efficiency index than that of Newton's method:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ t_n &= s_n - \frac{q(s_n)}{q'(s_n)}, \\ r_{n+1} &= t_n - \frac{q(t_n)}{q'(t_n)}. \end{aligned} \tag{2.2}$$

It is easily proved that scheme (2.2) is eighth-order convergent and it requires six evaluations of the function and its first derivative. Scheme (2.2) has an efficiency index is $\sqrt[6]{8}$ =1.414, which is the same as Newton's method. In other words, scheme (2.2) does not increase the computational efficiency. To derive a scheme with a higher efficiency index, Wang, X. *et al.* approximate $q'(r_n)$, and $q\prime(s_n)$ using a Hermit interpolation. To approximate $q'(s_n)$, and constructed a Hermit interpolation polynomial $H_2\ (r)$, that meets the interpolation conditions:

$$\begin{aligned} H_2(r_n) &= q(r_n), \\ H_2(s_n) &= q(s_n), \\ H_2'(r_n) &= q'(r_n), \end{aligned}$$

$H_2(r)$ can be written as

$$H_2(r) = l_o(r)q(r_n) + l_1(r)q(s_n) + l_0^-(r)q'(r_n).$$

Therefore, a new scheme is derived as follows:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ t_n &= s_n - \frac{q(s_n)}{2q[r_n, s_n] - q'(r_n)}, \\ r_{n+1} &= t_n - \frac{q(t_n)}{2q[r_n, t_n] + q[s_n, t_n] - 2q[r_n s_n] + (s_n - t_n)q[s_n, r_n, r_n]}. \end{aligned}$$

Order of convergence of above method is 8. The error equation meets the following equations is,

$$er_{n+1} = c_2^2(c_2^2 - c_3)(c_2^3 - c_2c_3 + c_4)er_n^8 + O(er_n^9).$$

- In 2010, Dehghan, M. *et al.* [19] considered the new iterative method for solving $q(r) = 0$ as follows:

$$r_{n+1} = r_n + \frac{q(r_n)(q^{'}(r_n) + q^{'}(r^*_{n+1}))}{2q^{'}(r_n)q^{'}(r^*_{n+1})} + \frac{2}{3}\frac{q(r_n)}{q^{'}((r_n) + r^*_{n+1})/2)} - \frac{4}{3}\frac{q(r_n)}{q^{'}(r_n) + q^{'}(r^*_{n+1})} \tag{2.3}$$

where

$$r^*_{n+1} = r_n - \frac{q(r_n)}{q^{'}(r_n)}$$

For an open interval D.

Theorem **:** let D be a straightforward zero of the sufficiently differential function $q : D \subseteq R \rightarrow R$. The approach specified by Equation (2.3) is fourth order if r_1 is sufficiently close to σ, and its error equation is given by

$$er_{n+1} = \frac{9(q^{''}(\sigma))^3 - (q^{'}(\sigma))^2 q^{(4)}(\sigma)}{72(q^{'}(\sigma))^3} er_n^4 + O(er_n^5)$$

- In 2012, Wang X. *et al.* [20] considered the iteration strategy given below:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{\lambda q(r_n) + q'(r_n)}, \\ r_{n+1} &= s_n - \frac{q(s_n)}{\gamma q(r_n) + q'(r_n)} G(v_n), \end{aligned} \tag{2.4}$$

where λ, $\gamma \in R$ and $G(v_n)$ is a weight function with $v_n = \frac{q(s_n)}{q(r_n)}$.

Order of convergence of (2.4) is 4. It complies with the following error equation,

$$\begin{aligned} er_{n+1} &= -1/2(c_2 + \lambda)(2c_3 + c_2^2(-10 + G''(0)) + 2c_2\lambda(-7 + G''(0)) \\ &\quad +\lambda^2(-4 + G''(0)))er_n^4 + 0(er_n^5). \end{aligned}$$

- In 2013, Dzunic, J. [21] approached the modification of Newton's method as follow:

$$r_{k+1} = r_k - \frac{q(r_k)}{q'(w_k)}, \tag{2.5}$$

where $w_k = r_k + \gamma q(r_k)$ and leads to the new method error equation,

$$\varepsilon_{k+1} = c_2 \epsilon_k (2\varepsilon_{k,w} - \varepsilon_k) + O(\varepsilon_k^3) = c_2(1 + 2\gamma q'(r_t))\varepsilon_k^2 + O(\varepsilon_k^3).$$

Order of the method (2.5) is 2.

- In 2014, Jaiswal, J. P. [22] used the concept of inverse function to derive variants of Newton's Method (BN). Jaiswal used here inverse function,

$$r = q^{-1}(s) = g(s)$$

instead of $s = q(r)$, to get the following

$$r_{n+1} = r_n - \frac{(q'(r_n) + q'(s_n)q(r_n)}{q'(r_n)q'(s_n) + \sqrt{(1 + q'(r_n)^2)(1 + q'(s_n)^2)} - 1}, n \geq 0, \tag{2.6}$$

where

$$s_n = r_n - \frac{q(r_n)}{q'(r_n)}$$

The method (2.6) is third order convergent for simple root and its efficiency index is $3^{1/3} = 1.442$.

Function $s = q(r)$ has been used in (2.6). Jaiswal used here inverse function

$$r = q^{-1}(s) = g(s)$$

instead of $s = q(r)$ to get the following:

$$\begin{aligned} g(s) &= g(s_n) + \int_{s_n}^{s} g'(s)ds, \qquad (2.7) \\ &= q(s_n) + (s - s_n)[\frac{g'(r_n)g'(s_n) + \sqrt{(1 + g'(r_n)^2)(1 + g'(s_n)^2)} - 1}{(g'(r_n) + g'(s_n))}], \end{aligned}$$

where $s_n = q(r_n)$. Now by using the fact that $g'(s) = (q^{-1})'(s) = [q(r)]^{-1}$ and that $s = q(r) = 0$, obtained the following method:

$$r_{n+1} = r_n - q(r_n)[\frac{1 + \sqrt{1 + q'(r_n)^2(1 + q'(s_n)^2)} - q'(r_n)q'(s_n)}{(q'(r_n) + q'(s_n))}], \tag{2.8}$$

where $s_n = r_n - \frac{q(r_n)}{q'(r_n)}$. If the function have sufficient number of continuous derivatives in neighborhood of r_t which is simple root of q'then the Method (2.8) has third-order convergence. Error equation meets from the (2.8) is

$$er_{n+1} = (\frac{c_2^2}{1 + q'(\alpha)^2} + \frac{c_3}{2})er_n^3 + O(er_n^4).$$

- In 2014, Eftekhari, T. *et al.* [23] main objective to presented some highly effective multipoint approaches with an index, one can approximate the nonlinear equation's root $q(r) = 0$ the multipoint techniques with memory then become:

$$\begin{aligned} w_n &= r_n + \beta_n q(r_n), \\ s_n &= r_n - \frac{\beta_n q(r_n)^2}{q(w_n) - q(r_n)}, \\ t_n &= s_n - \phi_k(\frac{q(s_n)}{q[r_n, s_n]}), k = 1, 2, 3 \\ r_{n+1} &= t_n - (1 - \frac{q(t_n)}{q'(w_n)})^{-1} \times (1 - \frac{q(s_n)^3}{q(w_n)^2 q(r_n)})(\frac{q[r_n, s_n]q(t_n)}{q[s_n, t_n]q[r_n, t_n]}), \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} \beta_{n1} &= -\frac{1}{q'(r_t)} = -\frac{1}{N_3'(r_n)} \\ \beta_{n2} &= -\frac{1}{q'(r_t)} = -\frac{1}{N_3'(r_n)} \\ \beta_{n3} &= -\frac{1}{q'(r_t)} = -\frac{1}{N_4'(r_n)} \end{aligned}$$

and

$$\begin{aligned} \phi_1 &= (1 - \frac{q(s_n)}{q(w_n)})^{-1} \\ \phi_1 &= 1 - \frac{q(s_n)}{q(w_n)} + (\frac{q(s_n)}{q(w_n)})^2 \\ \phi_3^{'} &= \frac{q[r_n, w_n]}{q[w_n, s_n]} \end{aligned}$$

error equation of the method (2.9) which is mentioned above is

$$er_{n+1} \sim c_{n,8} c_5^3 D_{n-1,p}^3 D_{n-1,s}^3 D_{n-1,r}^8 er_{n-1}^{3+3q+3p+3s+8r}.$$

Therefore, the R-order of the methods with memory (2.9), when β_{n1} is calculated is at least 10.7202, when β_{n2} is calculated is at least 11 and when β_{n3} is calculated is at least 11.2915.

- In 2016, Maroju, P. *et al.* [24] presented an eight order family of king's method. Considered the following three steps scheme:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ t_n &= s_n - \frac{q(r_n) + \beta q(s_n)}{q(r_n) + (\beta - 2) q(s_n)} \frac{q(s_n)}{q'(r_n)}, \beta \in \mathbb{R} \\ r_{n+1} &= t_n - \frac{q(t_n)}{q'(r_n)} G(u, v), \end{aligned} \tag{2.10}$$

where the above weight function $G : C^2 \to C$, is an analytic function in the neighborhood of $(0, 0)$ and

$$\begin{aligned} u &= \frac{q(t_n)}{q(s_n)}, \\ v &= \frac{q(s_n)}{q(r_n)}. \end{aligned} \tag{2.11}$$

Maroju, P. *et al.* also proved that the following error estimates for given weight functions that is the quotients defined in (2.11) satisfied the following error equations

$$u = \frac{q(t_n)}{q(s_n)} = O(er_n^2),$$
$$v = \frac{q(s_n)}{q(r_n)} = O(er_n).$$

Order of convergence of the above scheme (2.10) is 8 and error equation is

$$
\begin{aligned}
er_{n+1} &= -\frac{c_2((2\beta+1)c_{2-}^2 - c_3)}{2}[c_2^4(4\beta^3 - 32\beta^2 + 44\beta + 2\beta G_{12} + G_{12} \\
&\quad +(2\beta+1)^2 G_{20} - 82) + c_3^2(G_{20} - 2) - 2c_4c_2 - \\
&\quad c3c_2^2(-4\beta + G_{12} + 4\beta G_{20} + 2G_{20} - 30)]er_n^8 + O(er_n^9).
\end{aligned}
$$

- In 2019, Shengfeng, Li. *et al.* [25] used a modified iteration method as shown below:

$$
\begin{aligned}
t_k &= r_k - \frac{q(r_k)}{q'(w_k)}, \\
r_{k+1} &= r_k - \frac{r_k - t_k}{1 + 2q(t_k)q'^2(r_k)L^{-1}(r_k)},
\end{aligned}
\tag{2.12}
$$

where

$$
Lr_k = q(r_k)(q(r_k)q''(r_k) - 2q'^2(r_k)).
$$

It is at least 4-order convergence. The error equation meets the above equation is :

$$
er_{k+1} = (b_2^3 - 2b_2b_3)er_k^4 + O(er_k^5).
$$

- In 2019, Mir, N. A. *et al.* [26] constructed an eighth order derivative free simultaneous method which is more efficient than the similar methods existing in the literature. Considered the following scheme as given below,

$$
\begin{aligned}
s_i &= r_i - \frac{q^2(r_i)}{q(r_i + q(r_i)) - q(r_i)}, \\
t_i &= r_i - \frac{q^2(r_i) + q(s_i)q(r_i)}{q(r_i + q(r_i)) - q(r_i)},
\end{aligned}
\tag{2.13}
$$

Take well-known two-step fourth order Newton's method

$$\begin{aligned} v_i &= r_i - \frac{q(r_i)}{q'(r_i)}, \\ u_i &= v_i - \frac{q(v_i)}{q'(vi)}, \end{aligned} \tag{2.14}$$

Mir, N. A. *et al.* converted method (2.14) into derivative free simultaneous method for extracting all the distinct roots of non-linear equation $q(r) = 0$.

$$\begin{aligned} v_i &= r_i - \frac{q(r)}{\Pi_{\substack{j\neq i \\ j=1}}^{n}(r_i - H(r_{j^*}))}, \\ u_i &= v_i - \frac{q(vi)}{\Pi_{\substack{j\neq i \\ j=1}}^{n}(v_i - v_j)}. \end{aligned} \tag{2.15}$$

The convergence order of method (2.15) is eight.

- In 2019, Chand, B. p. *et al.* [27] proposed the following two-step method using a weight function, whose iterative expression is:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)} \\ r_{n+1} &= r_n - w(t_n)\frac{q(r_n) + q(s_n)}{q'(r_n)} \end{aligned} \tag{2.16}$$

where $w(t_n) = a_1 + a_2 t_n + a_3 t_n^2$ and $t_n = \frac{q(s_n)}{q'(r_n)}$. The convergence order of (2.16) is proved in the following theorem.

Theorem: Let q be a real or complex valued function defined in the interval I having a sufficient number of smooth derivatives. Let r_t be a simple root of the equation $q(r) = 0$ and the initial point r_0 is close enough to r_t. Then, the method (2.16) is fourth order of convergence if $a_1 = 1, a_2 = 0$ and $a_3 = 2$.

Therefore, the error equation of the method (2.16) becomes

$$er_{n+1} = (3c_2^3 - c_2c_3)er_n^4 + O(er_n^5).$$

In view of Theorem, the proposed fourth order method is

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)} \\ r_{n+1} &= r_n - (1 + 2(\frac{q(s_n)}{q(r_n)})^2)\frac{q(r_n) + q(s_n)}{q'(r_n)} \end{aligned} \tag{2.17}$$

which requires three function evaluations per iteration and consequently is optimal. In addition, the efficiency index of (2.17) is 1.5874.

Using the results obtained in (2.17), proposed a new method defined by:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)} \\ r_{n+1} &= r_n - (1 + 2(\frac{q(s_n)}{q(r_n)})^2)\frac{q(r_n) + q(s_n)}{q'(r_n)} \\ r_{n+1} &= t_n - w_1(t_n)\frac{q(t_n)}{q'(r_n)} \end{aligned} \tag{2.18}$$

where $w(t_n) = b_1 + b_2t_n$ is a new weight function and t_n is as in (2.16). The order of convergence is shown in the following result.

Theorem: Let q be a real or complex valued function defined in an interval I having a sufficient number of smooth derivatives. Let r_t be a simple root of the equation $q(r) = 0$ and the initial point r_o is close enough to r_t. Then, (2.18) has a sixth order of convergence if $b_1 = 1$ and $b_2 = 2$.

The error equation of the (2.18) is

$$er_{n+1} = c_2(18c_2^4 - 9c_2^2c_3 + c_2^3)er_n^6 + O(er_n^7).$$

- In 2019, Khdhr, F. W. *et al.* [28] suggested an effective bi-parametric iterative method with memory as follows:

$$\begin{aligned} w_k &= r_k - B_{kn} = q(r_k), \\ B_k &= \frac{1}{N_4'(r_k)}, \alpha_k = \frac{N_5'''(w_k)}{2N_5'(w_k)}, \ \ k \geq 2 \\ r_{k+1} &= r_k - \frac{q(r_k)}{q[r_k, w_{k]}}(1+\alpha_k)\frac{q(w_k)}{q[r_k, w_{k]}}, \ \ k \geq 0. \end{aligned} \tag{2.19}$$

Therefore, equation (2.19) one-step approach with memory has an R-order of convergence of 3.90057. The computational efficiency index of (2.19) is $3.90057^{1/2} \approx 1.97499 \approx 2$.

Error equation of the above scheme (2.19) is:

$$er_{k+1} \sim er_{k-2}^{2(r^{2+pr+r+p+1})}$$

- In 2020, Barrada, M. *et a*l. [29] arrived at the iterative procedure shown below, which served as a broad variant of Halley's approach HP for locating simple roots:

$$r_{n+1}^p = r_n^p - w_p(L_n)\frac{q(r_n)}{q'(r_n)}, \tag{2.20}$$

where

$$w_p(L_n) = \frac{S_p(L_n)}{S_{p+1}(L_n)}, \ \ n = 0, 1, 2, 3, ...$$

and

$$\begin{aligned}
S_0(L_n) &= 1,\\
S_1(L_n) &= 1-\frac{L_n}{2},\\
S_{p+2}(L_n)-S_{p+1}(L_n) &= -\frac{L_n}{2}S_p(L_n).
\end{aligned}$$

where p is a parameter, which is a nonnegative integer.

Error equation of the method (2.20) is

$$er_{n+1}=[2(1-w_p^{''}(0))c_2^2-c_3]er_n^3+O(er_n^4)$$

- In 2020, Barrada, M. *et al.* [30] proposed a new method for finding simple roots of nonlinear equations with cubical convergence as follows:

$$r_{n+1}=r_n-\frac{q(r_n)}{q'(r_n)+\frac{q''(r_n)}{2}(r_{n+1}-r_n)}, \tag{2.21}$$

In order to show the power and efficiency of method, a comparative analytic study is provided by author between the proposed method and other third and higher order method. The simplicity and power of the proposed formula pushed to do a first study of its global convergence.

By replacing $(r_{n+1}-r_n)$ remaining in the denominator of right-hand side of (2.21) by Halley's correction, author gets the following famous method of Super Halley,

$$\begin{aligned}
r_{n+1} &= r_n-\frac{q(r_n)}{2q'(r_n)}(\frac{2-L_n}{1-L_n}), && (2.22)\\
SH(r) &= r-\frac{q(r_n)}{2q'(r_n)}(\frac{2-L_{q(r)}}{1-L_{q(r)}}).
\end{aligned}$$

Now, by replacing $(r_{n+1}-r_n)$ located on the right-hand side of (2.21) by Super Halley's correction given in (2.22), to get the following,

$$\begin{aligned} r_{n+1}^2 &= r_n^2 - \frac{q(r_n^2)}{q'(r_n)} \cdot W_2(L_n), \\ W_2(L_n) &= \frac{4(1-L_n)}{L_n^2 - 6L_n + 4}. \end{aligned} \tag{2.23}$$

Above equation (2.23) has cubically convergent order and satisfied the error equation

$$er_{n+1} = -c_3 er_n^3 + O(er_n^4).$$

- In 2020, Wang, X. *et al.* [31] presented a novel way to construct the self-accelerating parameter and derived a modified newton method. Considered the following scheme:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ r_{n+1} &= s_n - T(s_n - r_n)^2. \end{aligned} \tag{2.24}$$

  where $T \in R$. The order of convergence of the iterative method (2.24) is two and its error equation meets the following equation is:

$$er_{n+1} = (c_2 - T)er_n^2 + O(er_{n-1}^3).$$

  where $er_n = r_n - r_t$, $c_2 = \frac{q^{(2)}(r_t)}{2q'(r_t)}$, *and* $T \in R - \{0\}$. If the variable parameter T_n satisfies, $\underset{n\to\infty}{Lim} T_n = c_2$ then the asymptotic convergence constant to be zero.

The new self accelerating parameter is given by the following

$$T_{n1} = \frac{s_{n-1} - s_n}{(r_n - r_{n-1})^2}, \tag{2.25}$$

$$T_{n2} = \frac{s_{n-1} - s_n}{(s_{n-1} - r_{n-1})^2}, \tag{2.26}$$

$$T_{n3} = \frac{s_{n-1} - s_n}{(s_{n-1} - r_{n-1})(r_n - r_{n-1})} \tag{2.27}$$

Replacing the parameter T in equation (2.24) with T_n, obtained the following iterative method with memory:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ r_{n+1} &= s_n - T_n(s_n - r_n)^2. \end{aligned} \tag{2.28}$$

R-order of convergence of iterative method (2.28) is atleast $1 + \sqrt{2} \approx 2.414$.

- In 2020 Wang, X. *et al.* [16] constructed the following modified iterative method:

$$\begin{aligned} w_n &= r_n + q(r_n), \\ t_n &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \\ s_n &= t_n - T(t_n - r_n)^2, \\ r_{n+1} &= s_n - \frac{q(s_n)}{q[r_n, s_n] + q[s_n, w_n] - q[r_n, w_n]}, \end{aligned} \tag{2.29}$$

where $T \in R$ is a self-accelerating parameter. If function $q : I \subset R \to R$ is sufficiently differentiable and has a simple zero r_t on an open interval I, then iterative method (2.29) is of fourth-order convergence and its error equation is as follows:

$$er_{n+1} = (c_2 + c_2 q'(r_t) - T)[c_2^2(1 + q'(r_t)) - c_3(1 + q'(r_t)) - c_2 T)]er_n^4 + O(er_n^5).$$

Where $er_n = r_n - r_t, T \in R$ and

$$c_m = (\frac{1}{m!})\frac{q^m(r_t)}{q'(r_t)}, n = 2, 3, 4, \ldots$$

- In 2020, Rafiq, N. *et al.* [32] proposed the following family of iterative methods.

$$\begin{aligned} s_i &= v_i - \frac{q(v_i)}{q'(v_i)}, \\ t_i &= s_i - \frac{q(v_i)}{q'(v_i)}\Gamma(u), \end{aligned} \tag{2.30}$$

where

$$u = \frac{q(v_i)}{q(r_i)}.$$

For iterative scheme (2.30), the following convergence theorem

Theorem: Let $r_t \in I$ be a simple root of sufficiently differential equation $q : I \subseteq R \rightarrow R$ in an open interval I. If v_o is sufficiently close to r_t and Γ be a real valued function satisfying $\Gamma(0) = 0$, $\Gamma^{'}(0) = 1$, $\Gamma^{''}(0) = 4$ and $\Gamma^{'''}(0) < \infty$, then the convergence method of the family of iterative method (2.30) is 4 and satisfies the following error equation :

$$er_{i+1} = (5c_2^3 - c_2c_3 - \frac{1}{6}\Gamma^{'''}(0)c_2^3)er_i^4 + O(er_i^5).$$

where

$$c_m = \frac{q^m(r_t)}{m!q'(r_t)} \quad : m \geq 2.$$

- In 2020, Chu, Y. *et al.* [33] constructed a ninth order derivative free simultaneous method which is more efficient than the similar methods existing in literature, author considered eighth order derivative free Kung–Traub's [34] family of iterative method (abbreviated as KF):

$$\begin{aligned}
\sigma^{(t)} &= \eta^{(t)} - \frac{r_t q(\eta^{(t)})^2}{q(v^{(t)}) - q(\eta^{(t)})}, \\
u^{(t)} &= \sigma^{(t)} - (\frac{q(\sigma^{(t)})q(v^{(t)})}{(q(v^{(t)}) - q(\sigma^{(t)})(\frac{q(\eta^{(t)} - q(\sigma^{(t)}}{\eta^{(t)} - \sigma^{(t)}})}), \\
z^{(t)} &= u^{(t)} - (\frac{q(\sigma^{(t)})q(v^{(t)})(\sigma^{(t)} - \eta^{(t)} + \frac{q(\sigma^{(t)})}{\frac{q(\eta^{(t)}) - q(u^{(t)})}{\eta^{(t)} - u^{(t)}}})}{(q(\sigma^{(t)}) - q(u^{(t)}))(q(v^{(t)} - q(u^{(t)}))}) + (\frac{q(\sigma^{(t)})}{\frac{q(\sigma^{(t)}) - q(u^{(t)})}{\sigma^{(t)} - u^{(t)}}}),
\end{aligned} \tag{2.31}$$

where

$$v^{(t)} = \eta^{(t)} + r_t q(\eta^{(t)}).$$

Let $\zeta_1, \zeta_2, \ldots \zeta_n$ be n simple roots of $q(\zeta) = 0$. If $\eta_1^{(0)}, \eta_2^{(0)}, \ldots \eta_n^{(0)}$ be the sufficiently close initial approximations to actual roots, then the order of convergence of (2.31) is nine.

- In 2021, Liu, C. S. *et al.* [35] decomposed $q(r) = 0$ by:

$$q(r) = g(r)r + ar - b,$$

and derived a a novel iterative scheme for solving a single non–linear equation $q(r) = 0$.

$$r_{n+1} = r_n - \frac{q(r_n)r_n}{(aw + c)r_n + (1 - w)[b + q(r_n)}. \tag{2.32}$$

Liu, C. S. *et al.* concerned with the local convergence property of the iterative scheme (2.32). The iterative scheme (2.32) for solving $q(r) = 0$ has third-order convergence, if the parameters w and c are given by

$$w = 1 - \frac{c_2 r_n}{c_1}, c = c_1 - a + \frac{a c_2 r_n}{c_1} - \frac{b c_2}{c_1},$$

where $c_1 = q'(r)$ and $c_2 = \frac{q''(r)}{2}$ and $q(r) = 0$ and $q'(r) \neq 0$ for a simple root r_t. The error equation of the above method is,

$$er_{n+1} = er_n - q(er_n) = er_n - er_n - \frac{q'''(0)}{6} er_n^3 + ... = -\frac{q'''(0)}{6} er_n^3 + \cdots$$

- In 2021, Neta B. [36] developed a derivative-free method with memory based on Traub's method

$$s_n = r_n - \frac{q(r_n)}{\frac{(q(r_{n-2})-q(r_n))}{(r_{n-2}-r_n)} - \frac{q(r_{n-2})-q(r_{n-1})}{r_{n-2}-r_{n-1}} + \frac{q(r_{n-1})-q(r_n)}{r_{n-1}-r_n}} \tag{2.33}$$

as the first step and the other two steps are based on replacing the derivative by the derivative of Newton interpolating polynomial of degree 3. In the next section, Neta B. discussed the order of the scheme and the computational order of convergence, COC, defined by

$$COC = \frac{\ln\left|\frac{r_i - r_t}{r_{i-1} - r_t}\right|}{\ln\left|\frac{r_{i-1} - r_t}{r_{i-2} - r_t}\right|}.$$

Neta B. suggested a 3-step method having (2.33) as the first step. The method is given by

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{\Psi(r_n, r_{n-1}, r_{n-2})} \\ t_n &= s_n - \frac{q(s_n)}{q'(s_n)}, \\ r_{n+1} &= t_n - \frac{q(t_n)}{q'(t_n)}, \end{aligned} \tag{2.34}$$

where $\Psi(r_n, r_{n-1}, r_{n-2}) = \frac{q(r_{n-2})-q(r_n)}{r_{n-2}-r_n} - \frac{q(r_{n-2})-q(r_{n-1})}{r_{n-2}-r_{n-1}} + \frac{q(r_{n-1})-q(r_n)}{r_{n-1}-r_n}$.

The error in the first step is given by

$$es_n = Cer_n^{1.839},$$

where C is the computational error constant.

The other two steps are of the same order as the Newton's method, i.e. $et_n = es_n^2$ and $er_{n+1} = et_n^2$. Therefore, the order of the method is $4 \times 1.839 = 7.356$. The efficiency index $I = p^{1/d} = (7.3561)^{\frac{1}{3}} = 1.945$ is higher than that of the 3-step optimal eighth order method.

The order of convergence is proved in form of following theorem.

Theorem 1 *Let $q : I \subset R \to R$ be a differential in an open interval I and $r_t \in I$ be a simple zero of q. Then iterative method (2.34) has order of convergence 7.356 and its error equation is as follow:*

$$er_{n+1} = (er_n)^{7.356}$$

Proof. Let, $er_n = r_n - r_t$, $er_{n-1} = r_{n-1} - r_t$, $er_{n-2} = r_{n-2} - r_t$ be error at n^{th}, $(n-1)^{th}$ and $(n-2)^{th}$ step. Now, using Taylor expansion $q(r_n)$ around r_t as follows:

$$q(r_n - r_t + r_t) = q(r_t) + (r_n - r_t)q^{'}(r_t) + \frac{(r_n - r_t)^2}{2!}q^{''}(r_t) + \cdots \ .$$

Since, $q(r_t) = 0$ as r_t is given as simple zero, thus we have:

$$q(r_n) = er_n + c_2 er_n^2 + c_3 er_n^3 + c_4 er_n^4 + c_5 er_n^5 + c_6 er_n^6 + c_7 er_n^7 + O(er_n^8), \qquad (2.35)$$

where

$$c_m = \frac{q^m(r_t)}{m! q^{'}(r_t)}, \quad m = 2, 3, \ldots \ .$$

Similarly, we calculate:

$$q(r_{n-1}) = er_{n-1} + c_2 er_{n-1}^2 + c_3 er_{n-1}^3 + c_4 er_{n-1}^4 + c_5 er_{n-1}^5 + c_6 er_{n-1}^6 c_7 er_{n-1}^7 + O(er_{n-1}^8), \tag{2.36}$$

and

$$q(r_{n-2}) = er_{n-2} + c_2 er_{n-2}^2 + c_3 er_{n-2}^3 + c_4 er_{n-2}^4 + c_5 er_{n-2}^5 + c_6 er_{n-2}^6 + c_7 er_{n-2}^7 + O(er_{n-2}^8). \tag{2.37}$$

From (2.35) and (2.36) we get:

$$\frac{q(r_{n-1}) - q(r_n)}{r_{n-1} - r_n} = 1 + c_2(er_n - er_{n-2}) + c_3(er_n^2 + er_{n-1}^2 + er_n er_{n-1}) + \cdots, \tag{2.38}$$

with the similar calculations we obtain the following results after simplifications:

$$\frac{q(r_{n-2}) - q(r_{n-1})}{r_{n-2} - r_{n-1}} = 1 + c_2(er_{n-1} + er_{n-2}) + c_3(er_{n-2}^2 + er_{n-1}^2 + er_{n-1} er_{n-2}) + \cdots \tag{2.39}$$

and

$$\frac{q(r_{n-2}) - q(r_n)}{r_{n-2} - r_n} = 1 + c_2(er_n + er_{n-2}) + c_3(er_n^2 + er_{n-2}^2 + er_n er_{n-2}) + \cdots . \tag{2.40}$$

Using algebraic manipulation with equations (2.38-2.40) that required to obtain $\Psi(r_n, r_{n-1}, r_{n-2})$

$$\begin{aligned} \Psi(r_n, r_{n-1}, r_{n-2}) &= 1 + c_2(er_{n-1} - er_{n-2}) - c_2(er_n + er_{n-2}) + c_2(er_n + er_{n-1}) \\ &\quad - c_3 er_{n-2} er_{n-1} \end{aligned} \tag{2.41}$$

Now, in order to obtain first step of method (2.34), by using equations (2.35) and (2.41) we get the final result:

$$\begin{aligned} s_n &= r_t + c_3 er_n er_{n-1} er_{n-2} + \ldots , \\ es_n &\approx c_3 er_n er_{n-1} er_{n-2}, \end{aligned} \tag{2.42}$$

where $es_n = s_n - r_t$ is an error in first step. Now by using definition of convergence order at nth step that is given in form of following expression:

$$|er_{n+1}| \approx C\,|er_n|^p\,. \tag{2.43}$$

Using (2.43) in (2.42), the error equation (2.42) will be modified as follows (for instance consider $er_{n+1} = es_n$):

$$\begin{aligned} C\,|er_n|^p &\approx |c_3|\,|er_n|\,|er_{n-1}|\,|er_{n-2}| \\ |er_n|^{p-1} &\approx \frac{|c_3|}{C}\,|er_{n-1}|\,|er_{n-2}|\,, \end{aligned}$$

again using $|er_{n+1}| \approx C\,|er_n|^p$, we simplified above expression as follows:

$$\begin{aligned} |er_{n-1}|^{p^2-p-1} &\approx \left(\frac{|c_3|}{C^2}\right)|er_{n-2}| \\ |er_{n-1}| &\approx \left(\frac{|c_3|}{C^2}\right)^{\frac{1}{p^2-p-1}}|er_{n-2}|^{\frac{1}{p^2-p-1}}\,. \end{aligned}$$

Following the definition of convergence order, we observed that $C = \left(\frac{|c_3|}{C^2}\right)^{\frac{1}{p^2-p-1}}$ and

$$p = \frac{1}{p^2-p-1}$$

which implies

$$p^3 - p^2 - p - 1 = 0. \tag{2.44}$$

Since condition on p is that $p > 0$, leads the following positive solution of nonlinear equation (2.44) is 1.839544. We conclude that for the first step of method (2.34)

$$es_n \approx (er_n)^{1.8395}. \tag{2.45}$$

Again by using Taylor series we obtain the following series:

$$q(s_n) = es_n + d_2 es_n^2 + d_3 es_n^3 + d_4 es_n^4 + d_5 es_n^5 + d_6 es_n^6 + d_7 es_n^7 + O(es_n^8), \qquad (2.46)$$

where

$$d_j = \frac{q^j(s_n)}{j! q'(s_n)}, \quad j = 2, 3, \dots .$$

and

$$q'(s_n) = 1 + 2d_2 es_n + 3d_3 es_n^2 + 4d_4 es_n^3 + 5d_5 es_n^4 + 6d_6 es_n^5 + 7d_7 es_n^6 + O(es_n^7). \quad (2.47)$$

After algebraic calculations of equations (2.46) and (2.47), we get an error equation for the 2nd step of method (2.34) as follows:

$$\begin{aligned} t_n &= r_t + d_2 es_n^2 + O(es_n)^3, \\ et_n &= d_2 es_n^2 + O(es_n)^3. \end{aligned}$$

Therefore,

$$et_n \approx es_n^2 \qquad (2.48)$$

In a similar way, error equation for the third step is obtained as follows:

$$er_{n+1} \approx et_n^2. \qquad (2.49)$$

Combining (2.45), (2.48) and (2.49), we have:

$$er_{n+1} \approx et_n^2 \approx \left(es_n^2\right)^2 \approx (er_n)^{1.8395 \times 4} = (er_n)^{7.356}.$$

Hence proved the theorem. ■

In 2022, Thota, S. *et al.* [37] used the following iterative schemes to find the approximate solution r_{n+1}.

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)} - \frac{(q(r_n))^2 q''(r_n)}{2(q'(r_n))^3}, \\ r_{n+1} &= r_n - \frac{q(r_n)}{q'(r_n)} - \frac{(q(r_n))^2 q''(r_n)}{2(q'(r_n))^3} - \frac{q(s_n)}{q'(s_n)}. \end{aligned} \tag{2.50}$$

Order of convergence of iterative method (2.50) is 6.

Chapter 3

Basic Definations

In order to facilitate the reader, we are going to employ some basic definitions and related concepts all throughout the dissertation.

3.1 Types of Equation

In this section, some definitions of equations at its types are given to clarify the ideas about it.

Linear Equation

An equation is said to be linear if it is linear in the involved variables [38]. For example,

$$s = -7r + \frac{5}{4}.$$

Non-linear Equation

An equation of the form,

$$q\left(r\right) = 0,$$

where $q(r)$ is a non-linear algebraic or transcendental function [38]. For example,

$$\sin(4r) + (e)^{5r} + 2 = 0$$

or

$$r^4 + 11 = 0.$$

Algebraic Equations

If $q(r)$ is a non-linear algebraic function [39], such as a polynomial equation of degree two or higher, then the non-linear equation $q(r) = 0$ is a non-linear algebraic equation

$$Dr^2 + Er + M = 0,$$

is a 2nd-degree polynomial equation in r.

Transcendental Equations

If the nonlinear equation $q(r) = 0$ [39] contains trigonometric, exponential, logarithmic, or any combination of these functions, it is referred to as a transcendental equations, such as

$$er^r \sin\left(r\right) + r = 0,$$

a transcendental equation, for instance.

3.2 Roots of an Equation

Consider $q(r) = 0$ as a nonlinear equation [40]. When a real number $r = r_t$ solves the equation, it is referred to as the root of equation. For example

$$r^2 - 9 = 0,$$

is an equation having roots 3 and -3.

Distinct Root

When an equation has real roots, its answers or roots are included in the set of real numbers [40]. We contend that if an equation has unique roots, then not all of its solutions or roots are equal. Any quadratic equation with a discriminant $b^2 - 4ac$ larger than has actual and distinct roots .

For example $(r-1)(r-2)(r-5) = 0$ has 3 distinct real roots: $1, 2$, and 5.

Multiple Root

Assume that a real function $q(r)$ and its derivatives are continuous near the point $r = \ell$ and an integer j [40], such that

$$q(\ell) = 0,\ q^{'}(\ell) = 0,\ q^{''}(\ell) = 0, ...,\ q^{(j-1)}(\ell) = 0,$$

and $q^j \neq 0$. If $j = 1$, then ℓ is referred to as the equation's "simple root" and if $j > 1$, then ℓ is referred to as the equation's "repeated or multiple root". For instance

$$s^2 + 2s^5 - 4s^4 + 4s^3 + 7s^2 + 4s - 4 = 0.$$

The roots of equation are $1, 2$ and 4, with the order of root 4 being 3. The order of roots 2 and -2 and -4 is a simple root.

3.3 Iterative Methods

Using initial values [41], the mathematical technique creates an increasing succession of approximations for a class of problems, with each approximation deriving from the one before it, this is called iterative method .

Newton Raphson Method

The most commonly used technique for solving single variable nonlinear equation [41] is defined by:

$$r_{n+1} = r_n - \frac{q(r_n)}{q'(r_n)}, \quad j = 0, 1, ...,$$

where,

$$q'(r_n) \neq 0.$$

Iterative Method With Memory

Methods with memory is based on the use of suitable two-valued functions and the variation of a free parameter in each iterative step [41]. This parameter is calculated using information from the current and previous iteration so that the developed methods may be regarded as methods with memory. For example,

$$r_{n+1} = r_n - q(r_n)\frac{r_n - r_{n-1}}{q(r_n) - q(r_{n-1})}, \quad n = 0, 1, ...$$

with convergence order 1.69 and efficiency index 1.69. The above method is referred as with memory method.

Iterative Method Without Memory

Without-memory methods which are extendable to with-memory methods without insertion of any extra functional evaluation by using Newton's interpolating polynomials have gained attention [41]. These iterative methods offer a choice to achieve higher convergence order and increased efficiency. For example,

$$r_{n+1} = r_n - \frac{q(r_n)}{q'(r_n)}, \quad n = 1, 2, ...$$

with convergence order 2 and efficiency index 1.414, is the well known Newton's method which referred as without memory method.

Self-Accelerating Parameter

Self-accelerating type method is a kind of efficient iterative method with memory for solving nonlinear equations, which chooses some varying parameters as self-accelerating parameters in the iteration processes [16]. The self-accelerating parameter is calculated by using information from previous and current iterations, which does not increase the computational cost of iterative method. Thus, self-accelerating type methods possess a very high computational efficiency. Self-accelerating parameter is very important to self-accelerating type method, which can make a big difference to the efficiency of iterative method. There are two ways to construct the self-accelerating parameter, which are divided difference and interpolation method. Traub's method [2] is one of the most representative methods for self-accelerating type method, which can be written as:

$$\begin{aligned} w_n &= r_n + T_n q(r_n), \\ T_n &= \frac{-1}{q[r_n - r_{n-1}]}. \end{aligned}$$

The parameter T_n is called the self accelerating parameter.

Taylor's Series

Let $q(r)$ be a function [42] having continuos derivative upto order n on $[a, a+h]$ then the Taylor series is defined as:

$$q(a+h) = q(a) + hq'(a) + \frac{h^2}{2!}q''(a) + \cdots + \frac{h^{n-1}}{(n-1)!}q^{(n-1)}(a) + \cdots .$$

This infinite series is called Taylor's series. If we substitute t instead of $a+h$ then it is alternatively given as :

$$q(t) = q(a) + (t-a)q'(a) + \frac{(t-a)^2}{2!}q''(a) + \cdots + \frac{(t-a)^{n-1}}{(n-1)!}q^{n-1}(a) + \cdots .$$

Efficiency Index

To compare different iterative methods, the efficiency index is widely used suggested by Ostrowski [2], in (1960).

$$EI = (\alpha)^{\frac{1}{n}}.$$

where α is the local order of convergence of the method and n represents the number of the evaluations of functions necessary to carry out the method per iteration.

First Order Divided Difference For Scaler Function

Divided differences are used in the replacement of derivative for discrete data [44]. The first order divided difference is denoted as $[r, s; q]$ or $q[r,s]$ and is calculated by the following formula:

$$[r, s; q] = \frac{q(s) - q(r)}{s - r}.$$

Finite Difference Method

In numerical analysis, a numerical formulation known as finite difference method are used to approximate the derivatives when the function is known only on the discrete set of points. The finite difference formulas are of the three types given as:

I. Forward Difference Formula

II. Backward Difference Formula

III. Central Difference Formula

First Derivative Forward Difference Formula Of O(h)

By using expansion of Taylor's series of $q(r)$ [45], about a single point $s = s_i$ with $s_{i+1} = s_i + h$, the first order forward difference formula is given as:

$$q'(s_i) \approx \frac{q_{i+1} - q_i}{h} + O(h^2).$$

First Derivative Backward Difference Formula Of O(h)

By using expansion of Taylor's series $q(r)$ [45], about a single point $s = s_i$ with $s_{i+1} = s_i + h$, the first order backward difference formula is defined as:

$$q'(s_i) \approx \frac{q_i - q_{i-1}}{h} + O(h^2).$$

First Derivative Central Difference Formula Of O(h)

By using expansion of Taylor's series $q(r)$ [45], about a single point $s = s_i$ with $s_{i+1} = s_i + h$ and $s_{i-1} = s_i - h$, the central difference formula is defined as:

$$q'(s_i) \approx \frac{q_{i+1} - q_{i-1}}{2h} + O(h^2).$$

Second Derivative Forward Difference Formula Of O(h)

By using expansion of Taylor's series $q(r)$ [45], about a point $s = s_i$ with $s_{i+1} = s_i + h$ and $s_{i+2} = s_i + 2h$, the second order forward difference formula is defined as:

$$q^{''}(s_i) \approx \frac{2q_{i+1} + q_{i+2} + q_i}{h^2} + O(h^2).$$

Second Derivative Backward Difference Formula Of O(h)

By using expansion of Taylor's series $q(r)$ [45], about a point $s = s_i$ with $s_{i-1} = s_i - h$ and $s_{i-2} = s_i - 2h$, the second order forward difference formula is defined as:

$$q^{''}(s_i) \approx \frac{q_{i-2} - 2q_{i-1} + q_i}{h^2} + O(h^2).$$

Second Derivative Central Difference Formula Of O(h)

By using expansion of Taylor's series of $q(r)$ [45], about a point $s = s_i$ with $s_{i-1} = s_i + h$ and $s_{i+1} = s_i + h$, the second order central difference formula is defined as:

$$q^{''}(s_i) \approx \frac{q_{i+1} - 2q_i + q_{i-1}}{h^2} + O(h^2).$$

3.4 Errors In Computations

There are three types of errors in computations, i.e. ,

(1) Truncation Error,

(2) Round-off Error, and

(3) Inherent Error

Truncation Error

The discrepancy between a function's true value and its truncated value is known as a truncation error [46]. The function's approximate value up to a specified number of digits is the truncated value. For instance, the vacuum speed of light is $2.99792458 \times 10^8 ms^{-1}$.

Round-Off Error

The round-off error is the discrepancy between a number's approximate value and its exact (right) value when it is employed in a calculation [46]. The discrepancy between the true value of an irrational number and the values of rational expressions like $\frac{22}{7}$, $\frac{355}{113}$, 3.14, or 3.14159 serves as an example of round-off error in numerical analysis.

Inherent Error

A program fault that occurs independently of what the user does and is frequently inevitable is known as an inherent error [46].

Let $r = 0.3333$ and $s = 3.1416$ serve as two approximations of the exact values for $\frac{1}{3}$ and. It goes without saying that if we execute an algebraic operation between these two approximations, the error will be introduced proportionally in the outcome.

3.5 Errors Measurement

There are three ways to calculate errors

(1) Absolute Error

(2) Relative Error

(3) Percentage Error

Absolute Error

Let us suppose that r_t be the true real value and r_i be an approximate value [47]. Then, the absolute error is denoted by E_a and is defined by:

$$E_a = |r_{i-}r_t| .$$

Relative Error

Let us consider that r_t be a true real value and r_i be an approximate value [47]. Then, the relative error is denoted by E_r and is defined by:

$$E_r = \frac{|r_i - r_t|}{|r_t|}.$$

Percentage Error

Let us consider that r_t be a exact real value and r_i be an approximate value [47]. Then, the percentage error is denoted by E_p and is defined by:

$$E_p = \frac{|r_i - r_t|}{|r_t|} * 100.$$

Asymptotic Error Constant

The difference between a sequence's term and its limit is what the asymptotic error constant (λ) reveals about the behavior of a sequence's errors [48]. Along with the order of convergence, the asymptotic error constant influences the rate of convergence.

$$\lambda = \lim_{n \to \infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}}.$$

Thus λ is asymptotic error constant and α is the order of convergence .

Newton Interpolating Polynomial

Newton polynomial, named after its inventor Isaac Newton is an interpolation polynomial for a given set of data points [49]. The Newton polynomial is sometimes called Newton's divided differences interpolation polynomial because the coefficients of the polynomial are calculated using Newton's divided differences method.

Order of Convergence

In numerical analysis [31], the order of convergence and the rate of convergence of a convergent sequence are quantities that represent how quickly the sequence approaches its limit .

A sequence $\{r_n\}$ of iterates is of a order of convergence p if

$$er_{n+1} \approx C er_n^P.$$

Error Equation

Let ℓ be a solution of an equation $q(r) = 0$, r_m and r_{m+1} be any two subsequent numerical iteration that are close to the root ℓ, er_m and er_{m+1} be their coincidence errors, i.e $er_m = r_m - \ell$ be the mth step error [31]. Usually the error equation is :

$$er_{m+1} = er_m^p + O(er_m^{p+1}).$$

The given equation describe that the numerical algorithm has order of convergence p.

R-order Of Convergence

Quantities that indicate how rapidly a convergent sequence approaches its limit include the order of convergence and convergence rate [31].

Computational Order Of Convergence (COC)

The computational order of convergence (COC) of a sequence $\{r_n\}$, $n \geq 0$ is defined by [31],

$$p \approx \frac{\ln(|r_{n+1} - r_n| \,/\, |r_{n-1} - r_{n-2}|)}{\ln(|r_n - r_{n-1}| \,/\, |r_{n-1} - r_{n-2}|)},$$

where p is the computational order of convergence.

Linear Convergence

A sequence $\{r_n\}$ is linearly convergent if $p = 1$ [50].

Quadratic Convergent

A sequence $\{r_n\}$ is of quadratic order if $p = 2$ [50].

Stopping Criteria Of Numerical Method

A stopping condition is necessary for a typical iterative method in numerical analysis and scientific computing [51]. A sequence of generated iterations or steps, an error tolerance, and a technique to compute (or estimate) a quantity related to the error are all included in such an algorithm:

$$|r_k - r_{k-1}| \leq \varepsilon^n.$$

Chapter 4

Construction of Methods

In 2020, Wang, X. *et a*l. [31] presented a novel way to construct the self-accelerating parameter and derived a modified Newton method. If the sequence $\{r_n\}$ is generated by newton method, which converges to a simple root r_t of a non linear equation, then the sequence $\{r_n\}$ satisfies the following relation:

$$\lim_{n\to\infty} \frac{r_{n+1} - r_t}{(r_n - r_t)^2} = \lim_{n\to\infty} \frac{er_n + 1}{er_n^2} = c_2$$

where,

$$c_2 = \frac{q''(r_t)}{(2q'(r_t))}$$

is the asymptotic error constant,

$$\begin{aligned} er_n &= r_n - r_t, \\ er_{n+1} &= r_{n+1} - r_t. \end{aligned}$$

Wang, X. *et al.* [31] considered the following scheme:

$$\begin{aligned} s_n &= r_n - \frac{q(r_n)}{q'(r_n)}, \\ r_{n+1} &= s_n - T(s_n - r_n)^2, \end{aligned} \tag{4.1}$$

where $T \in R$ is a self accelerating parameter.

Theorem 2 *Let $q : I \subset R \to R$ be a differential in an open interval I and $r_t \in I$ be a simple zero of q. Then order of convergence of iterative method (4.1) is two and its error equation meets the following equation:*

$$er_{n+1} = (c_2 - T_n)er_n^2 + O(er_n^3), \tag{4.2}$$

where

$$r_n = er_n + r_t, \tag{4.3}$$

$c_2 = \frac{q^2(r_t)}{(2q^{'}(r_t))}$ *and* $T \in R - \{0\}$.

Proof. Let

$$c_i = \frac{q^i(r_t)}{i!q^{'}(r_t)}, i = 2, 3, 4, \ldots$$

Using the Taylor expansion of $q(r)$ around $r = r_t$ and taking $q(r_t) = 0$ into account, we get:

$$\begin{aligned} q(r_n) &= er_n + c_2er_n^2 + c_3er_n^3 + c_4er_n^4 + c_5er_n^5 \\ &\quad +c_6er_n^6 + c_7er_n^7 + O(er_n^8), \end{aligned} \tag{4.4}$$

$$\begin{aligned} q'(r_n) &= 1 + 2c_2 er_n + 3c_3 er_n^2 + 4c_4 er_n^3 + \\ &\quad 5c_5 er_n^4 + 6c_6 er_n^5 + 7c_7 er_n^6 + O(er_n^7). \end{aligned} \tag{4.5}$$

Now taking inverse of $q'(r_n)$

$$\begin{aligned} \frac{1}{q'(r_n)} &= 1 - 2c_2 er_n + (-3c_3 + 4c_2^2)er_n^2 + (-4c_4 + 6c_2c_3 \\ &\quad +2(3c_3 + 4c_2^2)c_2)er_n^3 + O(er_n^4), \end{aligned} \tag{4.6}$$

now from (4.4) and (4.6) we get:

$$\begin{aligned} \frac{q(r_n)}{q'(r_n)} &= er_n - c_2 er_n^2 + (-2c_3 + 2c_2^2)er_n^3 \\ &\quad +(-4c_2^3 + 7c_2c_3 - 3c_4)er_n^4 + O(er_n^5), \end{aligned} \tag{4.7}$$

using Equation (4.3) and (4.7) in (4.1) , we get:

$$s_n = r_n - \frac{q(r_n)}{q'(r_n)} = c_2 er_n^2 + (2c_3 - 2c_2^2)er_n^3 + (4c_2^3 - 7c_2c_3 + 3c_4)er_n^4 + O(er_n^5). \tag{4.8}$$

$$s_n - r_n = -er_n + c_2 er_n^2 + (2c_3 - 2c_2^2)er_n^3 + (3c_4 - 7c_2c_3 + 4c_2^3)er_n^4 + O(er_n^5), \tag{4.9}$$

taking square of (4.9) we get:

$$(s_n - r_n)^2 = er_n^2 - 2c_2 er_n^3 + (5c_2^2 - 4c_3)er_n^4 + O(er_n^5),$$

now using Equation (4.1) and (4.8), we get:

$$\begin{aligned} er_{n+1} &= r_{n+1} - r_t = s_n - r_t - T(s_n - r_n)^2 = (c_2 - T)er_n^2 \\ &\quad +2(c_3 - c_2^2 + c_2 T)er_n^3 + O(er_n^4). \end{aligned}$$

Hence proof is completed. ■

Theorem 3 *Let the self accelerating parameter T_n be calculated by (2.25), (2.26) or (2.27) in the iterative method (2.28), respectively. If r_0 is an initial approximation, which is sufficiently close to a simple root r_t of $q(r)$, then the R-order of convergence of the iterative methods (2.28) is at least* $1+\sqrt{2}\approx 2.414$.

Proof. Let the sequence $\{r_n\}$ be generated by an iterative method, which converges to the root r_t of $q(r)$ with the R-order $OR(IM,a)\geq r$, we obtain:

$$er_{n+1}\sim D_{n,r}er_n^r,\ r_n-r_t=er_n, \tag{4.10}$$

when $n\to\infty$, $D_{n,r}$ tends to the asymptotic error constant in Equation (4.10). Therefore,

$$er_{n+1}\sim D_{n,r}(D_{n-1,r}er_{n-1}^r)^r=D_{n,r}D_{n-1,r}er_{n-1}^{r^2}. \tag{4.11}$$

The error equation of the method (2.28) with memory can be obtained by using Equation (4.2) which satisfies:

$$er_{n+1}=r_{n+1}-a\sim(c_2-T_n)er_n^2+O(er_n^3) \tag{4.12}$$

Substitute $er_n=er_{n-1}$ in r_{n+1} we get:

$$\begin{aligned} er_n &= (-T+c_2)er_{n-1}^2+(2c_3+2Tc_2-2c_2^2)er_{n-1}^3\\ &\quad +(4c_2^3+4Tc_3-5Tc_2^2+3c_4-7c_2c_3)er_{n-1}^4+O(er_{n-1}^5), \end{aligned}$$

substitute $er_n=er_{n-1}$ in s_n we get:

$$s_{n-1}=c_2er_{n-1}^2+(-2c_2^2+2c_3)er_{n-1}^3+(4c_2^3-7c_2c_3+3c_4)er_{n-1}^4+O(er_{n-1}^5), \tag{4.13}$$

$$\begin{aligned} s_{new} &= (c_2^3 - 2c_2^2 T_{n-1} + c_2 T_{n-1}^2) er_{n-1}^4 \\ &\quad + (-8c_2^4 + 20c_2^2 c_3 - 10c_2 c_4 - 6c_3^2 + 4c_5) er_{n-1}^6 + O(er_{n-1}^8), \end{aligned} \tag{4.14}$$

from equation(4.13) and (4.14) we get:

$$\begin{aligned} s_{n-1} - s_{new} &= c_2 er_{n-1}^2 + 2(c_3 - c_2^2) er_{n-1}^3 + (3c_3 + 3c_4 + 2c_2^2 T_{n-1} \\ &\quad - c_2(7c_3 + T_{n-1}^2)) er_{n-1}^4 + O(er_{n-1}^5), \end{aligned} \tag{4.15}$$

$$r_n = (c_2 - T_{n-1}) er_{n-1}^2 + (2c_3 + 2c_2 T_{n-1} - 2c_2^2) er_{n-1}^3 + O(er_{n-1}^4),$$

by simplify (4.15) we get:

$$s_{n-1} - s_{new} = c_2 + (2c_3 - 2c_2^2) er_{n-1} + O(er_{n-1}^2), \tag{4.16}$$

$$\begin{aligned} r_n - r_{n-1} &= -er_{n-1} + (c_2 - T_{n-1}) er_{n-1}^2 \\ &\quad + 2(-c_2^2 + c_3 + c_2 T_{n-1}) er_{n-1}^3 + O(er_{n-1}^4), \end{aligned} \tag{4.17}$$

taking square of (4.17) we get:

$$\begin{aligned} (r_n - r_{n-1})^2 &= -1 + (c_2 - T_{n-1}) er_{n-1} \\ &\quad + (-2c_2^2 + 2c_2 T_{n-1} + 2c_3) er_{n-1}^2 + O(er_{n-1}^3), \end{aligned} \tag{4.18}$$

taking inverse of (4.18)

$$\frac{1}{(r_n - r_{n-1})^2} = 1 + (2c_2 - 2T_{n-1}) er_{n-1} + O(er_{n-1}^2),$$

using Equations (4.17) and (4.16) we get:

$$T_{n1} = \frac{s_{n-1} - s_{new}}{(r_n - r_{n-1})^2} = c_2 + 2(c_3 - c_2 T_{n-1}) er_{n-1} + O(er_{n-1}^2), \tag{4.19}$$

$$c_2 - T_{n1} \sim -2(c_3 - c_2 T_{n-1})er_{n-1} + O(er_{n-1}^2). \tag{4.20}$$

$$s_{n-1} - r_{n-1} = -er_{n-1} + c_2 er_{n-1}^2 + 2(c_3 - c_2^2)er_{n-1}^3 + O(er_{n-1}^4), \tag{4.21}$$

simplify (4.21) we get:

$$(s_{n-1} - r_{n-1})^2 = -1 + c_2 er_{n-1} + (2c_3 - 2c_2^2)er_{n-1}^2 + (3c_4 - 7c_2c_3 + 4c_2^3)er_{n-1}^3 + O(er_{n-1}^4),$$

taking inverse of above equation

$$\frac{1}{(s_{n-1} - r_{n-1})^2} = 1 + 2c_2 er_{n-1} + O(er_{n-1}^2), \tag{4.22}$$

from equations (4.17) and (4.22) we get:

$$T_{n2} = \frac{s_{n-1} - s_{new}}{(s_{n-1} - r_{n-1})^2} = c_2 + 2c_3 er_{n-1} + O(er_{n-1}^2), \tag{4.23}$$

$$c_2 - T_{n2} \sim -2c_3 er_{n-1} + O(er_{n-1}^2). \tag{4.24}$$

Multiply (4.17) with (4.28) *we get* :

$$(s_{n-1} - r_{n-1})(r_n - r_{n-1}) = er_{n-1}^2 + (-2c_2 + T_{n-1})er_{n-1}^3 + O(er_{n-1}^4), \tag{4.25a}$$

simplify (4.25a) we get:

$$(s_{n-1} - r_{n-1})(r_n - r_{n-1}) = 1 + (-2c_2 + T_{n-1})er_{n-1} + O(er_{n-1}^2), \tag{4.26}$$

taking inverse of (4.26) we get:

$$\frac{1}{(s_{n-1} - r_{n-1})(r_n - r_{n-1})} = 1 + (2c_2 - T_{n-1})er_{n-1} + O(er_{n-1}^2), \tag{4.27}$$

from equation (4.16) and (4.27) we have:

$$T_{n3} = \frac{s_{n-1} - s_n}{(s_{n-1} - r_{n-1})(r_n - r_{n-1})} = c_2 + (2c_3 - c_2 T_{n-1})er_{n-1} + O(er_{n-1}^2), \tag{4.28}$$

$$c_2 - T_{n3} \sim -(2c_3 - c_2T_{n-1})er_{n-1} + O(er_{n-1}^2). \tag{4.29}$$

According to Equations (4.12),(4.24),(4.20) and (4.29) we get:

$$er_{n+1} = (c_2 - T_n)er_n^2 \sim -2(c_3 - c_2T_{n-1})D_{n-1,r}^2 er_{n-1}^{2r+1}. \tag{4.30}$$

Comparing exponents of er_{n-1} in relations Equations (4.11) and (4.30), we obtain the following equation:

$$r^2 - 2r - 1 = 0. \tag{4.31}$$

The positive solution of Equation (4.31) is given by $r = 1 + \sqrt{2} \approx 2.414$. Therefore, the R-order of convergence of the method (2.28), when T_{n1} is calculated by (2.25), is atleast $1+\sqrt{2} = 2.414$. From Equations (4.20), (4.24) and (4.29), we can see that T_{n1}, T_{n2}, T_{n3} have the same error level. Thus, the convergence order of iterative method (2.28) with memory is atleast 2.414, when Equations (2.26) and (2.27) is used to compute the parameter T_n, respectively. This completes the proof. ■

As a new Newton technique with memory, the use of a variable self-accelerating parameter is proposed. The Newton method's invariant parameter is replaced by an invariant parameter in a modified Newton methodology without memory that is used to solve nonlinear equations. We develop a new Newton method with memory employing a variable self-accelerating parameter without memory. Convergence order for the innovative Newton technique with memory is $1+\sqrt{2}$. Without performing any more function evaluations, the speeding is achieved. The self-accelerating parameter is developed in a straightforward method, which is the key development.

- In 2020 Wang X. *et al.* [16] constructed the following modified iterative method

$$\begin{aligned} w_n &= r_n + q(r_n), \\ t_n &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \\ s_n &= t_n - T(t_n - r_n)^2, \\ r_{n+1} &= s_n - \frac{q(s_n)}{q[r_n, s_n] + q[s_n, w_n] - q[r_n, w_n]}, \end{aligned} \tag{4.32}$$

where $T \in R$ is a self-accelerating parameter. For method (4.32), we have the following convergence analysis.

Theorem 4 *If function $q : I \subset R \to R$ is sufficiently differentiable and has a simple zero r_t on an open interval I, then iterative method (4.32) is of fourth-order convergence and its error equation is as follows:*

$$er_{n+1} = (c_2 + c_2 q'(r_t) - T)[c_2^2(1 + q'(r_t)) - c_3(1 + q'(r_t) - c_2 T)]er_n^4 + O(er_n^5), \tag{4.33}$$

where $er_n = r_n - r_t, T \in R$ and $c_m = (\frac{1}{m!})\frac{q^m(r_t)}{q'(r_t)}, m = 2, 3, 4, \ldots$

Proof. Using Taylor expansion of $q(r_t)$, we have

$$q(r_n) = q'(r_t)[er_n + c_2 er_n^2 + c_3 er_n^3 + c_4 er_n^4 + c_5 er_n^5 + O(er_n^6)], \tag{4.34}$$

$$er_{n,w} = w_n - r_t = er_n + q(r_n) = 1 + q'(r_t)er_n + c_2 q'(r_t)er_n^2 + (c_3 q'(r_t)er_n^3 + O(er_n^4)),$$

$$\begin{aligned} q(w_n) &= q'r_t)[(1 + q'(r_t))er_n + c_2(1 + 3q^{'}(r_t) + q'(r_t)^2)er_n^2 + \\ &\quad (2c_2^2 q'(r_t)(1 + q'(r_t)) + c_3(1 + 4q^{'}(r_t) \\ &\quad +3q^{'}(r_t)^2 + q'(r_t)^3))er_n^3 + O(er_n^4)], \end{aligned} \tag{4.35}$$

$$
\begin{aligned}
q[r_n, w_n] &= q'(r_t)[1 + c_2(2 + q'(r_t))er_n + c_2^2 q'(r_t) + c_3(3 + 3q^{'}(r_t) \\
&\quad + q'(r_t)^2))er_n^2 + (2 + q'(r_t))(2c_2c_3q'(r_t) \\
&\quad + c_4(2 + 2q^{'}(\alpha r_t + q'(r_t)^2))er_n^3 + O(er_n^4). \qquad (4.36)
\end{aligned}
$$

According to (4.32), (4.34) and (4.36), we get:

$$
\begin{aligned}
er_{n,t} &= t_n - r_t = c_2(1 + q'(r_t))er_n^2 + (-c_2^2(2 + 2q^{'}(r_t) + q'(r_t)^2) \qquad (4.37) \\
&\quad + c_3(2 + 3q^{'}(r_t) + q'(r_t)^2))er_n^3 + O(er_n^4).
\end{aligned}
$$

From (4.32) and (4.37), we obtain:

$$
\begin{aligned}
er_{n,s} &= s_n - r_t = (c_2 + c_2q'(r_t) - T)er_n^2 + (-c_2^2(2 + 2q^{'}(r_t) + q'(r_t)^2) \quad (4.38) \\
&\quad + c_3(2 + 3q^{'}(r_t) + q'(r_t)^2) + 2c_2(1 + q'(r_t))T)er_n^3 + O(er_n^4).
\end{aligned}
$$

By a similar argument to that of (4.34) we get:

$$
\begin{aligned}
q(s_n) &= q'(r_t)(c_2 + c_2q'(r_t) - T)er_n^2 + q'(r_t)(-c_2^2(2 + 2q^{'}(r_t) \qquad (4.39) \\
&\quad + q'(r_t)^2) + c_3(2 + 3q^{'}(r_t) + q'(r_t)^2) + 2c_2(1 + q'(r_t))T)er_n^3 + O(er_n^4).
\end{aligned}
$$

Using (4.34), (4.38) and (4.39) we have:

$$
q[r_n, s_n] = q'(r_t) + c_2q'(r_t)er_n + q'(r_t)(c_3 + c_2^2(1 + q'(r_t)) - c_2T)er_n^2 + O(er_n^3), \quad (4.40)
$$

$$
\begin{aligned}
q[s_n, w_n] &= q'(r_t) + c_2q'(r_t)(1 + q'(r_t)er_n + q'(r_t)(c_3((1 + q'(r_t))^2 \qquad (4.41) \\
&\quad + c_2^2(1 + 2q^{'}(r_t)) - c_2T)er_n^2 + O(er_n^3).
\end{aligned}
$$

Together with (4.32) and (4.38)–(4.41), we obtain the error equation:

$$\begin{aligned} er_{n+1} &= r_{n+1} - r_t = (c_2 + c_2 q'(r_t) - T)[c_2^2(1 + q'(r_t)) \\ &\quad - c_3(1 + q'(r_t)) - c_2 T] er_n^4. \end{aligned} \tag{4.42}$$

The proof is completed. ■

Remark 1. From (4.33), we can see that the convergence order of method (4.32) is at least five provided that $T = c_2(1 + q'(r_t))$. Hence, in order to obtain a new self-accelerating type method, we will use a self-accelerating parameter T_n to replace the parameter T if the parameter T_n satisfies the relation $\lim\limits_{n\to\infty} T_n = T = c_2(1+q'(r_t))$. we can use interpolation method to construct self-accelerating parameter. For example, the self-accelerating parameter T_n can be given by

$$T_n = \frac{N_2^{''}(r_n)}{2N_2^{'}(r_n)}(1 + N_2^{'}(r_n)), \tag{4.43}$$

where $N_2(t) = N_2(t; r_n, r_{n-1}, w_{n-1})$ is Newton's interpolatory polynomial of second degree,

$$N_2^{'}(r_n) = q[r_n, r_{n-1}] + q[r_n, r_{n-1}, w_{n-1}](r_n - r_{n-1})$$

and

$$N_2^{''}(r_n) = 2q[r_n, r_{n-1}, w_{n-1}]$$

Now, we obtain a new self-accelerating type method as follows:

$$\begin{aligned}
w_n &= r_n + q(r_n), \\
t_n &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \\
s_n &= t_n - T_n(t_n - r_n)^2, \\
r_{n+1} &= s_n - \frac{q(s_n)}{q[r_n, s_n] + q[s_n, w_n] - q[r_n, w_n]},
\end{aligned} \tag{4.44}$$

Theorem 5 *Let the varying parameter T_n be calculated by (4.43) in method (4.44). If an initial value r_0 is sufficiently close to a simple zero r of function $q(r)$, then the R-order of convergence of self-accelerating type method (4.32) is at least* $2+\sqrt{5} \approx 4.2361$.

Proof. If an iterative method (IM) generates sequence $\{r_n\}$ that converges to the zero r_t of $q(r)$ with the R-order $O_R(IM, a) \geq r$, then we can write

$$er_{n+1} \sim D_{n,r} er_n^r, \tag{4.45}$$

where $er_n = r_n - r_t$ and the limit of $D_{n,r}$ is the asymptotic error constant of iterative method, as $n \to \infty$ So,

$$er_{n+1} \sim D_{n,r}(D_{n-1,r} er_{n-1}^r)^r = D_{n,r} D_{n-1,r}^r er_{n-1}^{r^2}. \tag{4.46}$$

Similar to (4.46), if the R-order of iterative sequence $\{s_n\}$ is p, then

$$er_{n,s} = D_{n,p} er_n^p \sim D_{n,p}(D_{n-1,r} er_{n-1}^r)^p = D_{n,p} D_{n-1,r}^p er_{n-1}^{rp}. \tag{4.47}$$

According to (4.38) and (4.42), we obtain:

$$er_{n,s} = s_n - r_t \sim (c_2 + c_2 q'(r_t) - T_n) er_n^2 + O(er_n^3), \tag{4.48}$$

$$er_{n+1} = r_{n+1} - r_t \sim (c_2 + c_2 q'(r_t) - T_n)[c_2^2(1+q'(r_t)) - c_3(1+q'(r_t)) - c_2 T_n] er_n^4. \quad (4.49)$$

Here, we omit the higher-order terms in (4.48)–(4.49). Let $N_2(t)$ be the Newton interpolating polynomial of degree two that interpolates the function q at nodes r_n, w_{n-1}, r_{n-1} contained in interval. Then, the error of the Newton interpolation can be expressed as follows:

$$q(t) - N_2(t) = \frac{q^{(3)}(r_t)}{3!}(r - r_n)(r - w_{n-1})(r - r_{n-1}), r_t \in I. \quad (4.50)$$

Differentiating (4.50) at the point $t = r_n$, we get:

$$N_2^{'}(r_n) \sim q'(r_t)(1 - c_3((1 + q'(r_t))er_{n-1}^2 + O(er_{n-1}^3)), \quad (4.51)$$

$$N_2^{''}(r_n) \sim 2q^{'}(r_t)(c_2 + c_3(2 + q'(r_t))er_{n-1} + O(er_{n-1}^2)), \quad (4.52)$$

$$T_n = \frac{N_2^{''}(r_n)}{2N_2^{'}(r_n)}(1 + N_2^{'}(r_n)) \quad (4.53)$$

$$\sim c_2(1 + q'(r_t)) + c_3(1 + q'(r_t))(2 + q'(r_t))er_{n-1} + O(er_{n-1}^2). \quad (4.54)$$

Using (4.48), (4.49) and (4.54) we get:

$$er_{n,s} = (c_2 + c_2 q'(r_t) - T_n)er_n^2$$

$$\sim -c_3(1 + q'(r_t))(2 + q'(r_t))er_{n-1})(D_{n-1,r}^r er_{n-1}^r)^2$$

$$\sim -c_3(1 + q'(r_t))(2 + q'(r_t))(D_{n-1,r}^2 er_{n-1}^{2r+1}). \quad (4.55)$$

$$er_{n+1} = (c_2 + c_2 q'(r_t) - T_n)[c_2^2(1 + q'(r_t)) - c_3(1 + q'(v)) - c_2 T_n]er_n^4$$

$$\sim c_3^2(1 + q'(r_t))^2(2 + q'(r_t))er_{n-1}(D_{n-1,r}er_{n-1}^r)^4$$

$$\sim c_3^2(1+q'(r_t))^2(2+q'(r_t))D_{n-1,r}^4 er_{n-1}^{4r+1}. \tag{4.56}$$

By comparing exponents of er_{n-1} in relations (4.47), (4.55) and (4.46) and (4.56), we have:

$$\begin{aligned} 2r+1 &= rp, \\ 4r+1 &= r^2. \end{aligned} \tag{4.57}$$

Solving system (4.57), we obtain $r = 2+\sqrt{5} \approx 4.2361$ and $p = \sqrt{5} \approx 2.2361$. Therefore, the R-order of method (4.44) is at least 4.2361, when T_n is calculated by (4.53). The proof is completed. ■

- In this chapter, based on Wang, X. and Fan, Q. [16] method with memory a new multi-step iterative method using two self accelerating parameter is proposed. Computational order of convergence (COC) of the proposed method [10] is determined using the following formula:

$$COC \approx \frac{\ln(|r_{n+1}-r_n| \,/\, |r_n - r_{n-1}|)}{\ln(|r_n - r_{n-1}| \,/\, |r_{n-1}-r_{n-2}|)}. \tag{4.58}$$

  The computational convergence order determined using software package Maple 18.0.

It is important to note that very few with memory self accelerating parameters are proposed in the literature we mention here the following.

Zheng *et a*l. [52] introduced following with memory method:

$$\begin{aligned} w_n &= r_n + T_n q(r_n),\ T_n = \frac{1}{q[r_n, r_{n-1}]}, \\ s_n &= r_n - \frac{q(r_n)}{q(r_n, w_n)}, \\ r_{n+1} &= r_n - \frac{q(r_n)^2}{q[r_n, w_n](q(r_n) - q(s_n))}. \end{aligned} \tag{4.59}$$

with convergence order 3.3028.

Wang *et al.* [20], [53], [54] used an interpolating polynomial with self accelerating parameters and introduced the following method:

$$s_n = r_n - \frac{q(r_n)}{T_n q(r_n) + q'(r_n)}, \tag{4.60}$$

$$r_{n+1} = s_n - \frac{q(s_n)}{2T_n q(r_n) + q'(r_n)}(1 + 2\frac{q(s_n)}{q(r_n)} + (\frac{q(s_n)}{q(r_n)})^2), \tag{4.61}$$

where

$$T_n = -\frac{H_2^{''}(r_n)}{q'(r_n)},$$

$$H_2(r_n) = q(r_n) + q'(r_n)(r - r_n) + q[r_n, r_n, s_{n-1}](r - r_n)^2,$$

$$H_2^{''}(r_n) = 2q[r_n, r_n, s_{n-1}].$$

with convergence order 4.562.

Dzunic *et al.* [3], [55] presented the iterative method by using self-accelerating parameter as follows:

$$\begin{aligned} w_n &= r_n + T_n q(r_n), \\ r_{n+1} &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \end{aligned} \tag{4.62}$$

where

$$T_n = -\frac{1}{N_2'(r_n)},$$

based on newton's interpolation polynomial of degree two as follows:

$$N_2(t) = N_2(t; r_n, r_{n-1}, w_{n-1}),$$

and

$$N_2'(t) = q[r_n, r_{n-1}] + q[r_n, r_{n-1}, w_{n-1}](r_n - r_{n-1}).$$

The computational convergence order of method (4.62) is 3.

The main purpose of this chapter is to propose with memory method using self-accelerating parameter.

Petkovic *et al.* [56] gave the following with memory method using self-accelerating type parameter given by:

$$\begin{aligned} w_n &= r_n + T_n q(r_n) \\ s_n &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \\ r_{n+1} &= s_n - \frac{q(s_n)}{q[r_n, w_n]}(1 + \frac{q(s_n)}{q(r_n)} + \frac{q(s_n}{q(w_n)}). \end{aligned} \tag{4.63}$$

Based on Ren's method [8], Wang and Fan proposed modified without memory method of convergence order 4 as follows:

$$\begin{aligned} w_n &= r_n + q(r_n), \\ s_n &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \\ r_{n+1} &= s_n - \frac{q(s_n)}{q[r_n, s_n] + q[s_n, w_n] - q[r_n, w_n] + r_t(s_n - r_n)(s_n - w_n)}, r_t \in \mathbb{R} \end{aligned} \tag{4.64}$$

Mathematicians have worked hard over the past few decades to develop more accurate approximations methods that have higher convergence orders and better efficiency indices. The number of function evaluations per step rises along with the order of convergence. Thus, in order to quantify the equilibrium between these quantities, a new index known as the efficiency index is established.

The use of a variable self-accelerating parameter is suggested as a new Newton technique with memory. In order to solve nonlinear equations, a modified Newton technique without memory with an invariant parameter is first built replacing the Newton method's invariant parameter. By using a variable self-accelerating parameter without memory, we create a new Newton technique that memory. The novel Newton method with memory has a convergence order of $1 + \sqrt{2}$. The quickening without doing any additional function evaluations, is reached. The main advancement is that the self-accelerating parameter is built in a straight forward manner.

4.1 Proposed Methods

We proposed the following (4.66) and (4.67) with memory methods involving two self accelerating parameters:

Proposed method 1 $(NA-1):-$

$$\begin{aligned}
w_n &= r_n + q(r_n), \\
t_n &= r_n - \frac{q(r_n)}{q[r_n, w_n]}, \\
s_n &= t_n - a_n(t_n - r_n)^2, \\
u_n &= s_n - \frac{q(s_n)}{q[r_n, s_n] + q[s_n, w_n] - q[r_n, w_n]}, \\
v_n &= u_n - b_n(u_n - s_n)^2, \\
r_{n+1} &= v_n - \frac{q(v_n)}{q[r_n, v_n] + q[v_n, w_n] - q[r_n, w_n]}.
\end{aligned} \tag{4.65}$$

where the self accelerating parameter a_n and b_n given by

$$\begin{aligned}
a_n &= \frac{t_{n-1} - r_n}{(r_n - r_{n-2})^2}, \\
b_n &= \frac{u_{n-1} - s_n}{(s_n - s_{n-1})^2}.
\end{aligned} \tag{4.66}$$

Proposed method 2 $(NA-2):-$

With another set of following accelerating parameter, a_n and b_n, we have numerical method 2:

$$\begin{aligned}
a_n &= \frac{(t_{n-1} - r_n)(s_{n-1} - r_{n-1})}{(r_n - r_{n-1})^3}, \\
b_n &= \frac{(u_{n-1} - s_n)(v_{n-1} - s_{n-1})}{(s_n - s_{n-1})^3}.
\end{aligned} \tag{4.67}$$

Chapter 5

Numerical Discussion

We compare our methods namely $(NA-1)$ and $(NA-2)$ (4.65) ,(4.66) and (4.67) with ZG(4.59), WG (4.60), PT (4.63) and RN (4.64) by Zheng [52] , Wang *et al.* [16] [20] ,[53], and Petkovic [56], respective to solve the following non-linear equations with initial guess r_0 :

$$
\begin{aligned}
q_1(r) &= \cos r - r,\ r_0 = 0.5 \\
q_2(r) &= 10re^{-r^2} - 1,\ r_0 = 1.8 \\
q_3(r) &= \sin r - \frac{1}{3}r,\ r_0 = 2.0.
\end{aligned}
$$

The numerical results are shown in the tables 1-3. The tables 1-3 represent

$|r_k - r_{k-1}|$ as the absolute error and ∂ as the computational order of convergence.

We take initial guesses $a_0 = 0.1$, $b_0 = 0.1$ as initial values of parameter in the first iteration.

Comparison is given with cited method, on the basis of results of first four iterations. The numerical results are computed using 1200 floating point arithmetic.

The stopping criteria for the iterative process is used as follows :

$$|r_k - r_{k-1}| < 10^{-500}.$$

Comparison of $q_1(r)$

Comparison of $q_1(r)$					
$Method$	$it1$	$it2$	$it3$	$it4$	COC
ZG	0.79077×10^{-3}	0.14418×10^{-11}	0.25525×10^{-40}	0.25811×10^{-135}	3.3039563
WG	0.75634×10^{-3}	0.25811×10^{-15}	0.15116×10^{-67}	0.60634×10^{-289}	4.2386876
RN	0.30201×10^{-4}	0.96552×10^{-20}	0.10086×10^{-81}	0.12011×10^{-329}	4.0000000
PT	0.63702×10^{-3}	0.78493×10^{-16}	0.57495×10^{-70}	0.20389×10^{-299}	4.2384668
$NA-1$	0.14444×10^{-6}	0.27809×10^{-45}	0.76570×10^{-234}	0.0	6.3456
$NA-2$	0.35674×10^{-5}	0.41235×10^{-51}	0.71034×10^{-300}	0.0	6.3456

Table 1

Comparison of $q_2(r)$

Comparison of $q_2(r)$					
$Method$	$it1$	$it2$	$it3$	$it4$	COC
ZG	0.15866×10^{-2}	0.46751×10^{-9}	0.14185×10^{-30}	0.11659×10^{-101}	3.3035264
WG	0.10688×10^{-2}	0.25811×10^{-15}	0.34580×10^{-66}	0.35950×10^{-283}	4.2408357
RN	0.33251×10^{-3}	0.30709×10^{-13}	0.22312×10^{-53}	0.62179×10^{-214}	4.0000000
PT	0.34882×10^{-2}	0.10531×10^{-10}	0.36075×10^{-46}	0.14899×10^{-196}	4.2403195
$NA-1$	0.26205×10^{-6}	0.37213×10^{-41}	0.39421×10^{-256}	0.16796×10^{-498}	6.2035
$NA-2$	0.56731×10^{-4}	0.37153×10^{-38}	0.41760×10^{-386}	0.0	6.3456

Table 2

Comparison of $q_3(r)$

Comparison of $q_3(r)$					
$Method$	$it1$	$it2$	$it3$	$it4$	COC
ZG	0.51016×10^{-2}	0.2611×10^{-8}	0.52455×10^{-29}	0.21691×10^{-97}	3.3040229
WG	0.84480×10^{-2}	0.96292×10^{-11}	0.89256×10^{-48}	0.74204×10^{-205}	4.2416331
RN	0.14664×10^{-4}	0.12289×10^{-23}	0.60662×10^{-100}	0.36019×10^{-405}	4.0000000
PT	0.90035×10^{-2}	0.25682×10^{-10}	0.87635×10^{-46}	0.33663×10^{-196}	4.2410060
$NA-1$	0.10623×10^{-8}	0.28826×10^{-59}	0.29681×10^{-372}	0.0	6.5073
$NA-2$	0.28026×10^{-7}	0.31456×10^{-72}	0.43127×10^{-350}	0.0	6.3741

Table 3

Tables 1-3 showed that our methods have excellent behaviors as compare to existing method shown in the table.

5.1 Discussion

Two with memory methods $(NA-1)$ and $(NA-2)$ involving two self-accelerating parameters for solving non-linear equations based on Wang and Fan [16] methods are constructed. Tables 1-3 show that our methods have excellent behavior as compared to existing methods shown in the tables.

Consider iterative strategies to approximate both the single root and all of the roots in order to identify the roots and examine both varieties of iterative systems in this method. There are numerous iterative techniques with various orders of convergence that can be used in the literature to approximate the roots of Ostrowski determined the efficiency index I of these iterative approaches as $I = \frac{k}{u}$ where k is the number of function evaluations per iteration and u is the order of convergence.

Chapter 6

Concluding Remark

In this dissertation, two with memory methods $(NA-1)$ and $(NA-2)$ involving two self-accelerating parameters for solving non-linear equations based on Wang and Fan [16] methods are constructed.

6.1 Conclusions

Derivative free methods are the only option when the derivatives are difficult to compute or the derivatives are zero at certain points. There are so far very few methods with memory involving self accelerating parameters existing in the literature. Derivative free and with memory iterative methods involving self-accelerating parameter have proposed for the solution of distinct roots of single variable nonlinear equations. The technique of obtaining self-accelerating parameter is based on forth order iterative methods developed by Wang and Fan [16]. The analysis of numerical methods show that with memory methods have stable behavior and fast convergence as compared to without memory methods using the new methods $(NA-1)$ and $(NA-2)$. The proposed method shows better convergence behavior, efficiency and accuracy.

6.2 Future Direction

This thesis aims to explore the future direction of distinct roots of nonlinear equations using modified root finders in Numerical analysis.

A more derivative free method, with memory involving accelerating parameters with better convergence order for finding the distinct roots of non-linear equations, can be constructed. The self-accelerating parameter may be improved for getting higher convergence of iterative methods.

References

[1] Neta, B. (2020). Basin attractors for derivative-free methods to find simple roots of nonlinear equations. *Journal of Numerical Analysis and Approximation Theory*, 49(2), 177–189.

[2] Traub, J.F. (1964). Iterative Method for the Solution of Equations. *American Mathematics Society*, 1-99.

[3] Džunic, J. Petkovic, M.S. Petkovic, L.D. (2012). Three-point methods with and without memory for solving nonlinear equations. *Applied Mathematics and Computation*, 218, 4917–4927.

[4] Lotfi, T., Assari, P. (2015). New three- and four-parametric iterative with memory methods with efficiency index near 2. *Applied Mathematics and Computation*, 270, 1004-1010.

[5] Cordero, A., Jordan, C., Torregroda, and J. R. (2017). A dynamical comparison between iterative methods with memory: Are the derivative good for the memory. *Computational and Applied Mathematics,* 318, 335-347.

[6] Campos, B., Cordero, A.,Torregrosa, J.R., Vindel, P. (2015). A multi dimensional dynamical approach to iterative methods with memory. *Applied Mathematics and Computation*, 271, 701-715.

[7] Ullah, M. Z., Kosari, M., Soleylmani, S., Haghani F. K. and Al-Fhaid, A.S. (2016). A super-fast tri-parametric iterative mehod with memory. *Applied Mathematics and Computation*, 289, 486–491.

[8] Ren, H., Wu, Q., Bi, W. (2009). A class of two-step Steffensen type methods with fourth-order convergence. *Applied Mathematics and Computation,* 209, 206–210.

[9] Weerakoon, S., and Fernando, T. G. I. (2000). A Variant of Newton's Method with Accelerated Third-Order Convergence. *Applied Mathematics Letters*, 13, 87-93.

[10] Cordero, A., Torregrosa, J.R. (2007). Variants of Newton's Method using fifth-order quadrature formulas. *Applied Mathematics and Computation*, 190(1), 686-698.

[11] Chun, C. (2006). Construction of Newton-like iteration methods for solving nonlinear equations. *Numerical Mathematics*, 104, 297–315.

[12] Abbas bandy, S. (2006). Modified homotopy perturbation method for nonlinear equations and comparison with Adomian decomposition method. *Applied Mathematics and Computation*, 172, 431-438.

[13] Noor, M. A., Noor, K. I. and Waseem, M. (2014). A new decomposition technique for solving a system of linear equations. *Journal of the Association of Arab Universities for Basic and Applied Sciences*, 16, 27-33.

[14] Wang, X. and Zhang, T. (2015). Efficient n-point iterative methods with memory for solving nonlinear equations. *Numerical Algorithms*, 70, 357–375.

[15] Barto, L., Poli, A., and Sacco, E. (2005). A degree reduction based technique for finding all univariate polynomial roots. *Journal of complexity*, 21(1), 72-79.

[16] Wang, X. and Fan, Q. (2020). A Modified Ren's Method with Memory Using a Simple Self-Accelerating Parameter. *Mathematics*, 8(4), 540.

[17] Noor, M.A., Noor, K. I., Al-Said, E. and Waseem, M. (2010). Some New Iterative Methods for Nonlinear Equations. *Mathematical Problems in Engineering*, 2010, 12.

[18] Wang, X. and Liu, L. (2010). Modified Ostrowski's method with eighth-order convergence and high efficiency index. *Applied Mathematics Letters*, 23, 549-554.

[19] Dehghan, M. & Hajarian, M. (2010). New iterative method for solving nonlinear equations with fourth-order convergence. *International Journal of Computer Mathematics*, 87(4), 834-839.

[20] Wang, X., Zhang, T. (2014). A new family of Newton-type iterative methods with and without memory for solving nonlinear equations. *Calcolo*, 51, 1–15.

[21] Dzunic, J. (2013). Modified Newton's method with memory. *Mathematics and Informatics*, 28, 429–441.

[22] Jaiswal, J.P. (2014). A class of iterative methods for solving nonlinear equation with fourth-order convergence. *Universal Journal of Applied Mathematics*, 2(8), 283-289.

[23] Eftekhari, T. (2014). On Some Iterative Methods with Memory and High Efficiency Index for Solving Nonlinear Equations. *International Journal of Differential Equations*, 2014, 6 pages.

[24] Maroju, P., Behl, R. and Motsa, S.S. (2019). Some novel and optimal families of King's method with eighth and sixteenth-order of convergence. *Journal of Computational and Applied Mathematics*, 318, 136–.148.

[25] Li, S., Liu, X. and Zhang X. (2019). A Few Iterative Methods by Using [1, n]-Order Padé Approximation of Function and the Improvements. *Journal of Applied Mathematics*, 7(1), 55.

[26] Mir, N. A., Shams M., Rafiq, N., Shams M. and Akram, S. (2019). Derivative Free Iterative Simultaneous Method For Finding Distinct Root Of Non-linear Equation. *International Journal of Sciences and Research*, 75, 1-10.

[27] Chand, B.P., Chicharro F.I., Garrido, N. and Jain P. (2019). Design and Complex

Dynamics of Potra–Pták-Type Optimal Methods for Solving Nonlinear Equations and Its Applications. *Mathematics*, 7(10), 942.

[28] Khdhr, F.W., Saeed, R.K. and Soleymani, F. (2019). Improving the Computational Efficiency of a Variant of Steffensen's Method for Nonlinear Equations. *Mathematics*, 7(3), 306.

[29] Barrada, M., Ouaissa, M., Rhazali, M. and Ouaissa, M. (2020). A New Class of Halley's Method with Third-Order Convergence for Solving Nonlinear Equations. *Journal of Applied Mathematics*, 2020, 13.

[30] Barrada, M., Hasnaoui, M. L., and Ouaissa, M. (2020). On the Global Convergence of Improved Halley's Method. *Engineering Letters*, 28, 7.

[31] Wang, X. and Tao, Y. (2020). A new Newton method with memory for solving nonliear equations. *Mathematics*, 8(1), 108.

[32] Rafiq, N., Akram, S., Mir, N.A. and Shams, M. (2020). Study of Dynamical Behavior and Stability of Iterative Methods for Nonlinear Equation with Applications in Engineering. *Mathematical Problem in Engineering*, 2020, 20.

[33] Chu, Y., Rafiq, N., Shams, M., Akram, S., Mir, N.A. and Kalsoom, H. (2020). Computer Methodologies for the Comparison of Some Efficient Derivative Free Simultaneous Iterative Methods for Finding Roots of Non-Linear Equations. *Computers, Materials & Continua*. 66(1), 275-290.

[34] Lotfi, T., Sharifi, S., Salimi, M., and Siegmund, S. (2015). A new class of three-point methods with optimal convergence order eight and its dynamics. *Numerical Algorithms,* 68(2), 261–288.

[35] Liu, CS., Hong, HK., Lee, T.L. (2021). A splitting method to solve a single non-linear equation with derivative-free iterative schemes. *Mathematics and Computers in Simulation*, 190, 837–847.

[36] Neta, B. (2021). A New Derivative-Free Method to Solve Nonlinear Equations. *Journal of Applied Mathematics*, 9(6), 583.

[37] Thota, B., & Shanmugasundaram, P. (2022). On new sixth and seventh order iterative methods for solving non-linear equations using homotopy perturbation technique. *BMC Research Notes*, 15, 1-15.

[38] Kelley, C. T. (1995). Iterative Methods for Linear and Nonlinear Equations. United States: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

[39] Householder, A. S. (1966). The Numerical Solution of Algebraic and Transcendental Equations. (n.p.): (n.p.).

[40] Hoffman, J. D. (2001). Numerical Methods for Engineers and Scientists, Second Edition,. Germany: Taylor & Francis.

[41] Lozenetz and Bulgaria. (2019). Finite Difference Methods. Theory and Applications: 7th International Conference, FDM 2018, Revised Selected Papers. Germany: Springer International Publishing.

[42] Karris, S. T. (2004). Numerical Analysis Using MATLAB and Spreadsheets. Ukraine: Orchard Publications.

[43] Grau-Sanchez, M., Noguera, M. and Diaz-Barrero, J. L. (2015). Note on the efficiency of some iterative methods for solving nonlinear equations. *SeMA Journal* ,71, 15–22.

[44] Cordero, A., Soleymani, F., Torregrosa J.R. (2019). Iterative Methods for Solving Nonlinear Equations and Systems. Switzerland: MDPI AG.

[45] Sharma, A. (2021). Introduction to Computational Fluid Dynamics: Development, Application and Analysis. Switzerland: Springer International Publishing.

[46] Babu, R. (2010). Engineering Mathematics-I (For Wbut). India: Pearson Education.

[47] Peterson, J. C., Smith, R. D. (2012). Introductory Technical Mathematics. United States: Cengage Learning.

[48] Gautschi, W. (2011). Numerical Analysis. United States: Birkhäuser Boston.

[49] Gupta, S. K. (1995). Numerical Methods for Engineers. India: New Age International (P) Limited.

[50] Bulirsch, R., Stoer, J. (2002). Introduction to Numerical Analysis. Germany: Springer.

[51] Butt. (2009). Introduction to Numerical Analysis Using MATLAB®. United States: Jones & Bartlett Learning.

[52] Zheng, Q., Zhao, P., Zhang, L. and Ma,W. (2010). Variants of Steffensen-secant method and applications. *Applied Mathematics and Computation*, 216, 3486–3496.

[53] Wang, X., Zhang, T. (2014). Some Newton-type iterative methods with and without memory for solving nonlinear equations. *International Journal of Computation Methods*, 11(5), 20.

[54] Wang, X., Qin, Y., Qian, W., Zhang, S. and Fan, X. (2015). A Family of Newton Type Iterative Methods for Solving onlinear Equations. *Algorithms*, 8(3), 786–798.

[55] Džunic, J. Petkovic, M.S. (2015). On generalized multipoint root-solvers with memory. *International Journal of Applied Mathematics*, 236, 2909–2920.

[56] Petkovic, M.S. Ilic, S. Džunic, J. (2010). Derivative free two-point methods with and without memory for solving nonlinear equations. *Applied Mathematics and Computation*, 217, 1887–1895.