HIGHER ORDER MULTIPOINT METHOD FOR SOLVING NON-LINEAR EQUATIONS

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

Master of Science

in

Mathematics and Computing

Submitted by

ISHTPREET KAUR
Roll No. 301703013

Under the guidance of

Dr. Munish kansal

July 2019
School of Mathematics

Thapar Institute of Engineering and Technology
Patiala-147004
Punjab, India

CERTIFICATE

This is to certify that the thesis entitled **”HIGHER ORDER MULTIPOINT METHODS FOR SOLVING NONLINEAR EQUATIONS”** in partial fulfillment of the requirements for the award of the degree of the Masters of Science in Mathematics and Computing to the School of the Mathematics, Thapar Institute of Engineering and Technology, Patiala is a record of my work studied under the supervision of **Dr. Munish Kansal.**

The matter embedded in this thesis has not been submitted by me for the award of any other degree of this or any other university/institute.

Ishtpreet Kaur
(Ishtpreet Kaur)
Reg.No. 301703013

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

Munish Kansal

Dr. Munish Kansal
Assistant Professor
Thapar Institute of Engineering and Techology
Patiala

Acknowledgement

It is my genuine pleasure to express my deep sense of thanks and gratitude to my teachers and supervisor, Dr. Munish Kansal, Assistant Professor, School of Mathematics, Thapar Institute of Engineering and Technology, Patiala. His immense interest, motivation and encouragement helped me throughout my work. It was usually a fortunate experience to work under him and enrich from his vast knowledge. Finally, i would like to thank all those who knowingly and unknowingly helped me all throughout this period. I would like to express my sincere thanks to **Dr. Satish Kumar Sharma**, Associate Professor, Head, SOM and to entire faculty and staff members of School of Mathematics for their help, cooperation, love and affection. My sincere heartfelt gratitude to my family whose prayers, best wishes and encouragement has been constant source of insipiration. Nevertheless, i will always be grateful to my friends and batchmates for their unconditional love and care.

Date: 05/09/2019

Place Patiala

Ishtpreet Kaur

Ishtpreet Kaur

Reg.No. 301703013

Contents

ABSTRACT

The present thesis has been divided into three chapters including the introduction. In chapter one, we include basic results and definitions related to the iterative methods to solve nonlinear equations in the single variable. In chapter two, we reviewed a paper concerning third order iterative methods for multiple roots of nonlinear equations. Several numerical problems arising from real life applications have been included in the chapter to show the efficiency of the proposed scheme. In chapter three, a wide general class of optimal eighth-order methods for multiple zeros with known multiplicity is brought forward, which is based on weight function technique involving function-to-function ratio. An extensive convergence analysis is demonstrated to establish the eighth-order of the developed methods. The numerical experiments considered the superiority of the new methods for solving concrete variety of real life problems coming from different disciplines such as trajectory of an electron in the air gap between two parallel plates, the fractional conversion in a chemical reactor, continuous stirred tank reactor problem, Planck's radiation law problem, which calculates the energy density within an isothermal blackbody and the problem arising from global carbon dioxide model in ocean chemistry, in comparison with methods of similar characteristics appeared in the literature.

Chapter 1

Introduction

1.1 Overview

Computational mathematics is the component of applied mathematics that creates and analyses algorithms, numerical techniques or methods and symbolic computations for the solutions of problems related with mathematics generally from science and engineering. There are a few computational viewpoints in various areas like numerical methods, fluid dynamics, optimization, number theory, differential equations and other functional problems. One of the vital and entertaining work in computational arithmatic deals with finding efficiently and precisely the numerical solutions for nonlinear problems.

It is possible to discover the roots of the nonlinear equations, when the degree of the polynomial is atmost four. Otherwise, in common, there is no closed form solution available for solving polynomial equation having degree greater than four or transcendental equations which includes trigonometric functions and logarithmic functions. Subsesquently, we turn towards the numerical techniques for finding the approximate solutions of nonlinear equations of the form $f(x) = 0$ where $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ be a non-differentiable function defined on semi open interval D.

1.2 Bracketing and Open methods

1.2.1 Bracketing methods

Bracketing methods are also known as globally convergent methods. These methods rely upon finding an interval [a,b] such that $f(a)$ and $f(b)$ are of opposite signs. When interval has been found, regardless of how large, iterations will continue until we either find the required root or the interval which contains the root. Bracketing methods are always convergent but their rate of convergence is generally slow. The most well-known examples of bracketing methods are bisection method and regula-falsi method.

Bisection method

Bisection method is a root finding method that again and again cuts an interval and afterward it chooses a subinterval in which a root must lie for further processing. This is the technique based on Intermediate Value Property which expresses that if $f(x)$ is continuous in the interval [a,b] and $f(a), f(b)$ have opposite signs, then $f(x)$ has atleast one root that lies between $x = a$ and $x = b$. Order of convergence is linear. It is generally slow method. Because of this, it is oftenly used for acquiring a rough aproximation to a solution which is then utilized as a starting point for even more rapidly converging strategies.

Method of False position or Regula-falsi method

It rely only on choice of terminal points of interval [a,b]. The function $f(x)$ does not have any role in determining the point c that is only a mid point of a and b. It is utilized distinctly to choose the next smaller interval [a,c] and [c,b]. A better approximation to c is gotten by assuming the straight line L which joins the points $(a, (f(a))$ and $(b, (f(b))$ intersecting $x - axis$. The method of false position is the oldest method. It remains an effective alternative to the bisection method for solving nonlinear equation of the form $f(x) = 0$ for a real root in the interval (a,b) given that f is continuous on the interval [a,b] and $f(a)$ and $f(b)$ have opposite signs.

Like bisection method, the Regula-falsi method has convergence and it may converge to the root faster than the bisection method. It is clear that the regula-falsi method is linear convergent i.e. order of convergence of this method is linear.

1.2.2 Open methods

Open methods are different from bracketing methods. These methods need only a single starting value or sometimes it aquires two starting values which do not essentially bracket a root. As the computation proceeds, open methods may diverge. Open methods have no guaranteed convergence. But, when they do converge, then their convergence is faster than bracketing methods. In particular, the convergence of these open methods firmly relies on the decision of initial approximation and the nature of the function. These techniques are additionally known as locally convergent methods. Secant method and Newton method are the well-known instances of open methods.

Newton's method

Newton's method is a strategy used for discovering successively better approximations to the roots of the function. It is a method that helps us to find one of the zeroes of the function. This method is derived from the Taylor's series extension of the function $f(x)$ about x_1 point:

$$f(x) = f(x_1) + (x - x_1)f'(x_1) + \frac{(x - x_1)^2 f''(x_1)}{2!} + ...$$

By taking the first two terms of the Taylor's series, we get

$$f(x) = f(x_1) + (x - x_1)f'(x_1)$$

On setting $f(x) = 0$, we get

$$f(x_1) + (x - x_1)f'(x_1) = 0$$

After rearranging the above equation, we obtain the next approximation, i.e.

$$x = x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Generally, starting with the approximation x_n, the next approximation is given by

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

Here, we assume that x_n is well defined, if we proceed

$$h(x) = x - \frac{f(x)}{f'(x)},$$

then, Newton's method is special case of fixed point method.

Secant method

This strategy is an improvement over the technique for Regula falsi as does not need the condition $f(x_0)f(x_i) < 0$. Secant method resembles Regula-falsi method except that no attempt is made to ensure that the root ξ is enclosed between two approximations. Starting with two initial approximations x_0 and x_1 to the root ξ, the further approximations $x_1, x_2, x_3, ...$ are computed using the following formula:

$$x_{k+1} = \frac{x_{k-1}f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})} \qquad k = 1, 2, 3...,$$

or

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \qquad k = 1, 2, 3....,$$

Relation between Secant method and Newton method

Consider the equation of line joining (x_0, y_0) and (x_1, y_1) as follows:

$$\frac{y - y_1}{x - x_1} = \frac{y_1 - y_0}{x_1 - x_0}$$

On taking

$$y = 0$$

in above equation, we get

$$\frac{-y_1}{x - x_1} = \frac{y_1 - y_0}{x_1 - x_0}$$

$$(x - x_1)(y_1 - y_0) = -y_1(x_1 - x_0)$$

$$(x - x_1)\frac{(y_1 - y_0)}{(x_1 - x_0)} = -y_1$$

$$(x - x_1) = \frac{-(x_1 - x_0)y_1}{y_1 - y_0}$$

As

$$y = f(x)$$

So, it becomes,

$$x_2 = x_1 - \frac{(x_1 - x_0)g(x_1)}{f(x_1) - g(x_0)}$$

Similarly,

$$x_3 = x_2 - \frac{(x_2 - x_1)f(x_2)}{f(x_2) - f(x_1)}$$

In general, we have

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})} \ ; n > 1$$

1.2.3 Fundamental concepts and Some definitions

Fixed-point iteration

For a given function $g(x)$, A number 's' is known as fixed point of a given function $g(x)$ if $g(s) = s$. In some cases, the fixed-point problems are easier to analyze,and fixed-point algorithms frequently lead to exceptionally powerful root-finding techniques. Basically, root-finding problems and fixed-point problems are comparable in the following sense: We have root discovering problem $f(x) = 0$, one can determine function 'g' with a fixed-point at 's' as $g(x) = x - f(x)$. Conversely, if the function 'g' has a fixed point at 's', then the function defined by $f(x) = x - g(x)$ has a zero at 's'.

Fixed-point iteration is given by $x_{n+1} = g(x_n)$ where 'g' is known as iteration function. Sufficient conditions for existence and uniqueness of a fixed-point are given by the following theorems:

Theorem 1 *If $\xi \in C[a, b]$ and $\xi(x) \in [a, b]$ for all $x \in [a, b]$ then ξ has atleast one fixed-point in [a,b].*

Theorem 2 *If in addition, ξ' exists on the interval (a,b) and a positive constant $k < 1$ exists with $|\xi'(x)| < k$, for all $x \in (a, b)$ then there is exactly one fixed point in $[a, b]$*

Order of Convergence

Let the sequence x_n converges to the root ξ of an equation $f(x) = 0$. At that point, the sequence is said to converge with the order of convergence p if there is a constant $c > 0$, i.e.

$$\lim_{n\to\infty} \frac{|x_{n+1} - \xi|}{|x_n - \xi|^p} = c,$$

where, c denotes asymptotic error constant. In this, if $p = 1$ and $0 < c < 1$, then the sequence is said to have linear convergence. When $p = 2$ and $p = 3$, the sequence is known as quadratic and cubic convergence.

Error equation

Let $e_n = x_n - \xi$ denotes the error of estimation in the n^{th} iterative step. Then the Error equation is

$$e_{n+1}{=}ce_n^p + O(e_n^{p+1}).$$

On obtaining the error equation from convergence analysis of any iterative method, then c denotes the asymptotic error constant and the smallest exponent p of e_n on right side denotes the convergence order.

Computational Efficiency

Computational efficiency depends upon the speed of convergence (order), cost of evaluating f and its derivatives (problem cost), and the cost of building the iterative procedure. Let d denotes the functional evaluations per step. As indicated by Traub, informational efficiency or coefficient of efficiency of iterative strategy is proportional to its order and inversely proportional to the informational usage. In this way,

$$E = \frac{p}{d}$$

Ostrowski gives an alternative definition of efficiency index of an iterative method as:

$$E = p^{1/d}$$

Kung Traub's Conjecture

Kung Traub Conjecture expresses that the order of convergence of any multipoint itaerative strategy without memory which requires n practical evalutions per iteration, cannot go beyond the bound 2^{n-1}, known as optimal order. For instance, the optimal convergence order for a method without memory consuming three functional evaluations per step would be 4. Multipoint iterative methods without memory that requires n+1 functional evaluations per iteration, have at most $2n$ order of convergence. Multipoint methods that fulfill the Kung Traub Conjecture are generally known as Optimal methods. Thus, the optimal order is r=2^n. As indicated by Kung Traub Conjecture and the optimal efficiency index is

$$E_n^{(o)} = 2^{n/(n+1)}$$

1.2.4 Initial approximations to a root

It is well known that most of the one-point and higher-order multipoint methods have been created by keeping in mind Newton's method as the base step. Consequently, good convergence of newton like strategies cannot be described if an initial guess is not near enough to the ideal root. However, finding a reasonably good initial approximation that ensures the convergence is a very tough and non-trivial task. However, the bisection method and its modifications are the easiest techniques for finding the initial approximations.

Graphic approach

In general, graphical techniques are commonly used to get initial approximations to the required root. In this methodology, we plot the graph of functipn $f(x)$ using any computer system and note the purpose of point of intersection with the $x-axis$. It gives the approximate location of the root and these approximations can be utilized as the initial values in root-finding techniques. At that point, they might be refined further by utilizing higher-order numerical procedures to accomplish the required accuracy. However, this method won't be appropriate for determining the initial approximation in multidimensional case.

Trial and error approach

This procedure comprises of assuming a value $x = x_0$ and examine even if the function $f(x)$ is sufficiently near to zero. If this is not true, then one can substitute another supposition and $f(x)$ is evaluated again to decide if the new value gives us superior estimate of the root. This procedure is proceeded until a guess is achieved that conclude $f(x)$ that is near zero.

Incremental search approach

In this methodology, we can begin at one side of a region. We then compute function values at small increments over the region. It is guaranteed by the Intermediate value property, that a root lies within an increment only, when at that point, the function changes sign in an increment. Any value of x inside that increment can be indicated as an initial guess.

1.3 Multipoint methods

Multipoint methods are special class of iterative techniques that have or possess a greater computational productivity than one-point techniques. Multipoint techniques generate approximations of higher accuracy, the rapid improvement of advanced computer mathematics and symbolic calculation have permitted an even more efficient implementation of multipoint techniques.

1.3.1 Classiffication of Iterative Methods

One-point iterative method without memory

An iterative method is known as one-point iterative without memory, if it requires only new information at x_n, for evaluating the next approximation x_{n+1} to the required root and no old data is reused. Thus,

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

defines a one-point iteration without memory. As indicated by Traub, to construct a one-point strategy of order p, one has to utilize all derivatives upto order $p-1$. Newton's method is the most commonly used one point iterative method.

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)},$$

which is known as Newton's method. Chebyshev's method, Halley's method, super-Halley strategy are likewise outstanding instances of one-point strategies without memory.

One-point iterative method with memory

An iterative technique is known as a one-point iterative method with memory, when the approximation x_{n+1} of required root can be determined by using new data at x_n and previous data at approximations $x_{n-1}.......x_{n-s}$. The mapping

$$x_{n+1} = \phi(x_n; x_{n-1}............x_{n-s})$$

defines a one point intration with memory.

Multipoint iterative method without memory

An iterative technique is known as a multipoint iterative strategy without memory, if approximation x_{n+1} of the required root can be resolved distinctly by new data at number of points and no old data is used. In different way, x_{n+1} is discovered by new data at $x_n w_1(x_n), w_2(x_n......w_k(x_n), k > 1$ and furthermore, no old data is reused. Along these lines, the mappings

$$x_{n+1} = \phi(x_n w_1, (x_n) w_2(x_n), ...w_k(x_n))$$

characterizes a multipoint iteration without memory. Traub Steffensen's method is the example of multipoint iteration without memory.

$$x_{n+1} = x_n - \frac{\gamma f(x_n)^2}{f(w_1) - f(x_n)},$$

where, $w_1(x_n) = x_n + \gamma f(x_n)$ and $\gamma \in \mathbb{R} \setminus \{0\}$

Multipoint iterative method with memory

If an estimation of root is resolved by new data at number of points with reusing the old data at different points, at that point iterative strategy is called a multipoint iterative technique with memory. Mathematically, let z_j denotes the $k+1$ quantities $x_j, w_1(x_j), w_2(x_j), ...w_k(x_j)$, $k > 1$. Thus, the mapping

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

defines a multipoint iteration with memory. Specifically, in every iterative step we should keep data of the last n approximations x_j, and for each approximation we should compute n expressions $w_1(x_j), ...w_n(x_j)$ Traub was the first mathematician who proposed a multipoint method with memory. The iterative expression for multipoint method with memory is given by

$$\gamma_n = \frac{x_n - x_n - 1}{g(x_n) - g(x_n - 1)}, n > 1 \qquad \gamma_0 isgiven$$

$$x_{n+1} = x_n - \frac{\gamma_n g(x_n)^2}{g(x_n + \gamma_n g(x_n)) - g(x_n)}$$

1.3.2 Literature survey

One of the most challenging and appealing issues in numerical techniques is to finding approximate soliutions of nonlinear equations. A few specialists have demonstrated their enthusiasm for the exploration and development of numerical analysis to deal with these kind of problems. Direct or analytical methods for finding the solutions of nonlinear equations address just a minor class of problems. These problems do not appear to be very practical in most cases. Thus, Iterative methods are observed to be the most appropriate for computing nonlinear equations. The principal of these methods comprises in starting from atleast one initial approximations to the sought root and gradually, we get a sequence of successive iterations.

Each iterative method has its own advantages as well as disadvantages, and subsequently the selection for finding the "best algorithm" is very complicated task. A valid iteration must possess good convergence properties with respect to the solutions of the given problem. There are numerous significant properties such as convergence speed, numerical stability, computational efficiency, etc.that needs special focus in numerical analysis. According to our view point, these are a section of the real explanations behind further improvement and analysis of numerical strategies.

Iterative methods for simple roots

Regula-falsi method and bisection method are the easiest instances of iterative strategies used for finding easier roots of nonlinear equations numerically. Due to their slow speed of convergence, these strategies are not approved when accuracy is essential in multi-precision digits. Another limitation of bisection method and regula-falsi method is that they are not efficient for finding multiple roots having even multiplicity. Subsequently, one likes to utilize locally convergent strategies which converge more quickly than globally convergent methods. These days, hybrid algorithms are used in practice that starts from a globally convergent method and change to locally convergent method when approximations draw nearer to the required root. There are a wide range of methods to construct computationally effecient root-finding strategies. A few iterative strategies have been built using geometrical approach, quadrature approach, functional approach, Adomian decomposition approach, inverse interpolation approach, testing approach, compositional methodology, weight function approach etc.

For example, the geometric development of Newton's method for solving nonlinear equations is most common as it comprises in finding the points of intersection of the tangent line to the curve $y = f(x)$ with the x-axis. It is possibly one of the most prominent and broadly utilized algorithm for solving such problems. It requires just two functional evaluations and its efficiency index

$$E = \sqrt{2} \approx 1.414$$

There are many drawbacks of Newton's method when executed in practicable problems. Firstly, Newton's method requires the calculation of first-order derivatives at each iteration. To eliminate this type of problem, few researchers have introduced the idea of eliminating derivatives from iterations. Specifically, when the first-order derivative $f'(x_n)$

in Newton's technique is reestablished by forward-difference estimation $\frac{g(x_n+g(x_n))-g(x_n)}{g(x_n)}$. One gets the well-known Steffensen's method,

$$x_{n+1} = x_n - \frac{g^2(x_n)}{g(x_n + g(x_n)) - g(x_n)}$$

Now, Steffensen's method is derivative free, however, in actuality. Both techniques have quadratic convergence by utilizing just two functional evaluations per full step.

Iterative methods for multiple roots

This area handles the problem of finding multiple zeroes of a nonlinear equation $f(x)$. The iterative techniques for simple roots are either not efficient or their convergence rate lowers to linear in case of multiple roots. Consequently, it is imperative to establish new iterative calculations or adjust current ones for discovering multiple roots. There are essentially two sorts of problems that deals with the multiple zeroes, in particular
(i) when the multiplicity $'m'$ of the multiple root is known explicitly.
(ii) when the multiplicity $'m'$ of the multiple root is unknown explicitly. In this case, one can approximate the multiple roots as well as its multiplicity. The most common open method for multiple roots is modified Newton's method

$$x_{n+1} = x_n - m\frac{g(x)}{g'(x)}$$

If $f(x)$ has a multiple zero ξ_m of m multiplicity, at that point the above method can be gotten from Newton's method for the modified function

$$k(x) = \sqrt{h(x)}$$

have a simple zero at a similar point. In this manner, this method is otherwise called modified Newton's method in the literature. So, as to increase its local order of convergence, Laguerre introduced a cubically convergent family depends upon the same function. The famous well known strategies like Halley's method, Ostrowski's square-root method and Euler-Cauchy method are found to be special cases of this family.

Every one-point third order strategies for multiple roots with known multiplicity m, rely upon the evaluations of f, f' and f'' per step. On the other hand, from empirical point of o view, evaluation of second-derivative of every iteration might be a complicated task. In this way, many scientists have introduced a few non-ideal second-derivative free methods for multiple roots. These methods either require the calculation of (one f and two f') or (two f and one f') per step. Dong, Homeier, introduced two-point third order strategies using one f and two f' per step.Thus, Dong, Victory and Neta introduced strategies requiring two f and one f' per step. Neta and Jhonson established another non-optimal two-point fourth-order scheme(consuming one f and three f') based on the well known Jaratt's method after got influenced or motivated by research activities. It is imperative to note that these techniques are not optimal as per the Kung-Traub conjecture. Subsequently, getting new computationally efficient optimal fourth-order techniques is an impressive and challenging task in computational mathematics.

Presently, we talk about the case when the multiplicity m of the multiple root is not expressed explicitly. Traub proposed the following method to approximate the multiplicity m of the root as:

$$m \approx \frac{log|f(x)|}{log|f(x)/f'(x)|},$$

when m is sufficiently closed to the root

Lagouanelle introduced another approach to approximate m as: $m \approx \frac{f'(x)^2}{f'(x)^2 - f(x)f''(x)}$, when m is sufficiently closed to the root.

1.3.3 Stopping criteria and Computational perspectives

There are a different types of numerical methods which are used for approximating the solutions of scalar nonlinear equations and nonlinear conditions. A few measures have been executed by researchers in the literature to contrast various strategies. These measures are generally based on informational efficiency, efficiency index, number of functional evaluations, convergence order, and number of iterations required for convergence until a given tolerance is achieved. For good and fair examination of iterative techniques, we have shown the residual errors at first three iterations, number of iterations required to obtain ideal tolerance and computational order of convergence(COC) to check hypothetical convergence order. All the calculations in the current work are executed in the computer

algebra system Mathematica utilizing multiple-precision arithmetic. We use Mathematica command to minimize the round-off errors as much as feasible. We acquire an approximate solution upto specified degree rather than the accurate root.
In this way, we utilize the stopping criterion given below:

$$|x_{n+1} - x_n| + |g(x_n)| < \in,$$

where $\in = 10^{-200}$, for determining the value of iterations for each technique. Whenever the stopping criteria is satisfied, x_{n+1} is the required root.

Chapter 2

Third-order iterative method for the multiple roots

2.1 Introduction

In this chapter, we will review the iterative methods for finding multiple roots of the nonlinear equations proposed by Reza Dehghan. We also try to implement the proposed methods on different types of real-life problems.

As everyone knows, finding the roots of nonlinear equation is one of the most significant issues in numerical analysis. This chapter proposed an iterative method which helps us to find a multiple root t of multiplicity m, that is

$$f(t) = f'(t) = ... = f^{m-1}(t) = 0$$

and

$$f^{m}(t) \neq 0,$$

of a nonlinear equation $f(x) = 0$. The classical Newton's method given by:

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

We know, it has linear convergence in case of multiple root. Consequently, some of the modifications or alterations have been proposed to get a method with higher order of convergence. As it is hard to discover analytical solutions for non linear problems, howerver numerical techniques might be utilized to acquire approximate solutions. Along these lines, iterative methods give an interesting alternative for solving these kind of problems. When we examine iterative solvers to find multiple roots of nonlinear equation, which is of the form $f(x) = 0$, where $f(x)$ is a real function characterized in a domain $D \subseteq \mathbb{R}$, we review the classical modified Newton's method given by:

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}$$

Here, m denotes the multiplicity of the solution, Given the multiplicity $m \geq 1$, the order of convergence of this method is two for multiple roots. We see one-point iterative function in the literature, but in the extent of the present reality, these methods are no longer use of practical interest as a result of their theoretical drawbacks with respect to efficiency index and convergence order. In addition, most of the one-point strategies are computationally costly and ineffective when they are proved on numerical examples. In this manner, multipoint iterative strategies are better candidates to qualify as proficient.

One of the modified Newton's method for multiple root is given as below:

$$x_{n+1} = x_n - \frac{f^{m-1}(x_n)}{f^{m}(x_n)} \quad n = 0, 1, 2$$

since t is a simple root of $f^{(m-1)}(x) = 0$

Order if convergence of above equation is 2 i.e. it is quadratically convergent. In this, a variant of Newton's method have been introduced for multiple root of nonlinear equation which is cubically convergent.

2.2 Description of the method

In this section, the modified Newton's method is upgraded as given below:

$$f^{(m-1)}(x) = f^{(m-1)}(x_n) + \int_{x_n}^{x} f^{(m)}(\theta)d\theta$$

Now, Simpson's three-eigths formula is used to approximate the $\int_{x_n}^{x} f^{(m)}(\theta)d\theta$, as given below:

$$\int_{x_n}^{x} f^{(m)}(\theta)d\theta \cong \frac{x - x_n}{8}(f^{(m)}(x_n) + 3f^{(m)}(x_n + h) + 3f^{(m)}(x_n + 2h) + f^{(m)}(x))$$

where

$$h = \frac{x - x_n}{3}$$

Now, we assume that t is the multiple root of $f(x)$ with m multiplicity.

$$\begin{aligned} f^{(m-1)}(t) &= f^{(m-1)}(x_n) + \int_{x_n}^{t} f^{(m)}(\theta) d\theta \cong f^{(m-1)}(x_n) + \frac{t - x_n}{8}\bigg(f^{(m)}(x_n) \\ &\quad + 3f^{(m)}(x_n + \frac{t - x_n}{3}) + 3f^{(m)}(x_n + 2\frac{t - x_n}{3}) + f^{(m)}(t)\bigg) \end{aligned} \tag{2.1}$$

On setting, $x_{n+1} = t$ in (2.1), we get

$$\begin{aligned} f^{(m-1)}(t) &= f^{(m-1)}(x_n) + \frac{x_{n+1} - x_n}{8}\bigg(f^{(m)}(x_n) + 3f^{(m)}(x_n + \frac{x_{n+1} - x_n}{3}) \\ &\quad + 3f^{(m)}(x_n + \frac{x_{n+1} - x_n}{3}) + 3f^{(m)}(x_n + 2\frac{x_{n+1} - x_n}{3}) + f^{(m)}(x_{n+1})\bigg) \end{aligned}$$

$$x_{n+1} - x_n = \frac{-8f^{(m-1)}(x_n)}{f^{(m)}(x_n) + 3f^{(m)}(x_n + \frac{x_{n+1}-x_n}{3}) + 3f^{(m)}(x_n + 2\frac{x_{n+1}-x_n}{3}) + f^{(m)}(x_{n+1})}$$

$$x_{n+1} = x_n - \frac{8f^{(m-1)}(x_n)}{f^{(m)}(x_n) + 3f^{(m)}(x_n + \frac{x_{n+1}-x_n}{3}) + 3f^{(m)}(x_n + 2\frac{x_{n+1}-x_n}{3}) + f^{(m)}(x_{n+1})}$$

By replacing x_{n+1} in right hand side by,

$$x_{n+1} = x_n - \frac{f^{(m-1)}(x_n)}{f^{(m)}(x_n)}$$

$$x_{n+1} - x_n = \frac{-f^{(m-1)}(x_n)}{f^{(m)}(x_n)}$$

$$x_{n+1} = x_n - \frac{8f^{(m-1)}(x_n)}{f^{(m)}(x_n) + 3f^{(m)}(x_n - \frac{k(x_n)}{3}) + 3f^{(m)}(x_n - 2\frac{k(x_n)}{3}) + f^{(m)}(x_n - k(x_n)}$$

where

$$k(x_n) = \frac{f^{(m-1)}(x_n)}{f^{(m)}(x_n)}$$

$$\Rightarrow k(t) = \frac{f^{(m-1)}(r)}{f^{(m)}(r)}$$

$$\Rightarrow k(t) = 0$$

Remark 1 *We notice that*

$$F_m(x) = f^{(m)}(x) + 3f^{(m)}(x - \frac{k(x)}{3}) + 3f^{(m)}(x - 2\frac{k(x)}{3}) + f^{(m)}(x - k(x)$$

then

$$F_m(t) = f^{(m)}(t) + 3f^{(m)}(r - \frac{k(t)}{3}) + 3f^{(m)}(t - 2\frac{k(t)}{3}) + f^{(m)}(x - k(t)$$

We have, $k(t) = 0$

$$F_m(t) = f^{(m)}(t) + 3f^{(m)}(t - 0) + 3f^{(m)}(t - 0) + f^{(m)}(t - 0)$$

$$F_m(t) = 8f^{(m)}(t)$$

and

$$\begin{aligned} F'_m(x) &= f^{(m+1)}(x) + 3f^{(m+1)}(x - \frac{k(x)}{3})[1 - \frac{k'(x)}{3}] + 3f^{(m+1)}(x - 2\frac{k(x)}{3})[1 - \frac{2k'(x)}{3}] \\ &+ f^{(m+1)}(x - k(x)[1 - k'(x)] \end{aligned}$$

$$\begin{aligned} F'_m(t) &= f^{(m+1)}(t) + 3f^{(m+1)}(t - \frac{k(t)}{3})[1 - \frac{k'(t)}{3}] + 3f^{(m+1)}(t - 2\frac{k(t)}{3})[1 - \frac{2k'(t)}{3}] \\ &+ f^{(m+1)}(r - k(t))[1 - k'(t)] \end{aligned}$$

We know,

$$k'(x) = \frac{f^{(m)}(x)f^{(m)}(x) - f^{(m-1)}(x)f^{(m+1)}(x)}{(f^{(m)}(x))^2}$$

$$k'(x) = \frac{(f^{(m)}(x))^2 - f^{(m-1)}(x)f^{(m+1)}(x)}{(f^{(m)}(x))^2}$$

$$k'(t) = \frac{(f^{(m)}(t))^2 - f^{(m-1)}(t)f^{(m+1)}(t)}{(f^{(m)}(t))^2}$$

We know, $f^{(m-1)}(t) = 0$ *and* $f^{(m+1)}(t) = 0$

$$\Rightarrow k'(t) = 1$$

$$\begin{aligned} F'_m(t) &= f^{(m+1)}(t) + 3f^{(m+1)}(r - \frac{k(t)}{3})[1 - \frac{k'(t)}{3}] + 3f^{(m+1)}(t - 2\frac{k(t)}{3})[1 - \frac{2k'(t)}{3}] \\ &+ f^{(m+1)}(t - k(t)[1 - k'(t)] \end{aligned}$$

$$\begin{aligned} F'_m(t) &= f^{(m+1)}(t) + 3f^{(m+1)}(t)[1 - \frac{1}{3}] + 3f^{(m+1)}(t)[1 - \frac{2}{3}] \\ &+ f^{(m+1)}(t)[1 - 1] \end{aligned}$$

$$F'_m(t) = f^{(m+1)}(t) + 2f^{(m+1)}(t) + f^{(m+1)}(t)$$

$$\Rightarrow F'_m(t) = 4f^{(m+1)}(t)$$

2.2.1 Analysis of convergence

The conduct of the convergence of the proposed strategy for multiple root is considered in the following theorem.

Theorem 3 *Let*

$$f : D \subseteq \mathbb{R} \rightarrow \mathbb{R},$$

where D is an open interval. Assume that f is a sufficiently differentiable function on the interval D and f has a multiple root t of multiplicity $m > 1$ in D. If x_0 is sufficiently near to t, then the method defined above has at least cubic convergence.

Proof 1 *Assume that*

$$x_{n+1} = g(x_n)$$

therefore

$$g(x) = x - \frac{8f^{(m-1)}(x)}{F_{(m)}(x)}$$

$\Rightarrow$

$$g'(x) = 1 - \frac{8f^{(m)}(x)}{F_m(x)} + \frac{8f^{(m-1)}(x)F'_m(x)}{(F_m(x))^2}$$

Then,

$$\begin{aligned} g''(x) &= \frac{-8f^{(m+1)}(x)}{F_m(x)} + \frac{8f^{(m)}(x)F'_m(x)}{(F_m(x))^2} + \frac{8f^{(m)}(x)F'_m(x) + 8f^{(m-1)}(x)F''_m(x)}{(F_m(x))^2} \\ &- \frac{16f^{(m-1)}(x)((F'_m(x))^2}{(F_m(x))^3} \end{aligned}$$

Since the root t is the simple root of $f^{(m-1)}(x)$, therefore, on substituting x=t, we get

$$g'(t) = 1 - \frac{8f^{(m)}(t)}{F_m(t)} = 1 - \frac{8f^{(m)}(t)}{8f^{(m)}(t)} = 0$$

and

$$\begin{aligned} g''(t) &= \frac{-8f^{(m+1)}(t)}{F_m(t)} + \frac{8f^{(m)}(t)F'_m(t)}{(F_m(t))^2} + \frac{8f^{(m)}(t)F'_m(t)}{(F_m(t))^2} \\ &= -\frac{8f^{(m+1)}(t)}{8f^{(m+1)}(t)} + \frac{16f^{(m+1)}(t)4f^{(m+1)}(t)}{(8f^{(m+1)}(t))^2} = 0 \end{aligned}$$

Hence,the theorem is proved.

2.3 Some numerical techniques

Let m denotes the multiplicity of the function, and x_0 be the initial guess and n be the number of iterations.

Example 1: Consider the van der Waals equation of state,

$$P + \frac{an^2}{V^2}(V - nb) = nRT,$$

here a and b are denoted by the behaviour of a real gas by presenting in the perfect equations two parameters, a and b, known as van der Waal's constants. The assurance of the volume V of the gas in terms of the rest of the parameters which requires the solution of a nonlinear equation in V:

$$PV^3 - (nbP + nRT)V^2 + an^2V - abn^2 = 0$$

Here, the constants a and b that describe a specific gas, one can discover values for n, P, and T, such that this equation has three roots. By utilizing the specific values, we acquired the nonlinear function:

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

which have three zeroes, such that one is $x = 1.75$, of two multiplicity, and the other one is $x = 1.72$. Our ideal root was $x = 1.75$

x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1} - x_n\rvert$
1.8	3.47651×10^{-11}	0.00144667
	6.90198×10^{-8}	0.00144667
	3.47651×10^{-11}	-

Example 2: Fractional conversion in a chemical reactor: Let us examine the given equation

$$f_2(x) = \frac{x}{1-x} - 5log\frac{[0.4(1-x)]}{[0.4-0.5x]} + 4.45977$$

x indicates the fractional conversion of Species A in a chemical reactor. x is bounded in the region $0 \leq x \leq 1$.

x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$
1.4	0.0144435	0.0202976
	0.0183712	0.0202976
	0.0144435	-

Example 3: Continuous stirred tank reactor In this example, we have the isothermal continuous stirred tank reactor(CSTR) problem.
The given reaction method develops in the reactor

$$k_1 + k_4 \to k_2$$
$$k_2 + k_4 \to k_3$$
$$k_3 + k_4 \to k_4$$
$$k_3 + k_4 \to k_5$$

where parts k_1 and k_2 are fed to the reactor at rates of Q and q-Q, respectively. Douglas examined this model in detail so as to structure a simple feedback control systems. In the displaying study, the equation for the transfer function of reactor was as follows :

$$M_R\frac{2.98(x+2.25)}{(s+1.45)(s+2.85)^2(s+4.35)} = -1$$

M_R denotes the gain of proportional controller. If $M_R = 0$ is chosen, then at that point we get the poles of the open-loop transfer function as roots of the nonlinear equation.

$$f_3(x) = x^4 + 11.50x^3 + 47.49x^2 + 83.06325x + 51.23266875 = 0$$

for example $x = -1.45$, -2.85, -2.85, -4.35. In this manner, we observe that there is only ome root $x = 2.85$ with two multiplicity.

x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$
-5	0.724535	0.00634791
	0.780723	0.174805
	-0.422132	0.168457
	0.724535	-

x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$
-3	5.68434×10^{-14}	6.86019×10^{-9}
	5.68434×10^{-14}	9.48661×10^{-7}
	-1.93268×10^{-12}	9.41801×10^{-7}
	5.68434×10^{-14}	-

Example 4: Another nonlinear test function is given by:

$$f_4(x) = ((x-1)^3 - 1)^{50}$$

The above function has a multiple zero at $x = 2$ of multiplicity

x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$
2.1	0.	0.0000768083
	1.34319×10^{-182}	0.0000768024
	$2.481145090720014\times10^{-388}$	5.89891×10^{-9}

Example 5: Consider a 5×5 matrix:

$$f_5(x) = (x-2)^4(x+1)$$

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

x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$
1.9	0.	6.29892×10^{-8}
	4.72263×10^{-29}	6.29891×10^{-8}
	1.16682×10^{-61}	4.44089×10^{-16}
	0.	-

x_n	$\|f(x_n)\|$	$\|x_{n+1} - x_n\|$
2.2	0.	7.87944×10^{-7}
	1.15639×10^{-24}	7.87944×10^{-7}
	2.18648×10^{-53}	5.19584×10^{-14}
	0.	-

Chapter 3

A class of modified Newton-like eighth-order methods for multiple roots

3.1 Introduction

Importance of solving nonlinear problems is justified by numerous physical and technical applications and intense growth of the field over the past decades. These problems arise in many areas of natural and physical sciences, which include initial and boundary value problems, heat and fluid flow problems, electrostatics problems, as well as problems associated with global positioning systems (GPS). In absence of analytical solutions, one of the possible ways to tackle the problem is to use appropriate numerical methods for finding approximate solutions. An algorithm, which is a cornerstone to the modern study of root-finding algorithms was made by Newton through his 'method of fluxions'. Later on this method was polished by Raphson to produce what we now know as the Newton-Raphson method. It converges quadratically for simple roots but if the root is non-simple, the convergence becomes linear. Since then, a tremendous amount of effort has been made in the direction of improving the convergence resulting in modified Newton method (also known as Rall's method) for finding multiple roots of nonlinear equations of the form $f(x) = 0$, where $f(x)$ is real function defined in a domain $D \subseteq \mathbb{R}$. It is given by

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}. \tag{3.1}$$

Given the multiplicity $m \geq 1$ in advance, it converges quadratically for multipe roots. Although, there are many one-point iterative methods available in the literature but when seen from real context, they are not of practical interest because of their theoretical limitations regarding convergence order and efficiency index. Moreover, most of the one-point methods are computationally expensive and inefficient when they are tested on academic problems originating from real life. Therefore, multipoint iterative methods are better candidates to qualify as efficient solvers. The good thing with multipoint iterative methods without memory for scalar nonlinear equations is that we have a conjecture about their convergence order. According to the Kung-Traub conjecture, any multipoint method without memory can reach its convergence order of at most 2^{n-1} for n functional evaluations. A large community of researchers from the world wide turn towards the most important class of multipoint iterative methods and proposed various optimal fourth-order methods (requiring three functional evaluations, per iteration) and non-optimal methods for approximating multiple zeros of nonlinear functions. In the literature, there are limited number of multipoint point iterative methods having sixth-order of convergence. For instance, Thukral proposed the following sixth-order multipoint iteration scheme:

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - m\frac{f(x_n)}{f'(x_n)}\sum_{i=1}^{3}\left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{i}{m}}, \\ x_{n+1} &= z_n - m\frac{f(x_n)}{f'(x_n)}\left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}\left[\sum_{i=1}^{3}\left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{i}{m}}\right]^2. \end{aligned} \tag{3.2}$$

In 2015, Geum, presented a non-optimal class of two-point sixth-order methods as follows:

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - Q(u_n, s_n)\frac{f(y_n)}{f'(y_n)}, \end{aligned} \tag{3.3}$$

where $u_n = \sqrt[m]{\frac{f(y_n)}{f(x_n)}}$, $s_n = \sqrt[m-1]{\frac{f'(y_n)}{f'(x_n)}}$ $(m > 1)$, and $Q : \mathbb{C} \to \mathbb{C}$ is a holomorphic function in the neighborhood of origin $(0, 0)$. But, the main drawback of this scheme is that it does not work for simple zeros (i.e., for $m = 1$).

In 2016, Geum developed another non-optimal family of three-point sixth-order methods for multiple zeros and it is defined by

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ w_n &= y_n - mG(u_n)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= w_n - mK(u_n, v_n)\frac{f(x_n)}{f'(x_n)}, \end{aligned} \tag{3.4}$$

where $u_n = \sqrt[m]{\frac{f(y_n)}{f(x_n)}}$ and $v_n = \sqrt[m]{\frac{f(w_n)}{f(x_n)}}$. The weight functions $G : \mathbb{C} \to \mathbb{C}$ and $K : \mathbb{C}^2 \to \mathbb{C}$ are analytic in the neighborhood of 0 and $(0, 0)$, respectively. It can be seen that the methods (3.3) and (3.4) require four function evaluations to achieve sixth-order of convergence. Therefore, these methods are not optimal in accordance with Kung and Traub conjecture. It is needless to mention that in the last decades much effort has been done to develop and analyze optimal eighth-order methods for multiple zeros but with no success. Motivated and inspired by this fact, Behl introduced an optimal family of eighth-order iterative methods in case of multiple roots for the first time. Its iterative expression is given by

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - u_nQ(h_n)\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= z_n - u_nv_nG(h_n,\ v_n)\frac{f(x_n)}{f'(x_n)}, \end{aligned} \tag{3.5}$$

where the weight functions $Q : \mathbb{C} \to \mathbb{C}$ is analytic in the neighborhood of (0) and $G : \mathbb{C}^2 \to \mathbb{C}$ is holomorphic in the neighborhoods of $(0, 0)$, with $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $h_n = \frac{u_n}{a_1 + a_2u_n}$ and $v_n = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}$, where a_1 and a_2 are free disposable real parameters.

Furthermore, Zafar presented another optimal eighth-order family using the weight function approach as follows:

$$\begin{aligned}
y_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\\
z_n &= y_n - m u_n H(u_n)\frac{f(x_n)}{f'(x_n)},\\
x_{n+1} &= z_n - u_n v_n (A_2 + A_3 u_n) P(v_n) G(w_n)\frac{f(x_n)}{f'(x_n)},
\end{aligned} \tag{3.6}$$

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

It is clear from the above discussions that there are a very small number of optimal eighth-order methods which can handle the case of multiple zeros. Moreover, these type of methods have not been discussed in deep till date. Therefore, the main motivation of the current research work is to present a new optimal class of iterative methods having eighth-order convergence which exploit weight function technique for computing multiple zeros. Our proposed scheme requires only four function evaluations $\big(f(x_n),\ f'(x_n),\ f(y_n)$ and $f(z_n)\big)$ per full iteration which is in accordance with the classical Kung-Traub conjecture. Furthermore, we manifest that the proposed methods have good stability characteristics, reasonable errors in the estimation of multiple zeros.

Our presentation is unfolded in what follows. The new eighth-order scheme and its convergence analysis is presented in **Section 2**. In **Section 3**, some special cases are included based on the different choices of weight functions employed at second and third substeps of the designed family. In **Section 4**, numerical experiments are included which illustrate the efficiency, accuracy of the scheme in comparison to other methods proposed in the scientific literature. Finally, **Section 5** is devoted to some conclusions.

3.2 Construction of the family

In this section, we intend to develop a new optimal eighth-order scheme for multiple roots with known multiplicity $m \geq 1$. We here establish a main theorem describing the convergence analysis of the proposed family. So, we present the three-step scheme as follows:

$$
\begin{aligned}
y_n &= x_n - m\lambda_n,\\
z_n &= y_n - m\left(\frac{u_n}{1-u_n}\right)\lambda_n H(t_n),\\
x_{n+1} &= z_n - u_n\lambda_n\left(\frac{v_n}{1+\alpha_1 v_n+\alpha_2 v_n^2}\right)\Big(R(u_n)+P(w_n)\Big),
\end{aligned}
\tag{3.7}
$$

where $\lambda_n = \frac{f(x_n)}{f'(x_n)}$ and the weight functions $H, R, P : \mathbb{C} \to \mathbb{C}$ are analytic in the neighborhood of origin with $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $t_n = \frac{u_n}{b_1+b_2u_n}$, $v_n = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}$ and $w_n = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$. Here, $b_1,\ b_2,\ \alpha_1$ and α_2 are free disposable real parameters.

In the following theorem, we demonstrate that how to construct weight functions $H,\ R$ and P so that the proposed scheme (3.7) arrives at eighth order without consuming any additional functional evaluations.

Theorem 4 *Assume that $f : \mathbb{C} \to \mathbb{C}$ is an analytic function in the region enclosing the multiple zero $x = \alpha$ with multiplicity $m \geq 1$. Then, the iterative expression defined by* (3.7) *has eighth-order convergence when it satisfies the following conditions:*

$$
\begin{cases}
H(0)=1,\ \ H'(0)=b_1,\ \ R(0)=m-P(0),\ \ R'(0)=2m,\ \ R''(0)=\dfrac{6b_1^2m-2b_1b_2m+H''(0)}{b_1^2},\\
\alpha_1=-1,\ \ R'''(0)=\dfrac{12b_1^3m-18b_1^2b_2m+6b_1b_2^2m+(9b_1-6b_2)H''(0)+H'''(0)}{b_1^3},\ \ P'(0)=2m.
\end{cases}
\tag{3.8}
$$

Proof 2 *Let $x = \alpha$ be a multiple zero of $f(x)$. Using Taylor's series expansion of $f(x_n)$ and $f'(x_n)$ about α, we obtain*

$$
f(x_n) = \frac{f^{(m)}(\alpha)}{m!}e_n^m\left(1+c_1e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+c_5e_n^5+c_6e_n^6+c_7e_n^7+c_8e_n^8+O(e_n^9)\right) \tag{3.9}
$$

and

$$
\begin{aligned}
f'(x_n) =& \frac{f^m(\alpha)}{m!}e_n^{m-1}\Big(m+c_1(m+1)e_n+c_2(m+2)e_n^2+c_3(m+3)e_n^3+c_4(m+4)e_n^4+c_5(m+5)e_n^5\\
&+c_6(m+6)e_n^6+c_7(m+7)e_n^7+c_8(m+8)e_n^8+O(e_n^9)\Big),
\end{aligned}
\tag{3.10}
$$

respectively. Here, $e_n = x_n - \alpha$ and $c_k = \dfrac{1}{k!}\dfrac{f^{(k)}(\alpha)}{f'(\alpha)}$, $k = 1, 2, 3, \ldots$.
Using the above expressions (3.9) *and* (3.10) *in the first substep of* (3.7), *we get*

$$
y_n - \alpha = \frac{c_1e_n^2}{m} + \frac{(-(1+m)c_1^2+2mc_2)e_n^3}{m^2} + \sum_{j=1}^{5}\Gamma_j e_n^{j+3} + O(e_n^9), \tag{3.11}
$$

where $\Gamma_j = \Gamma_j(m, c_1, c_2, \ldots, c_8)$ *are given in terms of* $m, c_1, c_2, c_3, \ldots, c_8$ *for* $1 \leq j \leq 5$. *The explicit expressions for the first two terms* Γ_1 *and* Γ_2 *are given by* $\Gamma_1 = \frac{1}{m^3}\{3m^2c_3 + (m+1)^2c_1^3 - m(3m+4)c_2c_1\}$ *and* $\Gamma_2 = \frac{1}{m^4}\{(m+1)^3c_1^4 - 2m(2m^2+5m+3)c_2c_1^2 + 2m^2(2m+3)c_3c_1 + 2m^2(c_2^2(m+2) - 2c_4m)\}$.

Using again Taylor's series expansion, we obtain

$$\begin{aligned} f(y_n) =& f^{(m)}(\alpha)e_n^{2m}\Bigg[\frac{\left(\frac{c_1}{m}\right)^m}{m!} + \frac{(2c_2m - c_1^2(m+1))\left(\frac{c_1}{m}\right)^m e_n}{c_1 m!} + \left(\frac{c_1}{m}\right)^{1+m}\frac{1}{2m!c_1^3} \\ & \big\{(3+3m+3m^2+m^3)c_1^4 - 2m(2+3m+2m^2)c_1^2c_2 + 4(-1+m)m^2c_2^2 + 6m^2c_1c_3\big\}e_n^2 \\ & + \sum_{j=1}^{5}\bar{\Gamma}_j e_n^{j+3} + O(e_n^9)\Bigg] \end{aligned} \tag{3.12}$$

and

$$u_n = \frac{c_1e_n}{m} + \frac{(2c_2m - c_1^2(m+2))e_n^2}{m^2} + \gamma_1e_n^3 + \gamma_2e_n^4 + \gamma_3e_n^5 + O(e_n^6), \tag{3.13}$$

where

$$\begin{cases} \gamma_1 = \dfrac{1}{2m^3}\big[c_1^3(2m^2+7m+7) + 6c_3m^2 - 2c_2c_1m(3m+7)\big], \\ \gamma_2 = -\dfrac{1}{6m^4}\big[c_1^4(6m^3+29m^2+51m+34) - 6c_2c_1^2m(4m^2+16m+17) + 12c_3c_1m^2(2m+5) \\ \quad + 12m^2(c_2^2(m+3) - 2c_4m)\big], \\ \gamma_3 = \dfrac{1}{24m^5}\big[-24m^3(c_2c_3(5m+17) - 5c_5m) + 12c_3c_1^2m^2(10m^2+43m+49) + 12c_1m^2\{c_2^2(10m^2+47m \\ \quad + 53) - 2c_4m(5m+13)\} - 4c_2c_1^3m(30m^3+163m^2+306m+209) + c_1^5(24m^4+146m^3+355m^2 \\ \quad + 418m+209)\big]. \end{cases} \tag{3.14}$$

Now, using the above expression (3.14), *we get*

$$t_n = \frac{c_1}{mb_1}e_n + \sum_{i=1}^{4}\Theta_j e_n^{j+1} + O(e_n^6). \tag{3.15}$$

where $\Theta_j = \Theta_j(b_1, b_2, m, c_1, c_2, \ldots, c_8)$ *are given in terms of* $b_1, b_2, m, c_1, c_2, \ldots, c_8$, *where the two coefficients* Θ_1 *and* Θ_2 *are wriiten explicitly as* $\Theta_1 = -\frac{b_2c_1^2 + b_1((2+m)c_1^2 - 2mc_2)}{m^2b_1^2}$, *and*$\Theta_2 = \frac{1}{2m^3b_1^3}[2b_2^2c_1^3 + 4b_1b_2c_1((2+m)c_1^2 - 2mc_2) + b_1^2\{(7+7m+2m^2)c_1^3 - 2m(7+3m)c_1c_2 + 6m^2c_3\}]$.

Due to the fact that $t_n = \dfrac{u_n}{b_1 + b_2u_n} = \mathcal{O}(e_n)$, *therefore, it suffices to expand weight function* $H(t_n)$ *in the neighborhood of origin by Taylor's expansion up to fourth-order* *term as follows:*

$$H(t_n) \approx H(0) + H'(0)t_n + \frac{1}{2!}H''(0)t_n^2 + \frac{1}{3!}H^{(3)}(0)t_n^3 + \frac{1}{4!}H^{(4)}(0)t_n^4, \tag{3.16}$$

where $H^{(k)}$ *represents the k-th derivative. By inserting the expressions* (3.9)-(3.16) *in the second substep of scheme* (3.7)*, we have*

$$\begin{aligned} z_n - \alpha =& \frac{(m - H(0))c_1}{m^2}e_n^2 + \frac{2m(m - H(0))b_1c_2 - \left(H'(0) + (m + m^2 - 2H(0) - mH(0))b_1\right)c_1^2}{m^3b_1}e_n^3 \\ &+ \sum_{s=1}^{5} \Omega_s e_n^{s+3} + O(e_n^9). \end{aligned} \tag{3.17}$$

where $\Omega_s = \Omega_s(H(0), H'(0), H''(0), H^{(3)}(0), H^{(4)}(0), m, b_1, b_2, c_1, c_2, \ldots, c_8)$, $s = 1, 2, 3, 4, 5$.

It is clear from error equation (3.17) *that in order to obtain at least fourth-order convergence, the coefficients of* e_n^2 *and* e_n^3 *must vanish simultaneously. That is possible only for the following values of* $H(0)$ *and* $H'(0)$*, which can be calculated from the expression* (3.17)*:*

$$H(0) = m, \quad H'(0) = mb_1. \tag{3.18}$$

Substituting the above values of $H(0)$ *and* $H'(0)$ *in* (3.17)*, we obtain*

$$z_n - \alpha = \frac{(m(5+m)b_1^2 - H''(0) + 2mb_1b_2)c_1^3 - 2m^2b_1^2c_1c_2}{2m^4b_1^2}e_n^4 + \sum_{r=1}^{4} L_r e_n^{s+4} + O(e_n^9), \tag{3.19}$$

where $L_r = L_r(H''(0), H^{(3)}(0), H^{(4)}(0), m, b_1, b_2, c_1, c_2, \ldots, c_8)$, $r = 1, 2, 3, 4$.

Now, again by using the Taylor series expansion, we have

$$f(z_n) = f^{(m)}(\alpha)e_n^{4m}\left[\frac{2^{-m}}{m!}\left(\frac{(m(5+m)b_1^2 - H''(0) + 2mb_1b_2)c_1^3 - 2m^2b_1^2c_1c_2}{m^4b_1^2}\right)^m + \sum_{s=1}^{5} \bar{P}_s e_n^s + O(e_n^6)\right], \tag{3.20}$$

$$v_n = \frac{(m(5+m)b_1^2 - H''(0) + 2mb_1b_2)c_1^2 - 2m^2b_1^2c_2}{2m^3b_1^2}e_n^2 + \gamma_0 e_n^3 + O(e_n^4), \tag{3.21}$$

and

$$w_n = \frac{(m(5+m)b_1^2 - H''(0) + 2mb_1b_2)c_1^2 - 2m^2b_1^2c_1c_2}{2m^4b_1^2}e_n^3 + \gamma_1 e_n^4 + O(e_n^5), \tag{3.22}$$

where

$$\gamma_0 = \frac{1}{6b_1^3m^4}\Big[\big(6b_1^2b_2m(5+2m) + 2b_1^3m\big(19+15m+2m^2\big) - 6b_2H''(0) + 3b_1\big(2b_2^2m - 5H''(0) - 2mH''(0)\big) + H'''(0)\big)c_1^3 - 12b_1m\big(2b_1b_2m + b_1^2m(5+m) - H''(0)\big)c_1c_2 + 12b_1^3m^3c_3\Big],$$

$$\gamma_1 = \frac{1}{6b_1^3m^5}\Big[\big(6b_1^2b_2m(7+2m) + b_1^3m\big(68+51m+7m^2\big) - 6b_2H''(0) + 3b_1\big(2b_2^2m - 7H''(0) - 3mH''(0)\big) + H'''(0)\big)c_1^4 - 6b_1m\big(6b_1b_2m + b_1^2m(17+m) - 3H''(0)\big)c_1^2c_2 + 12b_1^3m^3c_2^2 + 12b_1^3m^3c_1c_3\Big].$$

It is clear from equations (3.13) and (3.22) that u_n and w_n are of order e_n and e_n^3, respectively. Therefore, we can expand weight function $R(u_n)$ and $P(w_n)$ in the neighborhood of origin by Taylor's series expansion up to fourth-order and third-order terms, respectively as follow:

$$R(u_n) \approx R(0) + R'(0)u_n + \frac{1}{2!}R''(0)u_n^2 + \frac{1}{3!}R^{(3)}(0)u_n^3 + \frac{1}{4!}R^{(4)}(0)u_n^4, \tag{3.23}$$

and

$$P(w_n) \approx P(0) + P'(0)w_n + \frac{1}{2!}P''(0)w_n^2 + \frac{1}{3!}P^{(3)}(0)w_n^3. \tag{3.24}$$

By using the expressions (3.9)-(3.24) in the last substep of the proposed scheme (3.7), we have

$$e_{n+1} = \frac{(m - P(0) - R(0))c_1((H''(0) - (m+5)b_1^2 - 2b_1b_2)c_1^2 - 2mb_1^2c_2)}{2b_1^2m^5}e_n^4 + \sum_{i=1}^{4}\psi_i e_n^{i+4} + O(e_n^9), \tag{3.25}$$

where $\psi_i = \psi_i(m, b_1, b_2, \alpha_1, \alpha_2, H(0), H'(0), H''(0), H^{(3)}(0), P(0), P'(0), P''(0), P^{(3)}(0), R(0), R'(0), R''(0), R^{(3)}(0), c_1, c_2, \ldots, c_8)$, $i = 1, 2, 3, 4$.

For obtaining at least fifth-order convergence, we need to choose $R(0) = m - P(0)$. Further, substituting $R(0) = m - P(0)$ in $\psi_1 = 0$, one can obtain

$$R'(0) = 2m. \tag{3.26}$$

Now inserting $R(0) = m - P(0)$ and $R'(0) = 2m$ in $\psi_2 = 0$, we obtain the following relations

$$2b_1b_2\alpha_1m - \alpha_1H''(0) + b_1^2\left((11+5\alpha_1)m + (1+\alpha_1)m^2 - R''(0)\right) = 0 \text{ and } 2b_1^2m^2(\alpha_1+1) = 0, \tag{3.27}$$

which further yields

$$\alpha_1 = -1, \qquad R''(0) = \frac{6b_1^2 m - 2b_1 b_2 m + H''(0)}{b_1^2}. \tag{3.28}$$

By inserting the values of $R(0)$, $R'(0)$, $R''(0)$ *and* α_1 *obtained from the above equations in* $\psi_3 = 0$, *we obtain the following independent expressions*

$$\begin{aligned}&-6b_1^2 b_2 m(m + P'(0)) + 3b_1 \bigl(2b_2^2 m^2 + (m + P'(0))H''(0)\bigr) + m(-6b_2 H''(0) + H'''(0)) + b_1^3 m \bigl(6m^2 - 3m(-14\\ &-15P'(0) - R^{(3)}(0)\bigr) = 0, 6b_1^3 m^2 (P'(0) - 2m) = 0,\end{aligned} \tag{3.29}$$

which further gives

$$R'''(0) = \frac{12b_1^3 m - 18b_1^2 b_2 m + 6b_1 b_2^2 m + 9b_1 H''(0) - 6b_2 H''(0) + H^{(3)}(0)}{b_1^3}, \quad P'(0) = 2m. \tag{3.30}$$

Finally, using equations (3.26)–(3.30) *in* (3.7), *one can get the following error equation:*

$$\begin{aligned}e_{n+1} = {}& \frac{1}{48m^{10}b_1^6}\Bigl[c_1\bigl((2b_1 b_2 m + b_1^2 m(m+5) - H''(0))c_1^2 - 2b_1^2 m^2 c_2\bigr)\Bigr]\Bigl(\bigl(24b_1^3 b_2 \alpha_2 m^2 (5+m) + 36b_2^2 m H''(0)\\ &+ 6\alpha_2 H''(0)^2 + 12b_1^2 m\bigl(2b_2^2(3+\alpha_2)m - \alpha_2(5+m)H''(0)\bigr) - 12b_1 m\bigl(2b_2^3 m + 2b_2(3+\alpha_2)H''(0) - H'''(0)\bigr)\\ &+ mH^{(4)}(0) + b_1^4 m\bigl(2(134 + 75\alpha_2)m + 12(7 + 5\alpha_2)m^2 + (8 + 6\alpha_2)m^3 - R^{(4)}(0)\bigr)\bigr)c_1^4 - 24b_1^2 m^2\bigl(2b_1 b_2 \alpha_2 m\\ &+ b_1^2 m(7 + m + \alpha_2(5+m)) - \alpha_2 H''(0)\bigr)c_1^2 c_2 + 24b_1^4 \alpha_2 m^4 c_2^2 + 24b_1^4 m^4 c_1 c_3\Bigr)e_n^8 + O(e_n^9).\end{aligned} \tag{3.31}$$

The consequence of the above error analysis is that the proposed scheme (3.7) *acquires eighth-order convergence using only four functional evaluations (viz.* $f(x_n)$, $f'(x_n)$, $f(y_n)$ *and* $f(z_n)$*) per full iteration. This completes the proof.* ☐

3.2.1 Some special cases of the proposed class

In this section, we will discuss some interesting special cases of our proposed class (3.7) by assigning different forms of weight functions $H(t_n)$, $R(u_n)$ and $P(w_n)$ employed at second and third step, respectively.

1. Let us consider the following optimal class of eighth-order methods for multiple

roots where weight functions are chosen directly from the proposed Theorem **??**:

$$\begin{aligned}
y_n &= x_n - m\lambda_n,\\
z_n &= y_n - m\left(\frac{u_n}{1-u_n}\right)\lambda_n\left[1+t_nb_1+\frac{1}{2}t_n^2H''(0)+\frac{1}{3!}t_n^3H^{(3)}(0)+\frac{1}{4!}t_n^4H^{(4)}(0)\right],\\
x_{n+1} &= z_n - u_n\left(\frac{v_n}{1-v_n+\alpha_2v_n^2}\right)\lambda_n\Bigg[m+2mu_n+\frac{6b_1^2m-2b_1b_2m+H''(0)}{2b_1^2}u_n^2\\
&+\frac{12b_1^3m-18b_1^2b_2m+6b_1b_2^2m+(9b_1-6b_2)H''(0)+H^{(3)}(0)}{6b_1^3}u_n^3+\frac{R^{(4)}(0)}{24}u_n^4+2mw_n\\
&+\frac{P''(0)}{2}w_n^2+\frac{P^{(3)}(0)}{6}w_n^3\Bigg],
\end{aligned}\tag{3.32}$$

where $\lambda_n=\frac{f(x_n)}{f'(x_n)}$, $b_1,b_2,\alpha_2,H''(0),H^{(3)}(0),H^{(4)}(0),\ R^{(4)}(0),P''(0)$ and $P^{(3)}(0)$ are free parameters.

Sub cases of the given scheme (3.32):

(a) Let us consider $H''(0)=H^{(3)}(0)=H^{(4)}(0)=R^{(4)}(0)=P''(0)=P^{(3)}(0)=0$ in expression (3.32), we obtain

$$\begin{aligned}
y_n &= x_n - m\lambda_n,\\
z_n &= y_n - m\left(\frac{u_n}{1-u_n}\right)\lambda_n\left[1+t_nb_1\right],\\
x_{n+1} &= z_n - mu_n\left(\frac{v_n}{1-v_n+\alpha_2v_n^2}\right)\lambda_n\Bigg[1+2u_n+\frac{3b_1-b_2}{b_1}u_n^2+\frac{2b_1^2-3b_1b_2+b_2^2}{b_1^2}u_n^3\\
&+2w_n\Bigg].
\end{aligned}\tag{3.33}$$

2. Moreover, a combination of rational functions produce another optimal eighth-order scheme as follows:

$$\begin{aligned}
y_n &= x_n - m\lambda_n,\\
z_n &= y_n - m\left(\frac{u_n}{1-u_n}\right)\lambda_n\left[1+t_nb_1\right],\\
x_{n+1} &= z_n - u_n\left(\frac{v_n}{1-v_n+\alpha_2v_n^2}\right)\lambda_n\left[\frac{k_1+k_2u_n}{1+k_3u_n+k_4u_n^2}+\frac{\tau_1+w_n+w_n^2}{1+\tau_2w_n}\right],
\end{aligned}\tag{3.34}$$

where

$$\begin{cases} k_1 = m - P(0), \\ k_2 = \dfrac{-2b_1^2m^2 + b_1b_2m^2 + b_2^2m^2 + (8b_1^2m + 2b_1b_2m - 2b_2^2m)P(0) + (2b_1^2 - 3b_1b_2 + b_2^2)P(0)^2}{b_1(b_1m + b_2m + (3b_1 - b_2)P(0))}, \\ k_3 = \dfrac{-4b_1^2m - b_1b_2m + b_2^2m - (2b_1^2 - 3b_1b_2 + b_2^2)P(0)}{b_1(b_1m + b_2m + (3b_1 - b_2)P(0))}, \\ k_4 = \dfrac{\left(5b_1^2 - b_2^2\right)m}{b_1(b_1m + b_2m + (3b_1 - b_2)P(0))}, \\ \tau_1 = P(0), \\ \tau_2 = \dfrac{1-2m}{\tau_1}. \end{cases} \tag{3.35}$$

3. Now, we suggest another rational function forms of weight functions satisfying the conditions as follow:

$$\begin{aligned} y_n &= x_n - m\lambda_n, \\ z_n &= y_n - m\left(\frac{u_n}{1-u_n}\right)\lambda_n\left[1 + t_nb_1\right], \\ x_{n+1} &= z_n - u_n\left(\frac{v_n}{1 - v_n + \alpha_2v_n^2}\right)\lambda_n\left[\frac{1 + \rho_1u_n + \rho_2u_n^2}{\rho_3 + \rho_4u_n} + \frac{l_1 + (l_1 + 2m)w_n}{1 + w_n}\right], \end{aligned} \tag{3.36}$$

where

$$\begin{cases} \rho_1 = \dfrac{4b_1^2m + b_1b_2m - b_2^2m + (2b_1^2 - 3b_1b_2 + b_2^2)P(0)}{b_1(3b_1 - b_2)(m - P(0))}, \\ \rho_2 = \dfrac{(5b_1^2 - b_2^2)\,m}{b_1(3b_1 - b_2)(m - P(0))}, \\ \rho_3 = \dfrac{1}{m - P(0)}, \\ \rho_4 = \dfrac{-2b_1^2 + 3b_1b_2 - b_2^2}{b_1(3b_1 - b_2)(m - P(0))}, \\ l_1 = P(0). \end{cases} \tag{3.37}$$

Remark 2 *Furthermore, it is important to note that weight functions* $H(t_n)$, $R(u_n)$ *and* $P(w_n)$ *play a significant role in the construction of eighth-order schemes. Therefore, it is customary to display different choices of weight functions, provided they must satisfy all the conditions in the Theorem . Hence, we have mentioned above some special cases of new eighth-order schemes* (3.33), (3.34) *and* (3.36) *having simple body structures so that they can be easily implemented in the numerical experiments.*

3.3 Numerical experiments

In this section, we will check the computational aspects of the following proposed methods: expression (3.33) for $(b_1 = 1,\ b_2 = -2,\ \alpha_2 = -3)$, family (3.34) for $(b_1 = 1,\ b_2 = -2,\ \alpha_2 = -3,\ P(0) = \tau_1 = \frac{1}{2})$ and expression (3.36) for $(b_1 = 1,\ b_2 = -2,\ \alpha_2 = -3,\ P(0) = l_1 = \frac{1}{2})$ denoted by $MM1$, $MM2$ and $MM3$, respectively, with some already existing techniques.

In this regard, we consider several test functions coming from real life problems which are mentioned in examples 1-6. We compare our proposed methods with optimal eighth-order method (3.5) given by Behl for $Q(h_n) = m(1 + 2h_n + 3h_n^2)$ and $G(h_n, t_n) = m\left(\frac{1+2t_n+3h_n^2+h_n(2+6t_n+h_n)}{1+t_n}\right)$ and the method (3.6) given by Zafar taking $H(u_n) = 6u_n^3 - u_n^2 + 2u_n + 1,\ P(v_n) = 1 + v_n$ and $G(w_n) = \frac{2w_n+1}{A_2P_0}$ for $(A_2 = P_0 = 1)$. We denote these methods by OM and ZM, respectively. We also compare our proposed methods with family of two-point sixth-order method given by Geum , out of them we choose the case 2A, which is given by:

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\ m > 1, \\ x_{n+1} &= y_n - \left[\frac{m + b_1u_n}{1 + a_1u_n + a_2s_n + a_3s_nu_n}\right]\frac{f(y_n)}{f'(y_n)}, \end{aligned} \tag{3.38}$$

where $u_n = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $s_n = \left(\frac{f'(y_n)}{f(x_n)}\right)^{\frac{1}{m-1}}$, $b_1 = \frac{2m}{m-1}$, $a_1 = -\frac{2m(m-2)}{m-1}$, $a_2 = 2(m-1)$ and $a_3 = 3$.

Finally, we compare them with the non-optimal family of sixth-order methods based on weight function approach presented by the same authors Geum et al. [?], out of them we consider the case 5YD, which is defined as follows:

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)},\ m \geq 1, \\ w_n &= x_n - m\left[\frac{(u_n - 2)(2u_n - 1)}{(u_n - 1)(5u_n - 2)}\right]\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - m\left[\frac{(u_n - 2)(2u_n - 1)}{(5u_n - 2)(u_n + v_n - 1)}\right]\frac{f(x_n)}{f'(x_n)}. \end{aligned} \tag{3.39}$$

We denote methods (3.38) and (3.39) by $GK1$ and $GK2$, respectively.

In the numerical tests presented in Tables 3.1-3.6, we have compared our methods with the known ones on the basis of approximated zeros, residual error of the involved functions, difference between the two consecutive iterations, asymptotic error constants. In Tables 3.1-3.6, we display the number of iteration indices (n), approximated zeros (x_n), absolute residual error of the corresponding function ($|f(x_n)|$), error in the consecutive iterations $|x_{n+1} - x_n|$, computational order of convergence $\rho \approx \frac{\log|f(x_{n+1})/f(x_n)|}{\log|f(x_n)/f(x_{n-1}|}$, $n \geq 2$, (the details of this formula can be seen in, $\left|\dfrac{x_{n+1} - x_n}{(x_n - x_{n-1})^p}\right|$ (where p is either 6 or 8 corresponding to the considered iteration function), the estimation of asymptotic error constant $\eta \approx \lim\limits_{n\to\infty} \left|\dfrac{x_{n+1} - x_n}{(x_n - x_{n-1})^p}\right|$ at the last iteration. We have maintained 4096 significant digits of minimum precision to minimize the round off error.

As mentioned in the above paragraph, we calculate the values of all the constants and functional residuals up to several number of significant digits but we display the value of approximated zero x_n up to 25 significant digits although minimum 4096 significant digits are available with us. The absolute residual error in the function $|f(x_n)|$ and error in the consecutive iterations $|x_{n+1} - x_n|$ are displayed up to 2 significant digits with exponent power which are mentioned in Tables 3.1-3.6. In addition, computational order of convergence is up to 5 significant digits. In addition, we also display $\left|\dfrac{x_{n+1} - x_n}{(x_n - x_{n-1})^p}\right|$ and η up to 10 significant digits. From tables 3.1–3.6, it can be observed that the smaller asymptotic error constant implies that the corresponding method converge faster than the other ones. Although, it may happen in some cases that the method have smaller residual errors and smaller errors difference between two consecutive iterations but have also larger asymptotic error constant. All computations in numerical experiments have been carried out with $Mathematica$ 10 programming package using multiple precision arithmetic. Further, the meaning of $a(\pm b)$ is $a \times 10^{(\pm b)}$ in Tables 3.1-3.6.

Example 1 (van der Waals equation of state):

$$\left(P + \frac{an^2}{V^2}\right)(V - nb) = nRT,$$

where a and b are explains the behavior of a real gas by introducing in the ideal gas equations two parameters, a and b (known as van der Wall's constants), specific for each gas. The determination of the volume V of the gas in terms of the remaining parameters requires the solution of a nonlinear equation in V

$$+PV^3-(nbP+nRT)V^2+an^2V-abn^2=0.$$

Given the constants a and b of a particular gas, one can find values for n, P and T, such that this equation has a three roots. By using the particular values, we obtain the following nonlinear function

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

having three zeros and out of them one is a multiple zero $\alpha=1.75$ of multiplicity of order two and other one simple zero $\alpha=1.72$. However, our desired root is $\alpha=1.75$.

Example 2(Fractional conversion in a chemical reactor):

Let us consider the following expression (please, see for more details of this problem)

$$f_2(x)=\frac{x}{1-x}-5\log\left[\frac{0.4(1-x)}{0.4-0.5x}\right]+4.45977. \tag{3.40}$$

In the above expression, x represents the fractional conversion of species A in a chemical reactor. Since, there is no physical meaning of above fractional conversion if x is less than zero or greater than one. In this sense, x is bounded in the region $0\le x\le 1$. In addition, our required zero (that is simple) to this problem is $\alpha\approx 0.757396246253753879459641297929$. Moreover, it is interesting to note that the above expression is undefined in the region $0.8\le x\le 1$ which is very close to our desired zero. Furthermore, there are some other properties to this function which make the solution more difficult. The derivative of the above expression is very close to zero in the region $0\le x\le 0.5$ and there is an infeasible solution for $x=1.098$.

We can see that the new methods possess smaller residual error and difference between the consecutive approximations in comparison to the existing ones. Moreover, the numerical estimation of the order of convergence coincide with the theoretical one in all cases. In Table 3.2, $*$ means that the corresponding method does not converge to the desired root.

Example 3(Continuous stirred tank reactor (CSTR)):

In our third example, we consider the isothermal continuous stirred tank reactor (CSTR) problem. The following reaction scheme develops in the reactor:

$$\begin{aligned} &A + R \to B \\ &B + R \to C \\ &C + R \to D \\ &C + R \to E, \end{aligned} \tag{3.41}$$

where, components A and R are fed to the reactor at rates of Q and q-Q respectively. The problem was analyzed in detail by Douglas in order to design simple feedback control systems. In the analysis, he gave the following equation for the transfer function of the reactor:

$$K_C \frac{2.98(x + 2.25)}{(s + 1.45)(s + 2.85)^2(s + 4.35)} = -1, \tag{3.42}$$

where K_C is the gain of the proportional controller. The control system is stable for values of K_C that yields roots of the transfer function having negative real part. If we choose $K_C = 0$, we get the poles of the open-loop transfer function as roots of the nonlinear equation:

$$f_3(x) = x^4 + 11.50x^3 + 47.49x^2 + 83.06325x + 51.23266875 = 0 \tag{3.43}$$

given as: $x = -1.45, -2.85, -2.85, -4.35$. So, we see that there is one multiple root $x = -2.85$ with multiplicity 2.

Example 4Let us consider another nonlinear test function, which is given as follows:

$$f_4(x) = ((x - 1)^3 - 1)^{50}.$$

The above function has a multiple zero at $x = 2$ of multiplicity 50.

Table 3.4 shows the numerical results for this example. It can be observed from the numerical tests showed in this table that results are very good for all the methods, being lower the residuals at the newly proposed methods. Moreover, the asymptotic error constant (η) displayed in the last column of Table 3.4 is large for the methods OM and ZM in comparison to the other schemes.

Example 5(Planck's radiation law problem):

Now, we consider the following Planck's radiation law problem which calculates the energy density within an isothermal blackbody:

$$\Psi(\lambda) = \frac{8\pi ch\lambda^{-5}}{e^{\frac{ch}{\lambda BT}} - 1}, \tag{3.44}$$

where λ is the wavelength of the radiation, T is the absolute temperature of the blackbody, B is the Boltzmann constant, h is the Planck's constant and c is the speed of light. We are interested in determining wavelength λ which corresponds to maximum energy density $\Psi(\lambda)$.

Further, $\Psi'(\lambda) = 0$ implies that the maximum value of Ψ occurs when

$$\frac{\frac{ch}{\lambda BT} e^{\frac{ch}{\lambda BT}}}{e^{\frac{ch}{\lambda BT}} - 1} = 5. \tag{3.45}$$

If $x = \frac{ch}{\lambda BT}$, then (3.45) is satisfied when

$$f_5(x) = e^{-x} + \frac{x}{5} - 1 = 0. \tag{3.46}$$

Therefore, the solutions of $f_5(x) = 0$ give the maximum wavelength of radiation λ by means of the following formula:

$$\lambda \approx \frac{ch}{\alpha BT}, \tag{3.47}$$

where α is a solution of (3.46). Our desired root is $x = 4.9651142317442$ with multiplicity $m = 1$.

The numerical results for the test equation $f_5(x) = 0$ are displayed in Table 3.5. It can be observed that the new methods $MM1$ and $MM2$ have small values of residual errors and asymptotic error constants (η) in comparison to the other methods when the accuracy is tested in multi-precision arithmetic.

Example 6 (Global CO_2 model by Bresnahan in ocean chemistry):

In this example, we will discuss the global CO_2 model by Bresnahan iin ocean chemistry which finally leads to the numerical solution of a nonlinear fourth order polynomial in the calculation of pH of the ocean. The effect of atmospheric CO_2 is very complex and varies with location of the ocean. Therefore, considered a simplified approach based on the following assumptions:

1. only the ocean upper layer is considered (not the deep layer),
2. Approximation of the ocean upper layer carbon distribution by perfect mixing so that spatial variations are neglected.

As, CO_2 dissolves in ocean water and undergoes a series of chemical changes that ultimately leads to increased hydrogen ion concentration, denoted as $[H^+]$ and thus acidification. The problem was analyzed by Babajee in order to find the solution of the following nonlinear function:

$$p([H^+]) = \sum_{n=0}^{4} r_n [H^+]^n, \tag{3.48}$$

where

$$\begin{cases} r_0 = 2K_0K_1K_2P_tK_B, \\ r_1 = K_0K_1P_tK_B + 2K_0K_1K_2P_t + K_WK_B, \\ r_2 = K_0K_1P_t + BK_B + K_W - AK_B, \\ r_3 = -K_B - A, \\ r_4 = -1. \end{cases} \tag{3.49}$$

Here, $K_0,\ K_1, K_2, K_W$ and K_B are equilibrium constants. The parameter A represents alkalinity which expresses the neutrality of ocean water and P_t is the gas phase CO_2 partial pressure. We assume the values of $A = 2.050$ and $B = 0.409$ taken by L & Gruber [?] and Bacastow and Keeling [?], respectively. Furthermore, choosing the values of $K_0,\ K_1, K_2, K_W,\ K_B$ and P_t given by Babajee [?], we obtain the following nonlinear equation:

$$f_7(x) = x^4 - \frac{2309x^3}{250} - \frac{65226608163x^2}{500000} + \frac{425064009069x}{25000} - \frac{10954808368405209}{62500000} = 0. \tag{3.50}$$

The roots of $f_7(x) = 0$ are given by $x = -411.452,\ 11.286,\ 140.771,\ 268.332$. Our desired root is -411.452 having multiplicity $m = 1$. Finally, we are interested to find the solution $x = [H^+]$ of the above equation (3.50) to calculate the pH of the ocean.

The numerical experiments of this example are given in Table 3.6. The methods $MM1$, $MM2$ and $MM3$ have small residual errors and asymptotic error constants as compared to the other existing methods. The computational order of convergence for all methods coincides with the theoretical ones in all cases.

3.4 Conclusions

In this paper, we have developed a wide general three-step class of methods for approximating multiple zeros of nonlinear functions numerically. Optimal iteration schemes having eighth-order for multiple zeros have been considered very seldom in the literature, so the presented methods may be regarded as an advancement in the topic. Weight functions based on function-to-function ratios and free parameters are employed at second and third steps of the family which enable us to achieve desired convergence order eight. In the numerical section, we have incorporated variety of real life problems to confirm the efficiency of the proposed technique in comparison to the existing robust methods. From the computational results, we find that the new methods show better performance in terms of precision, residual errors for the considered test functions $f_{1-6}(x)$. Finally, we point out that the easy structure and high convergence order of the proposed class, makes it not only interesting from theoretical point of view but also in practice.

Table 3.1: Convergence behavior of different iterative methods on the test function $f_1(x)$

Methods	n	x_n	$\|f(x_n)\|$	$\|x_{n+1} - x_n\|$	ρ	$\left\| \frac{x_{n+1} - x_n}{(x_n - x_{n-1})^p} \right\|$	η
$GK1$	0	1.8	2.0(−4)	4.9(−2)			
	1	1.750895258580091535641280	2.5(−8)	9.0(−4)		6.385691220(+4)	3.536522620(+7)
	2	1.750000000014299761415271	6.1(−24)	1.4(−11)		2.777396484(+7)	
	3	1.750000000000000000000000	2.7(−117)	3.0(−58)	5.9816	3.536522620(+7)	
$GK2$	0	1.8	2.0(−4)	5.0(−2)			
	1	1.750388172793891559741273	4.6(−9)	3.9(−4)		2.603237303(+4)	3.215020576(+6)
	2	1.750000000000010343224637	3.2(−30)	1.0(−14)		3.023468138(+6)	
	3	1.750000000000000000000000	4.6(−157)	3.9(−78)	5.9959	3.215020576(+6)	
OM	0	1.8	2.0(−4)	4.9(−2)			
	1	1.750388172319823575363680	9.9(−9)	5.7(−4)		1.599594295(+7)	1.462834362(+11)
	2	1.750000000000001356336629	5.5(−32)	1.4(−15)		3.750857339(+11)	
	3	1.750000000000000000000000	8.4(−218)	1.7(−108)	7.9903	1.462834362(+11)	
ZM	0	1.8	2.0(−4)	5.0(−2)			
	1	1.750388172319823575363680	4.6(−9)	3.9(−4)		1.057651892(+7)	1.178394347(+11)
	2	1.750000000000000051608567	8.0(−35)	5.2(−17)		1.001210273(+11)	
	3	1.750000000000000000000000	1.1(−240)	5.9(−120)	7.9928	1.178394347(+11)	
$MM1$	0	1.8	2.0(−4)	5.0(−2)			
	1	1.750078744729477065897963	1.9(−10)	7.9(−5)		2.041444221(+6)	1.754865398(+10)
	2	1.750000000000000000000000	1.9(−47)	2.5(−23)		1.705919057(+10)	
	3	1.750000000000000000000000	2.5(−343)	2.9(−171)	7.9991	1.754865398(+10)	
$MM2$	0	1.8	2.0(−4)	5.0(−2)			
	1	1.750023647624207742848767	1.7(−11)	2.4(−5)		6.076745870(+5)	2.545224623(+9)
	2	1.750000000000000000000000	1.8(−57)	2.5(−28)		2.526328798(+9)	
	3	1.750000000000000000000000	3.7(−425)	3.5(−212)	7.9998	2.545224623(+9)	
$MM3$	0	1.8	2.0(−4)	5.0(−2)			
	1	1.750031099258857162422275	2.9(−11)	3.1(−5)		8.001136411(+5)	2.11655213(+9)
	2	1.750000000000000000000000	1.0(−55)	1.8(−27)		2.095705097(+9)	
	3	1.750000000000000000000000	2.2(−411)	2.7(−205)	7.9997	2.11655213(+9)	

Table 3.2: Convergence behavior of different iterative methods on the test function $f_2(x)$

Methods	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$	ρ	$\left\lvert \frac{x_{n+1}-x_n}{(x_n-x_{n-1})^p} \right\rvert$	η
$GK1$	0	0.76	*	*			
	1		*	*		*	*
	2		*	*		*	
	3		*	*	*	*	
$GK2$	0	0.76	2.2(−1)	2.6(−3)			
	1	0.7573962460753336221899798	1.4(−8)	1.8(−10)		5.725910242(+5)	5.257130496(+5)
	2	0.7573962462537538794596413	1.4(−51)	1.7(−53)		5.257130467(+5)	
	3	0.7573962462537538794596413	1.0(−309)	1.3(−311)	6.0000	5.257130496(+5)	
OM	0	0.76	2.2(−1)	2.6(−3)			
	1	0.7573962463137703385994168	4.8(−9)	6.0(−11)		2.840999693(+10)	3.013467463(+10)
	2	0.7573962462537538794596413	4.0(−70)	5.1(−72)		3.013467461(+10)	
	3	0.7573962462537538794596413	1.1(−558)	1.3(−560)	8.0000	3.013467463(+10)	
ZM	0	0.76	2.2(−1)	2.6(−3)			
	1	0.7573962463048948508621891	4.1(−9)	5.1(−11)		2.420860580(+10)	3.421344786(+10)
	2	0.7573962462537538794596413	1.3(−70)	1.6(−72)		3.421344762(+10)	
	3	0.7573962462537538794596413	1.2(−562)	1.5(−564)	8.0000	3.421344786(+10)	
$MM1$	0	0.76	2.2(−1)	2.6(−3)			
	1	0.7573962462529556670756109	6.4(−11)	8.0(−13)		3.778498010(+8)	3.801147141(+8)
	2	0.7573962462537538794596413	5.0(−87)	6.3(−89)		3.801147141(+8)	
	3	0.7573962462537538794596413	7.2(−696)	9.0(−698)	8.0000	3.801147141(+8)	
$MM2$	0	0.76	2.2(−1)	2.6(−3)			
	1	0.7573962462537861829618272	2.6(−12)	3.2(−14)		1.529150906(+7)	1.263385529(+7)
	2	0.7573962462537538794596413	1.2(−99)	1.5(−101)		1.263385529(+7)	
	3	0.7573962462537538794596413	2.6(−798)	3.2(−800)	8.0000	1.263385529(+7)	
$MM3$	0	0.76	2.2(−1)	2.6(−3)			
	1	0.7573962462537905009805658	2.9(−12)	3.7(−14)		1.733552963(+7)	1.446059282(+7)
	2	0.7573962462537538794596413	3.7(−99)	4.7(−101)		1.446059282(+7)	
	3	0.7573962462537538794596413	2.1(−794)	3.3(−796)	8.0000	1.446059282(+7)	

Table 3.3: Convergence behavior of different iterative methods on the test function $f_3(x)$

Methods	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$	ρ	$\left\lvert\frac{x_{n+1}-x_n}{(x_n-x_{n-1})^p}\right\rvert$	η
$GK1$	0	−3.0	4.7(−2)	1.5(−1)			
	1	−2.850032149435759899649078	2.2(−9)	3.2(−5)		2.826079363(+0)	4.198827967(−5)
	2	−2.850000000000000000000000	4.5(−63)	4.6(−32)		4.191188565(−5)	
	3	−2.850000000000000000000000	3.6(−385)	4.1(−193)	6.0000	4.198827967(−5)	
$GK2$	0	−3.0	4.7(−2)	1.5(−1)			
	1	−2.845530536829933778640841	4.2(−5)	4.5(−3)		3.291554609(+2)	1.360955722(−3)
	2	−2.850002074441970615144759	9.0(−12)	2.1(−6)		2.595135041(+8)	
	3	−2.850000000000000000000000	2.5(−74)	1.1(−37)	9.3846	1.360955722(−3)	
OM	0	−3.0	4.7(−2)	1.6(−1)			
	1	−2.844042602118935658056506	7.5(−5)	6.0(−3)		1.703635813(+4)	6.783289282(−4)
	2	−2.850005050121091781574571	5.4(−11)	5.1(−6)		3.161585672(+12)	
	3	−2.850000000000000000000000	1.7(−91)	2.9(−46)	13.102	6.783289282(−4)	
ZM	0	−3.0	4.7(−2)	1.6(−1)			
	1	−2.840827596075196247341513	1.8(−4)	9.2(−3)		2.230697732(+4)	3.402776481(−4)
	2	−2.850019022777759525868734	7.6(−10)	1.9(−5)		3.734311208(+11)	
	3	−2.850000000000000000000000	7.1(−83)	5.8(−42)	13.609	3.402776481(−4)	
$MM1$	0	−3.0	4.7(−2)	1.5(−1)			
	1	−2.847981610389184901653897	8.6(−6)	2.0(−3)		7.077438026(+3)	9.674841648(−4)
	2	−2.850000186344162045752608	7.3(−14)	1.9(−7)		6.760129660(+14)	
	3	−2.850000000000000000000000	4.2(−114)	1.4(−57)	12.423	9.674841648(−4)	
$MM2$	0	−3.0	4.7(−2)	1.5(−1)			
	1	−2.847982098238578815439951	8.6(−6)	2.0(−3)		7.075908888(+3)	9.675353587(−4)
	2	−2.850000186252333907669066	7.3(−14)	1.9(−7)		6.769878897(+14)	
	3	−2.850000000000000000000000	4.1(−114)	1.4(−57)	12.423	9.675353587(−4)	
$MM3$	0	−3.0	4.7(−2)	1.5(−1)			
	1	−2.847981540231008673038257	8.6(−6)	2.0(−3)		7.077657926(+3)	9.675368192(−4)
	2	−2.850000186357369916831709	7.3(−14)	1.9(−7)		6.758728963(+14)	
	3	−2.850000000000000000000000	4.2(−114)	1.4(−57)	12.423	9.675368192(−4)	

Table 3.4: Convergence behavior of different iterative methods on the test function $f_4(x)$

Methods	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$	ρ	$\left\lvert \frac{x_{n+1}-x_n}{(x_n-x_{n-1})^p} \right\rvert$	η
$GK1$	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000002777374011867781357	1.1(−254)	2.8(−6)		2.777836885(+00)	5.504789671(+00)
	2	2.000000000000000000000000	9.6(−1607)	2.5(−33)		5.504677538(+00)	
	3	2.000000000000000000000000	4.5(−9719)	1.4(−195)	6.0000	5.504789671(+00)	
$GK2$	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000000200989638086020762	1.0(−311)	2.0(−7)		2.009920619(−1)	2.777777778(−1)
	2	2.000000000000000000000000	9.8(−2014)	1.8(−41)		2.777775861(−1)	
	3	2.000000000000000000000000	7.3(−12226)	1.0(−245)	6.0000	2.777777778(−1)	
OM	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000000785189010712446522	4.0(−282)	7.9(−7)		7.852383342(+1)	2.269259259(+2)
	2	2.000000000000000000000000	4.4(−2301)	3.3(−47)		2.269242109(+2)	
	3	2.000000000000000000000000	8.3(−18453)	3.0(−370)	8.0000	2.269259259(+2)	
ZM	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000000477890417235498042	6.6(−293)	4.8(−7)		4.779086880(+1)	2.084074074(+2)
	2	2.000000000000000000000000	3.4(−2389)	5.7(−49)		2.084057463(+2)	
	3	2.000000000000000000000000	1.6(−19159)	2.2(−384)	8.0000	2.084074074(+2)	
$MM1$	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000000073305887479606243	1.3(−333)	7.3(−8)		7.330631738(+00)	2.066666667(+01)
	2	2.000000000000000000000000	4.7(−2765)	1.7(−56)		2.066664998(+01)	
	3	2.000000000000000000000000	1.5(−22216)	1.6(−445)	8.0000	2.066666667(+01)	
$MM2$	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000000001927516381664629	1.3(−412)	1.9(−9)		1.927516679(−1)	1.326315789(+00)
	2	2.000000000000000000000000	9.7(−3457)	2.5(−70)		1.326315770(+00)	
	3	2.000000000000000000000000	1.1(−27809)	2.2(−557)	8.0000	1.326315789(+00)	
$MM3$	0	2.1	9.8(−25)	1.0(−1)			
	1	2.000000006966462333292930	1.0(−384)	7.0(−9)		6.966466216(−1)	1.466666667(+00)
	2	2.000000000000000000000000	2.4(−3231)	8.1(−66)		1.466666588(+00)	
	3	2.000000000000000000000000	2.2(−26004)	2.8(−521)	8.0000	1.466666667(+00)	

Table 3.5: Convergence behavior of different iterative methods on the test function $f_5(x)$.

Methods	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$	ρ	$\left\lvert\frac{x_{n+1}-x_n}{(x_n-x_{n-1})^p}\right\rvert$	η
$GK1$	0	5.0	$*$	$*$			
	1		$*$	$*$		$*$	$*$
	2		$*$	$*$		$*$	
	3		$*$	$*$	$*$	$*$	
$GK2$	0	5.0	6.7(−3)	3.5(−2)			
	1	4.9651142317442777568317118	2.4(−16)	1.3(−15)		7.015679382(−7)	7.468020979(−7)
	2	4.9651142317442776303698759	5.9(−97)	3.1(−96)		7.468020979(−7)	
	3	4.9651142317442776303698759	1.2(−580)	6.1(−580)	6.0000	7.468020979(−7)	
OM	0	5.0	6.7(−3)	3.5(−2)			
	1	4.9651142317442776303744811	8.9(−21)	4.6(−20)		2.099233812(−8)	2.312146664(−8)
	2	4.9651142317442776303698759	9.0(−164)	4.7(−163)		2.312146664(−8)	
	3	4.9651142317442776303698759	1.0(−1307)	5.3(−1307)	8.0000	2.312146664(−8)	
ZM	0	5.0	6.7(−3)	3.5(−2)			
	1	4.9651142317442776303727319	5.5(−21)	2.9(−20)		1.301869270(−8)	1.435568470(−8)
	2	4.9651142317442776303698759	1.2(−165)	6.4(−165)		1.435568470(−8)	
	3	4.9651142317442776303698759	7.4(−1323)	3.8(−1322)	8.0000	1.435568470(−8)	
$MM1$	0	5.0	6.7(−3)	3.5(−2)			
	1	4.9651142317442776303681372	3.4(−21)	1.7(−20)		7.925586248(−9)	8.529952965(−9)
	2	4.9651142317442776303698759	1.4(−167)	7.1(−167)		8.529952965(−9)	
	3	4.9651142317442776303698759	1.1(−1338)	5.7(−1338)	8.0000	8.529952965(−9)	
$MM2$	0	5.0	6.7(−3)	3.5(−2)			
	1	4.9651142317442776303680705	3.5(−21)	1.8(−20)		8.229748515(−9)	8.881048023(−9)
	2	4.9651142317442776303698759	1.9(−167)	1.0(−166)		8.881048023(−9)	
	3	4.9651142317442776303698759	1.7(−1337)	9.1(−1337)	8.0000	8.881048023(−9)	
$MM3$	0	5.0	6.7(−3)	3.5(−2)			
	1	4.9651142317442776303680702	3.5(−21)	1.8(−20)		8.231154237(−9)	8.882681023(−9)
	2	4.9651142317442776303698759	1.9(−167)	1.0(−166)		8.882681023(−9)	
	3	4.9651142317442776303698759	1.8(−1337)	9.2(−1337)	8.0000	8.882681023(−9)	

Table 3.6: Convergence behavior of different iterative methods on the test function $f_6(x)$.

Cases	n	x_n	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$	ρ	$\left\lvert \frac{x_{n+1}-x_n}{(x_n-x_{n-1})^p} \right\rvert$	η
$GK1$	0	−412	*	*			
	1		*	*		*	*
	2		*	*		*	
	3		*	*	*	*	
$GK2$	0	−412	1.3(+8)	8.5(−1)			
	1	−411.1521869660545671537300	9.6(−5)	6.1(−13)		1.636932403(−12)	1.668249119(−12)
	2	−411.1521869660539592549395	1.3(−77)	8.4(−86)		1.668249119(−12)	
	3	−411.1521869660539592549395	9.4(−515)	5.9(−523)	6.0000	1.668249119(−12)	
OM	0	−412	1.3(+8)	8.5(−1)			
	1	−411.1521869660539699457729	1.7(−6)	1.1(−14)		4.005076808(−14)	4.198566741(−14)
	2	−411.1521869660539592549395	1.1(−117)	7.2(−126)		4.198566741(−14)	
	3	−411.1521869660539592549395	4.6(−1007)	2.9(−1015)	8.0000	4.198566741(−14)	
ZM	0	−412	1.3(+8)	8.5(−1)			
	1	−411.1521869660539687486432	1.5(−6)	9.5(−15)		3.556599499(−14)	3.852323642(−14)
	2	−411.1521869660539592549395	4.0(−118)	2.5(−126)		3.852323642(−14)	
	3	−411.1521869660539592549395	1.1(−1010)	6.7(−1019)	8.0000	3.852323642(−14)	
$MM1$	0	−412	1.3(+8)	8.5(−1)			
	1	−411.1521869660539602280746	1.5(−7)	9.7(−16)		3.645628543(−15)	3.846055662(−15)
	2	−411.1521869660539592549395	4.9(−127)	3.1(−135)		3.846055662(−15)	
	3	−411.1521869660539592549395	5.1(−1083)	3.2(−1091)	8.0000	3.846055662(−15)	
$MM2$	0	−412	1.3(+8)	8.5(−1)			
	1	−411.1521869660539593310835	1.2(−8)	7.6(−17)		2.852561685(−16)	2.964107964(−16)
	2	−411.1521869660539592549395	5.3(−137)	3.3(−145)		2.964107964(−16)	
	3	−411.1521869660539592549395	7.4(−1164)	4.7(−1172)	8.0000	2.964107964(−16)	
$MM3$	0	−412	1.3(+8)	8.5(−1)			
	1	−411.1521869660539593268602	1.1(−8)	7.2(−17)		2.694344101(−16)	2.799008203(−16)
	2	−411.1521869660539592549395	3.2(−137)	2.0(−145)		2.799008203(−16)	
	3	−411.1521869660539592549395	1.2(−1165)	7.3(−1174)	8.0000	2.799008203(−16)	

Bibliography

[1] Petkovic, M.S., Neta, B.,:(2014)" Multipoint methods for solving nonlinear equations". *Applied Mathematics and Computation*, **226**, 635-660.

[2] Sharma, J.R., Guha, R.K., Sharma,R.: (2012)"A unified approach to generate weighted Newton third order methods for solving nonlinear equations".*J.Numer.Mathe.Stoch.*,**4(1)**,48-58.

[3] Chen,J.Y., Kincaid,D.R., Lin,B.R.:(2017) "A variant of Newton's method based on Simpson's three-eighths rule for nonlinear equations".*Appl.Math.Lett.*.

[4] Noor,M.A.,Noor,K.I.,Al-Said,E.,Waseem,M.:(2010)"Some new iterative methods for nonlinear equations.".*Math.Prob.Eng.*.

[5] Reza Dehghan :(2019)"A new iterative method for finding the multiple roots of nonlinear equations".*African Mathematical Union and Springer-Verlag GmbH Deutschland, ein Teil von Springer Nature 2019.*

[6] Choubey, N., Jaiswal, J.P.(2015):" A derivative-free method of eighth-order for finding simple root of nonlinear equations."*Commun. Numer. Anal.* 2015(2), 90–103.

[7] Mehdi Salimi,N. M. A. Nik Long,,Bruno Antonio Pansera.:(2017)"A multi-point iterative method for solving nonlinea requations with optimal order of convergence".*The JJIAM Publishing Committee and Springer Japan KK, part of Springer Nature.*

[8] Zhoua, X., Chena, X., Songa, Y.:(2011)" Constructing higher-order methods for obtaining the multiple roots of nonlinear equations".*J. Comput. Appl. Math.*.**235**, 4199–4206.

[9] Dong, C.:(1987)"A family of multipoint iterative functions for finding multiple roots of equations".*Int. J. Comput. Math.***21**,363–367.

[10] Sharma, R.,Bahl Ashu.:(2015) "An Optimal Fourth Order Iterative Method for Solving Nonlinear Equations and Its Dynamics".*Journal of Complex Analysis.*

[11] O zban, A.Y.:(2004)" Some new variants of Newton's method" *Appl. Math. Lett.***17(6)**, 677–682.

[12] King, R.F.:(1973)" A family of fourth order methods for nonlinear equations" *SIAM J. Numer. Anal.***10**, 876–879.

[13] Rostamy,D.,Bakhtiari,P.:(2015)"New efficient multipoint iterative methods for solving nonlinear systems" *Applied Mathematics and Computation*,**266**,350-356.

[14] Khan,Y.,Fardi,M.,Sayevand,K.:(2012)"A new general eighth-order family of iterative methods for solving nonlinear equations"*Applied Mathematics Letters*,**25**,2262-2266.

[15] Weihong, Bi., Hongmin, Ren., Qingbiao, Wu.:(2009)"Three-step iterative methods with eighth-order convergence for solving nonlinear equations"*Journal of Computational and Applied Mathematics***225**Issue 1,105-112