Communications in Mathematics and Applications
Vol. 15, No. 1, pp. 9–31, 2024
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v15i1.2432

Research Article

Two Novel With and Without Memory Multi-Point Iterative Methods for Solving Non-Linear Equations

Shahid Abdullah*, Neha Choubey and Suresh Dara

School of Advanced Sciences and Languages, VIT Bhopal University, Kothri-kalan, Sehore, 466114, Madhya Pradesh, India
***Corresponding author:** shahidibnabdullah786@gmail.com

Received: September 25, 2023 **Accepted:** December 27, 2023

Abstract. In this work, two new iterative methods are proposed for finding simple roots of non-linear equations. The new methods are the modifications of the existing work proposed by Rafiullah and Jabeen (New eighth and sixteenth order iterative methods to solve nonlinear equations, *International Journal of Applied and Computational Mathematics* **3** (2017), 2467 – 2476). The first method obtained is of fifth-order two-step with memory method while the second scheme is three-point eight-order optimal without memory method. Firstly, the Hermite interpolation polynomial is employed to eliminate the first derivative. To maintain order, the conversion to a memory scheme was accomplished by introducing self-accelerated parameters, all without requiring any new function evaluations. Additionally, the Gauss quadrature approach was incorporated for the first derivative, aiming to attain optimal eighth-order convergence. In particular, the efficiency index is increased from 1.4953 to 1.7099 and 1.5157 to 1.6817 for fifth- and eighth-orders respectively. Some real-life application-based problems, such as Kepler's equation, an ocean engineering problem, Planck's radiation law, a blood rheology model, and the charge between two parallel plates were presented to validate and demonstrate the superiority of the proposed scheme. Another benefit of the proposed scheme is on the restriction of the Newton's method that $f'(v) \neq 0$ can be eliminated close to the root.

Keywords. Iterative method, Non-linear equation, With memory scheme, Hermite interpolation polynomial, Gauss quadrature approach, Efficiency index

Mathematics Subject Classification (2020). 35A01, 65L10, 65L12, 65L20, 65L70

1. Introduction

In many contemporary scientific disciplines, especially in the realms of computational and applied mathematics, the usage of root-finding algorithms has grown importance. In this day and age, modern computer programs such as MATLAB, SAGEMATH, MAPLE, MATHCAD and MATHEMATICA among others can be used to perform a wide range of root-finding techniques. In order to find the simple root of non-linear equations of the type

$$f(v) = 0, \tag{1}$$

where $f : D \subset R \to R$ is a scalar function on an open interval D. The search for the polynomial's roots is the most significant among them, since it has an immense impact on a variety of other domains of modern research in addition to applied and computational mathematics. Mathematical study alone does not satisfy the need to solve nonlinear equations due to the fact that many partial, ordinary, integral, and integro-differential equations frequently result in systems of nonlinear equations after being discretized. Generally speaking, nonlinear phenomena are constructed using models containing nonlinear equations, and they are not just found in academics. These equations are used frequently in daily life for tasks like measuring rocket speed determining a system's eigenvalues, computing the simple harmonic oscillation (Avallone and Baumeister [3]), determining the compressibility of gases (Chaudhary *et al*. [5]), measuring the variation of the local heat (Jaturonglumlert and Kiatsiriroat [19]), discussing the aging model of a cell's energy-producing organelle (mitochondria) etc. Technical problems of this nature often cannot be solved analytically, demanding the employment of iterative techniques. They must therefore be managed using numerical methods having higher order of convergence and reasonable computational cost. In this way, Newton's method (Ortega [33]) is one of the oldest technique for solving such equations given by:

$$v_{n+1} = v_n - \frac{f(v_n)}{f'(v_n)}. \tag{2}$$

Traub [50] demonstrated that Scheme (2) possesses the second-order of convergence.

Recently, the research and development of multi-point iterative algorithms have recently attracted more attention to obtain higher order of convergence with decrease in computational cost. Nowadays, lot of attention is given to multi-point iterative methods with memory that employs one or more self accelerated parameters, e.g, Abdullah *et al*. [1], Choubey and Jaiswal [7], Choubey *et al*. [10], Kansal *et al*. [20], and Wang and Zhang [54], etc. to accelearte order of convergence without any extra function evaluations. Furthermore, the most popular approaches for comparing the effectiveness of iterative methods is the efficiency index, which can be calculated using the formula $r^{\frac{1}{q}}$, where r is the iterative scheme's convergence order and q is the number of functions that must be found during each iteration. As mentioned by Kung and Traub [26] in his conjecture that the iterative scheme is optimal if fulfills the convergence order as 2^{q-1} where q represents number of functional evaluations. Several authors have developed efficient iterative algorithms of various kinds. The composition technique is the preferred approach for building an optimal method, along with the use of various approximations and interpolations to reduce the number of functional evaluations. Various fourth- and eighth order optimal iterative algorithms have been developed, see e.g., Chun and Lee [11], Kung and Traub [26], Liu and Wang [28], Pandey *et al*. [35], Parimala *et al*. [36], Rafiullah and Jabeen [40], Sharma and Arora [45] and the references cited therein.

In this paper, we proposed a new two-point with memory scheme of fifth order and an optimal eighth-order without memory scheme. The main goal for employing these techniques is to construct an efficient root-finding approaches and to develop an approximation for derivatives. The new methods are the modifications of the existing work proposed by Rafiullah and Jabeen [40]. The primary driving force behind this research can be attributed by combining Hermite Interpolation polynomial to remove first derivative. Furthermore, self accelerated parameters are employed to preserve order of convergence without any new functional evaluations.In addition, we combine *Gauss Quadrature* approach for the first derivative, to achieve the eighth-order of convergence with optimality. The rest of the paper is discussed as follows. In Section 2 and its subsections, the new schemes and their convergence analysis is illustrated. In Section 3, some application based problems are provided to illustrate theoretical findings. Section 4 ends with the conclusion.

2. Construction of the Proposed Methods

Rafiullah [39] proposed the following iterative method without memory for solving non linear equations as:

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{f'(u_n)} - \frac{1}{2}\left[\frac{f(u_n)}{f'(v_n)}\right]^2 \frac{f''(u_n)}{f'(u_n)}. \end{aligned}\right\} \tag{3}$$

The iterative method (3) denoted by $MR1$ involves five evaluations of the function having an efficiency index $r^{\frac{1}{f}} = 5^{\frac{1}{5}} = 1.3797$. To reduce the number of function evaluations, Rafiullah and Jabeen [40] modified (3) and derived two new iterative methods of fifth and eighth-order without memory methods. Finite difference and Lagrange interpolation were utilized to remove $f''(u_n)$ and $f'(w_n)$ respectively and is given by:

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{f'(u_n)} - \frac{f(u_n)^2(f'(v_n)-f'(u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2} \end{aligned}\right\} \tag{4}$$

and

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)}, \\ w_n &= u_n - \frac{f(u_n)}{f'(u_n)} - \frac{f(u_n)^2(f'(v_n)-f'(u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2}, \\ v_{n+1} &= w_n - \frac{f(w_n)((v_n-u_n)(v_n-w_n)(u_n-w_n))}{-f(w_n)(v_n-u_n)(v_n-2w_n-u_n)+f(u_n)(v_n-w_n)^2-f(u_n)(v_n-z_n)^2}. \end{aligned}\right\} \tag{5}$$

The method defined by (4) denoted by $MR2$ holds the order as five and involves four function evaluations with efficiency index $5^{\frac{1}{4}} = 1.\ 4953$ while (5) is of eighth order involving five function evaluations per cycle with efficiency index $8^{\frac{1}{5}} = 1.5157$.

Removing one more derivative in (4) with respect to y plays a significant role to develop an optimal iterative methods of order four. At this point, we will approximate $f'(u_n)$ by the use Hermite interpolating polynomial of second degree. Towards this end, let us suppose that

$$R(k)=\sum_{j=0}^{2} c_j(k-u_n)^j, \tag{6}$$

$$R'(k)=c_1+2c_2(k-u_n), \tag{7}$$

where c_0, c_1 and c_2 are unknown parameters and can be found by applying the following conditions as

$$R(v_n)=f(v_n),\ R'(v_n)=f'(v_n),\ R(u_n)=f(u_n) \text{ and } R'(u_n)=f'(u_n).$$

While utilizing the interpolation conditions, the following systems of equations are obtained:

$$\left.\begin{aligned}
&R(v_n)=c_0=f(v_n),\\
&R'(v_n)=c_1=f'(v_n),\\
&R(u_n)=c_0+c_1(u_n-v_n)+c_2(u_n-v_n)^2=f(u_n),\\
&R'(u_n)=c_1+2c_2(u_n-v_n)=f'(u_n).
\end{aligned}\right\} \tag{8}$$

On solving (8), the desired outcome is:

$$f'(u_n)=\left(2\frac{f(u_n)-f(v_n)}{u_n-v_n}\right)-f'(v_n)=R(v_n,u_n). \tag{9}$$

So, the iterative method (4) and (5) becomes:

$$\left.\begin{aligned}
&u_n=v_n-\frac{f(v_n)}{f'(v_n)},\\
&v_{n+1}=u_n-\frac{f(u_n)}{R(v_n,u_n)}-\frac{f(u_n)^2(f'(v_n)-R(v_n,u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2}.
\end{aligned}\right\} \tag{10}$$

The iterative method defined by (10) is an optimal fourth-order method with only three function evaluations. The error equation of (10) is:

$$-d_2d_3e_n^4+O(e_n^5). \tag{11}$$

Since, the main motive of this study is to minimize the cost of function evaluations of (4) while preserving the order of convergence. So, we attempt to hold the order of convergence as five while converting (10) to with memory scheme without any new functional evaluation. For that purpose, parameter T is added in the initial step of method defined by (10) as:

$$\left.\begin{aligned}
&u_n=v_n-\frac{f(v_n)}{f'(v_n)-Tf(v_n)},\\
&v_{n+1}=u_n-\frac{f(u_n)}{R(v_n-u_n)}-\frac{f(u_n)^2(f'(v_n)-R(v_n-u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2}.
\end{aligned}\right\} \tag{12}$$

The error expression for the iterative scheme (12) is as follows:

$$\begin{aligned}
e_{n,u}&=u_n-\beta\\
&=(d_2-T)e_n^2+(-2d_2^2-T^2+2d_2T+2d_3)e_n^3+(T^3+5Td_2^2-4d_2^3\\
&\quad-4Tc_3+d_2(7d_3-3T^2)-3d_4)e_n^4+O(e_n^5),
\end{aligned} \tag{13}$$

$$\begin{aligned}
e_{n,v}&=v_n-\beta\\
&=(T-d_2)d_3e_n^4+(-10Td_2^3+5d_2^5+(T^2-2d_3)d_3+d_2^2(5T^2+2d_3)\\
&\quad+2Td_4-2d_2(Td_3+d_4))e_n^5+O(e_n^6).
\end{aligned} \tag{14}$$

where $e_{n,v} = v_v - \beta, e_{n,u} = u_n - \beta$ and $c_i = \frac{f^{(i)}(\beta)}{i!f'(\beta)}$ for $i = 2,3,\dots$ and $T \in R$. It can be observed that order of convergence remains consistent when $T \neq d_2$. We assume $T = d_2 = \frac{f''(\beta)}{f'(\beta)}$ to improve order of convergence and will the parameter T with T_n. Now, T_n is evaluated by the use of current and previous information in such a way that satisfies the condition $\lim_{n\to\infty} T_n = d_2 = f''(\beta)/2f'(\beta)$ such that fourth asymptotic convergence in the error expression (14) should be zero.

First, we consider distinct values of T are considered to improve order of (14) as fallows:

Method 1:

$$T_n = \frac{H_2''(v_n)}{2f'(v_n)}, \tag{15}$$

where $H_2(v) = f(v_n) + f[v_n, v_n](v - v_n) + f[v_n, v_n, u_{n-1}](v - v_n)^2$ and $H_2''(v) = 2f[v_n, v_n, u_{n-1}]$.

Method 2:

$$T_n = \frac{H_3''(v_n)}{2f'(v_n)} \tag{16}$$

where $H_3(v) = H_2(v) + f[v_n, v_n, u_{n-1}, v_{n-1}](v - v_n)^2(v - u_{n-1})$ and $H_3''(v) = 2f[v_n, v_n, u_{n-1}] + 2f[v_n, v_n, u_{n-1}, u_{n-1}](v_n - u_{n-1})$.

Method 3:

$$T_n = \frac{H_4''(v_n)}{2f'(v_n)} \tag{17}$$

where $H_4(v) = H_3(v) + f[v_n, v_n, u_{n-1}, v_{n-1}, v_{n-1}](v - v_n)^2(v - u_{n-1})(v - v_{n-1})$ and $H_4''(v_n) = 2f[v_n, v_n, u_{n-1}] - (2f[v_n, u_{n-1}, v_{n-1}, v_{n-1}](v_n - u_{n-1}) - 4f[v_n, v_n, u_{n-1}, v_{n-1}])$.

To obtain a with memory scheme, we replace T with T_n as:

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n) - T_n f(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{R(v_n - u_n)} - \frac{f(u_n)^2(f'(v_n) - R(v_n - u_n))}{2(f(v_n) - f(u_n))f'(v_n)^2}. \end{aligned}\right\} \tag{18}$$

The scheme (18) is represented by $AS1$.

Note. The condition $H_s'(v_n) = f'(v_n)$ fulfilled by the Hermite interpolation polynomial $H_s(v)$ for $s = 2,3,4$. So, $T_n = \frac{H_s''(v_n)}{2'(v_n)}$ can be expressed as $T_n = \frac{H_s''(v_n)}{2H_s'(v_n)}$ $(s = 2,3,4)$.

Theorem 2.1. *Let s be the degree of the Hermite interpolating polynomial H_s that interpolates a function f at interpolation nodes $v_n, v_n, l_0, \dots, l_{s-2}$ embedded in an interval I and the derivative $f^{(s+1)}$ is continuous in I and the Hermite polynomial $H_s(v_n) = f(v_n)$, $H_s'(v_n) = f'(v_n)$, $H_s(l_i) = f(l_i)$ $(i = 0,1,2,\dots,s-2)$. Let $e_{l,i} = l_i - \beta$ $(i = 0,1,2,\dots,s-2)$ denote the errors and assume that*

(i) *all nodes $v_n, l_0, \dots, l_{s-2}$ are fairly close to the root β.*

(ii) *The condition holds $e_n = O(e_{l,0}, \dots, e_{l,s-2})$ holds. Then*

$$H_s''(v_n) = 2f'(\beta)\Big(c_2 - (-1)^{s-1} c_{s+1} \prod_{i=0}^{s-2} e_{l,i} + 3d_3 e_n\Big), \tag{19}$$

$$T_n = \frac{H_s''(v_n)}{2f'(v_n)} \sim \left(d_2 - (-1)^{s-1} c_{s+1} \prod_{i=0}^{s-2} e_{l,i} + (3d_3 - 2d_2^2)e_n\right) \tag{20}$$

and

$$T_n - c_2 \sim \left(-(-1)^{s-1} c_{s+1} \prod_{i=0}^{s-2} e_{l,i} + (3d_3 - 2d_2^2)e_n\right). \tag{21}$$

Proof. The error equation of the Hermite interpolation polynomial is computed as fallows:

$$f(v) - H_s(v) = \frac{f^{(s+1)}(\psi)}{(s+1)!}(v - v_n)^2 \prod_{i=0}^{s-2}(v - l_i), \quad (\psi \in I). \tag{22}$$

On differentiating equation (22) twice at point $v = v_n$, we have

$$H_s''(v_n) = f''(v_n) - 2\frac{f^{(s+1)}(\psi)}{(s+1)!} \prod_{i=0}^{s-2}(v_n - l_i), \quad (\psi \in I). \tag{23}$$

At point $v_n \in I$ and $\psi \in I$ the Taylor's series expansion of f' about the zero β of f provides

$$f'(v_n) = f'(\beta)(1 + 2d_2 e_n + 3d_3 e_n^2 + O(e_n^3)), \tag{24}$$

$$f''(v_n) = f'(\beta)(2d_2 + 6d_3 e_n + O(e_n^2)) \tag{25}$$

and

$$f^{(s+1)}(\psi) = f'(\beta)((s+1)! c_{s+1} + (s+2)! c_{s+2} e_\psi + O(e_\psi^2)), \tag{26}$$

where $e_\psi = \psi - \beta$. Substituting (25), (26) in (23), we get

$$H_s''(v_n) = 2f'(\beta)\left(c_2 - (-1)^{(s-1)} c_{s+1} \prod_{j=0}^{s-2} e_{p,j} + 3c_3 e_n\right) \tag{27}$$

implies that

$$\frac{H_s''(v_n)}{2f'(v_n)} \sim \left(d_2 - (-1)^{(s-1)} d_{s+1} \prod_{i=0}^{s-2} e_{l,i} + (3d_3 - 2d_2^2)e_n\right), \tag{28}$$

hence

$$T_n \sim \left(d_2 - (-1)^{s-1} c_{s+1} \prod_{i=0}^{s-2} e_{l,i} + (3d_3 - 2d_2^2)e_n\right) \tag{29}$$

or

$$T_n - d_2 \sim \left(-(-1)^{s-1} d_{s+1} \prod_{i=0}^{s-2} e_{l,i} + (3d_3 - 2d_2^2)e_n\right). \tag{30}$$

To determine the convergence order of an iterative technique (18), we apply the idea of R-order of convergence (Ortega and Rheinboldt [34]) and the following statement (Alefeld and Herzberger [2]). □

Theorem 2.2. *If the approximation errors $e_i = v_i - \beta$ acquired in an iterative root finding method IM satisfy*

$$e_{q+1} \sim \prod_{j=0}^{s-2}(e_{q-j})^{s_i}, \quad q \geq q(\{e_q\}),$$

then the inequality is fulfilled $O_R(IM, \beta) \geq m^*$ where $O_R(IM, \beta)$ is the R-order of convergence of iterative method and m^* is unique positive solution of the expression $m_{n+1} - \sum_{j=0}^{n} s_i m^{n-i} = 0$.

For the new iterative scheme with memory (18), the following convergence theorem can be established.

Theorem 2.3. *Let T_n which is determined by* (15) *to* (17) *in the iterative technique iterative scheme* (18) *be the variable parameter. If an initial approximation v_0 is close enough to simple root β of $f(v)$ then for the iterative methods* (18) *the R-order of convergence for the corresponding expressions* (15) *to* (17) *of T_n with memory is at least* 4.56*,* 4.79*, and* 5*, respectively.*

Proof. Let $\{v_n\}$ be the sequence formed by the *Iterative Method* (*IM*) converging to β of $f(v)$ with R-order of convergence as $O_R(IM,\beta) \geq x$, then

$$e_{n+1} \sim D_{n,x} e_n^x . \tag{31}$$

By taking $n \to \infty$ then the asymptotic error constant D_x of *Iterative Method* (*IM*) will tend to $D_{n,x}$ and hence

$$e_{n+1} \sim D_{n,x}(D_{n-1,x} e_{n-1}^x)^x = D_{n,x}(D_{n-1,x} e_{n-1}^{x^2}). \tag{32}$$

With the aid of (15)-(17) and T_n, the error equation of with memory iterative scheme (15) can be constructed as:

$$e_{n,u} = u_n - \beta \sim (d_2 - T_n) e_n^2 , \tag{33}$$

$$e_{n+1} = v_n - \beta \sim B_{n,4}(T_n - d_2) e_n^4 , \tag{34}$$

where $B_{n,4}$ is a varying parameter evaluated from (14).

Method 1 (Evaluation of T_n by (15)): The computation of T_n is identical to the derivation of (31). Let us assume that y is the R-order of convergence of the iterative scheme $\{u_n\}$, thus

$$e_{n,u} \sim D_{n,y} e_n^y \sim D_{n,y}(D_{n-1,x} e_{n-1}^x)^y \sim D_{n,x} D_{n-1,y}^x e_{n-1}^{xy} . \tag{35}$$

By applying Theorem 2.1 for $s = 2$ and $l_0 = u_{n-1}$, we get

$$T_n - c_2 \sim d_3 e_{l,0} = d_3 e_{n-1,u} . \tag{36}$$

By using the expressions (33)-(34) and (36), we achieve the following:

$$e_{n,u} \sim -d_3 e_{n-1,u}(D_{n-1,x} e_{n-1}^x)^2 \sim -d_3 D_{n-1,x} D_{n-1,y}^2 e_{n-1}^{2x+y} \tag{37}$$

and

$$\begin{aligned} e_{n+1} &\sim d_3 e_{n-1,u} B_{n,4} e_{n,4} \\ &\sim B_{n,4} d_3 (D_{n-1,y} e_{n-1}^y)(D_{n-1,x} e_{n-1}^x)^4 \\ &\sim B_{n,4} d_3 D_{n-1,y} D_{n-1,x}^4 e_{n-1}^{4x+y} . \end{aligned} \tag{38}$$

By contrasting the powers of e_{n-1} in the relation pairs (35)-(37) and (32)-(38), the following systems of equations are obtained:

$$\left. \begin{aligned} 2x + y &= xy \\ 4x + y &= x^2 \end{aligned} \right\} . \tag{39}$$

$x = 4.5616$ and $y = 2.5615$ satisfy the solution of the system of equations (39). In light of this, when Ln is estimated,the R-order of convergence of the iterative method (35)-(37) is at least 4.5616.

Method 2 (Computation of T_n by (16)): Applying Theorem 2.1 for $s=3$, $l_0=u_{n-1}$ and $l_1=v_{n-1}$, we get

$$T_n - d_2 \sim -d_4 e_{l,0} e_{l,1} = -d_4 e_{n-1,u} e_{n-1}, \tag{40}$$

Now from (33), (34) and (40), we achieve

$$e_{n,u} \sim (d_2 - T_n)e_n^2 \sim d_4 e_{n-1} e_{n-1,u} (D_{n-1,x} e_{n-1}^x)^2 \sim c_4 D_{n-1,y} D_{n-1,x}^2 e_{n-1}^{2x+y+1} \tag{41}$$

and

$$\begin{aligned} e_{n+1} &\sim -B_{n,4} d_4 e_{n-1,u} e_{n-1} e_n^4 \\ &\sim -B_{n,4} c_4 (D_{n-1,y} e_{n-1}^y) e_{n-1} (D_{n-1,x} e_{n-1}^x)^4 \\ &\sim -B_{n,4} c_4 D_{n-1,y} D_{n-1,x}^4 e_{n-1}^{4x+y+1}. \end{aligned} \tag{42}$$

By equating the indices of e_{n-1} with the help of relations (35)-(41) and (32)-(42), resulting equations are obtained as:

$$\left.\begin{aligned} 2x+y+1 &= rp \\ 4x+2y+1 &= r^2 \end{aligned}\right\}. \tag{43}$$

$r = 4.7913$ and $p = 2.7912$ specify the solution to system of equations (43). Therefore, for with memory iterative technique (16)-(18) the R-order of convergence is at least 4.7913.

Method 3 (Computation of T_n by (17)): Applying Theorem 2.1 for $s=4$, $l_0=u_{n-1}$ and $l_1=l_2=v_{n-1}$, we get

$$T_n - d_2 \sim d_5 e_{l,0} e_{l,1} e_{l,2} = d_4 e_{n-1,u} e_{n-1}^2. \tag{44}$$

Now from (33), (34) and (44), we achieve

$$e_{n,u} \sim (d_2 - T_n)e_n^2 \sim -d_5 e_{n-1}^2 e_{n-1,u} (D_{n-1,x} e_{n-1}^x)^2 \sim -c_5 D_{n-1,y} D_{n-1,x}^2 e_{n-1}^{2x+y+2} \tag{45}$$

and

$$\begin{aligned} e_{n+1} &\sim B_{n,4} d_5 e_{n-1,u} e_{n-1}^2 e_n^4 \\ &\sim B_{n,4} d_4 (D_{n-1,y} e_{n-1}^y) e_{n-1}^2 (D_{n-1,x} e_{n-1}^x)^4 \\ &\sim B_{n,4} d_4 D_{n-1,y} D_{n-1,x}^4 e_{n-1}^{4x+y+2}. \end{aligned} \tag{46}$$

By equating the indices of e_{n-1} with the help of relations (35)-(41) and (32)-(42), the resulting equations are obtained as below:

$$\left.\begin{aligned} 2x+y+2 &= rp \\ 4x+y+2 &= r^2 \end{aligned}\right\}. \tag{47}$$

$r = 5.00$ and $p = 3.00$ specify the solution to system of equations (47). Therefore, for with memory iterative technique (17)-(18) the R-order of convergence is at least 4.7913. □

2.1 Optimal Eight-Order Method

In this subsection, we attempt to construct an optimal eighth-order iterative method while preserving the order. Since, the iterative method (5) is of eighth-order with five function evaluations. If we take Kung and Traub's notion of optimality into consideration then (5) denoted by $SS1$ is not optimal given as;

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)}, \\ w_n &= u_n - \frac{f(u_n)}{f'(u_n)} - \frac{f(u_n)^2(f'(v_n)-f'(u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2}, \\ v_{n+1} &= w_n - \frac{f(w_n)((v_n-u_n)(v_n-w_n)(u_n-w_n))}{-f(w_n)(v_n-u_n)(v_n-2w_n-u_n)+f(u_n)(v_n-w_n)^2-f(u_n)(v_n-z_n)^2}. \end{aligned}\right\} \tag{48}$$

By replacing the approximating value of $f'(u_n)$ defined by (9) into (48), we have

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{R(v_n-u_n)} - \frac{f(u_n)^2(f'(v_n)-R(v_n-u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2}, \\ v_{n+1} &= w_n - \frac{f(w_n)((v_n-u_n)(v_n-w_n)(u_n-w_n))}{-f(w_n)(v_n-u_n)(v_n-2w_n-u_n)+f(u_n)(v_n-w_n)^2-f(u_n)(v_n-z_n)^2}. \end{aligned}\right\} \tag{49}$$

The error equation for (49) is:

$$3d_2^2 d_3^2 e_n^7 + O(e_n^8). \tag{50}$$

It can be observed from (49) that the iterative method (48) is of seventh-order. Since, reduction of function evaluation also decreased the convergence order ,so our aim is not yet achieved. Main motive is to decrease both the function evaluation and preserving the order. To achieve goal for finding the most suitable approximation to make it optimal. Instead of using Lagrange interpolation in the third step of (5) proposed by Rafiullah and Jabeen [40] to approximate $f'(w_n)$, we utilized the Gauss quadrature approach by considering the Newton's formula in order to derive Gauss quadrature approximation as:

$$f'(w_n) = g'(v_n) + \int_{v_n}^{w_n} f''(s)ds. \tag{51}$$

By means of weight function the second derivative in (51) can be approximated:

$$\int_{v_n}^{w_n} f''(s)ds = c_1 f(v_n) + c_2 f(u_n) + c_3 f(w_n) + c_4 f'(v_n). \tag{52}$$

For finding the parameters c_1, c_2, c_3 and c_4. We employ four functions $f(s) = 1$, $f(s) = s$, $f(s) = s^2$ and $f(s) = s^3$ in order to obtain a family of four equations as:

$$\left.\begin{aligned} &c_1 + c_2 + c_3 = 0, \\ &c_1 v_n + c_2 u_n + c_3 w_n + c_4 = 0 \\ &c_1 v_n^2 + c_2 u_n^2 + c_3 w_n^2 + 2c_4 v_n = 2(w_n - v_n), \\ &c_1 v_n^3 + c_2 u_n^3 + c_3 w_n^3 + 3c_4 v_n^2 = 3(w_n^2 - v_n^2). \end{aligned}\right\} \tag{53}$$

The solution of the system of equations (53) is specified by four constants c_1, c_2, c_3 and c_4 and consequently by substituting these values into (51), we obtain the value of $f'(w_n)$ as:

$$\begin{aligned} f'(w_n) = &-\frac{u_n - w_n)(3v_n - 2u_n - w_n)}{(v_n - w_n)(v_n - u_n)^2} f(v_n) + \frac{(v_n - w_n)^2}{(u_n - w_n)(v_n - u_n)^2} f(v_n) \\ &- \frac{(v_n + 2u_n - 3w_n)}{(v_n - w_n)(u_n - w_n)} f(w_n) + \frac{(2v_n + u_n - w_n)}{(v_n - u_n)} f'(v_n). \end{aligned} \tag{54}$$

On simplifying the expression (54) and substituting it in the last step of (5). We achieve the following:

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)-f(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{R(v_n-u_n)} - \frac{f(u_n)^2(f'(v_n)-R(v_n-u_n))}{2(f(v_n)-f(u_n))f'(v_n)^2}, \\ v_{n+1} &= w_n - g(w_n)\frac{(v_n-w_n)(v_n-u_n)^2(u_n-w_n)}{Ef(v_n)+Ff(u_n)+Gf(w_n)+Hf'(v_n)}, \end{aligned}\right\} \tag{55}$$

where

$$E = -(u_n-w_n)^2(3v_n-2u_n-w_n),$$
$$F = (v_n-w_n)^3,$$
$$G = -(v_n-v_n)^2(v_n+2u_n-3w),$$
$$H = (u_n-w_n)^2(v_n-w_n)(v_n-u_n).$$

The error expression for (51) is:

$$d_2^2 d_3(d_2d_3-d_4)e_n^8 + O(e_n^9). \tag{56}$$

It may be noted that our proposed iterative method (49) denoted by $SZ1$ satisfies the hypothesis of Kung and Traub and hence is an optimal eighth-order iterative method requiring only four function evaluations.

2.2 Convergence Analysis of Optimal Eighth-Order Method

Theorem 2.4. *Let $\beta \in D$be a simple root of a sufficiently differentiable function $f : D \subseteq R \to R$ within an open interval D. Then, the three-step without memory method* (49) *possesses optimal eighth order of convergence with four functional evaluations per iteration, and the asymptotic error is given by:*

$$d_2^2 d_3(d_2d_3-d_4)e_n^8 + O(e_n^9). \tag{57}$$

Proof. Let $e_n = v_n - \beta$ be the error term in the nth iteration. Assuming that $f(\beta) = 0$ and applying the Taylor's series expansion for function $f(v_n)$ around β, we obtain:

$$f(v_n) = f'(\beta)[e_n + d_2e_n^2 + d_3e_n^3 + d_4e_n^4 + d_5e_n^5 + d_6e_n^6 + d_7e_n^7 + d_8e_n^8 + O(e_n)^9], \tag{58}$$

where $d_k = \frac{f^k(\beta)}{kf'(\beta)}$ for $k \in N$. Similarly, applying Taylor's series expansion for the function $g'(v_n)$ around β, we obtain

$$f'(v_n) = f'(\beta)[1+2d_2e_n+3d_3e_n^2+4d_4e_n^3+5d_5e_n^4+6d_6e_n^5+7d_7e_n^6+8d_8e_n^7+9d_9e_n^7+O(e_n)^9]. \tag{59}$$

On dividing (58) by (59), we obtain

$$\frac{g(v_n)}{g'(v_n)} = e_n - d_2e_n^2 + 2(d_2^2-d_3)e_n^3 + (-4d_2^3+7d_2d_3-3d_4)e_n^4 + O(e_n)^5. \tag{60}$$

Substituting (60) in the first step of (55), we have

$$v_n = \beta + d_2e_n^2 + (-2d_2^2+d_2)e_n^3 + (4d_2^3-7d_2d_3+3d_4)e_n^4 + O(e_n)^5] \tag{61}$$

Applying Taylor's series expansion for the function $f(v_n)$ around β, we have

$$f(u_n) = f'(\beta)[d_2e_n^2 + (-2d_2^2 + d_2)e_n^3 + (5d_2^3 - 7d_2d_3 + 3d_4)e_n^4 + O(e_n)^5]. \tag{62}$$

Also, from the second step of (55) with the aid of (58) and (62), we have

$$f(v_n) - f(u_n) = e_n + (2d_2^2 - d_3)e_n^3 + (-5d_2^3 + 7d_2d_3 - 2d_4)e_n^4 + O(e_n)^5, \tag{63}$$

$$2\Big(\frac{f(v_n) - f(v_n)}{v_n - u_n}\Big) - f'(v_n) = 1 + (-3d_3 + 2(d_2^2 + d_3)e_n^2 + (-4d_4 + 2(-4d_2^3 + d_2(2d_2^2 - d_3) + 4d_2d_3 + d_4)e_n^3 + O(e_n)^4)) \tag{64}$$

and with the help of (62), (63) and (64), we have

$$\frac{f(u_n)(f'(v_n) - f'(u_n))}{2(f(v_n) - f(u_n))f'(v_n)^2} = d - 2^3e_n^4 + \Big(-4d - 2^4 + \frac{1}{2}(4d_2^2)(-2d_2^2 + 2d_3) + d_2^2(6d_3 - 2(d_2^2 + d_3))\Big)e_n^5 + O(e_n)^6. \tag{65}$$

Substitute (62), (63), (64) and (65) in the second step of (55), we have

$$w_n = \beta + (3d_2^3 - d_2d_3)e_n^4 - 2(9d_2^4 + 10d_2^2d_3 + d_2^2 + d_2d_4)e_n^5 + O(e_n)^6. \tag{66}$$

Again, we apply Taylor's series for the function $f(w_n)$ around β, we obtain

$$f(w_n) = f'(\beta)[(3d_2^3 - d_2d_3)e_n^4 - 2(9d_2^4 + 10d_2^2d_3 + d_2^2 + d_2d_4)e_n^5 + O(e_n)^6]. \tag{67}$$

With the aid of (58), (59), (62), (63), (66) and (67), we obtain the following

$$Ef(v_n) = (-(u-w)^2(3v - 2u - w))f(v_n) = -3d_2^2e_n^5 + O(e_n)^6, \tag{68}$$

$$Ff(u_n) = (v-w)^3 f(u_n) = e_n^3 + (-9d_2^4 + 3d_2d_3)e_n^6 + O(e_n)^7, \tag{69}$$

$$Gf(w_n) = (v-u)^2(v + 2u - 3w)f(w_n) = e_n^3 + (d_2^2 + 4d_3 + 2(-2d_2^2 + d_2^3))e_n^5 + O(e_n)^6, \tag{70}$$

$$Hf'(v_n) = (u-w)^2(v-w)(v-u)g'(v_n) = d_2^2e_n^6 + (-d_2^3 + 2d_2 + (-2d_2^2 + 2d_2))e_n^7 + O(e_n)^8 \tag{71}$$

and

$$(v-w)(v-u)^2(u-w) = d_2e_n^5 + (-4d_2^2 + 2d_3)e_n^6 + O(e_n)^7. \tag{72}$$

Finally, we employ (66)-(72) in the last step of the proposed method (55), we obtain

$$e_{n+1} = d_2^2d_3(d_2d_3 - d_4)e_n^8 + O(e_n^9). \tag{73}$$

Therefore, our proposed method (55) is eighth-order optimal iterative method with the evaluation of three functions and one of the first derivative. Note that the efficiency index of our proposed method is $8^{\frac{1}{4}} \approx 1.6818$ which is better than Newton's method and others available in the literature. □

3. Numerical Implementations

In order to demonstrate the effectiveness of our proposed method, we performed some numerical simulations with existing with and without memory methods. A few numerical schemes that can be practically compared to the suggested scheme are briefly described in this section. Some fifth order methods are:

XW1 Fifth-order with memory scheme given in [54].

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n) + L_nf(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{2L_nf(v_n) + f'(v_n)}\Big(1 + \frac{2f(u_n)}{f(v_n)} + a\Big(\frac{f(u_n)}{f(v_n)}\Big)^2\Big), \end{aligned}\right\} \tag{74}$$

where $a \in R$ denoted by XW1, and

XW2 Fifth-order with memory scheme given [55].

$$\left.\begin{aligned} u_n &= v_n - \frac{f(v_n)}{f'(v_n)+L_n f(v_n)}, \\ v_{n+1} &= u_n - \frac{f(u_n)}{2L_n f(v_n)+f'(v_n)} + \left(\frac{f(v_n)+(2+b)f(u_n)}{f(v_n)+bf(u_n)}\right), \end{aligned}\right\} \tag{75}$$

where $b \in R$ denoted by XW2.

NN1: Modified Halley's method of fifth-order given in [30].

NN2: Fifth-order Noor's method given in [31].

KU1: Chebshew Haley's method of fifth-order given in [23].

KU2: Fifth-order method proposed by Kou *et al*. [24].

FG1: Newton type method with fifth-order convergence given in [15].

HC1: Fifth-order method by Ham and Chun [18].

MR1: Rafiullah's fifth order method given in [39].

MR2: Rafiullah and Jubeen's method of fifth-order given in [40],

and some optimal eight orders are CP1 [11], JK1 [45], KS1 [26], LT1 [28], NS2 [9], SM1 [36], SS1 [35], SQ1 [40] and our proposed method SZ1.

In Tables 1-5 and 6-10, we have presented the absolute residual error $|f(v_n)|$ for each test function, approximated roots (v_n), error in the consecutive iterations $|v_n - v_{n-1}|$, CPU time in seconds and the *Computational Order of Convergence* (*COC*) for all the compared methods. All the numerical results have been tabulated. Tables 1-5 and 6-10 shows the numerical comparisons of various two points iterative methods of fifth-order convergence and eighth-order optimal iterative methods respectively. It is clearly visible that our results are superior and efficient than existing with and without memory methods. Figures 1 and 2 shows that our proposed methods are quite superior than other existing schemes currently in use. Besides computational order of convergence, CPU time is also one of the best way to compare the effectiveness of the iterative methods. At this juncture, with the help of MATHEMATICA 11 the command "`Timeused[ ]`" is used to calculate the CPU time. The numerical results were performed with the MATHEMATICA 11 system running under WINDOWS 10 PRO with an installed memory of 10GB having a processor INTEL(R) Core(TM) i5-3727U CPU @ 1.80 GHz 2.30GHz speed and system type 64-bit operating system. The COC has been calculated by using the usual formula [57]:

$$COC \sim \frac{\ln|f(v_{n+1})/f(v_n)|}{\ln|f(v_n)/f(v_{n-1})|}, \tag{76}$$

to verify the hypothesized convergence rate and assess computational effectiveness.

Table 1. Numerical comparisons of two-point methods with memory for f_1

Fun.	Method	Guess	$\lvert v_n - v_{n-1}\rvert$	$\log\lvert f(v_n)\rvert$	COC	CPU
f_1	WX1 (74)-(15), $T_0 = 0.5$	1.3	3.4305e-54	8.0355e-203	3.7392	1.421
	WX1 (74)-(16), $T_0 = 0.5$		1.8380e-56	4.0614e-224	4.0000	1.657
	WX1 (74)-(17), $T_0 = 0.5$		3.0817e-56	3.5465e-223	4.0000	1.657
	WX2 (75)-(15), $T_0 = 0.5$		3.4304e-54	8.6015e-203	3.7392	0.938
	WX2 (75)-(16), $T_0 = 0.5$		1.8380e-56	4.3475e-224	4.0000	1.062
	WX2 (75)-(17), $T_0 = 0.5$		3.0816e-56	3.7963e-223	4.0000	1.169
	AS1 (18)-(15), $T_0 = 0.5$		4.0003e-145	7.1437e-663	4.5612	1.047
	AS1 (18)-(16), $T_0 = 0.5$		1.1630e-174	7.5549e-836	4.7917	1.0
	AS1 (18)-(17),$T_0 = 0.5$		1.4154e-195	1.1278e-977	5.0000	1.031
	NN1		4.9217e-146	1.1741e-729	5.0000	1.39
	NN2		NC			
	KU1		1.0191e-38	6.3923e-117	3.0000	1.231
	KU2		1.3669e-164	1.9747e-823	5.0000	1.169
	FG1		1.3321e-117	3.2378e-586	5.0000	1.293
	HC1		3.1028e-105	5.2842e-524	5.0000	1.124
	MR1		9.6964e-117	1.8264e-506	5.0000	1.454
	MR2		1.6557e-98	3.2631e-490	5.0000	1.422

Table 2. Numerical comparisons of two-point methods with memory for f_2

Fun.	Method	Guess	$\lvert v_n - v_{n-1}\rvert$	$\log\lvert f(v_n)\rvert$	COC	CPU
f_2	WX1 (74)-(15), $T_0 = -0.5$	17.3	3.9401e-55	6.4333e-218	4.0000	0.812
	WX1 (74)-(16), $T_0 = -0.5$		3.8674e-55	5.9712e-218	4.0000	0.751
	WX1 (74)-(17), $T_0 = -0.5$		3.8656e-55	5.9602e-218	4.0000	0.936
	WX2 (75)-(15), $T_0 = -0.5$		1.4073e-38	4.11085e-151	4.0000	0.907
	WX2 (75)-(16), $T_0 = -0.5$		1.3776e-38	3.7753e-151	4.0000	0.907
	WX2 (75)-(17), $T_0 = -0.5$		1.3769e-38	3.7672e-151	4.0000	0.843
	AS1 (18)-(15), $T_0 = -0.5$		6.2854e-105	8.6344e-479	4.5629	0.851
	AS1 (18)-(16), $T_0 = -0.5$		7.6668e-117	8.3397e-561	4.7914	0.797
	AS1 (18)-(17), $T_0 = -0.5$		4.0246e-122	6.8758e-611	5.0000	0.757
	NN1		1.4085e-126	1.5697e-634	5.0000	2.25
	NN2		NC			
	KU1		13.8548e-45	6.3923e-136	3.0000	2.164
	KU2		2.0640e-160	7.4235e-802	5.0000	1.999
	FG1		2.7055e-181	3.9086e-507	5.0000	2.314
	HC1		1.3321e-117	3.2378e-586	5.0000	1.093
	MR1		9.6964e-117	1.8264e-506	5.0000	1.454
	MR2		1.6557e-98	3.2631e-490	5.0000	1.422

Table 3. Numerical comparisons of two-point methods with memory for f_3

Fun.	Method	Guess	$\|v_n - v_{n-1}\|$	$\log\|f(v_n)\|$	COC	CPU
f_3	WX1 (74)-(15), $T_0 = -0.6$	0.1	1.7158e-37	3.9640e-140	3.7302	1.546
	WX1 (74)-(16), $T_0 = -0.6$		9.7898e-40	9.3927e-159	4.0000	1.984
	WX1 (74)-(17), $T_0 = -0.6$		1.1982e-39	2.3348e-158	4.0000	2.455
	WX2 (75)-(15), $T_0 = -0.6$		1.7724e-37	4.4735e-140	3.7302	2.017
	WX2 (75)-(16), $T_0 = -0.6$		1.0306e-39	1.1537e-158	4.0000	1.889
	WX2 (75)-(17), $T_0 = -0.6$		1.2612e-39	2.8657e-158	4.0000	2.032
	AS1 (18)-(15), $T_0 = -0.6$		1.5409e-140	9.5677e-640	4.5610	1.157
	AS1 (18)-(16), $T_0 = -0.6$		5.0078e-157	3.1845e-751	4.7914	1.093
	AS1 (18)-(17), $T_0 = -0.6$		4.1341e-169	5.8438e-844	5.0000	1.023
	NN1		4.8790e-152	3.3750e-758	5.0000	1.25
	NN2		NC			
	KU1		3.1751e-36	5.2098e-108	3.0000	1.157
	KU2		1.4914e-153	9.6075e-766	5.0000	1.124
	FG1		1.8883e-149	3.5412e-745	5.0000	1.172
	HC1		3.9969e-134	4.6174e-668	5.0000	1.218
	MR1		4.8790e-152	3.3750e-668	5.0000	1.328
	MR2		2.5948e-128	7.5386e-639	5.0000	1.157

Table 4. Numerical comparisons of two-point methods with memory for f_4

Fun.	Method	Guess	$\|v_n - v_{n-1}\|$	$\log\|f(v_n)\|$	COC	CPU
f_4	WX1 (74)-(15), $T_0 = -0.8$	-0.3	3.6201e-24	2.0609e-93	4.0000	1.031
	WX1 (74)-(16), $T_0 = -0.8$		3.6201e-24	2.0609e-93	4.0000	1.079
	WX1 (74)-(17), $T_0 = -0.8$		3.6201e-24	2.0609e-93	4.0000	0.938
	WX2 (75)-(15), $T_0 = -0.8$		6.7591e-18	9.1836e-68	4.0000	1.234
	WX2 (75)-(16), $T_0 = -0.8$		6.7591e-18	9.1836e-68	4.0000	1.234
	WX2 (75)-(17), $T_0 = -0.8$		6.7591e-18	9.1836e-68	4.0000	1.031
	AS1 (18)-(15), $T_0 = -0.8$		1.6606e-169	4.6319e-784	4.5687	1.187
	AS1 (18)-(16), $T_0 = -0.8$		6.1245e-157	9.2552e-790	4.7914	1.047
	AS1 (18)-(17), $T_0 = -0.8$		6.1245e-157	9.2552e-791	5.0000	1.016
	NN1		3.7926e-113	1.2642e-620	5.0000	1.116
	NN2		NC			
	KU1		1.1176e-48	1.6106e-153	3.0000	1.281
	KU2		1.7200e-143	5.2108e-646	5.0000	1.062
	FG1		1.5166e-76	1.2036e-379	5.0000	1.171
	HC1		2.8385e-67	6.4491e-333	5.0000	1.134
	MR1		3.7926e-113	1.2642e-620	5.0000	1.391
	MR2		5.1038e-65	1.7316e-321	5.0000	1.254

Table 5. Numerical comparisons of two-point methods with memory for f_5

Fun.	Method	Guess	$\lvert v_n - v_{n-1}\rvert$	$\log\lvert f(v_n)\rvert$	COC	CPU
f_5	WX1 (74)-(15), $T_0 = 0.2$	−0.7	4.2787e-23	2.1357e-90	4.0000	1.275
	WX1 (74)-(16), $T_0 = 0.2$		3.7591e-23	1.2723e-90	4.0000	1.312
	WX1 (74)-(17), $T_0 = 0.2$		3.8562e-23	1.4091e-90	4.0000	1.032
	WX2 (75)-(15), $T_0 = 0.5$		4.4061e-22	3.6113e-86	4.0000	1.126
	WX2 (75)-(16), $T_0 = 0.5$		4.2939e-22	3.2571e-86	4.0000	1.133
	WX2 (75)-(17), $T_0 = 0.5$		4.3252e-22	3.3532e-86	4.0000	1.173
	AS1 (18)-(15), $T_0 = 0.5$		3.7746e-98	7.5571e-448	4.5619	0.952
	AS1 (18)-(16), $T_0 = 0.5$		0.968e-124	1.7825e-592	4.7921	0.75
	AS1 (18)-(17), $T_0 = 0.5$		1.3804e-138	1.5096e-694	5.0000	0.985
	NN1		7.6041e-121	8.9942e-604	5.0000	1.328
	NN2		NC			
	KU1		1.7832e-31	5.6152e-95	3.0000	1.163
	KU2		5.0251e-117	8.5918e-585	5.0000	1.361
	FG1		2.6308e-80	8.7918e-581	5.0000	1.422
	HC1		1.7524e-101	3.2098e-506	5.0000	1.5
	MR1		6.6725e-96	3.2927e-478	5.0000	1.563
	MR2		7.6185e-95	7.0787e-473	5.0000	1.469

Table 6. Numerical comparisons of three-point eighth-order optimal iterative method for f_1

Fun.	Method	Guess	$\lvert v_n - v_{n-1}\rvert$	$\log\lvert f(v_n)\rvert$	COC	CPU
f_1	CP1	1.6	4.3250e-81	5.6478e-645	8.0000	0.64
	JK1		7.2905e-74	1.9965e-586	8.0000	0.625
	KS1		2.4890e-77	1.1976e-614	8.0000	0.594
	LT1		1.9319e-76	9.9145e-607	8.0000	0.688
	NS2		2.3239e-80	2.4319e-639	8.0000	1.031
	SM1		6.0467e-79	2.9856e-628	8.0000	0.827
	SS1		3.6491e-86	1.0338e-685	8.0000	0.953
	SQ1		7.2303e-94	1.7802e-748	7.9930	0.983
	SZ1		8.0760e-82	5.3290e-651	8.0000	0.506

Table 7. Numerical comparisons of three-point eighth-order optimal iterative method for f_2

Fun.	Method	Guess	$\|v_n - v_{n-1}\|$	$\log\|f(v_n)\|$	COC	CPU
f_2	CP1	21.2	3.0658e-13	1.5013e-103	8.0000	1.015
	JK1		2.2938e-8	6.3586e-64	8.0000	0.905
	KS1		1.0655e-17	8.2236e-139	8.0000	0.954
	LT1		1.3183e-13	9.9295e-106	7.9844	0.89
	NS2		4.6355e-15	1.3566e-117	8.0000	0.797
	SM1		7.0279e-16	2.2022e-124	7.9921	1.001
	SS1		5.1565e-7	9.7507e-53	8.0000	1.016
	SQ1		1.1754e-16	5.6704e-131	7.9930	1.014
	SZ1		6.4244e-20	2.5960e-157	8.0000	0.738

Table 8. Numerical comparisons of three-point eighth-order optimal iterative method for f_3

Fun.	Method	Guess	$\|v_n - v_{n-1}\|$	$\log\|f(v_n)\|$	COC	CPU
f_3	CP1	$-.2$	2.3578e-56	7.5862e-446	8.0000	0.704
	JK1		7.6198e-51	4.0917e-401	8.0000	0.971
	KS1		1.7676e-55	6.3227e-439	8.0000	0.716
	LT1		5.0788e-57	1.9377e-451	8.0000	0.703
	NS2		2.4596e-57	1.8498e-452	8.0000	0.702
	SM1		7.6703e-55	4.7560e-446	8.0000	0.687
	SS1		1.5906e-47	2.4078e-374	8.0000	0.688
	SQ1		1.0833e-55	7.5457e-451	7.9930	0.953
	SZ1		4.2680e-58	4.8195e-460	8.0000	0.61

Table 9. Numerical comparisons of three-point eighth-order optimal iterative method for f_4

Fun.	Method	Guess	$\|v_n - v_{n-1}\|$	$\log\|f(v_n)\|$	COC	CPU
f_4	CP1	0.4	3.2642e-35	2.1851e-281	8.0000	0.829
	JK1		3.6433e-28	2.9671e-224	8.0000	1.046
	KS1		1.9083e-35	2.2069e-289	8.0000	0.844
	LT1		2.7060e-35	1.1527e-289	8.0000	0.922
	NS2		6.2714e-18	9.8945e-142	7.9867	1.062
	SM1		2.8624e-37	2.4078e-297	8.0000	0.954
	SS1		3.6548e-36	3.1543e-289	8.0000	0.953
	SQ1		1.4656e-32	7.5457e-251	7.9930	0.957
	SZ1		1.9083e-37	2.2069e-299	8.0000	0.823

Table 10. Numerical comparisons of three-point eighth-order optimal iterative method for f_5

Fun.	Method	Guess	$\lvert v_n - v_{n-1}\rvert$	$\log\lvert f(v_n)\rvert$	COC	CPU
f_5	CP1	0.4	1.1039e-47	6.2072e-379	8.0000	0.923
	JK1		1.1087e-39	2.6875e-314	8.0000	0.937
	KS1		7.6727e-53	4.8119e-453	8.0000	0.953
	LT1		6.6309e-42	6.1128e-333	8.0000	0.969
	NS2		8.9850e-38	7.3697e-300	8.0000	0.937
	SM1		3.5670e-45	2.9461e-359	8.0000	0.954
	SS1		7.8895e-44	1.5630e-347	8.0000	0.923
	SQ1		7.8895e-55	9.1702e-439	8.0000	0.916
	SZ1		7.6727e-57	4.8119e-493	8.0000	0.891

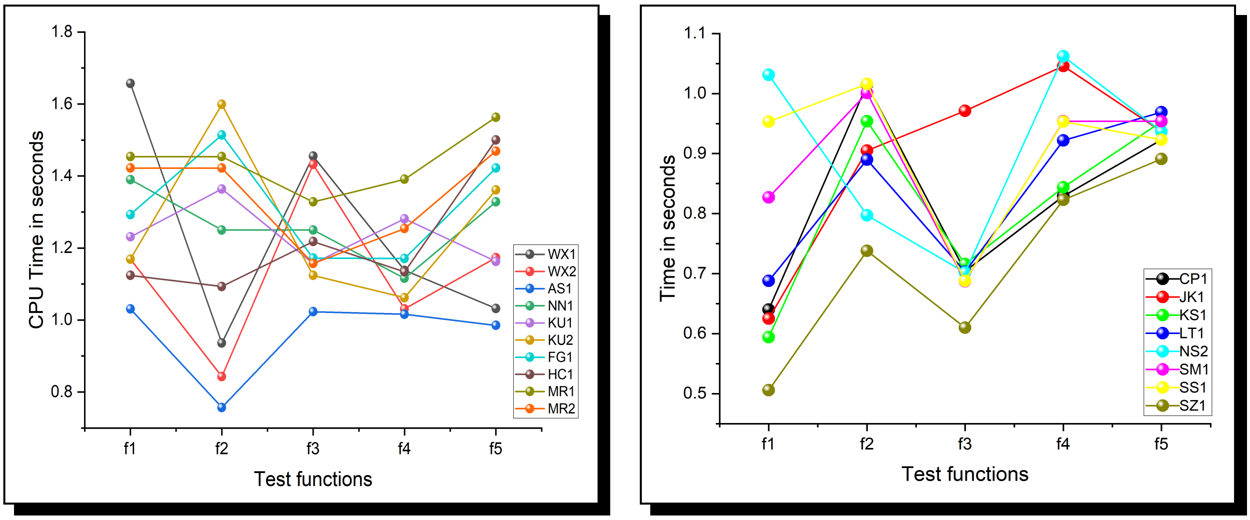

Figure 1. Graphs comparing the methods for each test function based on CPU time

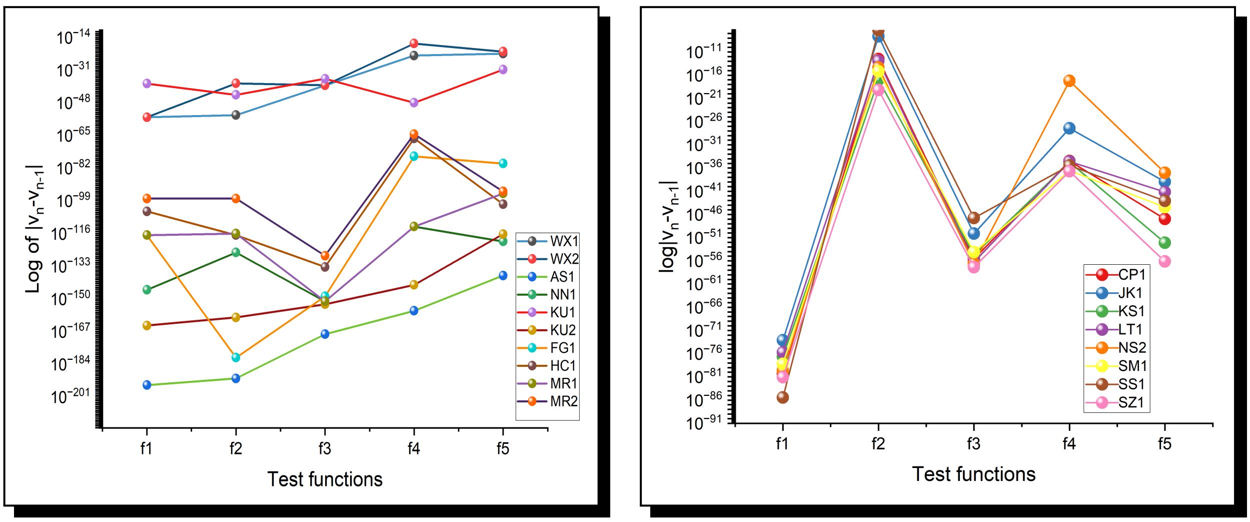

Figure 2. Graphs comparing the methods for each test function based on the error in consecutive iteration $\lvert v_n - v_{n-1}\rvert$

Example 3.1 (Kepler's Equation (Aerospace Engineering Problem) [48]). Let's look at the Kepler's equation in astronomy, which is written as follows.

$$P = M - e\sin(M), \quad M \in [0,2\pi],\ e \in [0,1]. \tag{77}$$

The eccentricity e and mean anomaly M are key components of the Kepler's equation. A point traveling in a Keplerian orbit can be located by using the eccentric anomaly E. In particular, when $e = 0.9$ and $M = 0.6$ the equation reduces to

$$f_1(v) = v - 0.9\sin(v) - 0.6, \tag{78}$$

where the unknown eccentric anomaly E is represented by the variable v. The above equation (78) converges to the 1.4975894133904085 using $v_0 = 1.8$ as the first guess. Table 1 presents the comparison findings.

Example 3.2. German physicist Max Planck developed the mathematical formula known as Planck's radiation law in 1900 to describe the spectral-energy distribution of radiation emitted by a black body given by:

$$f(\lambda) = \frac{8\pi c h_p \lambda^{-5}}{e^{\frac{ch_p}{\lambda B_k T}} - 1}, \tag{79}$$

where λ = Wavelength of the radiation,

h_p = Planck's constant,

B_k = Boltzmann constant,

T = Absolute temperature of the Blackbody raditaion,

C = Speed of light.

The wavelength that best fits the highest energy density is what we are looking for. For that purpose using (79), we obtain

$$\frac{8\pi C h_c}{\lambda^7 B_k T(\exp(Ch_c/\lambda B_k T - 1)^2}[5\lambda B_k T + (-5\lambda B_k T + Ch_c)\exp(Ch_c \lambda B_k T)] \tag{80}$$

or

$$\frac{8\pi C h_c \lambda^{-6}}{(\exp(Ch_c/\lambda B_k T - 1)}\left[\frac{Ch_c/\lambda B_k T(\exp(Ch_c/\lambda B_k T)}{(\exp(Ch_c/\lambda B_k T - 1)} - 5\right] = RS. \tag{81}$$

It is clear that when $S = 0$, there is a maximum for f, that is, when

$$\frac{Ch_c/\lambda B_k T(\exp(Ch_c/\lambda B_k T)}{(\exp(Ch_c/\lambda B_k T - 1)} = 5. \tag{82}$$

Let us take $v = \frac{CP_h}{\lambda B_k T}$, then equation (82) becomes

$$f_2(v) = e^{-v} + \frac{v}{5} - 1. \tag{83}$$

As mentioned in [38] the approximate root of the equation (83) is $x \approx 4.965114$. Consequently, using the following formula, it is possible to roughly predict the wavelength of radiation where the energy density is maximum:

$$\lambda \approx \frac{Ch_c}{4.965114 B_k T}. \tag{84}$$

Example 3.3 (Ocean Engineering Problem)**.** In ocean engineering. variable h in the equation below indicates the height of a reflected standing wave in a port

$$h = h_0\left[\sin\left(\frac{2\pi v}{\lambda}\right) + \cos\left(\frac{2\pi ts}{\lambda}\right) + e^{-v}\right], \tag{85}$$

where v is the distance from the wave's source and t is the amount of time after the wave was formed. The wave's velocity, its height at the source, and its wavelength are all represented by the symbols s, h_0, and λ, respectively. The aforementioned equation reduces to the following nonlinear equation for specific values of $\lambda = 16$, $t = 12$, $s = 48$, and $h = 0.4h_0$.

$$f_3(v) = e^{-v} + \sin\left(\frac{\pi v}{8}\right)\cos(72\pi) - 0.4\,. \tag{86}$$

The comparison results are presented in Table 2.

Example 3.4 (Fluid Permeability in Biogels [43])**.** The hydraulic permeability and the pressure gradient to fluid velocity in the extracellular fibre matrix can be defined using the nonlinear model shown below:

$$R_f(v)^3 - 20p(1-v^2) = 0\,, \tag{87}$$

where p denotes its particular hydraulic permeability, R_f denotes the radius of the fiber and $v \in [0,1]$ represents the porosity of the medium. Let us suppose that $p = 0.4655$ and $R_f = -100 \times 10^-9$, we achieve a polynomial of third degree as:

$$f_4(v) = -100 \times 10^{-9}v^3 + 9.3100v^2 - 18.6200v + 9.3100\,. \tag{88}$$

Example 3.5. The following equation describes the path of an electron traverses in the region between two parallel plates by considering the multi factor effect:

$$r(t) = r_0 + \left(v_0 + q_0\frac{E}{m\phi}\sin\phi t_0(t-t_0) + \psi\right)(t-t_0) + q_0\frac{E_0}{m\phi^2}(\cos(\phi t + \psi) + \sin(\phi t + \psi)), \tag{89}$$

where the position and velocity of an electron is given by r_0 and v_0 respectively at time t_0, q_0 and m are the charge and mass of an electron at rest and $E_0 \sin\phi(t) + \psi$ represents the RF electric field between the plates. If specific parameters are chosen, then equation (89) can be expressed as:

$$f_5(v) = \frac{\pi}{4} - \frac{\cos v}{2} + v = 0\,. \tag{90}$$

The root of equation (90) is $\zeta = -0.30909327154179 49$.

4. Conclusion

In this presented work, a novel family of two and three-point methods for evaluating simple roots of non-linear equations was introduced. As the primary objective of this research article is to increase the computational efficiency by reducing number of function evaluations while preserving the order. For that purpose, distinct approximations of self-accelerating parameters formed by Hermite Interpolating Polynomials were employed without performing any further calculations. The approximation of Gauss quadrature was utilized to eliminate an additional derivative for achieving optimal eighth-order convergence. Based on numerical data, the novel method appears to be an effective method for solving these issues and finding simple roots.

References

[1] S. Abdullah, N. Choubey and S. Dara, An efficient two-point iterative method with memory for solving non-linear equations and its dynamics, *Journal of Applied Mathematics and Computing* **70** (2024), 285 – 315, DOI: 10.1007/s12190-023-01953-w.

[2] G. Alefeld and J. Herzberger, *Introduction to Interval Computation*, 1st edition, Academic Press Inc., UK (1984).

[3] E. A. Avallone and T. Baumeister, *Marks' Standard Handbook for Mechanical Engineers*, 10th edition, McGraw-Hill, New York (1987).

[4] F. Awawdeh, On new iterative method for solving systems of nonlinear equations, *Numerical Algorithms* **54**(3) (2010), 395 – 409, DOI: 10.1007/s11075-009-9342-8.

[5] S. U. Chaudhary, S.-Y. Shin, J.-K. Won and K.-H. Cho, Multiscale modeling of tumorigenesis induced by mitochondrial incapacitation in cell death, *IEEE Transactions on Biomedical Engineering* **58**(10) (2011), 3028 – 3032, DOI: 10.1109/TBME.2011.2159713.

[6] J. Chen, Some new iterative methods with three-order convergence, *Applied Mathematics and Computation* **181**(2) (2006), 1519 – 1522, DOI: 10.1016/j.amc.2006.02.037.

[7] N. Choubey and J. Jaiswal, Two-and three-point with memory methods for solving nonlinear equations, *Numerical Analysis and Applications* **10**(1) (2017), 74 – 89, DOI: 10.1134/S1995423917010086.

[8] N. Choubey, J. Jaiswal and A. Choubey, Family of multipoint with memory iterative schemes for solving nonlinear equations, *International Journal of Applied and Computational Mathematics* **8** (2022), Article number 83, DOI: 10.1007/s40819-022-01283-8.

[9] N. Choubey and J. P. Jaiswal, An improved optimal eighth-order iterative scheme with its dynamical behaviour, *International Journal of Computing Science and Mathematics* **7**(4) (2016), 361 – 370, DOI: 10.1504/IJCSM.2016.078685.

[10] N. Choubey, B. Panday and J. P. Jaiswal, Several two-point with memory iterative methods for solving nonlinear equations, *Afrika Matematika* **29**(3) (2018), 435 – 449, DOI: 10.1007/s13370-018-0552-x.

[11] C. Chun and M. Y. Lee, A new optimal eighth-order family of iterative methods for the solution of nonlinear equations, *Applied Mathematics and Computation* **223** (2013), 506 – 519, DOI: 10.1016/j.amc.2013.08.033.

[12] A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa, A modified Newton-Jarratt's composition, *Numerical Algorithms* **55**(1) (2010), 87 – 99, DOI: 10.1007/s11075-009-9359-z.

[13] H.-F. Ding, Y.-X. Zhang, S.-F. Wang and X.-Y. Yang, A note on some quadrature based three-step iterative methods for non-linear equations, *Applied Mathematics and Computation* **215**(1) (2009), 53 – 57, DOI: 10.1016/j.amc.2009.04.036.

[14] J. Džunić, M. Petković and L. Petković, Three-point methods with and without memory for solving nonlinear equations, *Applied Mathematics and Computation* **218**(9) (2012), 4917 – 4927, DOI: 10.1016/j.amc.2011.10.057.

[15] L. Fang, L. Sun and G. He, An efficient Newton-type method with fifth-order convergence for solving nonlinear equations, *Computational & Applied Mathematics* **27**(3) (2008), 269 – 274, DOI: 10.1590/s1807-03022008000300003.

[16] C. Grosan and A. Abraham, A new approach for solving nonlinear equations systems, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **38**(3) (2008), 698 – 714, DOI: 10.1109/TSMCA.2008.918599.

[17] M. Hafiz, An efficient three-step tenth-order method without second order derivative, *Palestine Journal of Mathematics* **3**(2) (2014), 198 – 203, URL: https://pjm.ppu.edu/sites/default/files/papers/7_1.pdf.

[18] Y. Ham and C. Chun, A fifth-order iterative method for solving nonlinear equations, *Applied Mathematics and Computation* **194**(1) (2007), 287 – 290, DOI: 10.1016/j.amc.2007.04.005.

[19] S. Jaturonglumlert and T. Kiatsiriroat, Heat and mass transfer in combined convective and far-infrared drying of fruit leather, *Journal of Food Engineering* **100**(2) (2010), 254 – 260, DOI: 10.1016/j.jfoodeng.2010.04.007.

[20] M. Kansal, V. Kanwar and S. Bhatia, Efficient derivative-free variants of Hansen-Patrick's family with memory for solving nonlinear equations, *Numerical Algorithms* **73** (2016), 1017 – 1036, DOI: 10.1007/s11075-016-0127-6.

[21] E. D. Kaplan and C. J. Hegarty, *Understanding GPS/GNSS: Principles and Applications*, 3rd edition, Artech House, UK, xxi + 993 pages (2017).

[22] J. Kiusalaas, *Numerical Methods in Engineering with MATLAB®*, 2nd edition, Cambridge University Press, Cambridge, xii + 431 pages (2009), DOI: 10.1017/cbo9780511812200.

[23] J. Kou and Y. Li, The improvements of Chebyshev-Halley methods with fifth-order convergence, *Applied Mathematics and Computation* **188**(1) (2007), 143 – 147, DOI: 10.1016/j.amc.2006.09.097.

[24] J. Kou, Y. Li and X. Wang, A family of fifth-order iterations composed of Newton and third-order methods, *Applied Mathematics and Computation* **186**(2) (2007), 1258 – 1262, DOI: 10.1016/j.amc.2006.07.150.

[25] S. Kumar, V. Kanwar, S. K. Tomar and S. Singh, Geometrically constructed families of Newton's method for unconstrained optimization and nonlinear equations, *International Journal of Mathematics and Mathematical Sciences* **2011** (2011), Article ID 972537, 9 pages, DOI: 10.1155/2011/972537.

[26] H. Kung and J. F. Traub, Optimal order of one-point and multipoint iteration, *Journal of the ACM* **21**(4) (1974), 643 – 651, DOI: 10.1145/321850.321860.

[27] Y. Lin, L. Bao and X. Jia, Convergence analysis of a variant of the newton method for solving nonlinear equations, *Computers & Mathematics with Applications* **59**(6) (2010), 2121 – 2127, DOI: 10.1016/j.camwa.2009.12.017.

[28] L. Liu and X. Wang, Eighth-order methods with high efficiency index for solving nonlinear equations, *Applied Mathematics and Computation* **215**(9) (2010), 3449 – 3454, DOI: 10.1016/j.amc.2009.10.040.

[29] W. Nazeer, A. R. Nizami, M. Tanveer and I. Sarfaraz, A ninth-order iterative method for non-linear equations along with polynomiography, *Journal of Prime Research in Mathematics* **13**(1) (2017), 41 – 55, URL: https://jprm.sms.edu.pk/media/pdf/jprm/volume_13/05.pdf.

[30] M. A. Noor, W. A. Khan and A. Hussain, A new modified Halley method without second derivatives for nonlinear equation, *Applied Mathematics and Computation* **189**(2) (2007), 1268 – 1273, DOI: 10.1016/j.amc.2006.12.011.

[31] M. A. Noor and K. I. Noor, Fifth-order iterative methods for solving nonlinear equations, *Applied Mathematics and Computation* **188**(1) (2007), 406 – 410, DOI: 10.1016/j.amc.2006.10.007.

[32] R. Nourgaliev, P. Greene, B. Weston, R. Barney, A. Anderson, S. Khairallah and J.-P. Delplanque, High-order fully implicit solver for all-speed fluid dynamics, *Shock Waves* **29**(5) (2019), 651 – 689, DOI: 10.1007/s00193-018-0871-8.

[33] J. M. Ortega, *Numerical Analysis: A Second Course*, Classics in Applied Mathematics, SIAM, xv + 201 pages (1990), DOI: 10.1137/1.9781611971323.

[34] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Applied Mathematics, SIAM, xxvi + 572 pages (2000), DOI: 10.1137/1.9780898719468.

[35] S. Panday, A. Sharma and G. Thangkhenpau, Optimal fourth and eighth-order iterative methods for non-linear equations, *Journal of Applied Mathematics and Computing* **69**(1) (2023), 953 – 971, DOI: 10.1007/s12190-022-01775-2.

[36] S. Parimala, K. Madhu and J. Jayaraman, A new class of optimal eighth order method with two weight functions for solving nonlinear equation, *Journal Nonlinear Analysis and Application* **2018**(2) (2018), 83 – 94.

[37] M. S. Petković, B. Neta, L. D. Petković and J. Džunić, Multipoint methods for solving nonlinear equations: A survey, *Applied Mathematics and Computation* **226** (2014), 635 – 660, DOI: 10.1016/j.amc.2013.10.072.

[38] S. Qureshi, H. Ramos and A. K. Soomro, A new nonlinear ninth-order root-finding method with error analysis and basins of attraction, *Mathematics* **9**(16) (2021), 1996, DOI: 10.3390/math9161996.

[39] M. Rafiullah, *Multi-step Higher Order Iterative Methods for Solving Nonlinear Equations*, Master's Thesis, Higher Education Commission of Pakistan, Spring (2013).

[40] M. Rafiullah and D. Jabeen, New eighth and sixteenth order iterative methods to solve nonlinear equations, *International Journal of Applied and Computational Mathematics* **3** (2017), 2467 – 2476, DOI: 10.1007/s40819-016-0245-9.

[41] R. Rouzbar and S. Eyi, Reacting flow analysis of a cavity-based scramjet combustor using a Jacobian-free Newton-Krylov method, *The Aeronautical Journal* **122**(1258) (2018), 1884 – 1915, DOI: 10.1017/aer.2018.110.

[42] M. Salimi, T. Lotfi, S. Sharifi and S. Siegmund, Optimal Newton-Secant like methods without memory for solving nonlinear equations with its dynamics, *International Journal of Computer Mathematics* **94**(9) (2017), 1759 – 1777, DOI: 10.1080/00207160.2016.1227800.

[43] M. Shams, N. Rafiq, N. Kausar, N. A. Mir and A. Alalyani, Computer oriented numerical scheme for solving engineering problems., *Computer Systems Science and Engineering* **42**(2) (2022), 689 – 701, DOI: 10.32604/csse.2022.022269.

[44] H. Sharma, M. Kansal and R. Behl, An efficient two-step iterative family adaptive with memory for solving nonlinear equations and their applications, *Mathematical and Computational Applications* **27**(6) (2022), 97, DOI: 10.3390/mca27060097.

[45] J. R. Sharma and H. Arora, An efficient family of weighted-Newton methods with optimal eighth order convergence, *Applied Mathematics Letters* **29** (2014), 1 – 6, DOI: 10.1016/j.aml.2013.10.002.

[46] F. Soleymani, T. Lotfi, E. Tavakoli and F. K. Haghani, Several iterative methods with memory using self-accelerators, *Applied Mathematics and Computation* **254** (2015), 452 – 458, DOI: 10.1016/j.amc.2015.01.045.

[47] B. D. Stewart, *Attractor Basins of Various Root-Finding Methods*, Master's thesis, Naval Postgraduate School, Monterey, California (2001).

[48] A. Tassaddiq, S. Qureshi, A. Soomro, E. Hincal, D. Baleanu and A. A. Shaikh, A new three-step root-finding numerical method and its fractal global behavior, *Fractal and Fractional* **5**(4) (2021), 204, DOI: 10.3390/fractalfract5040204.

[49] V. Torkashvand, A two-step method adaptive with memory with eighth-order for solving nonlinear equations and its dynamic, *Computational Methods for Differential Equations* **10**(4) (2022), 1007 – 1026, DOI: 10.22034/cmde.2022.46651.1961.

[50] J. F. Traub, *Iterative Methods for the Solution of Equations*, Vol. **312**, *American Mathematical Society Chelsea Publishing*, (1964).

[51] I. Tsoulos and A. Stavrakoudis, On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods, *Nonlinear Analysis: Real World Applications* **11**(4) (2010), 2465 – 2471, DOI: 10.1016/j.nonrwa.2009.08.003.

[52] E. Vrscay and W. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and König rational iteration functions, *Numericshe Mathematik* **52** (1988), 1 – 16, DOI: 10.1007/BF01401018.

[53] X. Wang, An Ostrowski-type method with memory using a novel self-accelerating parameter, *Journal of Computational and Applied Mathematics* **330** (2018), 710 – 720, DOI: 10.1016/j.cam.2017.04.021.

[54] X. Wang and T. Zhang, A new family of Newton-type iterative methods with and without memory for solving nonlinear equations, *Calcolo* **51**(1) (2014), 1 – 15, DOI: 10.1007/s10092-012-0072-2.

[55] X. Wang and T. Zhang, Some Newton-type iterative methods with and without memory for solving nonlinear equations, *International Journal of Computational Methods* **11**(05) (2014), 1350078, DOI: 10.1142/S0219876213500783.

[56] X. Wang and T. Zhang, Efficient n-point iterative methods with memory for solving nonlinear equations, *Numerical Algorithms* **70**(2) (2015), 357 – 375, DOI: 10.1007/s11075-014-9951-8.

[57] S. Weerakoon and T. Fernando, A variant of Newton's method with accelerated third-order convergence, *Applied Mathematics Letters* **13**(8) (2000), 87 – 93, DOI: 10.1016/S0893-9659(00)00100-2.

[58] J. Zachary, *Introduction to Scientific Programming: Computational Problem Solving Using Maple and C*, Springer-Verlag, Berlin – Heidelberg, 380 pages (1996).

[59] H. Zhang, J. Guo, J. Lu, J. Niu, F. Li and Y. Xu, The comparison between nonlinear and linear preconditioning JFNK method for transient neutronics/thermal-hydraulics coupling problem, *Annals of Nuclear Energy* **132** (2019), 357 – 368, DOI: 10.1016/j.anucene.2019.04.053.