Applied Mathematics E-Notes, 21(2021), 320-343 © ISSN 1607-2510
Available free at mirror sites of http://www.math.nthu.edu.tw/∼amen/

Optimal Eighth And Sixteenth Order Iterative Methods For Solving Nonlinear Equation With Basins Of Attraction*

Parimala Sivakumar†, Kalyanasundaram Madhu‡, Jayakumar Jayaraman§

Received 20 May 2020

Abstract

This paper presents two optimal iterative methods for solving a nonlinear equation which are improved from well known fourth order Ostrowski's method. The first one is an eighth order method which uses three function evaluations and one first derivative evaluation. The second one is a sixteenth order method which uses four function evaluations and one derivative evaluation. Both methods satisfy the Kung-Traub optimality conjecture. The theoretical order of convergence of our schemes are derived. The performance and effectiveness of the new iterative methods have been tested and compared with few existing equivalent methods on some examples. In particular, we consider few wide variety of real life problems arising from different disciplines in order to check the applicability and effectiveness of the proposed methods. For the presented eighth order optimal method, a result which shows there exist a conjugacy mapping by using quadratic complex polynomial and a result on the extraneous fixed points are given. The basins of attraction in the complex plane for the eighth order methods are given to display the stability of the method with respect to the initial point.

1 Introduction

A common problem in engineering, scientific computing and applied mathematics, in general, is the problem of solving a nonlinear equation $f(x) = 0$. In most of the cases, whenever real problems are faced such as weather forecasting, accurate positioning of satellite systems in the desired orbit, measurement of earthquake magnitudes and other high-level engineering problems, only approximate solutions may get resolved. However, only in rare cases, it is possible to solve the governing equations exactly. The most familiar method of solving non linear equation is Newton's iteration method. The local order of convergence of Newton's method is two and it is an optimal method with two function evaluation per iterative step.

In the recent past, many higher order iterative methods are proposed and analyzed for solving nonlinear equations which are better than the classical methods such as Newton's method, Chebyshev method, Halley's iteration method, etc. Whenever the convergence order of a method increases, so does the number of function evaluations per step increases. Hence, Ostrowski [20] introduced a new index to determine the efficiency of a method called Efficiency Index (EI). Kung-Traub [14] conjectured that the order of convergence of any multi-point without memory method with d function evaluations cannot exceed the bound 2^{d-1}, the optimal order. Recently, there are many fourth and eighth order optimal iterative methods proposed in the literature (see [2, 7, 13, 15, 17, 21, 23, 24, 27] and references therein).

A detailed list of references as well as a survey on the progress made in the class of multi-point methods is found in the recent book by Petkovic et al. [21]. This book is a collection of theoretical results, algorithmic aspects and symbolic computation and serves as a text and a reference source for numerical analysts, engineers, physicists and computer scientists who are interested in the new developments and applications. In general, there are more number of eighth-order iterative methods for finding simple zeros of nonlinear equations in the available literature. But, unfortunately, there are few iterative methods of eighth-order for

*Mathematics Subject Classifications: 65H05, 65D05, 41A25.

†Department of Mathematics, Saradha Gangadharan College, Puducherry 605004, India
‡Department of Mathematics, Khalifa University, P.O.Box: 127788, Abu Dhabi, UAE
§Department of Mathematics, Pondicherry Engineering College, Puducherry 605014, India

multiple zeros with known or unknown multiplicity. Hence, Behl et al. [8] proposed an optimal scheme with eighth-order convergence based on weight function approach for multiple zeros.

Recently, there are many papers discussing the basins of attraction of various methods and ideas on how to choose the parameters appearing in the class of methods where weight functions are used. Amat et al. [4] studied the dynamics of a classical third-order Newton-type iterative method when it is applied to second and third degree polynomials. Affine conjugacy class of the method when it is applied to a differentiable function is given. Also, chaotic dynamics have been investigated in several examples. An eighth-order family improved from existing sixth-order method is given by Choubey et al. [11]. They also discussed the dynamics of the methods using basins of attraction for few complex polynomials.

Chun and Neta [12] collected many eighth-order schemes scattered in the literature and presented a quantitative comparison. They have compared all the methods in-terms of average number of function evaluations per iteration, CPU time and the number of divergent points. For more detailed study with many examples and dynamical behavior of the iterative methods, one can refer ([1, 6, 10]).

Motivated by optimization requirement, we develop iterative methods which agree the basic requirements of generating a quality numerical algorithm, that is, an algorithm which has high convergence speed, minimum computational cost and simple structure. Thus, two optimal methods having convergence order eight and sixteen for solving nonlinear equation are proposed in this work. Rest of the paper is organized as follows. In Section 2 the new methods are developed and their convergence analysis are discussed in section 3. Section 4 considers examples and numerical experimentation along with the comparison of the new methods with few existing methods of equivalent order. Four real life application problems are solved in Section 5, where all the listed methods and the proposed methods are numerically verified. In Section 6, we obtain the conjugacy mapping and all possible extraneous fixed points of the proposed eighth order method. In section 7, some visual graphical figures depicting the convergence for different initial points in a wide basins of attraction for the proposed eighth order method in comparison to some equivalent existing methods. Section 8 produces concluding remarks.

2 Development of the Methods

Definition 1 ([29]) *If the sequence $\{x_n\}$ tends to a limit x^* in such a way that*

$$\lim_{n\to\infty} \frac{x_{n+1} - x^*}{(x_n - x^*)^p} = C$$

for $p \geq 1$, then the order of convergence of the sequence is said to be p, and C is known as the asymptotic error constant. If $p = 1$, $p = 2$ or $p = 3$, the convergence is said to be linear, quadratic or cubic, respectively. Let $e_n = x_n - x^*$, then the relation

$$e_{n+1} = C\ e_n^p + O\Big(e_n^{p+1}\Big)$$

is called the error equation. The value of p is called the order of convergence of the method.

Definition 2 ([20]) *The Efficiency Index (EI) is given by*

$$EI = p^{\frac{1}{d}},$$

where d is the total number of new function evaluations (the values of f and its derivatives) per iteration.

Let $x_{n+1} = \psi(x_n)$ define an Iterative Function (I.F.). Let x_{n+1} be determined by new information at $x_n, \phi_1(x_n), ..., \phi_{\mathrm{i}}(x_n), \mathrm{i} \geq 1$ and no old information is reused. Thus, $x_{n+1} = \psi(x_n, \phi_1(x_n), ..., \phi_{\mathrm{i}}(x_n))$ is called a multi-point I.F. without memory.

The Newton-Raphson I.F. (*2NR*) is given by

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

The *2NR* is an one-point I.F. with two function evaluations and it is optimal as per Kung-Traub conjecture with $d = 2$. Further, $EI_{2NR} = 1.414$. The new methods are constructed based on Ostrowski's method and further developed by using divided difference approximations. Now, consider the well-known Ostrowski's method $(4OM)$[20],

$$y_n = x_n - u(x_n), \quad z_n = x_n - u(x_n)\Big[\frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)}\Big]. \tag{2}$$

The efficiency of the method (2) is $EI_{4OM} = 1.587$.

2.1 New Optimal Eighth order Method (8PM)

In order to increase the order of convergence, we add an additional Newton step in (2) then we obtain

$$y_n = x_n - u(x_n), \quad z_n = x_n - u(x_n)\Big[\frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)}\Big], \tag{3}$$
$$w_n = z_n - \frac{f(z_n)}{f'(z_n)}.$$

The above method is having eighth order convergence with five function evaluations. Consequently, this method is not optimal. In order to decrease the number of function evaluations, $f'(z_n)$ is approximated using divided differences. Hence we consider the following polynomial

$$q(t) = a_0 + a_1(t - x) + a_2(t - x)^2 + a_3(t - x)^3, \tag{4}$$

which satisfies

$$q(x_n) = f(x_n),\ q'(x_n) = f'(x_n),\ q(y_n) = f(y_n),\ q(z_n) = f(z_n). \tag{5}$$

Let us define the divided differences

$$f[y_n, x_n] = \frac{f(y_n) - f(x_n)}{y_n - x_n},\ f[y_n, x_n, x_n] = \frac{f[y_n, x_n] - f'(x_n)}{y_n - x_n}.$$

On implementing the above conditions (5) on (4), four linear equations with four unknowns a_0, a_1, a_2 and a_3 are obtained. From $q(x_n) = f(x_n)$, $q'(x_n) = f'(x_n)$, we get $a_0 = f(x_n)$ and $a_1 = f'(x_n)$. To find a_2 and a_3, the following equations are solved:

$$\begin{cases} f(y_n) = f(x_n) + f'(x_n)(y_n - x_n) + a_2(y_n - x_n)^2 + a_3(y_n - x_n)^3, \\ f(z_n) = f(x_n) + f'(x_n)(z_n - x_n) + a_2(z_n - x_n)^2 + a_3(z_n - x_n)^3. \end{cases}$$

Thus by applying divided differences, the above equations simplify into

$$\begin{cases} a_2 + a_3(y_n - x_n) = f[y_n, x_n, x_n], \\ a_2 + a_3(z_n - x_n) = f[z_n, x_n, x_n]. \end{cases} \tag{6}$$

Solving the above eqn. (6), we have

$$\begin{cases} a_2 = \frac{f[y_n,x_n,x_n](z_n - x_n) - f[z_n,x_n,x_n](y_n - x_n)}{z_n - y_n}, \\ a_3 = \frac{f[z_n,x_n,x_n] - f[y_n,x_n,x_n]}{z_n - y_n}. \end{cases} \tag{7}$$

Use eqn. (7) help to approximate $f'(z_n)$ in method (3) by $q'(z_n)$, where

$$f'(z_n) \approx q'(z_n) = a_1 + 2a_2(z_n - x_n) + 3a_3(z_n - x_n)^2.$$

Finally, we obtain a new optimal eighth order method $(8PM)$ given by

$$y_n = x_n - u(x_n), \quad z_n = x_n - u(x_n)\left[\frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)}\right],\ w_n = z_n - \frac{f(z_n)}{q'(z_n)}. \tag{8}$$

The efficiency of the method (8) is $EI_{8PM} = 1.682$, where it uses four function evaluations.

2.2 New Optimal Sixteenth order Method (16PM)

Extending the above three-point optimal eighth order scheme (8), a four-point optimal sixteenth order method is obtained in the following way. Consider a method with one more Newton step from (8) as given below:

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

The above method is having sixteenth order convergence with six function evaluations. However, this is not an optimal method. To get an optimal method, one need to reduce a function evaluation and preserve the same convergence order. Hence, approximate $f'(w_n)$ by the following polynomial

$$r(t) = b_0 + b_1(t-x) + b_2(t-x)^2 + b_3(t-x)^3 + b_4(t-x)^4, \tag{9}$$

where the parameters b_0, b_1, b_2, b_3 and b_4 are to be determined by imposing the conditions

$$r(x_n) = f(x_n),\ r'(x_n) = f'(x_n),\ r(y_n) = f(y_n),\ r(z_n) = f(z_n), r(w_n) = f(w_n).$$

On implementing the above conditions on (9), we obtain four linear equations with four unknowns b_0, b_1, b_2 and b_3. From the first two conditions, we get $b_0 = f(x_n)$ and $b_1 = f'(x_n)$. To find b_2, b_3 and b_4, we solve the following equation:

$$\begin{cases} f(y_n) = f(x_n) + f'(x_n)(y_n - x_n) + b_2(y_n - x_n)^2 + b_3(y_n - x_n)^3 + b_4(y_n - x_n)^4, \\ f(z_n) = f(x_n) + f'(x_n)(z_n - x_n) + b_2(z_n - x_n)^2 + b_3(z_n - x_n)^3 + b_4(z_n - x_n)^4, \\ f(w_n) = f(x_n) + f'(x_n)(w_n - x_n) + b_2(w_n - x_n)^2 + b_3(w_n - x_n)^3 + b_4(w_n - x_n)^4. \end{cases}$$

Thus by applying divided differences, the above equations simplify to

$$\begin{cases} b_2 + b_3(y_n - x_n) + b_4(y_n - x_n)^2 = f[y_n, x_n, x_n], \\ b_2 + b_3(z_n - x_n) + b_4(z_n - x_n)^2 = f[z_n, x_n, x_n], \\ b_2 + b_3(w_n - x_n) + b_4(w_n - x_n)^2 = f[w_n, x_n, x_n]. \end{cases}$$

Solving above equation, we have

$$\begin{cases} b_2 = \dfrac{\Big(f[y_n,x_n,x_n]\Big(-S_2^2S_3+S_2S_3^2\Big)+f[z_n,x_n,x_n]\Big(S_1^2S_3-S_1S_3^2\Big)+f[w_n,x_n,x_n]\Big(-S_1^2S_2+S_1S_2^2\Big)\Big)}{-S_1^2S_2+S_1S_2^2+S_1^2S_3-S_2^2S_3-S_1S_3^2+S_2S_3^2}, \\ b_3 = \dfrac{\Big(f[y_n,x_n,x_n]\Big(S_2^2-S_3^2\Big)+f[z_n,x_n,x_n]\Big(-S_1^2+S_3^2\Big)+f[w_n,x_n,x_n]\Big(S_1^2-S_2^2\Big)\Big)}{-S_1^2S_2+S_1S_2^2+S_1^2S_3-S_2^2S_3-S_1S_3^2+S_2S_3^2}, \\ b_4 = \dfrac{\Big(f[y_n,x_n,x_n]\Big(-S_2+S_3\Big)+f[z_n,x_n,x_n]\Big(S_1-S_3\Big)+f[w_n,x_n,x_n]\Big(-S_1+S_2\Big)\Big)}{-S_1^2S_2+S_1S_2^2+S_1^2S_3-S_2^2S_3-S_1S_3^2+S_2S_3^2}, \end{cases} \tag{10}$$

where $S_1 = y_n - x_n,\ S_2 = z_n - x_n,\ S_3 = w_n - x_n$. Further, using Equation (10), we have the approximation

$$f'(w_n) \approx r'(w_n) = b_1 + 2b_2(w_n - x_n) + 3b_3(w_n - x_n)^2 + 4b_4(w_n - x_n)^3.$$

Finally, we propose a new optimal sixteenth order method ($16PM$) given by

$$\begin{cases} y_n = x_n - u(x_n); \quad z_n = x_n - u(x_n)\left[\frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right], \\ w_n = z_n - \frac{f(z_n)}{q'(z_n)}; \quad x_{n+1} = w_n - \frac{f(w_n)}{r'(w_n)}. \end{cases} \tag{11}$$

The efficiency of the method (11) is $EI_{16PM} = 1.741$, where it uses five function evaluations.

3 Convergence Analysis

In this section, we prove the convergence of the proposed $I.F.s$ with the help of MATHEMATICA software.

Theorem 1 *Let $x^* \in D$ be a simple zero of sufficiently differentiable function $f : D \subset \mathbb{R} \to \mathbb{R}$, D is an open interval. If x_0 is sufficiently close to x^*, then the method (8) is of local eighth order convergence.*

Proof. Let $e_n = x_n - x^*$ and $c_j = \frac{f^{(j)}(x^*)}{j!f'(x^*)}$, $j = 2, 3, 4, ...$ Expanding $f(x_n)$ and $f'(x_n)$ about x^* by Taylor's method, we have

$$f(x_n) = f'(x^*)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + c_8e_n^8 + \ldots] \tag{12}$$

and

$$f'(x_n) = f'(x^*)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + 8c_8e_n^7 + \ldots]. \tag{13}$$

Now substituting (12) and (13) in (1), we get

$$\begin{aligned} y_n &= x^* + c_2e_n^2 - 2(c_2^2 - c_3)e_n^3 + (4c_2^3 - 7c_2c_3 + 3c_4)e_n^4 \\ &+(16c_2^5 - 52c_2^3c_3 + 33c_2c_3^2 + 28c_2^2c_4 - 17c_3c_4 - 13c_2c_5 + 5c_6)e_n^6 \\ &+(-8c_2^4 + 20c_2^2c_3 - 6c_3^2 - 10c_2c_4 + 4c_5)e_n^5 \\ &-2(16c_2^6 - 64c_2^4c_3 - 9c_3^3 + 36c_2^3c_4 + 6c_4^2 + 9c_2^2(7c_3^2 - 2c_5) + 11c_3c_5 + c_2(-46c_3c_4 + 8c_6) - 3c_7)e_n^7 \\ &+(64c_2^7 - 304c_2^5c_3 + 176c_2^4c_4 + 75c_3^2c_4 + c_2^3(408c_3^2 - 92c_5) - 31c_4c_5 - 27c_3c_6 \\ &+c_2^2(-348c_3c_4 + 44c_6) + c_2(-135c_3^3 + 64c_4^2 + 118c_3c_5 - 19c_7) + 7c_8)e_n^8 + \ldots. \end{aligned} \tag{14}$$

Expanding $f(y_n)$ about x^* and taking into account (14), we have

$$\begin{aligned} f(y_n) &= f'(x^*)[c_2e_n^2 - 2(c_2^2 - c_3)e_n^3 + (5c_2^3 - 7c_2c_3 + 3c_4)e_n^4 - 2(6c_2^4 - 12c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5)e_n^5 \\ &+(28c_2^5 - 73c_2^3c_3 + 34c_2^2c_4 - 17c_3c_4 + c_2(37c_3^2 - 13c_5) + 5c_6)e_n^6 - 2(32c_2^6 - 103c_2^4c_3 \\ &-9c_3^3 + 52c_2^3c_4 + 6c_4^2 + c_2^2(80c_3^2 - 22c_5) + 11c_3c_5 + c_2(-52c_3c_4 + 8c_6) - 3c_7)e_n^7 + \ldots]. \end{aligned} \tag{15}$$

Now, using (12), (13) and (14) in (2) then we have

$$\begin{aligned} z_n &= x^* + (c_2^3 - c_2c_3)e_n^4 - 2(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4)e_n^5 \\ &+(10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 - 7c_3c_4 + 3c_2(6c_3^2 - c_5))e_n^6 \\ &-2(10c_2^6 - 40c_2^4c_3 - 6c_3^3 + 20c_2^3c_4 + 3c_4^2 + 8c_2^2(5c_3^2 - c_5) + 5c_3c_5 + c_2(-26c_3c_4 + 2c_6))e_n^7 \\ &+(36c_2^7 - 178c_2^5c_3 + 101c_2^4c_4 + 50c_3^2c_4 + 3c_2^3(84c_3^2 - 17c_5) - 17c_4c_5 - 13c_3c_6 + c_2^2(-209c_3c_4 \\ &+20c_6) + c_2(-91c_3^3 + 37c_4^2 + 68c_3c_5 - 5c_7))e_n^8 + \ldots. \end{aligned} \tag{16}$$

Expanding $f(z_n)$ about x^* and making use of (16), we have

$$\begin{aligned} f(z_n) &= f'(x^*)\Big[(c_2^3 - c_2c_3)e_n^4 - 2(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4)e_n^5 + (10c_2^5 - 30c_2^3c_3 + 18c_2c_3^2 \\ &+12c_2^2c_4 - 7c_3c_4 - 3c_2c_5)e_n^6 - 2(10c_2^6 - 40c_2^4c_3 - 6c_3^3 + 20c_2^3c_4 + 3c_4^2 + 8c_2^2(5c_3^2 - c_5) \\ &+5c_3c_5 + c_2(-26c_3c_4 + 2c_6))e_n^7 + (37c_2^7 - 180c_2^5c_3 + 101c_2^4c_4 + 50c_3^2c_4 + c_2^3(253c_3^2 - 51c_5) \\ &-17c_4c_5 - 13c_3c_6 + c_2^2(-209c_3c_4 + 20c_6) + c_2(-91c_3^3 + 37c_4^2 + 68c_3c_5 - 5c_7))e_n^8 + \ldots\Big]. \end{aligned} \tag{17}$$

$$\begin{aligned} f[y_n, x_n] &= f'(x^*)\Big[1 + c_2e_n + (c_2^2 + c_3)e_n^2 + (-2c_2^3 + 3c_2c_3 + c_4)e_n^3 + (4c_2^4 - 8c_2^2c_3 + 2c_3^2 + 4c_2c_4 + c_5)e_n^4 \\ &+(-8c_2^5 + 20c_2^3c_3 - 9c_2c_3^2 - 11c_2^2c_4 + 5c_3c_4 + 5c_2c_5 + c_6)e_n^5 + (16c_2^6 - 48c_2^4c_3 - 2c_3^3 \\ &+29c_2^3c_4 + 3c_4^2 + c_2^2(31c_3^2 - 14c_5) + 6c_3c_5 + 6c_2(-4c_3c_4 + c_6 + c_7)e_n^6 \\ &+(-32c_2^7 + 112c_2^5c_3 - 74c_2^4c_4 - 7c_3^2c_4 + 7c_4c_5 + c_2^3(-94c_3^2 + 37c_5) + c_2^2(92c_3c_4 - 17c_6) \\ &+7c_3c_6 + c_2(11c_3^3 - 16c_4^2 - 30c_3c_5 + 7c_7) + c_8)e_n^7 + \ldots\Big]. \end{aligned} \tag{18}$$

$$\begin{aligned} f[y_n,x_n,x_n] &= f'(x^*)\Big[c_2+2c_3e_n+(c_2c_3+3c_4)e_n^2+2(-c_2^2c_3+c_3^2+c_2c_4+2c_5)e_n^3 \\ &+(4c_2^3c_3-7c_2c_3^2-3c_2^2c_4+7c_3c_4+3c_2c_5+5c_6)e_n^4+(-8c_2^4c_3-6c_3^3 \\ &+4c_2^3c_4+4c_2^2(5c_3^2-c_5)+10c_3c_5+4c_2(-5c_3c_4+c_6)+6(c_4^2+c_7)e_n^5+16c_2^5c_3 \\ &-4c_2^4c_4-25c_3^2c_4+17c_4c_5+c_2^3(-52c_3^2+5c_5)+c_2^2(46c_3c_4-5c_6) \\ &+13c_3c_6+c_2(33c_3^3-14c_4^2-26c_3c_5+5c_7)+7c_8)e_n^6+\ldots\Big]. \end{aligned} \tag{19}$$

Now

$$\begin{aligned} f[z_n,x_n] &= f'(x^*)\Big[1+c_2e_n+c_3e_n^2+c_4e_n^3+(c_2^4-c_2^2c_3+c_5)e_n^4+(-4c_2^5+9c_2^3c_3-3c_2c_3^2-2c_2^2c_4+c_6)e_n^5 \\ &+10c_2^6-34c_2^4c_3-2c_3^3+13c_2^3c_4-10c_2cc_4+c_2^2(26c_3^2-3c_5)+c_7)e_n^6 \\ &+(-20c_2^7+90c_2^5c_3-44c_2^4c_4-9c_3^2c_4+c_2^3(-110c_3^2+17c_5)+2c_2(15c_3^3-4c_4^2-7c_3 \\ &c_5)+c_2^2(72c_3c_4-4c_6)+c_8)e_n^7+\ldots\Big]. \end{aligned} \tag{20}$$

$$\begin{aligned} f[z_n,x_n,x_n] &= f'(x^*)\Big[c_2+2c_3e_n+3+(-4c_2^4c_3+8c_2^2c_3^2c_4e_n^2+4c_5e_n^3+(c_2^3c_3-c_2c_3^2+5c_6)e_n^4 \\ &-2c_3^3+2c_2^3c_4-4c_2c_3c_4+6c_7)e_n^5+(10c_2^5c_3-8c_2^4c_4+28c_2^2c_3c_4-11c_3^2c_4 \\ &+c_2^3(-30c_3^2+3c_5)+2c_2(9c_3^3-2c_4^2-3c_3c_5)+7c_8)e_n^6+\ldots\Big]. \end{aligned} \tag{21}$$

Now substituting (14), (16), (19) and (21) in (7) we obtain

$$\begin{aligned} a_2 &= f'(x^*)\Big[c_2+3c_3e_n+5c_4e_n^2+(c_2c_4+7c_5)e_n^3+(-2c_2^2c_4+2c_3c_4+2c_2c_5+9c_6)e_n^4 \\ &+(5c_2^3c_4-8c_2c_3c_4+3c_4^2-3c_2^2c_5+4c_3c_5+3c_2c_6+11c_7)e_n^5+\ldots\Big], \end{aligned} \tag{22}$$

and

$$\begin{aligned} a_3 &= f'(x^*)\Big[c_3+2c_4e_n+(c_2c_4+3c_5)e_n^2+(-2c_2^2c_4+2c_3c_4+2c_2c_5+4c_6)e_n^3 \\ &+(5c_2^3c_4-8c_2c_3c_4+3c_4^2-3c_2^2c_5+4c_3c_5+3c_2c_6+5c_7)e_n^4+\ldots\Big]. \end{aligned} \tag{23}$$

Consequently, we obtain the required error estimate $e_{n+1}=c_2^2(c_2^2-c_3)(c_2^3-c_2c_3+c_4))e_n^8+O(e_n^9)$. ∎

The following theorem can be proved similar to the above theorem with the help of Mathematica software and hence proof is not given.

Theorem 2 *Let $f: D\subset\mathbb{R}\to\mathbb{R}$ be a sufficiently smooth function having continuous derivatives. If $f(x)$ has a simple root x^* in the open interval D and x_0 chosen in sufficiently small neighborhood of x^*, then the method (11) is of local sixteenth order convergence and it satisfies the error equation*

$$e_{n+1}=c_2^4(c_2^2-c_3)^2(c_2^3-c_2c_3+c_4)(c_2^4-c_2^2c_3+c_2c_4-c_5)e_n^{16}+O(e_n^{17}).$$

4 Numerical Examples

The present section deals with the computation of nonlinear numerical examples which are furnished to corroborate the effectiveness of the proposed iterative methods. We compare them with $2NR$ and few existing eighth-order methods specifically, $8KTM$, $8LWM$, $8PNPD$, $8CFGT$, $8SAM$ and $8PMJ$. Numerical computations have been carried out in the Matlab software with 500 significant digits. We have used the stopping criteria for the iterative process satisfying $error=|x_N-x_{N-1}|<\epsilon$, where $\epsilon=10^{-50}$ and N is the number of iterations required for convergence.

The computational order of convergence is given by

$$\rho = \frac{\ln|(x_N - x_{N-1})/(x_{N-1} - x_{N-2})|}{\ln|(x_{N-1} - x_{N-2})/(x_{N-2} - x_{N-3})|}.$$

Tables 1-6 holds the values of initial approximation (x_0), number of iteratins (N), the absolute errors $|x_N - x_{N-1}|$ in the first three iterations and last iteration, computational order of convergence (ρ) and cpu time ($cpu(s)$). Here D implies that the method is divergent. The following eighth order existing methods are taken for the purpose of comparison:
Method proposed by Kung-Traub [14] ($8KTM$):

$$\begin{cases} y_n = x_n - u(x_n), \\ z_n = y_n - \frac{f(y_n)f(x_n)}{(f(x_n)-f(y_n))^2} u(x_n), \\ x_{n+1} = z_n - u(x_n) \frac{f(x_n)f(y_n)f(z_n)}{(f(x_n)-f(y_n))^2} \frac{f(x_n)^2+f(y_n)(f(y_n)-f(z_n))}{(f(x_n)-f(z_n))^2(f(y_n)-f(z_n))}. \end{cases} \tag{24}$$

Method given by Liu et al [15] ($8LWM$):

$$\begin{cases} y_n = x_n - u(x_n), \\ z_n = y_n - \frac{f(x_n)}{f(x_n)-2f(y_n)} \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)} \left(\left(\frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)} \right)^2 + \frac{f(z_n)}{f(y_n)-f(z_n)} + \frac{4f(z_n)}{f(x_n)+f(z_n)} \right). \end{cases} \tag{25}$$

Method suggested by Petkovic et al [21] ($8PNPD$):

$$\begin{cases} y_n = x_n - u(x_n), \\ z_n = x_n - \left(\left(\frac{f(y_n)}{f(x_n)} \right)^2 - \frac{f(x_n)}{f(y_n)-f(x_n)} \right) u(x_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)} \left(\varphi(t) + \frac{f(z_n)}{f(y_n)-f(z_n)} + \frac{4f(z_n)}{f(x_n)} \right), \end{cases} \tag{26}$$

where $\varphi(t) = 1 + 2t + 2t^2 - t^3$ and $t = \frac{f(y_n)}{f(x_n)}$. Method proposed by Sharma et al [23] ($8SAM$):

$$\begin{cases} y_n = x_n - u(x_n), \\ z_n = y_n - \left(3 - 2\frac{f[y_n,x_n]}{f'(x_n)} \right) \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)} \left(\frac{f'(x_n)-f[y_n,x_n]+f[z_n,y_n]}{2f[z_n,y_n]-f[z_n,x_n]} \right). \end{cases} \tag{27}$$

Method given by Cordero et al [13] ($8CFGT$):

$$\begin{cases} y_n = x_n - u(x_n), \\ z_n = y_n - \frac{f(y_n)}{f'(x_n)} \frac{1}{1-2t+t^2-t^3/2}, \\ x_{n+1} = z_n - \frac{1+3r}{1+r} \frac{f(z_n)}{f[z_n,y_n]+f[z_n,x_n,x_n](z-y)}, \\ r = \frac{f(z_n)}{f(x_n)}. \end{cases} \tag{28}$$

Method proposed by Parimala et al [25] ($8PMJ$):

$$\begin{cases} y_n = x_n - u(x_n), \\ z_n = x_n - \frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}u(x_n), \\ x_{n+1} = z_n - \frac{f(z_n)(z_n-y_n)}{f(z_n)-f(y_n)}(1+2\eta)\times(1+\tau^2+2\tau^3+(7/24)\tau^4), \end{cases} \tag{29}$$

where $\eta = \frac{f(z_n)}{f(x_n)}$, and $\tau = \frac{f(y_n)}{f(x_n)}$. The following numerical examples and their simple zeros for our study are given below:

$$\begin{cases} f_1(x) = \sin(2\cos x) - 1 - x^2 + e^{\sin(x^3)}, \quad x^* = -0.7848959876612125352..., \\ f_2(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5, \quad x^* = -1.2076478271309189270, \\ f_3(x) = \sqrt{x} - cosx, \quad x^* = 0.6417143708728826583..., \\ f_4(x) = x^3 + 4x^2 - 10, \quad x^* = 1.3652300134140968457..., \\ f_5(x) = \sqrt{x^2+2x+5} - 2\sin x - x^2 + 3, \quad x^* = 2.3319676558839640103..., \\ f_6(x) = ln(x^2+x+2) - x + 1, \quad x^* = 4.1525907367571583.... \end{cases}$$

Table 1: Numerical results for example $f_1(x)$.

Methods	x_0	N	$\lvert x_1 - x_0\rvert$	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_N - x_{N-1}\rvert$	ρ	$cpu(s)$
$2NR$	−1.2	7	0.3964	0.0185	2.4455e-04	1.5646e-60	2.00	1.3283
	−0.2	11	2.3823	3.3711	1.1619	6.8664e-60	1.99	2.0056
$8KTM$	−1.2	4	0.4151	5.8720e-06	3.7206e-42	0	7.99	0.6578
	−0.2	D	D	D	D	D	D	D
$8LWM$	−1.2	4	0.4151	1.4943e-05	1.1511e-38	1.4274e-303	7.99	0.6561
	−0.2	D	D	D	D	D	D	D
$8PNPD$	−1.2	4	0.4151	1.4188e-05	2.7614e-38	5.6862e-300	7.99	0.6592
	−0.2	32	1.8007e+09	9.0033e+08	4.5016e+08	2.8174e-65	7.93	4.6413
$8CFGT$	−1.2	4	0.4151	4.3579e-06	3.1296e-44	0	7.99	0.6627
	−0.2	26	4.1948	4.8161	13.2874	9.1518e-81	7.89	4.1650
$8SAM$	−1.2	4	0.4152	5.5316e-05	4.5477e-34	9.4823e-267	8.00	0.6622
	−0.2	22	1.1173e+05	5.5889e+04	2.7922e+04	1.0375e-322	7.99	3.1821
$8PMJ$	−1.2	4	0.4151	5.5187e-06	1.6794e-42	0	7.99	0.6752
	−0.2	D	D	D	D	D	D	D
$8PM$	−1.2	3	0.4151	2.7459e-07	4.6177e-54	4.6177e-54	7.57	0.5836
	−0.2	4	0.7178	0.1329	6.5073e-09	4.5930e-67	7.96	0.6918
$16PM$	−1.2	3	0.4151	1.3359e-12	3.0748e-192	3.0748e-192	15.63	0.9355
	−0.2	4	0.5610	0.0239	4.3829e-28	0	15.60	0.9755

The results from tables 1–6 show that for all the numerical examples $f_1(x) - f_6(x)$, the computational order of convergence agrees with the theoretical order of convergence. For the example $f_1(x)$, the methods $8KTM$, $8LWM$ and $8PMJ$ produce divergent results and for the example $f_2(x)$, the methods $8PNPD$, $8SAM$ and $8PMJ$ produce divergent results. For $f_3(x)$, the method $8PNPD$ produces divergent results, whereas the proposed methods $8PM$ and $16PM$ converge for all the examples. For the examples $f_1(x)$ and $f_4(x)$ at some initial points, $8PNPD$, $8CFGT$ and $8SAM$ methods take more number of iterations, whereas the $8PM$ and $16PM$ methods converge with less number of iterations. Also the presented methods converge with least error and consume less cpu time for most of the numerical examples.

Table 2: Numerical results for example $f_2(x)$.

Methods	x_0	N	$\|x_1 - x_0\|$	$\|x_2 - x_1\|$	$\|x_3 - x_2\|$	$\|x_N - x_{N-1}\|$	ρ	$cpu(s)$
$2NR$	−1.9	10	0.2192	0.2146	0.1697	2.1763e-57	2.00	1.6922
	−0.5	12	1.6075	0.2096	0.2193	9.2155e-58	1.99	1.9929
$8KTM$	−1.9	4	0.5855	0.1069	3.4154e-07	4.3663e-51	7.99	0.6421
	−0.5	6	1.4819	0.6053	0.1690	4.1821e-304	7.99	0.9080
$8LWM$	−1.9	4	0.6150	0.0774	3.5160e-09	5.1262e-68	8.01	0.6266
	−0.5	7	2.3500	0.5285	0.6339	3.4278e-130	8.00	1.0610
$8PNPD$	−1.9	5	0.5062	0.1859	2.6631e-04	2.0847e-203	7.99	0.8026
	−0.5	D	D	D	D	D	D	D
$8CFGT$	−1.9	5	1.0094	0.3187	0.0016	6.9651e-164	7.99	0.8461
	−0.5	5	0.9637	0.2546	0.0014	5.3513e-167	7.99	0.8128
$8SAM$	−1.9	4	0.7257	0.0334	6.2681e-10	7.1611e-72	8.01	0.6417
	−0.5	D	D	D	D	D	D	D
$8PMJ$	−1.9	4	0.6196	0.0728	5.6432e-09	8.2266e-66	7.99	0.6475
	−0.5	D	D	D	D	D	D	D
$8PM$	−1.9	4	0.1077	0.5846	4.3322e-08	1.8756e-59	8.03	0.6697
	−0.5	4	0.7799	0.0723	1.5303e-09	4.5457e-71	8.01	0.5498
$16PM$	−1.9	4	0.6851	0.0072	9.7199e-35	0	15.80	0.7967
	−0.5	4	0.7119	0.0042	1.7928e-38	0	15.79	0.6697

Table 3: Numerical results for example $f_3(x)$.

Methods	x_0	N	$\|x_1 - x_0\|$	$\|x_2 - x_1\|$	$\|x_3 - x_2\|$	$\|x_N - x_{N-1}\|$	ρ	$cpu(s)$
$2NR$	1.3	7	0.6224	0.0357	1.6994e-04	2.1346e-74	1.99	1.3826
	−0.1	9	0.6590	1.4249	0.6872	3.8733e-57	2.00	1.8372
$8KTM$	1.3	3	0.6583	6.1835e-07	3.0591e-54	3.0591e-54	7.85	0.6925
	−0.1	5	1.8403	1.3895	0.0025	2.4305e-202	8.00	0.8741
$8LWM$	1.3	3	0.6583	2.3033e-08	2.0526e-65	2.0526e-65	7.65	0.5417
	−0.1	5	0.4419	0.9444	0.4063	1.1123e-60	7.95	0.8953
$8PNPD$	1.3	3	0.6583	2.8069e-08	1.0135e-65	1.0135e-65	7.79	0.5985
	−0.1	D	D	D	D	D	D	D
$8CFGT$	1.3	3	0.6583	7.1472e-07	1.6838e-53	1.6838e-53	7.82	0.5958
	−0.1	5	1.3857	0.7111	9704e-06	0	7.99	0.9619
$8SAM$	1.3	4	0.6583	2.1843e-06	1.7971e-50	0	7.99	0.7488
	−0.1	5	1.6122	0.9985	4.1690e-05	3.5227e-321	8.00	0.8815
$8PMJ$	1.3	3	0.6583	5.9954e-07	3.1577e-54	3.1577e-54	7.83	0.5243
	−0.1	5	2.5915	2.0283	0.0120	2.7614e-157	8.00	0.8781
$8PM$	1.3	3	0.6583	8.8536e-07	6.5900e-53	6.5900e-53	7.86	0.5411
	−0.1	4	0.5218	0.2199	1.2629e-09	1.1297e-75	8.01	0.6235
$16PM$	1.3	3	0.6583	8.1917e-13	1.3003e-201	1.3003e-201	15.85	0.7739
	−0.1	4	1.8060	1.0878	8.2245e-10	1.3862e-153	15.76	1.1058

Table 4: Numerical results for example $f_4(x)$.

Methods	x_0	N	$\lvert x_1 - x_0\rvert$	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_N - x_{N-1}\rvert$	ρ	$cpu(s)$
$2NR$	1.7	7	0.2907	0.0432	9.2356e-04	6.3751e-54	1.99	1.0770
	0.3	10	3.6004	1.4340	0.7724	2.2871e-54	2.00	1.5962
$8KTM$	1.7	4	0.3348	2.0182e-06	1.2248e-47	0	7.99	0.6329
	0.3	5	3.0597	1.9599	0.0346	6.8622e-107	7.99	0.7291
$8LWM$	1.7	4	0.3348	2.0733e-06	1.5165e-47	0	7.99	0.6579
	0.3	11	59.2061	38.3806	12.6405	3.5289e-132	8.00	1.5608
$8PNPD$	1.7	4	0.3348	1.3930e-05	7.4081e-40	4.7394e-314	8.00	0.6399
	0.3	40	1.7890e+15	1.1151e+15	4.2002e+14	3.1065e-226	8.00	5.2991
$8CFGT$	1.7	3	0.3348	1.1454e-07	1.7803e-58	1.7803e-58	7.86	0.5091
	0.3	42	0.9411	1.8567	0.3822	1.1033e-311	8.00	5.8491
$8SAM$	1.7	4	0.3348	3.3964e-06	2.7532e-45	0	7.99	0.5982
	0.3	104	2.8115e+08	2.1317e+08	5.1542e+07	6.6248e-156	8.00	13.4957
$8PMJ$	1.7	4	0.3348	1.2449e-06	1.7603e-49	0	7.99	0.6294
	0.3	14	3.1474e+03	2.1591e+03	677.2203	3.2202e-106	7.99	1.9745
$8PM$	1.7	3	0.3348	2.4786e-07	5.4251e-56	5.4251e-56	7.94	0.5054
	0.3	4	1.2671	0.2018	6.0904e-09	7.2105e-69	7.97	0.6149
$16PM$	1.7	3	0.3348	3.0118e-14	0	0	15.80	0.7149
	0.3	3	1.0826	0.0174	4.5377e-34	0	15.82	0.6345

Table 5: Numerical results for example $f_5(x)$.

Methods	x_0	N	$\lvert x_1 - x_0\rvert$	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_N - x_{N-1}\rvert$	ρ	$cpu(s)$
$2NR$	3.0	7	0.5791	0.0880	9.1138e-04	2.1862e-64	1.99	1.2521
	1.8	6	0.5374	0.0055	3.0129e-06	6.6344e-52	1.99	1.0841
$8KTM$	3.0	3	0.6680	2.1477e-06	6.8605e-51	6.8605e-51	8.10	0.5737
	1.8	3	0.5320	1.8670e-09	2.2369e-75	2.2369e-75	7.80	0.5620
$8LWM$	3.0	4	0.6680	1.6093e-05	3.5614e-43	0	7.99	0.6558
	1.8	3	0.5320	1.8218e-09	9.6043e-75	9.6043e-75	7.71	0.5619
$8PNPD$	3.0	3	0.6680	1.5698e-06	1.0750e-51	1.0750e-51	8.02	0.5489
	1.8	3	0.5320	1.0413e-09	4.0285e-77	4.0285e-77	7.74	0.5294
$8CFGT$	3.0	4	0.6680	7.1235e-06	1.1381e-47	0	7.99	0.6982
	1.8	3	0.5320	2.5950e-09	3.5296e-75	3.5296e-75	7.92	0.5469
$8SAM$	3.0	4	0.6680	4.8563e-06	9.9155e-48	0	7.99	0.6633
	1.8	3	0.5320	3.5546e-08	8.1695e-65	8.1695e-65	7.89	0.5316
$8PMJ$	3.0	3	0.6680	1.2910e-06	5.2791e-53	5.2791e-53	8.12	0.5382
	1.8	3	0.5320	2.1305e-09	2.9042e-75	2.9042e-75	7.84	0.5925
$8PM$	3.0	3	0.6680	1.1557e-06	4.2999e-53	4.2999e-53	8.06	0.5442
	1.8	3	0.5320	3.6733e-09	4.4801e-73	4.4801e-73	7.83	0.5332
$16PM$	3.0	3	0.6680	8.8134e-13	0	0	15.83	0.6322
	1.8	3	0.5320	3.5854e-18	0	0	15.78	0.6310

Table 6: Numerical results for example $f_6(x)$.

Methods	x_0	N	$\|x_1 - x_0\|$	$\|x_2 - x_1\|$	$\|x_3 - x_2\|$	$\|x_N - x_{N-1}\|$	ρ	$cpu(s)$
$2NR$	4.6	7	0.4370	0.0104	6.5694e-06	6.1723e-102	1.99	1.1308
	2.8	7	1.5421	0.1874	0.0020	4.3796e-62	2.00	1.1680
$8KTM$	4.6	3	0.4474	1.5546e-10	6.4916e-86	6.4916e-86	7.97	0.5488
	2.8	4	1.3526	2.3216e-05	1.6058e-44	0	7.99	0.6228
$8LWM$	4.6	3	0.4474	2.5563e-10	5.7697e-84	5.7697e-84	7.97	0.5276
	2.8	4	1.3526	4.0279e-05	2.1917e-42	0	7.99	0.6406
$8PNPD$	4.6	3	0.4474	5.3558e-10	5.0666e-81	5.0666e-81	7.96	0.5216
	2.8	4	1.3527	1.4858e-04	1.7767e-37	7.4306e-301	7.99	0.6339
$8CFGT$	4.6	3	0.4474	2.4696e-11	2.7027e-93	2.7027e-93	7.99	0.5282
	2.8	3	1.3526	7.3177e-07	1.6060e-57	1.6060e-57	8.08	0.5220
$8SAM$	4.6	3	0.4474	9.0304e-11	6.1529e-88	6.1529e-88	7.96	0.5255
	2.8	4	1.3525	4.8168e-05	4.0320e-42	0	7.99	0.6286
$8PMJ$	4.6	3	0.4474	1.1670e-10	4.8809e-87	4.8809e-87	7.97	0.5211
	2.8	4	1.3526	1.9982e-05	3.6071e-45	0	7.99	0.6446
$8PM$	4.6	3	0.4474	4.5174e-11	8.3690e-91	8.3690e-91	7.98	0.5274
	2.8	3	1.3526	3.2509e-06	6.0202e-52	6.0202e-52	8.14	0.5287
$16PM$	4.6	3	0.4474	2.0795e-22	0	0	15.69	0.6384
	2.8	3	1.3526	7.0451e-13	0	0	15.78	0.6286

5 Some Real Life Applications

Generally, many problems in scientific and engineering which involve determination of any unknown appearing implicitly give rise to a root-finding problem. We start with one such simple application here.

Application 1: We consider the classical projectile problem [26] in which a projectile is launched from a tower of height $h > 0$, with initial speed v and at an angle ϕ with respect to the horizontal distance onto a hill, which is defined by the function ω, called the impact function which is dependent on the horizontal distance, x. We wish to find the optimal launch angle ϕ_m which maximizes the horizontal distance. In our calculations, we neglect air resistance.

The path function $y = P(x)$ that describes the motion of the projectile is given by

$$P(x) = h + x\tan\phi - \frac{gx^2}{2v^2}\sec^2\phi. \tag{30}$$

When the projectile hits the hill, there is a value of x for which $P(x) = \omega(x)$ for each value of x. We wish to find the value of ϕ that maximizes x.

$$\omega(x) = P(x) = h + x\tan\phi - \frac{gx^2}{2v^2}\sec^2\phi. \tag{31}$$

Differentiating Equation (31) implicitly w.r.t. ϕ, we have

$$\omega'(x)\frac{dx}{d\phi} = x\sec^2\phi + \frac{dx}{d\phi}\tan\phi - \frac{g}{v^2}\left(x^2\sec^2\phi\tan\phi + x\frac{dx}{d\phi}\sec^2\phi\right). \tag{32}$$

Setting $\frac{dx}{d\phi} = 0$ in Equation (32), we have

$$x_m = \frac{v^2}{g}\cot\phi_m \tag{33}$$

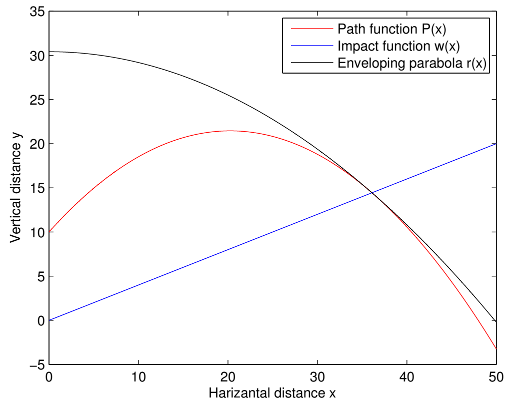

Figure 1: The enveloping parabola with linear impact function.

or

$$\phi_m = \arctan\left(\frac{v^2}{g\ x_m}\right). \tag{34}$$

An enveloping parabola is a path that encloses and intersects all possible paths. This enveloping parabola is obtained by maximizing the height of the projectile for a given horizontal distance x which will give the path that encloses all possible paths. Let $w = \tan\phi$, then Equation (30) becomes

$$y = P(x) = h + xw - \frac{gx^2}{2v^2}(1 + w^2). \tag{35}$$

Differentiating Equation (35) w.r.t. w and setting $y' = 0$, Henelsmith obtained

$$y' = x - \frac{gx^2}{v^2}(w) = 0, \ \ w = \frac{v^2}{gx}, \tag{36}$$

so that the enveloping parabola is defined by $y_m = \rho(x) = h + \frac{v^2}{2g} - \frac{gx^2}{2v^2}$.

The solution to the projectile problem requires first finding x_m which satisfies $\rho(x) = \omega(x)$ and solving for ϕ_m in Equation (34) because we want to find the point at which the enveloping parabola ρ intersects the impact function ω, and then find ϕ that corresponds to this point on the enveloping parabola. We choose a linear impact function $\omega(x) = 0.4x$ with $h = 10$ and $v = 20$. We let $g = 9.8$. Then we apply our I.F.s starting from $x_0 = 30$ to solve the non-linear equation

$$f(x) = \rho(x) - \omega(x) = h + \frac{v^2}{2g} - \frac{gx^2}{2v^2} - 0.4x,$$

whose root is given by $x_m = 36.102990117...$ and $\phi_m = \arctan\left(\frac{v^2}{g\ x_m}\right) = 48.5°$.

Figure 1 shows the intersection of the path function, the enveloping parabola and the linear impact function for this application when $5^{th}PJ$ method is applied.

Application 2: In the study of the multi-factor effect [18], the trajectory of an electron in the air gap between two parallel plates is given by

$$x(t) = x_0 + \big(v_0 + e\frac{E_0}{m\omega}sin(\omega t_0 + \Psi)\big)(t - t_0) + e\frac{E_0}{m\omega^2}\big(cos(\omega t + \Psi) + sin(\omega + \Psi)\big), \tag{37}$$

where $E_0 sin(\omega t + \Psi)$ is the RF electric field between plates at time t_0, x_0 and v_0 are the position and velocity of the electron, e and m are the charge and mass of the electron at rest respectively. For the particular parameters, one can deal with a simpler expression as follows:

$$f(x) = x - \frac{1}{2}cos(x) + \frac{\pi}{4}. \tag{38}$$

The required zero of the above function is $x^* \approx -0.3094661392082146514....$

Application 3: Van der Waals equation representing a real gas is given by [5]:

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

Here, a and b are parameters specific for each gas. This equation reduces to a nonlinear equation given by

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

By using the particular values for unknown constants, one can obtain the following nonlinear function

$$f(x) = 0.986x^3 - 5.181x^2 + 9.067x - 5.289, \tag{39}$$

having three zeros. Out of them, two are complex zeros and the third one is a real zero. However, our desired root is $x^* \approx 1.9298462428478622184875...$

Application 4: Generally, many problems in scientific and engineering which involve determination of any unknown appearing implicitly give rise to a root-finding problem. The Planck's radiation law problem appearing in [9, 16] is one among them and it is given by

$$\varphi(\lambda) = \frac{8\pi ch\lambda^{-5}}{e^{ch/\lambda kT} - 1}, \tag{40}$$

which calculates the energy density within an isothermal blackbody. Here, λ is the wavelength of the radiation; T is the absolute temperature of the blackbody; k is Boltzmann's constant; h is the Planck's constant; and c is the speed of light. Suppose we would like to determine wavelength λ, which corresponds to maximum energy density $\varphi(\lambda)$. From Equation (40), we get

$$\varphi'(\lambda) = \Big(\frac{8\pi ch\lambda^{-6}}{e^{ch/\lambda kT} - 1}\Big)\Big(\frac{(ch/\lambda kT)e^{ch/\lambda kT}}{e^{ch/\lambda kT} - 1} - 5\Big) = A \cdot B.$$

It can be checked that a maxima for φ occurs when $B = 0$, that is when $\Big(\frac{(ch/\lambda kT)e^{ch/\lambda kT}}{e^{ch/\lambda kT}-1}\Big) = 5$. Here, taking $x = ch/\lambda kT$, the above equation becomes

$$1 - \frac{x}{5} = e^{-x}. \tag{41}$$

Let us define

$$f(x) = e^{-x} - 1 + \frac{x}{5}. \tag{42}$$

The aim is to find a root of the equation $f(x) = 0$. Obviously, one of the root $x = 0$ is not taken for discussion. As argued in [9], the left-hand side of Equation (41) is zero for $x = 5$ and $e^{-5} \approx 6.74 \times 10^{-3}$. Hence, it is expected that another root of the equation $f(x) = 0$ might occur near $x = 5$. The approximate root of the Equation (42) is given by $x^* \approx 4.96511423174427630369$. Consequently, the wavelength of radiation (λ) corresponding to which the energy density is maximum is approximated as $\lambda \approx \frac{ch}{(kT)4.96511423174427630369}$.

Tables 7–10 display the numerical results with respect to number of iterations (N), $Error$, order of convergence (ρ) and CPU time (in seconds). The numerical experiments of the above real life problems demonstrate the validity and applicability of the proposed methods. It is observed that the presented methods take less CPU time and less than or equal number of iterations among the equivalent compared methods. This shows that the proposed methods are very much suitable for all the application problems. In most of the cases, the proposed methods show better performance in comparison to the existing methods.

Table 7: Comparison of results for Application 1.

Methods	N	error	ρ	$cpu(s)$
$2NR$	7	4.3980e-76	1.99	0.757150
$8KTM$	3	1.5610e-66	8.03	0.469166
$8LWM$	3	7.8416e-66	8.03	0.474202
$8PNPD$	3	0.474202	8.05	0.493403
$8CGFT$	3	3.3018e-89	9.03	0.509473
$8SAM$	3	1.2092e-61	8.06	0.497827
$8PMJ$	3	1.2696e-67	8.04	0.526071
$8PM$	3	4.3980e-76	8.02	0.500666
$16PM$	3	0	15.98	0.449256

Table 8: Comparison of results for Application 2.

Methods	N	error	ρ	$cpu(s)$
$2NR$	8	2.6110e-83	2.00	0.853926
$8KTM$	4	0	7.97	0.623539
$8LWM$	3	3.1704e-53	7.66	0.486217
$8PNPD$	4	3.8100e-210	7.99	0.613847
$8CGFT$	4	0	7.78	0.652285
$8SAM$	4	3.8586e-194	8.00	0.613134
$8PMJ$	4	0	7.89	0.654376
$8PM$	3	8.9487e-56	7.82	0.523860
$16PM$	3	0	15.99	0.456265

Table 9: Comparison of results for Application 3.

Methods	N	error	ρ	$cpu(s)$
$2NR$	10	3.1818e-79	1.99	1.030874
$8KTM$	4	2.1041e-60	7.95	0.605384
$8LWM$	4	7.1254e-60	7.95	0.651714
$8PNPD$	5	1.5540e-306	7.99	0.758008
$8CGFT$	4	6.3662e-74	8.03	0.636127
$8SAM$	4	4.5632e-122	7.99	0.599316
$8PMJ$	4	1.2474e-67	7.95	0.636168
$8PM$	4	5.9788e-88	7.98	0.532361
$16PM$	4	0	15.98	0.572305

Table 10: Comparison of results for Application 4.

Methods	N	error	ρ	$cpu(s)$
$2NR$	6	4.5985e-63	1.99	0.664445
$8KTM$	3	8.4034e-93	8.05	0.479138
$8LWM$	3	2.3181e-91	8.05	0.480359
$8PNPD$	3	5.4667e-91	8.05	0.491035
$8CGFT$	3	1.7403e-94	8.04	0.527014
$8SAM$	3	5.7651e-97	8.04	0.481452
$8PMJ$	3	2.6157e-93	8.05	0.512827
$8PM$	3	2.4364e-94	8.04	0.450025
$16PM$	3	0	16.02	0.439266

6 Conjugacy Maps and Extraneous Fixed Points

6.1 Conjugacy Maps for Quadratic Polynomials

In this section, we discuss the rational map R_p arising from $2NR$ and proposed method $8PM$ applied to a generic polynomial with simple roots.

Theorem 3 *For a rational map $R_p(z)$ arising from Newton's method (1) applied to $p(z) = (z-a)(z-b)$, $a \neq b$, $R_p(z)$ is conjugate via the Möbius transformation given by $M(z) = (z-a)/(z-b)$ to*

$$S(z) = z^2.$$

Proof. Let $p(z) = (z-a)(z-b)$, $a \neq b$, and let M be Möbius transformation given by $M(z) = (z-a)/(z-b)$ with its inverse $M^{-1}(z) = \frac{(zb-a)}{(z-1)}$, which may be considered as map from $\mathbb{C} \cup \{\infty\}$.
Then we have $S(z) = M \circ R_p \circ M^{-1}(z) = M\left(R_p\left(\frac{zb-a}{z-1}\right)\right) = z^2$. ∎

Theorem 4 *For a rational map $R_p(z)$ arising from Proposed Method (8) applied to $p(z) = (z-a)(z-b)$, $a \neq b$, $R_p(z)$ is conjugate via the Möbius transformation given by $M(z) = (z-a)/(z-b)$ to*

$$S(z) = z^8.$$

Proof. Let $p(z) = (z-a)(z-b)$, $a \neq b$, and let M be Möbius transformation given by $M(z) = (z-a)/(z-b)$ with its inverse $M^{-1}(z) = \frac{(zb-a)}{(z-1)}$, which may be considered as map from $\mathbb{C} \cup \{\infty\}$.

We then have $S(z) = M \circ R_p \circ M^{-1}(z) = M\left(R_p\left(\frac{zb-a}{z-1}\right)\right) = z^8$. ∎

Remark 1 *All the maps obtained above are of the form $S(z) = z^p R(z)$, where $R(z)$ is either unity or a rational function and p is the order of the method.*

Remark 2 *The conjugacy classes for the compared methods (24)–(29) can be calculated in a similar way with the help of* MATHEMATICA.

6.2 Extraneous Fixed Points

It is interesting to note that all the above discussed methods can be written as

$$x_{n+1} = x_n - G_f(x_n, y_n, w_n)u(x_n), \quad where\ u(x_n) = \frac{f(x_n)}{f'(x_n)}. \tag{43}$$

As per the definition, x^* is a fixed point of this method, since $u(x^*) = 0$. However, the points $\xi \neq x^*$ at which $G_f(\xi) = 0$ are also fixed points of the method, since $G_f(\xi) = 0$, second term on the right side of equation (43) vanishes. Hence, these points ξ are called extraneous fixed points.

Moreover, for a general iteration function given by

$$R(z) = z - G_f(z, y(z), w(z))u(z), \quad z \in \mathbb{C},$$

the nature of extraneous fixed points can be discussed. Based on the nature of the extraneous fixed points, the convergence of the iteration process will be determined. For example, Newton method does not have any extraneous fixed point, since $G_f = 1$. By following the works of Chun et al. [12] it is found that methods without extraneous fixed point or those having such points on the imaginary axis perform better than others. For the class of methods, they showed how to choose the parameter(s) such that the extraneous fixed points are on or close to the imaginary axis.

For more details on this aspect, the papers by Vrcsay *et al.* [28] and Neta et al. [19] will be useful. In fact, they investigated that if the extraneous fixed points are attractive then the method will give erroneous results. If the extraneous fixed points are repelling or neutral, then the method may not converge to a root near the initial guess. A point z_0 is called attracting if $|R'(z_0)| < 1$, repelling if $|R'(z_0)| > 1$ and neutral if $|R'(z_0)| = 1$. If the derivative is also zero then the point is super attracting. In order to find the extraneous fixed points, we substitute the quadratic polynomial $z^3 - 1$ for $f(z)$ and then find the zeros of G_f.

In this section, we have stated the theorem on extraneous fixed points for the methods $2NR$ and $8PM$ for the polynomial $z^3 - 1$. Extraneous Fixed Points for the methods $16PM$ and (24)–(29) can be calculated in a similar way with the help of MATHEMATICA.

Theorem 5 *There are no extraneous fixed points for Newton's Methods ($2NR$).*

Theorem 6 *There are twenty four extraneous fixed points for proposed method ($8PM$).*

Proof. For this $8PM$ method (8), we find that the extraneous fixed points are

$$\begin{aligned}&-0.5 \pm 0.866025i, \quad 0.25 \pm 0.433013i, \quad 0.411175 \pm 0.453532i, \quad -0.598358 \pm 0.129321i,\\ &-0.5, \quad 1, \quad 0.187183 \pm 0.582854i, 0.262684 \pm 0.401689i, \quad -0.479215 \pm 0.0266465i,\\ &0.216531 \pm 0.428336i, \quad 0.890221 \pm 1.04435i, \quad 0.459321 \pm 1.29313i, -1.34954 \pm 0.248781i.\end{aligned}$$

All these fixed points are repelling (since $|R'(z_0)| > 1$). ■

7 Basins of Attraction

The study on basins of attraction discussed below indicates that there are important aspects in which new method is superior than the other existing equivalent methods. This property of rational function associated to an iterative method acting on a polynomial gives us an important information about numerical features of the method for its stability and reliability. It is another way to compare the iterative methods. The basic definitions and dynamical concepts of rational function are found in [3, 22].

To obtain the basins of attraction of the root in terms of fractal graphs, consider a square $\mathbb{R} \times \mathbb{R} = [-3, 3] \times [-3, 3]$ in which we take $300 \times 300 = 90000$ initial points which contains all the roots $(z_j^*, j = 1, 2, 3, ...)$ of the concerned complex polynomial and we apply $8PM$ method starting at every initial point $z^{(0)}$ in the square. If the sequence generated by the iterative method converges to a root z_j^* of the polynomial with a tolerance $|f(z^{(k)})| < 10^{-4}$ and a maximum of 50 iterations, we decide that $z^{(0)}$ is in the basins of attraction of this root. If the iterative method starting in $z^{(0)}$ reaches a root in N iterations ($N \leq 50$), then this point $z^{(0)}$ is assigned with different light colors if $|z^{(N)} - z_j^*| < 10^{-4}$. If $N > 50$, we conclude that the starting point has diverged and it is assigned dark blue color. In the following, the basins of attraction for Newton's method, eighth order methods ($8KTM$, $8LWM$, $8PNPD$, $8CFGT$, $8SAM$ and $8PMJ$) and $8PM$ method

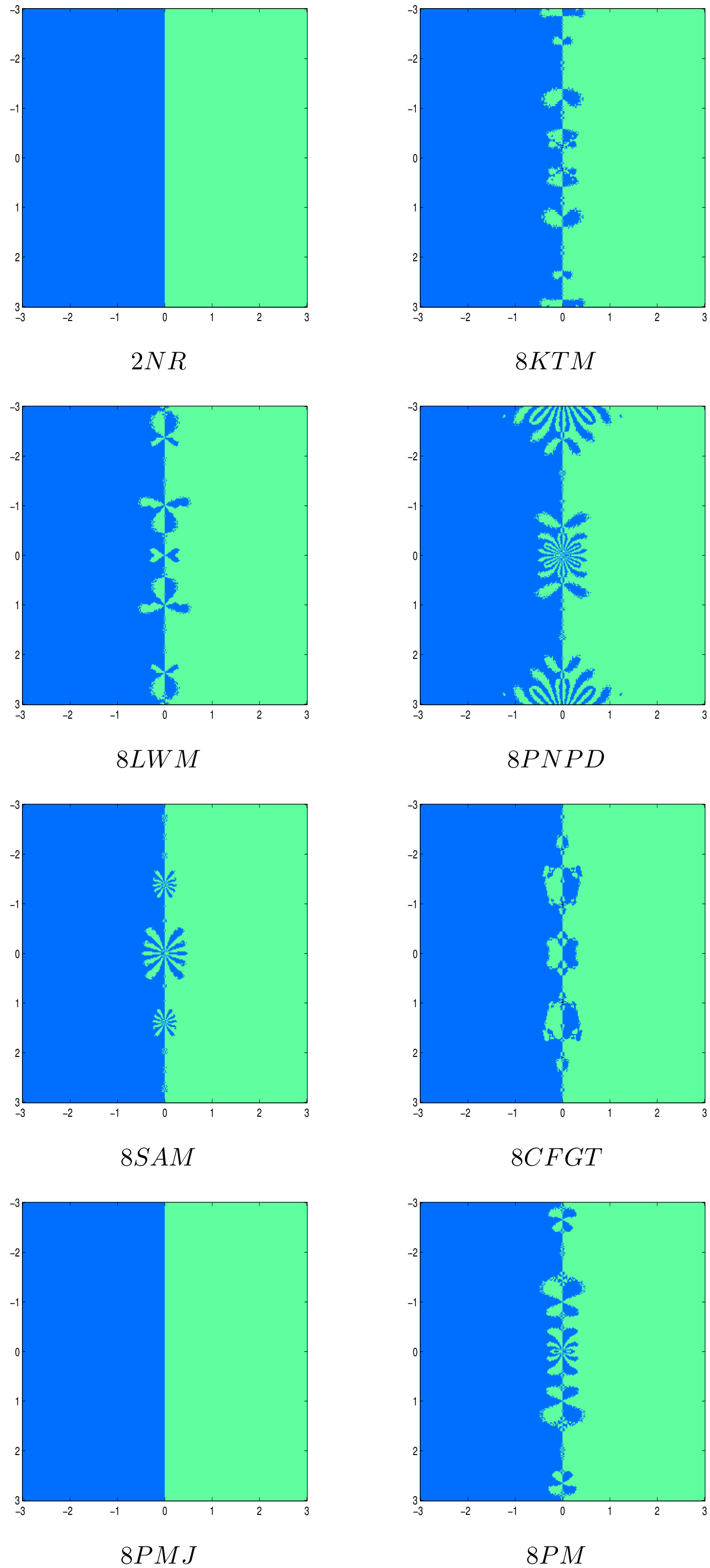

Figure 2: Basins of attraction for $p_1(z) = z^2 - 1$.

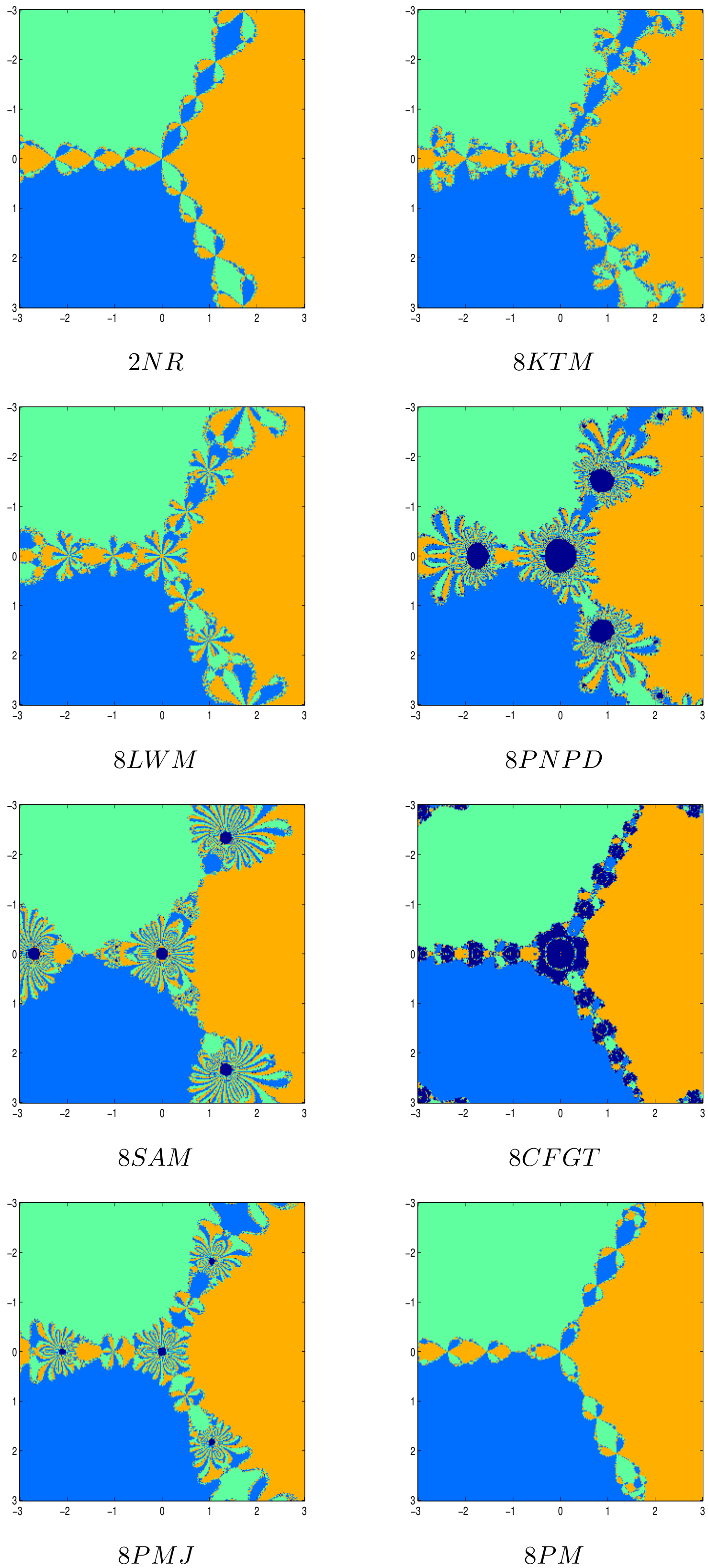

Figure 3: Basins of attraction for $p_2(z) = z^3 - 1$.

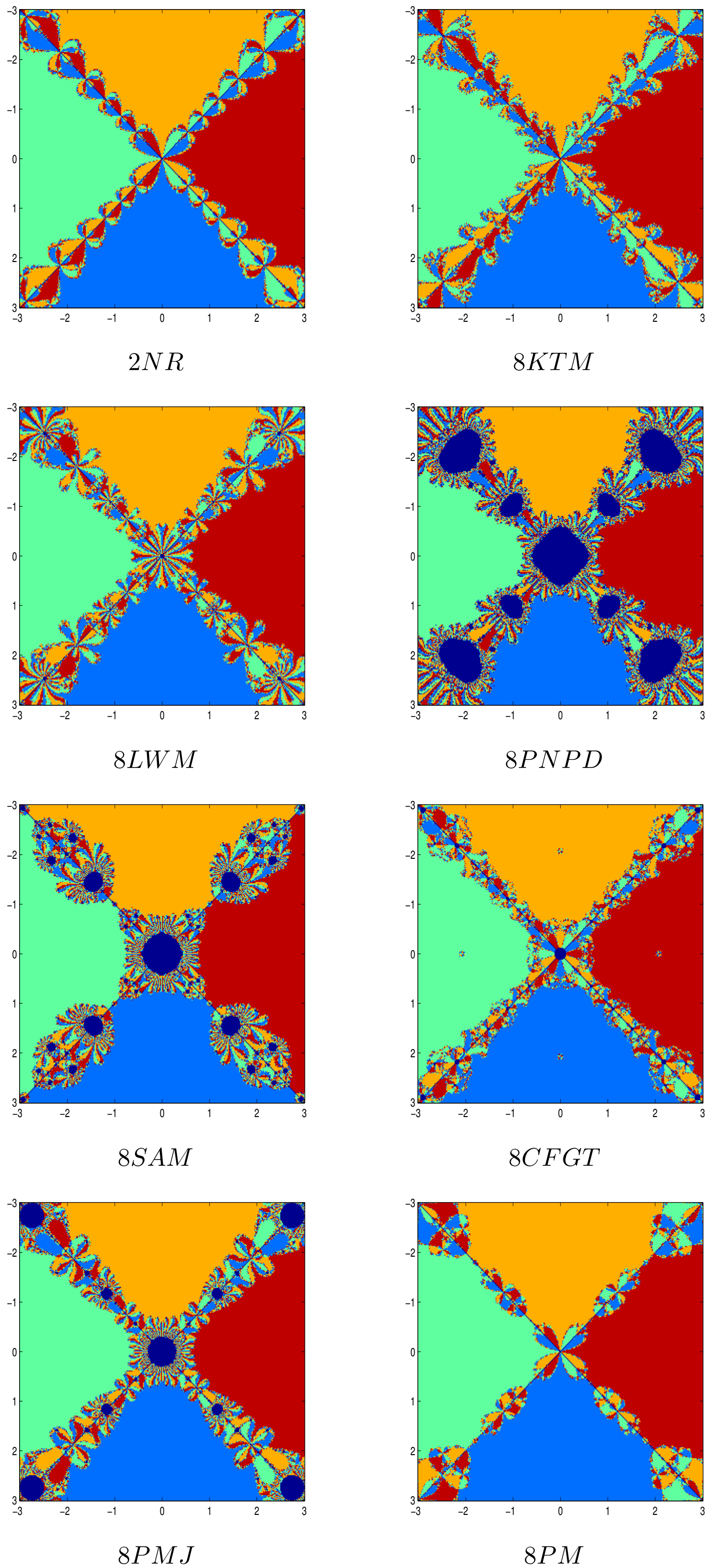

Figure 4: Basins of attraction for $p_3(z) = z^4 - 1$.

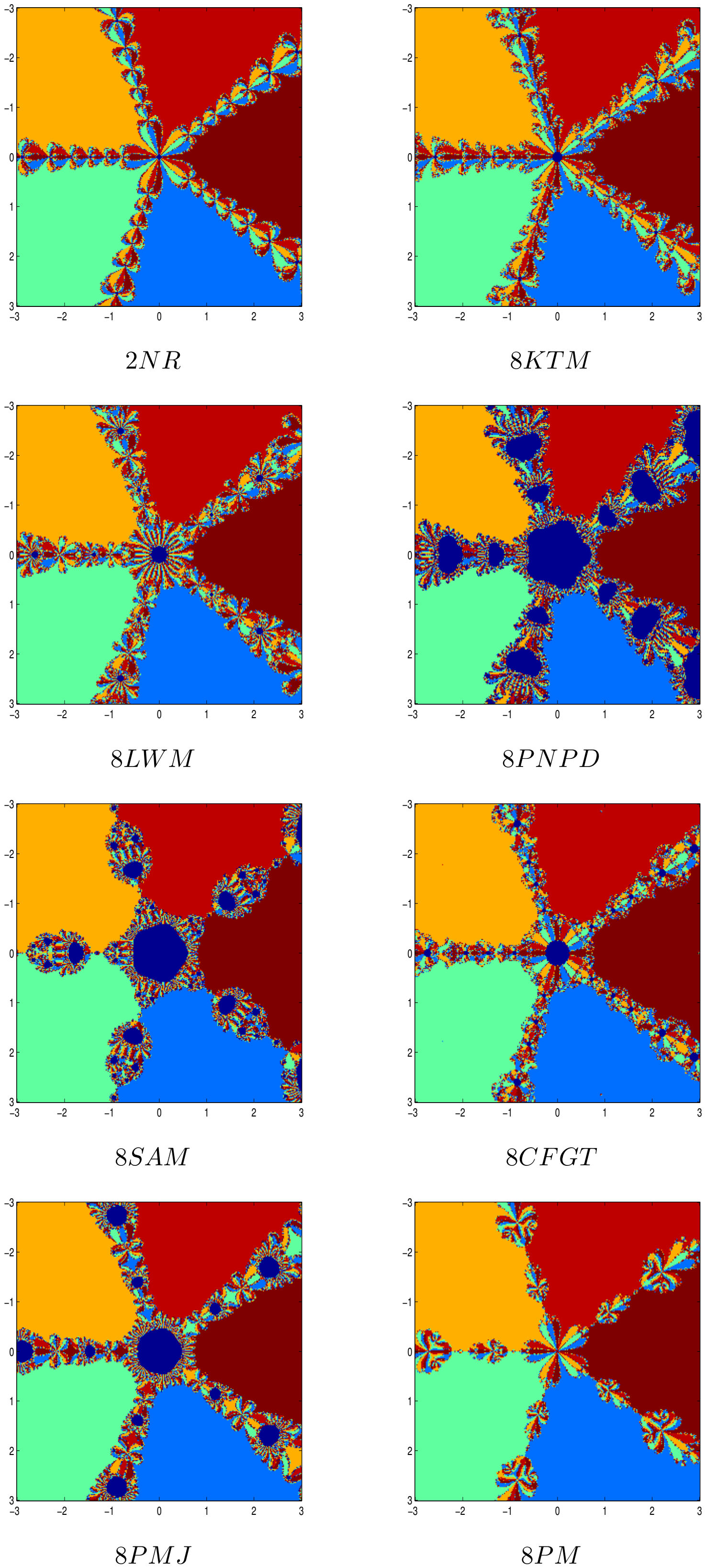

Figure 5: Basins of attraction for $p_4(z) = z^5 - 1$.

Table 11: Comparison of total number of convergent grid points and their average for $p_1(z) - p_4(z)$.

I.F.	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	Average
2NR	90000	90000	88704	89196	89475
8KTM	89984	90000	89026	89796	89701.5
8LWM	90000	90000	88660	88112	89193
8PNPD	90000	82114	63608	65916	75409.5
8SAM	90000	88296	79592	78172	84015
8CFGT	89944	83512	88679	88034	87542.25
8PMJ	90000	89462	81040	80114	85154
8PM	89992	90000	89024	90000	89754

Table 12: Comparison of mean number of iterations per convergent grid point for $p_1(z) - p_4(z)$.

I.F.	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	Average
2NR	5.1016	6.9759	9.6838	10.6046	8.091475
8KTM	2.3625	3.2639	4.5788	4.6225	3.706925
8LWM	2.3949	3.4772	5.7375	6.0645	4.418525
8PNPD	3.2421	7.7530	13.7335	11.5511	9.069925
8SAM	2.3459	5.5056	8.2523	7.4034	5.8768
8CFGT	2.166	2.8863	3.8561	3.8278	3.18405
8PMJ	2.0729	4.1832	7.7354	7.8420	5.458375
8PM	2.3963	2.4713	3.4510	3.8565	3.043775

Table 13: Comparison of CPU time and their average for $p_1(z) - p_4(z)$.

I.F.	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	Average
2NR	2.0515	2.7512	3.0423	3.9177	2.690675
8KTM	2.3426	3.0576	3.8564	3.2851	3.135425
8LWM	2.5355	3.1326	3.3531	3.4422	3.11585
8PNPD	2.7133	4.3497	6.0245	6.0368	4.781075
8SAM	2.3019	3.3592	3.6362	3.8710	3.292075
8CFGT	2.5273	3.2797	3.2527	3.3715	3.1078
8PMJ	2.6111	13.8762	20.4257	20.7798	14.4232
8PM	2.2514	3.0855	3.0961	3.2593	2.923075

are given for finding complex roots of the polynomials $p_1(z) = z^2 - 1$, $p_2(z) = z^3 - 1$, $p_3(z) = z^4 - 1$ and $p_4(z) = z^5 - 1$.

Note that a point z_0 belongs to the Julia set if and only if the dynamics in a neighborhood of z_0 displays sensitive dependence on the initial conditions, so that nearby initial conditions lead to wildly different behavior after a number of iterations. For this reason, some of the methods are getting divergent points. The common boundaries of these basins of attraction constitute the Julia set of the iteration function.

Figure 2 shows that the fractal graphs of the polynomial $p_1(z)$ for the proposed $8PM$ and other compared methods. From the fractal graphs we can see that the methods $2NR$ and $8PMJ$ perform very well since there is no chaotic behavior at all. For the methods $8KTM$, $8LWM$, $8PNPD$, $8SAM$, $8CFGT$ and $8PM$ show some chaotic behavior near the boundary points.

Figure 3 shows that the fractal graphs of the polynomial $p_2(z)$ for the proposed $8PM$ and other compared methods. It is seen from the fractal graphs that the methods $2NR$ and $8PM$ perform very well since there is no chaotic behavior at all. For the methods $8KTM$ and $8LWM$ show some chaotic behavior near the boundary points. The methods $8PNPD$, $8SAM$, $8CFGT$ and $8PMJ$ are sensitive according to the choice of initial guess in this case.

Figure 4 shows that the fractal graphs of the polynomial $p_3(z)$. We can see that the methods $2NR$ and $8PM$ perform well with least chaotic behavior. The methods $8KTM$, $8LWM$ and $8CFGT$ show some chaotic behavior near the boundary points. The methods $8PNPD$, $8SAM$ and $8PMJ$ are sensitive to the choice of initial guess in this case.

Figure 5 shows that the fractal graphs of the polynomial $p_4(z)$. We can see that $8PM$ method perform good. The methods $2NR$, $8KTM$ and $8CFGT$ show some chaotic behavior near the boundary points. The methods $8LWM$, $8PNPD$, $8SAM$ and $8PMJ$ are sensitive according to the choice of initial guess.

From the figures 2-5, one can see that the proposed method $8PM$ shows the best performance. It is clear that one has to use quantitative measures observed from the tables 11 to 13 to distinguish between the methods which is not possible only by viewing the fractal graphs using basins of attraction.

Besides basins of attraction, we have also done some quantitative comparison. For this, we have constructed three tables for all the considered methods on four polynomials $p_1(z)$, $p_2(z)$, $p_3(z)$ and $p_4(z)$. In table 11, we have calculated and compared total number of convergent grid points and their respective percentages. Table 12 displays mean number of iterations per convergent points. In Table 13 total cpu time (in seconds) with their average cpu time is compared. Based on the above information, we can conclude that, in terms of convergent points, the proposed method $8PM$ is the best for all polynomials. Interms of mean number of iterations per convergent points and interms of average cpu time also $8PM$ is the best and the fastest method compared with other equivalent methods.

8 Concluding Remarks

Based on the optimal two-point fourth order Ostrowski's scheme, two new optimal three-point and four-point methods without memory are developed for approximating a simple root of a given nonlinear equation. The methods use only four function evaluations and five function evaluations in each iteration and result in a method of convergence order eight and sixteen respectively. Therefore, the Kung and Traub's conjecture is found to be true for the new methods. Some numerical examples are tested using the proposed schemes and some existing schemes which illustrate the superiority of the proposed methods. Four application problems are solved where the new methods produce better results than other compared methods. Conjugacy mapping by using quadratic complex polynomial and extraneous fixed points of the proposed eighth order method are discussed. Further investigations have been made on the complex plane for such methods to reveal their basins of attraction for solving complex polynomials by presenting their corresponding fractal graphs. The numerical results of the proposed methods and their fractal graphs suggest that the new methods are valuable alternative for solving scalar nonlinear equation.

Acknowledgment. The authors would like to thank the anonymous reviewers for their constructive comments and useful suggestions which greatly helped to improve this paper.

References

[1] S. Amat, S. Busquier and Á.A. Magreñan, Reducing chaos and bifurcations in Newton-type methods, Abstr. Appl. Anal., Art., ID 726701 (2013), 10 pages.

[2] S. Amat, S. Busquier and S. Plaza, Dynamics of a family of third-order iterative methods that do not require using second derivatives, Appl. Math. Comp., 154(2004), 735–746.

[3] S. Amat, S. Busquier and S. Plaza. Review of some iterative root-finding methods from a dynamical point of view, SCIENTIA Series A: Mathematical Sciences, 10(2004), 3–35.

[4] S. Amat, S. Busquier and S. Plaza, Chaotic dynamics of a third-order Newton-type method, J. Math. Anal. Appl., 366(2010), 24–32.

[5] I. K. Argyros, M. Kansal, V. Kanwar and S. Bajaj, Higher-order derivative-free families of Chebyshev-Halley type methods with or without memory for solving nonlinear equations, Applied Mathematics and Computation, 315(2017), 224–245.

[6] D. K. R. Babajee, A. Cordero, F. Soleymani and J. R. Torregrosa, On improved three-step schemes with high efficiency index and their dynamics, Numer. Algor., 65(2014), 153–169.

[7] D. K. R. Babajee, K. Madhu and J. Jayaraman, A family of higher order multi-point iterative methods based on power mean for solving nonlinear equations, Afr. Mat., 27(2016), 865–876.

[8] R. Behl, A. S. Alshomrani and S. S. Motsa, An optimal scheme for multiple roots of nonlinear equations with eighth-order convergence, J. Math. Chem., 56(2018), 2069–2084.

[9] B. Bradie, A Friendly Introduction to Numerical Analysis, Pearson Education Inc., 2006.

[10] F. Chicharro, A. Cordero, J. M. Gutiérrez and J. R. Torregrosa, Complex dynamics of derivative-free methods for nonlinear equations, Appl. Math. Comput., 219(2013), 7023–7035.

[11] N. Choubey and J. P. Jaiswal, An improved optimal eighth-order iterative scheme with its dynamical behaviour, Int. J. Computing Science and Mathematics, 7(2016), 361–370.

[12] C. Chun and B. Neta, Comparative study of eighth-order methods for finding simple roots of nonlinear equations, Numer. Algor., 74(2017), 1169–1201.

[13] A. Cordero, M. Fardi, M. Ghasemi and J. R. Torregrosa, Accelerated iterative methods for finding solutions of nonlinear equations and their dynamical behavior, Calcolo, 51(2014), 17–30.

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

[15] L. Liu and X. Wang, Eighth-order methods with high efficiency index for solving nonlinear equations, Appl. Math. Comp., 215(2010), 3449–3454.

[16] T. Liu, X. Qin and Q. Li, An optimal fourth-order family of modified Cauchy methods for finding solutions of nonlinear equations and their dynamical behavior, Open Math., 17(2019), 1567–1598.

[17] K. Madhu and J. Jayaraman, Higher order methods for nonlinear equations and their basins of attraction, Mathematics, 4(2016), 22.

[18] P. Maroju, A. A. Magrenan, S. S. Motsa and I. Sarria, Second derivative free sixth order continuation method for solving nonlinear equations with application, J. Math. Chem., 56(2018), 2099–2116.

[19] B. Neta, M. Scott and C. Chun, Basin of attractions for several methods to find simple roots of nonlinear equations, Appl. Math. Comput., 218(2012), 10548–10556.

[20] A. M. Ostrowski, Solutions of Equations and System of equations, Academic Press, New York, 1960.

[21] M. S. Petković, B. Neta, L. D. Petković and J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Academic Press (an imprint of Elsevier), Waltham, MA, 2013.

[22] M. Scott, B. Neta and C. Chun, Basin attractors for various methods, Applied Mathematics and Computation, 218(2011), 2584–2599.

[23] J. R. Sharma and H. Arora, An efficient family of weighted-Newton methods with optimal eighth order convergence, Appl. Math. Lett., 29(2014), 1–6.

[24] J. R. Sharma and H. Arora, A new family of optimal eighth order methods with dynamics for nonlinear equations, Appl. Math. Compt., 273(2016), 924–933.

[25] P. Sivakumar, 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, 2(2018), 83–94.

[26] P. Sivakumar and J. Jayaraman, Some new higher order weighted Newton methods for solving nonlinear equation with applications, Math. Comput. Appl., 24(2019), 1567–1598.

[27] F. Soleymani, S. K. Khratti and S. Karimi Vanani, Two new classes of optimal Jarratt-type fourth-order methods, Appl. Math. Lett., 25(2011), 847–853.

[28] E. R. Vrscay and W. J. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for schroder and konig rational iteration functions, Numer. Math., 52(1988), 1–16.

[29] R. Wait, The Numerical Solution of Algebraic Equations, John Wiley & Sons, 1979.