MATHEMATICAL COMMUNICATIONS
Math. Commun. **25**(2020), 269–288

An optimal sixteenth order family of methods for solving nonlinear equations and their basins of attraction

Dejan Ćebić[1,*], Nebojša Ralević[2] and Marina Marčeta[2]

[1] *Faculty of Mining and Geology, University of Belgrade, Djušina 7, 11 000 Belgrade, Serbia*
[2] *Faculty of Engineering, University of Novi Sad, Trg Dositeja Obradovića 6, 21 000 Novi Sad, Serbia*

Received March 23, 2020; accepted August 24, 2020

Abstract. We propose a new family of iterative methods for finding simple roots of nonlinear equations. The proposed method is the four-point method with convergence order 16, which consists of four steps: the Newton step, an optional fourth order iteration scheme, an optional eighth order iteration scheme and the step constructed using the divided difference. By reason of the new iteration scheme requiring four function evaluations and one first derivative evaluation per iteration, the method satisfies the optimality criterion in the sense of Kung-Traub's conjecture and achieves a high efficiency index $16^{1/5} \approx 1.7411$. Computational results support theoretical analysis and confirm the efficiency. The basins of attraction of the new presented algorithms are also compared to the existing methods with encouraging results.

AMS subject classifications: 65H05, 65H99

Key words: nonlinear equation, sixteenth-order convergence, optimal methods, divided differences, basins of attraction

1. Introduction

The problem of solving nonlinear equations is frequent in many spheres of science and engineering. Solving this type of equations analytically is usually difficult. Consequently, many numerical methods for solving such problems have been developed. In this paper, we will focus on highly efficient multipoint iterative methods.

Newton's method is the best known iterative method for solving a nonlinear equation $f(x) = 0$, and it is defined by

$$f(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}.$$

If α is a simple root of the function $f(x)$, which means $f(\alpha) = 0$ and $f'(\alpha) \neq 0$, then Newton's method is quadratically convergent to α when the initial approximation x_0 is close enough to α.

*Corresponding author. *Email address:* `dejan.cebic@rgf.bg.ac.rs` (D. Ćebić), `nralevic@uns.ac.rs` (N. Ralević), `marceta.marina@gmail.com` (M. Marčeta)

Newton's method has been used as the foundation point for a significant number of multipoint methods constructed with the aim to improve the quadratic convergence order. Several multipoint methods were introduced by Ostrowski in [24], together with the coefficient $p^{\frac{1}{m}}$ as a measure of the efficiency of methods, where p is the convergence order and m is the number of functional evaluations per iteration. Later, Kung and Traub conjectured in [19] that any multipoint method with m functional evaluations per iteration can reach order 2^{m-1} at most. Methods that reach this order of convergence are called optimal methods. A systematic review of the most important aspects of multipoint methods with certain generalizations and historical notes can be found in a survey paper [26] by Petković et al. and a book [25] by the same authors.

In this paper, we are focused on the efficient and relevant 16th order optimal methods free from the second or any higher order derivatives. Therefore, we explore a new wide family of four-point methods, which uses four function evaluations and one derivative evaluation to achieve the 16th order of convergence. The structure of the paper is as follows. In Section 2, we develop a new optimal method. In Section 3, we present the numerical performance of the proposed method and compare it with already existing methods through several test examples. The basins of attraction of new algorithms are also displayed and compared visually and numerically to other methods. Finally, conclusion is provided in Section 4.

2. A new family of methods and its convergence

Inspired by recently established highly efficient eighth-order methods [27, 31, 32], we have used similar techniques based on the divided differences to develop a new class of four-point methods in the following form:

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = M_4(x_n, w_n), \\ y_n = M_8(x_n, w_n, z_n), \\ x_{n+1} = y_n - \dfrac{f(y_n)(2f[z_n,x_n] - 2f[y_n,x_n] + f[y_n,z_n])}{f'(x_n)(f[y_n,w_n] - f[z_n,w_n]) + f^2[z_n,x_n] - f^2[y_n,x_n] + f^2[y_n,z_n]}. \end{cases} \tag{1}$$

$M_4(\cdot,\cdot)$ and $M_8(\cdot,\cdot,\cdot)$ represent any optimal iterative scheme of fourth and eighth convergence order, respectively, with Newton's method as the first step, while $f[\cdot,\cdot]$ denotes the divided difference defined by $f[a,b] = \frac{f(a)-f(b)}{a-b}$.

Theorem 1. *Assume that function $f(x)$ is sufficiently differentiable in a neighborhood of its simple root α, and let $M_4(\cdot,\cdot)$ and $M_8(\cdot,\cdot,\cdot)$ be any optimal fourth and eighth order methods based on Newton's method, satisfying*

$$z_n - \alpha = \sum_{i=4}^{16} B_i e_n^i + O(e_n^{17}) \tag{2}$$

and

$$y_n - \alpha = \sum_{i=8}^{16} A_i e_n^i + O(e_n^{17}), \tag{3}$$

respectively, where $e_n = x_n - \alpha$, $B_4 \neq 0$ and $A_8 \neq 0$. Then for any starting approximation x_0 chosen close enough to α, method (1) is at least of sixteenth order.

Proof. Let $e_n = x_n - \alpha$ be the error of the n-th iteration. Then from Taylor's expansion of $f(x_n)$ and $f'(x_n)$ about α, we have

$$f(x_n) = f'(\alpha)\Big(e_n + \sum_{i=2}^{16} c_i e_n^i\Big) + O(e_n^{17}) \tag{4}$$

and

$$f'(x_n) = f'(\alpha)\Big(1 + \sum_{i=2}^{16} i c_i e_n^{i-1}\Big) + O(e_n^{16}), \tag{5}$$

where $c_i = \frac{f^{(i)}(\alpha)}{i! f'(\alpha)}$, for every integer $i \in \{2, ..., 16\}$.

Using (4) and (5) in Newton's step, we obtain its error $e_{w,n}$:

$$e_{w,n} = w_n - \alpha = \sum_{i=2}^{16} K_i e_n^i + O(e_n^{17}), \tag{6}$$

where $K_i = K_i(c_2, c_3, ..., c_i)$ with several explicitly written coefficients as follows:

$$\begin{aligned}
K_2 &= c_2,\\
K_3 &= -2c_2^2 + 2c_3,\\
K_4 &= 4c_2^3 - 7c_2^2 c_3 + 3c_4,\\
K_5 &= -8c_2^4 + 20c_2^2 c_3 - 6c_3^2 - 10c_2 c_4 + 4c_5,\\
K_6 &= 16c_2^5 - 52c_2^3 c_3 + 28c_2^2 c_4 - 17c_3 c_4 + c_2(33c_3^2 - 13c_5) + 5c_6,\\
K_7 &= -2(16c_2^6 - 64c_2^4 c_3 - 9c_3^3 + 36c_2^2 c_4 + 6c_4^2 + 9c_2^2(7c_3^2 - 2c_5)\\
&\quad +11c_3 c_5 + c_2(-46c_3 c_4 + 8c_6) - 3c_7),\\
&\vdots
\end{aligned}$$

Substituting $e_{w,n}$ from (6) into (4), we get

$$f(w_n) = f'(\alpha)\sum_{i=2}^{16} Q_i e_n^i + O(e_n^{17}), \tag{7}$$

where $Q_i = Q_i(c_2, c_3, ..., c_i, K_2, ..., K_i)$ with several explicitly written coefficients as follows:

$$
\begin{aligned}
Q_2 &= K_2 = c_2,\\
Q_3 &= K_3 = -2c_2^2 + 2c_3,\\
Q_4 &= c_2K_2^2 + K_4 = 5c_2^3 - 7c_2c_3 + 3c_4,\\
Q_5 &= 2c_2K_2K_3 + K_5 = -2(6c_2^4 - 12c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5),\\
Q_6 &= c_3K_2^3 + c_2(K_3^2 + 2K_2K_4) + K_6 = 28c_2^5 - 73c_2^3c_3 + 34c_2^2c_4 - 17c_3c_4\\
&\quad +c_2(37c_3^2 - 13c_5) + 5c_6,\\
Q_7 &= 3c_3K_2^2K_3 + 2c_2(K_3K_4 + K_2K_5) + K_7\\
&= -2(32c_2^6 - 103c_2^4c_3 - 9c_3^3 + 52c_2^3c_4 + 6c_4^2 + c_2^2(80c_3^2 - 22c_5) + 11c_3c_5\\
&\quad +c_2(-52c_3c_4 + 8c_6) - 3c_7),\\
&\ \ \vdots
\end{aligned}
$$

For any optimal method $M_4(x_n, w_n)$ that satisfies (2), from (4) we have

$$
\begin{aligned}
f(z_n) = f'(\alpha)\Big(&\sum_{i=4}^{7} B_i e_n^i + (B_8 + B_4^2c_2)e_n^8\\
&+(B_9 + 2B_4B_5c_2)e_n^9 + (B_{10} + (B_5^2 + 2B_4B_6)c_2)e_n^{10}\\
&+(B_{11} + 2(B_5B_6 + B_4B_7)c_2)e_n^{11} + (B_{12} + (B_6^2 + 2B_5B_7 + 2B_4B_8)c_2\\
&+B_4^3c_3)e_n^{12} + (B_{13} + 2(B_6B_7 + B_5B_8 + B_4B_9)c_2 + 3B_4^2B_5c_3)e_n^{13}\\
&+(B_{14} + (B_7^2 + 2(B_6B_8 + B_5B_9 + B_4B_{10}))c_2 + 3B_4(B_5^2 + B_4B_6)c_3)e_n^{14}\\
&+(B_{15} + 2(B_7B_8 + B_6B_9 + B_5B_{10} + B_4B_{11})c_2\\
&+(B_5^3 + 6B_4B_5B_6 + 3B_4^2B_7)c_3)e_n^{15}\\
&+(B_{16} + (B_8^2 + 2(B_7B_9 + B_6B_{10} + B_5B_{11} + B_4B_{12}))c_2\\
&+3(B_5^2B_6 + 2B_4B_5B_7 + B_4B_6^2 + B_4^2B_8)c_3 + B_4^4c_4)e_n^{16}\Big) + O(e_n^{17}).
\end{aligned}
\tag{8}
$$

Analogously, for any optimal $M_8(x_n, w_n, z_n)$ that satisfies (3), we get

$$
f(y_n) = f'(\alpha)\Big(\sum_{i=8}^{15} A_i e_n^i + (A_{16} + A_8^2c_2)e_n^{16}\Big) + O(e_n^{17}). \tag{9}
$$

Thus, using (4)-(9), with the aid of a Mathematica program package, it is uncomplicated to calculate any divided difference that appears in (1). Therefore, taking those results into account and substituting (4)-(9) into the fourth step of scheme (1), the following error equation is obtained:

$$
e_{n+1} = A_8\big(A_8c_2 + B_4(2B_4c_2^2 - c_2^2c_4 + c_3c_4 - c_2c_5)\big)e_n^{16} + O(e_n^{17}),
$$

which completes the proof. □

Remark 1. *Due to the robust length of some coefficients expressed in terms of c_i, such as K_8, K_9,...,K_{16}, Q_8, Q_9,...,Q_{16}, as well as the divided differences, we intentionally omit to display them for the sake of simplicity; still they can be efficiently derived using Mathematica symbolic computation.*

Seeing that the proposed scheme (1) requires five function/derivative evaluations per iteration, it is optimal in the sense of the Kung-Traub hypothesis. The efficiency index of the new method is $16^{1/5} \approx 1.7411$, which is better than the efficiency of optimal fourth order methods [4, 15, 16, 18, 21, 24] and optimal eighth order methods [2, 3, 5, 8, 9, 10, 17, 22, 27, 31, 32, 36] whose indices are 1.5874 and 1.6818, respectively. Although in further analysis this study will be concerned only with the optimal sixteenth order methods and their comparisons, several optimal methods of fourth and eighth order based on Newton's method are listed below since we have employed them as the second and the third step in (1) to construct concrete algorithms of the new family.

- Fourth order choices for $M_4(x_n, w_n)$

1) $M_4(x_n, w_n) = w_n - \dfrac{f(w_n)}{2f[w_n, x_n] - f'(x_n)}$, from [24],

2) $M_4(x_n, w_n) = w_n - \big(\dfrac{2}{f[w_n, x_n]} - \dfrac{1}{f'(x_n)}\big) f(w_n)$, from [13],

3) $M_4(x_n, w_n) = w_n - \big(3 - \dfrac{2f[w_n, x_n]}{f'(x_n)}\big)\dfrac{f(w_n)}{f'(x_n)}$, from [31].

- Eighth order choices for $M_8(x_n, w_n, z_n)$

A) $M_8(x_n, w_n, z_n) = z_n + \dfrac{f(z_n)}{f[z_n, x_n]} \dfrac{f[z_n, w_n]}{f[z_n, x_n] - 2f[z_n, w_n]}$, from [27, 32],

B) $M_8(x_n, w_n, z_n) = z_n - \dfrac{f(z_n)}{f'(x_n)} \dfrac{f'(x_n) - f[w_n, x_n] + f[z_n, w_n]}{2f[z_n, w_n] - f[z_n, x_n]}$, from [31].

Thus, we consider six special cases of (1) denoted by **NMXY**, where **X** suggests which function $M_4(x_n, w_n)$ has been used, while **Y** denotes the choice of $M_8(x_n, w_n, z_n)$. For example, the algorithm denoted by **NM2A** has the following form:

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \big(\dfrac{2}{f[w_n, x_n]} - \dfrac{1}{f'(x_n)}\big) f(w_n), \\ y_n = z_n + \dfrac{f(z_n)}{f[z_n, x_n]} \dfrac{f[z_n, w_n]}{f[z_n, x_n] - 2f[z_n, w_n]}, \\ x_{n+1} = y_n - \dfrac{f(y_n)(2f[z_n, x_n] - 2f[y_n, x_n] + f[y_n, z_n])}{f'(x_n)(f[y_n, w_n] - f[z_n, w_n]) + f^2[z_n, x_n] - f^2[y_n, x_n] + f^2[y_n, z_n]}. \end{cases}$$

Appropriate error equations for six methods obtained are displayed in Table 1.

method	error constant C
NM1A	$c_2^2(c_3-c_2^2)^2(c_2c_4-c_3^2)(2c_2^5-2c_2^3c_3+c_3c_4-c_2(c_3^2+c_5))$
NM2A	$c_2^2(c_3-3c_2^2)^2(c_2c_4-c_3^2)(6c_2^5-2c_2^3c_3+c_3c_4-c_2(c_3^2+c_5))$
NM3A	$c_2^2(c_3-5c_2^2)^2(c_2c_4-c_3^2)(10c_2^5-2c_2^3c_3+c_3c_4-c_2(c_3^2+c_5))$
NM1B	$-c_2^2(c_3-c_2^2)^2(c_2^4-c_2^2c_3+c_3^2-c_2c_4)(c_2^5-c_2^3c_3+c_3c_4-c_2(c_3^2+c_5))$
NM2B	$-c_2^2(c_3-3c_2^2)^2(3c_2^4-c_2^2c_3+c_3^2-c_2c_4)(3c_2^5-c_2^3c_3+c_3c_4-c_2(c_3^2+c_5))$
NM3B	$-c_2^2(c_3-5c_2^2)^2(5c_2^4-c_2^2c_3+c_3^2-c_2c_4)(5c_2^5-c_2^3c_3+c_3c_4-c_2(c_3^2+c_5))$

Table 1: *Error equations* $e_{n+1}=C\cdot e_n^{16}+O(e^{17})$ *for special members of family (1)*

3. Numerical results

3.1. Numerical implementation and comparison

The comparison methods used in this paper have been theoretically and numerically proven as the most efficient once through a vast number of test functions. Classes of methods given below are the ones suggested by the authors in the corresponding papers. The performance of our method is compared with the performance of the following methods. Some additional optimal sixteenth order methods that have not been included in this research can be found in [11, 23, 37].

The method developed by H.T. Kung and J.F. Traub [19] denoted by MKT is:

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - G_f(x_n), \\ y_n = z_n - f^2(x_n)f(w_n)H_f(x_n,w_n,z_n), \\ x_{n+1} = y_n + \dfrac{f^2(x_n)f(w_n)f(z_n)}{f(x_n)-f(y_n)} \\ \qquad\times\Big(H_f(x_n,w_n,z_n) - \dfrac{K_f(x_n,w_n,z_n)-L_f(x_n,w_n,z_n,y_n)}{f(x_n)-f(y_n)}\Big), \end{cases}$$

where

$$\begin{aligned} G_f(x_n) &= \frac{f^2(x_n)f(w_n)}{f'(x_n)(f(x_n)-f(w_n))^2}, \\ H_f(x_n,w_n,z_n) &= G_f(x_n)\Big(\frac{-1}{f^2(x_n)(f(x_n)-f(z_n))} \\ &\quad+\frac{f(w_n)-f(x_n)}{f(x_n)f(w_n)(f(x_n)-f(z_n))^2} \\ &\quad+\frac{1}{(f(w_n)-f(z_n))(f(x_n))-f(z_n))^2}\Big), \end{aligned}$$

$$
\begin{aligned}
K_f(x_n,w_n,z_n) &= \frac{f(x_n)(f(w_n)-f(z_n))(f(x_n)-f(w_n))-f^2(x_n)f(w_n)}{f'(x_n)(f(x_n)-f(z_n))(f(x_n)-f(w_n))^2(f(w_n)-f(z_n))},\\
L_f(x_n,w_n,z_n,y_n) &= \frac{G_f(x_n)}{(f(w_n)-f(y_n))(f(w_n)-f(z_n))}\\
&\quad -\frac{f(w_n)f^2(x_n)H_f(x_n,w_n,z_n)}{(f(w_n)-f(y_n))(f(z_n)-f(y_n))}.
\end{aligned}
$$

The method developed by Sharma et al. [33] denoted by MSGG is:

$$
\begin{cases}
w_n = x_n - \dfrac{f(x_n)}{f'(x_n)},\\
z_n = w_n - \dfrac{f(x_n)+\beta f(w_n)}{f(x_n)+(\beta-2)f(w_n)}\dfrac{f(w_n)}{f'(x_n)},\\
y_n = x_n - \dfrac{P+Q+R}{Pf[z_n,x_n]+Qf'(x_n)+Rf[w_n,x_n]}f(x_n),\\
x_{n+1} = x_n - \dfrac{P_1f[z_n,w_n]+Q_1f[x_n,w_n]+Rf[y_n,w_n]}{P_1L+Q_1M+RN}f(x_n),
\end{cases}
$$

where

$$
\begin{aligned}
&\beta=1, P=(x_n-w_n)f(x_n)f(w_n), Q=(w_n-z_n)f(z_n)f(w_n),\\
&R=(z_n-x_n)f(z_n)f(x_n), P_1=(x_n-y_n)f(x_n)f(y_n),\\
&Q_1=(y_n-z_n)f(y_n)f(z_n), L=\frac{f(w_n)f[x_n,z_n]-f(z_n)f[x_n,w_n]}{w_n-z_n},\\
&M=\frac{f(w_n)f'(x_n)-f(x_n)f[x_n,w_n]}{w_n-x_n} \quad \text{and} \quad N=\frac{f(w_n)f[x_n,y_n]-f(y_n)f[x_n,w_n]}{w_n-y_n}.
\end{aligned}
$$

Sharifi et al. have investigated a class of four-point methods in [30]. Here we employ a member of this class, denoted by MSSSL, which achieved the best numerical results in [30].

$$
\begin{cases}
w_n = x_n - \dfrac{f(x_n)}{f'(x_n)},\\
z_n = w_n - ((1+t_n^2)(1+2t_n+2t_n^2)+t_n^2(2-8t_n-2t_n^2))\cdot\dfrac{f(w_n)}{f'(x_n)},\\
y_n = z_n - \big(4u_n-5v_n+(6+v_n^3)(t_n^2+v_n)+(1+u_n^3)(1+2t_n)\big)\cdot\dfrac{f(z_n)}{f'(x_n)},\\
x_{n+1} = y_n - \Big[(1+t_n)(2t_n+t_n^3)+4t_n^2-t_n^3-t_n^4-2v_n^2+6u_n+2t_nr_n+2v_nu_n\\
\qquad +24t_n^4u_n+t_nu_n+\dfrac{2t_n^3u_n-10t_nu_n^2+6t_n^2u_n}{1+2t_nu_n}+\dfrac{1+2p_n+2q_n}{1-r_n}+\dfrac{6p_n}{1+q_n}\\
\qquad -\dfrac{2u_n+6u_n^2}{1+u_n}+\dfrac{v_n+2v_n^2}{1+v_n^2}+\dfrac{6t_n^2r_n+6t_n^3r_n-4v_n^2u_n}{1+t_n}\Big]\cdot\dfrac{f(y_n)}{f'(x_n)},
\end{cases}
$$

where $t_n = \dfrac{f(w_n)}{f(x_n)}$, $v_n = \dfrac{f(z_n)}{f(w_n)}$, $u_n = \dfrac{f(z_n)}{f(x_n)}$, $p_n = \dfrac{f(y_n)}{f(x_n)}$, $q_n = \dfrac{f(y_n)}{f(w_n)}$ and $r_n = \dfrac{f(y_n)}{f(z_n)}$.

Maroju et al. have proposed eighth and sixteenth-order families of King's methods [20]. For the purpose of comparison we use a sixteenth-order method with acronym MMBM, given by:

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \dfrac{f(x_n)+\beta f(w_n)}{f(x_n)+(\beta-2)f(w_n)}\dfrac{f(w_n)}{f'(x_n)}, \\ y_n = z_n - \dfrac{f(z_n)}{f'(x_n)} \cdot \dfrac{\theta_4}{2\beta+2(\beta^2-6\beta+6)v-5}, \\ x_{n+1} = x_n - \theta_5 f(x_n), \end{cases}$$

where

$$\begin{aligned} \theta_4 &= 2\beta + u(2\beta+2(\beta^2-2\beta-4)v-5) - (4\beta+1)v^2 + 2(\beta^2-4\beta+1)v - 5, \\ \theta_5 &= \frac{a_n b_n\big(u_1 f(x_n)^2 f(w_n) + u_2 f'(x_n) f(y_n) f(z_n)\big)}{v_1 f(x_n)^3 + v_2 f'(x_n) f(y_n) f(z_n)}, \end{aligned} \tag{10}$$

for

$$\begin{aligned} u_1 &= f(y_n)\big(b_n^2 f'(x_n) + b_n f(x_n) - c_n f(z_n)\big) + a_n\big(f(x_n) - a_n f'(x_n)\big) f(z_n), \\ u_2 &= a_n b_n c_n f'(x_n)\big(f(w_n) - f(x_n)\big) + c_n f(w_n) f(x_n)\big(a_n - b_n\big), \\ v_1 &= f(w_n)\big[b_n f(y_n)\big(b_n^2 f'(x_n) + b_n f(x_n) - c_n f(z_n)\big) \\ &\quad + \big(a_n^3 f'(x_n) + c_n a_n f(y_n) - a_n^2 f(x_n)\big) f(z_n)\big], \\ v_2 &= a_n^2 b_n^2 c_n f'(x_n)^2\big(2f(w_n) - f(x_n)\big) + a_n b_n c_n (2a_n - c_n) f'(x_n) f(w_n) f(x_n) \\ &\quad + c_n\big(a_n b_n - a_n c_n - b_n^2\big) f(w_n) f(x_n)^2, \\ \beta &= 1, a_n = x_n - z_n, b_n = y_n - x_n, c_n = y_n - z_n, u = \frac{f(z_n)}{f(w_n)} \quad \text{and} \quad v = \frac{f(w_n)}{f(x_n)}. \end{aligned}$$

Latterly, Behl et al. have proposed a more general family (see [1] for details). We have chosen a special case 1 created by authors of the original paper.

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \dfrac{f(x_n)}{f'(x_n)} \cdot \dfrac{f(x_n)-f(w_n)}{f(x_n)-2f(w_n)}, \\ y_n = u_n - \dfrac{f(z_n)}{f'(x_n)} \cdot \dfrac{3(\beta_2+\beta_3)(u_n-z_n)}{\beta_1(u_n-z_n)+\beta_2(w_n-x_n)+\beta_3(z_n-x_n)}, \\ x_{n+1} = x_n - \theta_5 f(x_n), \end{cases}$$

where $u_n = z_n - \dfrac{f(z_n)}{f'(x_n)}\left[\dfrac{f(x_n)-f(w_n)}{f(x_n)-2f(w_n)} + \dfrac{1}{2}\dfrac{f(z_n)}{f(w_n)-2f(z_n)}\right]^2$ and $\beta_2+\beta_3 \neq 0$, while θ_5 has the form (10). This method is denoted by MBAMM with parametric values $\beta_1 = 0$, $\beta_2 = 1$ and $\beta_3 = 0$.

Recently, Geum et al. have constructed an optimal class of generic simple root finders. They have tested a vast number of methods, here we take the one with the lowest CPU time and an average number of iterations as reported by the test examples in the original research [12]. The initialism that we use for this method is MGKN.

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \dfrac{f(x_n)}{f'(x_n)} \cdot Q_f(s), \quad s = \dfrac{f(y_n)}{f(x_n)}, \\ w_n = z_n - \dfrac{f(y_n)}{f'(x_n)} \cdot K_f(s,u), \quad u = \dfrac{f(z_n)}{f(y_n)}, \\ x_{n+1} = w_n - \dfrac{f(w_n)}{f'(x_n)} \cdot J_f(s,u,v), \quad v = \dfrac{f(w_n)}{f(z_n)}, \end{cases}$$

where

$$Q_f(s) = \frac{1}{1-2s}, \quad K_f(s,u) = Q_f(s) \cdot \frac{(s-1)^2}{1-2s-u+2s^2u},$$

$$J_f(s,u,v) = K_f(s,u) \cdot \frac{1-s-s^2-2s^3+(-1-s+s^2)u+2su^2}{1-s-s^2-2s^3+(-1-s-s^3-s^4)u+(-1+s+s^2+2s^3)v}.$$

Salimi and Behl have developed an optimal family in [28], from this family we use a special member that has shown the best numerical performance in the original paper. This method is denoted by MSB with the following form:

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \dfrac{f(x_n)}{f'(x_n)} \cdot \dfrac{f(w_n)}{f(x_n)-2f(w_n)}, \\ t_n = z_n - \dfrac{f(x_n)f(w_n)f(z_n) \cdot Q}{f'(x_n)(-2f(w_n)+f(x_n))(f(w_n)-f(z_n))}, \\ x_{n+1} = x_n + \dfrac{\theta_2 P_1}{\theta_3 + \theta_2 P_2}, \end{cases}$$

where

$$Q = 1 - \frac{f(x_n)}{2f(x_n)} - \frac{bf(x_n)(f(w_n)+4f(z_n))}{2(2f(w_n)-f(x_n))(bf(x_n)-f(z_n))}, \quad (b=-0.5),$$

$$\theta_2 = f(x_n)(f(t_n)-f(x_n))(f(x_n)-f(w_n))(f(x_n)-f(z_n)),$$

$$\begin{aligned}\theta_3 &= f'(x_n)f(t_n)f(z_n)(f(t_n)-f(z_n))(t_n-x_n)(x_n-z_n)\\&\quad\times\big[(f(x_n)-f(z_n))\big(-f(x_n)(f(w_n)+2f(x_n)-2f(z_n))+f(w_n)^2\\&\quad+(f(w_n)+2f(x_n))(f(x_n)-f(z_n))\big)+f(x_n)(f(t_n)-f(x_n))(f(x_n)-f(z_n))\big],\\P_1 &= (f(t_n)-f(w_n))(f(t_n)-f(z_n))(f(w_n)-f(z_n))(t_n-x_n)(z_n-x_n),\\P_2 &= f(w_n)[f(t_n)(t_n-x_n)(f(t_n)-f(w_n))-f(z_n)(x_n-z_n)(f(w_n)-f(z_n))].\end{aligned}$$

Tao and Madhu have proposed the optimal fourth, eighth and sixteenth order methods [35]. The sixteenth order method denoted by MTM could be written as follows:

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)},\\ z_n = y_n - \dfrac{f(y_n)}{f'(x_n)+2f[y,x,x](y_n-x_n)},\\ w_n = z_n - \dfrac{f(z_n)}{f'(x_n)+2b_2(z_n-x_n)+3b_3(z_n-x_n)^2},\\ x_{n+1} = w_n - \dfrac{f(w_n)}{f'(x_n)+2a_2(w_n-x_n)+3a_3(w_n-x_n)^2+4a_4(w_n-x_n)^3},\end{cases}$$

where

$$\begin{aligned}b_2 &= \frac{f[y,x,x](z_n-x_n)-f[z,x,x](y_n-x_n)}{z_n-y_n},\\ b_3 &= \frac{f[y,x,x]-f[z,x,x]}{z_n-y_n},\\ a_2 &= \frac{f[y,x,x](-s_2^2s_3+s_2s_3^2)+f[z,x,x](s_1^2s_3-s_1s_3^2)+f[w,x,x](-s_1^2s_2+s_1s_2^2)}{-s_1^2s_2+s_1s_2^2+s_1^2s_3-s_2^2s_3-s_1s_3^2+s_2s_3^2},\\ a_3 &= \frac{f[y,x,x](s_2^2-s_3^2)+f[z,x,x](-s_1^2+s_3^2)+f[w,x,x](s_1^2-s_2^2)}{-s_1^2s_2+s_1s_2^2+s_1^2s_3-s_2^2s_3-s_1s_3^2+s_2s_3^2},\\ a_4 &= \frac{f[y,x,x](-s_2+s_3)+f[z,x,x](s_1-s_3)+f[w,x,x](-s_1+s_2)}{-s_1^2s_2+s_1s_2^2+s_1^2s_3-s_2^2s_3-s_1s_3^2+s_2s_3^2},\\ s_1 &= y_n-x_n, s_2=z_n-x_n, s_3=w_n-x_n \quad \text{and} \quad f[t,x,x]=\frac{f[t_n,x_n]-f'(x_n)}{t_n-x_n}.\end{aligned}$$

For the interpretation of the numerical behaviour and computational efficiency of the proposed methods, we have used test examples and appropriate initial approximations displayed in Table 2. Functions f_1, f_2 and f_3 are derived from the acclaimed real-life problems such as the real gas behavior explained by the van der Waals equation of state, the fractional conversion and the equation derived from Plank's radiation law, respectively.

$f_n(x)$	α	x_0
$f_1(x) = 0.986x^3 - 5.181x^2 + 9.067x - 5.289$; [20]	1.9298462...	2
$f_2(x) = x^4 - 7.79075x^3 + 14.7445x^2 + 2.511x - 1.674$; [20]	$3.9... + i \cdot 0.3...$	$3.7 + i/4$
$f_3(x) = e^{-x} - 1 + x/5$; [14]	4.965114...	3
$f_4(x) = \log(x^2 + x + 2) - x + 1$; [32]	4.1525907...	3
$f_5(x) = x\log(1 + x\sin x) + e^{-1+x^2+\cos x}\sin(\pi x)$; [30]	0	0.01
$f_6(x) = \frac{-2}{27}(9\sqrt{2}+7\sqrt{3}) + \sqrt{1-x^2} + (1+x^3)\cos(\pi x/2)$; [30]	1/3	0.35
$f_7(x) = \log(1 + x^2) + e^x \sin x$; [30]	0	0.1

Table 2: *Test functions*

All computations have been carried out by *Mathematica* using the *SetPrecision* function with 10000 significant digits. A computer with the Windows 10 Pro 64-bit operating system and the AMD Ryzen 7 1700 Eight-Core CPU @ 3.00 GHz processor has been used for all numerical calculations.

method	it	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	COC	CPU
MKT	3	$5.1027 \cdot 10^{-8}$	$1.0011 \cdot 10^{-103}$	$4.8228 \cdot 10^{-1635}$	16.000	0.0106
MSGG	3	$8.3927 \cdot 10^{-10}$	$1.8082 \cdot 10^{-134}$	$3.8979 \cdot 10^{-2129}$	16.000	0.0144
MSSSL	3	$7.5196 \cdot 10^{-7}$	$1.3646 \cdot 10^{-84}$	$1.8883 \cdot 10^{-1328}$	16.000	0.0156
MMBM	3	$1.1749 \cdot 10^{-8}$	$1.9665 \cdot 10^{-114}$	$7.4638 \cdot 10^{-1807}$	16.000	0.0156
MBAMM	3	$5.6460 \cdot 10^{-10}$	$1.0170 \cdot 10^{-137}$	$1.2492 \cdot 10^{-2181}$	16.000	0.0156
MGKN	3	$3.8085 \cdot 10^{-10}$	$2.2160 \cdot 10^{-140}$	$3.8251 \cdot 10^{-2224}$	16.000	0.0100
MSB	3	$2.1280 \cdot 10^{-9}$	$3.3608 \cdot 10^{-128}$	$5.0337 \cdot 10^{-2029}$	16.000	0.0125
MTM	3	$9.1473 \cdot 10^{-10}$	$3.3763 \cdot 10^{-134}$	$4.0077 \cdot 10^{-2125}$	16.000	0.0250
NM1A	3	$1.8044 \cdot 10^{-10}$	$4.4746 \cdot 10^{-146}$	$9.1519 \cdot 10^{-2316}$	16.000	0.00816
NM2A	3	$2.1597 \cdot 10^{-10}$	$4.8969 \cdot 10^{-143}$	$2.3902 \cdot 10^{-2265}$	16.000	0.00812
NM3A	3	$3.5589 \cdot 10^{-9}$	$7.7187 \cdot 10^{-123}$	$1.8504 \cdot 10^{-1941}$	16.000	0.00872
NM1B	3	$9.2506 \cdot 10^{-10}$	$3.9672 \cdot 10^{-134}$	$5.1935 \cdot 10^{-2124}$	16.000	0.00752
NM2B	3	$1.9928 \cdot 10^{-8}$	$1.9741 \cdot 10^{-110}$	$1.6981 \cdot 10^{-1742}$	16.000	0.00812
NM3B	3	$5.9879 \cdot 10^{-8}$	$8.0420 \cdot 10^{-102}$	$9.0108 \cdot 10^{-1604}$	16.000	0.00876

Table 3: *Numerical results for* $f_1(x)$

method	it	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	COC	CPU
MKT	3	$2.7372 \cdot 10^{-4}$	$4.7461 \cdot 10^{-52}$	$3.1852 \cdot 10^{-816}$	16.000	0.0431
MSGG	3	$4.7686 \cdot 10^{-6}$	$9.8749 \cdot 10^{-82}$	$1.1294 \cdot 10^{-1292}$	16.000	0.0569
MSSSL	–	–	–	–	–	–
MMBM	3	$2.1478 \cdot 10^{-4}$	$7.6017 \cdot 10^{-54}$	$4.5984 \cdot 10^{-845}$	16.000	0.0644
MBAMM	3	$1.5788 \cdot 10^{-6}$	$5.9057 \cdot 10^{-90}$	$8.6768 \cdot 10^{-1425}$	16.000	0.0662
MGKN	3	$8.2841 \cdot 10^{-7}$	$8.5845 \cdot 10^{-95}$	$1.5178 \cdot 10^{-1502}$	16.000	0.0388
MSB	3	$1.9090 \cdot 10^{-5}$	$7.0180 \cdot 10^{-73}$	$7.8102 \cdot 10^{-1152}$	16.000	0.0494
MTM	3	$8.2254 \cdot 10^{-7}$	$6.9022 \cdot 10^{-95}$	$4.1718 \cdot 10^{-1504}$	16.000	0.0912
NM1A	3	$1.5704 \cdot 10^{-7}$	$2.9890 \cdot 10^{-107}$	$8.8647 \cdot 10^{-1703}$	16.000	0.0313
NM2A	3	$9.4718 \cdot 10^{-6}$	$3.1312 \cdot 10^{-77}$	$6.3690 \cdot 10^{-1221}$	16.000	0.0312
NM3A	3	$3.4343 \cdot 10^{-5}$	$1.3675 \cdot 10^{-67}$	$5.4670 \cdot 10^{-1066}$	16.000	0.0344
NM1B	3	$1.3351 \cdot 10^{-6}$	$1.6749 \cdot 10^{-91}$	$6.3068 \cdot 10^{-1450}$	16.000	0.0312
NM2B	3	$6.4308 \cdot 10^{-5}$	$1.5839 \cdot 10^{-62}$	$2.9033 \cdot 10^{-984}$	16.000	0.0325
NM3B	3	$2.0108 \cdot 10^{-4}$	$1.0927 \cdot 10^{-53}$	$6.3700 \cdot 10^{-842}$	16.000	0.0331

Table 4: *Numerical results for* $f_2(x)$

method	it	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	CPU
MKT	3	$7.4786 \cdot 10^{-8}$	$5.8399 \cdot 10^{-132}$	$1.1162 \cdot 10^{-2117}$	16.000	0.0406
MSGG	3	$2.7003 \cdot 10^{-8}$	$2.8804 \cdot 10^{-139}$	$8.0888 \cdot 10^{-2235}$	16.000	0.0438
MSSSL	3	$4.9791 \cdot 10^{-6}$	$1.0851 \cdot 10^{-101}$	$2.8105 \cdot 10^{-1632}$	16.000	0.0462
MMBM	3	$1.7448 \cdot 10^{-7}$	$8.0141 \cdot 10^{-126}$	$3.1452 \cdot 10^{-2019}$	16.000	0.0463
MBAMM	3	$2.5006 \cdot 10^{-9}$	$2.8656 \cdot 10^{-156}$	$2.5341 \cdot 10^{-2507}$	16.000	0.0456
MGKN	3	$3.4023 \cdot 10^{-10}$	$4.1685 \cdot 10^{-170}$	$1.0747 \cdot 10^{-2728}$	16.000	0.0394
MSB	3	$5.6297 \cdot 10^{-9}$	$3.5586 \cdot 10^{-150}$	$2.3115 \cdot 10^{-2409}$	16.000	0.0406
MTM	3	$3.9566 \cdot 10^{-9}$	$8.3176 \cdot 10^{-153}$	$1.2099 \cdot 10^{-2451}$	16.000	0.0550
NM1A	3	$2.7734 \cdot 10^{-10}$	$6.1675 \cdot 10^{-172}$	$2.2063 \cdot 10^{-2758}$	16.000	0.035
NM2A	3	$7.4584 \cdot 10^{-10}$	$5.6036 \cdot 10^{-165}$	$5.7763 \cdot 10^{-2647}$	16.000	0.0362
NM3A	3	$1.5157 \cdot 10^{-9}$	$5.6559 \cdot 10^{-160}$	$7.9948 \cdot 10^{-2567}$	16.000	0.0369
NM1B	3	$7.5911 \cdot 10^{-10}$	$7.5856 \cdot 10^{-165}$	$7.4985 \cdot 10^{-2645}$	16.000	0.0375
NM2B	3	$3.2961 \cdot 10^{-9}$	$1.5000 \cdot 10^{-154}$	$5.0770 \cdot 10^{-2480}$	16.000	0.0375
NM3B	3	$1.1336 \cdot 10^{-8}$	$6.9956 \cdot 10^{-146}$	$3.0927 \cdot 10^{-2341}$	16.000	0.0356

Table 5: *Numerical results for* $f_3(x)$

method	it	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	CPU
MKT	3	$5.8981 \cdot 10^{-12}$	$2.2937 \cdot 10^{-194}$	$6.2783 \cdot 10^{-3113}$	16.000	0.0344
MSGG	3	$6.9171 \cdot 10^{-13}$	$4.8597 \cdot 10^{-210}$	$1.7125 \cdot 10^{-3364}$	16.000	0.0375
MSSSL	3	$2.1173 \cdot 10^{-10}$	$1.4873 \cdot 10^{-169}$	$5.2291 \cdot 10^{-2716}$	16.000	0.0388
MMBM	3	$8.1818 \cdot 10^{-12}$	$5.1860 \cdot 10^{-192}$	$3.5210 \cdot 10^{-3075}$	16.000	0.0356
MBAMM	3	$1.6411 \cdot 10^{-13}$	$1.3589 \cdot 10^{-220}$	$6.6380 \cdot 10^{-3534}$	16.000	0.0388
MGKN	3	$3.0172 \cdot 10^{-14}$	$6.5425 \cdot 10^{-236}$	$1.5631 \cdot 10^{-3782}$	16.000	0.0325
MSB	3	$6.9534 \cdot 10^{-14}$	$2.6903 \cdot 10^{-226}$	$6.7836 \cdot 10^{-3625}$	16.000	0.0344
MTM	3	$3.4214 \cdot 10^{-14}$	$7.8919 \cdot 10^{-232}$	$5.0688 \cdot 10^{-3714}$	16.000	0.0475
NM1A	3	$3.3634 \cdot 10^{-16}$	$5.5495 \cdot 10^{-267}$	$1.6741 \cdot 10^{-4279}$	16.000	0.0287
NM2A	3	$1.3202 \cdot 10^{-14}$	$8.4901 \cdot 10^{-240}$	$7.2663 \cdot 10^{-3843}$	16.000	0.0294
NM3A	3	$5.2518 \cdot 10^{-14}$	$1.2980 \cdot 10^{-229}$	$2.5167 \cdot 10^{-3679}$	16.000	0.0312
NM1B	3	$1.3169 \cdot 10^{-14}$	$2.5432 \cdot 10^{-239}$	$9.5228 \cdot 10^{-3835}$	16.000	0.0275
NM2B	3	$2.5289 \cdot 10^{-14}$	$9.1704 \cdot 10^{-235}$	$8.1986 \cdot 10^{-3762}$	16.000	0.0294
NM3B	3	$1.0830 \cdot 10^{-12}$	$8.6488 \cdot 10^{-208}$	$2.3670 \cdot 10^{-3329}$	16.000	0.03

Table 6: *Numerical results for* $f_4(x)$

method	it	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	CPU
MKT	3	$2.9007 \cdot 10^{-32}$	$6.4779 \cdot 10^{-505}$	$2.4802 \cdot 10^{-8067}$	16.000	0.111
MSGG	3	$6.5287 \cdot 10^{-32}$	$7.2926 \cdot 10^{-499}$	$4.2832 \cdot 10^{-7970}$	16.000	0.114
MSSSL	3	$9.3684 \cdot 10^{-30}$	$8.6553 \cdot 10^{-462}$	$2.4384 \cdot 10^{-7374}$	16.000	0.114
MMBM	3	$5.9620 \cdot 10^{-31}$	$1.5491 \cdot 10^{-482}$	$6.6811 \cdot 10^{-7708}$	16.000	0.118
MBAMM	3	$5.1813 \cdot 10^{-34}$	$1.2984 \cdot 10^{-534}$	$3.1386 \cdot 10^{-8544}$	16.000	0.118
MGKN	3	$1.2622 \cdot 10^{-33}$	$4.5689 \cdot 10^{-528}$	$3.9696 \cdot 10^{-8439}$	16.000	0.108
MSB	3	$5.2367 \cdot 10^{-33}$	$1.4395 \cdot 10^{-517}$	$1.5301 \cdot 10^{-8270}$	16.000	0.110
MTM	3	$6.4722 \cdot 10^{-34}$	$5.1403 \cdot 10^{-533}$	$1.2883 \cdot 10^{-8518}$	16.000	0.126
NM1A	3	$1.5714 \cdot 10^{-33}$	$1.6613 \cdot 10^{-526}$	$4.0466 \cdot 10^{-8414}$	16.000	0.109
NM2A	3	$3.8002 \cdot 10^{-31}$	$7.3983 \cdot 10^{-486}$	$3.1508 \cdot 10^{-7761}$	16.000	0.104
NM3A	3	$2.2830 \cdot 10^{-30}$	$1.3482 \cdot 10^{-472}$	$2.9502 \cdot 10^{-7548}$	16.000	0.108
NM1B	3	$1.3844 \cdot 10^{-33}$	$1.8892 \cdot 10^{-527}$	$2.7335 \cdot 10^{-8429}$	16.000	0.104
NM2B	3	$5.3049 \cdot 10^{-31}$	$2.1212 \cdot 10^{-483}$	$9.0552 \cdot 10^{-7722}$	16.000	0.105
NM3B	3	$4.6200 \cdot 10^{-30}$	$2.1639 \cdot 10^{-467}$	$1.1603 \cdot 10^{-7464}$	16.000	0.107

Table 7: *Numerical results for* $f_5(x)$

method	it	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	CPU
MKT	3	$2.1129\cdot10^{-25}$	$1.5197\cdot10^{-391}$	$7.8004\cdot10^{-6250}$	16.000	0.0575
MSGG	3	$2.3148\cdot10^{-26}$	$6.7216\cdot10^{-408}$	$1.7167\cdot10^{-6512}$	16.000	0.0606
MSSSL	3	$8.2594\cdot10^{-27}$	$2.4290\cdot10^{-415}$	$7.6065\cdot10^{-6632}$	16.000	0.0619
MMBM	3	$2.7484\cdot10^{-25}$	$1.3390\cdot10^{-389}$	$1.3490\cdot10^{-6218}$	16.000	0.0644
MBAMM	3	$4.1228\cdot10^{-27}$	$1.2421\cdot10^{-420}$	$5.7231\cdot10^{-6717}$	16.000	0.0662
MGKN	3	$1.2434\cdot10^{-28}$	$3.6925\cdot10^{-446}$	$1.3517\cdot10^{-7126}$	16.000	0.0544
MSB	3	$4.0450\cdot10^{-27}$	$7.9911\cdot10^{-421}$	$4.3019\cdot10^{-6720}$	16.000	0.0575
MTM	3	$1.5451\cdot10^{-27}$	$6.6975\cdot10^{-428}$	$1.0407\cdot10^{-6833}$	16.000	0.0694
NM1A	3	$7.9516\cdot10^{-28}$	$8.3706\cdot10^{-433}$	$1.9038\cdot10^{-6912}$	16.000	0.0538
NM2A	3	$3.1879\cdot10^{-26}$	$1.5174\cdot10^{-405}$	$1.0529\cdot10^{-6474}$	16.000	0.0538
NM3A	3	$1.5100\cdot10^{-25}$	$4.8583\cdot10^{-394}$	$6.4056\cdot10^{-6290}$	16.000	0.0518
NM1B	3	$8.1132\cdot10^{-29}$	$6.2355\cdot10^{-450}$	$9.2398\cdot10^{-7188}$	16.000	0.0538
NM2B	3	$2.2842\cdot10^{-26}$	$6.2176\cdot10^{-408}$	$5.6483\cdot10^{-6513}$	16.000	0.0532
NM3B	3	$2.3504\cdot10^{-25}$	$1.0464\cdot10^{-390}$	$2.4901\cdot10^{-6236}$	16.000	0.0537

Table 8: *Numerical results for* $f_6(x)$

method	it	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	CPU
MKT	3	$1.7675\cdot10^{-11}$	$5.8749\cdot10^{-166}$	$1.3039\cdot10^{-2637}$	16.000	0.0581
MSGG	3	$5.3823\cdot10^{-13}$	$6.7571\cdot10^{-192}$	$2.5728\cdot10^{-3054}$	16.000	0.0594
MSSSL	3	$1.3817\cdot10^{-10}$	$5.9340\cdot10^{-153}$	$7.9534\cdot10^{-2431}$	16.000	0.065
MMBM	3	$8.2567\cdot10^{-12}$	$1.8483\cdot10^{-171}$	$7.3510\cdot10^{-2726}$	16.000	0.0613
MBAMM	3	$2.6453\cdot10^{-13}$	$2.4795\cdot10^{-197}$	$8.8006\cdot10^{-3142}$	16.000	0.0625
MGKN	3	$9.0931\cdot10^{-14}$	$4.9413\cdot10^{-205}$	$2.8565\cdot10^{-3265}$	16.000	0.0531
MSB	3	$4.7536\cdot10^{-13}$	$1.0734\cdot10^{-193}$	$4.9060\cdot10^{-3084}$	16.000	0.0569
MTM	3	$1.3916\cdot10^{-13}$	$3.9851\cdot10^{-202}$	$8.1489\cdot10^{-3219}$	16.000	0.0700
NM1A	3	$2.5080\cdot10^{-14}$	$8.5415\cdot10^{-215}$	$2.7982\cdot10^{-3422}$	16.000	0.0519
NM2A	3	$1.6031\cdot10^{-13}$	$2.1458\cdot10^{-200}$	$2.2768\cdot10^{-3190}$	16.000	0.0513
NM3A	3	$1.3220\cdot10^{-13}$	$4.7015\cdot10^{-201}$	$3.0797\cdot10^{-3200}$	16.000	0.0537
NM1B	3	$1.7548\cdot10^{-13}$	$1.9906\cdot10^{-200}$	$1.4957\cdot10^{-3191}$	16.000	0.0507
NM2B	3	$4.7330\cdot10^{-12}$	$1.5343\cdot10^{-175}$	$2.2821\cdot10^{-2791}$	16.000	0.0531
NM3B	3	$1.9276\cdot10^{-11}$	$7.0415\cdot10^{-165}$	$7.0783\cdot10^{-2620}$	16.000	0.0525

Table 9: *Numerical results for* $f_7(x)$

Numerical results are listed in tables 3 - 9, where "it" represents the number of iterations required for each method to satisfy the stopping criterion $|f(x_n)| < 10^{-500}$ (except for $f_5(x)$, where it is $|f(x_n)| < 10^{-1000}$). The following three columns display the errors of the first, second and third iteration. Cases when the method diverges or converges to an undesired root are denoted by "–". The computational order of convergence "COC" [38] has been calculated by the formula:

$$\mathrm{COC} = \frac{\log|(x_n-\alpha)/(x_{n-1}-\alpha)|}{\log|(x_{n-1}-\alpha)/(x_{n-2}-\alpha)|}.$$

Finally, the last column CPU shows the average computational time of 25 performances of each method.

From these tables, it is clear that all the methods of family (1) reach the 16th convergence order, which agrees with the theoretical conclusions derived in Section 2. CPU times of the new methods are mostly better than the CPU times of the other considered methods. Moreover, according to the error values $|x_n - \alpha|$, a new method NM1A performs favorably in comparison to the majority of existing methods for the given particular choice of test functions.

3.2. Dynamical behaviour

In the following section, we compare the above given iterative methods in the complex plane by using basins of attractions. The description of the dynamical behaviour and comparison of the method through basins of attraction have been previously used in [6, 23, 30, 32, 34]. Before presenting the numerical results, we give a brief review of the basic concepts regarding basins of attraction.

Let $f : \mathbb{C} \to \mathbb{C}$ be the rational map of the complex plain. Point x_0 is called a fixed point for f if $f(x_0) = x_0$. Fixed point x_0 is called attracting if $|f'(x_0)| < 1$, repelling if $|f'(x_0)| > 1$, and neutral if $|f'(x_0)| = 1$. Qualitative behaviors of non-fixed starting points can be interpreted in relation to the fixed points. Further, the orbit for $x \in \mathbb{C}$ is defined as the set $orb(x) = \{x, f(x), f^2(x), ...\}$ and point y_0 is named periodic with the minimal period n if $f^n(y_0) = y_0$. It is evident that a fixed point is a periodic point with its minimal period being 1.

Each attracting region is called the basin of attraction $A(\alpha)$:

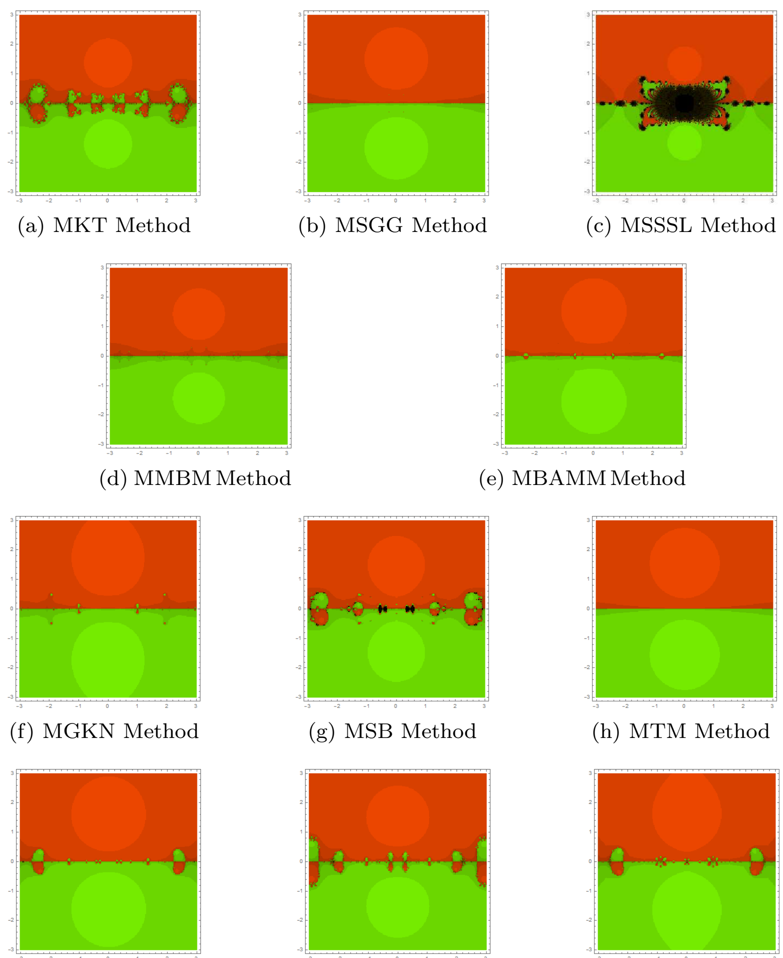
$$A(\alpha) = \{x_0 \in \mathbb{C} : f^m(x_0) \to \alpha, m \to \infty\},$$

where α is an attracting fixed point of function f. In other words, the basin of attraction is the set of starting points whose trajectories are asymptotic to a bounded region. The points whose orbits tend to an attracting fixed point α define a set named the Fatou set. The closure of the set consisting of repelling periodic points is denoted as the Julia set. The Julia set is the complement to the Fatou set, and it establishes the borders between the basins of attraction. This implies that the basin of attraction of any fixed point belongs to the Fatou set and the boundaries of these basins of attraction belong to the Julia set.

We observe a 256×256 mesh of a rectangle $R = [-3, 3] \times [-3, 3]$ with uniformly distributed complex starting points (without pure real and pure imaginary starting points). When considering the sensitive dependence on starting conditions, one needs to observe the "decorations" along the basin boundaries for each method's geometry in terms of frequency, size, and structure. Methods with rather clean boundaries are considered more desirable since they show increased behavior predictability in the sense that the observed starting point converges to the closest solution. In order to visualize the dynamical behaviour, we assign a color to each starting point $x_0 \in R$ according to the root at which the corresponding iterative method starting from x_0 converges, and we mark the point as black if the method does not converge, in the sense that after at most 100 iterations it has a distance larger than 10^{-5} to any of the roots. Furthermore, the number of iterations necessary to converge to a root is shown through a variety of color intensities. Points requiring fewer iterations appear with lower intensity. We have chosen three members of the family (1) for a dynamical comparison with other methods presented above, namely NM1A, NM2A and NM1B.

The following test examples have been employed to analyze the dynamical behavior:

- $p_1(z) = z^2 + 1$ with roots $\pm i$, [29];
- $p_2(z) = z^5 + z$ with roots $0, \pm 0.70710678 \pm 0.70710678i$, [32];
- $p_3(z) = (e^{z+1} - 1)(z - 1)$ with roots ± 1, [7];



(a) MKT Method (b) MSGG Method (c) MSSSL Method

(d) MMBM Method (e) MBAMM Method

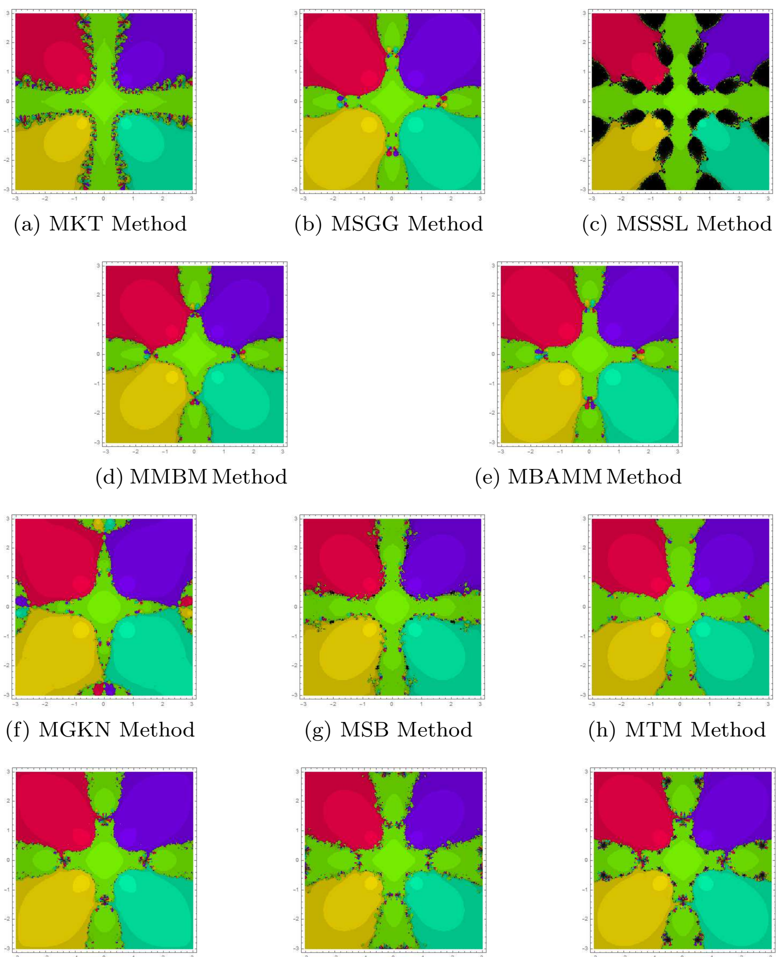
(f) MGKN Method (g) MSB Method (h) MTM Method

(i) NM1A Method (j) NM2A Method (k) NM1B Method

Figure 1: *Basins of attraction of different methods for polynomial p_1*

		$p_1(z)$			$p_2(z)$	
method	black(%)	average (it)	CPU	black(%)	average (it)	CPU
MKT	0	2.1559	662.829	0	2.9437	671.549
MSGG	0	1.8477	667.829	0	2.2695	681.313
MSSSL	1.605	5.7024	761.235	12.341	17.1359	1052.218
MMBM	0	1.9606	681.250	0	2.5404	699.953
MBAMM	0	1.8256	679.750	0	2.2971	694.891
MGKN	0	1.7726	663.203	0	2.3175	677.765
MSB	0	2.3825	682.172	0.452	3.0569	702.984
MTM	0	1.7867	685.516	0	2.4971	713.750
NM1A	0	1.8030	650.906	0	2.3083	656.266
NM2A	0	1.9623	653.594	0	2.6659	659.625
NM1B	0	1.8416	652.359	0	2.4894	657.454

Table 10a: *Numerical results for $p_1(z)$ and $p_2(z)$*



(a) MKT Method (b) MSGG Method (c) MSSSL Method

(d) MMBM Method (e) MBAMM Method

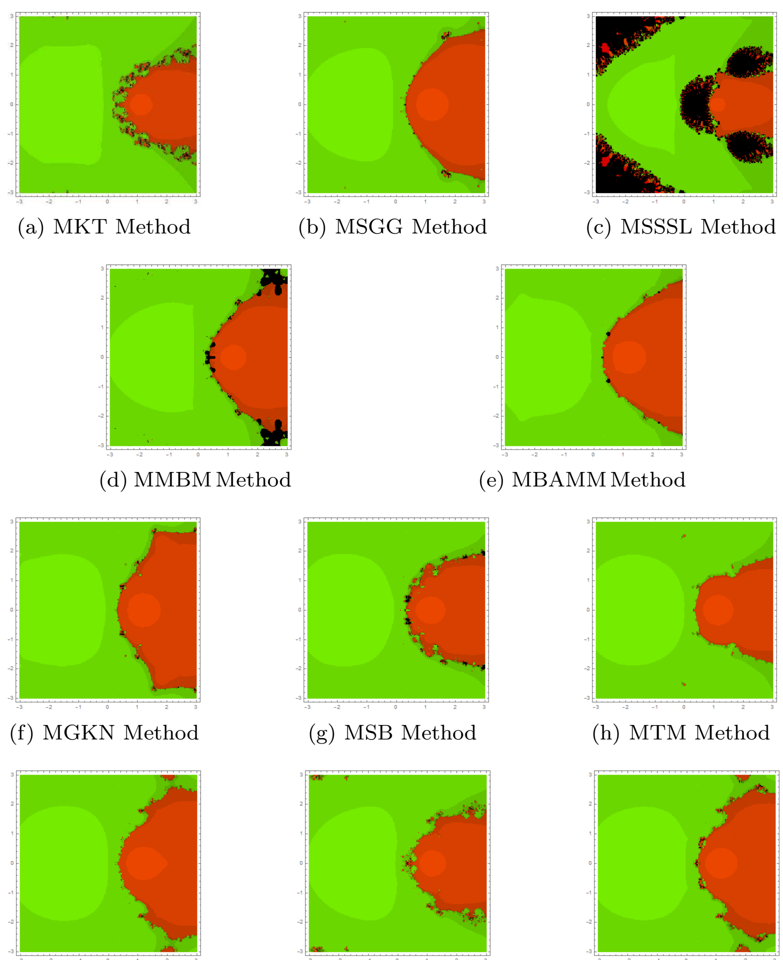
(f) MGKN Method (g) MSB Method (h) MTM Method

(i) NM1A Method (j) NM2A Method (k) NM1B Method

Figure 2: *Basins of attraction of different methods for polynomial p_2*

	$p_3(z)$			Total	
method	black(%)	average (it)	CPU	average (it)	average (CPU)
MKT	0.198	2.1505	672.844	2.4167	669.089
MSGG	0.098	1.9157	680.171	2.0110	676.438
MSSSL	20.932	22.9360	745.719	15.2581	853.057
MMBM	3.174	4.9801	690.079	3.1604	690.427
MBAMM	0.154	1.9314	691.687	2.0180	688.776
MGKN	0.046	1.8567	677.063	1.9823	672.677
MSB	0.380	2.2142	692.750	2.5512	692.635
MTM	0	1.7861	698.750	2.0233	699.339
NM1A	0	1.8143	663.219	1.9752	656.797
NM2A	0.031	1.9111	666.015	2.1798	659.745
NM1B	0.095	1.9570	665.078	2.0960	658.297

Table 10b: *Numerical results for $p_3(z)$ and total average*



(a) MKT Method (b) MSGG Method (c) MSSSL Method

(d) MMBM Method (e) MBAMM Method

(f) MGKN Method (g) MSB Method (h) MTM Method

(i) NM1A Method (j) NM2A Method (k) NM1B Method

Figure 3: *Basins of attraction of different methods for polynomial p_3*

Tables 10a and 10b show a quantitative comparison of the methods. The percent of black points (out of 65536 equally distributed starting points on the rectangle $[-3,3]\times[-3,3]$), the average number of iterations per starting point and CPU time (in seconds) required for the depiction of the graph is calculated for each method and each test example. In order to summarize the presented results, the last two columns in Table 10b display the average number of iterations and the average CPU time determined across all three test examples.

According to those results, it can be said that NM1A, NM2A and NM1B methods are very competitive with already existing methods, especially in the sense of the CPU time, the new methods are faster than the others in all tests. In terms of the average number of iterations, the best method overall is NM1A, closely followed by MGKN, MSGG, MBAMM, MTM and NM1B. Note that NM1A and MTM are the only methods with no black points in all test examples. Aside from that, NM1A has the best CPU time results.

4. Conclusions

In this paper, we have given a simple yet efficient family of multipoint methods of order sixteen with four steps, by using an optional fourth order and an optional eighth order iteration scheme for solving nonlinear equations. One requires four evaluations of the function and one of its first derivative per step, accordingly, the family is of the 16th convergence order. Some examples of members of the family are given and their performance is compared with the existing optimal sixteenth order methods over numerical experiments. The presented methods show competitive results in the comparison to the existing methods by numerical results displayed in Table 3 - Table 9. Moreover, the presented basins of attraction have also confirmed good performance of the methods as compared to other methods established in the literature.

References

[1] R. Behl, S. Amat, A. A. Magreñán, S.S. Motsa, *An efficient optimal family of sixteenth order methods for nonlinear models*, J. Comput. Appl. Math. **354**(2019), 271–285.

[2] W. Bi, H. Ren, Q. Wu, *Three-step iterative methods with eighth-order convergence for solving nonlinear equations*, J. Comput. Appl. Math. **225**(2009), 105–112.

[3] W. Bi, Q. Wu, H. Ren, *A new family of eighth-order iterative methods for solving nonlinear equations*, Appl. Math. Comput. **214**(2009), 236–245.

[4] C. Chun, M. Y. Lee, B. Neta, J. Džunić, *On optimal fourth-order iterative methods free from second derivative and their dynamics*, Appl. Math. Comput. **218**(2012), 6427–6438.

[5] C. Chun, B. Neta, *An analysis of a new family of eighth-order optimal methods*, Appl. Math. Comput. **245**(2014), 86–107.

[6] C. Chun, B. Neta, *Basins of attraction for Zhou–Chen–Song fourth order family of methods for multiple roots*, Math. Comput. Simul. **109**(2015), 74–91.

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

[8] A. Cordero, J. R. Torregrosa, M. Vassileva, *A family of modified Ostrowski's methods with optimal eighth order of convergence*, Appl. Math. Lett. **24**(2011), 2082–2086.

[9] J. Džunić, M. Petković, Lj. Petković, *A family of optimal three-point methods for solving nonlinear equations using two parametric functions*, Appl. Math. Comput. **217**(2011), 7612–7619.

[10] Y. H. Geum, Y. I. Kim, *A multi-parameter family of three-step eighth-order iterative methods locating a simple root*, Appl. Math. Comput. **215**(2010), 3375–3382.

[11] Y. H. Geum, Y. I. Kim, *A biparametric family of four-step sixteenth-order root-finding methods with the optimal efficiency index*, Appl. Math. Lett. **24**(2011), 1336-1342.

[12] Y. H. Geum, Y. I. Kim, B. Neta, *Developing an Optimal Class of Generic Sixteenth-Order Simple-Root Finders and Investigating Their Dynamics*, Mathematics **7**(2019), 8.

[13] M. Grau-Sanchez, A. Grau, M. Noguera, *Ostrowski type methods for solving systems of nonlinear equations*, Appl. Math. Comput. **218**(2011), 2377-2385.

[14] D. Jain, *Families of Newton-like methods with fourth-order convergence*, Int. J. Comput. Math. **90**(2013), 1072–1082.

[15] P. Jarratt, *Some fourth order multipoint methods for solving equations*, Math. Comput. **20**(1966), 434-437.

[16] P. Jarratt, *Some efficient fourth-order multipoint methods for solving equations*, BIT. **9**(1969), 119-124.

[17] Y. Khan, M. Fardi, K. Sayevand, *A new general eighth-order family of iterative methods for solving nonlinear equations*, Appl. Math. Lett. **25**(2012), 2262–2266.

[18] R. King, *A family of fourth order methods for nonlinear equations*, SIAM J. Num. Anal. **10**(1973), 876–879.

[19] H. T. Kung, J. F. Traub, *Optimal order of one-point and multi-point iteration*, Appl. Math. Comput. **21**(1974), 643-651.

[20] P. Maroju, R. Behl, S. S. Motsa, *Some novel and optimal families of King's method with eighth and sixteenth-order of convergence*, J. Comput. Appl. Math. **318**(2017), 136–148.

[21] A. K. Mashewari, *A fourth-order iterative method for solving nonlinear equations*, Appl. Math. Comput. **211**(2009), 383-391.

[22] G. Matthies, M. Salimi, S. Sharifi, J. L. Varona, *An optimal class of eighth-order iterative methods based on Kung and Traub's method with its dynamics*, Japan J. Indust. Appl. Math. **33**(2016), 751–766.

[23] B. Neta, *On a family of multipoint methods for non-linear equations*, Int. J. Comput. Math. **9**(1981), 353–361.

[24] A. M. Ostrowski, *Solutions of Equations and Systems of Equations*, Academic Press, New York-London, 1960.

[25] M. S. Petković, B. Neta, Lj. Petković, J. Džunić, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier, Amsterdam, 2013.

[26] M. S. Petković, B. Neta, Lj. Petković, J. Džunić, *Multipoint methods for solving nonlinear equations: A survey*, Appl. Math. Comput. **226**(2014), 635–660.

[27] N. Ralević, D. Ćebić, *A new optimal family of three-step methods for efficient finding of a simple root of a nonlinear equation*, Math. Commun. **21**(2016), 189–197.

[28] M. Salimi, R. Behl, *Sixteenth-Order Optimal Iterative Scheme Based on Inverse Interpolatory Rational Function for Nonlinear Equations*, Symmetry **11**(2019), 691.

[29] S. Sharifi, M. Ferrara, M. Salimi, S. Siegmund, *New modification of Maheshwari's method with optimal eighth order convergence for solving nonlinear equations*, Open Math. J. **14**(2016), 443–451.

[30] S. Sharifi, M. Salimi, S. Siegmund, T. Lofti, *A new class of optimal 4 point methods with convergence 16 for solving nonlinear equations*, Math. Comput. Simul. **119**(2016), 69–90.

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

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

[33] J. R. Sharma, R. Guha, P. Gupta, *Improved King's methods with optimal order of convergence based on rational approximations*, Appl. Math. Lett. **26**(2013), 473–480.

[34] B. D. Stewart, *Attractor basins of various root-finding methods*, Tech. rep., DTIC Document, 2001.

[35] Y. Tao, K. Madhu, *Optimal Fourth, Eighth and Sixteenth Order Methods by Using Divided Difference Techniques and Their Basins of Attraction and Its Application*, Mathematics **7**(2019), 322.

[36] R. Thukral, *A new eighth-order iterative method for solving nonlinear equations*, Appl. Math. Comput. **217**(2010), 222–229.

[37] M. Z. Ullah, A. S. Al-Fhaid, F. Ahmad, *Four-point optimal sixteenth-order iterative method for solving nonlinear equations*, J. Appl. Math. **2013**(2013), Article ID 850365.

[38] S. Weerakoon, T. G. I. Fernando, *A variant of Newtons method with accelerated third-order convergence*, Appl. Math. Lett. **13**(2000), 87–93.