Journal of Applied Analysis and Computation
Volume 15, Number 6, December 2025, 3649–3676
Website:http://www.jaac-online.com
DOI:10.11948/20250049

OPTIMAL FOURTH-ORDER ITERATIVE METHODS FOR SOLVING NONLINEAR EQUATIONS: AN INNOVATIVE GENERAL CLASS WITH STABLE MEMBERS AND ENGINEERING APPLICATIONS

Niveen Abu Ghalioun[1] and Ali Zein[1,†]

Abstract In this work, we construct a new class of two-step fourth-order iterative methods for solving nonlinear equations. Each iteration requires two function evaluations and one evaluation of the first derivative. Consequently, this family is optimal according to the Kung-Traub conjecture. The first step of the family coincides with the classical Newton's method, while the second step involves three parameters and a weight function, offering a wide range of options and including several well-known methods as special cases. Additionally, we identify three new particular cases that perform well compared to existing methods within the same family. The analysis of complex dynamics and basins of attraction shows that these methods have a wider range of initial points that ensure convergence. Furthermore, numerical examples using various test functions and real-life applications illustrate that, in general, the new methods produce good results in terms of accuracy.

Keywords Iterative methods, optimal methods, basins of attraction, engineering applications.

MSC(2010) 41A25, 65H05, 65K05.

1. Introduction

Developing efficient and high-order iterative methods for solving nonlinear equations is a fundamental challenge in numerical analysis. This is because these equations often cannot be solved analytically, and many problems in engineering and applied sciences are modeled by nonlinear equations.

Iterative methods fall into two main categories: One-point and multi-point schemes. One-point iterative methods can achieve high orders of convergence by using higher derivatives of the function, often resulting in a significant computational cost. Multi-point methods, on the other hand, improve both convergence order and computational efficiency by leveraging previously computed information, see Petković et al. [26]. In recent decades, a variety of multi-step iterative methods have been introduced to achieve higher orders of convergence, see e.g. [5, 7, 8, 14, 15, 17, 19, 25].

†The corresponding author.
1Department of Applied Mathematics, Palestine Polytechnic University, P.O. Box 198, Dahiat albaladia, Hebron, Palestine
Email: 216370@ppu.edu.ps(N. Abu Ghalioun), alizein@ppu.edu(A. Zein)

The efficiency indicator of an iterative method is measured by *the efficiency index*, defined as $E.I. = p^{\frac{1}{d}}$, where p is the order of convergence and d is the total number of functional evaluations per iteration. In 1974, Kung and Traub [18] proposed what is now known as *the Kung-Traub conjecture*: A multi-point iterative method without memory for finding simple roots of a function can achieve a maximum convergence order of 2^{d-1}. Methods achieving this bound are called *optimal methods*. Naturally, they are of particular interest. The well-known Newton's method is considered an optimal method, converging quadratically to simple roots.

In the literature, considerable attention has been devoted to developing optimal fourth-order iterative methods [2, 4–8, 12, 13, 15–17, 19, 20, 23–25, 27–29, 31]. Some of these methods adopt an initial step similar to Jarratt's step [15, 23, 25, 28, 31], while others employ Newton's method as their initial step [2,7,8,16,17,19,20,24,27]. Among these optimal methods, some use two evaluations of the function and one evaluation of the first derivative per iteration [2,5–8,17,19,20], while others use one evaluation of the function and two evaluations of the first derivative [15,23,25,28,31].

In this paper, we introduce a new class of two-step optimal fourth-order iterative methods for solving nonlinear equations. These methods utilize Newton's method in the first step and involve a weight function in the second step. Each iteration requires two evaluations of the function and one evaluation of the first derivative. The general formulation of the second step allows for a diverse range of options. We show that the proposed family encompasses several well-known methods as special cases. Additionally, three distinct methods have been derived from this proposed family, which generally achieve higher accuracy than existing methods from the same family, based on the test cases considered, which include various nonlinear functions and engineering applications. Furthermore, through the analysis of basins of attraction, we demonstrate that these new methods exhibit better stability, that is, they possess wider sets of initial points leading to convergence.

The paper is structured as follows: Section 2 introduces the construction and convergence analysis of a new family of optimal fourth-order methods. In Section 3, some well-known schemes are listed as particular cases of the proposed family, and new specific methods within the proposed family are established. Section 4 is devoted to study the stability of particular methods by using the basins of attraction technique. Finally, in Section 5, numerical examples are presented to illustrate the performance of these new methods. Additionally, real-life applications are discussed.

2. Design and convergence analysis of the new family

The new family of fourth-order iterative methods consists of two steps. The first step is the well-known Newton's method, and the second step involves a weight function, as follows:

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \theta \dfrac{f(x_n)}{f'(x_n)} - G(\eta_n)\dfrac{Af(x_n) + Bf(y_n)}{f'(x_n)}, \end{cases} \tag{2.1}$$

where θ, A and B are parameters, and G is a weight function in terms of

$$\eta_n = \frac{f(y_n)}{f(x_n)}. \tag{2.2}$$

The parameters θ, A and B are chosen arbitrarily. The weight function G is then designed to ensure fourth-order convergence, as demonstrated in the following theorem.

Theorem 2.1. *Let $\alpha \in I$ be a simple root of a sufficiently differentiable function $f : I \subseteq \mathbf{R} \to \mathbf{R}$ for an open interval I. If x_0 is sufficiently close to α, and the weight functions $G(\eta)$ satisfies*

$$G(0) = \frac{1-\theta}{A}, \quad G'(0) = \frac{B\theta + A - B}{A^2}, \quad G''(0) = \frac{2(-B^2\theta + 2A^2 - AB + B^2)}{A^3}, \tag{2.3}$$

then the scheme (2.1) *converges to α with order of convergence four and satisfies the error equation*

$$e_{n+1} = -\left[\frac{\left(A^4G'''(0) - 30A^3 + 12A^2B - 6AB^2 - 6B^3(\theta-1)\right)c_2^3}{6A^3} + c_2c_3\right]e_n^4 + O(e_n^5),$$

where $e_n = x_n - \alpha$ and $c_j = \dfrac{f^{(j)}(\alpha)}{j!\,f'(\alpha)}$, $j = 2, 3, \ldots$, provided $A \neq 0$.

Proof. Applying Taylor's expansion to $f(x_n)$ and $f'(x_n)$ around α, we obtain

$$f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4] + O(e_n^5), \tag{2.4}$$

and

$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4] + O(e_n^5). \tag{2.5}$$

Then from (2.4) and (2.5), we get

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2e_n^2 + (2c_2^2 - 2c_3)e_n^3 + (-4c_2^3 + 7c_2c_3 - 3c_4)e_n^4 + O(e_n^5). \tag{2.6}$$

Subtracting α from both sides of the first equation in (2.1) and using (2.6), we obtain

$$y_n - \alpha = c_2e_n^2 - (2c_2^2 - 2c_3)e_n^3 + (4c_2^3 - 7c_2c_3 + 3c_4)e_n^4 + O(e_n^5). \tag{2.7}$$

Using the expansion of $f(y_n)$ about α and (2.7), we obtain

$$f(y_n) = f'(\alpha)[c_2e_n^2 + (-2c_2^2 + 2c_3)e_n^3 + (5c_2^3 - 7c_2c_3 + 3c_4)e_n^4] + O(e_n^5). \tag{2.8}$$

From (2.4), (2.5), and (2.8), we have

$$\begin{aligned}&\frac{Af(x_n) + Bf(y_n)}{f'(x_n)}\\ =&Ae_n - c_2(A-B)e_n^2 + ((2A-4B)c_2^2 - 2c_3(A-B))e_n^3\\ &+ ((-4A+13B)c_2^3 + 7c_3(A-2B)c_2 - 3c_4(A-B))e_n^4 + O(e_n^5).\end{aligned} \tag{2.9}$$

Using (2.4) and (2.8), the expansion of the weight function variable η in (2.2) is as follows

$$\eta_n = \frac{f(y_n)}{f(x_n)} = c_2e_n + (-3c_2^2 + 2c_3)e_n^2 + (8c_2^3 - 10c_2c_3 + 3c_4)e_n^3 + O(e_n^4).$$

Then, expanding the weight function $G(\eta_n)$ around zero results in

$$\begin{aligned}
G(\eta_n) &= G(0) + G'(0)\eta_n + G''(0)\frac{\eta_n^2}{2} + G'''(0)\frac{\eta_n^3}{6} + G^{(4)}(0)\frac{\eta_n^4}{24} + \cdots \\
&= G(0) + G'(0)c_2e_n + \left[\frac{(-6G'(0)+G''(0))c_2^2}{2} + 2c_3G'(0)\right]e_n^2 \\
&\quad + \left[\left(8G'(0) - 3G''(0) + \frac{G'''(0)}{6}\right)c_2^3 - \left(10G'(0) - 2G''(0)\right)c_2c_3 \right. \\
&\quad \left. + 3c_4G'(0)\right]e_n^3 + \cdots . \qquad (2.10)
\end{aligned}$$

Finally, according to (2.6), (2.9) and (2.10) the error equation of the scheme (2.1) is

$$\begin{aligned}
e_{n+1} &= x_n - \alpha - \theta\frac{f(x_n)}{f'(x_n)} - G(\eta_n)\frac{Af(x_n)+Bf(y_n)}{f'(x_n)} \\
&= \Big[1-\theta-AG(0)\Big]e_n + \Big[(A-B)G(0) - AG'(0) + \theta\Big]c_2e_n^2 \\
&\quad + \left[\left(A\Big(-4G(0) + 8G'(0) - G''(0)\Big) + 8BG(0) - 2BG'(0) - 4\theta\right)\frac{c_2^2}{2}\right. \\
&\quad \left. + 2\Big(A(G(0)-G'(0)) - BG(0) + \theta\Big)c_3\right]e_n^3 \\
&\quad + \left[\left(A\Big(24G(0) - 78G'(0) + 21G''(0) - G'''(0)\Big) - 78BG(0)\right.\right. \\
&\quad \left. + B\Big(42G'(0) - 3G''(0)\Big) + 24\theta\right)\frac{c_2^3}{6} \\
&\quad - \left(A\Big(7G(0) - 14G'(0) + 2G''(0)\Big) - 14BG(0) + 4BG'(0) + 7\theta\right)c_2c_3 \\
&\quad \left. + 3\Big(A(G(0)-G'(0)) - BG(0) + \theta\Big)c_4\right]e_n^4 + O(e_n^5). \qquad (2.11)
\end{aligned}$$

For fourth-order convergence, the coefficients of e_n, e_n^2, and e_n^3 in (2.11) must vanish, giving:

$$\begin{aligned}
&1-\theta-AG(0) = 0, \\
&(A-B)G(0) - AG'(0) + \theta = 0, \\
&\left(A\Big(-4G(0) + 8G'(0) - G''(0)\Big) + 8BG(0) - 2BG'(0) - 4\theta\right)\frac{c_2^2}{2} \\
&\quad + 2\Big(A(G(0)-G'(0)) - BG(0) + \theta\Big)c_3 = 0.
\end{aligned}$$

Upon solving this system of equations for $G(0), G'(0)$ and $G''(0)$, we obtain

$$\begin{aligned}
G(0) &= \frac{1-\theta}{A},\ G'(0) = \frac{B\theta + A - B}{A^2},\ \text{and} \\
G''(0) &= \frac{2(-B^2\theta + 2A^2 - AB + B^2)}{A^3}.
\end{aligned}$$

These conditions in (2.3), when substituted into (2.11), yield the following error equation

$$e_{n+1} = -\left[\frac{\left(A^4G'''(0) - 30A^3 + 12A^2B - 6AB^2 - 6B^3(\theta - 1)\right)c_2^3}{6A^3} + c_2c_3\right]e_n^4 + O(e_n^5),$$

and the proof is complete. □

3. Particular cases within the proposed family

Numerous specific fourth-order methods can be derived from the family (2.1) by adjusting the parameters θ, A, and B, as well as by selecting different weight functions. This section presents several well-known methods as particular instances of the proposed family (2.1), along with the introduction of three new schemes.

1. Suppose $\theta = 0$, $A = 1$, and $B = 0$. According to (2.3) the weight function should satisfy

   $$G(0) = 1,\ G'(0) = 1,\ G''(0) = 4.$$

   By Selecting the weight function $G(\eta) = \dfrac{\eta - 1}{2\eta - 1}$, then we obtain the well-known Traub-Ostrowski's method (TOM) [24,30], which is given by

   $$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\dfrac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)}. \end{cases} \tag{3.1}$$

   If we consider the weight function $G(\eta) = \dfrac{1 + \eta^2}{1 - \eta}$, we get the following method

   $$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{1}{f'(x_n)}\dfrac{f^2(x_n) + f^2(y_n)}{f(x_n) - f(y_n)}. \end{cases} \tag{3.2}$$

   This is the method proposed by Kou et al. (KLM) [17].
   Another choice for the weight function $G(\eta) = \eta^2 + \dfrac{1}{1 - \eta}$, leads to the following method

   $$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}\left[\dfrac{f(x_n)}{f(x_n) - f(y_n)} + \left(\dfrac{f(y_n)}{f(x_n)}\right)^2\right], \end{cases} \tag{3.3}$$

   which is the method of Maheshwari (MM) [19].

2. For $\theta = 1, A = 1$, and $B = 0$, the conditions in (2.3) hold if
$$G(0) = 0,\ G'(0) = 1,\ G''(0) = 4.$$
By taking the weight function $G(\eta) = \dfrac{\eta}{1 - 2\eta + \eta^2}$, the resulting scheme is given by
$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f'(x_n)} \left[\dfrac{f(x_n)}{f(x_n) - f(y_n)}\right]^2. \end{cases} \tag{3.4}$$
This is the method proposed by Chun (Chun1) [5].
3. Let $\theta = 1$, $A = 1$, and $B = 2$. The conditions in (2.3) are satisfied if
$$G(0) = 0,\ G'(0) = 1,\ G''(0) = 0.$$
Assuming $G(\eta) = \eta$, we get the scheme of Chun (Chun2) [6, 8], which is expressed as
$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f'(x_n)} \dfrac{f(x_n) + 2f(y_n)}{f(x_n)}. \end{cases} \tag{3.5}$$
4. If $\theta = 1$, $A = 2$, and $B = -1$, the conditions in (2.3) hold as follows
$$G(0) = 0,\ G'(0) = \frac{1}{2},\ G''(0) = \frac{5}{2}.$$
Using the weight function $G(\eta) = \dfrac{\eta}{2 - 5\eta}$, results in the following method
$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f'(x_n)} \dfrac{2f(x_n) - f(y_n)}{2f(x_n) - 5f(y_n)}. \end{cases} \tag{3.6}$$
This is the method introduced by Chun and Ham (CHM) [7].
5. Suppose $\theta = 0$, $A = 1$, and $B = 1$, according to (2.3)
$$G(0) = 1,\ G'(0) = 0,\ G''(0) = 4.$$
Assuming $G(\eta) = 1 + 2\eta^2$, we obtain the scheme developed by Chand et al. (PBM) [2], which is given by
$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{f(y_n) + f(x_n)}{f'(x_n)} \left[1 + 2\left(\dfrac{f(y_n)}{f(x_n)}\right)^2\right]. \end{cases} \tag{3.7}$$

Now, by selecting different values for θ, A, and B, we derive three new specific methods within the proposed family (2.1) as detailed below:

i. Assuming $\theta = 1$, $A = 1$, and $B = 0$, then the weight function $G(\eta)$ satisfies the conditions in (2.3) when

$$G(0) = 0,\ G'(0) = 1,\ G''(0) = 4.$$

Selecting the weight function $G(\eta) = \dfrac{\eta}{1 - 2\eta - \eta^2}$, then we obtain the following scheme (NA1)

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f'(x_n)} \dfrac{f(x_n)f(y_n)}{f^2(x_n) - f^2(y_n) - 2f(x_n)f(y_n)}. \end{cases} \tag{3.8}$$

ii. Let $\theta = 1$, $A = -1$, and $B = -2$, the conditions in (2.3) are satisfied when

$$G(0) = 0,\ G'(0) = -1,\ G''(0) = 0.$$

Choosing the weight function $G(\eta) = \dfrac{\eta}{6\eta^2 - 1}$, the resulting scheme (NA2) is expressed as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{-f(x_n) - 2f(y_n)}{f'(x_n)} \dfrac{f(x_n)f(y_n)}{6f^2(y_n) - f^2(x_n)}. \end{cases} \tag{3.9}$$

iii. Suppose $\theta = 0$, $A = 1$, and $B = 0$, according to (2.3)

$$G(0) = 1,\ G'(0) = 1,\ G''(0) = 4.$$

Taking the following weight function $G(\eta) = \dfrac{\eta}{\ln(\eta + 1) - 3\eta + 1} + 1$, then the resulting scheme (NA3) is given as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \left[\dfrac{f(y_n)}{f(x_n) \ln\left(\frac{f(y_n)}{f(x_n)} + 1\right) - 3f(y_n) + f(x_n)} + 1 \right] \dfrac{f(x_n)}{f'(x_n)}. \end{cases} \tag{3.10}$$

4. Basins of attraction

Many authors have utilized the basins of attraction technique to analyze the stability of iterative methods for nonlinear equations. This approach illustrates how different initial estimates within a specified region of the complex plane affect the behavior of the function, providing graphical comparisons between various methods.

To generate basins of attraction, we select a rectangular region D in the complex plane that contains all the roots of the nonlinear polynomial $p(z)$. Using iterative methods, we start from each initial guess $z_0 \in D$ and assign a color to each point based on the root to which the sequence converges. The color intensity reflects the number of iterations required for convergence: Brighter colors indicate fewer iterations, while darker colors represent more iterations. Points that do not converge are colored black.

We compare the outcomes of our newly proposed methods: NA1 (3.8), NA2 (3.9), and NA3 (3.10), with those of existing methods: TOM (3.1), KLM (3.2), MM (3.3), Chun1 (3.4), Chun2 (3.5), CHM (3.6), and PBM (3.7). The basins of attraction for these methods are analyzed by applying them to seven complex functions listed in Table 1, over the region $D = [-2, 2] \times [-2, 2]$ with a grid of 800×800 points. Convergence is determined by a tolerance of 10^{-3}, with a maximum of 20 iterations allowed.

Tables 2 and 3 respectively report the number of black points, indicating divergence, and their proportion to the total number of initial points for each test case. Tables 4 and 5 present the average number of iterations per convergent point and the processing time (in seconds) required to generate the basins of attraction, respectively.

Figure 1 displays the basins of attraction for the polynomial $p_1(z)$. The methods TOM, CHM, NA1, NA2, NA3, Chun1, and KLM perform well, with a low percentage of divergent points (approximately 0.125%), followed by MM, Chun2, and finally PBM, which exhibits a divergent point percentage of 0.3562%. The new methods NA1, NA2, and NA3, as well as the TOM method, produce clear boundaries between the two basins. In contrast, the remaining methods display regions near the basin boundaries where interweaving between basins occurs.

Figure 2 illustrates the basins of attraction for the polynomial $p_2(z)$. The methods TOM, NA1, NA2, and NA3 perform very well, followed by Chun1, which shows a small percentage of divergent points. The CHM method exhibits some chaotic behavior. In contrast, the methods KLM, MM, Chun2, and PBM display larger black regions, highlighting their sensitivity to the initial guess in this test case. Additionally, TOM, NA1, NA2, and NA3 exhibit smaller areas of interweaving between basins compared to the remaining methods.

Figure 3 represents the basins of attraction for the polynomial $p_3(z)$. The methods NA1, NA2, and NA3 perform well, showing the fewest black points. Chun1 and CHM follow, with a small number of black points. The methods KLM and TOM exhibit some chaotic behavior, while the remaining methods display larger black regions.

Figure 4 shows the basins of attraction for the polynomial $p_4(z)$. The methods NA2, NA1, NA3, and TOM perform well, with a low percentage of divergent points (approximately 0.26%). They are followed by Chun1, which exhibits some chaotic behavior. The remaining methods are highly sensitive to the initial guess, with significant percentages of divergent points, ranging from 5.776% for CHM to 29.57% for PBM.

Figure 5 displays the basins of attraction for the polynomial $p_5(z)$. The methods TOM, NA1, NA2, and NA3 perform very well, with no black points observed in the specified region. Chun1 and CHM follow, exhibiting a very low percentage of black points (approximately 0.024%). In contrast, the methods KLM, MM, Chun2, and PBM show greater sensitivity to the initial guess. Moreover, NA1, NA2, NA3, TOM, and CHM display smaller regions of interwoven basins compared to the other methods.

Figure 6 illustrates the basins of attraction for the polynomial $p_6(z)$. The methods NA1, NA2, NA3, and TOM perform very well. The Chun1 method shows some chaotic behavior. Meanwhile, the remaining methods are more sensitive to the initial guess, exhibiting higher percentages of divergent points, ranked from best to worst as follows: CHM, KLM, MM, Chun2, and PBM.

Figure 7 shows the basins of attraction for the polynomial $p_7(z)$. The methods NA2, NA1, NA3, and TOM exhibit some chaotic behavior, with increasing percentages of black points in that order. The remaining methods show significantly larger black regions, indicating a higher sensitivity to the initial guess. The proportion of divergent points ranges from 5.313% for Chun1 to approximately 32% for Chun2 and PBM.

In conclusion, based on Figures 1, 2, 3, 4, 5, 6, 7 and the quantitative comparisons in Tables 2, 3, and 4, we find that our new methods, NA1, NA2, and NA3, exhibit superior dynamical behavior, encompassing broader sets of initial points that lead to convergence. Based on the number of black points, the new methods, NA2, NA1, and NA3, demonstrate the best performance, with averages of 905, 1205, and 1376, respectively, followed by the TOM method with an average of 1979. In contrast, the MM, Chun2, and PBM methods exhibit higher averages, indicating greater sensitivity to the choice of initial guesses in several cases.

Regarding the mean number of iterations required for convergence, as shown in Table 4, the methods, NA1, NA2, NA3 and TOM outperform the others, with averages of 3.20, 3.24, 3.29, and 3.29 respectively. On the other hand, methods such as PBM and Chun2 require significantly higher average numbers of iterations.

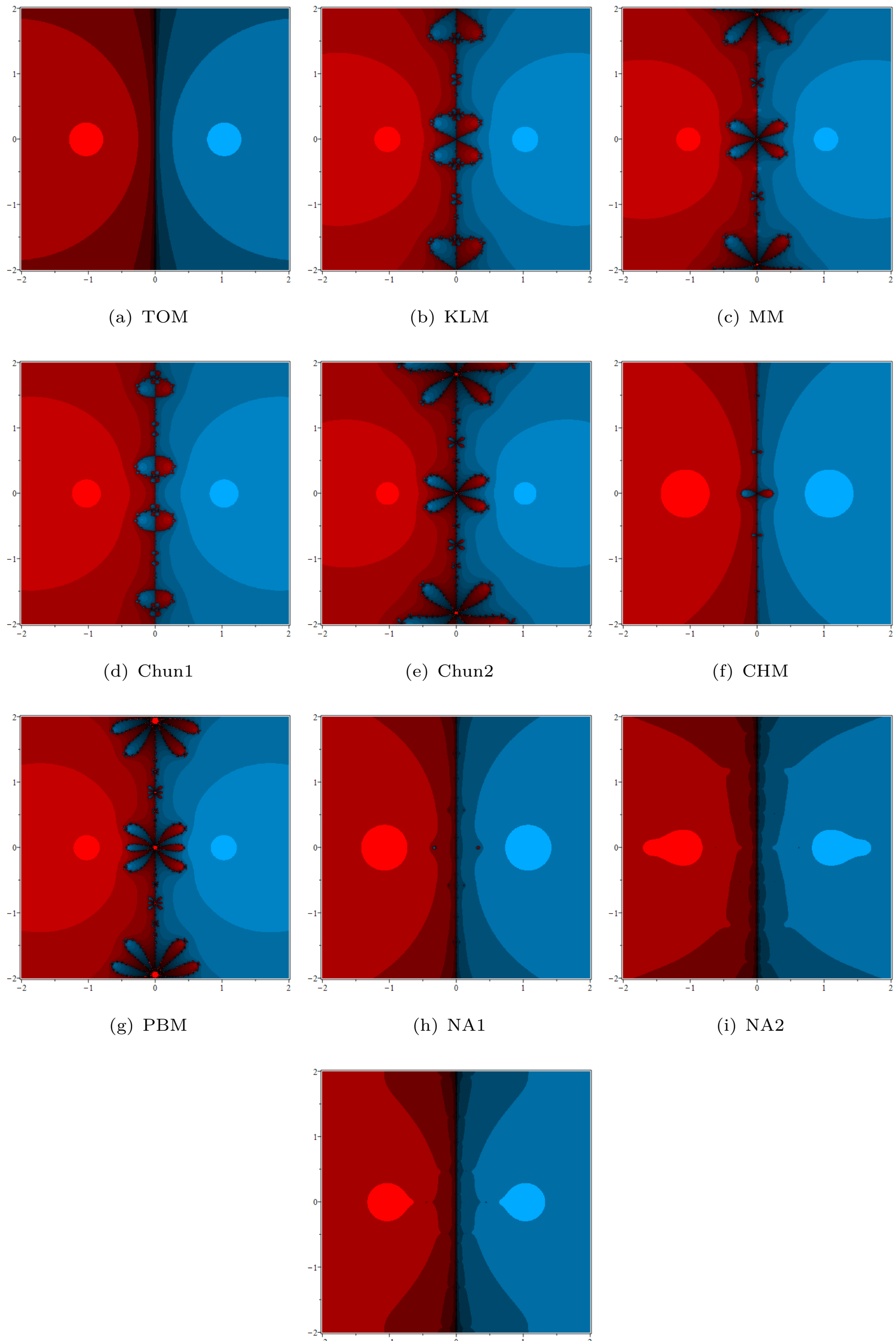
Concerning the processing time for generating the basins of attraction, as shown in Table 5, the CHM method requires the shortest time, followed by NA1, NA2, TOM, and then NA3.

Table 1. Complex polynomials and their roots accurate to 6 decimal digits.

Function	**Root**
$p_1(z) = z^2 - 1$	$1, -1$
$p_2(z) = z^3 - 1$	$1, -0.5 \pm 0.866025i$
$p_3(z) = z^3 + z + i$	$-0.562280 - 0.662359i, 0.562280 - 0.662359i, 1.324718i$
$p_4(z) = z^4 + 16$	$-1.414214 \pm 1.414214i, 1.414214 \pm 1.414214i$
$p_5(z) = z^4 - z + i$	$-0.532605 - 1.088288i, -0.759845 + 0.592595i,$
	$0.181924 + 0.732098i, 1.110525 - 0.236405i$
$p_6(z) = z^5 - 1$	$1, -0.809017 \pm 0.587785i, 0.309017 \pm 0.951057i$
$p_7(z) = z^6 - 4$	$\pm 1.259921, -0.629961 \pm 1.091124i, 0.629961 \pm 1.091124i$

5. Numerical results and engineering applications

This section investigates the effectiveness of the newly developed methods by testing various functions, including several engineering applications. Results are compared with those of well-known methods in the same family.



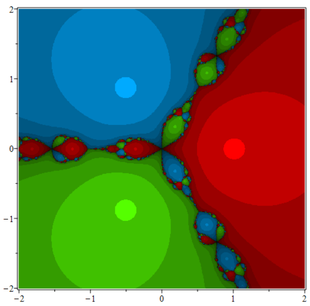
(a) TOM (b) KLM (c) MM

(d) Chun1 (e) Chun2 (f) CHM

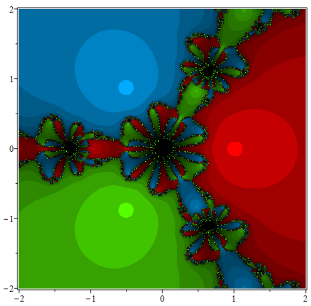
(g) PBM (h) NA1 (i) NA2

(j) NA3

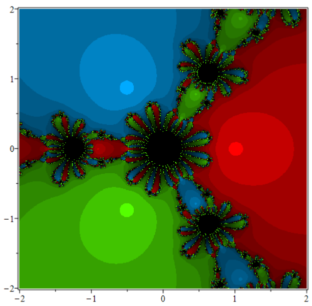
Figure 1. Basins of attraction for $p_1(z) = z^2 - 1$.



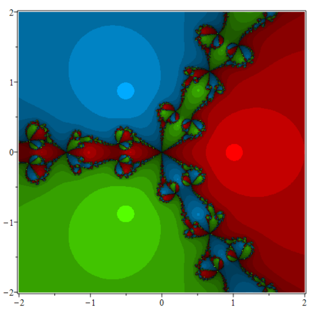
(a) TOM



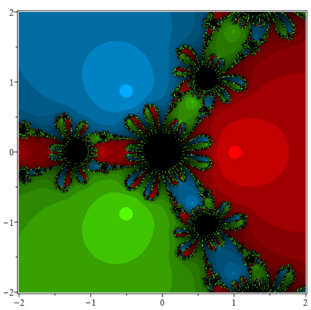
(b) KLM



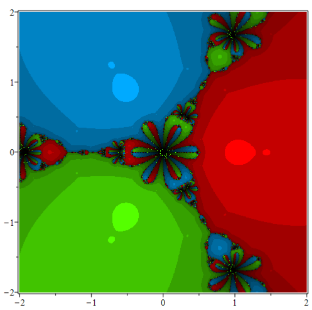
(c) MM



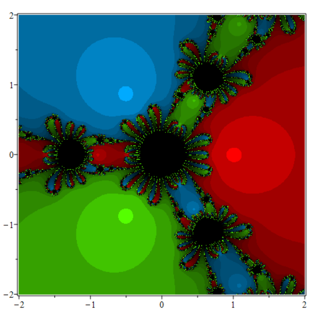
(d) Chun1



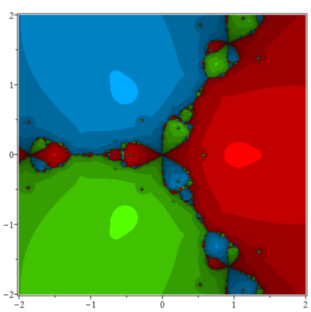
(e) Chun2



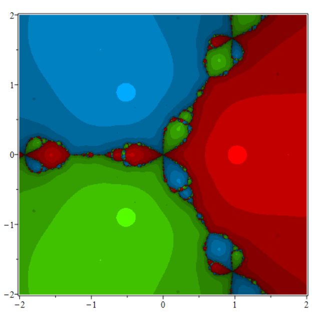
(f) CHM



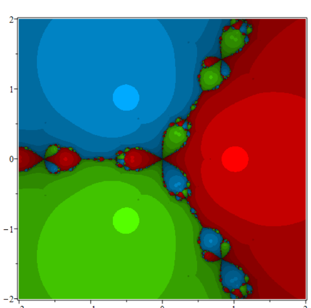
(g) PBM



(h) NA1

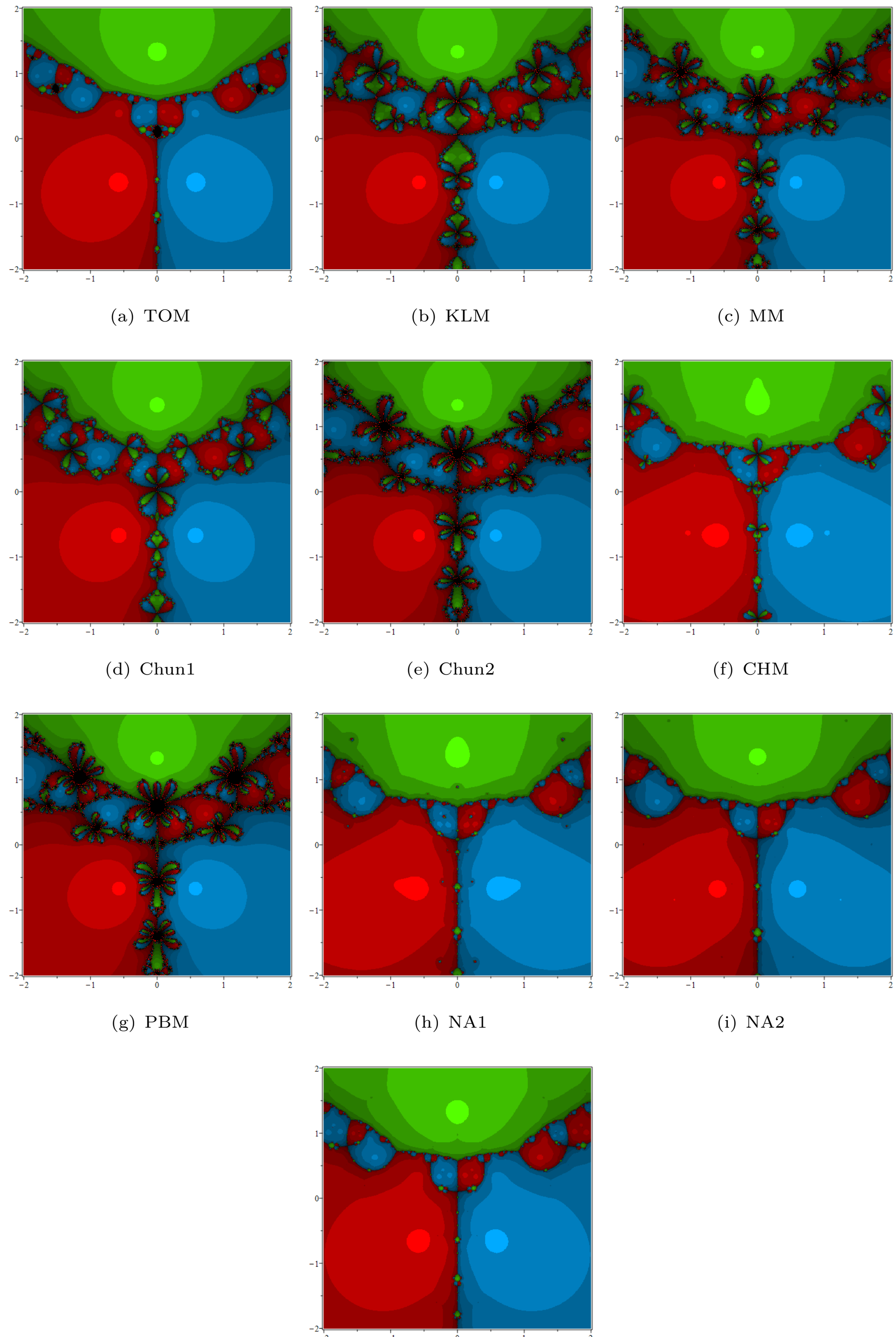


(i) NA2



(j) NA3

Figure 2. Basins of attraction for $p_2(z) = z^3 - 1$.



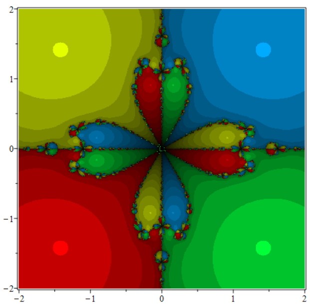
(a) TOM (b) KLM (c) MM

(d) Chun1 (e) Chun2 (f) CHM

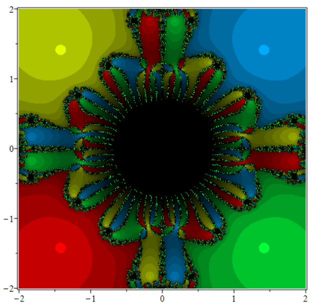
(g) PBM (h) NA1 (i) NA2

(j) NA3

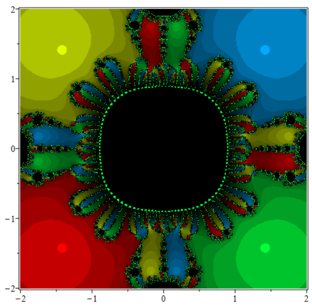
Figure 3. Basins of attraction for $p_3(z) = z^3 + z + I$.



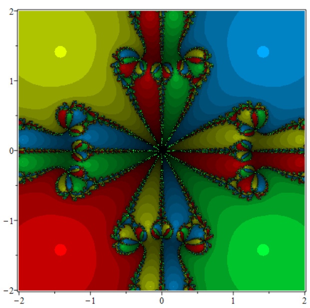
(a) TOM



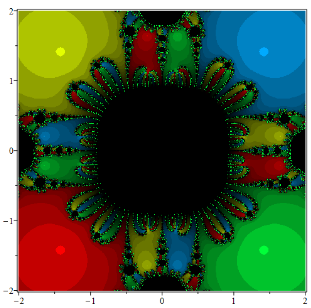
(b) KLM



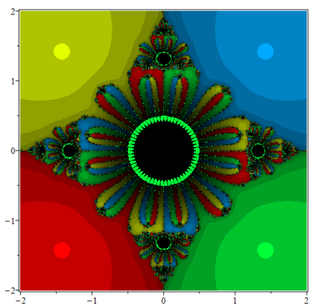
(c) MM



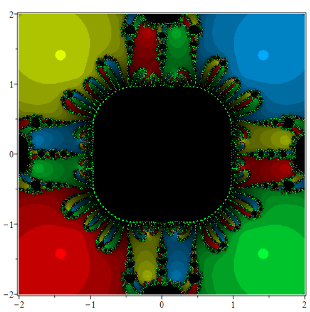
(d) Chun1



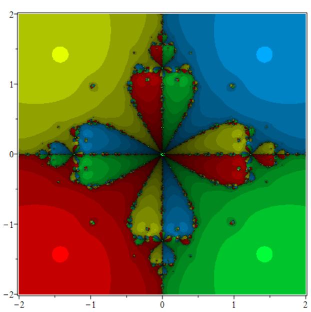
(e) Chun2



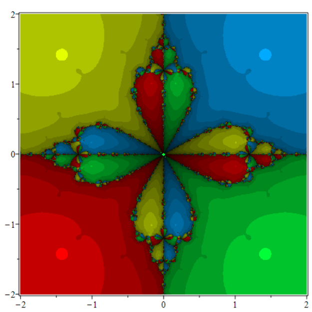
(f) CHM



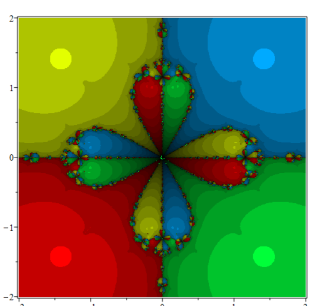
(g) PBM



(h) NA1

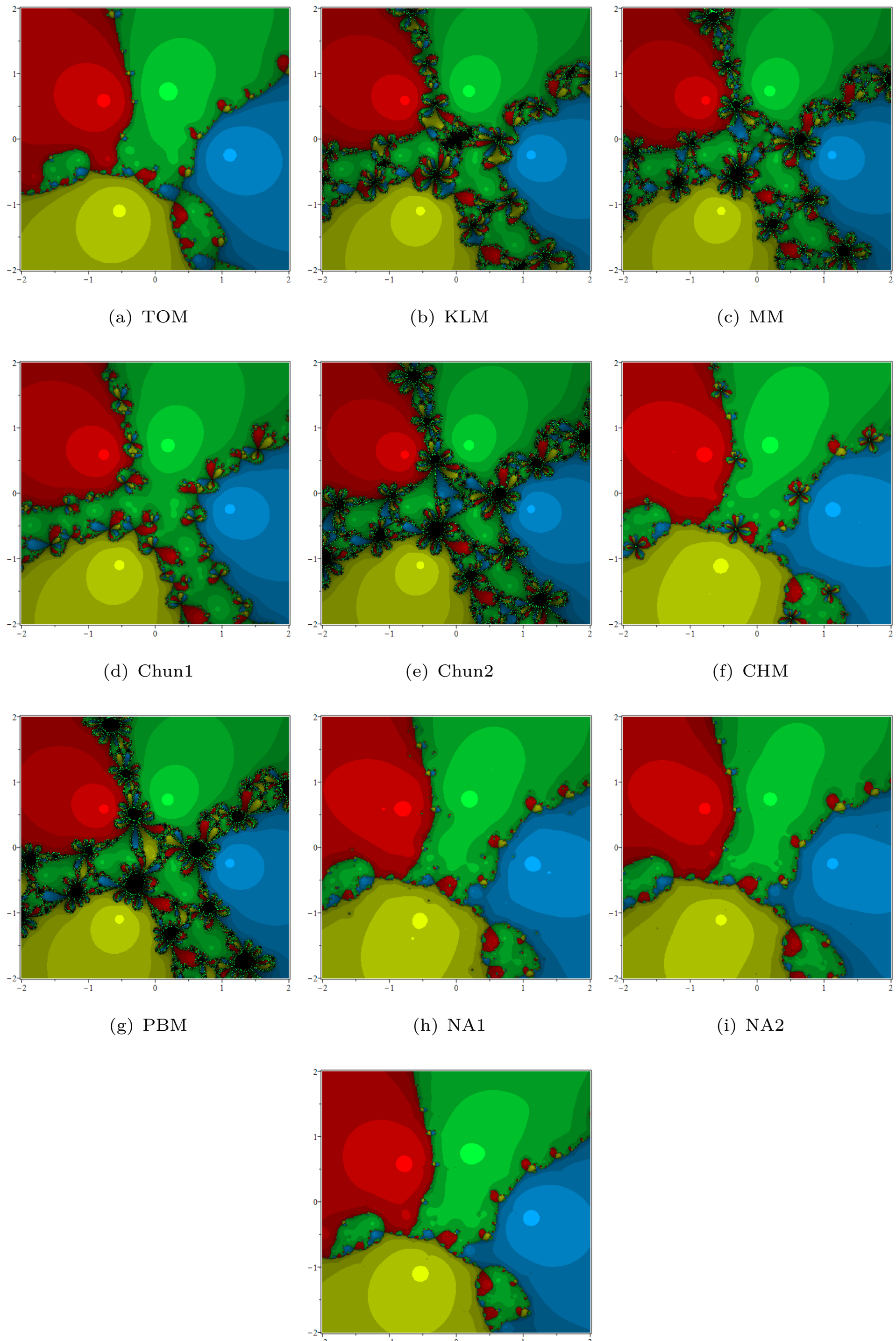


(i) NA2



(j) NA3

Figure 4. Basins of attraction for $p_4(z) = z^4 + 16$.



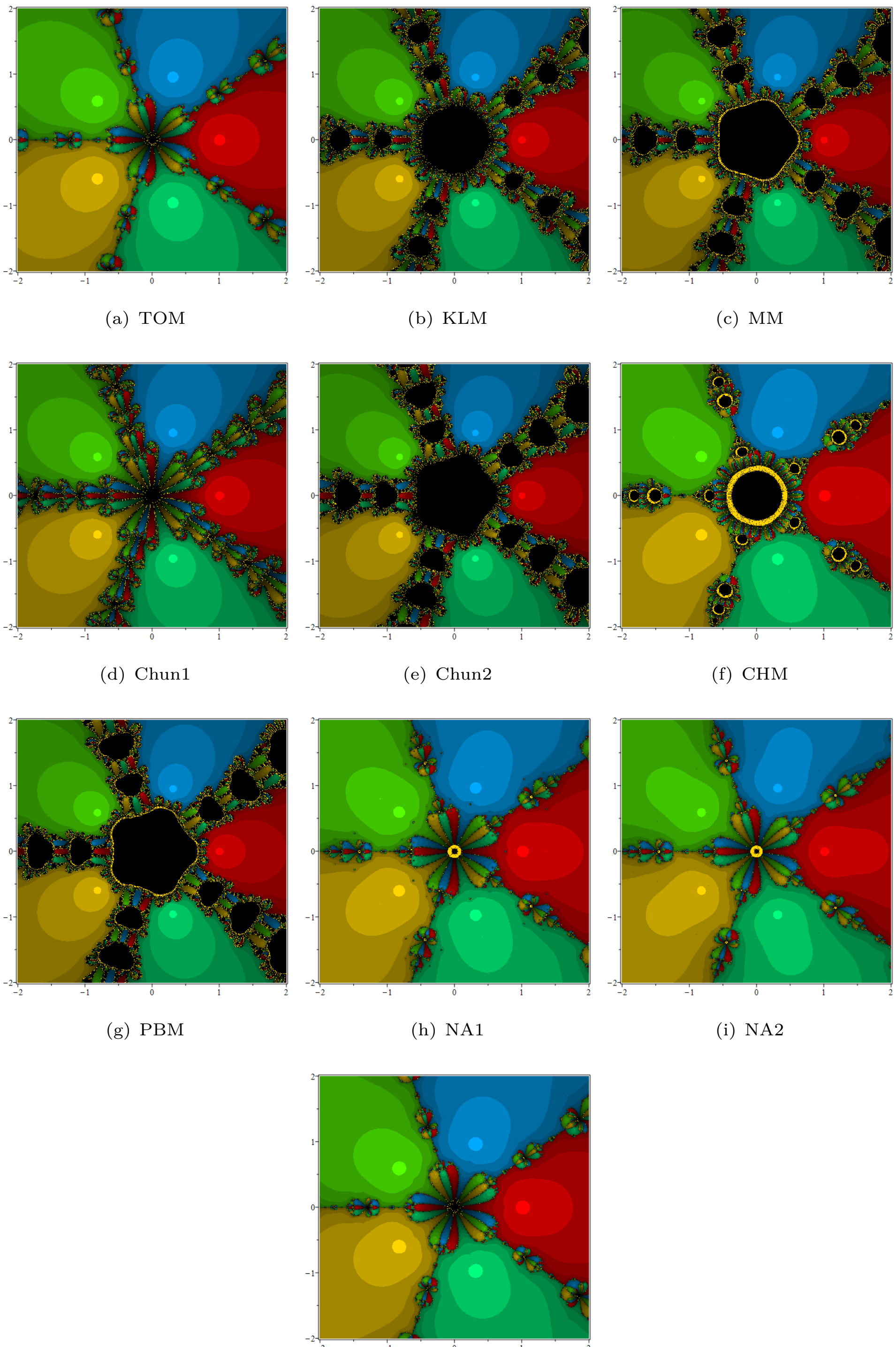
(a) TOM (b) KLM (c) MM

(d) Chun1 (e) Chun2 (f) CHM

(g) PBM (h) NA1 (i) NA2

(j) NA3

Figure 5. Basins of attraction for $p_5(z) = z^4 - z + i$.



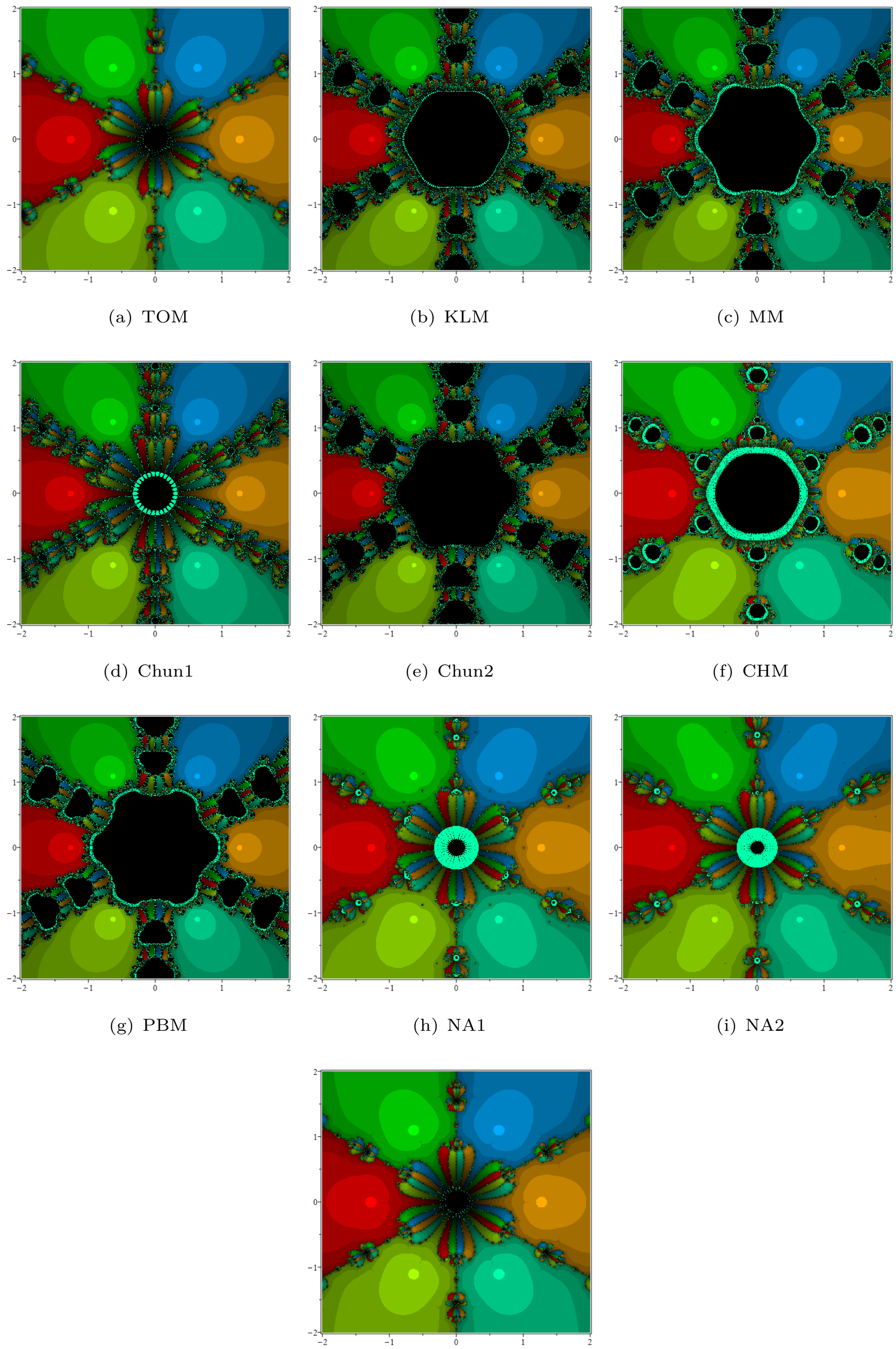
(a) TOM (b) KLM (c) MM

(d) Chun1 (e) Chun2 (f) CHM

(g) PBM (h) NA1 (i) NA2

(j) NA3

Figure 6. Basins of attraction for $p_6(z) = z^5 - 1$.



(a) TOM (b) KLM (c) MM

(d) Chun1 (e) Chun2 (f) CHM

(g) PBM (h) NA1 (i) NA2

(j) NA3

Figure 7. Basins of attraction associated with $p_7(z) = z^6 - 4$.

Table 2. Comparison of the number of black points in specific fourth-order iterative methods.

	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	$p_5(z)$	$p_6(z)$	$p_7(z)$	**Average**
TOM	798	0	2093	1756	0	1474	7730	1979
KLM	832	9500	1290	91632	14772	91734	148582	51192
MM	974	28248	5922	149216	19460	125524	178990	72619
Chun1	808	184	234	5332	154	11937	34006	7522
Chun2	1246	40270	10875	179228	31405	150336	209526	88984
CHM	800	1599	382	36968	157	39744	88340	23999
PBM	2280	48388	17879	189240	39046	154623	206852	94044
NA1	800	0	101	1652	0	737	5146	1205
NA2	800	4	78	1624	0	410	3416	905
NA3	800	8	91	1712	0	1078	5944	1376

Table 3. The ratio of black points to the total number of initial points.

	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	$p_5(z)$	$p_6(z)$	$p_7(z)$	**Average**
TOM	1.246×10^{-3}	0	3.270×10^{-3}	2.744×10^{-3}	0	2.303×10^{-3}	1.208×10^{-2}	3.696×10^{-3}
KLM	1.300×10^{-3}	1.484×10^{-2}	2.016×10^{-3}	1.432×10^{-1}	2.308×10^{-2}	1.433×10^{-1}	2.322×10^{-1}	8.193×10^{-2}
MM	1.522×10^{-3}	4.414×10^{-2}	9.253×10^{-3}	2.332×10^{-1}	3.041×10^{-2}	1.961×10^{-1}	2.797×10^{-1}	1.117×10^{-1}
Chun1	1.262×10^{-3}	2.875×10^{-4}	3.656×10^{-4}	8.331×10^{-3}	2.406×10^{-4}	1.865×10^{-2}	5.313×10^{-2}	1.249×10^{-2}
Chun2	1.947×10^{-3}	6.292×10^{-2}	1.699×10^{-2}	2.800×10^{-1}	4.907×10^{-2}	2.349×10^{-1}	3.274×10^{-1}	1.274×10^{-1}
CHM	1.250×10^{-3}	2.498×10^{-3}	5.969×10^{-4}	5.776×10^{-2}	2.453×10^{-4}	6.210×10^{-2}	1.380×10^{-1}	4.256×10^{-2}
PBM	3.562×10^{-3}	7.561×10^{-2}	2.794×10^{-2}	2.957×10^{-1}	6.101×10^{-2}	2.416×10^{-1}	3.232×10^{-1}	1.348×10^{-1}
NA1	1.250×10^{-3}	0	1.578×10^{-4}	2.581×10^{-3}	0	1.152×10^{-3}	8.041×10^{-3}	1.460×10^{-3}
NA2	1.250×10^{-3}	6.250×10^{-6}	1.219×10^{-4}	2.538×10^{-3}	0	6.406×10^{-4}	5.338×10^{-3}	1.149×10^{-3}
NA3	1.250×10^{-3}	1.250×10^{-5}	1.422×10^{-4}	2.675×10^{-3}	0	1.684×10^{-3}	9.288×10^{-3}	1.501×10^{-3}

Table 4. Comparison of the mean number of iterations per convergent point in specific fourth-order iterative methods.

	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	$p_5(z)$	$p_6(z)$	$p_7(z)$	**Average**
TOM	2.45	2.97	2.98	3.57	3.14	3.79	4.13	3.29
KLM	3.07	4.44	4.23	5.88	4.60	5.37	5.53	4.73
MM	3.30	4.60	4.41	5.60	4.91	5.22	5.36	4.77
Chun1	2.81	3.85	3.80	5.21	3.92	5.35	5.95	4.41
Chun2	3.60	5.95	4.80	5.81	5.37	5.37	5.42	5.19
CHM	2.26	3.12	2.76	4.84	3.02	3.89	4.15	3.43
PBM	3.48	4.68	4.75	5.50	5.08	5.10	5.19	4.82
NA1	2.29	2.85	2.73	3.76	2.93	3.65	4.20	3.20
NA2	2.37	2.94	2.85	3.71	3.01	3.65	4.14	3.24
NA3	2.33	2.87	2.81	3.52	2.95	3.60	3.97	3.29

All computations are performed using Maple 2021 with a precision of 2000 significant digits. The hardware platform is an ASUS laptop equipped with an Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz and 8 GB of RAM, running Microsoft Windows 10 Pro operating system.

To check the order of convergence, the computational order of convergence

Table 5. Processing time for generating the basins of attraction.

	$p_1(z)$	$p_2(z)$	$p_3(z)$	$p_4(z)$	$p_5(z)$	$p_6(z)$	$p_7(z)$	**Average**
TOM	17.71	311.91	311.01	404.11	674.84	900.53	2712.29	761.77
KLM	17.23	490.39	504.83	1077.44	957.04	1407.50	1810.91	895.62
MM	28.55	661.42	653.36	1252.45	1766.00	1569.46	2255.06	1183.47
Chun1	26.19	503.54	486.04	689.92	2011.13	1122.02	2056.43	984.18
Chun2	13.51	638.46	565.71	1148.30	933.25	1190.65	1570.22	865.44
CHM	28.69	433.12	366.03	706.61	680.84	838.23	1099.87	593.34
PBM	23.64	684.33	690.19	1268.73	2082.41	1482.91	2732.61	1380.55
NA1	18.10	373.50	403.36	544.57	1326.30	813.68	1283.86	680.48
NA2	21.00	374.41	319.30	452.63	1413.52	751.32	1507.59	691.11
NA3	19.85	399.88	311.02	700.60	1541.01	1118.03	1806.11	842.36

(COC) can be approximated using the following formula [9]:

$$\mathrm{COC} = \frac{\ln \mid (x_{n+1} - x_n)/(x_n - x_{n-1}) \mid}{\ln \mid (x_n - x_{n-1})/(x_{n-1} - x_{n-2}) \mid}.$$

The number of iterations is fixed at $n = 5$ for all examples. Tables 6 to 13 present comparisons among iterative methods, reporting the error estimation $\mid x_5 - x_4 \mid$, the computational order of convergence (COC), and the CPU time in seconds (Time). The processing time represents the mean of 1000 executions to ensure reliable values. In each test case, the best values of $|x_5 - x_4|$ and CPU time are highlighted in bold.

Example 5.1. (Normal depth in trapezoidal open channels [31]) Consider an open channel with a trapezoidal cross-section, a bed width of b, and side slopes of m horizontal to 1 vertical. If Q represents the water flow under uniform flow conditions and x denotes the depth of water in the channel, then according to Manning's equation [10, 21], we have

$$Q = \frac{C\sqrt{S}}{n} \frac{\left(bx + mx^2\right)^{5/3}}{\left(b + 2x\sqrt{1 + m^2}\right)^{2/3}}. \tag{5.1}$$

Here, C equals 1.0 for SI units and 1.486 for BG units, n is the Manning roughness coefficient, and S is the longitudinal channel slope. See [31] for further details.

Assigning specific values to the parameters [10, 31]: $b = 10\ ft, m = 2, S = 0.0006, n = 0.016, Q = 225\ ft^3/s$, we can determine the depth of water in the channel using the following equation

$$f_1(x) = 98.9027 \left(10 + 2\sqrt{5}x\right)^{2/3} - \left(10x + 2x^2\right)^{5/3} = 0. \tag{5.2}$$

The solution to this equation is $\alpha = 3.406284331340969 \cdots$. Table 6 shows the numerical results for an initial guess of $x_0 = 3.0$. For this test case, the top four methods are ranked by accuracy as follows: NA3, CHM, NA1, and TOM. In terms of CPU time, the methods rank as follows: TOM, followed by NA1 and Chun2, then NA2 and PBM.

Table 6. Numerical comparison of distinct methods applied to the test function $f_1(x)$.

Method	$\lvert x_5 - x_4 \rvert$	$\lvert f(x_5) \rvert$	**COC**	**Time**
TOM	1.55×10^{-246}	4.23×10^{-983}	4.00	**0.0027**
KLM	3.42×10^{-191}	3.73×10^{-761}	4.00	0.0034
MM	7.99×10^{-177}	1.53×10^{-703}	4.00	0.0033
Chun1	8.46×10^{-212}	8.91×10^{-844}	4.00	0.0033
Chun2	1.51×10^{-166}	2.49×10^{-662}	4.00	0.0030
CHM	2.08×10^{-271}	4.94×10^{-1083}	4.00	0.0035
PBM	6.55×10^{-196}	5.04×10^{-780}	4.00	0.0031
NA1	4.80×10^{-266}	1.42×10^{-1061}	4.00	0.0030
NA2	8.27×10^{-203}	5.92×10^{-808}	4.00	0.0031
NA3	$\mathbf{6.59 \times 10^{-333}}$	4.43×10^{-1329}	4.00	0.0036

Example 5.2. (Parachutist's problem [3]) The total force F acting on a descending parachutist is the result of two opposing forces: The downward gravitational force F_d and the upward air resistance force F_u, such that $F = F_d + F_u$.

The gravitational force $F_d = mg$, where $g \approx 9.8m/s^2$ represents the acceleration due to gravity, and m is the mass of the parachutist. The upward air resistance is modeled as $F_u = -xv$, where v is the velocity and x is the drag coefficient (kg/s). The negative sign indicates that this force acts in the upward direction, opposing the motion.

Therefore, the total force can be expressed as

$$F = mg - xv.$$

By the Newton's second law of motion $F = m\dfrac{dv}{dt}$. Substituting the expression for F, we have

$$\frac{dv}{dt} = g - \frac{x}{m}v.$$

Solving this differential equation with with the initial condition $v(0) = 0$, we get

$$v(t) = \frac{gm}{x}\left(1 - e^{-\frac{x}{m}t}\right). \tag{5.3}$$

Assume the parachutist's mass is $m = 70\ kg$. To determine the drag coefficient x required for the velocity to reach $v = 40\ m/s$ at $t = 11\ s$, we need to solve the following nonlinear equation

$$f_2(x) = \frac{686}{x}\left(1 - e^{-\frac{11x}{70}}\right) - 40 = 0. \tag{5.4}$$

We begin with an initial approximation $x_0 = 12.0\ kg/s$. Note that the solution of (5.4) is $\alpha = 15.69380132331276\cdots$. Table 7 shows the numerical results. In this test case, the top four methods are ranked by accuracy as follows: TOM, NA3, NA1, and CHM. With respect to CPU time, the ranking is as follows: CHM, followed by NA1 and PBM, then KLM.

Table 7. Numerical comparison of distinct methods applied to the test function $f_2(x)$.

Method	$\lvert x_5 - x_4 \rvert$	$\lvert f(x_5) \rvert$	COC	Time
TOM	$\mathbf{1.28 \times 10^{-273}}$	6.04×10^{-1097}	4.00	0.0032
KLM	1.19×10^{-189}	5.57×10^{-760}	4.00	0.0031
MM	3.43×10^{-178}	5.71×10^{-714}	4.00	0.0035
Chun1	1.27×10^{-208}	3.96×10^{-836}	4.00	0.0037
Chun2	1.74×10^{-169}	4.98×10^{-679}	4.00	0.0034
CHM	1.42×10^{-220}	4.40×10^{-884}	4.00	**0.0027**
PBM	1.92×10^{-187}	3.85×10^{-751}	4.00	0.0030
NA1	7.51×10^{-223}	3.41×10^{-893}	4.00	0.0030
NA2	2.13×10^{-199}	4.90×10^{-799}	4.00	0.0037
NA3	8.73×10^{-264}	2.48×10^{-1057}	4.00	0.0038

Example 5.3. (Sheet-pile walls designing model [11]) The embedment depth x of a sheet-pile wall is determined by the equation

$$x = \frac{x^3 + 2.87x^2 - 10.28}{4.62}.$$

To determine the depth, we must solve the equivalent equation

$$f_3(x) = \frac{x^3 + 2.87x^2 - 10.28}{4.62} - x = 0.$$

The required root is $\alpha = 2.0021187789538 27\cdots$. For this study, we start with an initial approximation of $x_0 = 2.5$. The numerical results and comparisons are presented in Table 8. The top four methods are ranked by accuracy in the following order: NA1, CHM, NA3, and NA2. Regarding CPU time, the methods rank as follows: TOM, followed by KLM and Chun1, then Chun2.

Table 8. Numerical comparison of distinct methods applied to the test function $f_3(x)$.

Method	$\lvert x_5 - x_4 \rvert$	$\lvert f(x_5) \rvert$	COC	Time
TOM	1.25×10^{-189}	7.84×10^{-757}	4.00	**0.0023**
KLM	1.00×10^{-151}	1.17×10^{-604}	4.00	0.0025
MM	1.76×10^{-143}	1.53×10^{-571}	4.00	0.0030
Chun1	6.25×10^{-164}	1.14×10^{-653}	4.00	0.0025
Chun2	8.40×10^{-137}	1.01×10^{-544}	4.00	0.0027
CHM	8.21×10^{-284}	4.62×10^{-1134}	4.00	0.0034
PBM	4.99×10^{-150}	7.27×10^{-598}	4.00	0.0035
NA1	$\mathbf{6.27 \times 10^{-299}}$	1.57×10^{-1194}	4.00	0.0031
NA2	1.49×10^{-202}	2.56×10^{-808}	4.00	0.0033
NA3	2.56×10^{-215}	4.73×10^{-860}	4.00	0.0028

Example 5.4. (Colebrook-White Equation) The Colebrook-White equation is used to determine the friction factor in gas pipelines with turbulent flow, see e.g. [22], and is expressed as

$$\frac{1}{\sqrt{x}} = -2\log\left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{Re\sqrt{x}}\right), \tag{5.5}$$

where x is the friction factor (dimensionless), ε is the absolute pipe roughness (mm), D is the pipe inside diameter (mm), and Re is the Reynolds number of flow (dimensionless).

Consider a gas pipeline with the following parameters [22], $D = 476$ mm, $\varepsilon = 0.03$ mm, and $Re = 11347470$. To determine the friction factor equation (5.5) is reformulated as

$$f_4(x) = \frac{1}{\sqrt{x}} + 2\log\left(\frac{0.030/476}{3.7} + \frac{2.51}{11347470\sqrt{x}}\right) = 0.$$

As an initial guess, we choose $x_0 = 0.0088$. The numerical results of different iterative methods converging to the root $\alpha = 0.0112288447313997\cdots$ are presented in Table 9. The top four methods are ranked by accuracy in this order: TOM, NA3, Chun1, and NA1. In terms of CPU time, the methods are ranked as follows: KLM, followed by Chun1, CHM, and Chun2.

Table 9. Numerical comparison of distinct methods applied to the test function $f_4(x)$.

Method	$\lvert x_5 - x_4\rvert$	$\lvert f(x_5)\rvert$	**COC**	**Time**
TOM	$\mathbf{1.38 \times 10^{-277}}$	1.18×10^{-1100}	4.00	0.0076
KLM	8.17×10^{-183}	2.44×10^{-720}	4.00	**0.0068**
MM	6.81×10^{-170}	1.81×10^{-668}	4.00	0.0076
Chun1	2.60×10^{-206}	1.18×10^{-814}	4.00	0.0071
Chun2	2.26×10^{-160}	2.96×10^{-630}	4.00	0.0075
CHM	1.59×10^{-193}	2.07×10^{-763}	4.00	0.0073
PBM	7.20×10^{-180}	1.48×10^{-708}	4.00	0.0084
NA1	1.92×10^{-195}	4.39×10^{-771}	4.00	0.0078
NA2	4.73×10^{-178}	3.08×10^{-701}	4.00	0.0079
NA3	1.34×10^{-219}	5.68×10^{-868}	4.00	0.0081

Example 5.5. (Benedict-Webb-Rubin equation) The Benedict-Webb-Rubin equation is an equation of state that describes the behavior of real gases and applies to both vapor and liquid phases. The basic form of this equation is expressed as (see e.g. [1])

$$\begin{aligned} P =& \frac{R_u T}{\bar{v}} + \left(B_0 R_u T - A_0 - \frac{C_0}{T^2}\right)\frac{1}{\bar{v}^2} + (bR_uT - a)\frac{1}{\bar{v}^3} \\ &+ \frac{\alpha a}{\bar{v}^6} + \frac{c}{\bar{v}^3 T^2}\left(1 + \frac{\gamma}{\bar{v}^2}\right)\exp\left(-\frac{\gamma}{\bar{v}^2}\right), \end{aligned} \tag{5.6}$$

where $\bar{v}$ represents the molar specific volume $(m^3/kmol)$, $R_u = 8.314\, kPa{\cdot}m^3/kmol{\cdot}K$ is the universal gas constant, T is the temperature (K), P is the pressure (kPa), and $A_0,\ B_0,\ C_0,\ a,\ b,\ c, \alpha$ and γ are eight empirical parameters specific to the fluid.

For example, consider nitrogen gas at $T = 175\ K$ and $P = 10009\ kPa$. The eight empirical parameters for nitrogen are provided in [1]. To determine the molar specific volume by (5.6), we obtain

$$f_5(x) = \frac{1454.950}{x} - \frac{74.11329618}{x^2} + \frac{0.8471236}{x^3} + \frac{0.000323088}{x^6}$$

$$+ 2.409469388 \left(1 + \frac{0.0053}{x^2}\right) \exp\left(-\frac{0.0053}{x^2}\right) \frac{1}{x^3} - 10009$$
$$=0.$$

The solution to this equation is $\alpha = 0.1050449929263866\cdots$. Using $x_0 = 0.12$, the results are shown in Table 10. Based on accuracy, the top four methods are TOM, NA3, Chun1, and CHM. In terms of CPU time, the ranking is: Chun2, followed by NA1 and PBM, then NA2.

Table 10. Numerical comparison of distinct methods applied to the test function $f_5(x)$.

Method	$\|x_5 - x_4\|$	$\|f(x_5)\|$	**COC**	**Time**
TOM	$\mathbf{1.17 \times 10^{-256}}$	6.20×10^{-1017}	4.00	0.0079
KLM	4.93×10^{-171}	1.59×10^{-673}	4.00	0.0095
MM	3.57×10^{-151}	6.84×10^{-594}	4.00	0.0093
Chun1	9.59×10^{-207}	1.00×10^{-816}	4.00	0.0081
Chun2	2.26×10^{-138}	1.49×10^{-542}	4.00	**0.0073**
CHM	2.67×10^{-192}	9.42×10^{-759}	4.00	0.0082
PBM	2.56×10^{-181}	1.15×10^{-714}	4.00	0.0076
NA1	5.27×10^{-190}	1.42×10^{-749}	4.00	0.0076
NA2	3.02×10^{-152}	2.78×10^{-598}	4.00	0.0077
NA3	3.41×10^{-207}	1.47×10^{-818}	4.00	0.0086

Example 5.6. We now apply the proposed methods to a set of nonlinear functions. Consider the following six commonly test functions and their simple zeros:

$$\begin{aligned}
&f_6(x) = x^3 + 4x^2 - 10, && \alpha = 1.365230013414096\cdots,\\
&f_7(x) = \sin^2(x) - x^2 + 1, && \alpha = 1.404491648215341\cdots,\\
&f_8(x) = e^x \sin(x) + \ln(1 + x^2), && \alpha = 0,\\
&f_9(x) = \sqrt{x^2 + x + 3} - 2\sin(x - 2) - x^2 + 1, && \alpha = 2,\\
&f_{10}(x) = \cos\left(\frac{\pi x}{2}\right) + \frac{\ln(x^2 + 2x + 2)}{x^2 + 1}, && \alpha = -1,\\
&f_{11}(x) = \sin(x) + \cos(x) + x, && \alpha = -0.4566247045676308\cdots.
\end{aligned}$$

The comparison results for Example 5.6 are shown in Tables 11, 12, and 13.

Overall, considering all test functions and given initial points in Examples 5.1-5.6, the top methods, ranked by the accuracy measure $|x_5 - x_4|$, are generally as follows: NA2, NA1, CHM, NA3, and TOM. Concerning processing time, the best method is generally TOM, followed by KLM.

For further investigations, we set a stopping criterion of $|x_n - x_{n-1}| < 10^{-50}$ for the computer programs. Table 14 presents the number of iterations required by different methods to satisfy this criterion across all test functions. It is observed that the number of iterations required by the new methods NA1, NA2, and NA3 is generally less than or equal to that required by the other methods. In general, with respect to the number of iterations, NA3 is ranked first, followed by NA1, NA2, TOM, and CHM, which all share approximately the same rank.

Table 11. Numerical comparison of different methods applied to the test functions $f_6(x)$ and $f_7(x)$ with different initial approximations.

Function	**Method**	x_0	$\lvert x_5 - x_4 \rvert$	$\lvert f(x_5) \rvert$	COC	**Time**
$f_6(x)$	TOM	1.5	2.68×10^{-318}	7.53×10^{-1271}	4.00	**0.0016**
	KLM		1.47×10^{-273}	2.52×10^{-1091}	4.00	0.0018
	MM		2.83×10^{-263}	4.66×10^{-1050}	4.00	0.0019
	Chun1		1.20×10^{-288}	6.96×10^{-1152}	4.00	0.0020
	Chun2		3.01×10^{-255}	7.62×10^{-1018}	4.00	0.0019
	CHM		1.73×10^{-367}	4.37×10^{-1468}	4.00	0.0022
	PBM		1.32×10^{-272}	1.60×10^{-1087}	4.00	0.0018
	NA1		$\mathbf{7.66 \times 10^{-372}}$	1.69×10^{-1485}	4.00	0.0019
	NA2		4.54×10^{-309}	1.04×10^{-1233}	4.00	0.0024
	NA3		1.49×10^{-352}	2.39×10^{-1408}	4.00	0.0023
$f_6(x)$	TOM	2.5	3.32×10^{-114}	1.76×10^{-454}	4.00	**0.0016**
	KLM		4.45×10^{-84}	2.09×10^{-333}	3.99	0.0020
	MM		5.75×10^{-78}	8.00×10^{-309}	3.99	0.0018
	Chun1		2.80×10^{-93}	2.08×10^{-370}	4.00	0.0021
	Chun2		6.64×10^{-73}	1.80×10^{-288}	3.99	**0.0016**
	CHM		1.17×10^{-176}	9.25×10^{-705}	4.00	0.0022
	PBM		2.44×10^{-82}	1.88×10^{-326}	3.99	0.0020
	NA1		1.86×10^{-162}	5.88×10^{-648}	4.00	**0.0016**
	NA2		$\mathbf{8.20 \times 10^{-179}}$	1.10×10^{-712}	4.00	0.0026
	NA3		3.94×10^{-134}	1.16×10^{-534}	4.00	0.0022
$f_7(x)$	TOM	1.8	4.33×10^{-160}	3.60×10^{-638}	4.00	**0.0017**
	KLM		6.76×10^{-131}	7.13×10^{-521}	4.00	0.0018
	MM		6.02×10^{-124}	6.05×10^{-493}	4.00	0.0021
	Chun1		7.38×10^{-141}	6.56×10^{-561}	4.00	0.0024
	Chun2		3.37×10^{-118}	7.46×10^{-470}	4.00	0.0025
	CHM		2.17×10^{-205}	3.79×10^{-820}	4.00	0.0024
	PBM		1.79×10^{-129}	3.47×10^{-515}	4.00	0.0021
	NA1		1.21×10^{-201}	3.70×10^{-805}	4.00	0.0028
	NA2		$\mathbf{1.51 \times 10^{-229}}$	7.11×10^{-916}	4.00	0.0025
	NA3		1.60×10^{-176}	2.79×10^{-704}	4.00	0.0020
$f_7(x)$	TOM	2.5	1.96×10^{-86}	1.52×10^{-343}	4.00	0.0020
	KLM		7.95×10^{-71}	1.36×10^{-280}	3.99	**0.0018**
	MM		7.47×10^{-67}	1.43×10^{-264}	3.99	0.0021
	Chun1		2.24×10^{-76}	5.62×10^{-303}	3.99	0.0020
	Chun2		1.63×10^{-63}	4.09×10^{-251}	3.99	0.0026
	CHM		8.39×10^{-113}	8.43×10^{-450}	4.00	0.0022
	PBM		4.47×10^{-70}	1.36×10^{-277}	3.99	0.0021
	NA1		$\mathbf{1.53 \times 10^{-114}}$	9.36×10^{-457}	4.00	0.0027
	NA2		6.04×10^{-93}	1.82×10^{-369}	4.00	0.0026
	NA3		3.82×10^{-95}	9.10×10^{-379}	4.00	0.0021

Table 12. Numerical comparison of different methods applied to the test functions $f_8(x)$ and $f_9(x)$ with different initial approximations.

Function	Method	x_0	$\lvert x_5 - x_4 \rvert$	$\lvert f(x_5) \rvert$	COC	Time
$f_8(x)$	TOM	0.5	4.52×10^{-77}	3.05×10^{-305}	3.99	0.0024
	KLM		4.51×10^{-58}	9.66×10^{-229}	3.99	**0.0021**
	MM		8.32×10^{-54}	1.50×10^{-211}	3.99	0.0029
	Chun1		2.70×10^{-64}	8.15×10^{-254}	3.99	0.0025
	Chun2		3.29×10^{-50}	4.60×10^{-197}	3.99	0.0023
	CHM		4.97×10^{-121}	4.06×10^{-482}	4.00	0.0026
	PBM		7.30×10^{-57}	6.63×10^{-224}	3.99	0.0034
	NA1		$\mathbf{1.78 \times 10^{-153}}$	6.76×10^{-612}	4.00	0.0029
	NA2		7.62×10^{-96}	2.92×10^{-380}	4.00	0.0024
	NA3		2.65×10^{-87}	1.64×10^{-346}	4.00	0.0027
$f_8(x)$	TOM	1.5	6.16×10^{-34}	1.06×10^{-132}	3.99	**0.0021**
	KLM		2.03×10^{-25}	3.99×10^{-98}	3.99	0.0026
	MM		1.92×10^{-23}	4.25×10^{-90}	3.99	0.0030
	Chun1		2.93×10^{-28}	1.13×10^{-109}	3.99	0.0036
	Chun2		9.19×10^{-22}	2.81×10^{-83}	3.99	0.0022
	CHM		1.06×10^{-48}	8.31×10^{-193}	3.99	0.0023
	PBM		7.38×10^{-25}	6.91×10^{-96}	3.99	0.0026
	NA1		2.57×10^{-47}	2.91×10^{-187}	3.99	0.0022
	NA2		$\mathbf{1.50 \times 10^{-55}}$	4.35×10^{-219}	3.99	0.0024
	NA3		2.43×10^{-38}	1.15×10^{-150}	3.99	0.0029
$f_9(x)$	TOM	1.5	6.85×10^{-204}	2.00×10^{-814}	4.00	**0.0024**
	KLM		4.62×10^{-172}	7.07×10^{-687}	4.00	0.0030
	MM		1.30×10^{-160}	5.26×10^{-641}	4.00	0.0034
	Chun1		1.87×10^{-186}	1.50×10^{-744}	4.00	0.0035
	Chun2		5.23×10^{-152}	1.64×10^{-606}	4.00	0.0028
	CHM		1.21×10^{-238}	1.26×10^{-953}	4.00	0.0027
	PBM		1.34×10^{-176}	4.92×10^{-705}	4.00	0.0029
	NA1		$\mathbf{2.30 \times 10^{-244}}$	1.65×10^{-976}	4.00	0.0029
	NA2		9.93×10^{-206}	2.59×10^{-822}	4.00	0.0028
	NA3		2.12×10^{-220}	1.50×10^{-880}	4.00	0.0031
$f_9(x)$	TOM	2.5	7.22×10^{-250}	2.46×10^{-998}	4.00	0.0026
	KLM		8.34×10^{-238}	7.50×10^{-950}	4.00	**0.0024**
	MM		2.23×10^{-233}	4.58×10^{-932}	4.00	0.0031
	Chun1		4.30×10^{-243}	4.19×10^{-971}	4.00	0.0026
	Chun2		1.82×10^{-229}	2.40×10^{-916}	4.00	0.0029
	CHM		1.06×10^{-258}	7.37×10^{-1034}	4.00	0.0031
	PBM		1.61×10^{-237}	1.05×10^{-948}	4.00	0.0032
	NA1		1.44×10^{-258}	2.54×10^{-1033}	4.00	0.0031
	NA2		$\mathbf{2.36 \times 10^{-270}}$	8.28×10^{-1081}	4.00	0.0029
	NA3		8.01×10^{-254}	3.07×10^{-1014}	4.00	0.0030

Table 13. Numerical comparison of different methods applied to the test functions $f_{10}(x)$ and $f_{11}(x)$ with different initial approximations.

Function	**Method**	x_0	$\|x_5 - x_4\|$	$\|f(x_5)\|$	COC	**Time**
$f_{10}(x)$	TOM	-0.1	8.67×10^{-96}	5.49×10^{-382}	4.00	0.0044
	KLM		1.85×10^{-82}	2.31×10^{-328}	3.99	0.0050
	MM		4.10×10^{-76}	7.02×10^{-303}	3.99	0.0041
	Chun1		4.99×10^{-89}	9.16×10^{-355}	4.00	0.0042
	Chun2		1.30×10^{-70}	8.59×10^{-281}	3.99	0.0041
	CHM		5.15×10^{-106}	3.28×10^{-423}	4.00	**0.0040**
	PBM		9.54×10^{-85}	1.64×10^{-337}	3.99	0.0047
	NA1		1.72×10^{-106}	4.08×10^{-425}	4.00	0.0049
	NA2		$\mathbf{1.26 \times 10^{-133}}$	1.04×10^{-534}	4.00	0.0041
	NA3		1.11×10^{-100}	1.10×10^{-401}	4.00	0.0048
$f_{10}(x)$	TOM	-0.8	3.40×10^{-292}	1.29×10^{-1167}	4.00	0.0042
	KLM		1.36×10^{-271}	6.83×10^{-1085}	4.00	**0.0039**
	MM		6.45×10^{-265}	4.32×10^{-1058}	4.00	0.0050
	Chun1		4.21×10^{-280}	4.65×10^{-1119}	4.00	0.0049
	Chun2		2.79×10^{-259}	1.81×10^{-1035}	4.00	0.0046
	CHM		3.61×10^{-311}	7.85×10^{-1244}	4.00	0.0053
	PBM		4.00×10^{-271}	5.06×10^{-1083}	4.00	0.0045
	NA1		8.28×10^{-311}	2.18×10^{-1242}	4.00	0.0041
	NA2		$\mathbf{4.71 \times 10^{-349}}$	2.06×10^{-1396}	4.00	0.0048
	NA3		4.42×10^{-300}	2.75×10^{-1199}	4.00	0.0042
$f_{11}(x)$	TOM	0.0	1.26×10^{-232}	6.04×10^{-930}	4.00	**0.0011**
	KLM		1.14×10^{-215}	4.86×10^{-862}	4.00	0.0013
	MM		1.18×10^{-208}	5.88×10^{-834}	4.00	0.0014
	Chun1		9.30×10^{-224}	1.95×10^{-894}	4.00	0.0016
	Chun2		7.10×10^{-203}	8.31×10^{-811}	4.00	0.0015
	CHM		1.77×10^{-244}	2.12×10^{-977}	4.00	0.0016
	PBM		3.70×10^{-217}	5.30×10^{-868}	4.00	0.0013
	NA1		3.51×10^{-245}	3.32×10^{-980}	4.00	0.0021
	NA2		$\mathbf{5.05 \times 10^{-292}}$	1.27×10^{-1167}	4.00	0.0020
	NA3		1.10×10^{-238}	3.38×10^{-954}	4.00	0.0017
$f_{11}(x)$	TOM	-1.5	2.51×10^{-161}	9.46×10^{-645}	4.00	**0.0012**
	KLM		5.85×10^{-154}	3.31×10^{-615}	4.00	0.0013
	MM		3.69×10^{-150}	5.65×10^{-600}	4.00	0.0015
	Chun1		8.75×10^{-158}	1.53×10^{-630}	4.00	0.0019
	Chun2		1.05×10^{-146}	3.96×10^{-586}	4.00	0.0016
	CHM		4.03×10^{-165}	5.74×10^{-660}	4.00	0.0018
	PBM		8.87×10^{-155}	1.75×10^{-618}	4.00	0.0015
	NA1		2.60×10^{-165}	9.89×10^{-661}	4.00	0.0021
	NA2		$\mathbf{1.65 \times 10^{-172}}$	1.44×10^{-689}	4.00	0.0022
	NA3		2.13×10^{-163}	4.70×10^{-653}	4.00	0.0019

Table 14. Methods and number of iterations (n) required for $\mid x_n - x_{n-1} \mid < 10^{-50}$.

Method	$f_1(x)$ $x_0 = 3.0$	$f_2(x)$ $x_0 = 12.0$	$f_3(x)$ $x_0 = 2.5$	$f_4(x)$ $x_0 = 0.0088$	$f_5(x)$ $x_0 = 0.12$	$f_6(x)$ $x_0 = 1.5$	$f_7(x)$ $x_0 = 1.8$	$f_8(x)$ $x_0 = 1.5$	$f_9(x)$ $x_0 = 1.5$	$f_{10}(x)$ $x_0 = -0.8$	$f_{11}(x)$ $x_0 = 0.0$
TOM	4	4	5	4	4	4	5	6	4	4	4
KLM	5	5	5	5	5	4	5	6	5	4	4
MM	5	5	5	5	5	4	5	6	5	4	4
Chun1	4	4	5	4	4	4	5	6	5	4	4
Chun2	5	5	5	5	5	4	5	6	5	4	4
CHM	4	4	4	5	5	4	4	6	4	4	4
PBM	5	5	5	5	5	4	5	6	5	4	4
NA1	4	4	4	4	5	4	5	6	4	4	4
NA2	4	5	4	5	5	4	4	5	4	4	4
NA3	4	4	4	4	4	4	5	6	4	4	4

6. Conclusion

A new family of optimal fourth-order iterative methods for finding the roots of non-linear equations has been introduced in this work. The error equation has been theoretically proven to show that the proposed methods have fourth-order convergence. We highlight that several well-known methods are special cases within this family. Additionally, three new specific methods have been derived from this family: NA1, NA2, and NA3. Comparisons with other known fourth-order methods in the same family, along with complex dynamics and basin of attraction analysis, show that the newly developed methods provide better results. Overall, NA2 is the best in terms of stability and accuracy. Several numerical examples involving real-life problems further demonstrate the applicability of the newly proposed methods.

References

[1] Y. A. Çengel, M. A. Boles and M. Kanoğlu, *Thermodynamics: An Engineering Approach*, McGraw Hill, New York, 2019.

[2] P. B. Chand, F. I. Chicharro, N. Garrido and P. Jain, *Design and complex dynamics of Potra-Ptak-type optimal methods for solving nonlinear equations and its applications*, Mathematics, 2019, 7(10), 942.

[3] P. B. Chand, F. I. Chicharro and P. Jain, *On the design and analysis of high-order Weerakoon-Fernando methods based on weight functions*, Comput. Math. Methods, 2020, 2(5), e1114, 17 pp.

[4] W. H. Chanu, S. Panday and G. Thangkhenpau, *Development of optimal iterative methods with their applications and basins of attraction*, Symmetry, 2022, 14(10), 2020.

[5] C. Chun, *Some variants of King's fourth-order family of methods for nonlinear equations*, Appl. Math. Comput., 2007, 190(1), 57–62.

[6] C. Chun, *Some fourth-order iterative methods for solving nonlinear equations*, Appl. Math. Comput., 2008, 195, 454–459.

[7] C. Chun and Y. Ham, *Some fourth-order modifications of Newton's method*, Appl. Math. Comput., 2008, 197, 654–658.

[8] A. Cordero, M. Leonardo Sepúlveda and J. Torregrosa, *Dynamics and stability on a family of optimal fourth-order iterative methods*, Algorithms, 2022, 15(387), 1–13.

[9] A. Cordero and J. R. Torregrosa, *Variants of Newton's method using fifth-order quadrature formulas*, Appl. Math. Comput., 2007, 190, 686–698.

[10] J. B. Franzini and E. J. Finnemore, *Fluid Mechanics with Engineering Applications*, McGraw-Hill, 1997.

[11] D. V. Griffithms and I. M. Smith, *Numerical Methods for Engineers*, Chapman and Hall/CRC, Taylor and Francis Group, Boca Raton, FL, USA, 2011.

[12] P. Jarratt, *Some fourth order multipoint methods for solving equations*, Appl. Math. Comput., 1966, 20(95), 434–437.

[13] P. Jarratt, *Some efficient fourth order multipoint methods for solving equations*, BIT Numer. Math., 1969, 9(2), 119–124.

[14] N. Kakarlapudi, M. Mylapalli, R. Sri and S. Marapaga, *Applications of an efficient iterative scheme for finding zeros of nonlinear equations and its basins of attraction*, Commun. Math. Appl., 2023, 14(1), 67–79.

[15] M. Khirallah and A. Alkhomsan, *Convergence and stability of optimal two-step fourth-order and its expanding to sixth order for solving nonlinear equations*, Eur. J. Pure Appl. Math., 2022, 15(3), 971–991.

[16] R. F. King, *A family of fourth order methods for nonlinear equations*, SIAM J. Numer. Anal., 1973, 10(5), 876–879.

[17] J. Kou, Y. Li and X. Wang, *A composite fourth-order iterative method for solving non-linear equations*, Appl. Math. Comput., 2007, 184(2), 471–475.

[18] H. T. Kung and J. F. Traub, *Optimal order of one-point and multipoint iteration*, J. Assoc. Comput. Math., 1974, 21, 634–651.

[19] A. K. Maheshwari, *A fourth order iterative method for solving nonlinear equations*, Appl. Math. Comput., 2009, 211(2), 383–391.

[20] M. Moscoso-Martínez, F. I. Chicharro, A. Cordero, et al., *Achieving optimal order in a novel family of numerical methods: Insights from convergence and dynamical analysis results*, Axioms, 2024, 13(7), 458.

[21] E. Maurer, *Hydraulics and Water Resources: Examples Using R*, Maurer, Ed, 2024. https://edm44.github.io/hydr-watres-book/.

[22] S. Menon, *Gas Pipeline Hydraulics*, Taylor and Francis Group, Boca Raton, London, New York, Singapore, 2005.

[23] O. Ogbereyivwe, F. J. Atajeromavwo and S. S. Umar, *Jarratt and Jarratt-variant families of iterative schemes for scalar and system of nonlinear equations*, Iran. J. Numer. Anal. Optim., 2024, 14(2), 391–416.

[24] A. M. Ostrowski, *Solution of Equations in Euclidean and Banach Space*, Academic Press, New York, 1973.

[25] A. Y. Ozban and B. Kaya, *A new family of optimal fourth-order iterative methods for nonlinear equations*, Results Control Optim., 2022, 8, 100157.

[26] M. S. Petkovic, B. Neta, L. D. Petkovic and J. Dzunic, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier, Amsterdam, Boston, Heidelberg, London, New York, 2014.

[27] S. Qureshi, I. K. Argyros, A. Soomro, et al., *A new optimal root-finding iterative algorithm: Local and semilocal analysis with polynomiography*, Numer. Algorithms, 2023, 95(4), 1715–1745.

[28] R. Sharma and A. Bahl, *An optimal fourth order iterative method for solving nonlinear equations and its dynamics*, J. Complex Anal., 2015, 1–9. DOI: 10.1155/2015/259167.

[29] G. Thangkhenpau, S. Panday and W. H. Chanu, *New efficient bi-parametric families of iterative methods with engineering applications and their basins of attraction*, Result. Control Optim., 2023, 12, 100243.

[30] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing Company, New York, 1977.

[31] A. Zein, *A new family of optimal fourth-order iterative methods for solving nonlinear equations with applications*, J. Appl. Math., 2024, 1–22. DOI: 10.1155/2024/9955247.

Received February 2025; Accepted June 2025; Available online June 2025.