Journal of Applied and Numerical Analysis
2024, Vol. 2, pp. 30–40
doi: 10.30970/ana.2024.2.30

A new optimal eighth-order iterative technique for solving simple roots of nonlinear equations

K. Devi[1], P. Maroju[1], H. Kalita[2]

[1]*Department of Mathematics, VIT-AP University, Amaravati, 522237, Andhra Pradesh, India,*
[2]*Mathematics Division, VIT Bhopal University, Madhya Pradesh, India*

Анотація. У цьому дослідженні ми представляємо ітераційну техніку восьмого порядку для розв'язання нелінійного рівняння. Запропонований нами метод є оптимальним згідно з гіпотезою Кунга-Трауба, що вимагає лише чотирьох оцінок функції на ітерацію для техніки восьмого порядку. Ми проаналізували теоретичні аспекти нашої схеми, ретельно досліджуючи її властивості збіжності через основну теорему, яка служить для демонстрації порядку збіжності. Щоб перевірити практичну корисність наших функцій оптимальної ітерації, ми проводимо порівняльний аналіз із існуючими конкурентами, використовуючи стандартні академічні задачі. Це дозволяє нам підкреслити чудову продуктивність і ефективність нашого підходу до розв'язання нелінійних рівнянь.

Abstract. In this study, we present an iterative technique of eighth order for solving a non-linear equation. Our proposed method is optimal according to Kung-Traub conjecture, requiring only four function evaluations per iteration for the eighth order technique. We analysed the theoretical aspects of our scheme, thoroughly exploring its convergence properties through the main theorem, which serves to demonstrate the convergence order. To check the practical utility of our optimal iteration functions, we conduct a comparative analysis against existing competitors using standard academic problems. This enables us to highlight the superior performance and effectiveness of our approach in solving non-linear equations.

1 Introduction

The common problem of solving nonlinear equations in various scientific and engineering fields are in the form of $\theta(\alpha) = 0$. In real-world applications, exact solutions to these equations are often impractical or even impossible to obtain, so iterative methods are utilized to find approximate solutions. Newton's iteration method is a well-known approach for solving such equations, with a order of convergence of two, making it an optimal method in terms of efficiency, requiring two function evaluations per iterative step.

In recent years, researchers have developed higher-order iterative methods that improve upon classical methods like Newton's method. These higher-order methods achieve increased convergence rates, but they also require a greater number of function evaluations per step. To balance efficiency and convergence order, a measure is called the efficiency index has been introduced. The Kung-Traub conjecture states that the order of convergence for any multi-point iterative method cannot exceed the bound of 2^{n-1}, where n is the number of functions evaluations per iteration, which is considered as the optimal order.

The experts [2–4] discussed a recent development in the field of iterative methods, specifically the elimination of derivatives from the iteration functions. The conventional challenge associated with iterative methods involving derivatives is the computational effort required to calculate these derivatives at each step. This difficulty is particularly pronounced for high-order derivatives, making such methods impractical and time-consuming for real-world problems.

Key words: non-linear equations, order of convergence, Local convergence, Lipschitz continuity condition, Fréchet derivative, Basin of attraction.

In [9], the researchers discussed the third order iterative method. In [1, 10, 14] the experts discussed the fourth order iterative method, in [11] fifth order itertaive method, sixth order iterative algorithm discussed in [5, 12] and seventh order iterative method discussed in [13]. Inspired by all these higher order iterative methods we developed the eighth order optimal iterative method.

We emphasizes the significance of optimal eighth-order multi-point derivative-free methods as a noteworthy class of iterative methods. This method offers faster convergence towards the desired root and demonstrate a superior efficiency index compared to conventional methods like Newton's method. Moreover, this method enable the achievement of a specified level of accuracy which is expressed in terms of digits within a relatively small number of iterations, further enhancing the practical applicability.

In this paper, we focuses on developing optimal eighth order iterative methods using divided difference techniques. After that we analyzes the convergence order of this method and provides numerical examples to compare them with existing optimal method. The discussion extends to the application of these methods to solve the problem of fraction conversion of nitrogen and hydrogen to ammonia and Planck's radiation law.

This paper is structured as follows. In Section 2, we focuses on the development and analysis of eighth-order iterative methods for solving nonlinear equations. In Section 3, we discusses the convergence speed that is order of convergence of our proposed method. In Section 4, we apply our technique to real-world problems, providing numerical examples and comparing it with existing methods. Finally, in Section 5 we present the conclusions drawn from the proposed method.

2 Eighth-order proposed method

In this section, we introduce the derivative-free optimal eighth order iterative method. This technique significantly enhances the rate at which the algorithm converges to the root. In 2007 Jisheng Kou et.al [14] proposed fourth order optimal technique

$$\begin{aligned} \beta_n &= \alpha_n - \frac{\theta(\alpha_n)}{\theta'(\alpha_n)}, \\ \gamma_n &= \alpha_n - \frac{\big(\theta(\alpha_n)\big)^2 + \big(\theta(\beta_n)\big)^2}{\theta'(\alpha_n)\big(\theta(\alpha_n) - \theta(\beta_n)\big)}. \end{aligned} \tag{2.1}$$

To increase the order of convergence of (2.1) we add the Newton's method as

$$w_n = \gamma_n - \frac{\theta(\gamma_n)}{\theta'(\gamma_n)}.$$

By adding third step to (2.1), we get eighth-order convergence

$$\begin{aligned} \beta_n &= \alpha_n - \frac{\theta(\alpha_n)}{\theta'(\alpha_n)}, \\ \gamma_n &= \alpha_n - \frac{\big(\theta(\alpha_n)\big)^2 + \big(\theta(\beta_n)\big)^2}{\theta'(\alpha_n)\big(\theta(\alpha_n) - \theta(\beta_n)\big)}, \\ w_{n+1} &= \gamma_n - \frac{\theta(\gamma_n)}{\theta'(\gamma_n)}. \end{aligned} \tag{2.2}$$

In equation (2.2), the number of function is five. According to the Kung-Traub conjecture the order of convergence should be 16. So to get the optimal order of convergence, we need to reduce the computational cost that is to approximate $\theta'(\gamma_n)$ by using Newton interpolation

$$p(s) = a_0 + a_1(s-\alpha) + a_2(s-\alpha)^2 + a_3(s-\alpha)^3,$$

which satisfies $p(\alpha) = \theta(\alpha)$, $p'(\alpha) = \theta'(\alpha)$, $p(\beta) = \theta(\beta)$, $p(\gamma) = \theta(\gamma)$.
From this, we get

$$p(\alpha) = \theta(\alpha) = a_0,$$
$$p'(\alpha) = \theta'(\alpha) = a_1.$$

To determine the value of a_2 and a_3, we define the first order and second order divided difference as

$$\theta[\beta, \alpha] = \frac{\theta(\beta) - \theta(\alpha)}{\beta - \alpha}, \qquad \theta[\alpha, \beta, \gamma] = \frac{\theta[\beta, \gamma] - \theta[\alpha, \beta]}{\gamma - \alpha}.$$

Clearly, we have

$$\theta[\beta, \alpha, \alpha] = \frac{\theta[\alpha, \alpha] - \theta[\beta, \alpha]}{\alpha - \beta}$$
$$= \frac{\theta[\beta, \alpha] - \theta'(\alpha]}{\beta - \alpha}.$$

Now,

$$p(\beta) = \theta(\beta) = \theta(\alpha) + \theta'(\alpha)(\beta - \alpha) + a_2(\beta - \alpha)^2 + a_3(\beta - \alpha)^3, \tag{2.3}$$
$$p(\gamma) = \theta(\gamma) = \theta(\alpha) + \theta'(\alpha)(\gamma - \alpha) + a_2(\gamma - \alpha)^2 + a_3(\gamma - \alpha)^3. \tag{2.4}$$

From equation (2.3) it follows

$$\frac{\theta(\beta) - \theta(\alpha)}{\beta - \alpha} = \theta'(\alpha) + a_2(\beta - \alpha) + a_3(\beta - \alpha)^2$$

and

$$\frac{\theta[\beta, \alpha] - \theta'(\alpha)}{\beta - \alpha} = a_2 + a_3(\beta - \alpha).$$

Thus

$$\theta[\beta, \alpha, \alpha] = a_2 + a_3(\beta - \alpha). \tag{2.5}$$

Similarly, from equation (2.4), we get

$$\theta[\gamma, \alpha, \alpha] = a_2 + a_3(\gamma - \alpha). \tag{2.6}$$

From (2.5) and (2.6), we get the value of a_3

$$a_3 = \frac{\theta[\beta, \alpha, \alpha] - \theta[\gamma, \alpha, \alpha]}{\beta - \gamma}. \tag{2.7}$$

Now using equation (2.7) in equation (2.5), we get the value of a_2

$$a_2 = \frac{\theta[\beta, \alpha, \alpha](\gamma - \alpha) - \theta[\gamma, \alpha, \alpha](\beta - \alpha)}{\gamma - \beta}. \tag{2.8}$$

Using the approximation $\theta'(\gamma_n) \approx p'(\gamma_n)$ and substituting the value of a_1, a_2, and a_3 in equation (2.2), our proposed eighth-order iterative method (PM) is

$$\begin{aligned}
\beta_n &= \alpha_n - \frac{\theta(\alpha_n)}{\theta'(\alpha_n)}, \\
\gamma_n &= \alpha_n - \frac{\big(\theta(\alpha_n)\big)^2 + \big(\theta(\beta_n)\big)^2}{\theta'(\alpha_n)\big(\theta(\alpha_n) - \theta(\beta_n)\big)}, \\
w_{n+1} &= \gamma_n - \frac{\theta(\gamma_n)}{\theta'(\alpha_n) + 2a_2(\gamma_n - \alpha_n) + 3a_3(\gamma_n - \alpha_n)^2},
\end{aligned} \tag{2.9}$$

where a_2 and a_3 is given by (2.8) and (2.7) respectively.

Now equation (2.9) is optimal as per Kung-Traub conjecture. The total number of function is four, and the order of convergence is equal 8. The efficiency index of our proposed method is $8^{\frac{1}{4}} = 1.681792830507$.

3 Convergence analysis

In this section, we prove the convergence analysis of (2.9). By employing Taylor's theorem we conduct a comprehensive analysis of the iterative technique and we derive the error equation which presents eighth order of convergence.

Theorem 3.1. *Let $\theta : D \subset \mathbb{R} \to \mathbb{R}$ be a sufficiently smooth function having continuous derivatives. If $\theta(\alpha)$ has a simple root α^* in the open interval D and α_0 chosen in sufficiently small neighbourhood of α^*, then the method* (2.9) *is of eighth-order convergence and the error equation is*

$$e_{n+1} = \left(c_2^2\left(3c_2^2 - c_3\right)\right)\left(3c_2^3 - c_3c_2 + c_4\right)e_n^8 + O\left(e_n^9\right).$$

Proof. Let $e_n = \alpha_n - \alpha^*$ be the error in n-th iteration. Using Taylor's series expansion around α^*, we get

$$\theta(\alpha_n) = \theta(\alpha^*) + \frac{\theta'(\alpha^*)}{1!}(\alpha_n - \alpha^*) + \frac{\theta''(\alpha^*)}{2!}(\alpha_n - \alpha^*)^2 + \frac{\theta'''(\alpha^*)}{3!}(\alpha_n - \alpha^*)^3 + \cdots.$$

Since α^* is the simple root, so $\theta(\alpha^*) = 0$,

$$\theta(\alpha_n) = \theta'(\alpha^*)\left[e_n + \frac{\theta''(\alpha^*)}{2!\theta'(\alpha^*)}e_n^2 + \frac{\theta'''(\alpha^*)}{3!\theta'(\alpha^*)}e_n^3 + \frac{\theta^{(IV)}(\alpha^*)}{4!\theta'(\alpha^*)}e_n^4 + \cdots\right]. \tag{3.1}$$

Let $c_k = \frac{\theta^{(k)}(\alpha^*)}{k!\theta'(\alpha^*)}$, $k = 2, 3, \cdots$. Using the value of c_k in equation (3.1), we get

$$\theta(\alpha_n) = \theta'(\alpha^*)\Big[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + \cdots\Big],$$

and clearly

$$\theta'(\alpha_n) = \theta'(\alpha^*)\Big[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + \cdots\Big].$$

From the first step of equation (2.9), we get

$$\begin{aligned}
\beta_n &= c_2e_n^2 + \left(2c_3 - 2c_2^2\right)e_n^3 + \left(4c_2^3 - 7c_3c_2 + 3c_4\right)e_n^4 \\
&+ \left(-8c_2^4 + 20c_3c_2^2 - 10c_4c_2 - 6c_3^2 + 4c_5\right)e_n^5 \\
&+ \left(16c_2^5 - 52c_3c_2^3 + 28c_4c_2^2 + \left(33c_3^2 - 13c_5\right)c_2 - 17c_3c_4 + 5c_6\right)e_n^6 \\
&- 2\big(16c_2^6 - 64c_3c_2^4 + 36c_4c_2^3 + 9\left(7c_3^2 - 2c_5\right)c_2^2 + \left(8c_6 - 46c_3c_4\right)c_2 \\
&- 9c_3^3 + 6c_4^2 + 11c_3c_5\big)e_n^7 + \big(64c_2^7 - 304c_3c_2^5 + 176c_4c_2^4 \\
&+ \left(408c_3^2 - 92c_5\right)c_2^3 + \left(44c_6 - 348c_3c_4\right)c_2^2 + \left(-135c_3^3 + 118c_5c_3 + 64c_4^2\right)c_2 \\
&+ c_4\left(75c_3^2 - 31c_5\right) - 27c_3c_6\big)e_n^8 + O\left(e_n^9\right).
\end{aligned}$$

From the second step of equation (2.9), we get

$$\begin{aligned}
\gamma_n &= \left(3c_2^3 - c_2c_3\right)e_n^4 - 2\left(9c_2^4 - 10c_3c_2^2 + c_4c_2 + c_3^2\right)e_n^5 \\
&+ \left(70c_2^5 - 130c_3c_2^3 + 30c_4c_2^2 + \left(42c_3^2 - 3c_5\right)c_2 - 7c_3c_4\right)e_n^6 \\
&- 2\big(111c_2^6 - 288c_3c_2^4 + 92c_4c_2^3 + 4\left(43c_3^2 - 5c_5\right)c_2^2 \\
&+ 2\left(c_6 - 31c_3c_4\right)c_2 - 14c_3^3 + 3c_4^2 + 5c_3c_5\big)e_n^7 \\
&+ \Big(624c_2^7 - 2076c_3c_2^5 + 799c_4c_2^4 + \left(1862c_3^2 - 239c_5\right)c_2^3 + \left(50c_6 - 965c_3c_4\right)c_2^2 \\
&+ \left(-395c_3^3 + 164c_5c_3 + 91c_4^2\right)c_2 + c_4\left(122c_3^2 - 17c_5\right) - 13c_3c_6\Big)e_n^8 + O\left(e_n^9\right).
\end{aligned}$$

From the final step of equation (2.9), we get

$$\begin{aligned} w_{n+1} = \gamma_n &- \left(3c_2^3 - c_2c_3\right)e_n^4 - 2\left(9c_2^4 - 10c_3c_2^2 + c_4c_2 + c_3^2\right)e_n^5 \\ &+ \Big(70c_2^5 - 130c_3c_2^3 + 30c_4c_2^2 + \left(42c_3^2 - 3c_5\right)c_2 - 7c_3c_4\Big)e_n^6 \\ &- 2\Big(111c_2^6 - 288c_3c_2^4 + 92c_4c_2^3 + 4\left(43c_3^2 - 5c_5\right)c_2^2 \\ &+ 2\left(c_6 - 31c_3c_4\right)c_2 - 14c_3^3 + 3c_4^2 + 5c_3c_5\Big)e_n^7 \\ &+ \Big(615c_2^7 - 2070c_3c_2^5 + 796c_4c_2^4 + \left(1861c_3^2 - 239c_5\right)c_2^3 \\ &+ \left(50c_6 - 964c_3c_4\right)c_2^2 + \big(-395c_3^3 + 164c_5c_3 \\ &+ 91c_4^2\big)c_2 + c_4\left(122c_3^2 - 17c_5\right) - 13c_3c_6\Big)e_n^8 + O\left(e_n^9\right). \end{aligned} \tag{3.2}$$

Hence, from equation (3.2), we concluded that the convergence order of the proposed method (2.9) is equal 8 and it is represented by the error equation

$$e_{n+1} = c_2^2\left(3c_2^2 - c_3\right)\left(3c_2^3 - c_3c_2 + c_4\right)e_n^8 + O\left(e_n^9\right).$$

□

4 Numerical examples

In this section, we validate the efficacy of our proposed eighth-order method through a series of illustrative numerical examples. The chosen examples are carefully selected to showcase the method's superior convergence properties and efficiency in comparison to existing techniques. Employing the Mathematica 11.3 software, we present detailed numerical results that highlight the accelerated convergence rates achieved by our method. We denote our approach as the proposed method (PM) and compare its numerical results with two existing methods, denoted as 8KHT and 8LLW. In Tables 4.1-4.4, we present the numerical comparison between our results and the results obtained from existing techniques. In the Tables 4.1-4.4, the first column represents the method names, the second column shows the number of iterations required by each method, the third column presents the approximate root, the fourth column presents the absolute value of the function at each iteration, the fifth column highlights the absolute difference between consecutive iterations, the sixth column indicates the Computational Order of Convergence (COC), and the final column reports the CPU time taken for each method.

Based on the comparison of absolute functional values, the differences between consecutive iterations, and CPU time, our proposed method (PM) demonstrates superior performance compared to the existing methods. PM achieves more accurate results with smaller differences between consecutive steps, and lower computational time as compared to existing techniques. In Figures 4.1-4.4, we present the visual comparative analysis between our technique and existing methodologies.

4.1 Real-life application problem

We compared our results with the following methods: Kung and Traub proposed the method [7] denoted by (8KHT)

$$\begin{aligned} \beta_n &= \alpha_n - \frac{\theta(\alpha_n)}{\theta'(\alpha_n)}, \\ \gamma_n &= \beta_n - \frac{\theta(\beta_n)\theta(\alpha_n)}{\left(\theta(\alpha_n) - \theta(\beta_n)\right)^2}\frac{\theta(\alpha_n)}{\theta'(\alpha_n)}, \end{aligned}$$

$$\alpha_{n+1} = \gamma_n - \frac{\theta(\alpha_n)}{\theta'(\alpha_n)} \frac{\theta(\alpha_n)\theta(\beta_n)\theta(\gamma_n)(\theta(\alpha_n))^2 + \theta(\beta_n)(\theta(\beta_n)) - \theta(\gamma_n)}{\big(\theta(\alpha_n) - \theta(\beta_n)\big)^2\big(\theta(\alpha_n) - \theta(\gamma_n)\big)^2(\theta(\beta_n) - \theta(\gamma_n))}.$$

Liu and Wang proposed the method [8] denoted by (8LLW)

$$\begin{aligned}
\beta_n &= \alpha_n - \frac{\theta(\alpha_n)}{\theta'(\alpha_n)}, \\
\gamma_n &= \beta_n - \frac{\theta(\alpha_n)}{\theta(\alpha_n) - 2\theta(\beta_n)} \frac{\theta(\beta_n)}{\theta'(\alpha_n)}, \\
\alpha_{n+1} &= \gamma_n - \frac{\theta(\gamma_n)}{\theta'(\gamma_n)} \Big(\frac{\theta(\alpha_n) - \theta(\beta_n)}{\theta(\alpha_n) - 2\theta(\beta_n)} \Big)^2 + \frac{\theta(\gamma_n)}{\theta(\beta_n) - \theta(\gamma_n)} + \frac{4\theta(\gamma_n)}{\theta(\alpha_n) + \theta(\gamma_n)}.
\end{aligned}$$

Example 4.1. Converting nitrogen and hydrogen into ammonia using a fractional approach, see [6]. The proportion that represents the amount of nitrogen and hydrogen that have been converted into ammonia in a chemical process is called the fractional conversion. We may apply this metric to track the reaction's progress and find the best circumstances for getting the highest yield. How much ammonia is produced as a percentage of the total nitrogen and hydrogen that were added to the reaction system is called the fractional conversion. To optimize ammonia production, this ratio serves as a clear indicator of the reaction's efficiency and directs the adjustment of conditions. As the quantity of ammonia produced is divided by the total quantity of nitrogen and hydrogen input into the reaction system, we get the equation for the fractional conversion of nitrogen and hydrogen to ammonia.

One way to calculate the fractional conversion is to divide the total amount of nitrogen and hydrogen delivered into the reaction system by the amount of ammonia produced. This relationship gives a ratio that shows how much ammonia (A) has been produced from nitrogen and hydrogen. We can use the fractional conversion to track how far along the reaction is and when it will be finished. Decimal or percentage forms are common for expressing fractional conversions. In this problem, the values of temperature and pressure have been taken as $500°C$ and $250\,atm$, respectively. This problem has the following non-linear form:

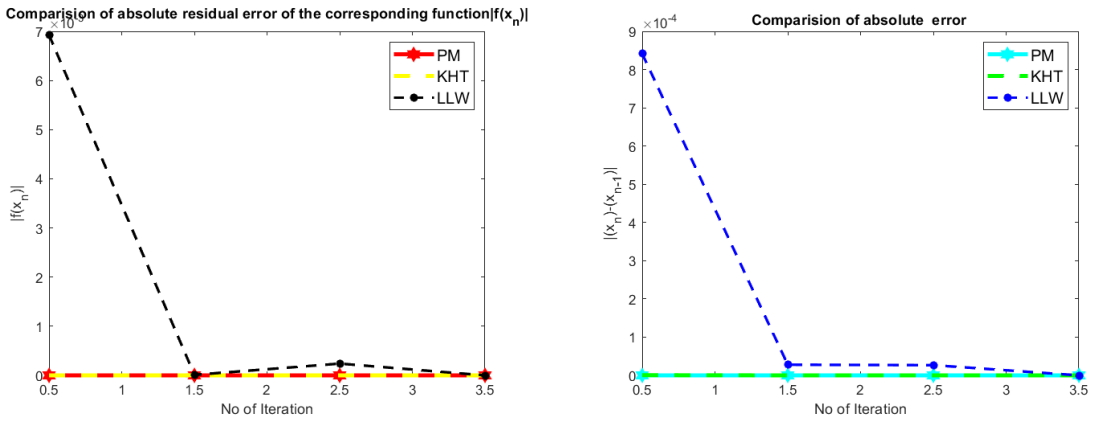
$$\theta(\alpha) = \alpha^4 - 7.79075\alpha^3 + 14.7445\alpha^2 + 2.511\alpha - 1.674 = 0. \tag{4.1}$$

The equation (4.1) has four roots. The roots are 0.384094, 0.27776, and $3.94854 \pm 0.316124i$. For the initial value α_0=0.3, the following Table 4.1 is provided.

Table 4.1. Numerical comparison of our results with existing techniques

method	n	α_n	$\mid \theta(\alpha_n) \mid$	$\mid \alpha_{n-1} - \alpha_n \mid$	COC	CPU Time
PM	1	0.27776	4.3705×10^{-12}	7.21367×10^{-13}		
	2	0.27776	2.11053×10^{-12}	2.69063×10^{-13}	8	0.156
8KHT	1	0.27776	4.71531×10^{-8}	5.24834×10^{-7}		
	2	0.27776	1.11022×10^{-6}	Indeterminate	8	0.157
8LLW	1	0.278602	0.00757591	0.000841346		
	2	0.277761	0.0000107518	0.0000278803	8	0.235
	3	0.277733	2.3973×10^{-4}	2.67015×10^{-5}		

The Table 4.1 shows that the proposed method (PM) performs significantly better than the existing techniques, (8KHT and 8LLW). In PM, both the absolute value of the function and the difference between consecutive errors are consistently smaller which indicates higher accuracy. In



Residual Error for equation (4.1) Absolute Error for equation (4.1)

Fig. 4.1. Graphical comparison of our results with existing techniques

comparison with 8KHT, the error of PM is almost half. Additionally, PM takes less time to compute, making it more efficient. The Figure 4.1 also clearly demonstrate that the error is less in PM as compared to existing techniques.

Example 4.2. Planck's blackbody radiation law, see [10]. Planck's radiation law, which is sometimes called Planck's blackbody radiation law, describes the electromagnetic radiation spectrum density emitted by a blackbody at a given temperature when it is in thermal equilibrium. According to this law, the rate of radiation and the blackbody's temperature are directly proportional to the spectrum radiance, which is the quantity of radiation emitted per unit area, unit solid angle, and unit frequency. The mathematical foundations of the emission of radiation from a blackbody under different conditions are provided by this basic law

$$\theta(\alpha) = e^{-\alpha} + \frac{\alpha}{5} - 1 = 0, \tag{4.2}$$

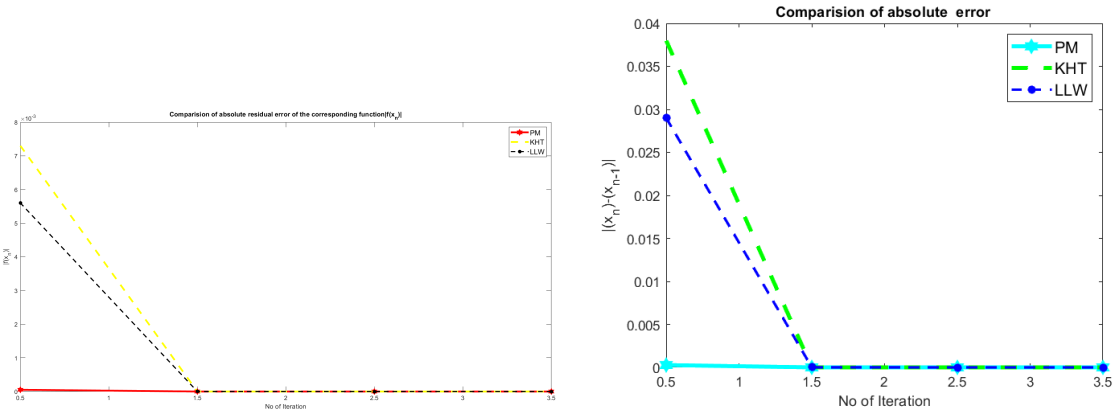
where α stands for the maximal wavelength. The exact roots of equation (4.2) are 0 and 4.96511. For the initial value α_0= 3, the following Table 4.2 is provided.

Table 4.2. Numerical comparison of our results with existing techniques

method	n	α_n	$\mid \theta(\alpha_n) \mid$	$\mid \alpha_{n-1} - \alpha_n \mid$	COC	CPU Time
PM	1	4.9653832924707295940	5.2×10^{-5}	2.7×10^{-4}	8	0.828
	2	4.9651142317442763037	2.0×10^{-38}	1.0×10^{-37}		
	3	4.9651142317442763037	8.8×10^{-306}	4.6×10^{-305}		
8KHT	1	5.0029184879164884514	0.0073	0.038	8	0.843
	2	4.9651142319790888075	4.5×10^{-11}	2.3×10^{-10}		
	3	4.9651142317442763037	7.0×10^{-44}	3.7×10^{-43}		
8LLW	1	4.9940825929871859263	0.0056	0.029	8	0.843
	2	4.9651194888792980915	1.0×10^{-6}	5.3×10^{-6}		
	3	4.9651142317444518319	3.4×10^{-14}	1.8×10^{-13}		

The Table 4.2 shows that, in PM from first to the second iteration, and the second to the third iteration, the error is approximately reduced eight times. When compared with 8KHT method, the

error is reduced by approximately six times, while compared with 8LLW method, the error reduction is approximately 20 times which indicates higher accuracy in PM. The Figure 4.2 provides the visual representation of less error of PM as compared to existing techniques.



Residual Error for equation (4.2) Absolute Error for equation (4.2)

Fig. 4.2. Graphical comparison of our results with existing techniques

4.2 Academic problem

Example 4.3. Consider the nonlinear equation

$$\theta(\alpha) = \ln(\alpha^2 + \alpha + 2) - \alpha + 1 = 0. \tag{4.3}$$

The exact root of the equation (4.3) is 4.15259. For the initial value $\alpha_0 = 4$, the following Table 4.3 is provided.

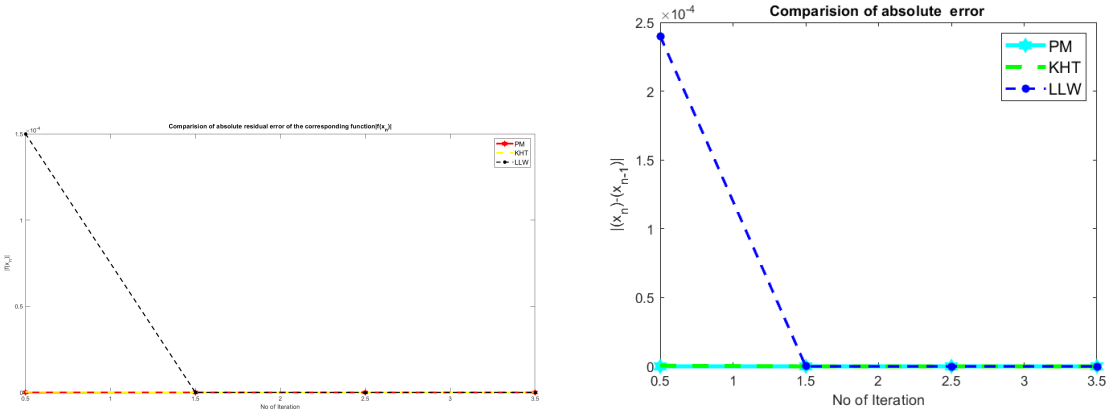
Table 4.3. Numerical comparison of our results with existing techniques

Method	n	α_n	$\mid \theta(\alpha_n) \mid$	$\mid \alpha_{n-1} - \alpha_n \mid$	COC	CPU Time
PM	1	4.1525907367571995494	2.5×10^{-14}	4.1×10^{-14}	8	1.61
	2	4.1525907367571582750	5.6×10^{-115}	9.3×10^{-115}		
	3	4.1525907367571582750	3.8×10^{-920}	6.3×10^{-920}		
8KHT	1	4.1525912474738348606	3.1×10^{-7}	5.1×10^{-7}	8	1.673
	2	4.1525907367571582750	3.4×10^{-29}	5.7×10^{-29}		
	3	4.1525907367571582750	5.3×10^{-117}	8.8×10^{-117}		
8LLW	1	4.1528316692180028950	0.00015	0.00024	8	1.766
	2	4.1525907373512851664	3.6×10^{-10}	5.9×10^{-10}		
	3	4.1525907367571582750	2.2×10^{-21}	3.6×10^{-21}		

From Table 4.3, we can see that, the error in PM decreases approximately eight times from the first to the second iteration, and from the second to third iteration. Compared to the 8KHT method and 8LLW method, the error in PM is reduced to approximately seven times and 43 times respectively. That gives better result. The Figure 4.3 visually shows, the smaller error in each iteration of PM as compared to existing techniques.

Example 4.4. Consider the nonlinear equation

$$\theta(\alpha) = \alpha^3 + 4\alpha^2 - 10 = 0. \tag{4.4}$$



Residual Error for equation (4.3) Absolute Error for equation (4.3)

Fig. 4.3. Graphical comparison of our results with existing techniques

The exact root of equation (4.4) is 1.36523. For the initial value $\alpha_0 = 1$, the following Table 4.4 is provided.

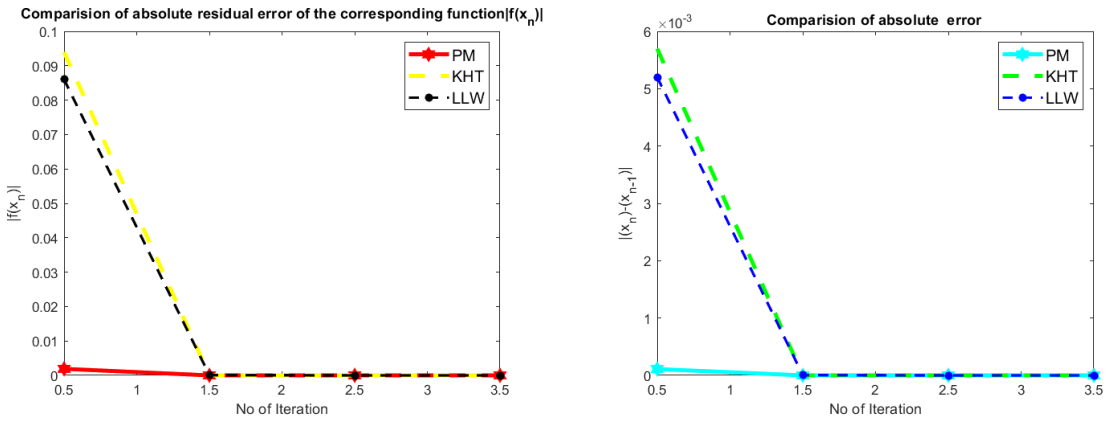
Table 4.4. Numerical comparison of our results with existing techniques

method	n	α_n	$\mid \theta(\alpha_n) \mid$	$\mid \alpha_{n-1} - \alpha_n \mid$	COC	CPU Time
PM	1	1.3653426948260834139	0.0019	0.00011	8	1.414
	2	1.3652300134140968458	2.2×10^{-32}	1.3×10^{-33}		
	3	1.3652300134140968458	8.6×10^{-264}	5.2×10^{-265}		
8KHT	1	1.3709171842544731067	0.094	0.0057	8	1.572
	2	1.3709171842544731067	3.5×10^{-9}	2.1×10^{-10}		
	3	1.3652300134140968458	7.0×10^{-39}	4.3×10^{-40}		
8LLW	1	1.3600211459109570607	0.086	0.0052	8	1.429
	2	1.3652348576908087136	0.000080	4.8×10^{-6}		
	3	1.3652300134183159594	7.0×10^{-11}	4.2×10^{-12}		

In Table 4.4, we can see that the error in PM decreases approximately eight times from the first to the second iteration, and from the second to third iteration. Compared PM with 8KHT method, the error decreases approximately seven times, while with 8LLW method PM achieves a much larger error reduction approximately 24 times which indicates that PM is much effective at minimizing error. The Figure 4.4 provides a clear visual of the error decreases with each iteration, showing the ability of PM to reduce errors efficiently.

5 Conclusion

In this study, we have introduced an optimal eighth-order iterative method for solving nonlinear equations, by using the divided difference approximation. The method involves performing four function evaluations per iteration, achieving a convergence order of eighth. Through convergence analysis and numerical examples, we demonstrate the proposed method satisfied of Kung-Traub conjecture. Our proposed eighth-order PM method has been tested against known schemes, showcasing its superiority. We applied the newly developed method, along with existing ones, to solve



Residual Error for equation (4.4) Absolute Error for equation (4.4)

Fig. 4.4. Graphical comparison of our results with existing techniques

problems such as Conversion of nitrogen and hydrogen into ammonia and Planck's radiation law. The results obtained highlighting the effectiveness of the eighth order PM method. Our numerical experiments suggest that this method offer a valuable alternative for efficiently solving non-linear equations.

Declarations

Conflict of Interest.
The authors declare that there is no conflict of interest.
Funding.
This research did not receive any funding.
Author Contributions.
All authors contributed equally to the research and preparation of this article.

References

1. Solaiman, Obadah Said, Samsul Ariffin Abdul Karim, Ishak Hashim: Optimal fourth-and eighth-order of convergence derivative-free modifications of King's method. Journal of King Saud University-Science. **31** (4), 1499-1504 (2019)
2. Yasmin, Nusrat, Fiza Zafar, Saima Akram: Optimal derivative-free root finding methods based on the Hermite interpolation. J. Nonlinear. Sci. Appl. **9**. 4427-4435 (2016)
3. Cordero, Alicia, et al.: A new technique to obtain derivative-free optimal iterative methods for solving nonlinear equations. Journal of Computational and Applied Mathematics. **252**, 95-102 (2013)
4. Zafar, Fiza, et al.: A general class of derivative free optimal root finding methods based on rational interpolation. The Scientific World Journal. 934260, 1-12 (2015)
5. Rehman, M.A., Amir Naseem, Thabet Abdeljawad: Some novel sixth order iteration schemes for computing zeros of nonlinear scalar equations and their applications in engineering. Journal of Function Spaces. 5566379, 1-11 (2021)
6. Solaiman, Obadah Said, Ishak Hashim: Optimal Eighth-Order Solver for Nonlinear Equations with Applications in Chemical Engineering. Intelligent Automation and Soft Computing. **27** (2), 379-390 (2021)
7. Kung, H.T., Traub, J.F.: Optimal order of one-point and multipoint iteration. Journal of the ACM (JACM). **21** (4), 643-651 (1974)

8. Liu, Liping, Xia Wang: Eighth-order methods with high efficiency index for solving nonlinear equations. Applied Mathematics and Computation. **215** (9), 3449-3454 (2010)
9. Darvishi, Mohammad Taghi, Ali Barati: A third-order Newton-type method to solve systems of nonlinear equations. Applied Mathematics and Computation. **187** (2), 630-635 (2007)
10. Naseem, Amir, Rehman, M.A., Jihad Younis: Some real life applications of a newly designed algorithm for nonlinear equations and its dynamics via computer tsools. Complexity. 9234932, 1-9 (2021)
11. Rafiullah, M: A fifth-order iterative method for solving nonlinear equations. Numerical Analysis and Applications. **4** (3), 239-243 (2011)
12. Yaseen, Saima, Fiza Zafar: A new sixth-order Jarratt-type iterative method for systems of nonlinear equations. Arabian Journal of Mathematics. **11** (3), 585-599 (2022)
13. Wang, Xiaofeng, Tie Zhang: A family of Steffensen type methods with seventh-order convergence. Numerical Algorithms. **62**, 429-444 (2013)
14. Kou, Jisheng, Yitian Li, Xiuhua Wang: A composite fourth-order iterative method for solving non-linear equations. Applied Mathematics and Computation. **184** (2), 471-475 (2007)

Received 19.09.2024
Revised 08.10.2024