Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.015285
Article

Tech Science Press

Optimal Eighth-Order Solver for Nonlinear Equations with Applications in Chemical Engineering

Obadah Said Solaiman and Ishak Hashim[*]

Department of Mathematical Sciences, Universiti Kebangsaan Malaysia, Bangi Selangor, 43600, Malaysia
[*]Corresponding Author: Ishak Hashim. Email: ishak_h@ukm.edu.my

Received: 13 November 2020; Accepted: 01 December 2020

Abstract: A new iterative technique for nonlinear equations is proposed in this work. The new scheme is of three steps, of which the first two steps are based on the sixth-order modified Halley's method presented by the authors, and the last is a Newton step, with suitable approximations for the first derivatives appeared in the new scheme. The eighth-order of convergence of the new method is proved via Mathematica code. Every iteration of the presented scheme needs the evaluation of three functions and one first derivative. Therefore, the scheme is optimal in the sense of Kung-Traub conjecture. Several test nonlinear problems are considered to compare the performance of the proposed method according to other optimal methods of the same order. As an application, we apply the new scheme to some nonlinear problems from the field of chemical engineering, such as a chemical equilibrium problem (conversion in a chemical reactor), azeotropic point of a binary solution, and volume from van der Waals equation. Comparisons and examples show that the presented method is efficient and comparable to the existing techniques of the same order.

Keywords: Nonlinear equations; root finding method; iterative methods; Halley's method; optimal order of convergence

1 Introduction

Searching for a solution of $g(x) = 0$, when $g(x)$ is nonlinear is highly significant in mathematics. Newton's iterative technique for solving such equations is defined as

$$z_{n+1} = z - \frac{g(z_n)}{g'(z_n)}. \tag{1}$$

It was shown by Traub [1] that the scheme given by (1) has the second-order of convergence. Many researchers have improved the method of Newton to attain better results and to increase the convergence order, for instance see [2–5] and the references therein. Petković [6] has presented a general class of multipoint root finding methods of arbitrary order 2^n. Because of the huge number of iterative techniques that appear in the literature, Petković et al. [7] have presented a review for the most efficient iterative methods and developed techniques in a general sense. Also, Cordero et al. [8] have presented a general survey on optimal iterative schemes and how to design optimal methods of different orders.

One of the most famous improvements of Newton's scheme is the technique of order three given by Halley [9]:

$$z_{n+1} = z_n - \frac{2g(z_n)g'(z_n)}{2(g'(z_n))^2 - g(z_n)g''(z_n)}. \tag{2}$$

Halley's method has been studied widely and improved in different ways. For example, a two-step Halley's scheme with order of convergence equals six was implemented by Noors et al. [10] using a predictor-corrector technique. But finding the second derivative is not always an easy task. Because of that Noor et al. [11] have improved the previous technique with the help of the finite difference and implemented a new second derivative-free scheme of order five. Very recently, Said Solaiman et al. [12] have established two sixth-order modifications of Halley's method, with one of them without second derivative.

One of the most common ways to compare the efficiency of iterative methods is the efficiency index which can be determined by $q^{1/r}$, where q is convergence order of the iterative scheme and r represents number of functions needed to be found at each iteration. Kung et al. [13] mentioned in a conjecture that the iterative scheme with the number of functional evaluations equals r is optimal if its order of convergence equals 2^{r-1}. Many authors have constructed optimal iterative methods of different orders. The default way for constructing optimal method is the composition technique together with the usage of some interpolations and approximations to minimize the number of functional evaluations. Different optimal fourth-order iterative methods have been constructed, see for examples [14–16]. Optimal eighth-order of convergence methods have been presented by many authors, see [2,17,18–22]. A comparison using the dynamics of different families of optimal eighth-order of convergence methods was proposed by Chun et al. [23].

We propose in this work a new optimal eighth-order iterative technique for nonlinear equations. The new method is a modification of the modified Halley method (MH2) introduced by Said Solaiman et al. [12]. We use the composition technique with Hermite's interpolation for the first derivative to reach the eighth-order of convergence with optimality, which can be considered as the major motivation of this research. The work in this paper is distributed as follows. In Section 2, below the new scheme is illustrated. In Section 3, the order of convergence of the new scheme is determined. In Section 4, four chemical engineering problems in addition to six nonlinear examples are used to demonstrate the efficiency of the proposed scheme, and tables are used to illustrate the comparison between our optimal method with other techniques having equal order. Lastly, in Section 5 the conclusion is given.

2 The New Method

Let $g(x) = 0$ be an equation such that $g(x)$ is a nonlinear function defined on some open interval A and sufficiently differentiable. Let $\alpha \in A$ be a simple root of $g(x)$, and consider x_0 as an initial guess which is sufficiently close to α. Said Solaiman and Hashim [12] obtained the following iterative scheme using Taylor's expansion of $g(x)$ with Newton's and Halley's methods.

Algorithm 1: *Let x_0 be an initial guess of the solution of $g(x) = 0$. Then we can approximate x_{n+1} by the iterative method defined by:*

$$\begin{cases} y_n = x_n - \dfrac{g(x_n)}{g'(x_n)}, \\ x_{n+1} = y_n - \dfrac{g(y_n)}{g'(y_n)} - \dfrac{2(g(y_n))^2 g'(y_n) g''(y_n)}{4(g'(y_n))^4 - 4g(y_n)(g'(y_n))^2 g''(y_n) + (g(y_n))^2 (g''(y_n))^2}. \end{cases} \tag{3}$$

Said Solaiman et al. [12] named Algorithm 1 as MH1. They proved that MH1 is of order of convergence equals six. MH1 needs at each iteration the evaluation of two functions, two first derivatives, and one second derivative evaluation. So, the efficiency index of MH1 is $(6)^{\frac{1}{5}} \approx 1.431$, which is not as good as Halley's method.

To make the efficiency index of Algorithm 1 better, Hermite's approximation of the second derivative is used to produce a second derivative-free method given by:

Algorithm 2: *Let x_0 be an initial guess of the solution of $g(x) = 0$. Then we can approximate x_{n+1} by the iterative method defined by:*

$$\begin{cases} y_n = x_n - \dfrac{g(x_n)}{g'(x_n)}, \\ x_{n+1} = y_n - \dfrac{g(y_n)}{g'(y_n)} - \dfrac{2(g(y_n))^2 g'(y_n) R(x_n, y_n)}{4(g'(y_n))^4 - 4g(y_n)(g'(y_n))^2 R(x_n, y_n) + (g(y_n))^2 (R(x_n, y_n))^2}, \end{cases} \tag{4}$$

where the second derivative $g''(y_n)$ is approximated by

$$R(x_n, y_n) = \left[3\frac{g(y_n) - g(x_n)}{y_n - x_n} - 2g'(y_n) - g'(x_n)\right]\frac{2}{x_n - y_n}. \tag{5}$$

Algorithm 2 is called MH2. Said Solaiman et al. [12] proved that it is of order six. MH2 needs at each iteration the computation of two functions, and two first derivatives only. So, MH2 has efficiency index equals $(6)^{\frac{1}{4}} \approx 1.565$, which is better than $(6)^{\frac{1}{5}} \approx 1.431$ of MH1 and $3^{\frac{1}{3}} \approx 1.442$ of Halley's method.

In order to reach the optimality, we reduce the number of functions needed to be evaluated at each iteration by using divided differences, Hermite's interpolation, and the composition of Algorithm 2 with Newton's method. Now, by using Algorithm 2 as a predictor, and Newton's technique as a corrector one obtains the following algorithm:

Algorithm 3: *Let x_0 be an initial guess of the solution of $g(x) = 0$. Then we can approximate x_{n+1} by the iterative method defined by:*

$$\begin{cases} y_n = x_n - \dfrac{g(x_n)}{g'(x_n)}, \\ w_n = y_n - \dfrac{g(y_n)}{g'(y_n)} - \dfrac{2(g(y_n))^2 g'(y_n) R(x_n, y_n)}{4(g'(y_n))^4 - 4g(y_n)(g'(y_n))^2 R(x_n, y_n) + (g(y_n))^2 (R(x_n, y_n))^2}, \\ x_{n+1} = w_n - \dfrac{g(w_n)}{g'(w_n)}. \end{cases} \tag{6}$$

Algorithm 3 has order of convergence equals twelve with error term $e_{n+1} = c_2^5(c_4 - c_2c_3)^2 e_n^{12} + O(e_n^{13})$. At each iteration, Algorithm 3 requires three function evaluations and needs three first derivatives. Our goal is to rewrite $g'(y_n)$ and $g'(w_n)$ by using a combination of already evaluated functions.

Using the second-order polynomial interpolation of the function $g(y_n)$ one simply obtains

$$g'(y_n) \approx g[y_n, x_n] + (y_n - x_n)g[y_n, x_n, x_n], \tag{7}$$

where $g[y_n, x_n, x_n] = (g[y_n, x_n] - g'(x_n))/(y_n - x_n)$. Simplifying (7) and using $q(y_n) = g'(y_n)$ gives

$$g'(y_n) = q(y_n) \approx 2g[y_n, x_n] - g'(x_n). \tag{8}$$

Now, we will use the technique which proposed before by Petković [6] and Petković et al [7] to approximate $g'(w_n)$, consider Hermite's interpolating polynomial of order 3

$$k(t) = c_1 + c_2(t - x_n) + c_3(t - x_n)^2 + c_4(t - x_n)^3, \tag{9}$$

where c_1, c_2, c_3, and c_4 need to be found. With the conditions

$g(x_n) = k(x_n), \quad g(y_n) = k(y_n), \quad g(w_n) = k(w_n), \quad g'(x_n) = k'(x_n)$, and by solving the system of linear equations resulted from the above conditions, we get

$$\begin{aligned}
c_1 &= g(x_n),\\
c_2 &= g'(x_n),\\
c_3 &= \frac{(x_n - w_n)g[w_n, y_n]}{(y_n - w_n)(x_n - y_n)} - \frac{(x_n - y_n)g[x_n, w_n]}{(y_n - w_n)(x_n - w_n)} - g'(x_n)\left(\frac{1}{w_n - x_n} + \frac{1}{y_n - x_n}\right),\\
c_4 &= \frac{g[x_n, w_n]}{(x_n - w_n)(y_n - w_n)} - \frac{g[y_n, x_n]}{(x_n - y_n)(y_n - w_n)} + \frac{g'(x_n)}{(x_n - w_n)(x_n - y_n)}.
\end{aligned}$$

Substituting these into Eq. (9) and using the approximation $g'(w_n) = k'(w_n)$, one can write

$$k'(w_n) = g[w_n, x_n]\left(2 + \frac{x_n - w_n}{y_n - w_n}\right) - \frac{(x_n - w_n)^2}{(x_n - y_n)(y_n - w_n)}g[x_n, y_n] + g'(x_n)\frac{y_n - w_n}{x_n - y_n} \approx g'(w_n). \tag{10}$$

Replacing $g'(y_n)$ and $g'(w_n)$ in Algorithm 3 and in Eq. (5) with the approximations (8) and (10) respectively, the following algorithm is obtained.

Algorithm 4: *Let x_0 be an initial guess of the solution of $g(x) = 0$. Then we can approximate x_{n+1} by the iterative method defined by:*

$$\begin{cases}
y_n = x_n - \dfrac{g(x_n)}{g'(x_n)},\\
w_n = y_n - \dfrac{g(y_n)}{q(y_n)} - \dfrac{2(g(y_n))^2 q(y_n) R(x_n, y_n)}{4(q(y_n))^4 - 4g(y_n)(q(y_n))^2 R(x_n, y_n) + (g(y_n))^2 (R(x_n, y_n))^2},\\
x_{n+1} = w_n - \dfrac{g(w_n)}{k'(w_n)}.
\end{cases} \tag{11}$$

We call the above scheme the third modified Halley's method MH3, which has convergence order equals eight as we will see in the next section. Each iteration in Algorithm 4 requires the evaluation of three functions, and one first derivative only. Based on the conjecture of Kung et al. [13], MH3 attains optimality and has efficiency index $(8)^{\frac{1}{4}} = 1.6818$.

3 Order of Convergence

In this section we establish the order of convergence of the presented method MH3 given by Algorithm 4 by using Mathematica codes to prove the order of convergence. Consider for the next theorem that α is a root of $g(x)$, and let $e_n = x_n - \alpha$ be the error at the n-th iteration. Using Taylor's series expansion of $g(x)$ about $x = \alpha$: $g(x_n) = g'(\alpha)\left[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + \cdots\right]$, where $c_k = \frac{1}{k!}\frac{g^{(k)}(\alpha)}{g'(\alpha)}$, $k = 2, 3, \ldots$ we have the following:

Theorem 1 *Let $\alpha \in A$ be a simple root of the function $g : A \subseteq \mathrm{R} \to \mathrm{R}$, where $g(x)$ is sufficiently differentiable in an open interval A. Let x_0 be an initial guess close enough to the root α. The proposed scheme given by (11) has at least eighth-order of convergence.*

Proof. The following Mathematica code proves the theorem

In [1]:= g[e_]:= dg[α] (e+c $_2$e ^{2}+c $_3$e ^{3}+c $_4$e ^{4}); (*g(x) series with dg[α]=$g'(\alpha)$ *).

In [2]:= g[x_,y_]:= $\frac{\mathrm{g[x]-g[y]}}{\mathrm{x-y}}$; (*This is the finite difference*).

In [3]:= q[x_,y_]:= 2 g[x,y]-g ′[x]; (*$g'(y)$ approximation *).

In [4]:= R[x_,y_]:= (3$\frac{\mathrm{g[y]-g[x]}}{\mathrm{y-x}}$-2q[x,y]-g ′[y])$\frac{2}{\mathrm{x-y}}$; (*Second derivative approximation*).

In [5]:= k[x_,y_,w_]:= g[w,x] (2+$\frac{\mathrm{w-x}}{\mathrm{w-y}}$)-$\frac{\mathrm{(w-x)^2}}{\mathrm{(w-y)(y-x)}}$g[x,y]+g’[x] $\frac{\mathrm{w-y}}{\mathrm{y-x}}$;(*$g'(w)$ approximation*).

In [6]:= y =e-Series[$\frac{\mathrm{g[e]}}{\mathrm{g'[e]}}$,{e,0,8}]; (*First step of Alg. 4 *).

In [7]:= w =y-$\frac{\mathrm{g[y]}}{\mathrm{q[e,y]}} - \frac{\mathrm{2(g[y])^2q[e,y]R[e,y]}}{\mathrm{4(q[e,y])^4 - 4g[y](q[e,y])^2R[e,y] + (g[y])^2(R[e,y])^2}}$;(*Second step of Alg. 4 *).

In [8]:= e $_{n+1}$= w-$\frac{\mathrm{g[w]}}{\mathrm{k[e,y,w]}}$// FullSimplify (* Third step of Alg. 4 *).

Out [8]:= c $_2^2$c $_3$(c $_2$c $_3$-c $_4$)e ^{8}+O[e ^{9}]

Hence, MH3 technique given by Algorithm 4 is of eighth-order of convergence.

4 Applications and Numerical Examples

To show the efficiency of the new optimal eighth-order method MH3, several examples will be tested including some chemical engineering problems. Comparison will be done against the following schemes of optimal eighth-order of convergence: the method proposed by Kung et al. [13], the method presented by Cordero et al. [2] with $\beta = 1$, the second case of the first family with $\beta = 1$ proposed by Sharma et al. [19], the method presented by Behl et al. [21] with $\beta = 1$, and the special case 2 with $\beta = 1$ from the method presented by Behl et al. [18]. We denote the methods by the following abbreviations respectively: KT, CLMT, SA, BGMM, and BAM.

We consider $|x_n - x_{n-1}| < 10^{-30}$ and $|f(x_n) - f(x_{n-1})| < 10^{-30}$ at the same time as a stopping criterion of the computer programs. Mathematica 9 was used to carry out all computations with 10000 significant digits.

Tabs. 1–5 illustrate the comparisons between the iterative methods, where n indicates number of the iterations such that the stopping criterion is affirmed, x_n is the approximate root, $|x_n - x_{n-1}|$ is the absolute difference between two successive approximations of the root such that $|x_n - x_{n-1}| < 10^{-30}$ and $|f(x_n) - f(x_{n-1})| < 10^{-30}$, $f(x_n)$ is the value of the approximate root, the approximated computational order of convergence (ACOC) given by Cordero et al. [24], which can be estimated as follows

$\text{ACOC} \approx \dfrac{\ln|(x_{n+1} - x_n)/(x_n - x_{n-1})|}{\ln|(x_n - x_{n-1})/(x_{n-1} - x_{n-2})|}$,and finally, the time in seconds required to satisfy the stopping criterion using the built-in function "TimeUsed" in Mathematica 9 software. All calculations have been performed under the same conditions on Intel Core i7-3770 CPU @3.40 GHz with 4GB RAM, with Microsoft Windows 10, 64 bit based on X64-based processor.

Consider the following test examples:

Example 1 (A chemical equilibrium problem) Consider the equation from [25] which describes the fraction of the nitrogen-hydrogen feed that gets converted to ammonia (this fraction is called fractional conversion). Also, consider that we have pressure of 250 atm and temperature of 500°C, the original problem consists of finding the root of the function

$f_1(x) = \dfrac{8(4-x)^2 x^2}{(6-3x)^2(2-x)} - 0.186$, which can be reduced in polynomial form as:

$$f_1(x) = x^4 - 7.79075x^3 + 14.7445x^2 + 2.511x - 1.674.$$

The four roots of this function are: $x_1 = 0.27776, x_2 = -0.384094, x_3 = 3.94854 + 0.316124i$ and $x_4 = 3.94854 + 0.316124i$. By the definition, the factional conversion must be between 0 and 1. So, only the first real root $x_1 = 0.27776$ is acceptable and physically meaningful. We started by $x_0 = 0.3$ as an initial guess. The results are concluded in Tab. 1.

Table 1: Comparisons between different methods on test function $f_1(x)$

Method	n	x_n	$\|x_n - x_{n-1}\|$	$\|f(x_n)\|$	ACOC	CPU Time
			$f_1(x);\ x_0 = 0.3$			
KT	3	0.27775954284172066	5.82E-96	2.41E-760	8	0.171
CLMT	3	0.27775954284172066	1.72E-103	1.87E-821	8	0.203
SA	3	0.27775954284172066	6.38E-67	2.21E-524	8	0.203
BGMM	3	0.27775954284172066	9.36E-70	2.17E-547	8	0.422
BAM	3	0.27775954284172066	3.81E-96	7.95E-762	8	0.203
MH3	3	0.27775954284172066	3.41E-109	9.49E-868	8	0.188

Example 2 (Azeotropic point of a binary solution) Consider the problem obtained by Shacham et al. [26] to determine the azeotropic point of a binary solution:

$$f_2(x) = \frac{AB\left[B\left(1-x\right)^2 - Ax^2\right]}{\left[x(A-B)+B\right]^2} + 0.14845,$$

where A and B are coefficients in the Van Laar equation which describes phase equilibria of liquid solutions. Consider for this problem that $A = 0.38969$ and $B = 0.55954$.

The root of this equation is $x = 0.6914737357$. We took the initial approximation $x_0 = 1$. See Tab. 2 for the results and comparisons.

Table 2: Comparisons between different methods on test function $f_2(x)$

Method	n	x_n	$\|x_n - x_{n-1}\|$	$\|f(x_n)\|$	ACOC	CPU Time
			$f_2(x);\ x_0 = 1$			
KT	3	0.69147373574714142	7.41E-48	3.05E-379	8	0.187
CLMT	3	0.69147373574714142	1.74E-56	1.87E-449	8	0.234
SA	3	0.69147373574714142	7.73E-83	6.19E-665	8	0.172
BGMM	3	0.69147373574714142	4.61E-76	1.77E-609	8	0.453
BAM	3	0.69147373574714142	1.38E-44	1.36E-352	8	0.188
MH3	3	0.69147373574714142	8.37E-54	7.36E-428	8	0.171

Example 3 (Conversion in a chemical reactor) In this example from [27], the following nonlinear equation is to be solved

$$f_3(x) = \frac{x}{1-x} - 5\ln\left[\frac{0.4(1-x)}{0.4-0.5x}\right] + 4.45977,$$

where x is the fractional conversion of species in a chemical reactor. Therefore, x should be bounded between 0 and 1.

The solution of this equation is $x = 0.7573962463$. As an initial solution, we selected $x_0 = 0.77$. Check the results in Tab. 3.

Table 3: Comparisons between different methods on test function $f_3(x)$

Method	n	x_n	$\|x_n - x_{n-1}\|$	$\|f(x_n)\|$	ACOC	CPU Time
			$f_3(x);\ x_0 = 0.77$			
KT	3	0.75739624625375388	6.61E-47	6.39E-360	8	0.312
CLMT	3	0.75739624625375388	6.24E-53	2.86E-409	8	0.765
SA	4	0.75739624625375388	1.75E-106	1.08E-831	8	0.280
BGMM	4	0.75739624625375388	5.99E-173	6.79E-1363	8	7.484
BAM	3	0.75739624625375388	4.82E-42	2.29E-320	8	0.312
MH3	3	0.75739624625375388	2.37E-48	2.79E-372	8	0.264

Example 4 (Volume from Van Der Waals equation) Van Der Waals' equation is given by

$$\left(p + \frac{n^2 a}{V^2}\right)(V - nb) = nRT,$$

where p, V, T, n are the pressure, volume, temperature in Kelvin and number of moles of the gas. R is the gas constant equals 0.0820578. Finally, a and b are called Van Der Waals constants and they depend on the gas type. Its clear that the above equation is nonlinear in V. It can be reduced to the following function of V.

$$f(V) = pV^3 - n(RT + bp)V^2 + n^2aV - n^3ab.$$

For instance, if one has to find the volume of 1.4 moles of benzene vapor under pressure of 40 atm and temperature of 500°C, given that Van Der Waals constants for benzene are $a = 18$ and $b = 0.1154$, then the problem arises is to find roots of this polynomial

$$f_4(x) = 40x^3 - 95.26535116x^2 + 35.28x - 5.6998368.$$

The above equation has three roots: $x = 1.97078$ and $x = 0.205425 \pm 0.173507i$. As V is a volume, therefore only the positive real roots are physically meaningful, that is the first root. We considered the initial approximation $x_0 = 2$ for this problem. The results and comparisons are concluded in Tab. 4.

Table 4: Comparisons between different methods on test function $f_4(x)$

Method	n	x_n	$\lvert x_n - x_{n-1}\rvert$	$\lvert f(x_n)\rvert$	ACOC	CPU Time
			$f_4(x);\ x_0 = 2$			
KT	3	1.9707842194070294	2.41E-89	2.11E-706	8	0.171
CLMT	3	1.9707842194070294	1.10E-98	3.31E-782	8	0.171
SA	3	1.9707842194070294	3.43E-40	2.07E-305	8	0.187
BGMM	3	1.9707842194070294	3.86E-44	1.35E-337	8	0.312
BAM	3	1.9707842194070294	2.83E-88	1.04E-697	8	0.187
MH3	3	1.9707842194070294	7.22E-107	1.32E-848	8	0.156

Example 5 To study the proposed method on some nonlinear functions, consider the following six test functions:

$$f_5(x) = (x - 1)^3 - 1, \quad f_6(x) = x^3 - 10, \quad f_7(x) = \cos(x) - x,$$
$$f_8(x) = 1 - x^2 + \sin^2(x), \quad f_9(x) = (2 + x)e^x - 1,$$
$$f_{10}(x) = \ln(x^2 - x + 1) - 4\sin(x - 1).$$

Comparisons' results of Example 5 are presented in Tab. 5.

It is clear from Tabs. 1–5 that MH3 needs less iterations to satisfy the stopping criterion than the other tested methods, or in some cases it needs the same number of iterations. Based on the numerical experiments, the iterative scheme given by MH3 is comparable to the tested schemes of equal order. Note that even if MH3 has the same number of iterations needed to satisfy the convergence criterion, it is still superior to the other schemes considered in this study since $|x_n - x_{n-1}|$ and $f(x_n)$ are less for MH3 than the other tested methods of the same order. Also, in the last column of Tabs. 1–5, the CPU time required to satisfy the convergence condition of MH3 is less in nine out of 10 functions than that of the other tested methods. Overall, based on either the number of iterations or CPU time needed to satisfy the convergence criterion, the new method would be preferable as compared to the tested methods.

For the test functions in Example 5, we test another convergence condition, that is number of required iterations such that $|x_n - x_{n-1}| < 10^{-200}$. It is obvious from Tab. 6 that MH3 requires number of iterations which is fewer or equal to the number needed by the tested methods of equal order of convergence to satisfy the convergence criterion. Overall, MH3 is comparable to the other tested methods if we want to take in account the accuracy of the approximate zero with the CPU time needed to satisfy the stopping criterion.

Table 5: Comparisons between different methods on test functions $f_5(x) - f_{10}(x)$

Method	n	x_n	$\|x_n - x_{n-1}\|$	$\|f(x_n)\|$	ACOC	CPU Time
$f_5(x);\ x_0 = 2.5$						
KT	4	2	6.26E-188	3.94E-1497	8	0.187
CLMT	4	2	2.50E-241	2.06E-1925	8	0.156
SA	4	2	9.54E-105	5.85E-830	8	0.203
BGMM	4	2	3.50E-131	5.28E-1042	8	0.203
BAM	4	2	2.16E-185	1.13E-1476	8	0.172
MH3	3	2	4.68E-32	7.73E-252	8	0.156
$f_6(x);\ x_0 = 2$						
KT	3	2.1544346900318837	3.63E-65	1.09E-516	8	0.171
CLMT	3	2.1544346900318837	1.21E-75	1.30E-601	8	0.188
SA	4	2.1544346900318837	5.84E-175	4.83E-1391	8	0.188
BGMM	4	2.1544346900318837	3.10E-236	6.70E-1882	8	0.218
BAM	3	2.1544346900318837	2.62E-63	1.16E-501	8	0.203
MH3	3	2.1544346900318837	1.56E-81	2.55E-649	8	0.156
$f_7(x);\ x_0 = 1.7$						
KT	3	0.7390851332151606	4.05E-46	1.06E-366	8	0.281
CLMT	3	0.7390851332151606	2.55E-52	1.21E-417	8	0.281
SA	3	0.7390851332151606	2.94E-34	7.85E-273	8	0.281
BGMM	3	0.7390851332151606	4.47E-42	2.58E-363	8	1.313
BAM	3	0.7390851332151606	1.22E-47	5.96E-379	8	0.329
MH3	3	0.7390851332151606	4.13E-53	2.35E-424	8	0.249
$f_8(x);\ x_0 = 1$						
KT	4	1.4044916482153412	2.97E-123	2.04E-980	8	0.281
CLMT	4	1.4044916482153412	2.34E-226	2.05E-1806	8	0.390
SA	3	1.4044916482153412	2.21E-32	2.30E-253	8	0.265
BGMM	3	1.4044916482153412	8.57E-40	5.22E-314	8	1.657
BAM	4	1.4044916482153412	8.75E-92	1.46E-728	8	0.374
MH3	3	1.4044916482153412	6.83E-38	1.23E-299	8	0.250
$f_9(x);\ x_0 = 0.5$						
KT	3	0.4428544010023886	2.74E-85	1.24E-677	8	0.313
CLMT	3	0.4428544010023886	2.30E-95	2.55E-759	8	0.328
SA	3	0.4428544010023886	2.47E-79	2.76E-629	8	0.328
BGMM	3	0.4428544010023886	1.83E-88	2.48E-703	8	5.126
BAM	3	0.4428544010023886	1.59E-82	3.06E-655	8	0.454
MH3	3	0.4428544010023886	2.57E-96	5.13E-767	8	0.313

(Continued)

Table 5 (continued).

Method	n	x_n	$\lvert x_n - x_{n-1}\rvert$	$\lvert f(x_n)\rvert$	ACOC	CPU Time
			$f_{10}(x); x_0 = 1.5$			
KT	3	1	1.78E-42	5.04E-338	8	0.282
CLMT	3	1	2.53E-47	3.47E-377	8	0.282
SA	3	1	5.25E-48	2.07E-381	8	0.344
BGMM	3	1	6.53E-51	5.40E-405	8	4.359
BAM	3	1	5.28E-42	4.30E-334	8	0.328
MH3	3	1	1.80E-54	2.40E-487	9	0.281

Table 6: Comparisons between different methods such that $\lvert x_n - x_{n-1}\rvert < 10^{-200}$

Method	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
	$x_0 = 2.5$	$x_0 = 2$	$x_0 = 1.7$	$x_0 = 1$	$x_0 = -0.5$	$x_0 = 1.5$
KT	5	4	4	5	4	4
CLMT	4	4	4	4	4	4
SA	5	5	4	4	4	4
BGMM	4	4	4	4	4	4
BAM	5	4	4	5	4	4
MH3	4	4	4	4	4	4

5 Conclusion

A new optimal root finding scheme for nonlinear equations has been established in this work. The optimality of the proposed method was reached by using composition technique with Hermite's polynomial and finite differences. The software Mathematica has been used to show that the optimal technique is convergent with convergence order equals eight. Several numerical examples with four real life problems from the field of chemical engineering were examined, demonstrating the strength of the proposed method. Overall, the implemented method is comparable to the tested iterative schemes of equal order of convergence.

Funding Statement: We are grateful for the financial support from UKM's research Grant GUP-2019-033.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. F. Traub, *Iterative Methods for the Solution of Equations*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1964.

[2] A. Cordero, T. Lotfi, K. Mahdiani and J. R. Torregrosa, "A stable family with high order of convergence for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 254, pp. 240–251, 2015.

[3] A. Kumar, P. Maroju, R. Behl, D. K. Gupta and S. S. Motsa, "A family of higher order iterations free from second derivative for nonlinear equations in R," *Journal of Computational and Applied Mathematics*, vol. 330, pp. 676–694, 2018.

[4] O. S. Solaiman and I. Hashim, “Efficacy of optimal methods for nonlinear equations with chemical engineering applications,” *Mathematical Problems in Engineering*, vol. 2019, pp. 11, 2019.

[5] S. Sharifi, M. Salimi, S. Siegmund and T. Lotfi, “A new class of optimal four-point methods with convergence order 16 for solving nonlinear equations,” *Mathematics and Computers in Simulation*, vol. 119, pp. 69–90, 2016.

[6] M. S. Petković, “On a general class of multipoint root-finding methods of high computational efficiency,” *SIAM Journal on Numerical Analysis*, vol. 47, no. 6, pp. 4402–4414, 2010.

[7] M. S. Petković, B. Neta, L. D. Petković and J. Džunić, “Multipoint methods for solving nonlinear equations: a survey, ” in *Applied Mathematics and Computation*, vol. 226, Elsevier Inc., pp. 635–660, 2014.

[8] A. Cordero and J. R. Torregrosa, “On the design of optimal iterative methods for solving nonlinear equations, ” in *Advances in Iterative Methods for Nonlinear Equations, SEMA SIMAI Springer Series*, S. Amat, S. Busquier, 10. New York City, NY: Springer International Publishing, pp. 79–111, 2016.

[9] E. Halley, “A new; exact and easy method of finding the roots of equations generally and that without any previous reduction,” *Philosophical Transactions of the Royal Society*, vol. 18, no. 210, pp. 136–148, 1694.

[10] K. I. Noor and M. A. Noor, “Predictor-corrector Halley method for nonlinear equations,” *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1587–1591, 2007.

[11] M. A. Noor, W. A. Khan and A. Hussain, “A new modified Halley method without second derivatives for nonlinear equation,” *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1268–1273, 2007.

[12] O. S. Solaiman and I. Hashim, “Two new efficient sixth order iterative methods for solving nonlinear equations,” *Journal of King Saud University—Science*, vol. 31, no. 4, pp. 701–705, 2019.

[13] H. T. Kung and J. F. Traub, “Optimal order of one-point and multipoint iteration,” *Journal of the ACM*, vol. 21, no. 4, pp. 643–651, 1974.

[14] C. Chun, “Some fourth-order iterative methods for solving nonlinear equations,” *Applied Mathematics and Computation*, vol. 195, no. 2, pp. 454–459, 2008.

[15] R. Sharma and A. Bahl, “An optimal fourth order iterative method for solving nonlinear equations and its dynamics,” *Journal of Complex Analysis*, vol. 2015, no. 8, pp. 1–9, 2015.

[16] R. Behl, P. Maroju and S. S. Motsa, “A family of second derivative free fourth order continuation method for solving nonlinear equations,” *Journal of Computational and Applied Mathematics*, vol. 318, pp. 38–46, 2017.

[17] O. S. Solaiman, S. A. A. Karim and I. Hashim, “Optimal fourth- and eighth-order of convergence derivative-free modifications of King’s method,” *Journal of King Saud University—Science*, vol. 31, no. 4, pp. 1499–1504, 2019.

[18] R. Behl, I. K. Argyros and S. S. Motsa, “A new highly efficient and optimal family of eighth-order methods for solving nonlinear equations,” *Applied Mathematics and Computation*, vol. 282, pp. 175–186, 2016.

[19] J. R. Sharma and H. Arora, “Some novel optimal eighth order derivative-free root solvers and their basins of attraction,” *Applied Mathematics and Computation*, vol. 284, pp. 149–161, 2016.

[20] G. Matthies, M. Salimi, S. Sharifi and J. L. Varona, “An optimal three-point eighth-order iterative method without memory for solving nonlinear equations with its dynamics,” *Japan Journal of Industrial and Applied Mathematics*, vol. 33, no. 3, pp. 751–766, 2016.

[21] R. Behl, D. González, P. Maroju and S. S. Motsa, “An optimal and efficient general eighth-order derivative free scheme for simple roots,” *Journal of Computational and Applied Mathematics*, vol. 330, pp. 666–675, 2018.

[22] Y. H. Geum, Y. I. Kim and B. Neta, “Constructing a family of optimal eighth-order modified Newton-type multiple-zero finders along with the dynamics behind their purely imaginary extraneous fixed points,” *Journal of Computational and Applied Mathematics*, vol. 333, pp. 131–156, 2018.

[23] C. Chun and B. Neta, “Comparison of several families of optimal eighth order methods,” *Applied Mathematics and Computation*, vol. 274, pp. 762–773, 2016.

[24] A. Cordero and J. R. Torregrosa, “Variants of Newton’s method using fifth-order quadrature formulas,” *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 686–698, 2007.

[25] G. V. Balaji and J. D. Seader, “Application of interval Newton’s method to chemical engineering problems,” *Reliable Computing*, vol. 1, no. 3, pp. 215–223, 1995.

[26] M. Shacham and E. Kehat, “An iteration method with memory for the solution of a non-linear equation,” *Chemical Engineering Science*, vol. 27, no. 11, pp. 2099–2101, 1972.

[27] M. Shacham, “Numerical solution of constrained non-linear algebraic equations,” *International Journal for Numerical Methods in Engineering*, vol. 23, no. 8, pp. 1455–1481, 1986.