Global Analysis and Discrete Mathematics
Volume 8, Issue 1, pp. 1–10
ISSN: 2476-5341 *Research Article*

Optimal Parameterised Families of Modified Householder's Method with and without Restraint on Function Derivative

Oghovese Ogbereyivwe · Shehu Salisu Umar

Received: 24 May 2023/ Accepted: 24 July 2023

Abstract This paper introduces two families of modified Householder's method (HM) that are optimal in line with Kung-Traub conjecture given in [4]. The modification techniques employed involved approximation of the function derivatives in the HM with divided difference operator, a polynomial function approximation and the modified Wu function approximation in [17]. These informed the formation of two families of methods that that are optimal and do not or require function derivative evaluation. The both families do not breakdown when $f(\cdot) \approx 0$ as in the case with the HM and many existing modified HM. From the convergence investigation carried out on the methods, the sequence of approximations produced by the methods, converged to solution of nonlinear equation with order four. The implementation of the methods was illustrated and numerical results obtained were compared with that of some recently developed methods.

Keywords Iterative method · Householder method · Derivative free · Optimal Order

Mathematics Subject Classification (2010) 65Hxx · 65H05

O. Ogbereyivwe
Department of Mathematics, Delta State University of Science & Tech., Ozoro, Nigeria.
Tel.: +234-0706-494-3779
E-mail: ogho2015@gmail.com, ogbereyivweo@dsust.edu.ng

S. S. Umar
Department of Statistics, Federal Polytechnic, Auchi, Nigeria.
Tel.: +234-0806-366-1595
E-mail: umarss83@yahoo.com

1 Introduction

The Householder method (HM) in [2] is one of the traditional iterative methods for obtaining solution s_0 of a nonlinear (NL) equation $f(s) = 0$. It is given as

$$s_{j+1} = s_j - \eta_j \left(1 + \frac{\eta_j}{2} \frac{f''(s_j)}{f'(s_j)} \right), \quad j = 0, 1, 2, \ldots, \tag{1}$$

where $\eta_j = \frac{f(s_j)}{f'(s_j)}$. Starting with initial guess s^*, the HM iteratively generates sequence of approximations that converges to the solution s_0 of a NL equation with convergence order (CO) three. Some major setbacks of the HM includes:

(i) its involvement of second derivative ($f''(s_j)$) evaluation,

(ii) failure of the method when $f'(\cdot) \approx 0$ or $f'(\cdot) = 0$ and,

(iii) non-optimal in the sense of Kung-Traub conjecture in [4].

The Kung-Traub conjecture [4] posits that, an IM without memory that require n distinct function evaluation in one iteration cycle, is optimal if it attain CO $\nu = 2^{n-1}$.

Sequel to the setbacks itemized in (i)-(iii) above, many researchers have put forward modifications of the HM with the motivation of dealing with them. For instance, in the work [1, 6–8, 13, 16], modifications of HM were presented that resolved problem (i) with no recourse to resolving problems (ii) and (iii). While the authors in [11] put forward family of methods with reduced number of function derivative evaluation $f'(\cdot)$ at two different points from two to one point, they failed to explicitly eliminate the setback in (ii). In [5, 15], the authors considered modifications of the HM that have the advantages of resolving the setbacks (i) and (iii), but failed to deal with setback (ii) also. The modifications of the HM that have the advantages of resolving the setbacks itemised in (i)-(iii) above are scarcely available. This is the main motivation of this work.

From the foregoing, two families of modified HM that does not require second derivative of function, fail when $f(\cdot) \approx 0$ and optimal are developed in this manuscript. The approach used in the methods development, involved the use of the divided difference operator, modified Wu function approximation and a newly introduced polynomial approximation of function derivative.

This manuscript is structured in the following order: Section 1 contains the review of related literature, while Section 2 presents techniques employed in the methods development. The test for methods convergence is given in Section 3. Section 4 put forward results of numerical implementation of the developed and compared methods. The last section of the manuscript contains conclusion.

2 Method Formulation

We acknowledge the traditional HM [2] put forward as in (1). The HM generates sequence of approximations that converges to solution of NL equations when implemented. However, the presence of the first and second derivatives in its iterative procedure, hinders its practical utilisation. This is because the method collapses when the derivatives $f'(\cdot) \approx 0$ and evaluation of the second derivative of function incurs additional cost to the iterative process.
To circumvent the presence of second derivative in the HM, Noor and Gupta in [8] set $s_j = y_j$ and approximates $f''(s_j)$ as

$$f''(s_j) \approx \frac{f'(y_j) - f'(s_j)}{y_j - s_j} = G\left(s, y\right), \tag{2}$$

and then obtained a fourth order convergence method as:

$$s_{j+1} = s_j - \eta_j \left(1 + \frac{\eta_j}{2} \frac{G\left(s, y\right)}{f'(s_j)}\right). \tag{3}$$

Although the method in (3) successfully modified HM to a method that does not require second derivative, its efficiency index (EI) is far less than that of the HM. The efficiency of Iterative Algorithms for solving NL equations, is measured using the Ostrowski efficiency index [14] given as $\nu^{\frac{1}{n}}$. Thus, method (3) has $EI \approx 1.4142$ while HM is $EI \approx 1.4417$.
Further, the method (3) fails or breakdown whenever $f'(\cdot) \approx 0$. To eliminate the problem of breakdown of the iterative process and also make the method optimal as conjectured by Kung-Traub in [4], we used an obtained polynomial function approximation $P(s, t)$ to approximate $f'(y_j)$ as

$$f'(y_j) \approx f'(s_j)\left[1 - 2t + t^2\right] = P\left(s, t\right), \tag{4}$$

where $t = \frac{f(y_j)}{f(s_j)}$ and the Wu approximation for $f'(s_j)$ in [17] that is given as

$$f'(s_j) \approx f'(s_j) + \delta f(s_j), \qquad \delta \in (-1, 1) - \{0\}. \tag{5}$$

Define the real-valued (RV) function $Q\left(s, t\right)$ (a modified Wu second iteration stage approximation for $f'(y_j)$) as

$$Q\left(s, t\right) = \left[P\left(s, t\right) + \alpha f(s_j)\right] + \delta f(y_j), \tag{6}$$

then a family of modified HM is put forward next.

Algorithm 1 *Suppose s^* an initial guess, then the solution s_{j+1} of NL equation can be obtained using the iterative procedure*

$$\begin{aligned} y_j &= s_j - \frac{f(s_j)}{f'(s_j) + \delta f(s_j)}, \\ s_{j+1} &= y_j - \frac{f(y_j)}{Q(s,t)}\left[1 + \frac{1}{2}\frac{f(y_j)}{Q(s,t)}\left(\frac{(f'(s_j) + \delta f(s_j)) - Q(s,t)}{f(s_j)}\right)\right]. \end{aligned} \tag{7}$$

To put forward the derivative free version of Algorithm 1, consider the approximation of the derivative $f'(s_j)$ as

$$f'(s_j) \approx f\left[s_j, \omega_j\right] + \delta f(s_j), \quad \omega_j = s_j + \delta\left(f(s_j)\right)^m, \tag{8}$$

where $m \geq 2$ and $f\left[\cdot,\cdot\right]$ a divided difference operator. By the definition

$$f'(y_j) \approx \left(f\left[s_j, \omega_j\right] + \delta f(s_j)\right)\left[1 - 2t + t^2\right] = R\left(s,t\right), \tag{9}$$

and the RV functions $\varPhi\left(s,t\right)$ defined as:

$$\varPhi\left(s,t\right) = \left[R\left(\alpha, s\right) + \alpha f(s_j)\right] + \delta f(y_j), \tag{10}$$

a new family of derivative free modified HM is provided in Algorithm 2.

Algorithm 2 *Suppose s^* an initial guess and $m = 3$, then the solution s_{j+1} of NL equation can be obtained using the iterative procedure*

$$\begin{aligned} y_j &= s_j - \frac{f(s_j)}{f[s_j, \omega_j] + \delta f(s_j)}, \\ s_{j+1} &= y_j - \frac{f(y_j)}{\varPhi(s,t)}\left[1 + \frac{1}{2}\frac{f(y_j)}{\varPhi(s,t)}\left(\frac{(f[s_j, \omega_j] + \delta f(s_j)) - \varPhi(s,t)}{f(s_j)}\right)\right], \end{aligned} \tag{11}$$

3 Convergence Analysis

To prove the convergence of Algorithm 1 and Algorithm 2, suffice to deriving an equation in the form $D_{j+1} = \phi D_j^{\nu} + O(D_j^{\nu+1})$ (where the error at jth iteration point is $D_j = s_j - s_0$) from the Algorithms by the use of the Taylor series expansion of the functions $f(s)$ and $f'(s)$. When this equation is derived, then ν is referred to as the Algorithm CO. For more details on this technique see [9,10,12].

Theorem 1 *Consider a sufficiently differentiable scalar function $f : \varOmega \subset \Re \rightarrow \Re$ in domain $\varOmega$ with a simple solution s_0. Then, for s^* (an initial guess) close to s_0 and utilised in Algorithm 1 implementation, a sequence $\{s_j\}_{j\geq 0}, (s_j \in D)$ of approximations of s_0 will be generated that converges to s_0 with CO four, provided the free parameters $\alpha, \delta \in (-1, 1) - \{0\}$ are equal.*

Proof . Consider replacing s with s_j in Taylor expansion of $f'\left(s\right)$ and $f\left(s\right)$ about s_0, the expansions are obtained next.

$$f(s_j) = f'(s_0)\left[D_j + \sum_{m=2}^{4} c_m D_j^m + O(D_j^5)\right], \quad j{=}0, 1, 2, \ldots \tag{12}$$

and

$$f'(s_j) = f'(s_0)\left[1 + \sum_{m=2}^{4} m c_m D_j^{m-1} + O(D_j^5)\right], \quad j{=}0, 1, 2, \ldots \tag{13}$$

where $c_m = \frac{1}{m!}\frac{f^m(s_0)}{f'(s_0)}$, $m \geq 2$.
From the results in (12) and (13), the next expression for y is obtained.

$$\begin{aligned} y_j =& s_j - \frac{f(s_j)}{f'(s_j) + \delta f(s_j)} = \left(\delta + c_2\right) D_j^2 \\ &+ \left(-\delta^2 - 2\delta c_2 - 2c_2^2 + 2c_3\right) D_j^3 \\ &+ \left(\delta^3 + 3\delta^2 c_2 + 5\delta c_2^2 + 4c_2^3 - 4\delta c_3 - 7c_2c_3 + 3c_4\right) D_j^4 + O\left(D_j^5\right). \end{aligned} \tag{14}$$

Using (14), the Taylor expansion for $f(y_j)$ is obtained as

$$\begin{aligned} f(y_j) =& \left(\delta + c_2\right) D_j^2 + \left(-\delta^2 - 2\delta c_2 - 2c_2^2 + 2c_3\right) D_j^3 \\ &+ \Big(\delta^3 + 3\delta^2 c_2 + 5\delta c_2^2 + 4c_2^3 + c_2\left(\delta + c_2\right)^2 - 4\delta c_3 - 7c_2c_3 + 3c_4\Big) D_j^4 \\ &+ O\left(D_j^5\right). \end{aligned} \tag{15}$$

From (15) and (12), the expansion for t_j is obtained next.

$$\begin{aligned} t_j =& \frac{f(y_j)}{f(s_j)} = \left(\delta + c_2\right) D_j + \left(-\delta^2 - 3\delta c_2 - 3c_2^2 + 2c_3\right) D_j^2 \\ &+ \left(\delta^3 + 5\delta^2 c_2 + 10\delta c_2^2 + 8c_2^3 - 5\delta c_3 - 10c_2c_3 + 3c_4\right) D_j^3 + O\left(D_j^4\right). \end{aligned} \tag{16}$$

From (16), we obtained the expansion for $P(s,t)$ as

$$\begin{aligned} P(s,t) =& 1 - \delta D_j + (\delta^2 + 3\delta c_2 + 3c_2^2 - c_3)\delta^2 \\ &+ (-\delta^3 - 6\delta^2 c_2 - 11\delta c_2^2 - 8c_2^3 + 5\delta c_3 + 10c_2c_3 - 2c_4)D_j^3 \\ &+ (\delta^4 + 9\delta^3 c_2 + 21c_2^4 + \delta^2(25c_2^2 - 7c_3) - 37c_2^2c_3 + 8c_3^2 + 14c_2c_4 \\ &+ \delta(35c_2^3 - 32c_2c_3 + 7c_4) - 3c_5)D_j^4 + O\left(D_j^4\right). \end{aligned} \tag{17}$$

By combining (15) and (17), the expansion for $Q(s,t)$ is obtained next as

$$\begin{aligned} &Q(s,t) = [P(s,t) + \alpha f(s)] + \delta f(y_j) = 1 + (\alpha - \delta)D_j \\ &+ (2\delta^2 + \alpha c_2 + 4\delta c_2 + 3c_2^2 - c_3)D_j^2 \\ &+ (-2\delta^3 - 8\delta^2 c_2 - 13\delta c_2^2 - 8c_2^3 + \alpha c_3 + 7\delta c_3 + 7\delta c_3 + 10c_2c_3 - 2c_4)D_j^3 \\ &+ (2\delta^4 + 13\delta^3 c_2 + 21c_2^4 + \delta(32c_2^2 - 11c_3) - 37c_2^2c_3 + 8c_3^2 + \alpha c_4 + 14c_2c_4 \\ &+ \delta(40c_2^3 - 39c_2c_3 + 10c_4) - 3c_5)D_j^4 + O(D_j^5). \end{aligned} \tag{18}$$

From (12), (13), (14), (15) and (18), we have:

$$\begin{aligned} s_{j+1} =& y_j - \frac{f(y_j)}{Q(\alpha,s)}\left[1 + \frac{1}{2}\frac{f(y_j)}{Q(\alpha,s)}\left(\frac{(f'(s_j) + \delta f(s_j)) - Q(\alpha,s)}{f(s_j)}\right)\right] \\ &= s_0 + (\alpha - \delta)(\delta + c_2)D_j^3 \\ &+ (\delta^3 + 3\delta^2 c_2 + 4\delta c_2^2 + c_2^3 - \alpha(\delta + c_2) - \delta c_3 - c_2c_3 \\ &+ \frac{\alpha}{2}(3\delta^2 + 4\delta c_2 + 4c_3))D_j^4 + O(D_j^5). \end{aligned} \tag{19}$$

Our desire is to reduce the coefficients of $D_j^n,\ n = 2, 3$ to zero. This is achievable when the parameters α and δ are equal. That is $\alpha = \delta$. Consequently, (19) is reduced to

$$s_{j+1} = s_0 + \left(\frac{(\delta + c_2)(3\delta^2 + 5\delta c_2 + 2c_2^2 - 2c_3)}{2} \right) D_j^4 + O(D_j^5). \tag{20}$$

But $D_{j+1} = s_{j+1} - s_0$. Therefore, from (20) we have

$$D_{j+1} = \left(\frac{(\delta + c_2)(3\delta^2 + 5\delta c_2 + 2c_2^2 - 2c_3)}{2} \right) D_j^4 + O(D_j^5). \tag{21}$$

By comparing the error expression in (21) with the general error equation $D_{j+1} = \phi D_j^{\nu} + O(D_j^{\nu+1})$, we conclude that the CO (ν) of Algorithm 1 is four. This completes the proof.

3.1 Algorithm 1 concrete member

For $\delta = 0.001$, a concrete member that falls under Algorithm 1 is as following:

Algorithm 3 *Suppose s_0 an initial guess, then the solution s_{k+1} of NL equation can be obtained using the iterative procedure*

$$\begin{aligned} y_j &= s_j - \frac{f(s_j)}{f'(s_j) - 0.001f(s_j)}, \\ s_{j+1} &= y_j - \frac{f(y_j)}{Q(s,t)} \left[1 + \frac{1}{2} \frac{f(y_j)}{Q(s,t)} \left(\frac{(f'(s_j) - 0.001f(s_j)) - Q(s,t)}{f(s_j)} \right) \right]. \end{aligned} \tag{22}$$

Theorem 2 *Consider a sufficiently differentiable scalar function $f : \Omega \subset \Re \to \Re$ in domain Ω with a simple solution s_0. Then, for s^* (an initial guess) close to s_0 and utilised in Algorithm 2 implementation, a sequence $\{s_j\}_{j\geq 0}, (s_j \in D)$ of approximations of s_0 will be generated that converges to s_0 with CO four, provided the free parameters $\alpha, \delta \in (-1, 1) - \{0\}$ are equal.*

Proof . The proof follows same procedures used in the proof of Theorem 1. Consequently, the error equation is obtained as:

$$s_{j+1} = s_0 + \left(\frac{(\delta + c_2)(3\delta^2 + 5\delta c_2 + 2c_2^2 - 2c_3)}{2} \right) D_j^4 + O(D_j^5). \tag{23}$$

3.2 Algorithm 2 concrete member

For $\delta = 0.001$, a typical member of Algorithm 2 is put forward as:

Algorithm 4 *Suppose s_0 an initial guess, then the solution s_{k+1} of NL equation can be obtained using the iterative procedure*

$$\begin{aligned} y_j &= s_j - \frac{f(s_j)}{f[s_j,\omega_j] - 0.001 f(s_j)}, \\ s_{j+1} &= y_j - \frac{f(y_j)}{\Phi(s,t)}\left[1 + \frac{1}{2}\frac{f(y_j)}{\Phi(s,t)}\left(\frac{(f[s_j,\omega_j] - 0.001 f(s_j)) - \Phi(s,t)}{f(s_j)}\right)\right]. \end{aligned} \tag{24}$$

4 Numerical Results

The concrete members (Algorithm 3 (Alg 3) and Algorithm 4 (Alg 4)) of methods developed (Algorithm 1 and Algorithm 2) in this work are tested in this section to verify their performance when utilised to solve NL equations. For performance comparison sake, the obtained computation outputs by the developed methods were put side by side with the outputs of some existing CO four methods that are also modification of HM and are optimal. The compared methods includes Nadeem et al., method (NM) in [5] given as:

$$\begin{aligned} s_{j+1} &= y_j - \frac{f(y_j)}{F(s_j,y_j)} - \left(\frac{f^2(y_j)G(s_j,y_j)}{2F^3(s_j,y_j)}\right), \\ F(s_j,y_j) &= \frac{2[f(s_j) - f(y_j)]}{x_j - y_j} - f'(s_j), \\ G(s_j,y_j) &= \frac{6[f(s_j) - f(y_j)] - 2(s_j - y_j)[2f'(s_j) + F(s_j,y_j)]}{(s_j - y_j)^2}. \end{aligned} \tag{25}$$

and Sarima et al., method (SM) in [15] and presented as:

$$\begin{aligned} s_{j+1} &= y_j - \frac{f(y_j)}{F(s_j,y_j)} - M(s_j,y_j)\frac{f^2(y_j)}{2f'^3(s_j)}, \\ M(s_j,y_j) &= \frac{10f(y_j) + 4f(s_j)}{(y_j - s_j)^2}, \end{aligned} \tag{26}$$

where $y_j = s_j - (f(s_j)/f'(s_j))$.
The computational CO ν_{coc} due to Jay in [3] given as

$$\nu_{coc} = \frac{\log|f(s_{j+1})|}{\log|f(s_j)|}, \tag{27}$$

number of iterations to achieve convergence (NIT) and absolute function of last iteration point value ($|f(s_j)|$) obtained from each method were used for

comparison. To obtain computation results, programs were written and implemented in MAPLE 2017 software domain using a computer with the specifications: 2GB RAM processor Intel Celeron(R). To terminate all programs execution, a tolerance of $|f\ (s_j)| \leq 10^{-1000}$ was adopted. In order to minimize truncation error, all computation outputs were given in 2000 digits of precision. The format for computation result presentation is $X_{-Y} = X \times 10^{-Y}$, where $X, Y \in \Re$.
Table 1 presents the NL equations $f_j(s) = 0$ utilized for testing the methods applicability and computational comparison.

Table 1 Test equations

$f_i(s) = 0$	s^*
$f_1(s) = 2s - ln\ s - 7s = 0$	$4.2199064837\ldots$
$f_2(s) = s^3 - 9s + 1 = 0$	$2.9428200577\ldots$
$f_3(s) = s - 3\ ln\ (s) = 0$	$1.8571838602\ldots$
$f_4(s) = -20s^5 - \frac{s}{2} + \frac{1}{2} = 0$	$0.4276772969\ldots$
$f_5(s) = s(s^2 - 1) + 3 = 0$	$-1.6716998816\ldots$
$f_6(s) = 1 - (s^2 - sin^2(s)) = 0$	$1.4044916482\ldots$

4.1 Results Discussion

The computational outcomes obtained when the developed Alg 3, Alg 4 and the compared methods (NM and SM) were used to solve the NL equations in Table 1, are presented in Table 2-5. From Table 2-4, observe that Alg 3 and Alg 4 solved the NL equations $f_1(s)$, $f_2(s)$ and $f_3(s)$ when $f'(s) = 0$. While the compared methods failed because of the presence of evaluation of quotients with zero as divisor. That is, the evaluation of $f'(\cdot)$ vanished. Furthermore, for NL equations with non-vanishing $f'(\cdot)$, the Alg 3 and Alg 4 in most cases, performed better than the compared methods.

Table 2 Methods results comparison for $f_1(s) = 0$

$Methods$	s_0	NIT	$\lvert f(s_{j+1})\rvert$	ν_{coc}
NM	0.5		Failed	-
SM			Failed	-
Alg 3		7	1.5_{-1131}	4.0
Alg 4		10	2.4_{-1276}	4.0
NM	3.0	5	5.6_{-1264}	4.0
SM		5	4.1_{-1310}	4.0
Alg 3		5	2.0_{-1290}	4.0
Alg 4		5	8.9_{-1297}	4.0

Table 3 Methods results comparison for $f_2(s) = 0$

$Methods$	s_0	NIT	$\lvert f(s_{j+1})\rvert$	ν_{coc}
NM	$\sqrt{3}$		Failed	-
SM			Failed	-
Alg 3		12	9.3_{-0813}	4.0
Alg 4		6	4.2_{-1584}	4.0
NM	2.5	5	2.1_{-0495}	4.0
SM		6	1.8_{-0846}	4.0
Alg 3		5	3.8_{-0632}	4.0
Alg 4		5	6.3_{-0694}	4.0

Table 4 Methods results comparison for $f_3(s) = 0$

Methods	s_0	NIT	$\lvert f(s_{j+1})\rvert$	ν_{coc}
NM	3.0		Failed	-
SM			Failed	-
Alg 3		10	1.0_{-1999}	4.0
Alg 4		10	7.2_{-1480}	4.0
NM	0.5	9	8.2_{-1050}	4.0
SM		6	3.9_{-0698}	4.0
Alg 4		6	1.5_{-0993}	4.0
Alg 6		6	5.3_{-1030}	4.0

Table 5 Methods results comparison for $f_4(s) = 0$, $f_5(s) = 0$ and $f_6(s) = 0$

$f(s)$	$Methods$	s_0	NIT	$\lvert f(s_{j+1})\rvert$	ν_{coc}
$f_4(s)$	NM	0.26	6	5.73_{-0576}	4.0
	SM		27	3.2_{-1878}	4.0
	Alg 3		6	5.1_{-1328}	4.0
	Alg 4		6	1.1_{-1197}	4.0
$f_5(s)$	NM	0.0		Failed	-
	SM		22	8.7_{-0610}	4.0
	Alg 3		8	7.3_{-1634}	4.0
	Alg 4		13	8.8_{-1025}	4.0
$f_6(s)$	NM	0.5	7	8.7_{-0762}	4.0
	SM		17	2.9_{-0650}	4.0
	Alg 3		7	7.4_{-1054}	4.0
	Alg 4		7	9.4_{-0907}	4.0

5 Conclusion

In this manuscript, the HM have been successfully modified to attain optimal CO four and further enhanced to not require the evaluation of function derivatives in its implementation. The presented methods, also possess the advantages of obtaining solution of NL equations even when $f'(s) = 0$ or $f'(s) \approx 0$. These results, are the major advantages of the methods presented herein, over existing modified HM in literature.

Acknowledgements

The Authors are grateful to the Editors and anonymous reviewers for good reviews and suggestions that eventually improved this work.

References

1. M. A. Hafiz, S. M. H. Al-Goria, Solving nonlinear equations using a new tenth-and seventh-order methods free from second derivatives, Int. J. of Differential Equations and Applications, 12, 169–183 (2013).
2. A. S. Householder, The numerical treatment of a single nonlinear equation, McGraw-Hill, New York, 1970.
3. L, O. Jay, A note on Q-order of convergence BIT Numerical Math., 41, 422–429 (2001).
4. H. Kung, J. F. Traub, Optimal order of one-point and multi-point iteration, Applied Mathematics and Computation, 21, 643–651 (1974).
5. G. A. Nadeem, W. Aslam, F. Ali, An optimal fourth-order second derivative free iterative method for nonlinear scientific equations, Kuwait J. Sci., 50, 1–15 (2023).
6. W. Nazeer, A. R. Nizami, M. Tanveer, I. Sarfraz, A ninth-order iterative method for nonlinear equations along with polynomiography, J. of Prime Research in Mathematics, 13, 41–55 (2017).
7. W. Nazeer, M. Tanveer, S. M. Kang, A. Naseem, A new Householder's method free from second derivatives for solving nonlinear equations and polynomiography, J. of Nonlinear Science and Applications, 9, 998–1007 (2016).
8. M. A. Noor, V. Gupta, Modified Householder iterative method free from second derivatives for nonlinear equations, Applied Mathematics and Computation, 190, 1701–1706 (2007).
9. O. Ogbereyivwe, J. O. Emunefe, An optimal family of methods for obtaining the zero of nonlinear equation, Mathematics and Computational Sciences , 3, 17–24 (2022).
10. O. Ogbereyivwe, O. Izevbizua, A three-parameter class of power series based iterative method for approximation of nonlinear equations solution, Iran. J. Numer. Anal. Optim., 13, 157–169 (2023).
11. O. Ogbereyivwe, S. S. Umar, O. Izevbizua, Some high-order convergence modifications of the Householder method for nonlinear equations, Communications in Nonlinear Analysis, 5, 1–16 (2023).
12. O. Ogbereyivwe, V. Ojo-Orobosa, Families of Means-Based Modified Newtons Method for Solving Nonlinear Models, Punjab University Journal of Mathematics, 53, 779–791 (2021).
13. O. Ogbereyivwe, S. S. Umar, Modified Householder iterative scheme requiring no function derivative for solving nonlinear equations, Mathematical Anal. and Convex Optimization, 3, 77–82 (2022).
14. A. M. Ostrowski, Solutions of Equations and System of Equations, Academic Press, New York, London, (1966).
15. S. A. Sarima, I. Hashim, New optimal Newton-Householder methods for solving nonlinear equations and their dynamics, Computers, Materials and Continua, 65, 69–85 (2020).
16. M. Tanveer, I. Ahmad, W. Aslam, O. Mehmood, Predictor corrector iterative method for solving nonlinear equations, Sci. Int. (Lahore), 28, 891–898 (2016).
17. X. Wu, A significant improvement on Newton's iterative method, J. Appl. Math. Mech., 20, 924–927 (1999).