International Journal of the Physical Sciences Vol. 6(8), pp. 1887-1893, 18 April, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.425
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Higher-order iterative methods free from second derivative for solving nonlinear equations

Muhammad Aslam Noor[1, 2]*, Waseem Asghar Khan[1], Khalida Inayat Noor[1] and Eisa Al-said[2]

[1]Mathematics Department, COMSATS Institute of Information Technology, Park Road, Islamabad, Pakistan.
[2]Mathematics department, College of Science, King Saud University, Riyadh, Saudi Arabia.

Accepted 04 April, 2011

In this paper, we suggest and analyze some new higher-order iterative methods free from second derivative for solving nonlinear equations. Per iteration, these new methods require three evaluations of the function and two of its first-derivative with efficiency index = $9^{1/5} \approx 1.552.$ Convergence of their methods is also considered. Several numerical examples are given to illustrate the efficiency and performance of these new methods. These new iterative methods may be viewed as an alternative to the known methods.

Key words: Nonlinear equations, Newton method, convergence criteria, root finding method, numerical examples.

INTRODUCTION

It is well known that a wide class of problem which arises in several branches of pure and applied science can be studied in the general framework of the nonlinear equations $f(x)=0,$ Due to their importance, several numerical methods have been suggested and analyzed under certain conditions. These numerical methods have been constructed using different techniques such as Taylor series, homotopy perturbation method and its variant forms, quadrature formula, variational iteration method, and decomposition method (Chun, 2005, 2007; Ham et al., 2008; Javidi, 2009; Noor, 2006, 2010, 2010a; Noor et al. (2007, 2007a, 2007b). Using the technique of updating the solution and Taylor series expansion, Noor et al. (2007b) have suggested and analyzed a sixth-order predictor-corrector iterative type Halley method for solving the nonlinear equations. Ham et al. (2008) and Chun (2007) have also suggested a class of fifth-order and sixth-order iterative methods. In the implementation of the method of Noor et al. (2007b), one has to evaluate the second derivative of the function, which is a serious drawback of these methods. To overcome these drawbacks, we modify the predictor-corrector Halley method by replacing the second derivatives of the function by its suitable finite difference scheme. We prove that the new modified predictor-corrector method is of sixth-order convergence. We also present the comparison of these new methods with the methods of Noor et al. (2007a), Ham et al. (2008) and Chun (2005, 2007). We discus the efficiency index and computational order of convergence of new methods. Several examples are given to illustrate the efficiency and performance of these new methods. These new results may stimulate further research in this area.

*Corresponding author. E-mail: mnoor.c@ksu.edu.sa, noormaslam@hotmail.com.

ITERATIVE METHODS

For the sake of completeness, we recall the Newton method and Hellay method. These methods are as follows:

Algorithm 2.1: For a given x_0, compute approximates solution x_{n+1} by the iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{1}$$

Algorithm 2.1 is the well-known Newton method, which has a quadratic convergence.

Algorithm 2.2: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2f'^2(x_n) - f(x_n)f''(x_n)}. \quad (2)$$

This is known as Halley's method and has cubic convergence (Halley, 1964).

Noor et al. (2007b) have suggested the following two-step method, using Algorithm 2.1 as predictor and Algorithm 2.2 as a corrector.

Algorithm 2.3: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)f''(y_n)}. \quad (3)$$

If $f''(y_n) = 0$, then Algorithm 2.3 is called the predictor-corrector Newton method and has fourth-order convergence (Traub, 1964). In order to implement Algorithm 2.3, one has to find the second derivative of this function, which may create some problems. To overcome this drawback, several authors have developed involving only the first derivative. This idea plays a significant part in developing some iterative methods free from second derivatives. To be more precise, we consider:

$$f''(y_n) = \frac{2}{y_n - x_n}\left\{2f'(y_n) + f'(x_n) - 3\frac{f(y_n) - f(x_n)}{y_n - x_n}\right\} \equiv P_f(x_n, y_n). \quad (4)$$

Combining Equations (3) and (4), we suggest the following new iterative method for solving the nonlinear Equation (1).

Algorithm 2.4: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)}.$$

Algorithm 2.4 is called the new two-step modified Halley's method free from second derivative for solving nonlinear Equation (1). This method has sixth-order convergence. Per iteration, this method requires two evaluations of the function and two evaluations of its first-derivative, so its efficiency index equals to $6^{1/4} \approx 1.565.$

Following the technique of predictor-corrector of the solution (Chen, 2007; Ham et al., 2008), we derive the three-step iterative method for solving the nonlinear equations.

Algorithm 2.5: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$z_n = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)}$$
$$x_{n+1} = z_n - \frac{f'(x_n) + 3f'(y_n)}{6f'(y_n) - 2f'(x_n)}\frac{f(z_n)}{f'(x_n)}.$$

This new method has seventh-order convergence. Per iteration, this method requires two evaluations of the function and two evaluations of its first-derivative, so its efficiency index equals to $7^{1/4} \approx 1.475.$

Algorithm 2.6: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$z_n = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)}$$
$$x_{n+1} = z_n - \frac{f'(x_n)}{f'(y_n)}\frac{f(z_n)}{f'(x_n)}.$$

Algorithm 2.7: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$z_n = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)}$$
$$x_{n+1} = z_n - \frac{f'(y_n)}{2f'(y_n) - f'(x_n)}\frac{f(z_n)}{f'(x_n)}.$$

These new Algorithms 2.6 and 2.7 have eighth-order convergence. Per iteration these methods requires two evaluations of the function and two evaluations of its first-derivative, so its efficiency index equals to $8^{1/4} \approx 1.515.$

In the similar way, we can suggest the following new three-step iterative methods for solving the nonlinear equations using the predictor-corrector technique.

Algorithm 2.8: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

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

$$z_n = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)} \quad (5)$$

$$x_{n+1} = z_n - \frac{f'(x_n) + f'(y_n)}{3f'(y_n) - f'(x_n)} \frac{f(z_n)}{f'(x_n)} \quad (6)$$

Algorithm 2.9: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

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

$$z_n = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)}$$

$$x_{n+1} = z_n + \frac{2f'^2(x_n)}{f'^2(x_n) - 4f'(x_n)f'(y_n) + f'^2(y_n)} \frac{f(z_n)}{f'(x_n)}.$$

Algorithm 2.10: For a given x_0, compute approximates solution x_{n+1} by the iterative schemes:

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

$$z_n = y_n - \frac{2f(y_n)f'(y_n)}{2f'^2(y_n) - f(y_n)P_f(x_n, y_n)}$$

$$x_{n+1} = z_n - \frac{2f'(x_n)f'(y_n)}{f'^2(x_n) + 2f'(x_n)f'(y_n) - f'^2(y_n)} \frac{f(z_n)}{f'(x_n)}.$$

Convergence criteria

Now, we consider the convergence criteria of Algorithm 2.8. In a similar way, we can discuss the convergence of other algorithms.

Theorem 3.1: Let $\alpha \in D$ be a simple zero of sufficiently differentiable function $f: D \subset R \to R$ for an open interval D. And x_0 is initial choice, then Algorithm 2.8 has ninth-order convergences.

Proof: If α is the root and e_n be the error at *nth* iteration, than $e_n = x_n - \alpha$, using Taylor's expansion, we have:

$$f(x_n) = f'(x_n)e_n + \frac{1}{2!}f''(x_n)e_n^2 + \frac{1}{3!}f'''(x_n)e_n^3 + \frac{1}{4!}f^{(iv)}(x_n)e_n^4 + \frac{1}{5!}f^{(v)}(x_n)e_n^5 + \frac{1}{6!}f^{(vi)}(x_n)e_n^6 + O(e_n^7),$$

$$f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + O(e_n^6)], \quad (7)$$

$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + O(e_n^6)], \quad (8)$$

Where: $c_k = \frac{1}{k!}\frac{f^{(k)}(\alpha)}{f'(\alpha)}, \quad k = 2,3,...,$ let $e_n = x_n - \alpha.$

From Equations (7) and (8), we have:

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2e_n^2 - (2c_3 - 2c_2^2)e_n^3 - (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 + (-6c_3^2 + 20c_3c_2^2 - 10c_2c_4 + 4c_5 - 8c_2^4)e_n^5 + O(e_n^6). \quad (9)$$

From Equation (9), we have:

$$y_n = \alpha + c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 + O(e_n^5). \quad (10)$$

$$f(y_n) = f'(\alpha)[c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (3c_4 - 7c_2c_3 + 5c_2^3)e_n^4 + O(e_n^5)], \quad (11)$$

and,

$$f'(y_n) = f'(\alpha)[1 + c_2^2e_n^2 + 4(c_2c_3 - c_2^3)e_n^3 + (8c_2^4 + 6c_2c_4 - 11c_2^2c_3)e_n^4 + O(e_n^5). \quad (12)$$

$$\frac{f(y_n)P_f(x_n, y_n)}{2f'^2(y_n)} = c_2^2e_n^2 + (2c_2c_3 - 2c_2^3)e_n^3 + (c_2^4 + 2c_2c_4 - 4c_2^2c_3)e_n^4 + (-4c_3c_2^3 + 4c_2^5 + 2c_2c_5 - 6c_2^2c_4 + 6c_2c_3^2 - 2c_3c_4)e_n^5 + (36c_3c_2^4 + 6c_4c_2^3 - 12c_2^6 - 45c_3^2c_2^2 - 8c_2^2c_5 - 3c_4^2 - 4c_5c_3 + 2c_2c_6 + 16c_2c_3c_4 + 6c_3^3)e_n^6 + O(e_n^7). \quad (13)$$

$$z_{n+1} = y_n - f(y_n)\Big/ f'(y_n)\left[1 - \frac{f(y_n)P_f(x_n, y_n)}{2f'^2(y_n)}\right] = (c_2^2c_4 - c_3c_2^3 + c_2^5)e_n^6 + (2c_5c_2^2 + 4c_2c_3c_4 - 6c_4c_2^3 - 6c_3^2c_2^2 + 12c_3c_2^4 - 6c_2^6)e_n^7 + (21c_2^7 + 3c_6c_2^2 - 43c_2^2c_3c_4 + 288c_2c_3c_5 - 9c_2^3c_5 - 63c_3c_2^5 - 12c_2c_3^3 + 29c_4c_2^4 + 6c_2c_4^2 + 4c_4c_3^2 + 57c_3^2c_2^3)e_n^8 + O(e_n^9). \quad (14)$$

Table 1. Approximate solution of Example 1.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	6	1.36523001341409684576080682	3.98235e-43	2.21790e-22	2
NN1	3	1.36523001341409684576080682	0	5.58014e-26	6.11
NK	3	1.36523001341409684576080682	0	2.78777e-19	5.16
Alg. 2.4	3	1.36523001341409684576080682	0	1.81330e-55	6.02
Alg. 2.5	3	1.36523001341409684576080682	0	4.52450e-50	6.99
Alg. 2.6	3	1.36523001341409684576080682	0	1.81330e-55	7.96
Alg. 2.7	3	1.36523001341409684576080682	0	2.41159e-44	8.01
Alg. 2.8	3	1.36523001341409684576080682	0	1.81330e-55	9.11
Alg. 2.9	3	1.36523001341409684576080682	0	4.52450e-50	9.16
Alg. 2.10	3	1.36523001341409684576080682	0	1.81330e-55	9.06
JM1	4	1.36523001341409684576080682	0	2.41159e-44	4
JM2	4	1.36523001341409684576080682	0	0	-
LJM	4	1.36523001341409684576080682	0	9.46019e-37	5.02
CM1	4	1.36523001341409684576080682	0	0	-
CM2	3	1.36523001341409684576080682	0	7.07294e-23	6.08
CM3	3	1.36523001341409684576080682	0	2.08447e-20	5.03

Table 2. Approximate solution of Example 2.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	5	1.4044916482153412260350868	-4.87943e-34	1.58381e-17	2
NN1	3	1.4044916482153412260350868	0	4.15448e-39	6.04
NK	3	1.4044916482153412260350868	0	4.04252e-30	5.07
Alg. 2.4	3	1.4044916482153412260350868	0	1.58381e-17	6
Alg. 2.5	3	1.4044916482153412260350868	0	4.93796e-25	6.89
Alg. 2.6	3	1.4044916482153412260350868	0	4.76144e-19	7.99
Alg. 2.7	3	1.4044916482153412260350868	0	5.69635e-21	8.01
Alg. 2.8	3	1.4044916482153412260350868	0	1.84366e-30	9
Alg. 2.9	3	1.4044916482153412260350868	0	2.01280e-55	9.13
Alg. 2.10	3	1.4044916482153412260350868	0	1.58381e-17	9.01
JM1	3	1.4044916482153412260350868	0	1.58381e-17	4.03
JM2	3	1.4044916482153412260350868	0	4.93796e-25	5.05
LJM	3	1.4044916482153412260350868	0	4.76144e-19	5.12
CM1	3	1.4044916482153412260350868	0	3.75135e-25	5.04
CM2	3	1.4044916482153412260350868	0	9.56327e-36	6.03
CM3	3	1.4044916482153412260350868	0	4.82529e-35	4.86

Using Equations (7) to (14) in Algorithm 2.8, we have:

$$x_{n+1}=\alpha+(2c_2^8+2c_4c_2^5-4c_3c_2^6+2c_3^2c_2^4-2c_3c_4c_2^3)e_n^9+O(e_n^{10}).$$

Thus, we have:

$$e_{n+1}=(2c_2^8+2c_4c_2^5-4c_3c_2^6+2c_3^2c_2^4-2c_3c_4c_2^3)e_n^9+O(e_n^{10}).$$

which shows that Algorithm 2.8 has ninth-order convergence.

NUMERICAL EXAMPLES

In this study, we present some numerical examples to illustrate the efficiency and the accuracy of the new developed iterative methods (Tables 1 to 7). We compare our new methods obtained in Algorithm 2.4 to Algorithm 2.10 with Newton's method (NM), method of Noor et al. ((2007b) NN1), method of Noor et al. (2007c, NK), methods of Chun (2008 CM1, CM2 and CM3), method of Li et al.(2009, LJ) and method of Javidi (2009, JM1 and

Table 3. Approximate solution of Example 3.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	6	0.2575302854398607604553673	2.92590e-55	9.10261e-28	2
NN1	4	0.2575302854398607604553673	0	2.00000e-60	3.48
NK	4	0.2575302854398607604553673	0	4.15360e-40	5.02
Alg. 2.4	3	0.2575302854398607604553673	0	1.90768e-29	
Alg. 2.5	3	0.2575302854398607604553673	0	1.00000e-59	
Alg. 2.6	3	0.2575302854398607604553673	0	1.00000e-59	
Alg. 2.7	3	0.2575302854398607604553673	0	5.69635e-21	
Alg. 2.8	3	0.2575302854398607604553673	-1.00000e-59	1.00000e-59	-
Alg. 2.9	3	0.2575302854398607604553673	0	3.49682e-42	4.98
Alg. 2.10	3	0.2575302854398607604553673	0	2.36963e-54	5
JM1	4	0.2575302854398607604553673	0	1.90768e-29	5.05
JM2	4	0.2575302854398607604553673	1.00000e-59	1.00000e-59	-
LJM	4	0.2575302854398607604553673	1.00000e-59	1.00000e-59	-
SM	4	0.2575302854398607604553673	0	5.69635e-21	5.09
CM1	4	0.2575302854398607604553673	0	3.75135e-25	5.04
CM2	4	0.2575302854398607604553673	0	9.56327e-36	6.03
CM3	4	0.2575302854398607604553673	0	4.82529e-35	4.86

Table 4. Approximate solution of Example 4.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	5	0.7390851332151606416553121	-2.03197e-32	2.34491e-16	1.99
NN1	3	0.7390851332151606416553121	-1.00000e-60	5.72241e-34	5.60
NK	3	0.7390851332151606416553121	-1.00000e-60	2.50589e-21	4.66
Alg. 2.4	3	0.7390851332151606416553121	0	1.90768e-29	6
Alg. 2.5	3	0.7390851332151606416553121	0	1.00000e-59	7.02
Alg. 2.6	3	0.7390851332151606416553121	0	1.00000e-59	8
Alg. 2.7	3	0.7390851332151606416553121	0	5.69635e-21	8.12
Alg. 2.8	3	0.7390851332151606416553121	-1.00000e-60	1.00000e-60	9
Alg. 2.9	3	0.7390851332151606416553121	-1.00000e-60	1.00000e-60	-
Alg. 2.10	3	0.7390851332151606416553121	0	1.90768e-29	9.02
JM1	3	0.7390851332151606416553121	-1.00000e-60	2.34491e-16	3.60
JM2	3	0.7390851332151606416553121	-1.00000e-60	5.83737e-24	4.58
LJM	3	0.7390851332151606416553121	1.00000e-60	1.70292e-22	4.45
SM	3	0.7390851332151606416553121	1.00000e-60	2.24183e-21	4.48
CM1	3	0.7390851332151606416553121	-1.00000e-60	3.28792e-23	4.75
CM2	3	0.7390851332151606416553121	1.00000e-60	2.52820e-24	5.90
CM3	3	0.7390851332151606416553121	-1.00000e-60	2.17244e-17	4.79

JM2). All computations have been done by using the Maple 11 package with 25 digit floating point arithmetic. We accept an approximate solution rather than the exact root, depending on the precision (ε) of the computer. We use the following stopping criteria for computer programs: (i) $|x_{n+1}-x_n|<\varepsilon,\quad (ii)\quad |f(x_{n+1})|<\varepsilon$ and so, when the stopping criterion is satisfied, x_{n+1} is taken as the exact root α computed. For numerical illustrations we have used the fixed stopping criterion $\varepsilon = 10^{-15}$. As for the convergence criteria, it was required that the distance of two consecutive approximations δ. Also displayed are the number of iterations to approximate the zero (IT), the approximate root x_n, the value $f(x_n)$ and the computational order of convergence (COC) can be approximated using the formula,

Table 5. Approximate solution of Example 5.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	7	2	5.03100e-56	1.29484e-28	2
NN1	3	2	0	1.80249e-17	5.79
NK	3	2	0	4.90379e-24	5
Alg. 2.4	3	2	0	1.29484e-28	6
Alg. 2.5	3	2	0	2.35729e-49	6.88
Alg. 2.6	3	2	0	9.03074e-34	7.98
Alg. 2.7	3	2	0	6.68430e-38	7.79
Alg. 2.8	3	2	0	3.97550e-36	8.68
Alg. 2.9	3	2	0	7.32429e-34	8.61
Alg. 2.10	3	2	0	1.90768e-29	8.78
JM1	4	2	0	1.29484e-28	4
JM2	4	2	0	2.35729e-49	5
LJM	4	2	0	9.03074e-34	4.98
SM	4	2	0	6.68430e-38	5
CM1	4	2	0	4.31570e-35	5
CM2	4	2	0	0	-
CM3	4	2	0	3.52600e-55	5

Table 6. Approximate solution of Example 6.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	5	2.15443469003188372175929 36	3.29189e-35	2.25681e-18	2
NN1	3	2.1544346900318837217592936	-8.00000e-59	4.50228e-42	6.02
NK	3	2.1544346900318837217592936	1.00000e-58	4.89248e-30	5.04
Alg. 2.4	3	2.1544346900318837217592936	0	1.29484e-28	6
Alg. 2.5	3	2.1544346900318837217592936	0	2.35729e-49	7.01
Alg. 2.6	3	2.1544346900318837217592936	0	9.03074e-34	8.02
Alg. 2.7	3	2.1544346900318837217592936	0	6.68430e-38	8
Alg. 2.8	3	2.1544346900318837217592936	1.00000e-58	1.00000e-58	-
Alg. 2.9	3	2.1544346900318837217592936	-8.00000e-59	8.00000e-59	-
Alg. 2.10	3	2.1544346900318837217592936	0	1.90768e-29	8.99
JM1	3	2.1544346900318837217592936	0	1.29484e-28	4
JM2	3	2.1544346900318837217592936	1.00000e-58	1.64194e-26	5.04
LJM	3	2.1544346900318837217592936	-8.00000e-59	1.51742e-20	5.09
SM	3	2.1544346900318837217592936	1.00000e-58	1.44631e-22	5.06
CM1	3	2.1544346900318837217592936	-8.00000e-59	1.89488e-26	5.02
CM2	3	2.1544346900318837217592936	1.00000e-58	2.54320e-38	6.02
CM3	3	2.1544346900318837217592936	-8.00000e-59	2.39906e-30	4.99

$COC \approx \frac{\ln\left|(x_{n+1}-x_n)/(x_n-x_{n-1})\right|}{\ln\left|(x_n-x_{n-1})/(x_{n-1}-x_{n-2})\right|}$. For the sake of comparison, all examples are the same as in Chen (2005).

Example 1. Consider the equation (Table 1) $f_1(x)=x^3+4x^2-10, \quad x_0=1.$

Example 2. Consider the equation (Table 2) $f_2(x)=\sin^2 x-x^2+1, \quad x_0=1.3.$

Example 3. Consider the equation (Table 3) $f_3(x)=x^2-e^x-3x+2, \quad x_0=2.$

Example 4. Consider the equation (Table 4) $f_4(x)=\cos x-x, \quad x_0=1.7.$

Table 7. Approximate solution of Example 7.

Methods	IT	x_n	$F(x_n)$	δ	COC
NM	9	3	1.37562e-53	4.01112e-28	2
NN1	4	3	0	5.44868e-34	5.98
NK	5	3	0	4.21818e-37	5.00
Alg. 2.4	3	3	0	1.29484e-28	5.98
Alg. 2.5	3	3	0	2.35729e-49	6.78
Alg. 2.6	3	3	0	9.03074e-34	8.00
Alg. 2.7	3	3	0	6.68430e-38	7.89
Alg. 2.8	3	3	0	2.19280e-55	9
Alg. 2.9	3	3	0	3.36801e-24	8.52
Alg. 2.10	3	3	0	1.90768e-29	8.98
JM1	5	3	0	4.01112e-28	4
JM2	5	3	0	1.52828e-54	5
LJM	3	3	2.00000e-58	4.86190e-25	4.96
SM	5	3	0	3.73390e-31	4.99
CM1	7	3	0	1.99393e-30	5
CM2	5	3	0	0	-
CM3	5	3	0	2.19280e-55	5

Example 5. Consider the equation (Table 5) $f_5(x) = (x-1)^3 - 1, \quad x_0 = 2.5.$

Example 6. Consider the equation (Table 6) $f_6(x) = x^3 - 10, \quad x_0 = 2.$

Example 7. Consider the equation (Table 7) $f_7(x) = e^{x^2+7x-30} - 1, \quad x_0 = 3.2.$

CONCLUSIONS

In this paper, we have suggested new higher-order iterative methods free from second derivative for solving nonlinear equation. We also discussed the efficiency index and computational order of convergence of these new methods. Several examples are given to illustrate the efficiency of Algorithms 2.4-2.10. Using the idea of this paper, one can suggest and analyze higher-order multi-step iterative methods for solving nonlinear equations.

ACKNOWLEDGEMENT

This research is supported by the Visiting Professor Program of King Saud University, Riyadh, Saudi Arabia and Research Grant No: VPP.KSU.108. The authors are grateful to the referees for their constructive comments and suggestions.

REFERENCES

Chun C (2005). Iterative methods improving Newton's method by the decomposition method, Comput. Math. Appl., 50: 1559–1568

Chun C (2007). Some improvements of Jarrat's methods with sixth order convergences. Appl. Math. Comput., 190: 1432–1437.

Halley E (1964). A new exact and easy method for finding the roots of equations generally and without any previous reduction, Phil. Roy. Soc. London, 18: 136–147.

Javidi M (2009). Fourth-order and fifth-order iterative methods for nonlinear algebraic equations, Math. Comput. Model., 50: 66-71.

Noor KI, Noor MA (2007). Modified Householder iterative method for nonlinear equations, Appl. Math. Comput., 190: 1534–1539.

Noor MA (2007). New family of iterative methods for nonlinear equations, Appl. Math. Comput., 190: 553–558.

Noor MA (2010). Some iterative methods for solving nonlinear equations using homotopy perturbation method, Int. J. Comp. Math., 87: 141-149.

Noor MA (2010a). On iterative methods for nonlinear equations using homotopy perturbation technique, Appl. Math. Inform. Sci., 4: 227-235.

Noor MA, Noor KI (2007a). Predicot-corrector Halley method for nonlinear equations, Appl. Math. Comput., 188: 1587–1591.

Noor MA, Khan WA, Hussain A (2007b). A new modified Halley method without second derivatives for nonlinear equation, Appl. Math. Comput., 189: 1268-1273.

Traub JF (1964). Iterative methods for solution of equations, Prentice-Hall, Englewood Cliffs, NJ, USA.