Some iterative methods free from second derivatives
for nonlinear equations

Muhammad Aslam Noor

Mathematics Department, COMSATS Institute of Information Technology, Islamabad, Pakistan

Abstract

In a recent paper, Noor [M. Aslam Noor, New classes of iterative methods for nonlinear equations, Appl. Math. Comput., 2007, doi:10.1016/j.amc:2007], suggested and analyzed a generalized one parameter Halley method for solving nonlinear equations using. In this paper, we modified this method which has fourth order convergence. As special cases, we obtain a family of third-order iterative methods for appropriate and suitable choice of the parameter. We have compared this modified Noor method with some other iterative methods which shows that this new iterative method is robust and efficient one. Several examples are given to illustrate the efficiency and the performance of this new method.

Keywords: Modified Halley method; Iterative method; Convergence; Newton method; Taylor series; Examples

1. Introduction

Iterative methods for finding the approximate solutions of the nonlinear equation \( f(x) = 0 \) are being developed using several different techniques including Taylor series, quadrature formulas, homotopy and decomposition techniques, see [1–14] and the references therein. In a recent paper, Noor [9] has suggested and analyzed a one parameter family of iterative methods using the variational iteration technique. It has been shown that this method can be viewed as generalization of the Halley method and is robust one. In the implementation of these methods, one has to evaluate the second derivative of the function, which is itself a serious and difficult problem. To overcome these drawback, we present some modifications of the method of Noor [9] by replacing the second derivative of the function by an appropriate finite difference schemes. We also discuss several cases of this new method, which include several known and new ones. In particular, we show that this method includes the two-step Newton method \([2,6,14]\) as a special case, which is of fourth-order. We also discuss the convergence analysis of these new methods. Several examples are given to illustrate the efficiency and performance of these new methods and their comparison with other iterative methods. These numerical experiments show that these new methods are robust and can be viewed as an alternative to other methods for solving nonlinear equations.

E-mail address: noormaslam@hotmail.com

0096-3003/S - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2007.02.138
2. Iterative methods

Consider the nonlinear equation of the type

\[ f(x) = 0. \] (1)

For simplicity, we assume that \( a \) is a simple root of \( f \) and \( c \) is an initial guess sufficiently close to \( a \). For the sake of completeness and to give the idea, we consider the approximate solution \( x_n \) of (1) such that

\[ f(x_n) \neq 0. \]

Let \( g(x) \) be an arbitrary auxiliary function and \( \lambda \) be a parameter, which is usually called the Lagrange multiplier and can be identified by the optimality condition. Consider the following iterative relation:

\[ x_{n+1} = x_n + \lambda g(x_n) f(x_n). \] (2)

Using the optimality criteria, from (2), we have

\[ \lambda = - \frac{1}{g'(x_n)f(x_n) + g(x_n)f'(x_n)}. \] (3)

From (2) and (3), we have

\[ x_{n+1} = x_n - \frac{g(x_n)f(x_n)}{g'(x_n)f(x_n) + g(x_n)f'(x_n)}. \] (4)

Eq. (4) is the main recurrence relation for the iterative methods. Here the function \( g(x) \) is called the auxiliary function and one has the flexibility to choose this function arbitrary. This is the main novelty and feature of this technique.

By selecting a suitable auxiliary function \( g(x) \), Noor [9] has suggested and analyzed the following iterative method.

**Algorithm 2.1.** For a given \( x_0 \), find the approximate solution \( x_{n+1} \) by the iterative scheme

\[ x_{n+1} = x_n - \frac{f(x_n)[f'(x_n)]^2}{[f'(x_n)]^3 - \frac{2}{3} f(x_n)f''(x_n)}. \]

If \( x = f'(x_n) \), then Algorithm 2.1 reduces to the following iterative method.

**Algorithm 2.2.** For a given \( x_0 \), find the approximate solution \( x_{n+1} \) by the iterative scheme

\[ x_{n+1} = x_n - \frac{2f(x_n)[f'(x_n)]^2}{2[f'(x_n)]^3 - f(x_n)f''(x_n)}, \]

which is well known Halley method and has cubic convergence, see [5,7,10] and the references therein. In addition, for \( f''(x_n) \approx 0 \), Algorithms 2.1 and 2.2 collapse to the well-known Newton method.

It is clear that to implement Algorithms 2.1 and 2.2, one has to evaluate the second derivative of the function. This can create some problems. In order to overcome this drawback, several techniques have been developed. In this paper, we use the following technique for replacing \( f''(x_n) \) in Algorithm 2.1. Using the Taylor series, we have

\[ f(y_n) \approx f(x_n) + (y_n - x_n)f'(x_n) + \frac{(y_n - x_n)^2}{2} f''(x_n) = \frac{1}{2} \left[ f(x_n) \right]^2 f''(x_n), \] (5)

where

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}. \]

Using (5), we can rewrite Algorithm 2.1 in the following equivalent form.
Algorithm 2.3. For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative scheme
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{f(x_n)}{\left[1 - \frac{3}{2} \frac{f(x_n)}{f'(x_n)} \right] f'(x_n)} = x_n - \frac{f(x_n)}{\left[1 - \frac{3}{2} \frac{f(x_n)}{f'(x_n)} \right] f'(x_n)}
\]
This allows us to suggest the following two-step iterative method free from second derivative for solving non-linear equations and this is the main motivation of this paper.

Algorithm 2.4. For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative schemes:
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f'(y_n)}{f'(x_n)}
\]
We would like to point out that for the choice of $\alpha = \frac{f'(y_n)^2}{f'(x_n)^2}$ and neglecting the second powers of $\alpha$, we have the well known [6,14] two-step Newton method of the following type:

Algorithm 2.5. For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative schemes:
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]
\[
x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}
\]
which has fourth-order convergent method, see [14] for more details.

If we take $\alpha = \frac{f'(x_n)}{f'(x_n)}$, then Algorithm 2.4 reduces to the following new fourth-order convergent method.

Algorithm 2.6. For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative schemes:
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f'(y_n)}{f'(x_n)} - \frac{f''(y_n)}{f'(x_n)^2}
\]
Algorithm 2.6 is quite general and include several recently obtained iterative methods, see [8,9], as special cases. In brief, for suitable and appropriate choice of the parameter $\alpha$, one can obtain several new iterative methods for solving the nonlinear equations. This shows that Algorithm 2.4 is quite general and flexible.

We now discuss the convergence criteria for Algorithm 2.4 for an arbitrary parameter $\alpha$. It is worth mentioning that the convergence criteria of Algorithms 2.5 and 2.6 can be discussed for the special values of the parameter $\alpha$. This is the main advantage of this result.

Theorem 2.1. Let $r \in I$ be a simple zero of sufficiently differentiable function $f : I \subseteq R \rightarrow R$ for an open interval $I$. If $x_0$ is sufficiently close to $r$, then the two-step iterative method defined by Algorithm 2.4 satisfies
\[
e_{n+1} = \left\{ \frac{c_2 - 2c_2}{f'(r)} \right\} e_n^2 - \left[ 2c_3 - 2c_2^2 + (6c_2^2 - 2c_3) \frac{1}{f'(r)} - \frac{1}{f'(r)} \frac{c_2}{f'(r)} \right] e_n^3
\]
\[+ \left[ 4c_2^3 - 7c_2c_3 + 3c_4 + (21c_2c_3 - 3c_4 - 25c_2^2) \frac{1}{f'(r)} + (11c_2^2 - 4x^2c_3c_2) \frac{1}{f'(r)} \right] e_n^4 + \cdots \]
The technique is given by

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$  \hspace{1cm} (6)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \alpha \frac{f(y_n)}{f(x_n) f'(x_n)} - \alpha^2 \frac{f^2(y_n)}{f(x_n) f'(x_n) f''(x_n)}.$$  \hspace{1cm} (7)

Let $r$ be a simple zero of $f$. Since $f$ is sufficiently differentiable, by expanding $f(x_n)$ and $f'(x_n)$ about $r$, we get

$$f(x_n) = f'(r)[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + \cdots],$$  \hspace{1cm} (8)

$$f'(x_n) = f'(r)[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + \cdots],$$  \hspace{1cm} (9)

where $c_k = \frac{f^{(k)}(r)}{f'(r)}$, $k = 1, 2, 3, \ldots$ and $e_n = x_n - r$.

Now, from (8) and (9), we have

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2 e_n^2 + 2(c_2 - c_3) e_n^3 + (7c_2 c_3 - 4c_2^2 - 3c_4) e_n^4 + \cdots$$ \hspace{1cm} (10)

From (6) and (10), we get

$$y_n = [r + c_2 e_n^2 + 2(c_3 - c_2^2) e_n^3 + (-7c_2 c_3 + 4c_2^2 + 3c_4) e_n^4 + \cdots].$$ \hspace{1cm} (11)

From (11), we get

$$f(y_n) = f'(r)[(y_n - r) + c_2(y_n - r)^2 + c_3(y_n - r)^3 + c_4(y_n - r)^4 + \cdots]$$

$$= [c_2 e_n^2 + 2(c_2 - c_3) e_n^3 + (-7c_2 c_3 + 5c_2^2 + 3c_4) e_n^4 + \cdots],$$ \hspace{1cm} (12)

$$f'(y_n) = f'(r)[1 + 2c_2(y_n - r) + 3c_3(y_n - r)^2 + 4c_4(y_n - r)^3 + 5c_5(y_n - r)^4 + \cdots]$$

$$= f'(r)[1 + 2c_2 e_n + 4(c_2 c_3 - c_2^2) e_n^3 + (-11c_2 c_3 + 8c_2^2 + 6c_2 c_4) e_n^4 + \cdots].$$ \hspace{1cm} (13)

From (9) and (12), we get

$$\alpha \frac{f(y_n)}{f'(x_n)} = \frac{1}{f'(r)} \left\{ c_2 e_n^2 + (2c_3 - 6c_2^2) e_n^3 + (3c_4 - 21c_2 c_3 + 25c_2^3) e_n^4 + \cdots \right\}.$$ \hspace{1cm} (14)

From (8), (9) and (12), we get

$$\alpha^2 \frac{f^2(y_n)}{f(x_n) f'(x_n) f''(x_n)} = \frac{1}{f'(r)^2} \left[ \alpha^2 c_2^2 e_n^3 + (4\alpha^2 c_2 c_3 - 11\alpha^2 c_2^3) e_n^4 + \cdots \right].$$ \hspace{1cm} (15)

From (11), (14) and (15), we obtain

$$e_{n+1} = \left\{ \left( c_2 - \frac{2c_3}{f'(r)} \right) e_n^2 - \left[ 2c_3 - 2c_2^2 + (6c_3 - 2axc_3) \frac{1}{f'(r)} - \alpha^2 c_2^2 \frac{1}{f'(r)^2} \right] e_n^3 \right. \left. + \left[ 4c_2^3 - 7c_2 c_3 + 3c_4 + (21xc_2c_3 - 3xc_4 - 25c_2^3) \frac{1}{f'(r)} + (11\alpha^2 c_2^3 - 4\alpha^2 c_2^3 c_2) \frac{1}{f'(r)^2} \right] e_n^4 + \cdots \right\},$$

the required result.

We remark that if $\alpha = f'(x_n)(1 + \frac{f(y_n)}{f'(x_n)})$, then Algorithm 2.6 has fourth-order convergent. In a similar way, one can discuss the convergence criteria of other iterative methods which can be obtained as special cases of Algorithm 2.4. \hspace{1cm} \Box

3. Numerical results

We now present some examples to illustrate the efficiency of the new developed two-step iterative methods, see Table 1. We compare the Newton method (NM), Abbasbandy’s method (AM [1]), Chun’s method (CN...
Table 1
Examples and comparison of various iterative schemes

<table>
<thead>
<tr>
<th>IT</th>
<th>$x_n$</th>
<th>$f(x)_n$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1, x_0 = -1$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>7</td>
<td>1.4044916482153412260350868178</td>
<td>-1.04e-50</td>
</tr>
<tr>
<td>AM</td>
<td>5</td>
<td>1.4044916482153412260350868178</td>
<td>-5.81e-55</td>
</tr>
<tr>
<td>GM</td>
<td>3</td>
<td>1.4044916482153412260350868178</td>
<td>1.3e-63</td>
</tr>
<tr>
<td>CM</td>
<td>5</td>
<td>1.4044916482153412260350868178</td>
<td>-2.0e-63</td>
</tr>
<tr>
<td>NR1</td>
<td>4</td>
<td>1.4044916482153412260350868178</td>
<td>-2.00e-63</td>
</tr>
<tr>
<td>NR2</td>
<td>5</td>
<td>1.4044916482153412260350868178</td>
<td>1.30e-63</td>
</tr>
<tr>
<td>$f_2, x_0 = 2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>6</td>
<td>0.25753028543986076045536730494</td>
<td>2.93e-55</td>
</tr>
<tr>
<td>AM</td>
<td>5</td>
<td>0.25753028543986076045536730494</td>
<td>1.0e-63</td>
</tr>
<tr>
<td>GM</td>
<td>4</td>
<td>0.25753028543986076045536730494</td>
<td>0</td>
</tr>
<tr>
<td>CM</td>
<td>4</td>
<td>0.25753028543986076045536730494</td>
<td>1.0e-63</td>
</tr>
<tr>
<td>NR1</td>
<td>4</td>
<td>0.25753028543986076045536730494</td>
<td>0</td>
</tr>
<tr>
<td>NR2</td>
<td>4</td>
<td>0.25753028543986076045536730494</td>
<td>0</td>
</tr>
<tr>
<td>$f_3, x_0 = 1.7$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>5</td>
<td>0.73908513321564165537208767</td>
<td>-2.03e-32</td>
</tr>
<tr>
<td>AM</td>
<td>4</td>
<td>0.73908513321564165537208767</td>
<td>-7.14e-47</td>
</tr>
<tr>
<td>GM</td>
<td>2</td>
<td>0.73908513321564165537208767</td>
<td>0</td>
</tr>
<tr>
<td>CM</td>
<td>4</td>
<td>0.73908513321564165537208767</td>
<td>0</td>
</tr>
<tr>
<td>NR1</td>
<td>3</td>
<td>0.73908513321564165537208767</td>
<td>0</td>
</tr>
<tr>
<td>NR2</td>
<td>4</td>
<td>0.73908513321564165537208767</td>
<td>0</td>
</tr>
<tr>
<td>$f_4, x_0 = 3.5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>8</td>
<td>2</td>
<td>2.06e-42</td>
</tr>
<tr>
<td>AM</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>GM</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>CM</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>NR1</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>NR2</td>
<td>4</td>
<td>2</td>
<td>-1.81e-60</td>
</tr>
<tr>
<td>$f_5, x_0 = 1.5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>7</td>
<td>2.154436900318837217592935665</td>
<td>2.06e-54</td>
</tr>
<tr>
<td>AM</td>
<td>5</td>
<td>2.154436900318837217592935665</td>
<td>-5.0e-63</td>
</tr>
<tr>
<td>GM</td>
<td>2</td>
<td>2.154436900318837217592935665</td>
<td>-1.0e-61</td>
</tr>
<tr>
<td>CM</td>
<td>5</td>
<td>2.154436900318837217592935665</td>
<td>-5.0e-63</td>
</tr>
<tr>
<td>NR1</td>
<td>4</td>
<td>2.154436900318837217592935665</td>
<td>-5.00e-63</td>
</tr>
<tr>
<td>NR2</td>
<td>5</td>
<td>2.154436900318837217592935665</td>
<td>1.00e-62</td>
</tr>
<tr>
<td>$f_6, x_0 = -2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>9</td>
<td>-1.2076478271309189270094167584</td>
<td>-2.27e-40</td>
</tr>
<tr>
<td>AM</td>
<td>6</td>
<td>-1.2076478271309189270094167584</td>
<td>-4.0e-63</td>
</tr>
<tr>
<td>GM</td>
<td>3</td>
<td>-1.2076478271309189270094167584</td>
<td>-4.0e-63</td>
</tr>
<tr>
<td>CM</td>
<td>6</td>
<td>-1.2076478271309189270094167584</td>
<td>-4.0e-63</td>
</tr>
<tr>
<td>NR1</td>
<td>5</td>
<td>-1.2076478271309189270094167584</td>
<td>-4.00e-63</td>
</tr>
<tr>
<td>NR2</td>
<td>5</td>
<td>-1.2076478271309189270094167584</td>
<td>-4.00e-63</td>
</tr>
<tr>
<td>$f_7, x_0 = 3.5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>13</td>
<td>3</td>
<td>1.52e-47</td>
</tr>
<tr>
<td>AM</td>
<td>7</td>
<td>3</td>
<td>-4.33e-48</td>
</tr>
<tr>
<td>GM</td>
<td>Diverges</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CM</td>
<td>8</td>
<td>3</td>
<td>2.0e-62</td>
</tr>
<tr>
<td>NR2</td>
<td>7</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>NR2</td>
<td>7</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

[3], Grau and DB’s method (GM [4]), Algorithm 2.5 (NR1) and Algorithm 2.6 (NR2), introduced in this present paper. We use $\varepsilon = 10^{-15}$. The following stopping criteria is used for computer programs:

(i) $|x_{n+1} - x_n| < \varepsilon$,  
(ii) $|f(x_{n+1})| < \varepsilon$. 


The examples are the same as in [3,5–9].

\[ f_1(x) = \sin^2 x - x^2 + 1, \]
\[ f_2(x) = x^2 - e^x - 3x + 2, \]
\[ f_3(x) = \cos x - x, \]
\[ f_4(x) = (x - 1)^3 - 1, \]
\[ f_5(x) = x^3 - 10, \]
\[ f_6(x) = xe^{x^2} - \sin^2 x + 3 \cos x + 5, \]
\[ f_7(x) = e^{x^2 + 7x - 30} - 1. \]

From Table 1, we see that our methods are compatible with the two-step Newton, Chun and Grau et al. methods. It is clear from the table that the method of Grau and Diaz-Barrero [4] diverges whereas new method converges and has less number of iterations as compared with Chun’s method. In view of this fact, these new methods can be viewed as a significant improvement of the previously known methods and can be considered as alternative method to that of Newton and its variant forms.

Acknowledgement

This research is supported by the Higher Education Commission, Pakistan, through research Grant No: I-28/HEC/HRD/2005/90.

References