A modification of Newton's method with third-order convergence

Husnia Mohamed Eldanfour
Department of Mathematics- Faculty of Science- Misurata University –PO Box 996 Misurata -Libya
hsniam@yahoo.com

Abstract
In this paper, we present an improvement to the Newton’s method that it has third-order convergence for solving non-linear equations. From error analysis the higher order of convergence is proving. Per iteration, the modified method requires two evaluations of function and one evaluation of its first derivative. Numerical examples are illustrating the performance of the modified method.

Keywords: Newton's method, Third-order convergence, Function Evaluations.

1- Introduction
Newton’s method for the approximation of the root $\xi$ of a nonlinear equation (or system of non-linear equations) is well known, in paper [1] Newton’s method may be seen as the approximation of the indefinite integral arising from Newton’s theorem

$$f(x) = f(x_n) + \int_{x_n}^{x} f'(t)dt$$

by using the rectangular rule (the Newton- Cotes quadrature formula of order zero) for the computation of the integral in (1).

$$\int_{x_n}^{x} f'(t)dt \approx (x - x_n)f'(x_n)$$

And, looking for $f(x) = 0$, we obtain the new value

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

Some modified Newton's methods with third-order convergence have been developed in [1,2,3], by considering different quadrature formula for the computation of the indefinite integral of (1).

Weerakoom and Fernando [3] used the trapezoidal rule to compute the integral of (1) to obtain the implicit method
\textbf{Husnia Mohamed Eldanfour: A modification of Newton's method with third-order ...}

\begin{equation}
x_{n+1} = x_n - 2 \frac{f(x_n)}{f'(x_{n+1}) + f'(x_n)}
\end{equation}

and, replacing \( f'(x_{n+1}) \) with \( f'(x^*_n) \); where \( x^*_n = x_n - f(x_n)/f'(x_n) \) is the classical Newton's iterate, they obtained the explicit with third-order convergence:

\begin{equation}
x_{n+1} = x_n - 2 \frac{f(x_n)}{f'(x^*_n) + f'(x_n)} \tag{3}
\end{equation}

The midpoint rule for the integral of (1) gives that [1]

\begin{equation}
x_{n+1} = x_n - \frac{f(x_n)}{f' \left( \frac{x_n + x^*_n}{2} \right)} \tag{4}
\end{equation}

For any quadrature formula of order higher than zero to approximating the integral of (1), the following theorem gives that modified Newton's methods have third-order convergence [1].

**Theorem 1** [1]. The modified Newton's methods obtained by approximating the integral by the quadrature formula of order at least one, and writing the explicit form by replacing \( x_{n+1} \) with \( x^*_n \) in the obtaining implicit method to obtain;

\[ x^*_n = x_n - \frac{f(x_n)}{f'(x_n)} \]

if \( \xi \) is a simple root of \( f(x_n) \), of order three with

\[ g_m^{ss}(\xi) = \frac{f^{ss}(\xi)}{f'(\xi)} \left[ 3 \sum_{i=1}^{m} A_i \tau_i^2 - 1 \right] + \frac{3}{2} \left( \frac{f^s(\xi)}{f'(\xi)} \right)^2 \]

where \( g_m(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \), \( \eta^*_i = x_n - \tau_i (x^*_n - x_n) \), \( \tau_i \) are the knots, in \([0,1]\), and \( A_i \) are the weight of the quadrature formula.

In this paper, we will present new modification of Weerakoom-.Fernando [3] by using Newton divided-difference formulas to approximate the first derivative of the function \( f(x) \) at points \( x_{n+1} \) and \( x_n \), and obtain a new modification of Newton’s method.

**2- A new modification of Newton’s method and analysis of convergence**

To derive the modified, we using Newton backward-difference formula and Newton forward-difference formula to approximate the first derivative \( f'(x) \) at points \( x_{n+1} \) and \( x_n \) respectively of (3), we obtain

\begin{equation}
f'(x_{n+1}) = \frac{f(x_{n+1}) - f(x_n)}{x_{n+1} - x_n}, \tag{5}
\end{equation}

\begin{equation}
f'(x_n) = \frac{f(x_{n+1}) - f(x_n)}{x_{n+1} - x_n}
\end{equation}

thus, we have
\[ x_{n+1} = x_n - 2\frac{f(x_n)}{f(x_{n+1}) - f(x_n)} \left( \frac{f(x_{n+1}) - f(x_n)}{x_{n+1} - x_n} \right) \]

(6)

Obviously, this an implicit method, may be written in explicit method by replacing \( f(x_{n+1}) \) with \( f(x^*_{n+1}) \), so the new method is

\[ x_{n+1} = x_n - \frac{f(x_n)(x^*_{n+1} - x_n)}{f(x^*_{n+1}) - f(x_n)} \]

(7)

**Theorem 2.** Assume that the function \( f : D \subset R \rightarrow R \) has a simple root \( \xi \in D \), where \( D \) is an open interval. If \( f(x) \) has first, second and third derivatives in the interval \( D \), then the method defined by (7) converges cubically to \( \xi \) in a neighborhood of \( \xi \).

**Proof:** let \( \xi \) be a simple root of \( f(x) \) (i.e., \( f(\xi) = 0 \) and \( f'(\xi) \neq 0 \)) and \( e_n = x_n - \xi \).

Using the Taylor's expansion, we have

\[ f(x_n) = f(\xi) + f'(\xi)e_n + \frac{1}{2!}f''(\xi)e_n^2 + \frac{1}{3!}f'''(\xi)e_n^3 + O(e_n^4) \]

(8)

where \( C_i = \frac{f^{(i)}(\xi)}{i!} \).

Furthermore, we have

\[ f'(x_n) = f'(\xi)\left[ 1 + 2C_2 e_n^2 + 3C_3 e_n^3 + 4C_4 e_n^4 + O(e_n^5) \right] \]

(9)

Dividing (8) by (9)

\[ \frac{f(x_n)}{f'(x_n)} = \left[ e_n + C_2 e_n^2 + C_3 e_n^3 + O(e_n^4) \right] \left[ 1 + 2C_2 e_n^2 + 3C_3 e_n^3 + 4C_4 e_n^4 + O(e_n^5) \right]^{-1} \]

(10)

Using (10) we obtain

\[ x^*_n = x_n - \frac{f(x_n)}{f'(x_n)} \]

(11)

by considering (11) we have the Taylor's expansion of

\[ f(x^*_n) = f(\xi) + f'(\xi)\left( C_2 e_n^2 + \left( 2C_3 - 2C_2 \right) e_n^3 + O(e_n^4) \right) + \]

\[ \frac{1}{2!}f''(\xi)\left( C_2 e_n^2 + \left( 2C_3 - 2C_2 \right) e_n^3 + O(e_n^4) \right)^2 + O(e_n^5) \]

(12)

from equations (8) and (12)

\[ f(x^*_n) - f(x_n) = f'(\xi)\left[ - e_n + \left( C_3 - 2C_2 \right) e_n^3 + O(e_n^4) \right] \]

\[ \Rightarrow \]
Thus, equation (15) establishes the third-order convergence of method defined by (7).

3- Numerical examples

In this section we compare the behavior of the modified Newton's method (7) with three methods: classical Newton's method (2), the method Weerakoom - Fernando (3), and the method obtained by (the mid-point quadrature formula ) (4). We remark that chosen for comparison are only the methods which do not require the computation of second or higher derivative of the function to carry out iterations. All computations were done by C++.

<table>
<thead>
<tr>
<th>Table- 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( f(x_n) )</td>
<td>( x_0 )</td>
<td>C-NM</td>
<td>mNm (3)</td>
<td>mNm (4)</td>
<td>mNm (7)</td>
<td>Root</td>
</tr>
<tr>
<td>( x^3 + 4x^2 - 10 )</td>
<td>(-0.3)</td>
<td>54</td>
<td>108</td>
<td>6</td>
<td>18</td>
<td>18</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>(-0.25)</td>
<td>25</td>
<td>50</td>
<td>8</td>
<td>24</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>8</td>
<td>16</td>
<td>6</td>
<td>18</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>( x^{11} + x^5 + x - 1 )</td>
<td>(-3)</td>
<td>36</td>
<td>72</td>
<td>15</td>
<td>45</td>
<td>22</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>13</td>
<td>26</td>
<td>9</td>
<td>27</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>18</td>
<td>36</td>
<td>12</td>
<td>36</td>
<td>11</td>
<td>33</td>
</tr>
<tr>
<td>( \cos(x) - x )</td>
<td>(-10)</td>
<td>21</td>
<td>42</td>
<td>6</td>
<td>18</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>(-3)</td>
<td>24</td>
<td>48</td>
<td>4</td>
<td>12</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>(-2)</td>
<td>7</td>
<td>14</td>
<td>4</td>
<td>12</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>(-1.5)</td>
<td>38</td>
<td>76</td>
<td>7</td>
<td>21</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>9</td>
<td>27</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>( (x + 1)^2 e^{(x^2 - 2)} - 1 )</td>
<td>0.75</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>( (x - 1)^3 - 1 )</td>
<td>(-10)</td>
<td>19</td>
<td>38</td>
<td>8</td>
<td>24</td>
<td>12</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>16</td>
<td>32</td>
<td>4</td>
<td>12</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>12</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>( xe^{x^2} - \sin^2(x) + 3\cos(x) + 5 )</td>
<td>(-3)</td>
<td>15</td>
<td>30</td>
<td>9</td>
<td>27</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>12</td>
<td>4</td>
<td>12</td>
</tr>
</tbody>
</table>

- n/ Number of iterations to approximate the root to 15 decimal places.
- m/ Number of function evaluations.
- C-NM/ The classical Newton's method.
4- Conclusions

We have shown that by using Newton divided-difference formulas to approximate the first derivative of the function \( f(x) \) in Weerakoom-.Fernando formula obtained a new modification of Newton's method for solving non-linear equations, and proved that the order of convergence of the new method is three in the case of simple roots.

References