Modified Iterative Method for Solving Nonlinear Equation

Rostam K. Saeed
Shno O. Ahmed
Department of Mathematics / College of Science - Salahaddin University
received: 2010/3/12
Accepted: 2010/10/5

Abstract

In this paper, we present new one- and two-steps iterative methods for solving nonlinear equation \( f(x) = 0 \). It is proved here that the iterative methods converge of order three and six respectively. Several numerical examples are given to illustrate the performance and to show that the iterative methods in this paper give better result than the compared methods.

Introduction

Solving non-linear equations is one of the most important problems in numerical analysis. Multi-step iterative methods for finding solutions of nonlinear equations have been a constant interesting field of study in numerical analysis. Important research on these methods was carried out during the last ten years (Babajee, 2008). Several authors (Ostrowski 1973, Abbasbandy 2003, Noor 2007, Chun 2005, Kanwar et. al 2005, Noor & Ahmed 2006 and Noor et. al 2006) attempted to develop new higher order methods of higher efficiency index by adding finite evaluations of the function in the multi-step methods to obtain less iteration than the classical Newton method. Further, Noor and Ahmed (2006) derived a new two step iterative method of order four depending on an auxiliary equation, Newton’s method and by considering terms up to first-order in Taylor series. The purpose of this paper is to present one- and two- steps iterative methods having third and sixth order of convergence respectively in a similar manner of (Noor & Ahmed, 2006) by considering terms up to second-order in Taylor series.

Iterative methods

Consider the nonlinear equation
\[ f(x) = 0. \]...

Let \( \alpha \) be a simple root and \( x_0 \) be the initial guess known for the required root. To derive our iterative methods in a similar manner of (Noor & Ahmed, 2006), assume...
\[ x_i = x_0 + h, \quad |h| \leq 1, \quad \ldots (2) \]

be the first approximation to the root.

Consider the following auxiliary equation with a parameter \( p \):
\[ g(x) = p^3(x - x_0)^2 f(x)^2 - f(x) = 0, \quad \ldots (3) \]
where \( p \in \mathbb{R} \).

It is clear that the root of (1) is also the root of (3) and vice versa. If \( x_i = x_0 + h \) is the better approximation for the required root, then (3) gives
\[ p^3 h^2 f(x_0 + h)^2 - f(x_0 + h) = 0. \quad \ldots (4) \]

Expanding \( f(x_0 + h) \) by the Taylor’s theorem and simplifying, we get
\[ h = \frac{-2f(x_0)}{f'(x_0) \pm \sqrt{f''^2(x_0) + 4f(x_0)(p^3 f^2(x_0) - \frac{f^n(x_0)}{2})}}, \quad \ldots (5) \]
in which sign should be chosen to make the denominator largest in magnitude (see Noor and Ahmed, 2006).

We suggest the following one step iteration method for solving nonlinear equation \( f(x) = 0 \).

**Algorithm 1:** Here \( p \) is chosen so that \( f(n) \) and \( p \) have the same sign. For a given \( x_0 \), calculate \( x_1, x_2, \ldots \) such that
\[ x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n) \pm \sqrt{f''^2(x_n) + 4f(x_n)(p^3 f^2(x_n) - \frac{f^n(x_n)}{2})}}, \]
where sign is chosen such as to make the denominator largest in magnitude.

Now, by combining Algorithm 2.1 and the Newton’s method we obtain the following two step iterative method:

**Algorithm 2:** Here \( p \) is chosen so that \( f(n) \) and \( p \) have the same sign. For a given \( x_0 \), calculate \( x_1, x_2, \ldots \) such that
\[ y_n = x_n - \frac{2f(x_n)}{f'(x_n) \pm \sqrt{f''^2(x_n) + 4f(x_n)(p^3 f^2(x_n) - \frac{f^n(x_n)}{2})}}, \]
\[ x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}, \]
where sign is chosen such as to make the denominator largest in magnitude.

**Convergence analysis**

In this section, we consider the convergence analysis of iterative technique given by Algorithm 2.1 and Algorithm 2.2 by the following theorems respectively:
Theorem 1: Assume that the function $f: I \subseteq R \rightarrow R$ for an open interval $I$ is sufficiently differentiable near a simple root $\alpha \in I$. If $x_0$ is sufficiently closed to $\alpha$, then the one step iterative method defined by Algorithm 2.1 has third-order convergence.

Proof. The technique is given by

$$x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n) \pm \sqrt{f'^2(x_n) + 4f(x_n) \left( p^3 f^2(x_n) - \frac{f''(x_n)}{2} \right)}}, \quad \ldots(6)$$

From (6) (see [6] and [7]), we get

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) \left( 1 + \frac{f(x_n)}{f'^2(x_n)} \left( p^3 f^2(x_n) - \frac{f''(x_n)}{2} \right) \right)}, \quad \ldots(7)$$

If we let $e_n = x_n - \alpha$, then by Taylor’s expansion, we have

$$f(x_n) = f'(\alpha)[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + O(e_n^5)], \quad \ldots(8)$$

$$f'(x_n) = f'(\alpha)[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + O(e_n^5)], \quad \ldots(9)$$

and

$$f''(x_n) = f'(\alpha)[2c_2 + 6c_2 e_n^2 + 12c_4 e_n^2 + 20c_4 e_n^3 + 30c_6 e_n^4 + O(e_n^5)], \quad \ldots(10)$$

where

$$c_j = \frac{f^{(j)}(\alpha)}{j! f'(\alpha)} \quad \text{for } j=2, 3, \ldots.$$ 

From (8), (9) and (10) we have

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2 e_n^2 + (2c_2 - 2c_3)e_n^3 + (7c_2 c_3 - 4c_3^2 - 3c_4) e_n^4 + O(e_n^5), \quad \ldots(11)$$

$$\frac{f^3(x_n)}{f'^2(x_n)} = f'(\alpha)[e_n^3 - c_2 e_n^4 + O(e_n^5)], \quad \ldots(12)$$

and

$$\frac{f(x_n) f''(x_n)}{2 f'^2(x_n)} = c_2 e_n + 3(c_3 - c_2^2)e_n^2 + (8c_3^2 + 6c_4 - 14c_2 c_3) e_n^3 + 30c_6 e_n^4 + O(e_n^5). \quad \ldots(13)$$

From (11), (12) and (13) we have

$$\frac{f(x_n)}{f'(x_n) \left( 1 + \frac{f(x_n)}{f'^2(x_n)} \left( p^3 f^2(x_n) - \frac{f''(x_n)}{2} \right) \right)} = e_n + (c_3 - c_2^2) e_n^2 + (3c_3^2 - 6c_2 c_3 + 3c_4 - p^3 f'(\alpha)) e_n^3 + O(e_n^5). \quad \ldots(14)$$

From (7) and (14) we get

$$x_{n+1} = \alpha + (c_2^2 - c_3) e_n^3 + (6c_2 c_3 - 3c_3^2 - 3c_4 + p^3 f'(\alpha)) e_n^4 + O(e_n^5).$$

This implies that

$$e_{n+1} = (c_2^2 - c_3) e_n^3 + (6c_2 c_3 - 3c_3^2 - 3c_4 + p^3 f'(\alpha)) e_n^4 + O(e_n^5).$$

148
This shows that the method given in Algorithm 2.1 has third-order convergences. □

**Theorem 2:** Assume that the function \( f : I \subseteq \mathbb{R} \rightarrow \mathbb{R} \) for an open interval \( I \) is sufficiently differentiable near a simple root \( \alpha \in I \). If \( x_0 \) is sufficiently closed to \( \alpha \), then the one step iterative method defined by Algorithm 2.2 has sixth-order convergence.

**Proof.** The technique is given by

\[
y_n = x_n - \frac{2f(x_n)}{f'(x_n) + \sqrt{f'^2(x_n) + 4f(x_n)\left(p^3 f'^2(x_n) - \frac{f''(x_n)}{2}\right)}}.
\]  

...(15)

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

...(16)

From (15) we get

\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)\left(1 + \frac{f(x_n)}{f'^2(x_n)\left(p^3 f'^2(x_n) - \frac{f''(x_n)}{2}\right)}\right)}.
\]  

...(17)

From (14) and (17), we have

\[
y_n = \alpha + (c_2^2 - c_3)e_n^3 + (6c_2c_3 - 3c_2^3 - 3c_4 + p^3 f'(\alpha))e_n^4 + O(e_n^5).
\]  

...(18)

Also expanding \( f(y_n) \) and \( f'(y_n) \) about \( \alpha \), we have

\[
f(y_n) = f'(\alpha)[(c_2^2 - c_3)e_n^3 + (6c_2c_3 - 3c_2^3 - 3c_4 + p^3 f'(\alpha))e_n^4 + O(e_n^5)].
\]  

...(19)

\[
f'(y_n) = f'(\alpha)[1 + 2(c_2^2 - c_3)c_2e_n^3 + 2(6c_2c_3 - 3c_2^3 - 3c_4 + p^3 f'(\alpha))c_2e_n^4 + O(e_n^5)].
\]  

...(20)

From (16), (18), (19) and (20) we have

\[
x_{n+1} = \alpha + (c_2c_3^2 - 2c_3c_2^3 + c_2^5)e_n^6 + O(e_n^7).
\]

This implies that

\[
e_{n+1} = (c_2c_3^2 - 2c_3c_2^3 + c_2^5)e_n^6 + O(e_n^7).
\]

This shows that the two step iterative method given in Algorithm 2.2 has sixth-order convergences. □

**Numerical examples**

We now present some examples to illustrate the efficiency of our developed methods in this paper which are given by the Algorithm 2.1 and Algorithm 2.2. We compare the Newton’s method (NM) Ostrowski (1973), the method of Noor and Ahmed (2006) (NA) and the methods of Noor et al. (2006) (NRM), with the iterative methods introduced in this paper. We used \( \varepsilon = 10^{-15} \). The following stopping criteria are used for computer programs:

\( (i) \ |x_{n+1} - x_n| < \varepsilon. \)
The following examples are used for numerical testing, see (Noor and Ahmed (2006)) and (Noor et. al (2006)):

\[ f_1(x) = (x-1)^3 - 1, \quad \alpha = 2. \]
\[ f_2(x) = x^3 - 10, \quad \alpha = 2.1544346900318837217592935665. \]
\[ f_3(x) = \sin^2(x) - x^2 + 1, \quad \alpha = 1.4044916482153412260350868178. \]
\[ f_4(x) = e^x + 2^{-x} + 2\cos(x) - 6, \quad \alpha = 1.8293836019338488171362129468. \]
\[ f_5(x) = x^3 + 4x - 10, \quad \alpha = 1.3652300134140968457608068290. \]
\[ f_6(x) = 1 - x \cos(x) + \cos(x), \quad \alpha = -1.5707963267948966192313216916. \]
\[ f_7(x) = x^3 - x^2 - 1, \quad \alpha = 1.4655712318767680266567312252. \]
\[ f_8(x) = \tan^{-1}(x) + \sin(x) - 2, \quad \alpha = 0.718586769063581588178526552270138. \]
\[ f_9(x) = x^2 - (1-x)^5, \quad \alpha = 0.34595481584824201795820440645. \]

Table 1: Comparison of various iterative methods

<table>
<thead>
<tr>
<th>( f(x) )</th>
<th>( x_0 )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( x_6 )</th>
<th>( x_7 )</th>
<th>( x_8 )</th>
<th>( x_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>2.5</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_2 )</td>
<td>1.5</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_3 )</td>
<td>-2.3</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-2</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_4 )</td>
<td>1.5</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_5 )</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_6 )</td>
<td>-1</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_7 )</td>
<td>1.8</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_8 )</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_9 )</td>
<td>0.3</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion**

From Table 1, we conclude that the one-step iterative method (Algorithm 1) performs better than the Newton's method and two-steps
iterative method (Algorithm 2) performs better than the Newton's method and the methods presented by (Noor & Ahmed, 2006 and Noor et al., 2006).

References


الطريقة التكرارية المُعدلة لحل المعادلة اللاخطية

روستم كريم سعيد
قسم الرياضيات / كلية العلوم - جامعة صلاح الدين
قسن الرياضيات
تاريخ الاقتراح: 5/10/2010
تاريخ القبول: 12/3/2010

الخلاصة

في هذا البحث، نُقدم طرق تكرارية جديدة ذات خطوة واحدة وخطوتين لحل المعادلة اللاخطية 0 = f(x).

وأثبتنا بأن الطرق التكرارية مقارنة من المرتبة الثالثة والسادسة على التوالي. وقد أعطينا عدداً أمثلة عددية
لبيان كفاءة وفضلية طرائقنا على الطرق التي تم مقارنتها.