SOME MULTI-STEP ITERATIVE SCHEMES FOR SOLVING NONLINEAR EQUATIONS

ARIF RAFIQ\textsuperscript{a}, AYESHA INAM PASHA\textsuperscript{b} AND BYUNG-SOO LEE\textsuperscript{c,*}

Abstract. In this paper, we suggest and analyze a family of multi-step iterative methods which do not involve the high-order differentials of the function for solving nonlinear equations using a different type of decomposition (mainly due to Noor and Noor [15]). We also discuss the convergence of the new proposed methods. Several numerical examples are given to illustrate the efficiency and the performance of the new iterative method. Our results can be considered as an improvement and refinement of the previous results.

1. Introduction

In recent years, much attention has been given to develop several iterative methods for solving nonlinear equations (see for example [1, 6-16]). These methods can be classified as one-step and two-step methods.

Abbasbandy [1] and Chun [6] have proposed and studied several one-step and two-step iterative methods with higher order convergence by using the decomposition technique of Adomian [2-5].

In their methods, they have used the higher order differential derivatives which is a serious drawback. To overcome this drawback, Noor and Noor [14-15] developed two-step and three step iterative methods by combining the well-known Newton method with other one-step and two-step methods.

Following the lines of [14-15], we suggest and analyze a family of multi-step iterative methods which do not involve the high-order differentials of the function for solving nonlinear equations using a different type of decomposition (mainly due to Noor and Noor [15]). We also discuss the convergence of the new proposed methods.
Several numerical examples are given to illustrate the efficiency and the performance of the new iterative method. Our results can be considered as an improvement and refinement of the previous results.

2. Iterative Methods

Consider the nonlinear equation

\[ f(x) = 0. \]

(1)

We assume that \( \alpha \) is a simple root of (1) and \( \beta \) is an initial guess sufficiently close to \( \alpha \). We can rewrite (1) as a coupled system using the Taylor series

\[ f(\beta) + (x - \beta)f'(\beta) + g(x) = 0, \]

(2)

\[ g(x) = f(x) - f(\beta) - (x - \beta)f'(\beta). \]

(3)

We can rewrite (3) in the following form

\[ x = \beta - \frac{f(\beta)}{f'(\beta)} - \frac{g(x)}{f'(\beta)} \]

(4)

\[ = c + N(x), \]

(5)

where

\[ c = \beta - \frac{f(\beta)}{f'(\beta)} \]

(6)

and

\[ N(x) = -\frac{g(x)}{f'(\beta)}. \]

(7)

In order to prove the multi-step iterative methods, He [9] and Lao [11] have considered the case with the definition that

\[ g(x_0) = 0, \]

(8)

and Noor and Noor [15] have considered the case

\[ f(x_0) = g(x_0). \]

(9)

For the derivation of multi-step iterative methods for solving nonlinear equations, the condition (9) introduced by Noor and Noor [15] which is actually

\[ f(\sum_{i=0}^{\infty} x_i) = g(\sum_{i=0}^{\infty} x_i) \]
is the stronger one. We rectify this error and also remove such kind of conditions. For this purpose, we substitute (3) into (7) to obtain

\[ N(x) = x - \beta - \frac{f(x)}{f'(\beta)} + \frac{f(\beta)}{f'(\beta)}. \]  

(11)

We now construct a sequence of higher order iterative methods by using the following decomposition method which is mainly due to Noor and Noor [15]. This decomposition of the nonlinear operator \( N(x) \) is quite different than that of Adomian decomposition. The main idea of this technique is to look for a solution of (4) having the series form

\[ x = \sum_{i=0}^{\infty} x_i. \]  

(12)

The nonlinear operator \( N \) can be decomposed as

\[ N(x) = N(\sum_{i=0}^{\infty} x_i) = N(x_0) + \sum_{i=1}^{\infty} \left[ N\left( \sum_{j=0}^{i} x_j \right) \right]. \]  

(13)

Combining (4), (12) and (13), we have

\[ x = c + N(x_0) + \sum_{i=1}^{\infty} \left[ N\left( \sum_{j=0}^{i} x_j \right) \right]. \]  

(14)

Thus we have the following iterative scheme

\[ x_0 = c, \]
\[ x_1 = N(x_0), \]
\[ x_2 = N(x_0 + x_1), \]
\[ x_3 = N(x_0 + x_1 + x_2), \]
\[ \vdots \]
\[ x_{n+1} = N(x_0 + x_1 + \cdots + x_n); \quad n = 1, 2, \ldots. \]  

(15)

Then

\[ x_1 + x_2 + \cdots + x_{n+1} = N(x_0) + N(x_0 + x_1) + N(x_0 + x_1 + x_2) + \cdots + N(x_0 + x_1 + x + \cdots + x_n); \quad n = 1, 2, \ldots \]  

(16)
and

\[ x = c + \sum_{i=1}^{\infty} x_i. \]  

(17)

It follows from (6), (11) and (15), that

\[ x_0 = c = \beta - \frac{f(\beta)}{f'(\beta)} \]

(18)

and

\[ x_1 = N(x_0) = -\frac{f(x_0)}{f'(\beta)}. \]

(19)

From (14), (16) and (17) we have

\[ x \approx c = x_0 = \beta - \frac{f(\beta)}{f'(\beta)}. \]

This allows us to suggest the following one-step iterative method for solving (1).

**Algorithm 1**

For a given \( x_0 \), compute the approximate solution \( x_{n+1} \) by the iterative scheme

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0, \quad n = 0, 1, 2, \ldots \]

which is known as “Newton’s Method” and it has the second order convergence.

Again using (15), (17)-(19), we conclude that

\[ x \approx c + x_1 = x_0 + N(x_0) \]

\[ = \beta - \frac{f(\beta)}{f'(\beta)} - \frac{f(x_0)}{f'(\beta)}. \]

Using this relation, we can suggest the following two-step iterative methods for solving (1).

**Algorithm 2**

For a given \( x_0 \), compute the approximate solution \( x_{n+1} \) by the iterative scheme

**Predictor-Step**

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0, \]

**Corrector-Step**

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

This Algorithm is commonly known as “Double-Newton Method” with the third order convergence.
Again using (11) and (15), we can calculate
\[ x_2 = N(x_0 + x_1) \]
(20)
\[ = -\frac{f(x_0)}{f'(\beta)} - \frac{f(x_0 + x_1)}{f'(\beta)}. \]
From (11), (15)-(20), we conclude that
\[ x \approx c + x_1 + x_2. \]
\[ = x_0 + N(x_0) + X(x_0 + x_1). \]
\[ = \beta - \frac{f(\beta)}{f'(\beta)} - 2\frac{f(x_0)}{f'(\beta)} - \frac{f(x_0 + x_1)}{f'(\beta)}. \]
Using this, we can suggest and analyze the following two-step iterative method for solving (1).

**Algorithm 3 (AA)** For a given \( x_0 \), compute the approximate solution \( x_{n+1} \) by the iterative scheme

**Predictor-Step**

(21) \[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, f'(x_n) \neq 0, \ n = 0, 1, 2, \ldots. \]

(22) \[ z_n = -\frac{f(y_n)}{f'(x_n)}, \]

**Corrector-Step**

(23) \[ x_{n+1} = y_n + 2z_n - \frac{f(y_n + z_n)}{f'(x_n)}. \]

Again using (11) and (15), we have
\[ x_3 = N(x_0 + x_1 + x_2) \]
(20)
\[ = -\frac{f(x_0)}{f'(\beta)} - \frac{f(x_0 + x_1)}{f'(\beta)} - \frac{f(x_0 + x_1 + x_2)}{f'(\beta)}. \]
From (11), (17)-(20), we have
\[ x \approx c + x_1 + x_2 + x_3 \]
\[ = \beta - \frac{f(\beta)}{f'(\beta)} - 2\frac{f(x_0)}{f'(\beta)} - 2\frac{f(x_0 + x_1)}{f'(\beta)} - \frac{f(x_0 + x_1 + x_2)}{f'(\beta)}. \]
Using this, we can suggest and analyze the following iterative method for solving (1).
Algorithm 4 For a given $x_0$, compute the approximate solution $x_{n+1}$ by the iterative scheme

**Predictor-Step**

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0, \quad n = 0, 1, 2, \ldots. \]  

(24)

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

(25)

\[ w_n = -\frac{f(y_n + z_n)}{f'(x_n)}, \]  

(26)

**Corrector-Step**

\[ x_{n+1} = y_n + 2z_n + 2w_n - \frac{f(y_n + z_n + w_n)}{f'(x_n)}. \]  

(27)

3. **Convergence Analysis**

**Theorem 1.** Let $\beta \in I$ be a simple zero of a sufficiently differentiable function $f : I \subseteq \mathbb{R} \to \mathbb{R}$ for an open interval $I$. If $x_0$ is sufficiently close to $\beta$, then the three-step iterative method defined by Algorithm 3 has the second-order convergence.

**Proof.** Let $\beta \in I$ be a simple zero of $f$. Since $f$ is sufficiently differentiable function, by expanding $f(x_n)$ and $f'(x_n)$ about $\beta$, we get

\[ f(x_n) = f'(\beta)[e_n + c_2 e_n^2 + c_3 e_n^3 + \cdots], \]  

(28)

\[ f'(x_n) = f'(\beta)[12c_2 e_n + 3c_3 e_n^2 + \cdots], \]  

(29)

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

Now from (28) and (29), we have

\[ \frac{f(x_n)}{f'(x_n)} = e_n - c_2 e_n^2 - 2(c_3 + c_2^2)e_n^3 - (3c_4 + 5c_2c_3)e_n^4 + \cdots. \]  

(30)

From (18) and (30), we have

\[ y_n = \beta + c_2 e_n^2 + 2(c_3 + c_2^2)e_n^3 + (3c_4 + 5c_2c_3)e_n^4 + \cdots. \]  

(31)

Now expanding $f(y_n)$ about $\beta$ and using (31), we get

\[ f(y_n) = f'(\beta)[c_2 e_n^2 + 2(c_3 + c_2^2)e_n^3 + (3c_4 + 5c_2c_3)e_n^4 + \cdots]. \]  

(32)

Now from (29) and (32), we have

\[ z_n = -[c_2 e_n^2 + (2c_3 + 4c_2^2)e_n^3 + (3c_4 + 12c_2c_3 + 4c_2^3)e_n^4 + \cdots]. \]  

(33)
Now again expanding \( f(y_n + z_n) \) about \( \beta \) and using (31) and (33), we have

\[
(34) \quad f(y_n + z_n) = f'(\beta)[-2c_2^2e_n^3 + (-4c_2^3 - 7c_2c_3)e_n^4 + \cdots].
\]

From (29) and (34), we get

\[
(35) \quad \frac{f(y_n + z_n)}{f'(x_n)} = 2c_2^2e_n^3 + 7c_2c_3e_n^4 + \cdots.
\]

From (23), (33) and (35), one obtains

\[
(36) \quad e_{n+1} = c_2e_n^2 + 2(c_3 + 4c_2e_n^3 + \cdots.
\]

Hence it is proved.

**Theorem 2.** Let \( \beta \in I \) be a simple zero of a sufficiently differentiable function \( f : I \subseteq \mathbb{R} \to \mathbb{R} \) for an open interval \( I \). If \( x_0 \) is sufficiently close to \( \beta \), then the four-step iterative method defined by Algorithm 4 has the second-order convergence.

**Proof.** From (26) and (35), we have

\[
(37) \quad w_n = 2c_2^2e_n^3 + 7c_2c_3e_n^4 + \cdots.
\]

Now again expanding \( f(y_n + z_n + w_n) \) about \( \beta \) and using (23), (25) and (37), we get

\[
(38) \quad f(y_n + z_n + w_n) = f'(\beta)[-4c_2^3e_n^4 + (-17c_2^2c_3 - 8c_2^4)e_n^5 + \cdots].
\]

From (29) and (38), we have

\[
(39) \quad \frac{f(y_n + z_n + w_n)}{f'(x_n)} = -4c_2^3e_n^4 - 17c_2^2c_3e_n^5 + \cdots.
\]

From (27), (31), (33), (37) and (39), one obtains

\[
(40) \quad e_{n+1} = 4c_2e_n^2 + (8c_3 + 24c_2^2)e_n^3 + \cdots.
\]

Hence it is proved.

## 4. Numerical Examples

We present some examples to illustrate the efficiency of the new developed three-step iterative methods. We compare the Newton method (NM), the method (NR1) \[14\], the method (NR2) \[15\] and the method (AA). Put \( \epsilon = 10^{-15} \).

The following stopping criteria is used for computer programs

\[
(1) \quad |x_{n+1} - x_n| < \epsilon,
\]

\[
(2) \quad |f(x_{n+1})| < \epsilon.
\]
As for the convergence criteria, it was required that the distance of two consecutive approximations $\delta$ for the zero was less than $10^{-15}$. Also displayed are the number of iterations (IT) to approximate the zero, the approximate zero $x_0$ and the value $f(x_0)$ and $\delta$ (see Table 1).

The examples are the same as in Chun [6]:

\begin{align*}
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) &= x \cdot e^x - \sin^2 x + 3 \cos x + 5, \\
F_7(x) &= e^{x^2 + 7x - 30} - 1.
\end{align*}

<table>
<thead>
<tr>
<th>$F_i$ : $x_0$</th>
<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>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>7</td>
<td>1.404491648215341</td>
<td>3.331e-16</td>
<td>3.059e-13</td>
</tr>
<tr>
<td>NR1</td>
<td>6</td>
<td>1.404491648215341</td>
<td>-8.882e-16</td>
<td>2.22e-16</td>
</tr>
<tr>
<td>NR2</td>
<td>8</td>
<td>1.404491648215341</td>
<td>-4.441e-16</td>
<td>7.349e-14</td>
</tr>
<tr>
<td>AA</td>
<td>8</td>
<td>1.404491648215341</td>
<td>3.331e-16</td>
<td>4.244e-11</td>
</tr>
</tbody>
</table>

| $F_2$ : $x_0 = 2$ |     |            |                |             |
| NM    | 6   | 0.257530285439861 | 0              | 9.864e-14  |
| NR1   | 5   | 0.257530285439861 | 0              | 2.209e-12  |
| NR2   | 7   | 0.257530285439861 | -4.409e-16    | 5.884e-15  |
| AA    | 6   | 0.257530285439861 | -4.41e-16     | 8.109e-08  |

| $F_3$ : $x_0 = 1.7$ |     |            |                |             |
| NM    | 5   | 0.739085133215161 | -4.41e-16     | 3.259e-08  |
| NR1   | 4   | 0.739085133215161 | 0              | 1.849e-08  |
| NR2   | 6   | 0.739085133215161 | 0              | 7.627e-14  |
| AA    | 5   | 0.739085133215161 | 1.1102e-16    | 1.827e-08  |

| $F_4$ : $x_0 = 3.5$ |     |            |                |             |
| NM    | 8   | 2           | 0              | 2.878e-11  |
| NR1   | 7   | 2           | 0              | 2.2204e-15 |
| NR2   | 6   | 2           | 0              | 6.702e-11  |
| AA    | 7   | 2           | 0              | 2.22e-15   |
5. Conclusions

We have suggested a family of one-step, two-step, three-step and four-step iterative methods for solving nonlinear equations. It is important to note that the implementation of these multi-step methods does not require the computation of higher order derivatives compared to most other methods of the same order.

References


*Department of Mathematics, Lahore Leads University, Lahore, Pakistan  
Email address: aarafiq@gmail.com*

*bCIIT, Lahore, Pakistan  
Email address: ayeshams20@gmail.com*

*cDepartment of Mathematics, Kyungsung University, Busan 608-736, Korea  
Email address: bslee@ks.ac.kr*