ITERATIVE METHOD FOR SOLVING NONLINEAR FUNCTIONS WITH CONVERGENCE OF ORDER FOUR

M. Saqib Khan 1, Afshan Nazir 2 and Waqas Nazeer 3

1 Department of Mathematics, Lahore Leads University, Lahore-Pakistan
E-mail: khan.saqib@me.com

2 Department of Mathematics, Lahore Leads University, Lahore-Pakistan
E-mail: afshanmalik173@gmail.com

3 Division of Science and Technology, University of Education, Lahore, Pakistan
E-mail: nazeer.waqs@ue.edu.pk

ABSTRACT: In this paper, we describe an iterative method for solving nonlinear functions and analyzed. The iterative method is modification of Golbabai and Javidi’s method and has convergence of order four. This iterative method converges faster than Newton’s method, Halley’s method and Householder’s method. The comparison tables for different test function demonstrate the faster convergence of this method.

AMS Subject Classification: 65H05.
Key words: Fixed point method, Newton Raphson’s Method, nonlinear equations, convergence analysis.

1. INTRODUCTION

One of the most frequently problems in Sciences and more specifically in Mathematics is solving a nonlinear equation.

\[ f(x) = 0 \]  

(1.1)

with \( f: D \subseteq \mathbb{R} \rightarrow \mathbb{R} \), where D is an open connected set. Except special cases, the solutions of these kinds of equations cannot be obtained in a direct way. That is why, most of the methods for solving these equations are iterative.

Equation (1.1) is solvable iteratively by Newton’s method and a range of its variants [13] as well as by other techniques. The Newton's method, defined by

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad n \geq 0 \]  

(1.2)

converges quadratically in some neighborhood of \( \alpha \).

Some Newton-type methods, with third-order convergence, that do not require the computation of second-order derivatives, have been developed in [2, 3, 7, 9, 12, 14, 15]. Other classes of those iterative methods invoke the Adomian decomposition method as in [1]; He’s homotopy perturbation method [5] and Liao’s homotopy analysis method [2]. One class of those methods have been derived based on quadrature formulas for the computation of the integral

\[ f(x) = f(x) + \int_{x_0}^{x} f'(t)dt \]  

(1.3)

arising from Newton's theorem. In [14], by solving (1.3), Weerakoon and Fernando derived following modified Newton’s method.

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

(1.4)

which converge cubically. In [3, 12], solving (1.3), the authors yields the following

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

(1.5)

Method (1.5) has also been derived by Homeier in [7]. A further multivariate version of this method has been discussed in [4, 6].

By applying Newton's theorem to the inverse function \( x = f(y) \), in [7], Homeier derived the following cubically convergent iteration scheme:

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

(1.6)

The method leading to (1.6) has also been derived in [12]. Finally, in [9], Kou, et al. considered Newton’s theorem on a new interval of integration and arrived at the following cubically convergent iteration scheme

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

(1.7)

Any of the aforementioned methods require only first order derivative of the given function. The iterative methods with a higher-order convergence are important which do not require second derivative from the practical point of view and is an area of current active research.

During the last century, the numerical techniques for solving nonlinear equations have been successfully applied (see, e. g., [2-25] and the references therein). The order of convergence for a sequence of approximations derived from iteration method is defined in the literature, as
Let \( \{x_n\} \) converges to \( \alpha \), if there exist an integer constant \( p \), and a real positive constant \( C \) such that
\[
\lim_{n \to \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} = C,
\]
then \( p \) is called the order and \( C \) the constant of convergence.

To determine the order of convergence of the sequence \( \{x_n\} \), let us consider the Taylor expansion of \( g(x) \)
\[
g(x) = \sum_{k=0}^{\infty} \frac{g^{(k)}(x)}{k!} (x - \alpha)^k + \ldots
\]

And we can state the following result.

**Theorem 1.1**

Suppose \( g(x) \in C^p[a,b] \). If \( g^{(k)}(x) = 0 \), for \( k=1,2,\ldots, p-1 \) and \( g^{(p)}(x) \neq 0 \), then the sequence \( \{x_n\} \) is of order \( p \).

In [17] we have

**Algorithm 1.1.**

For a given \( x_0 \), we can calculate the approximation solutions \( x_{n+1} \) by the iterative scheme given by Golbabai and Javidi
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n) f''(x_n)}{2 (f^3(x_n) - f(x_n) f'(x_n) f''(x_n))},
\]

For \( k \geq 1 \), let
\[
x_{k+1} = \phi_{k+1}(x_k) = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{3 f(x_k) f''(x_k) + 2 f(x_k) f''(x_k)}{2 (f^3(x_k) - f(x_k) f'(x_k) f''(x_k))},
\]

On simplification, we have
\[
x_{k+1} = x_k + \frac{3 f(x_k) f'(x_k) (-2 f'(x_k)^2 + f(x_k) f''(x_k)) (f'(x_k)^2 + f(x_k) f''(x_k))}{-6 f'(x_k)^6 + 12 f(x_k) f'(x_k)^3 f''(x_k) - 6 f'(x_k)^2 f'(x_k)^2 f''(x_k) + f(x_k)^3 f''(x_k) + f(x_k)^3 f''(x_k) + f(x_k)^2 f''(x_k)^3 f''(x_k)}
\]

**Algorithm 2.1**

For a given \( x_0 \), we can calculate the approximation solutions \( x_{n+1} \) by the iterative scheme given by
\[
x_{n+1} = x_n + \frac{3 f(x_n) f'(x_n) (-2 f'(x_n)^2 + f(x_n) f''(x_n)) (f'(x_n)^2 + f(x_n) f''(x_n))}{-6 f'(x_n)^6 + 12 f(x_n) f'(x_n)^3 f''(x_n) - 6 f'(x_n)^2 f'(x_n)^2 f''(x_n) + f(x_n)^3 f''(x_n) + f(x_n)^3 f''(x_n) + f(x_n)^2 f''(x_n)^3 f''(x_n)}
\]

**3. Convergence Analysis**

In this section, we show that the convergence of our iterative method is at least four.

**Theorem 3.1**

Let \( f:D \subseteq \mathbb{R} \to \mathbb{R} \) be a scalar function defined on \( D \). Let the nonlinear equation \( f(x) = 0 \) has a simple root \( \alpha \in D \), such that \( f \) be sufficiently smooth in the neighborhood of \( \alpha \), then the convergence order of the iterative method (2.1) (Modified Golbabai and Javidi’s method MGJM) is at least four.
Proof.
To analyze the convergence of the Modified Golbabai and Javidi’s method MGJM, let
\[ H(x) = x + \frac{3f(x)f'(x)(-2f'(x)^2 + f(x)f''(x))(-f'(x)^2 + f(x)f''(x))}{-6f'(x)^6 + 12f(x)f'(x)^4 f''(x) - 6f'(x)^2 f''(x)^2 f(x)^2 + f(x)^3 f''(x)^3 - f'(x)^2 f(x)^3 f^{(3)}(x)} \]

Let \( \alpha \) be a simple zero of \( f \) i.e. \( f(\alpha) = 0 \), then by using the following commands in Mathematica, we get the following results:

Now, from the above equation, it can easily be seen that \( H^{(4)}(\alpha) \neq 0 \). Then, according to theorem 1.1, algorithm 2.1 (MGJM), has fourth order convergence.

4. Applications
In this section, we included some test functions to illustrate the efficiency of our developed Modified Golbabai and Javidi’s method (MGJM). We compare MGJM with Newton-Raphson method (NR) and Golbabai and Javidi’s method (GJM) as shown in the examples below.

Example 4.1. Consider \( f(x) = 1 - x^2 + \sin(x)^2 \) with initial guess \( x_0 = 2.5 \). The following table shows that NRM gives root after 4 iterations; GJM gives root after 3 iterations, while our Algorithm 2.1 converges after 2 iterations.

\[
\begin{array}{cccc}
\text{n} & \text{NR} & \text{GJM} & \text{MGJM} \\
1 & 2 & 3 & 4 \\
1 & 1.679074801549118 & 1.578139659937841 & 1.45927001366479 \\
2 & 1.446506508098619 & 1.40467401425747 & 1.40448936452083 \\
3 & 1.40579955812068 & 1.40491648214807 & 1.404492987075196 \\
4 & 1.404492987075196 & 1.404492987075196 & 1.404492987075196 \\
\end{array}
\]

Example 4.2. Consider \( f(x) = \ln(x) + x \) with initial guess \( x_0 = 1.5 \). The following table shows that NRM gives root after 4 iterations; GJM gives root after 3 iterations, while our Algorithm 2.1 converges after 2 iterations.

\[
\begin{array}{cccc}
\text{n} & \text{NR} & \text{GJM} & \text{MGJM} \\
1 & 2 & 3 & 4 \\
1 & 0.356720935135103 & 0.490280354374409 & 0.536770220033134 \\
2 & 0.539560092058768 & 0.5675535797293318 & 0.567142454928770 \\
3 & 0.564975664109221 & 0.5671432903641805 & 0.5671430556594482 \\
4 & 0.5671430556594482 & 0.5671430556594482 & 0.5671430556594482 \\
\end{array}
\]

Example 4.3. Consider \( f(x) = e^x - 5x^2 \) with initial guess \( x_0 = 2 \). The following table shows that NRM gives root after 5 iterations; GJM gives root after 3 iterations, while our Algorithm 2.1 converges after 2 iterations.

\[
\begin{array}{cccc}
\text{n} & \text{NR} & \text{GJM} & \text{MGJM} \\
1 & 2 & 3 & 4 \\
1 & 0.6866511285126958 & 0.608288762160259 & 0.6052657597005336 \\
2 & 0.6107413432557138 & 0.605267123389045 & 0.605267123389045 \\
3 & 0.605267123389045 & 0.605267123389045 & 0.605267123389045 \\
4 & 0.605267123389045 & 0.605267123389045 & 0.605267123389045 \\
5 & 0.605267123389045 & 0.605267123389045 & 0.605267123389045 \\
\end{array}
\]

Example 4.4. Consider \( f(x) = x^3 - 10 \) with initial guess \( x_0 = 6 \). The following table shows that NRM gives root after 6 iterations; GJM gives root after 3 iterations, while our Algorithm 2.1 converges after 2 iterations.

\[
\begin{array}{cccc}
\text{n} & \text{NR} & \text{GJM} & \text{MGJM} \\
1 & 2 & 3 & 4 \\
1 & 4.092502592592593 & 2.42765228499686 & 2.162971797390206 \\
2 & 2.927408185947801 & 2.13442890105894 & 2.15443468956685 \\
3 & 2.340571939749109 & 2.154434690102043 & 2.15443468956685 \\
4 & 2.168845109472783 & 2.168845109472783 & 2.168845109472783 \\
\end{array}
\]
**Example 4.5.** Consider \( f(x) = x^3 + 4x^2 + 8x + 8 \) with initial guess \( x_0 = 0.4 \). The following table shows that NRM gives root after 5 iterations; GJM gives root after 4 iterations while our Algorithm 2.1 converges after 3 iterations.

<table>
<thead>
<tr>
<th>n</th>
<th>NR</th>
<th>GJM</th>
<th>MGJM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.61917808219178</td>
<td>-5.61790547116176</td>
<td>2.30284751472449</td>
</tr>
<tr>
<td>2</td>
<td>-1.65396717530791</td>
<td>-2.48449618959117</td>
<td>-2.021391519849168</td>
</tr>
<tr>
<td>3</td>
<td>-2.05264064301376</td>
<td>-1.981169265978878</td>
<td>-1.99999989545223</td>
</tr>
<tr>
<td>4</td>
<td>-2.00138278854393</td>
<td>-2.000001684751276</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-2.00000095605071</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example 4.6.** Consider \( f(x) = x^3 + x^2 - 2 \) with initial guess \( x_0 = 6 \). The following table shows that NRM gives root after 7 iterations; GJM gives root after 3 iterations while our Algorithm 2.1 converges after 3 iterations.

<table>
<thead>
<tr>
<th>n</th>
<th>NR</th>
<th>GJM</th>
<th>MGJM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.916666666666666</td>
<td>1.897108843537415</td>
<td>1.436518042838706</td>
</tr>
<tr>
<td>2</td>
<td>2.553298946915968</td>
<td>0.994380369689159</td>
<td>0.99777138162588</td>
</tr>
<tr>
<td>3</td>
<td>1.695180663693167</td>
<td>1.000000036007899</td>
<td>0.9999999999767</td>
</tr>
<tr>
<td>4</td>
<td>1.216882334553059</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.030330127286111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.00071214745941</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.000000405399111</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 CONCLUSIONS

A new MFPI for solving nonlinear functions has been established. We can conclude from table that

1. The modified MFPI has a convergence of order two.
2. By using some examples the performance of MFPI is also discussed. The MFPI is performing very well in comparison to FPM as discussed in Table below.

REFERENCES

[17] A. Golbabai, M. Javid, Department of Mathematics, Iran University of Science and Technology, Narmak, Tehran 16844, Iran.

Jan.-Feb


