A modification on Newton's method for solving systems of nonlinear equations

Jafar Biazar, Behzad Ghanbari

Abstract—In this paper, we are concerned with the further study for system of nonlinear equations. Since systems with inaccurate function values or problems with high computational cost arise frequently in science and engineering, recently such systems have attracted researcher's interest. In this work we present a new method which is independent of function evaluations and has a quadratic convergence. This method can be viewed as a extension of some recent methods for solving mentioned systems of nonlinear equations. Numerical results of applying this method to some test problems show the efficiently and reliability of method.

I. INTRODUCTION

Consider a system of nonlinear equations

\[ \begin{align*}
  f_1(x_1, x_2, \ldots, x_n) &= 0, \\
  f_2(x_1, x_2, \ldots, x_n) &= 0, \\
  \vdots \\
  f_n(x_1, x_2, \ldots, x_n) &= 0.
\end{align*} \]  

(1)

This system can be referred by \( F(x) = 0 \), where \( F = (f_1, f_2, \ldots, f_n) : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n \) is continuously differentiable on an open neighborhood \( D^* \subset D \) of a solution \( x^* = (x_1^*, \ldots, x_n^*) \in D \) of the system (1). Each function \( f_i \) maps a vector \( x = (x_1, x_2, \ldots, x_n) \) from the \( n \)-dimensional space \( \mathbb{R}^n \) into \( \mathbb{R} \). We assume that the system (1) admits a unique solution. The most known iterative method for solving systems of nonlinear equations is the classical Newton's method, given by

\[ x^{p+1} = x^p - F'(x^p)^{-1} F(x^p), \quad p \geq 0 \]

(2)

where \( F'(x^p) \) denote the Jacobian matrix at the current approximation \( x^p = (x_1^p, x_2^p, \ldots, x_n^p) \) and \( x^{p+1} \) is the next approximation. In general, there exists no method that yields an exact solution for such equations. In recent years, considerable interest in system of nonlinear equation has been stimulated due to their numerous applications in the areas of science and engineering [1] and many powerful methods have been presented [6–29]. For example by using essentially Taylor's polynomial [1,2], decomposition [3], quadrature formulas [4,5] and other techniques [6–10].

In real life applications, there exist many problems where the system is known with some precision only, e.g. when the function and derivative values depend on the results of numerical simulations [11] or the precision of the desired function is available at a prohibitive cost, for example where function value results from the sum of an infinite series (e.g. Bessel or Airy functions [12,13,14]). So, it is very important to obtain methods which are function evaluations free. These methods are ideal for situations with unavailable accurate function values or high computational cost. In this direction, several methods have been proposed for example in [15] a method proposed which applied for polynomial only.

Also, there are some methods where, the function values in Newton's method are not directly evaluated from the corresponding component functions \( f_i(x) \), but are approximated by using appropriate quantities, which called WFEN method [16] and IWFEN [17].

This paper is structured as follows. In Section 2, a brief outline of the WFEN, IWFEN methods for systems of nonlinear equations has discussed. In Section 3 we modify some presented notations, in [16-17] and by using a geometrical interpretation, a method which can be viewed as a new improved Newton's method without direct function evaluations is presented. Some numerical examples are stated in Section 4 and a comparison between proposed method and WFEN, IWFEN, Newton's methods on these examples is given. Finally, Conclusions are drawn in Section 5.

II. WFEN AND IWFEN METHODS

For \( i = 1, 2, \ldots, n \), \( p = 1, 2, \ldots \) and by using the point \( x^p = (x_1^p, x_2^p, \ldots, x_n^p) \) the pivot points has been defined in [16-18], as

\[ x_{i, j}^{p} = (x_1^p, \ldots, x_{i-1}^p, x_{i+1}^p, \ldots, x_n^p) = (y^p, x_{i}^{p, j}) \]  

(3)

Such points have the same \( n - 1 \) components with the point \( x^p = (x_1^p, \ldots, x_{n-1}^p, x_n^p) \) and differ only at the \( n \)-th component. These points have imposed lying on the solution surfaces of the corresponding functions \( f_i(x) \), that is

\[ f_i(x_{i, j}^{p}) = 0 \]

(4)
Hence, the unknown \( n \)-th component \( x_s^p \) of pivot points can be found by solving each of the corresponding one-dimensional equations
\[
f_i(\mathbf{x}^p) = 0
\]
Implicit Function Theorem [2], guarantees the existence the unique mappings \( \varphi_i \) such that
\[
x_s = \varphi_i(y) \hspace{1em} f_i(y; \varphi_i(y)) = 0 \quad \text{and} \quad x_s^p = \varphi_i(y^p)
\]
Based on the definition of pivot points, the method namely WFEN (Without direct Function Evaluations Newton), is given by
\[
x^{p+1} = x^p - f'(x^p)^{-1}W(x^p), \quad p \geq 0.
\]
Where \( w_i(x) = \frac{\partial f_i}{\partial x_i}(x^p) \mathbf{q}^p \hspace{1em} (\text{for more details refer to} \ [16]).
\]
The other iterative scheme namely IWEN (Improved Without direct Function Evaluations Newton), as a modification of method (5) has presented, is given by
\[
x^{p+1} = x^p - L'(x^p)^{-1}L(x^p), \quad p \geq 0
\]
Where
\[
L(x^p) = \begin{bmatrix}
x^p - \varphi_i(y)
\vdots
x^p - \varphi_i(y^p)
\end{bmatrix},
\]
\[
L'(x^p) = \begin{bmatrix}
\frac{\partial f_i(x^p)}{\partial x_i} & \frac{\partial f_i(x^p)}{\partial x_i} & \cdots & \frac{\partial f_i(x^p)}{\partial x_i} \\
\frac{\partial f_i(x^p)}{\partial x_i} & \frac{\partial f_i(x^p)}{\partial x_i} & \cdots & \frac{\partial f_i(x^p)}{\partial x_i} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_i(x^p)}{\partial x_i} & \frac{\partial f_i(x^p)}{\partial x_i} & \cdots & \frac{\partial f_i(x^p)}{\partial x_i}
\end{bmatrix}
\]
also \( \varphi_i(y^p) \)’s are the mappings were mentioned above. (for more details refer to [17]).

III. DERIVATION OF THE NEW METHOD

A. Some new definitions

To develop a new method, let us have the following definitions.

Definition 1. For any \( i, j \in \{1, 2, \ldots, n\} \) and \( p = 1, \ldots, \) we define functions
\[
g_{i,j}^p(t) = f_i(x_{i,j}^p(t)) \quad R \rightarrow R
\]

Where \( x_{i,j}^p(t) = (x_{1}^p, x_{2}^p, \ldots, x_{i}^p, x_{j}^p, \ldots, x_{n}^p) \) and \( j \in \{1, 2, \ldots, n\} \) introduce as followed in next subsection.

Definition 2. For any \( i, j \in \{1, 2, \ldots, n\} \) and \( p = 1, \ldots, \) we extend the notion of pivot points (3) as the following form
\[
x_{i,j}^p(t) = (x_{1}^p, x_{1}^p, \ldots, x_{i}^p, x_{j}^p, \ldots, x_{n}^p, x_{n}^p)
\]

Form \( n-1 \) components of current point \( x_s^p \). The \( j \)-th unknown component \( x_{i,j}^p \) of modified pivot points can be found by solving each of the corresponding one dimensional equation
\[
g_{i,j}^p(t) = 0.
\]
According to Implicit Function Theorem there exist unique mappings \( \varphi_i \) such that
\[
x_{i,j}^p = \varphi_i(y) \hspace{1em} f_i(y; \varphi_i(y)) = 0 \quad \text{and therefore} \quad x_{i,j}^p = \varphi_i(y^p)
\]

In this paper, we use Newton’s method using the initial guess \( x_{i,j}^0 \) for solving each of the corresponding one-dimensional equations (8) (For simplicity, the Maple command NewtonsMethod can be used).

It is clear that the solution of (8) is depending on the expression of the components \( f_i \) and the current approach \( x^p \).

That is, if any of the Eq. (8) has no zeros, we are not able to apply our proposed method on a system of equations. Here, similar to what brought in [17], we can adopted some techniques to guarantee the existence of pivot points. For example choosing a different component for solving (8) as stated in [19], or applying either a reordering technique like in [20] or a linear combination between the components \( f_i \) like in [21]. For the needs of this work we consider that we are always able to find the zeros of (8) is possible.

B. Illustration of new method

The key idea in this paper is to define new quantities to approximate function values in Newton’s method (2).

At the first, we use the first order’s Taylor expansion of \( g_{i,j}^p(t) \) around the point \( t = x_{i,j}^p \) as
\[
g_{i,j}^p(t) \approx g_{i,j}^p(x_{i,j}^p)
\]

Due to the definition of pivot points and (7) and \( g_{i,j}^p(x_{i,j}^p) = 0 \), the above relation (10) becomes
\[
f_i(x^p) \approx \hat{e}_{i,j} f_i(x_{i,j}^p)(x_{i,j}^p - x_{i,j}^p)
\]

In Fig. 1, for any \( j \in \{1, 2, \ldots, n\} \) we can see the behavior of function \( g_{i,j}^p(x) \) around the point \( A = (x_{i,j}^p, 0) \). Also we have the If we bring from the pivot point \( B = (x_{i,j}^p, 0) \), the parallel line to the tangent of the function at the point \( P = (x_{i,j}^p, f_i(x_{i,j}^p)) \), this line cuts the segment AP at the point Q = (x_{i,j}^p, \hat{e}_{i,j} f_i(x_{i,j}^p)(x_{i,j}^p - x_{i,j}^p)).
Also it can be easily verified that the coordinate of point C is
\[ (x_{j(i)}^{p} - \frac{f_j(x^p)}{\partial_{j(i)} f_j(x^p)}, 0). \]
From the similar triangles, the function value \( f_j(x^p) \), denoted by the segment AP, can be approximated by the quantity \( \partial_{j(i)} f_j(x^p)(x_{j(i)}^{p'} - x_{j(i)}^{p}) \), denoted by the segment AQ, the suitable direction of \( f_j(x^p) \) instead of \( f_j(x^p) \) is more valid. So, we should choose that direction \( j(i) \) which minimizes the expression:
\[
[BC] = \left| x_{j(i)}^{p'} - x_{j(i)}^{p} + \frac{f_j(x^p)}{\partial_{j(i)} f_j(x^p)} \right|
\]
From triangular inequality, we have
\[
[BC] = \left| x_{j(i)}^{p'} - x_{j(i)}^{p} + \frac{f_j(x^p)}{\partial_{j(i)} f_j(x^p)} \right| \leq \left| x_{j(i)}^{p'} - x_{j(i)}^{p} \right| + \left| \frac{f_j(x^p)}{\partial_{j(i)} f_j(x^p)} \right|
\]
It is clear that, the expression \( \left| \frac{f_j(x^p)}{\partial_{j(i)} f_j(x^p)} \right| \) minimizing
whenever the numerator expression achieves its maximum value. Hence, in this paper we set \( j(i) = J \) when
\[
\partial_{j(i)} f_j(x^p) \geq \partial_{j(i)} f_j(x^p), \quad \text{for any } k \in \{1, \ldots, n\}, \quad \text{i.e. } \text{that direction which has to steepest slope of the gradient vector at the point } x^p.
\]
Now, using (11) in Newton method (2), we have
\[
\tilde{V}(x^p)\tilde{L}(x^p) + F(x^p)(x - x^p) = 0.
\]
Under the assumptions of Implicit Function Theorem the diagonal matrix \( \tilde{V}(x^p) \) is invertible and (12) becomes
\[
\tilde{V}(x^p)^{-1}F(x^p)(x - x^p) = -L(x^p).
\]
Now, we consider the function
\[
\tilde{L}(x) = \left( x_{j(i)} - \varphi_i(y), \ldots, x_{j(i)} - \varphi_i(y) \right)
\]
Utilizing again the Implicit Function Theorem to derive \( \partial_{j(i)} \varphi_i(x) \) we get
\[
(\tilde{L}(x^p))_{k,n} = \frac{\partial_{j(i)} f_j(x)}{\partial_{j(i)} f_j(x^p)}
\]
Eqs. (14) and (15) introduce iterative method given by
\[
x^p = x^p - \tilde{L}(x^p)^{-1}\tilde{L}(x^p), \quad p \geq 0.
\]
We will refer to this iteration iterative scheme (16) as BGM method, as the modified Newton method to solve systems of nonlinear equations.

**Theorem 1.** Suppose that \( F = (f_1, f_2, f_n) : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n \) is twice continuously differentiable on an open neighborhood \( D \subset D' \) of a point \( x^* = (x_1^*, x_2^*, \ldots, x_n^*) \in D \) for which \( F(x^*) = 0 \) and \( F'(x^*) \) is nonsingular. Then the iterations \( x^p, p = 1, 2, \ldots \) of the new method, given by (16) will converge to \( x^* \) provided the initial guess \( x^0 \) is sufficiently close to \( x^* \). Moreover the order of convergence will be two.

*Proof.* Refer to [17].

**IV. NUMERICAL EXAMPLES**

In this section, some examples are presented to illustrate the efficiency of proposed iterative family. In order to compare the results, we take the same examples were presented in [16-18].

In Tables 1-4 we present the results obtained, for various initial points, by Newton's method and the schemes (5), (6) and (16).

**Example 1.** The first system has two roots \( r_1 = (0,1,0,1,0,1) \) and \( r_2 = (-0.1,-0.1,-0.1) \). It is given by
\[
\begin{align*}
    f_1(x_1, x_2, x_3) &= x_1 - x_2 x_3 = 0 \\
    f_2(x_1, x_2, x_3) &= x_2 - x_1 x_3 = 0 \\
    f_3(x_1, x_2, x_3) &= 10 x_2 x_3 + x_2 - x_1 - 1 = 0
\end{align*}
\]

**Example 2.** The second example is
\[
\begin{align*}
    f_1(x_1, x_2, x_3) &= x_1 - x_2 x_3 = 0 \\
    f_2(x_1, x_2, x_3) &= x_2 - x_1 x_3 = 0 \\
    f_3(x_1, x_2, x_3) &= 10 x_2 x_3 + x_2 - x_1 - 1 = 0
\end{align*}
\]
with the solution
\[
    r = (-0.9999001 \times 10^{-4}, -0.9999001 \times 10^{-4}, 0.9999001 \times 10^{-4})
\]
Due to the definition of pivot points(3), It is clear that so
called “WFEN” and “IWFEN” methods are sensitive to the order of place of unknowns in component functions \( f_i(x) \).

This is a clear shortcoming of this methods in some systems. This sensitivity are shown in the followings examples.

**Example 3.** Rewrite Example 1. by changing the rule of \( x_1, x_2, \ldots \) as

\[
\begin{align*}
 f_1(x_1, x_2, x_3) &= x_3^3 - x_2 x_1 x_3 = 0 \\
 f_2(x_1, x_2, x_3) &= x_2^2 - x_1 x_2 = 0 \\
 f_3(x_1, x_2, x_3) &= 10 x_1 x_2 + x_3 - x_1 - 0.1 = 0
\end{align*}
\]

According to example 1., this system has the following stopping criteria

\[
\begin{align*}
 \text{FE} \text{ the number of the function evaluations}
\end{align*}
\]

\[
\begin{align*}
 (r_1, r_2) &= (0.1, 0.1, 0.1) \text{ and } r_2 = (-0.1, -0.1, -0.1) \text{ roots}.
\end{align*}
\]

In Tables 1 and 4, “IT” indicates the number of the iterations, “FE” the number of the function evaluations (including derivatives). Results were found using Maple software via 30 digit floating point arithmetic (Digits:=30) and following stopping criteria

\[
\| x^{k+1} - x^k \| + \| f(x^k) \| < 10^{-14}
\]

\section*{V. Conclusion}

In this paper, a modification of some existing method for solving system of nonlinear equations are presented. This method is independent of function evaluation and can be used in some systems that function calculations are quite costly or can’t be done precisely. As seen in tables [1-4], the numerical results of proposed method are quite satisfactory and admit the geometrical explanations. In some cases the results of our are very acceptable and there is a sufficient reduction on the number of iterations and hence the proposed method look be a reliable refinement for Newton’s method. It can be viewed as an improvement and refinement of the Newton’s methods and some resent methods.

\section*{References}


### Table I
**Comparison of different methods for Example 1**

<table>
<thead>
<tr>
<th>( x_1^0 )</th>
<th>( x_2^0 )</th>
<th>( x_3^0 )</th>
<th>Newton</th>
<th>WFEN</th>
<th>IMFEN</th>
<th>BGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1^0 )</td>
<td>( x_2^0 )</td>
<td>( x_3^0 )</td>
<td>IT</td>
<td>FE</td>
<td>IT</td>
<td>FE</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
<td>53</td>
<td>636</td>
<td>20</td>
<td>240</td>
</tr>
<tr>
<td>-4</td>
<td>-2</td>
<td>1</td>
<td>33</td>
<td>396</td>
<td>33</td>
<td>396</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>0.6</td>
<td>51</td>
<td>612</td>
<td>51</td>
<td>612</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>1</td>
<td>29</td>
<td>384</td>
<td>29</td>
<td>384</td>
</tr>
<tr>
<td>0.5</td>
<td>2</td>
<td>1</td>
<td>54</td>
<td>648</td>
<td>54</td>
<td>648</td>
</tr>
<tr>
<td>5</td>
<td>-2</td>
<td>-2</td>
<td>38</td>
<td>456</td>
<td>38</td>
<td>456</td>
</tr>
<tr>
<td>10</td>
<td>-2</td>
<td>-2</td>
<td>39</td>
<td>468</td>
<td>39</td>
<td>468</td>
</tr>
</tbody>
</table>

### Table II
**Comparison of different methods for Example 2**

<table>
<thead>
<tr>
<th>( x_1^0 )</th>
<th>( x_2^0 )</th>
<th>( x_3^0 )</th>
<th>Newton</th>
<th>WFEN</th>
<th>IMFEN</th>
<th>BGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1^0 )</td>
<td>( x_2^0 )</td>
<td>( x_3^0 )</td>
<td>IT</td>
<td>FE</td>
<td>IT</td>
<td>FE</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>42</td>
<td>504</td>
<td>38</td>
<td>456</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>27</td>
<td>324</td>
<td>27</td>
<td>324</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>5</td>
<td>92</td>
<td>1104</td>
<td>43</td>
<td>516</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>73</td>
<td>876</td>
<td>26</td>
<td>312</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>46</td>
<td>552</td>
<td>32</td>
<td>384</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>37</td>
<td>444</td>
<td>37</td>
<td>444</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>28</td>
<td>336</td>
<td>26</td>
<td>312</td>
</tr>
</tbody>
</table>

### Table III
**Comparison of different methods for Example 3**

<table>
<thead>
<tr>
<th>( x_1^0 )</th>
<th>( x_2^0 )</th>
<th>( x_3^0 )</th>
<th>Newton</th>
<th>WFEN</th>
<th>IMFEN</th>
<th>BGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1^0 )</td>
<td>( x_2^0 )</td>
<td>( x_3^0 )</td>
<td>IT</td>
<td>FE</td>
<td>IT</td>
<td>FE</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
<td>53</td>
<td>636</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>-4</td>
<td>1.5</td>
<td>5</td>
<td>38</td>
<td>456</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>-1</td>
<td>6</td>
<td>-2</td>
<td>15</td>
<td>180</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
<td>-2</td>
<td>11</td>
<td>132</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>54</td>
<td>648</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>180</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>-3</td>
<td>10</td>
<td>39</td>
<td>468</td>
<td>Div</td>
</tr>
</tbody>
</table>

Div = Divergent

### Table IV
**Comparison of different methods for Example 3**

<table>
<thead>
<tr>
<th>( x_1^0 )</th>
<th>( x_2^0 )</th>
<th>( x_3^0 )</th>
<th>Newton</th>
<th>WFEN</th>
<th>IMFEN</th>
<th>BGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1^0 )</td>
<td>( x_2^0 )</td>
<td>( x_3^0 )</td>
<td>IT</td>
<td>FE</td>
<td>IT</td>
<td>FE</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>42</td>
<td>504</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>27</td>
<td>324</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>92</td>
<td>1104</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>73</td>
<td>876</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>46</td>
<td>552</td>
<td>Div</td>
<td>Div</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>37</td>
<td>444</td>
<td>Div</td>
<td>Div</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-4</td>
<td>28</td>
<td>336</td>
<td>Div</td>
<td>Div</td>
</tr>
</tbody>
</table>

Div = Divergent