A NINTH-ORDER NEWTON-TYPE METHOD TO SOLVE SYSTEMS OF NONLINEAR EQUATIONS

Xiaowu Li1,*, Zhinan Wu2, Lin Wang3 & Qian Zhang3
1School of Information Engineering, Guizhou Minzu University, Guiyang 550025, PR China
2School of Mathematics and Computer Science, Yichun University, Yichun 336000, PR China
3College of Science, Guizhou Minzu University, Guiyang 550025, PR China
*Corresponding Author E-mail address: lixiaowu002@126.com.

ABSTRACT
In this paper, modification of Newton’s method with ninth-order convergence is presented. The modification of Newton’s method is based on Darvishi and Barati’s third-order method. The new method requires three-step per iteration. Analysis of convergence demonstrates that the order of convergence is 9. Some numerical examples illustrate that the algorithm is more efficient and performs better than classical Newton’s method and other methods.

Keywords: System of non-linear equations, Iterative methods, Newton-like’s method, Order of convergence.

1. INTRODUCTION
Consider the 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)

where each function \( f_i \) maps a vector \( x = (x_1, x_2, \ldots, x_n) \) of the \( n \)-dimensional space \( \mathbb{R}^n \) into the real line \( \mathbb{R} \). The system (1.1) of \( n \) nonlinear equations in \( n \) unknowns can also be represented by defining a function \( F \) mapping \( \mathbb{R}^n \) into \( \mathbb{R}^n \) as

\[ F(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T. \]

(2)

Thus, the system (1.1) can be written in the form \( F(x) = 0 \), where the functions \( f_1(x), f_2(x), \ldots, f_n(x) \) are the coordinate functions of \( F \). In recent years, several iterative methods have been developed to solve the nonlinear system of equations \( F(x) = 0 \) including essentially Taylor’s polynomial, decomposition, homotopy perturbation method, quadrature formulas methods. For more details, see[1-26] and the references therein. M.T. Darvishi and A. Barati [5] developed third-order method, which is written as:

\[ x_{k+1} = x_k - F'(x_k)^{-1} (F(x_k) + F(x_{k+1}^*)), \]

(3)

where \( x_{k+1}^* = x_k - F'(x_k)^{-1} F(x_k) \). Motivated and inspired by the on-going activities in this direction, we construct a modification (based on the above M.T. Darvishi and A. Barati’s method) of Newton’s method with higher-order convergence for solving the nonlinear system of equations. It has been shown that this three-step iterative method is ninth-order convergence. Several numerical examples are given to illustrate the efficiency and the performance of the new iterative methods. Our results can be viewed as an improvement and refinement of the previously known results.

2. THE METHOD AND ANALYSIS OF CONVERGENCE
From (3), we construct a three-step iterative method
where $x_k = (x_{1k}, x_{2k}, ..., x_{nk})'$, $y_k = (y_{1k}, y_{2k}, ..., y_{nk})'$, $z_k = (z_{1k}, z_{2k}, ..., z_{nk})'$ and $F(x_k)$ is Jacobian matrix. We let $F(x)^{-1} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}, \frac{\partial f_1}{\partial x_2}, ..., \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1}, \frac{\partial f_2}{\partial x_2}, ..., \frac{\partial f_2}{\partial x_n} \\ \vdots \\ \frac{\partial f_n}{\partial x_1}, \frac{\partial f_n}{\partial x_2}, ..., \frac{\partial f_n}{\partial x_n} \end{bmatrix}$. So the (4) can be expressed by

$$\begin{align*}
y_k &= x_k - F(x_k) F'(x_k), \\
z_k &= x_k - G(z_j) F(x_k), \\
F(x_k) &= F'(x_k) F'(x_k) - F'(x_k) F'(x_k), \\
x_{k+1} &= z_k - G(z_k) F(x_k), \\
F(x_k) &= F'(x_k) F'(x_k) - F'(x_k) F'(x_k).
\end{align*}$$

We can easily prove that scheme (5) is ninth-order convergent. In order to avoid the computation of the first derivative and inverse function $F'(z_k)^{-1}$, we use $G(z_k)$ instead of $F'(z_k)^{-1}$. Therefore, a new scheme is as follows:

$$\begin{align*}
y_k &= x_k - G(x_k) F(x_k), \\
z_k &= x_k - G(x_k) F(x_k) - G(x_k) F(y_k), \\
F(x_k) &= F'(x_k) F'(x_k) - G(z_k) F(z_k), \\
x_{k+1} &= z_k - G(z_k) F(x_k) - G(z_k) F(z_k).
\end{align*}$$

This proves that the method defined by (6) has ninth-order convergence.

**Theorem 1.** The iterative method (6) has local order of convergence at least nine with the following error equation

$$z_k = G(z_k) F(z_k) - G(z_k) F(z_k) - z_k = \alpha + 16c_8 e_8^8 + o\left(\|e_k\|^9\right).$$

**Proof:** Let $\alpha$ be a simple zero of $F$. As $F$ is a sufficiently differentiable function, by expanding $F(x_k)$ and $F'(x_k)$ about $\alpha$, we get

$$F(x_k) = F'(\alpha)(e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7 + c_8 e_k^8) + o\left(\|e_k\|^9\right),$$

and

$$F'(x_k) = F'(\alpha)(1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4 + 6c_6 e_k^5 + 7c_7 e_k^6 + 8c_8 e_k^7) + o\left(\|e_k\|^8\right).$$

where $c_k = (1/k)! F'(\alpha)^{-1} F^{(k)}(\alpha)$. As the terms in the square brackets are polynomials in terms of $e_k$, direct division gives us

$$F'(x_k)^{-1} F(x_k) = G(x_k) F(x_k)$$

$$= e_k - c_2 e_k^2 + 2c_3 e_k^3 + (7c_3 c_4 - 3c_4 - 4c_3^2) e_k^4 + (10c_3 c_4 - 4c_5 + 6c_2^2 - 20c_3 c_4^2 + 8c_2 c_2^2) e_k^5 + (17c_3 c_4 - 33c_3 c_4^2 + 52c_2 c_4^3 - 28c_4 c_4^2 + 13c_2 c_5 - 5c_6 - 16c_2 c_4) e_k^6 + o\left(\|e_k\|^8\right).$$
From (10), we have
\[ y_k = \alpha + c_2 e_k^2 - 2(c_2^2 - c_3^2) e_k^4 - (7c_2 c_3 - 3c_4 - 4c_2^2) e_k^6 - (10c_2 c_4 - 4c_3 + 6c_2^2 - 20c_3 c_2^2 + 8c_2^4) e_k^8 - (17c_3 c_4 - 33c_2 c_3^2 + 52c_3 c_2^2 - 28c_4 c_2^2 + 13c_2 c_3 - 5c_4 - 16c_2^4) e_k^{10} + o(\|e_k^{10}\|). \] (11)

By (9) and (11), we have
\[ G(x_k) F(y_k) = c_2 e_k^2 + (2c_2 - 4c_2^2) e_k^4 + (3c_4 - 14c_2 c_3 + 13c_2^2) e_k^6 + (-2c_3 + 64c_3 c_2^2) e_k^8 - 20c_2 c_4 - 38c_2^2 e_k^6 + (-34c_3 c_4 + 90c_2 c_3^2 - 240c_2^2 c_3^2 + 104c_2^3) e_k^8 + 103c_2 c_4 e_k^6 + (288c_2 c_4 - 24c_2^2 - 36c_4 c_3^2 - 558c_2^2 c_3^2 + 800c_3 c_2^4) e_k^{10} + 54c_3^3 - 272c_2^6 e_k^7 + o(\|e_k^{10}\|). \] (12)

Combining (10) with (12), we get
\[ z_k = \alpha + 2c_2 e_k^2 + (7c_2 c_3 - 9c_3^2) e_k^4 + (6c_2 - 44c_2 c_3^2 + 10c_2 c_4 + 30c_4^2) e_k^5 + (17c_3 c_4 - 62c_4 c_2^2 + 188c_3 c_2^3 - 88c_2^5 - 70c_2 c_3^2) e_k^6 + o(\|e_k^6\|). \] (13)

With (13), we obtain
\[ F(z_k) = F'(\alpha)(2c_2 e_k^2 + (7c_2 c_3 - 9c_3^2) e_k^4 + (6c_2^2 - 44c_2 c_3^2 + 10c_2 c_4 + 30c_4^2) e_k^5 - (84c_2^5 - 17c_3 c_4 + 62c_4 c_2^2 - 188c_3 c_2^3 + 70c_2 c_3^2) e_k^6) + o(\|e_k^7\|), \] (14)

\[ G(z_k) F(z_k) = 2c_2^2 e_k^2 + (7c_2 c_3 - 9c_3^2) e_k^4 + (6c_2^2 - 44c_2 c_3^2 + 10c_2 c_4 + 30c_4^2) e_k^5 + (17c_3 c_4 - 92c_4 c_2^2 + 188c_3 c_2^3 - 70c_2 c_3^2) e_k^6 + o(\|e_k^7\|), \] (15)

and
\[ z_k - G(z_k) F(z_k) = \alpha + 4c_2^2 e_k^6 + (28c_3 c_2^4 - 36c_3^2) e_k^7 + o(\|e_k^7\|). \] (16)

With (16), we get
\[ F(z_k - G(z_k) F(z_k)) = 4c_2^5 e_k^6 + (28c_3 c_2^4 - 36c_3^2) e_k^7 + o(\|e_k^8\|). \] (17)

From (13),(15),(17), we have
\[ z_k - G(z_k) F(z_k) - G(z_k) F(z_k) = \alpha + 16c_2^8 e_k^6 + o(\|e_k^{10}\|). \] (18)

This shows the ninth-order convergence of the method. Hence, the proof is completed.

3. NUMERICAL RESULTS

We now present some examples to illustrate the efficiency and the comparison of the newly developed method, see Table 1. We compare Newton’s method (NM), the Darvishi and Barati method [5], the methods NAD1 and NAD2 [26] and our Newton-type method. All computations were done using Visual C++ 6.0. We used \( \epsilon = 10^{-12} \). The following stopping criteria is used for computer programs[26]:

(i) \( \|x_{k+1} - x_k\|_\infty + \|F(x_k)\|_\infty < \epsilon \)
(ii) \( F(x_1, x_2) = (\exp(x_1) - \exp(\sqrt{2} x_1), x_1 - x_2), \bar{x}_1 = (\sqrt{\frac{\pi}{2}}, \sqrt{\frac{\pi}{2}})^T, \bar{x}_2 = (0, 0, 0, 0)^T \).
(iii) \( F(x_1, x_2) = (x_1 + \exp(x_2) - \cos(x_2), 3x_1 - x_2 - \sin(x_2)), \bar{x} = (0, 0, 0, 0)^T \).
(iv) \( F(x_1, x_2) = (x_1^2 - 2x_1 - x_2 + 0.5, x_2^2 + 4x_2^2 - 4), \bar{x} = (1.9007, 0.3112)^T \).
(v) \( F(x_1, x_2) = (x_1^2 + x_2^2 - 1, x_1 - x_2 + 0.5), \bar{x} = (\frac{1}{2}, \frac{1}{2})^T, \bar{x}_2 = (-\frac{1}{2} - \frac{\sqrt{3}}{2})^T \).
(5) \( F(x_1, x_2) = (\sin(x_1) + x_2 \cos(x_1), x_1 - x_2), \bar{x} = (0,0,0,0)^T \).

(6) \( F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \),

where \( x = (x_1, x_2, x_3, \ldots, x_n)^T \) and \( f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \ldots, n, \) such that

\[
\begin{align*}
&f_i(x) = x_i x_{i+1} - 1, i = 1, 2, \ldots, n - 1 \\
&f_n(x) = x_n x_1 - 1.
\end{align*}
\]

When \( n \) is odd, the exact zeros of \( \bar{F}(x) \) are \( \bar{x}_1 = (1, 1, \ldots, 1)^T \) and \( \bar{x}_2 = (-1, -1, \ldots, -1)^T \). Results appearing in Table 1 are obtained for \( n = 9 \).

(7) \( F(x) = (f_1(x), f_2(x), f_3(x), f_4(x)) \),

where \( x = (x_1, x_2, x_3, x_4)^T \) and \( f_i : \mathbb{R}^4 \to \mathbb{R}, i = 1, 2, 3, 4 \), such that

\[
\begin{align*}
&f_1(x) = x_2 x_3 + x_4 (x_2 + x_3) \\
&f_2(x) = x_1 x_3 + x_4 (x_1 + x_3) \\
&f_3(x) = x_1 x_2 + x_4 (x_1 + x_2) \quad \text{The zeros of } F(x) \text{ are } \bar{x}_1 = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{2\sqrt{3}}\right)^T \text{ and} \\
&f_4(x) = x_1 x_2 + x_3 x_1 + x_2 x_3 - 1.
\end{align*}
\]

\[
\bar{x}_2 = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{2\sqrt{3}}\right)^T.
\]

Table 1: Numerical results for nonlinear systems.

<table>
<thead>
<tr>
<th>( F(x) )</th>
<th>( x^{(0)} )</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NM</td>
<td>Darvishi and Barati</td>
</tr>
<tr>
<td>(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.3,2.3)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.8,1.8)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.8,0.8)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.5,2.0)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.3,0.5)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3.0,2.0)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.6,0.0)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.7,1.2)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(−1.0,−2.0)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.2,−1.5)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(−0.6,0.6)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.0,−2.0)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(−4.0,−4)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(−1,−1,−1,−1)^T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.020,2.00,0.0)^T</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 is obtained for \( n = 9 \).

Table 1: Numerical results for nonlinear systems.
4. **CONCLUSION**

We have obtained a Newton-type iterative method for nonlinear systems and proved that its convergence order is nine. This method is efficient and effective for solving systems of nonlinear equations, with moderate computation and good accuracy.

5. **ACKNOWLEDGEMENTS**

This work was supported by Scientific and Technology Foundation Funded Project of Guizhou Province([2012]2193), Introduced Talents Scientific Research Foundation Funded Project of Guizhou Minzu University, Key Laboratory of Pattern Recognition and Intelligent System of Construction Project of Guizhou Province([2009]4002), Information Processing and Pattern Recognition for Graduate Education Innovation Base of Guizhou Province and the National Science Foundation of China (61263034).

6. **REFERENCES**


