

Article

An Efficient Sixth-Order Newton-Type Method for Solving Nonlinear Systems

Xiaofeng Wang ^{1,*} and Yang Li ²¹ School of Mathematics and Physics, Bohai University, Jinzhou 121013, China² Department of Mathematics and Information Engineering, Puyang Vocational and Technical College, Puyang 457000, China; wanghuanqing@bhu.edu.cn

* Correspondence: w200888w@163.com; Tel.: +86-150-4169-9258

Academic Editors: Alicia Cordero, Juan R. Torregrosa and Francisco I. Chicharro

Received: 26 January 2017; Accepted: 20 April 2017; Published: 25 April 2017

Abstract: In this paper, we present a new sixth-order iterative method for solving nonlinear systems and prove a local convergence result. The new method requires solving five linear systems per iteration. An important feature of the new method is that the LU (lower upper, also called LU factorization) decomposition of the Jacobian matrix is computed only once in each iteration. The computational efficiency index of the new method is compared to that of some known methods. Numerical results are given to show that the convergence behavior of the new method is similar to the existing methods. The new method can be applied to small- and medium-sized nonlinear systems.

Keywords: nonlinear systems; iterative method; Newton's method; computational efficiency

MSC: 65H10

1. Introduction

We consider the problem of finding a zero of a nonlinear function $F : D \subset R^m \rightarrow R^m$, that is, a solution α of the nonlinear system $F(x) = 0$ with m equations and m unknowns. Newton's method [1,2] is the well-known method for solving nonlinear systems, which can be written as:

$$x^{(k+1)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \quad (1)$$

where $F'(x^{(k)})$ is the Jacobian matrix of the function F evaluated in the k th iteration and $F'(x^{(k)})^{-1}$ is the inverse of $F'(x^{(k)})$. Newton's method is denoted by NM. Newton's method converges quadratically if $F'(\alpha)$ is nonsingular and $F'(x)$ is Lipschitz continuous on D . The method (1) can be written as:

$$\begin{cases} F'(x^{(k)})\gamma^{(k)} = F(x^{(k)}), \\ x^{(k+1)} = x^{(k)} - \gamma^{(k)}, \end{cases} \quad (2)$$

which requires $(m^3 - m)/3$ multiplications and divisions in the LU decomposition (lower upper, also called LU factorization) and m^2 multiplications and divisions for solving two triangular linear systems. So, the computational cost (multiplications and divisions) of the method (1) is $(m^3 - m)/3 + m^2$.

In order to accelerate the convergence or to reduce the computational cost and function evaluation in each step of the iterative process, many efficient methods have been proposed for solving nonlinear

systems, see [3–21] and the references therein. Cordero et al. [3,4] developed some variants of Newton’s method. One of the methods is the following fourth-order method [3]:

$$\begin{cases} y^{(k)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ x^{(k+1)} = y^{(k)} - [2I - F'(x^{(k)})^{-1}F'(y^{(k)})]F'(x^{(k)})^{-1}F(y^{(k)}), \end{cases} \quad (3)$$

where I is the identity matrix. Method (3) is denoted by CM4 and requires LU decomposition of the Jacobian matrix only once per full iteration. Based on method (3), Cordero et al. [4] presented the following sixth-order method:

$$\begin{cases} y^{(k)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} = y^{(k)} - [2I - F'(x^{(k)})^{-1}F'(y^{(k)})]F'(x^{(k)})^{-1}F(y^{(k)}), \\ x^{(k+1)} = z^{(k)} - F'(y^{(k)})^{-1}F(z^{(k)}), \end{cases} \quad (4)$$

where I is the identity matrix. Method (4) is denoted by CHM and requires two LU decompositions of the Jacobian matrix, one for $F'(x^{(k)})$ and one for $F'(y^{(k)})$. Grau-Sánchez et al. [5] obtained the following sixth-order method:

$$\begin{cases} y^{(k)} = x^{(k)} - F[x^{(k)} + F(x^{(k)}), x - F(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} = y^{(k)} - \left\{ 2F[x^{(k)}, y^{(k)}] - F[x^{(k)} + F(x^{(k)}), x - F(x^{(k)})] \right\}^{-1}F(y^{(k)}), \\ x^{(k+1)} = z^{(k)} - \left\{ 2F[x^{(k)}, y^{(k)}] - F[x^{(k)} + F(x^{(k)}), x - F(x^{(k)})] \right\}^{-1}F(z^{(k)}), \end{cases} \quad (5)$$

where $F[\cdot, \cdot]$ denotes the first-order divided difference of D . Method (5) is denoted by SNAM and requires two LU decompositions for solving the linear systems involved.

It is well known that the computational cost of the iterative method greatly influences the efficiency of the iterative method. The number of LU decompositions that are used in the iterative method thus plays an important role when it comes to measuring the computational cost. So, we can reduce the computational cost of the iterative method by reducing the number of LU decompositions in each iteration.

The purpose of this paper is to construct a new sixth-order iterative method for solving small- and medium-sized systems. The theoretical advantages of the new method are based on the assumption that the Jacobian matrix is dense and that LU factorization is used to solve systems with the Jacobian. This assumption is not correct for sparse Jacobian matrices. An important feature of the new method is that the LU decomposition is computed only once per full iteration. This paper is organized as follows. In Section 2, based on the well-known fourth-order method (3), we present a sixth-order iterative method for solving nonlinear systems. The new method only increases one function evaluation, F . For a system of m equations, each iteration uses $3m + 2m^2$ evaluations of scalar functions. The computational efficiency is compared to some well-known methods in Section 3. In Section 4, numerical examples are given to illustrate the convergence behavior of our method. Section 5 offers a short conclusion.

2. The New Method and Analysis of Convergence

Based on the method (3), we construct the following iterative scheme:

$$\begin{cases} y^{(k)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} = y^{(k)} - (2I - F'(x^{(k)})^{-1}F'(y^{(k)}))F'(x^{(k)})^{-1}F(y^{(k)}), \\ x^{(k+1)} = z^{(k)} - (2I - F'(x^{(k)})^{-1}F'(y^{(k)}))F'(x^{(k)})^{-1}F(z^{(k)}), \end{cases} \quad (6)$$

where I is the identity matrix. We note that the first two steps of method (6) are the same as those of method (3), and the third step of method (6) increases one function evaluation $F(z^{(k)})$. Method (6) will be denoted by M6. For method (6), we have the following convergence analysis.

Theorem 1. *Let $\alpha \in R^m$ be a solution of the system $F(x) = 0$ and $F : D \subset R^m \rightarrow R^m$ be sufficiently differentiable in an open neighborhood D of α . Suppose that $F'(x)$ is nonsingular in D . Then, for an initial approximation sufficiently close to α , the iterations converge with order 6.*

Proof. By using the notation introduced in [6], we have the following Taylor’s expansion of $F(x^{(k)})$ around α :

$$F(x^{(k)}) = F'(\alpha)[e + A_2e^2 + A_3e^3 + O(e^4)], \tag{7}$$

where $A_i = \frac{1}{i!}F'(\alpha)^{-1}F^{(i)}(\alpha) \in L_i(R^m, R^m)$, $e = x^{(k)} - \alpha$, $A_i e^i \in R^m$, $F^{(i)}(\alpha) \in L(R^m \times \dots \times R^m, R^m)$, $F^{-1}(\alpha) \in L(R^m)$ and e^i denotes (e, \dots, e) . From (7) the derivatives of $F(x^{(k)})$ can be written as:

$$F'(x^{(k)}) = F'(\alpha)[I + 2A_2e + 3A_3e^2 + O(e^3)] = F'(\alpha)D(e) + O(e^3), \tag{8}$$

where $D(e) = I + 2A_2e + 3A_3e^2$. The inverse of (8) can be written as:

$$F'(x^{(k)})^{-1} = D(e)^{-1}F'(\alpha)^{-1} + O(e^3). \tag{9}$$

Then, we compel the inverse of $D(e)$ to be (see [6]):

$$D(e)^{-1} = I + X_2e + X_3e^2 + O(e^3), \tag{10}$$

such that X_2 verifies:

$$D(e)D(e)^{-1} = D(e)^{-1}D(e) = I. \tag{11}$$

Solving system (11), we get:

$$X_2 = -2A_2, \tag{12}$$

$$X_3 = 4A_2^2 - 3A_3, \tag{13}$$

then,

$$F'(x^{(k)})^{-1} = [I - 2A_2e + (4A_2^2 - 3A_3)e^2]F'(\alpha)^{-1} + O(e^3). \tag{14}$$

Let us denote $E = y^{(k)} - \alpha$. From (6), (7) and (14), we arrive at:

$$E = e - F'(x^{(k)})^{-1}F(x^{(k)}) = A_2e^2 + O(e^3). \tag{15}$$

By a similar argument to that of (7), we obtain:

$$F(y^{(k)}) = F'(\alpha)[E + A_2E^2 + O(E^3)] = F'(\alpha)[A_2e^2 + A_2^3e^4 + O(e^5)], \tag{16}$$

$$F'(y^{(k)}) = F'(\alpha)[I + 2A_2E + O(E^2)] = F'(\alpha)[I + 2A_2^2e^2 + O(e^3)]. \tag{17}$$

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

$$(2I - F'(x^{(k)})^{-1}F'(y^{(k)}))F'(x^{(k)})^{-1} = [I - 2A_2E - 4A_2^2e^2]F'(\alpha)^{-1} + O(e^3) = [I - 6A_2^2e^2]F'(\alpha)^{-1} + O(e^3). \tag{18}$$

Let us denote $\varepsilon = z^{(k)} - \alpha$. From (14), (15) and (18), we get:

$$\begin{aligned} \varepsilon = z^{(k)} - \alpha &= E - [I - 2A_2E - 4A_2^2e^2][E + A_2E^2 + O(E^3)] \\ &= A_2E^2 + 4A_2^2e^2E + O(e^5) \\ &= 5A_2^3e^4 + O(e^5), \end{aligned} \tag{19}$$

$$F(z^{(k)}) = F'(\alpha)[\varepsilon + A_2\varepsilon^2 + O(\varepsilon^3)] = F'(\alpha)[5A_2^3\varepsilon^4 + O(\varepsilon^5)]. \tag{20}$$

Therefore, from (15) and (18)–(20), we obtain the error equation:

$$\begin{aligned} e_{n+1} = x^{(k+1)} - \alpha &= \varepsilon - [I - 2A_2E - 4A_2^2\varepsilon^2][\varepsilon + O(\varepsilon^2)] \\ &= 2A_2E\varepsilon + 4A_2^2\varepsilon^2\varepsilon + O(\varepsilon^7) \\ &= 30A_2^5\varepsilon^6 + O(\varepsilon^7). \end{aligned} \tag{21}$$

This implies that method (6) is of sixth-order convergence. This completes the proof.

In order to simplify the calculation, the new method (6) can be written as:

$$\left\{ \begin{array}{l} F'(x^{(k)})\gamma^{(k)} = F(x^{(k)}), \ y^{(k)} = x^{(k)} - \gamma^{(k)}, \\ F'(x^{(k)})\delta_1^{(k)} = F(y^{(k)}), \ \delta_2^{(k)} = F'(y^{(k)})\delta_1^{(k)}, \ F'(x^{(k)})\delta_3^{(k)} = \delta_2^{(k)}, \\ z^{(k)} = y^{(k)} - 2\delta_1^{(k)} + \delta_3^{(k)}, \\ F'(x^{(k)})\beta_1^{(k)} = F(z^{(k)}), \ \beta_2^{(k)} = F'(y^{(k)})\beta_1^{(k)}, \ F'(x^{(k)})\beta_3^{(k)} = \beta_2^{(k)}, \\ x^{(k+1)} = z^{(k)} - 2\beta_1^{(k)} + \beta_3^{(k)}. \end{array} \right. \tag{22}$$

From (22), we can see that the LU decomposition of the Jacobian matrix $F'(x^{(k)})$ would be computed only once per iteration.

3. Computational Efficiency

Here, we want to compare the computational efficiency index of our sixth-order method (M6, (6)) with Newton’s method (NM, (1)), Cordero’s fourth-order method (CM4, (3)), Grau-Sánchez’s sixth-order method (SNAM, (5)) and Cordero’s sixth-order methods (CHM, (4)) and (CTVM, (18)). The method CTVM [7] is as follows:

$$\left\{ \begin{array}{l} y^{(k)} = x^{(k)} - 1/2F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} = x^{(k)} + [F'(x^{(k)}) - 2F'(y^{(k)})]^{-1}[3F(x^{(k)}) - 4F(y^{(k)})], \\ x^{(k+1)} = z^{(k)} + [F'(x^{(k)}) - 2F'(y^{(k)})]^{-1}F(z^{(k)}). \end{array} \right. \tag{23}$$

We define, respectively, the computational efficiency index (CEI) of the methods NM, CM4, SNAM, CHM, CTVM and M6 as:

$$CEI_i(\mu, m) = \rho_i \overline{C_i(\mu, m)}^{-1}, \ i = 1, 2, 3, 4, 5, 6, \tag{24}$$

where ρ_i is the convergence order of the method and $C_i(\mu, m)$ is the computational cost of method. The $C_i(\mu, m)$ is given by [8]:

$$C_i(\mu, m) = a_i(m)\mu + p_i(m), \tag{25}$$

where $a_i(m)$ represents the number of evaluations of the scalar functions used in the evaluations of F , F' and $[y, x; F]$. The $p_i(m)$ represents the computational cost per iteration. $\mu > 0$ is the ratio between multiplications (and divisions) and evaluations of functions that are required to express $C_i(\mu, m)$ in terms of multiplications (and divisions). The divided difference $[y, x; F]$ is defined by [9]:

$$[y, x; F]_{ij} = (F_i(y_1 \cdots, y_{j-1}, y_j, x_{j+1}, \cdots, x_m) - F_i(y_1 \cdots, y_{j-1}, x_j, x_{j+1}, \cdots, x_m)) / (y_j - x_j), \ 1 \leq i, j \leq m,$$

where $F(x)$ and $F(y)$ are computed separately. When we compute a divided difference, we need m^2 quotients and $m(m - 1)$ scalar function evaluations. We must add m multiplications for the scalar product, m^2 multiplications for the matrix-vector multiplication and m^2 evaluations of scalar functions for any new derivative F' . In order to factorize a matrix, we require $(m^3 - m)/3$ multiplications

and divisions in the LU decomposition. We require m^2 multiplications and divisions for solving two triangular linear systems.

Taking into account the previous considerations, we give the computational cost of each iterative method in Table 1.

Table 1. Computational cost of the iterative methods.

Methods	ρ	$a(m)$	$p(m)$	$C(\mu, m)$
NM	2	$m(m + 1)$	$(m^3 - m)/3 + m^2$	$C_1 = m(m + 1)\mu + (m^3 - m)/3 + m^2$
CM4	4	$2m(m + 1)$	$(m^3 - m)/3 + 4m^2 + m$	$C_2 = 2m(m + 1)\mu + (m^3 - m)/3 + 4m^2 + m$
SNAM	6	$m(2m + 3)$	$2(m^3 - m)/3 + 6m^2$	$C_3 = m(2m + 3)\mu + 2(m^3 - m)/3 + 6m^2$
CHM	6	$m(2m + 3)$	$2(m^3 - m)/3 + 5m^2 + m$	$C_4 = m(2m + 3)\mu + 2(m^3 - m)/3 + 5m^2 + m$
CTVM	6	$m(2m + 3)$	$2(m^3 - m)/3 + 4m^2 + 3m$	$C_5 = m(2m + 3)\mu + 2(m^3 - m)/3 + 4m^2 + 3m$
M6	6	$m(2m + 3)$	$(m^3 - m)/3 + 7m^2 + 2m$	$C_6 = m(2m + 3)\mu + (m^3 - m)/3 + 7m^2 + 2m$

From Table 1, we can see that these sixth-order iterative methods have the same number of function evaluations. Our method (M6) needs less LU decompositions than other methods with the same order. Therefore, the computational cost of our method is lower.

We use the following expressions [10] to compare the computational efficiency index of each iterative method:

$$R_{i,j} = \frac{\log CEI_i}{\log CEI_j} = \frac{\log(\rho_i)C_j(\mu, m)}{\log(\rho_j)C_i(\mu, m)}, i, j = 1, 2, 3, 4, 5, 6. \tag{26}$$

For $R_{i,j} > 1$ the iterative method M_i is more efficient than M_j . We have the following theorem:

Theorem 2. For all $\mu > 0$ we have:

1. $CEI_6 > CEI_3$ for all $m \geq 5$,
2. $CEI_6 > CEI_4$ for all $m \geq 7$,
3. $CEI_6 > CEI_5$ for all $m \geq 9$.

Proof. We note that the methods SNAM, CHM, CTVM and method M6 have the same order $\rho_3 = \rho_4 = \rho_5 = \rho_6 = 6$ and the same number of function evaluations $a_3(m) = a_4(m) = a_5(m) = a_6(m) = m(2m + 3)$.

1. Based on the expression (26), the relation between SNAM and M6 can be given by:

$$R_{6,3} = \frac{\log(\rho_6)C_3(\mu, m)}{\log(\rho_3)C_6(\mu, m)} = \frac{m(2m + 3)\mu + 2(m^3 - m)/3 + 6m^2}{m(2m + 3)\mu + (m^3 - m)/3 + 7m^2 + 2m}. \tag{27}$$

Subtracting the denominator from the numerator of (27), we have:

$$\frac{1}{3}m(m^2 - 3m - 7). \tag{28}$$

Equation (28) is positive for $m \geq 4.541$. Thus, we get $CEI_6 > CEI_3$ for all $m \geq 5$ and $\mu > 0$.

2. The relation between CHM and M6 is given by:

$$R_{6,4} = \frac{\log(\rho_6)C_4(\mu, m)}{\log(\rho_4)C_6(\mu, m)} = \frac{m(2m + 3)\mu + 2(m^3 - m)/3 + 5m^2 + m}{m(2m + 3)\mu + (m^3 - m)/3 + 7m^2 + 2m}. \tag{29}$$

Subtracting the denominator from the numerator of (29), we have:

$$\frac{1}{3}m(m^2 - 6m - 4). \tag{30}$$

Equation (30) is positive for $m \geq 6.606$. Thus, we get $CEI_6 > CEI_4$ for all $m \geq 7$ and $\mu > 0$.

3. The relation between CHM and M6 can be given by:

$$R_{6,4} = \frac{\log(\rho_6)C_5(\mu, m)}{\log(\rho_5)C_6(\mu, m)} = \frac{m(2m + 3)\mu + 2(m^3 - m)/3 + 4m^2 + 3m}{m(2m + 3)\mu + (m^3 - m)/3 + 7m^2 + 2m}. \tag{31}$$

Subtracting the denominator from the numerator of (31), we have:

$$\frac{1}{3}m(m^2 - 9m + 2). \tag{32}$$

Equation (32) is positive for $m \geq 8.772$. Thus, we obtain $CEI_6 > CEI_5$ for all $m \geq 9$ and $\mu > 0$. This completes the proof.

Theorem 3. For all $m \geq 2$ we have:

1. $CEI_6 > CEI_1$ for all $\mu > \frac{m^2 \log(3) + 3(\log(3) - 6 \log(2))m - (6 \log(2) + \log(3))}{3(m \log(2/3) + \log(4/3))}$,
2. $CEI_6 > CEI_2$ for all $\mu > \frac{m^2 \log(3/2) + 6(2 \log(3) - 5 \log(2))m + 2(\log(3) - 4 \log(2))}{6(m \log(2/3) + \log(4/3))}$.

Proof.

1. From expression (26) and Table 1, we get the following relation between NM and M6:

$$R_{6,1} = \frac{\log(\rho_6)C_1(\mu, m)}{\log(\rho_1)C_6(\mu, m)} = \frac{\log(6)}{\log(2)} \frac{m(m + 1)\mu + (m^3 - m)/3 + m^2}{m(2m + 3)\mu + (m^3 - m)/3 + 7m^2 + 2m}. \tag{33}$$

We consider the boundary $R_{6,1} = 1$. The boundary can be given by the following equation:

$$\mu = H_{6,1}(m) = \frac{m^2 \log(3) + 3(\log(3) - 6 \log(2))m - (6 \log(2) + \log(3))}{3(m \log(2/3) + \log(4/3))}, \tag{34}$$

where $CEI_6 > CEI_1$ over it (see Figure 1). Boundary (34) cuts axes at points $(\mu, m) = (0, 8.8948)$ and $(12.2470, 2)$. Thus, we get $CEI_6 > CEI_1$ since $R_{6,1} > 1$ for all $m \geq 2$ and $\mu > H_{6,1}(m)$.

2. The relation between CM4 and M6 is given by:

$$R_{6,2} = \frac{\log(\rho_6)C_2(\mu, m)}{\log(\rho_2)C_6(\mu, m)} = \frac{\log(6)}{\log(4)} \frac{2m(m + 1)\mu + (m^3 - m)/3 + 4m^2 + m}{m(2m + 3)\mu + (m^3 - m)/3 + 7m^2 + 2m}. \tag{35}$$

We consider the boundary $R_{6,2} = 1$. The boundary can be given by the following equation:

$$\mu = H_{6,2}(m) = \frac{m^2 \log(3/2) + 6(2 \log(3) - 5 \log(2))m + 2(\log(3) - 4 \log(2))}{6(m \log(2/3) + \log(4/3))}, \tag{36}$$

where $CEI_6 > CEI_2$ over it (see Figure 1). Boundary (36) cuts axes at points $(\mu, m) = (0, 19.2012)$ and $(5.3984, 2)$. Thus, we get $CEI_6 > CEI_2$ since $R_{6,2} > 1$ for all $m \geq 2$ and $\mu > H_{6,2}(m)$.

This completes the proof.

In Tables 2 and 3, we show the computational efficiency indices of NM, CM4, SNAM, CHM, CTVM and M6, for different values of the size of the nonlinear systems.

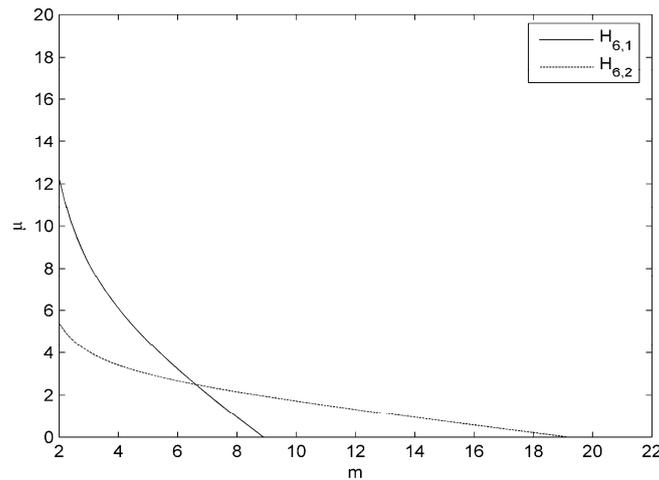


Figure 1. The boundary functions $H_{6,1}$ and $H_{6,2}$ in (μ, m) —plain.

Table 2. Computational efficiency indices of the methods ($\mu = 2$).

m	CEI_1	CEI_2	CEI_3	CEI_4	CEI_5	CEI_6
5	1.0055606	1.0052450	1.0049895	1.0052838	1.0055283	1.0050600
7	1.0025422	1.0025753	1.0023729	1.0025126	1.0026423	1.0025375
9	1.0013845	1.0014870	1.0013340	1.0014096	1.0014831	1.0014905
11	1.0008405	1.0009480	1.0008314	1.0008761	1.0009207	1.0009643
20	1.0001777	1.0002326	1.0001898	1.0001978	1.0002060	1.0002482
50	1.0000141	1.0000224	1.0000165	1.0000169	1.0000173	1.0000258
100	1.0000019	1.0000034	1.0000023	1.0000024	1.0000024	1.0000040
200	1.0000002	1.0000005	1.0000003	1.0000003	1.0000003	1.0000006

Table 3. Computational efficiency indices of the methods ($\mu = 6$).

m	CEI_1	CEI_2	CEI_3	CEI_4	CEI_5	CEI_6
5	1.0028332	1.0027489	1.0028941	1.0029907	1.0030675	1.0029177
7	1.0013956	1.0014055	1.0014554	1.0015068	1.0015525	1.0015157
9	1.0008054	1.0008390	1.0008536	1.0008839	1.0009123	1.0009150
11	1.0005124	1.0005505	1.0005504	1.0005697	1.0005882	1.0006057
20	1.0001242	1.0001488	1.0001391	1.0001434	1.0001476	1.0001681
50	1.0000117	1.0000168	1.0000139	1.0000141	1.0000144	1.0000199
100	1.0000017	1.0000028	1.0000021	1.0000021	1.0000022	1.0000034
200	1.0000002	1.0000004	1.0000003	1.0000003	1.0000003	1.0000005

The results shown in Tables 2 and 3 are in concordance with the Theorems 2 and 3. We can see that, for small- and medium-sized systems, the computational efficiency index of our method (M6) is similar to the other methods in this paper.

4. Numerical Examples

In this section, we compare the related methods by numerical experiments. The numerical experiments are performed using the MAPLE computer algebra system with 2048 digits. The method M6 is compared with NM, CM4, SNAM, CHM, CTVM by solving some nonlinear systems. The stopping criterion used is $\|x^{(k+1)} - x^{(k)}\| < 10^{-200}$ or $\|F(x^{(k)})\| < 10^{-200}$.

Following nonlinear systems are used: $F_1(x_1, x_2) = (2 - e^{x_1} + \arctan(x_2), \arctan(x_1^2 + x_2^2 - 5))$ and the initial value is $x^{(0)} = (1.35, 2)^T$. The solution is $\alpha \approx (1.1290650391602, 1.9300808629035)^T$.

$F_2(x_1, x_2, x_3) = (x_2 + x_3 - e^{-x_1}, x_1 + x_3 - e^{-x_2}, x_1 + x_2 - e^{-x_3})$ and the initial value is $x^{(0)} = (0.2, 1.5, 1.5)^T$. The solution is $\alpha \approx (0.351733711249, 0.351733711249, 0.351733711249)^T$.

$F_3(x) = (f_1(x), f_2(x), \dots, f_n(x))$, where $x = (x_1, x_2, \dots, x_n)^T$ and $f_i : R^n \rightarrow R, i = 1, 2, \dots, n$, such that

$$f_i(x) = x_i x_{i+1} - 1, i = 1, 2, \dots, n - 1,$$

$$f_n(x) = x_n x_1 - 1.$$

When n is odd, the exact zeros of $F_4(x)$ are $\alpha_1 = (1, 1, \dots, 1)$ and $\alpha_2 = (-1, -1, \dots, -1)$. The initial value is $x^{(0)} = (2.5, 0.5, 1.5, 2.5, 2.5, 1.5, 2.5, 0.5, 2.5, 1.5, 8.5)^T$.

Table 4 presents the results showing the following information: the number of iterations k needed to converge to the solution, the value of the stopping factors at the last step and the computational order of convergence ρ . The computational order of convergence ρ is defined by [18]:

$$\rho \approx \frac{\ln\left(\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)} - x^{(k-1)}\|}\right)}{\ln\left(\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|}\right)}. \tag{37}$$

Table 4. Numerical results for $F_i (i = 1, 2, 3)$ by the methods.

Function	Method	k	$\ x^{(k)} - x^{(k-1)}\ $	$\ F(x^{(k)})\ $	ρ
F_1	NM	8	2.42128×10^{-192}	1.06480×10^{-383}	1.99667
	CM4	5	5.59843×10^{-147}	2.69120×10^{-586}	4.00129
	SNAM	4	3.76810×10^{-39}	3.25655×10^{-227}	6.09363
	CHM	4	4.18959×10^{-123}	4.03125×10^{-736}	5.99962
	CTVM	4	2.07203×10^{-100}	2.63883×10^{-597}	6.00033
	M6	4	7.65662×10^{-119}	1.55028×10^{-710}	6.00589
F_2	NM	9	3.41596×10^{-116}	2.48971×10^{-232}	1.97549
	CM4	5	3.73825×10^{-90}	1.20501×10^{-359}	4.02761
	SNAM	4	9.18821×10^{-35}	6.76819×10^{-207}	5.98999
	CHM	4	8.31995×10^{-52}	8.11818×10^{-310}	5.72008
	CTVM	4	3.82928×10^{-42}	4.59455×10^{-251}	5.85429
	M6	4	8.13364×10^{-65}	6.14607×10^{-387}	5.99644
F_3	NM	22	2.71070×10^{-196}	2.20459×10^{-392}	1.99900
	CM4	6	2.26562×10^{-115}	1.03777×10^{-460}	4.00061
	SNAM	nc			
	CHM	5	2.79450×10^{-99}	4.68047×10^{-594}	5.92903
	CTVM	5	5.12075×10^{-193}	$1.30600 \times 10^{-1157}$	5.97091
	M6	5	1.99499×10^{-161}	3.41913×10^{-967}	6.08153

The numerical results shown in Table 4 are in concordance with the theory developed in this paper. The order of convergence of our method (M6) is 6, which is higher than the methods NM and CM4. The iterative method SNAM is not convergent (nc) for F_3 with the corresponding initial value. The convergence behavior of our method is similar to the existing methods in this paper.

5. Conclusions

In this paper, we have proposed a new iterative method of order six for solving nonlinear systems. Although five linear systems are required to be solved in each iteration, the LU decomposition of the linear systems are computed only once per full iteration. Numerical results are given to show that our method has a similar convergence behavior as the existing methods in this paper. The new method is suitable for solving small- and medium-sized systems. In order to obtain an efficient iterative method for solving nonlinear systems, we should make the iterative method achieve an as high as possible convergence order consuming an as small as possible computational cost. In addition, the theoretical advantages of the new method are based on the assumption that the Jacobian matrix is dense and that LU factorization is used to solve the systems with the Jacobian. This assumption is not correct for sparse Jacobian matrices.

Acknowledgments: This project is supported by the National Natural Science Foundation of China (Nos. 11547005 and 61572082), the PhD Start-up Fund of Liaoning Province of China (No. 201501196) and the Educational Commission Foundation of Liaoning Province of China (No. L2015012).

Author Contributions: Xiaofeng Wang conceived and designed the experiments; Yang Li performed the experiments; Xiaofeng Wang wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kelley, C.T. *Solving Nonlinear Equations with Newton's Method*; SIAM: Philadelphia, PA, USA, 2003.
2. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA, 1970.
3. Cordero, A.; Martínez, E.; Torregrosa, J.R. Iterative methods of order four and five for systems of nonlinear equations. *J. Comput. Appl. Math.* **2009**, *231*, 541–551. [[CrossRef](#)]
4. Cordero, A.; Hueso, J.L.; Martínez, E.; Torregrosa, J.R. Increasing the convergence order of an iterative method for nonlinear systems. *Appl. Math. Lett.* **2012**, *25*, 2369–2374. [[CrossRef](#)]
5. Sánchez, M.G.; Noguera, M.; Amat, S. On the approximation of derivatives using divided difference operators preserving the local convergence order of iterative methods. *J. Comput. Appl. Math.* **2013**, *237*, 263–272.
6. Cordero, A.; Hueso, J.L.; Martínez, E.; Torregrosa, J.R. Efficient high-order methods based on golden ratio for nonlinear systems. *Appl. Math. Comput.* **2011**, *217*, 4548–4556. [[CrossRef](#)]
7. Cordero, A.; Torregrosa, J.R.; Vassileva, M.P. Pseudocomposition: A technique to design predictor-corrector methods for systems of nonlinear equations. *Appl. Math. Comput.* **2012**, *218*, 11496–11504. [[CrossRef](#)]
8. Ezquerro, J.A.; Grau, À.; Sánchez, M.G.; Hernández, M.A. On the efficiency of two variants of Kurchatov's method for solving nonlinear systems. *Numer. Algorithms* **2013**, *64*, 685–698. [[CrossRef](#)]
9. Potra, F.A.; Pták, V. *Nondiscrete Induction and Iterative Processes*; Pitman Publishing: Boston, MA, USA, 1984.
10. Ezquerro, J.A.; Grau, À.; Sánchez, M.G.; Hernández, M.A.; Noguera, M. Analysing the efficiency of some modifications of the secant method. *Comput. Math. Appl.* **2012**, *64*, 2066–2073. [[CrossRef](#)]
11. Sánchez, M.G.; Grau, À.; Noguera, M. On the computational efficiency index and some iterative methods for solving systems of nonlinear equations. *J. Comput. Appl. Math.* **2011**, *236*, 1259–1266. [[CrossRef](#)]
12. Sánchez, M.G.; Grau, À.; Noguera, M. Frozen divided difference scheme for solving systems of nonlinear equations. *J. Comput. Appl. Math.* **2011**, *235*, 1739–1743.
13. Ezquerro, J.A.; Hernández, M.A.; Romero, N. Solving nonlinear integral equations of Fredholm type with high order iterative methods. *J. Comput. Appl. Math.* **2011**, *236*, 1449–1463. [[CrossRef](#)]
14. Argyros, I.K.; Hilout, S. On the local convergence of fast two-step Newton-like methods for solving nonlinear equations. *J. Comput. Appl. Math.* **2013**, *245*, 1–9. [[CrossRef](#)]
15. Amat, S.; Busquier, S.; Grau, À.; Sánchez, M.G. Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications. *Appl. Math. Comput.* **2013**, *219*, 7954–7963. [[CrossRef](#)]
16. Wang, X.; Zhang, T. A family of Steffensen type methods with seventh-order convergence. *Numer. Algorithms* **2013**, *62*, 429–444. [[CrossRef](#)]
17. Sharma, J.R.; Guha, R.K.; Sharma, R. An efficient fourth order weighted-Newton method for systems of nonlinear equations. *Numer. Algorithms* **2013**, *62*, 307–323. [[CrossRef](#)]
18. Weerakoon, S.; Fernando, T.G.I. A variant of Newton's method with accelerated third-order convergence. *Appl. Math. Lett.* **2000**, *13*, 87–93. [[CrossRef](#)]
19. Wang, X.; Zhang, T.; Qian, W.; Teng, M. Seventh-order derivative-free iterative method for solving nonlinear systems. *Numer. Algorithms* **2015**, *70*, 545–558. [[CrossRef](#)]
20. Păvăloiu, I. On approximating the inverse of a matrix. *Creative Math.* **2003**, *12*, 15–20.
21. Diaconu, A.; Păvăloiu, I. Asupra unor metode iterative pentru rezolvarea ecuațiilor operaționale neliniare. *Rev. Anal. Numer. Teor. Aproximației* **1973**, *2*, 61–69.

