Research Article

Convergence Analysis and Numerical Study of a Fixed-Point Iterative Method for Solving Systems of Nonlinear Equations

Na Huang and Changfeng Ma

School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China

Correspondence should be addressed to Changfeng Ma; macf@fjnu.edu.cn

Received 12 February 2014; Accepted 20 February 2014; Published 24 March 2014

Academic Editor: Pu-yan Nie

Copyright © 2014 N. Huang and C. Ma. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a fixed-point iterative method for solving systems of nonlinear equations. The convergence theorem of the proposed method is proved under suitable conditions. In addition, some numerical results are also reported in the paper, which confirm the good theoretical properties of our approach.

1. Introduction

One of the basic problems in mathematics is how to solve nonlinear equations
\[ f(x) = 0. \]  
(1)

In order to solve these equations, we can use iterative methods such as Newton’s method and its variants. Recently, there has been some progress on iterative methods with higher order of convergence using decomposition techniques; see [1–15] and the reference therein.

The zeros of a nonlinear equation cannot in general be expressed in closed form; thus we have to use approximate methods. Nowadays, we often use iterative methods to get the approximate solution of the system (1); the best known method is the classical Newton’s method. Recently, there has been some progress on solving the system (1), which allows us to get the iterative formula by using essentially Taylor’s polynomial (see [16, 17]), quadrature formulas (see [7, 9–12]), homotopy perturbation method (see [8]), and so on.

In this paper, we will present a new fixed point iterative method for solving the system (1) and prove that the method is cubic convergent under suitable conditions.

This paper is organized as follows. In Section 2, we introduce new iterative methods to solve (1). In Section 3 we extend these methods to solve systems of nonlinear equations, and we also prove convergence of the proposed method. Some numerical results are reported in Section 4, while the paper is concluded in the last section.

2. Iterative Methods and Convergence Analysis

We now consider the following nonlinear equation:
\[ f(x) = 0, \quad x \in [a, b]. \]  
(2)

Assume that \( \alpha^* \) is a simple root of (2); that is, \( f(\alpha^*) = 0 \). For \( \alpha, \alpha_k \in [a, b] \), using Taylor’s formula, we have
\[
  f(\alpha) = f(\alpha_k) + f'(\alpha_k)(\alpha - \alpha_k) + \frac{1}{2!} f''(\alpha_k)(\alpha - \alpha_k)^2 + \frac{1}{3!} f'''(\alpha_k)(\alpha - \alpha_k)^3 + \cdots + \frac{1}{(r-1)!} f^{(r-1)}(\alpha_k)(\alpha - \alpha_k)^{r-1} 
  + \int_0^1 (1-t)^{r-1} f^{(r)}(\alpha_k + t(\alpha - \alpha_k)) (\alpha - \alpha_k)^r dt.
\]  
(3)

Taking \( r = 1 \) in the above equality, we get
\[
  f(\alpha) = f(\alpha_k) + \int_0^1 f'(\alpha_k + t(\alpha - \alpha_k)) (\alpha - \alpha_k) dt. \]  
(4)
If the value of \( f'(\alpha_k + t(\alpha - \alpha_k)) \) in the interval \([0, 1]\) is replaced with its value in \( t = 0\), that is, with \( f'(\alpha_k)\), then we have
\[
\int_0^1 f'(\alpha_k + t(\alpha - \alpha_k))(\alpha - \alpha_k)\,dt = f'(\alpha_k)(\alpha - \alpha_k). \tag{5}
\]
By using (5) in (4), we have
\[
f(\alpha) \equiv f(\alpha_k) + f'(\alpha_k)(\alpha - \alpha_k) . \tag{6}
\]
We can get an iterative method from (6) to solve the system (2); it is the famous Newton's formula
\[
\alpha_{k+1} = \alpha_k - \frac{f(\alpha_k)}{f'(\alpha_k)} . \tag{7}
\]
The formula (7) has already been proved to be quadratically convergent. Now we begin to deduce a higher order iterative method. In fact, if we estimate \( f'(\alpha_k + t(\alpha - \alpha_k)) \) in the interval \([0, 1]\) by its value in \( t = 1\), that is, by \( f'(\alpha)\), then we have
\[
\int_0^1 f'(\alpha_k + t(\alpha - \alpha_k))(\alpha - \alpha_k)\,dt = f'(\alpha)(\alpha - \alpha_k) + R[f(\alpha)] . \tag{8}
\]
By using (8) in (4), and cutting off the error \( R[f(\alpha)] \) (still use \( \approx\)), we have
\[
f(\alpha) = f(\alpha_k) + f'(\alpha)(\alpha - \alpha_k) . \tag{9}
\]
Let \( \alpha_{k+1} \) be the solution of (9); we can obtain a new iterative method
\[
\alpha_{k+1} = \alpha_k - \frac{f(\alpha_k) - f(\alpha_{k+1})}{f'(\alpha_k)} . \tag{10}
\]
Since the iterative method (10) is implicit-type method, we use classical Newton's formula (7) as predictor and then use the above scheme (10) as corrector; in this way, we can get a workable iterative method.

**Algorithm 1.**

**Step 0 (initialization).** Choose the initial value \( \alpha_0 \in N(\alpha^*, \delta) \), where \( \alpha^* \) is a certain real zero of nonlinear mapping \( f(\alpha) \) and \( \delta > 0 \) is a sufficiently small constant. Take the stopping criteria \( \epsilon_1, \epsilon_2 > 0 \). Set \( k := 0 \).

**Step 1 (the predictor step).** Compute the predictor
\[
\beta_k = \alpha_k - \frac{f(\alpha_k)}{f'(\alpha_k)} . \tag{11}
\]
**Step 2 (the corrector step).** Computing the corrector
\[
\alpha_{k+1} = \alpha_k - \frac{f(\alpha_k) - f(\beta_k)}{f'(\beta_k)} . \tag{12}
\]
**Step 3.** If \( |f(\alpha_{k+1})| \leq \epsilon_1 \) or \( |\alpha_{k+1} - \alpha_k| \leq \epsilon_2 \) then stop; otherwise, set \( \alpha_k := \alpha_{k+1}, k := k + 1 \); go to Step 1.

In this section, we consider the convergence and convergent rate of Algorithm 1. We obtain a convergence theorem as follows.

**Theorem 2.** Let \( \alpha^* \in I \) be a simple zero of sufficiently differentiable function \( f : [a, b] \subseteq \mathbb{R} \rightarrow \mathbb{R} \). If \( \alpha_k \) is sufficiently close to \( \alpha^* \), then the two-step iterative method defined by (11)-(12) has three-order convergence and satisfies the following error equation:
\[
e_{k+1} = e_k^3 + O(e_k^4) , \tag{13}
\]
where \( e_k = \alpha_k - \alpha^* \) and \( c = f'''(\alpha^*)/6f'(\alpha^*) \).

**Proof.** By (12) we get
\[
e_{k+1} = e_k - \frac{f(\beta_k) - f(\alpha_k)}{f'(\beta_k)} . \tag{14}
\]
Multiplying the above equation by \( f'/(\beta_k) \), we can get
\[
f'(\beta_k)e_{k+1} = f'(\beta_k)e_k + [f(\beta_k) - f(\alpha_k)] . \tag{15}
\]
Let \( \alpha = \alpha^* \) in (3) and \( f(\alpha^*) = 0 \); we have
\[
0 = f(\alpha_k) + f'(\alpha_k)\left(\alpha - \alpha_k + \frac{1}{2!}f''(\alpha_k)(\alpha - \alpha_k)^2 + \frac{1}{3!}f'''(\alpha_k)(\alpha - \alpha_k)^3 + O\left((\alpha - \alpha_k)^4\right)\right); \tag{16}
\]
thus,
\[
f(\alpha_k) = f'(\alpha_k)e_k - \frac{1}{2}f''(\alpha_k)e_k^2 + \frac{1}{6}f'''(\alpha_k)e_k^3 + O(e_k^4) . \tag{17}
\]
Dividing both sides of the above equation by \( f'(\alpha_k) \), we can get
\[
f(\alpha_k)f'(\alpha_k) = e_k - \frac{1}{2}f''(\alpha_k)e_k^2 + \frac{1}{6}f'''(\alpha_k)e_k^3 + O(e_k^4) \tag{18}
\]
\[
= e_k - \frac{1}{2}f''(\alpha_k)e_k^2 + O(e_k^3) .
\]
Furthermore, let \( \alpha = \beta_k \) in (3); we have
\[
f(\beta_k) = f(\alpha_k) + f'(\alpha_k)(\beta_k - \alpha_k) + \frac{1}{2!}f''(\alpha_k)(\beta_k - \alpha_k)^2 + \frac{1}{3!}f'''(\alpha_k)(\beta_k - \alpha_k)^3 + O((\beta_k - \alpha_k)^4) . \tag{19}
\]
From (11) we get
\[
\beta_k - \alpha_k = \frac{f(\alpha_k)}{f'(\alpha_k)} . \tag{20}
\]
It follows from the above equation that
\[
f(β_k) = f(α_k) - f'(α_k) \left[ \frac{f(α_k)}{f'(α_k)} \right] + \frac{1}{2} f''(α_k) \left[ \frac{f(α_k)}{f'(α_k)} \right]^2 \\
- \frac{1}{6} f'''(α_k) \left[ \frac{f(α_k)}{f'(α_k)} \right]^3 + O \left( \left[ \frac{f(α_k)}{f'(α_k)} \right]^4 \right) \\
= \frac{1}{2} f''(α_k) \left[ \frac{f(α_k)}{f'(α_k)} \right]^2 - \frac{1}{6} f'''(α_k) \left[ \frac{f(α_k)}{f'(α_k)} \right]^3 \\
+ O \left( \left[ \frac{f(α_k)}{f'(α_k)} \right]^4 \right).
\] (21)

By applying (18) we have
\[
f(β_k) = \frac{1}{2} f''(α_k) \left[ e_k - \frac{1}{2} \frac{f''(α_k)}{f'(α_k)} e_k^2 + O(e_k^3) \right]^2 \\
- \frac{1}{6} f'''(α_k) \left[ e_k - \frac{1}{2} \frac{f''(α_k)}{f'(α_k)} e_k^2 + O(e_k^3) \right]^3 \\
+ O(e_k^4).
\] (22)

After some manipulations we obtain
\[
f(β_k) = \frac{1}{2} f''(α_k) e_k^2 - \frac{1}{2} \left[ \frac{f''(α_k)}{f'(α_k)} \right]^2 e_k^3 \\
- \frac{1}{6} f'''(α_k) e_k^3 + O(e_k^4).
\] (23)

By substituting (17), (23), and (26) into (15) we have
\[
f'(β_k) e_{k+1} \\
= f'(β_k) e_k + [f(β_k) - f(α_k)] \\
= \left\{ f'(α_k) - f''(α_k) e_k + \frac{1}{2} \left[ \frac{f''(α_k)}{f'(α_k)} \right]^2 e_k^2 \\
+ \frac{1}{2} f'''(α_k) e_k^2 + O(e_k^3) \right\} e_k \\
+ \left\{ \frac{1}{2} f''(α_k) e_k^2 - \frac{1}{2} \left[ \frac{f''(α_k)}{f'(α_k)} \right]^2 e_k^3 \\
- \frac{1}{6} f'''(α_k) e_k^3 + O(e_k^4) \right\} \\
- \left\{ f'(α_k) e_k - \frac{1}{2} f''(α_k) e_k^2 \\
+ \frac{1}{6} f'''(α_k) e_k^3 + O(e_k^4) \right\} \\
= \frac{1}{6} f'''(α_k) e_k^3 + O(e_k^4).
\] (27)

Note that
\[
f'(β_k) = f'(α^*) + O(e_k),
\] (28)
\[
f'''(α_k) = f'''(α^*) + O(e_k).
\]

Then, it follows from (27) that
\[
e_{k+1} = \frac{1}{6} \left[ \frac{f'''(α^*)}{f'(α^*)} + O(e_k) \right] e_k^3 \\
= \frac{1}{6} \left[ \frac{f'''(α^*)}{f'(α^*)} + O(e_k) \right] e_k^3 \\
+ O(e_k^4) = \frac{1}{6} f'''(α^*) e_k^3 + O(e_k^4).
\] (29)

This proves the conclusion of the theorem and the proof is completed. □

3. The n-Dimensional Case

In this section, we consider the n-dimensional case of the method, and we also study these iterative methods’ order of convergence. Consider the system of nonlinear equations
\[
\begin{align*}
f_1(x_1, x_2, \ldots, x_n) &= 0, \\
f_2(x_1, x_2, \ldots, x_n) &= 0, \\
&\vdots \\
f_n(x_1, x_2, \ldots, x_n) &= 0,
\end{align*}
\] (30)
where each function \( f_i(x_1, x_2, \ldots, x_n), \ i = 1, 2, \ldots, n \), maps a vector \( x = (x_1, x_2, \ldots, x_n)^T \) of the \( n \)-dimensional space \( \mathbb{R}^n \) into the real line \( R \). The system (30) 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 \( F(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T \). Thus, the system (30) can be written in the form

\[
F(x) = 0. \tag{31}
\]

Let \( F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \) be a sufficiently differentiable function on a convex set \( D \subseteq \mathbb{R}^n \) and let \( x^* \) be a real zero of the nonlinear mapping \( F(x) \); that is, \( F(x^*) = 0 \). For any \( x, x_k \in D \), we may write Taylor’s expansion for \( F \) as follows (see [17]):

\[
F(x) = F(x_k) + F'(x_k)(x - x_k)
+ \frac{1}{2!}F''(x_k)(x - x_k)^2
+ \frac{1}{3!}F'''(x_k)(x - x_k)^3 + \cdots + \frac{1}{(r-1)!}F^{(r-1)}(x_k) + \int_0^1 \frac{(1-t)^{r-1}}{(r-1)!} F^{(r)}(x_k + t(x - x_k))(x - x_k)^r \, dt;
\]

for \( r = 1 \), we have

\[
F(x) = F(x_k) + \int_0^1 F'(x_k + t(x - x_k))(x - x_k) \, dt. \tag{33}
\]

If we estimate \( F'(x_k + t(x - x_k)) \) in the interval \([0, 1]\) by its value in \( t = 0 \), that is, by \( F'(x_k) \), then we have

\[
\int_0^1 F'(x_k + t(x - x_k))(x - x_k) \, dt \equiv F'(x_k)(x - x_k). \tag{34}
\]

By using (34) in (33), we have

\[
F(x) \equiv F(x_k) + F'(x_k)(x - x_k). \tag{35}
\]

We can get an iterative method from (35) to solve the system (31); it is known as Newton’s method

\[
x_{k+1} = x_k - F'(x_k)^{-1}F(x_k). \tag{36}
\]

The method (36) has already been proved that it has quadratic convergence. Now we begin to deduce a higher order iterative method. If we estimate \( F'(x_k + t(x - x_k)) \) in the interval \([0, 1]\) by its value in \( t = 1 \), that is, by \( F'(x) \), then we have

\[
\int_0^1 F'(x_k + t(x - x_k))(x - x_k) \, dt
= F'(x)(x - x_k) + R[F(x)]. \tag{37}
\]

By using (37) in (33), and cutting off the error \( R[F(x)] \) (still use “=”), we have

\[
F(x) = F(x_k) + F'(x)(x - x_k). \tag{38}
\]

Let \( x_{k+1} \) be the solution of (38); we can obtain a new iterative method

\[
x_{k+1} = x_k - F'(x_k)^{-1}[F(x_k) - F(x_{k+1})]. \tag{39}
\]

On the other hand, from the easy Newton’s method (see [4]), we obtain

\[
x_{k+1} = x_k - F'(x_k)^{-1}[F(x_k) + F(x_{k+1})]. \tag{40}
\]

Now we consider the convex combination of (39) and (40). Let \( \alpha \geq 0, \ \beta \geq 0, \ \text{and} \ \alpha + \beta = 1; \ \text{then we can deduce from (39) and (40) that the following iterative formula holds:}

\[
x_{k+1} = x_k - \left[ \alpha F'(x_{k+1})^{-1} + \beta F'(x_k)^{-1} \right] x_k + \left[ F(x_k) + F(x_{k+1}) \right] \tag{41}
+ 2\alpha F'(x_{k+1})^{-1}F(x_{k+1}).
\]

Since the iterative method (41) is implicit-type method, we use Newton’s method as predictor and then use the new method (41) as corrector; in this way, we can get a workable iterative method.

**Algorithm 3.**

*Step 0 (initialization).* Choose the initial value \( x_0 \in \mathcal{N}(x^*, \delta), \ \alpha \in [0, 1] \) and \( \beta = 1 - \alpha \), where \( x^* \) is certain real zero of nonlinear mapping \( F(x) \) and \( \delta > 0 \) is a sufficiently small constant. Take the stopping criterions \( \epsilon_1, \epsilon_2 > 0 \). Set \( k := 0 \).

*Step 1 (the predictor step).* Compute the predictor

\[
y_k = x_k - F'(x_k)^{-1}F(x_k). \tag{42}
\]

*Step 2 (the corrector step).* Computing the corrector

\[
x_{k+1} = x_k - \left[ \alpha F'(\beta_k)^{-1} + \beta F'(x_k)^{-1} \right] x_k + \left[ F(\beta_k) + F(x_k) \right] \tag{43}
+ 2\alpha F'(\beta_k)^{-1}F(\beta_k).
\]

*Step 3.* If \( \|F(x_{k+1})\| \leq \epsilon_1 \) or \( \|x_{k+1} - x_k\| \leq \epsilon_2 \) then stop; otherwise, set \( k := k + 1; \) go to Step 1.

**Remark 4.** If we take \( \beta = 0, \alpha = 1 \), our algorithms (42) and (43) can be written in the following form:

\[
y_k = x_k - F'(x_k)^{-1}F(x_k),
\]

\[
x_{k+1} = x_k - \left[ F'(y_k) \right]^{-1} \left[ F(x_k) - F(y_k) \right]. \tag{44}
\]

Notice that, at each iteration, the number of functional evaluations is \( 2n^2 + 2n \).

In this section, we consider the convergence and convergent rate of Algorithm 1. We obtain a convergence theorem as follows.
Theorem 5. Let $F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a sufficiently differentiable function on a convex set $D \subseteq \mathbb{R}^n$ containing a root $x^*$ of the nonlinear system (31). The iterative method (42)-(43) has cubic convergence and satisfies the error equation

$$e_{k+1} = F'(x^*)^{-1} \times \left[ \frac{1}{6} \alpha F''(x^*) + \frac{1}{2} \beta F''(x^*) F'(x^*)^{-1} \right] e_k^3 + O \left( \|e_k^2\| \right).$$

Proof. Defining $e_k = x_k - x^*$, from (42) and (43), we have

$$e_{k+1} = e_k - \left[ \alpha F'(x_k)^{-1} + \beta F'(x_k)^{-1} \right] \times \left[ F(y_k) + F(x_k) \right] + 2 \alpha F'(y_k)^{-1} F(y_k).$$

Premultiplying the above equation by $F'(y_k)$, we can get

$$F'(y_k) e_{k+1} = F'(y_k) e_k - \left[ \alpha I + \beta F'(y_k) F'(y_k)^{-1} \right] \times \left[ F(y_k) + F(x_k) \right] + 2 \alpha F'(y_k)^{-1} F(y_k).$$

Let $x = x^*$ in (32) and $F(x^*) = 0$; we have

$$0 = F(x_k) + F'(x_k) (x^* - x_k)$$

$$+ \frac{1}{2} F''(x_k) (x^* - x_k)^2$$

$$+ \frac{1}{3} F'''(x_k) (x^* - x_k)^3 + O \left( \|x^* - x_k\|^4 \right);$$

thus,

$$F(x_k) = F'(x_k) e_k - \frac{1}{2} F''(x_k) e_k^2$$

$$+ \frac{1}{6} F'''(x_k) e_k^3 + O \left( \|e_k^2\| \right).$$

Multiplying $F'(x_k)^{-1}$ to the two sides of the above equation, we obtain

$$F'(x_k)^{-1} F(x_k) = e_k - \frac{1}{2} F'(x_k)^{-1} F''(x_k) e_k^2$$

$$+ \frac{1}{6} F'(x_k)^{-1} F'''(x_k) e_k^3 + O \left( \|e_k^2\| \right)$$

$$= e_k - \frac{1}{2} F'(x_k)^{-1} F''(x_k) e_k^2 + O \left( \|e_k^2\| \right).$$

(50)

On the other hand, let $x = y_k$ in (32); we have

$$F(y_k) = F(x_k) + F'(x_k) (y_k - x_k)$$

$$+ \frac{1}{2} F''(x_k) (y_k - x_k)^2$$

$$+ \frac{1}{3} F'''(x_k) (y_k - x_k)^3 + O \left( \|y_k - x_k\|^4 \right).$$

(51)

From (42) we get $y_k - x_k = -F'(x_k)^{-1} F(x_k)$. It follows from the above equation that

$$F(y_k) = F(x_k) - F'(x_k) \left[ F'(x_k)^{-1} F(x_k) \right]$$

$$+ \frac{1}{2} F''(x_k) \left[ F'(x_k)^{-1} F(x_k) \right]^2$$

$$- \frac{1}{6} F'''(x_k) \left[ F'(x_k)^{-1} F(x_k) \right]^3$$

$$+ O \left( \left\| F'(x_k)^{-1} F(x_k) \right\|^4 \right).$$

(52)

By applying (50) we have

$$F(y_k) = \frac{1}{2} F''(x_k)$$

$$\times \left[ e_k - \frac{1}{2} F'(x_k)^{-1} F''(x_k) e_k^2 + O \left( \|e_k^2\| \right) \right]^2$$

$$- \frac{1}{6} F'''(x_k)$$

$$\times \left[ e_k - \frac{1}{2} F'(x_k)^{-1} F''(x_k) e_k^2 + O \left( \|e_k^2\| \right) \right]^3$$

$$+ O \left( \|e_k^2\| \right).$$

(53)

After some manipulations we obtain

$$F(y_k) = \frac{1}{2} F''(x_k) e_k^2$$

$$- \frac{1}{2} F''(x_k) F'(x_k)^{-1} F''(x_k) e_k^3$$

$$- \frac{1}{6} F'''(x_k) e_k^3 + O \left( \|e_k^2\| \right).$$

(54)
Now, applying Taylor's formula for $F'(y_k)$ at the point $x_k$, we have

$$F' (y_k) = F' (x_k) + F'' (x_k) (y_k - x_k)$$
$$+ \frac{1}{2} F''' (x_k) (y_k - x_k)^2$$
$$+ O \left( \| y_k - x_k \|^3 \right). \quad (55)$$

From (42), (50), and (55), we obtain

$$F' (y_k) = F' (x_k) - F'' (x_k)$$
$$\times \left[ e_k - \frac{1}{2} F'(x_k)^{-1} F'' (x_k) e_k^2 + O \left( \| e_k^3 \| \right) \right]$$
$$+ \frac{1}{2} F''' (x_k)$$
$$\times \left[ e_k - \frac{1}{2} F'(x_k)^{-1} F'' (x_k) e_k^2 + O \left( \| e_k^3 \| \right) \right]^2$$
$$+ O \left( \| e_k^3 \| \right). \quad (56)$$

And after some manipulations we can get

$$F' (y_k) = F' (x_k) - F'' (x_k) e_k$$
$$+ \frac{1}{2} F'' (x_k) F'(x_k)^{-1} F'' (x_k) e_k^2$$
$$+ \frac{1}{2} F''' (x_k) e_k^2 + O \left( \| e_k^3 \| \right). \quad (57)$$

Therefore, we have

$$\alpha I + \beta F' (y_k) F'(x_k)^{-1}$$
$$= I - \beta \left[ F'' (x_k) e_k - \frac{1}{2} F''' (x_k) F'(x_k)^{-1} \right.$$  
$$\times F'' (x_k) e_k^2 + \frac{1}{2} F''' (x_k) e_k^2 \left. \right] F'(x_k)^{-1}$$
$$+ O \left( \| e_k^3 \| \right). \quad (58)$$

By substituting (57), (58), (59), and (54) into (47) we have

$$F' (y_k) e_{k+1} = F' (y_k) e_k$$
$$- \left[ \alpha I + \beta F' (y_k) F'(x_k)^{-1} \right]$$
$$\times \left[ F (y_k) + F (x_k) \right] + 2 \alpha F (y_k)$$
$$= F' (x_k) e_k - F'' (x_k) e_k^2$$
$$+ \frac{1}{2} F''' (x_k) F'(x_k)^{-1} F'' (x_k) e_k^3$$
$$+ \frac{1}{2} F''' (x_k) e_k^3 + O \left( \| e_k^4 \| \right)$$
$$- \left[ I - \beta \left[ F'' (x_k) e_k - \frac{1}{2} F''' (x_k) \right.$$  
$$\times F'(x_k)^{-1} F'' (x_k) e_k^2$$
$$- \frac{1}{2} F''' (x_k) e_k^2 \left. \right] \right.$$  
$$\times F'(x_k)^{-1} + O \left( \| e_k^3 \| \right) \right.$$  
$$\left. \right]$$
$$\times \left[ F' (x_k) e_k - 2 F'' (x_k) F'(x_k)^{-1} \right.$$  
$$\times F'' (x_k) e_k^2 + O \left( \| e_k^3 \| \right) \right.$$  
$$+ 2 \alpha \left[ \frac{1}{2} F''' (x_k) e_k^2 - \frac{1}{2} F''' (x_k) \right.$$  
$$\times F'(x_k)^{-1} F'' (x_k) e_k^3$$
$$- \frac{1}{6} F''' (x_k) e_k^3 + O \left( \| e_k^4 \| \right) \right.$$  
$$\left. \right]$$
$$\times F'(x_k)^{-1} + O \left( \| e_k^3 \| \right) \right.$$  
$$\left. \right]$$
$$\times e_k^3 + O \left( \| e_k^4 \| \right); \quad (60)$$

that is,

$$e_{k+1} = F'(y_k)^{-1}$$
$$\times \left[ \frac{1}{6} \alpha F''' (x_k) + \frac{1}{2} \beta F'' (x_k) F'(x_k)^{-1} F'' (x_k) \right.$$  
$$\times e_k^3 + O \left( \| e_k^4 \| \right) \right.] \quad (61)$$

Notice that

$$F' (y_k) = F' (x^*) + O (e_k)$$
$$F^{(k)} (x_k) = F^{(k)} (x^*) + O (e_k), \quad k = 1, 2, 3. \quad (62)$$
Equation (61) can be written as follows:
\[
\begin{align*}
    e_{k+1} &= F'(x^*)^{-1} \\
    &\times \left[ \frac{1}{6} \alpha F'''(x^*) + \frac{1}{2} \beta F''(x^*) F'(x^*)^{-1} F''(x^*) \right] \\
    &\times e_k + O(\|e_k\|),
\end{align*}
\]
(63)
which proves the conclusion of our theorem. The proof is completed.

4. Numerical Examples

In this section we present some examples to illustrate the efficiency and the performance of the newly developed method (42)-(43) (present study HM). This new method was compared with Newton’s method (NM), the method of Aslam Noor and Waseem [12] (NRI), the method of Cordero et al. [13] (NAd), the method of Darvishi and Barati [4] (DV), and the method of Cordero and Torregrosa [10] (CT) in the number of iterations, CPU time, error, and convergence order. All computations were done using the PC with Pentium(R) Dual-Core CPU T4400 @2.20 GHz. All the programming is implemented in MATLAB 7.9. The convergence order \( p \) is computed approximately by the following formula:
\[
    p \approx \begin{cases} 
        \ln(\|x_{k+1} - x_k\|/\|x_k - x_{k-1}\|), & \text{if } k \geq 3; \\
        +\infty, & \text{otherwise}.
    \end{cases}
\]
(64)
As the iterative formula (43) contains parameters \( \alpha \) and \( \beta \), we make the numerical examples based on \( \alpha = 1 \) and \( \beta = 0 \).

Example 1. The test function is as follows (see [10]):
\[
    F(x) = (x - 1)^6 - 1.
\]
(65)
This problem has a solution \( x^* = 2 \). We test this problem by using \( x_0 = 4 \) as a starting point. The test results are listed in Table 1.

Example 2. The test function is as follows (see [9, 10, 13, 14]):
\[
    F(x_1, x_2) = \left( \frac{x_1 + e^{x_2} - \cos x_2}{3x_1 - x_2 - \sin x_2} \right).
\]
(66)
This problem has a solution \( x^* = (0, 0)^T \). We test this problem by using initial value \( x_0 = (0.3, -0.3)^T \) as a starting point. The test results are listed in Table 2.

Example 3. The test function is as follows [4]:
\[
\begin{align*}
    x_1^2 + x_2^2 + x_3^2 - 1 &= 0, \\
    2x_1^2 + x_2^2 - 4x_3 &= 0, \\
    3x_1^2 - 4x_2^2 + x_3^2 &= 0.
\end{align*}
\]
(67)
We test this problem by using \( x_0 = (0.5, 0.5, 0.5)^T \). The test results are listed in Table 3.

Example 4. The test function is \( F(x) = (f_1(x), f_2(x), f_3(x), f_4(x))^T \), where \( x = (x_1, x_2, x_3, x_4)^T \in \mathbb{R}^4 \) and \( f_i : \mathbb{R}^4 \to \mathbb{R}, i = 1, 2, 3, 4, \) such that (see [9, 10, 13])
\[
\begin{align*}
    f_1(x) &= x_2x_3 + x_4(x_2 + x_3), \\
    f_2(x) &= x_1x_3 + x_4(x_1 + x_3), \\
    f_3(x) &= x_1x_2 + x_4(x_1 + x_2), \\
    f_4(x) &= x_1x_2 + x_3x_4 + x_2x_3 - 1.
\end{align*}
\]
(68)
This problem has two solutions: they are \( x^* = \left( -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{2\sqrt{3}} \right)^T \) and \( x^{**} = (1, 1, 1, 2)^T \). We test this problem by using \( x_0 = (1, 1, 1, -2)^T \) (the iterative sequence converges to \( x^* \)) and \( x_0 = (1, 1, 1, -2)^T \) (the iterative sequence converges to \( x^{**} \)) as starting point, respectively. The test results are listed in Tables 4 and 5.

Example 5. The test function is as follows (see [10, 13, 14]):
\[
    F_i(x) = x_i x_{i+1} - 1, \quad i = 1, \ldots, n - 1, \quad F_n(x) = x_1 x_n - 1.
\]
(70)
This problem has two solutions: they are \( x^* = (1, 1, \ldots, 1)^T \) and \( x^{**} = (-1, -1, \ldots, 1)^T \). We test this problem by using \( x_0 = (2, 2, \ldots, 2)^T \) (the iterative sequence converges to \( x^* \)) and \( x_0 = (-2, -2, \ldots, -2)^T \) (the iterative sequence converges to \( x^{**} \)) as starting point, respectively. The test results are listed in Tables 6 and 7, which are obtained for \( n = 101 \).

Example 6. Consider the unrestrained optimum problem (see [18]):
\[
    \min_{x \in \mathbb{R}^n} f(x),
\]
(71)
where \( f : \mathbb{R}^n \to \mathbb{R} \) is defined by
\[
    f(x) = \sum_{i=2}^{n} \left[ (x_{i-1}^2 + x_i^2)^2 - 4x_{i-1} + 3 \right].
\]
(72)
By KKT condition we have
\[
    F(x) = \frac{1}{4} \sqrt[n]{f(x)} = 0,
\]
(73)
where \( F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T \),
\[
    F_1(x) = x_1 \left( x_1^2 + x_2^2 \right) - 1,
\]
(74)
\[
    F_i(x) = x_i \left( x_{i-1}^2 + 2x_i^2 + x_{i+1}^2 \right) - 1, \quad i = 2, \ldots, n - 1,
\]
\[
    F_n(x) = x_n \left( x_{n-1}^2 + x_n^2 \right).
\]
Table 1: Numerical results of Example 1 with initial value $x_0 = 4$.

| Method | Iterations | CPU time | Error         | Convergence order $(p)$ |
|--------|------------|----------|---------------|-------------------------|
| NR1   | 6          | 0.0006   | $8.2276e - 009$ | 2.95                   |
| NAdl  | 6          | 0.0007   | $7.1942e - 014$ | 3.91                   |
| DV    | 7          | 0.0018   | $7.1783e - 012$ | 2.98                   |
| CT    | 6          | 0.0005   | $7.1828e - 009$ | 2.95                   |
| HM    | 6          | 0.0003   | $6.6613e - 015$ | 3.00                   |

Table 2: Numerical results of Example 2 with initial value $x_0 = (0.3, -0.3)^T$.

| Method | Iterations | CPU time | Error         | Convergence order $(p)$ |
|--------|------------|----------|---------------|-------------------------|
| NR1   | 3          | 0.0007   | $1.3502e - 006$ | 3.15                   |
| NAdl  | 3          | 0.0011   | $1.4541e - 006$ | 4.71                   |
| DV    | 4          | 0.0008   | $2.8501e - 013$ | 3.02                   |
| CT    | 3          | 0.0007   | $1.3487e - 006$ | 3.15                   |
| HM    | 3          | 0.0004   | $1.0359e - 014$ | 5.69                   |

Table 3: Numerical results of Example 3 with initial value $x_0 = (0.5, 0.5, 0.5)^T$.

| Method | Iterations | CPU time | Error         | Convergence order $(p)$ |
|--------|------------|----------|---------------|-------------------------|
| NR1   | 4          | 0.0008   | $5.5511e - 017$ | 3.24                   |
| NAdl  | 4          | 0.0007   | $5.6038e - 006$ | 4.51                   |
| DV    | 4          | 0.0007   | $0.0000e + 000$ | 3.62                   |
| CT    | 4          | 0.0009   | $5.5511e - 017$ | 3.24                   |
| HM    | 3          | 0.0005   | $3.8599e - 012$ | Inf                    |

Table 4: Numerical results of Example 4 with initial value $x_0 = (-1, -1, -1, -1)^T$.

| Method | Iterations | CPU time | Error         | Convergence order $(p)$ |
|--------|------------|----------|---------------|-------------------------|
| NR1   | 4          | 0.0013   | $2.9032e - 014$ | 3.32                   |
| NAdl  | 3          | 0.0011   | $5.6038e - 006$ | 4.40                   |
| DV    | 4          | 0.0008   | $1.2469e - 011$ | 3.38                   |
| CT    | 4          | 0.0015   | $2.9088e - 014$ | 3.32                   |
| HM    | 3          | 0.0006   | $3.3483e - 008$ | 4.68                   |

Table 5: Numerical results of Example 4 with initial value $x_0 = (1, 1, 1, -2)^T$.

| Method | Iterations | CPU time | Error         | Convergence order $(p)$ |
|--------|------------|----------|---------------|-------------------------|
| NR1   | 4          | 0.0016   | $2.9365e - 014$ | 3.29                   |
| NAdl  | 3          | 0.0010   | $5.7635e - 006$ | 4.13                   |
| DV    | 4          | 0.0008   | $1.2666e - 011$ | 3.33                   |
| CT    | 4          | 0.0014   | $2.9421e - 014$ | 3.29                   |
| HM    | 3          | 0.0006   | $3.4298e - 008$ | 4.43                   |

Table 6: Numerical results of Example 5 with initial value $x_0 = (2, 2, \ldots, 2)^T$.

| Method | Iterations | CPU time | Error         | Convergence order $(p)$ |
|--------|------------|----------|---------------|-------------------------|
| NR1   | 4          | 0.0104   | $2.6223e - 013$ | 2.98                   |
| NAdl  | 3          | 0.0105   | $5.2789e - 006$ | 3.23                   |
| DV    | 4          | 0.0072   | $6.9607e - 011$ | 2.96                   |
| CT    | 4          | 0.0085   | $2.6223e - 013$ | 2.98                   |
| HM    | 3          | 0.0047   | $4.6461e - 008$ | 3.58                   |
We test this problem by using $x_0 = (1, 1, \ldots, 1)^T$ as starting point. The test results are listed in Table 8, which are obtained for $n = 512$.

**Example 7.** Consider the discrete two-point boundary value problems (see [19]):

$$F(x) = Ax + \frac{1}{(n+1)^2}G(x) = 0,$$

where $A$ is an $n \times n$ tridiagonal matrix defined by

$$A = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 \end{pmatrix},$$

$G(x) = (G_1(x), G_2(x), \ldots, G_n(x))^T$, and $G_i(x) = \sin x_i - 1$, $i = 1, 2, \ldots, n$. We test this problem by using $x_0 = (1, 1, \ldots, 1)^T$ as starting point. The test results are listed in Table 9, which are obtained for $n = 1024$.

### 5. Conclusion

From the seven examples in Section 4, we can see that the newly developed method (42)-(43) has the advantages of fast convergence speed (we can get from the CPU time), small number of iterations. Especially, the value of convergence order $p$ that appears in Tables 2–7 is the highest compared to the other four methods. Although our method's convergence order is not always higher than the method of Cordero and Martínez and Torregrosa (NAd1), ours is superior in the number of iterations and CPU time to the other four methods. In a word, our method (42)-(43) is quite robust and effective.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Authors’ Contribution

All authors contributed equally and significantly in writing this paper. All authors read and approved the final paper.

### Acknowledgments

This project is supported by National Natural Science Foundation of China (Grant no. 11071041), Fujian Natural Science Foundation (Grant no. 2013J01006), and R & D of Key Instruments and Technologies for Deep Resources Prospecting (the National R & D Projects for Key Scientific Instruments) under Grant no. ZDYZ2012-1-02-04.
References

[1] F. A. Potra and V. Pták, Nondiscrete Induction and Iterative Processes, vol. 103 of Research Notes in Mathematics, Pitman, Boston, Mass, USA, 1984.

[2] M. A. Noor, “New iterative schemes for nonlinear equations,” Applied Mathematics and Computation, vol. 187, no. 2, pp. 937–943, 2007.

[3] S. Abbasbandy, “Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method,” Applied Mathematics and Computation, vol. 145, no. 2-3, pp. 887–893, 2003.

[4] M. T. Darvishi and A. Barati, “A third-order Newton-type method to solve systems of nonlinear equations,” Applied Mathematics and Computation, vol. 187, no. 2, pp. 630–635, 2007.

[5] C. Chun, “A new iterative method for solving nonlinear equations,” Applied Mathematics and Computation, vol. 178, no. 2, pp. 415–422, 2006.

[6] C. Chun, “Iterative methods improving newton's method by the decomposition method,” Computers and Mathematics with Applications, vol. 50, no. 10–12, pp. 1559–1568, 2005.

[7] M. T. Darvishi and A. Barati, “Super cubic iterative methods to solve systems of nonlinear equations,” Applied Mathematics and Computation, vol. 188, no. 2, pp. 1678–1685, 2007.

[8] A. Golbabai and M. Javidi, “A new family of iterative methods for solving system of nonlinear algebraic equations,” Applied Mathematics and Computation, vol. 190, no. 2, pp. 1717–1722, 2007.

[9] A. Cordero and J. R. Torregrosa, “Variants of Newton's method for functions of several variables,” Applied Mathematics and Computation, vol. 183, no. 1, pp. 199–208, 2006.

[10] A. Cordero and J. R. Torregrosa, “Variants of Newton's Method using fifth-order quadrature formulas,” Applied Mathematics and Computation, vol. 190, no. 1, pp. 686–698, 2007.

[11] M. Frontini and E. Sormani, “Third-order methods from quadrature formulae for solving systems of nonlinear equations,” Applied Mathematics and Computation, vol. 149, no. 3, pp. 771–782, 2004.

[12] M. Aslam Noor and M. Waseem, “Some iterative methods for solving a system of nonlinear equations,” Computers and Mathematics with Applications, vol. 57, no. 1, pp. 101–106, 2009.

[13] A. Cordero, E. Martínez, and J. R. Torregrosa, “Iterative methods of order four and five for systems of nonlinear equations,” Journal of Computational and Applied Mathematics, vol. 231, no. 2, pp. 541–551, 2009.

[14] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, “Accelerated methods of order $2^p$ for systems of nonlinear equations,” Journal of Computational and Applied Mathematics, vol. 233, no. 10, pp. 2696–2702, 2010.

[15] N. Huang, C. F. Ma, and Z. G. Liu, “A new extragradient-like method for solving variational inequality problems,” Fixed Point Theory and Applications, vol. 2012, article 223, 2012.

[16] R. L. Burden and J. D. Faires, Numerical Analysis, PWS, Boston, Mass, USA, 7th edition, 2001.

[17] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, 1970.

[18] E. Yamakawa and M. Fukushima, “Testing parallel variable transformation,” Computational Optimization and Applications, vol. 13, no. 1-3, pp. 253–274, 1999.

[19] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, “Testing unconstrained optimization software,” ACM Transactions on Mathematical Software, vol. 7, pp. 17–41, 1981.