

Research Article

A New Second-Order Iteration Method for Solving Nonlinear Equations

Shin Min Kang,¹ Arif Rafiq,² and Young Chel Kwun³

¹ Department of Mathematics and RINS, Gyeongsang National University, Jinju 660-701, Republic of Korea

² School of Computer Science and Mathematics, Hajvery University, 43-52 Industrial Area, Gulberg III, Lahore 54660, Pakistan

³ Department of Mathematics, Dong-A University, Busan 614-714, Republic of Korea

Correspondence should be addressed to Young Chel Kwun; yckwun@dau.ac.kr

Received 26 January 2013; Accepted 13 March 2013

Academic Editor: Gue Lee

Copyright © 2013 Shin Min Kang et al. 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 establish a new second-order iteration method for solving nonlinear equations. The efficiency index of the method is 1.4142 which is the same as the Newton-Raphson method. By using some examples, the efficiency of the method is also discussed. It is worth to note that (i) our method is performing very well in comparison to the fixed point method and the method discussed in Babolian and Biazar (2002) and (ii) our method is so simple to apply in comparison to the method discussed in Babolian and Biazar (2002) and involves only first-order derivative but showing second-order convergence and this is not the case in Babolian and Biazar (2002), where the method requires the computations of higher-order derivatives of the nonlinear operator involved in the functional equation.

1. Introduction

Our problem, to recall, is solving equations in one variable. We are given a function f and would like to find at least one solution to the equation $f(x) = 0$. Note that, priorly, we do not put any restrictions on the function f ; we need to be able to evaluate the function; otherwise, we cannot even check that a given solution $x = \alpha$ is true, that is, $f(\alpha) = 0$. In reality, the mere ability to be able to evaluate the function does not suffice. We need to assume some kind of “good behavior.” The more we assume, the more potential we have, on the one hand, to develop fast algorithms for finding the root. At the same time, the more we assume, the fewer the functions are going to satisfy our assumptions! This is a fundamental paradigm in numerical analysis.

We know that one of the fundamental algorithm for solving nonlinear equations is so-called fixed-point iteration method [1].

In the fixed-point iteration method for solving the nonlinear equation $f(x) = 0$, the equation is usually rewritten as

$$x = g(x), \quad (1)$$

where

- (i) there exists $[a, b]$ such that $g(x) \in [a, b]$ for all $x \in [a, b]$,
- (ii) there exists $[a, b]$ such that $|g'(x)| \leq L < 1$ for all $x \in [a, b]$.

Considering the following iteration scheme:

$$x_{n+1} = g(x_n), \quad n = 0, 1, 2, \dots, \quad (2)$$

and starting with a suitable initial approximation x_0 , we build up a sequence of approximations, say $\{x_n\}$, for the solution of the nonlinear equation, say α . The scheme will converge to the root α , provided that

- (i) the initial approximation x_0 is chosen in the interval $[a, b]$,
- (ii) g has a continuous derivative on (a, b) ,
- (iii) $|g'(x)| < 1$ for all $x \in [a, b]$,
- (iv) $a \leq g(x) \leq b$ for all $x \in [a, b]$ (see [1]).

The order of convergence for the sequence of approximations derived from an iteration method is defined in the literature, as follows.

Definition 1. Let $\{x_n\}$ converge to α . If there exist an integer constant p and real positive constant C such that

$$\lim_{n \rightarrow \infty} \left| \frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} \right| = C, \quad (3)$$

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 $g(x_n)$

$$\begin{aligned} g(x_n) &= g(x) + \frac{g'(x)}{1!} (x_n - x) + \frac{g''(x)}{2!} (x_n - x)^2 + \dots \\ &\quad + \frac{g^{(k)}(x)}{k!} (x_n - x)^k + \dots \end{aligned} \quad (4)$$

Using (1) and (2) in (4) we have

$$\begin{aligned} x_{n+1} - x &= g'(x) (x_n - x) + \frac{g''(x)}{2!} (x_n - x)^2 + \dots \\ &\quad + \frac{g^{(k)}(x)}{k!} (x_n - x)^k + \dots, \end{aligned} \quad (5)$$

and we can state the following result [1].

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

It is well known that the fixed point method has first order convergence.

During the last many years, the numerical techniques for solving nonlinear equations have been successfully applied (see, e.g., [2–4] and the references therein).

In [4], Babolian and Biazar modified the standard Adomian decomposition method for solving the nonlinear equation $f(x) = 0$ to derive a sequence of approximations to the solution, with nearly superlinear convergence. However, their method requires the computation of higher-order derivatives of the nonlinear operator involved in the functional equation.

In this paper, a new iteration method extracted from the fixed point method is proposed to solve nonlinear equations. The proposed method has second-order convergence and then applied to solve some problems in order to assess its validity and accuracy. It is worth to mention that our method involves only first-order derivative but showing second-order convergence.

2. New Iteration Method

Consider the nonlinear equation

$$f(x) = 0, \quad x \in \mathbb{R}. \quad (6)$$

We assume that α is simple zero of $f(x)$, and x_0 is an initial guess sufficiently close to α . Equation (6) is usually rewritten as

$$x = g(x). \quad (7)$$

Following the approach of [4], if $g'(x) \neq 1$, we can modify (7) by adding $\theta \neq -1$ to both sides as follows:

$$x + \theta x = \theta x + g(x), \quad (1 + \theta)x = \theta x + g(x), \quad (8)$$

which implies that

$$x = \frac{\theta x + g(x)}{1 + \theta} = g_\theta(x). \quad (9)$$

In order for (9) to be efficient, we can choose θ such that $g'_\theta(x) = 0$, we yields

$$\theta = -g'(x), \quad (10)$$

so that (9) takes the form

$$x = \frac{-g'(x)x + g(x)}{1 - g'(x)}. \quad (11)$$

This formulation allows us to suggest the following iteration methods for solving nonlinear equation (6).

Algorithm 3. For a given x_0 , we calculate the approximation solution x_{n+1} , by the iteration scheme

$$x_{n+1} = \frac{-g'(x_n)x_n + g(x_n)}{1 - g'(x_n)}, \quad g'(x_n) \neq 1. \quad (12)$$

3. Convergence Analysis

Now we discuss the convergence analysis of Algorithm 3.

Theorem 4. Let $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval D and consider that the nonlinear equation $f(x) = 0$ (or $x = g(x)$) has a simple root $\alpha \in D$, where $g(x) : D \subset \mathbb{R} \rightarrow \mathbb{R}$ be sufficiently smooth in the neighborhood of the root α ; then the order of convergence of Algorithm 3 is at least 2.

Proof. The iteration scheme is given by

$$x_{n+1} = \frac{-g'(x_n)x_n + g(x_n)}{1 - g'(x_n)}, \quad g'(x_n) \neq 1. \quad (13)$$

Let α be a simple zero of f , $e_n = x_n - \alpha$, where e_n is the error term involved at the n th step of Algorithm 3 and

$$g(\alpha) = \alpha. \quad (14)$$

By Taylor's expansion, we have

$$\begin{aligned}
 \alpha - e_{n+1} &= \frac{-g'(\alpha - e_n)(\alpha - e_n) + g(\alpha - e_n)}{1 - g'(\alpha - e_n)} \\
 &= \left(- \left(g'(\alpha) - e_n g''(\alpha) + e_n^2 \frac{g'''(\alpha)}{2!} - \dots \right) \right. \\
 &\quad \times (\alpha - e_n) + g(\alpha) - e_n g'(\alpha) + e_n^2 \frac{g''(\alpha)}{2!} \\
 &\quad \left. - \dots \right) \times \left(1 - \left(g'(\alpha) - e_n g''(\alpha) \right. \right. \\
 &\quad \left. \left. + e_n^2 \frac{g'''(\alpha)}{2!} - \dots \right) \right)^{-1} \\
 &= \left(- \left(g'(\alpha) - e_n g''(\alpha) + e_n^2 \frac{g'''(\alpha)}{2!} - \dots \right) \right. \\
 &\quad \times (\alpha - e_n) + \alpha - e_n g'(\alpha) + e_n^2 \frac{g''(\alpha)}{2!} \\
 &\quad \left. - \dots \right) \times \left(1 - g'(\alpha) + e_n g''(\alpha) \right. \\
 &\quad \left. - e_n^2 \frac{g'''(\alpha)}{2!} + \dots \right)^{-1} \\
 &= \left(\alpha - \alpha g'(\alpha) + \alpha e_n g''(\alpha) \right. \\
 &\quad \left. - e_n^2 \frac{(g''(\alpha) + \alpha g'''(\alpha))}{2!} + \dots \right) \\
 &\quad \times \left(1 - g'(\alpha) + e_n g''(\alpha) - e_n^2 \frac{g'''(\alpha)}{2!} + \dots \right)^{-1} \\
 &= \alpha - \rho(\alpha, g'(\alpha), g''(\alpha), g'''(\alpha)) e_n^2 - \dots,
 \end{aligned} \tag{15}$$

this shows that

$$e_{n+1} = \rho(\alpha, g'(\alpha), g''(\alpha), g'''(\alpha)) e_n^2 + \dots \tag{16}$$

This completes the proof. $\square$

Remark 5. For

$$G(x) = \frac{-g'(x)x + g(x)}{1 - g'(x)} \tag{17}$$

TABLE 1

FPM	Algorithm 3
$x_1 = -1.9476$	$x_1 = -2.0050$
$x_2 = -1.9731$	$x_2 = -2$
$x_3 = -1.9864$	
$x_4 = -1.9932$	
$x_5 = -1.9966$	
$x_6 = -1.9983$	
$x_7 = -1.9991$	
$x_8 = -1.9995$	
$x_9 = -1.9997$	
$x_{10} = -1.9998$	
$x_{11} = -1.9999$	

TABLE 2

FPM	Algorithm 3
$x_1 = 2.1225$	$x_1 = 2.12$
$x_2 = 2.1197$	
$x_3 = 2.1201$	
$x_4 = 2.12$	

and using the software Maple, we can easily deduce that

$$G'(x) = \frac{g''(x)(-x + g(x))}{(1 - g'(x))^2}, \tag{18}$$

$$\begin{aligned}
 G''(x) &= \left((1 - g'(x)) (g'''(x)(-x + g(x)) \right. \\
 &\quad \left. - g''(x)(1 - g'(x))) \right. \\
 &\quad \left. + 2g''^2(x)(-x + g(x)) \right) \\
 &\quad \times (1 - g'(x))^{-1}.
 \end{aligned} \tag{19}$$

Now, it can be easily seen that for (14) we obtain $G(\alpha) = \alpha$, $G'(\alpha) = 0$, and $G''(\alpha) = -(1 - g'(\alpha))g''(\alpha) \neq 0$. Hence, according to Theorem 2, Algorithm 3 has second-order convergence.

4. Applications

Now we present some examples [4] to illustrate the efficiency of the developed methods namely, Algorithm 3. We compare the fixed point method (FPM) with Algorithm 3.

Example 6. Consider the equation $x^3 + 4x^2 + 8x + 8 = 0$. We have $g(x) = -(1 + (1/2)x^2 + (1/8)x^3)$ and $g'(x) = -x - (3/8)x^2$. The exact solution of this equation is -2 . Take $x_0 = -1.9$; then the comparison of the two methods is shown in Table 1 correct up to four decimal places.

Example 7. Consider the equation $x + \ln(x - 2) = 0$. We have $g(x) = 2 + e^{-x}$ and $g'(x) = -e^{-x}$. The graphical solution of this equation is 2.1200 (4D). Take $x_0 = 2.1$, then the comparison of the two methods is shown in Table 2 up to four decimal places.

TABLE 3

FPM	Algorithm 3
$x_1 = 0.30302$	$x_1 = 0.34046$
$x_2 = 0.31755$	$x_2 = 0.34592$
$x_3 = 0.32699$	$x_3 = 0.34595$
$x_4 = 0.33322$	
$x_5 = 0.33737$	
$x_6 = 0.34016$	
$x_7 = 0.34203$	
$x_8 = 0.34330$	
$x_9 = 0.34416$	
$x_{10} = 0.34474$	
$x_{11} = 0.34513$	
$x_{12} = 0.34540$	
$x_{13} = 0.34558$	
$x_{14} = 0.3457$	
$x_{15} = 0.34578$	
$x_{16} = 0.34584$	
$x_{17} = 0.34588$	
$x_{18} = 0.3459$	
$x_{19} = 0.34592$	
$x_{20} = 0.34593$	
$x_{21} = 0.34594$	

TABLE 4

FPM	Algorithm 3
$x_1 = 0.86131$	$x_1 = 0.90768$
$x_2 = 0.88812$	$x_2 = 0.91001$
$x_3 = 0.9001$	
$x_4 = 0.90551$	
$x_5 = 0.90796$	
$x_6 = 0.90908$	
$x_7 = 0.90959$	
$x_8 = 0.90982$	
$x_9 = 0.90992$	
$x_{10} = 0.90997$	
$x_{11} = 0.90999$	
$x_{12} = 0.91000$	
$x_{13} = 0.91$	

Example 8. Consider the equation $x^3 + 4x^2 + 8x + 8 = 0$. We have $g(x) = 0.2 + 1.8x^2 - 2x^3 + x^4 - 0.2x^5$ and $g'(x) = 3.6x - 6x^2 + 4x^3 - x^4$. The graphical solution of this equation is 0.34595 (5D). Take $x_0 = 0.28$, then the comparison of the two methods is shown in Table 3 correct up to five decimal places.

Example 9. Consider the equation $e^x - 3x^2 = 0$. We have $g(x) = \sqrt{e^x/3}$ and $g'(x) = (1/2\sqrt{3})e^{x/2}$. The graphical solution of this equation is 0.91 (2D). Take $x_0 = 0.8$; then the comparison of the two methods is shown in Table 4 corrected up to five decimal places.

5. Conclusions

A new iteration method for solving nonlinear equations is established. By using some examples the performance of the method is also discussed. The method is performing very well in comparison to the fixed point method and the method discussed in [4]. The method can be studied for functional equations and can be extended to a system of nonlinear equations.

Acknowledgments

The authors would like to thank the editor and referees for useful comments and suggestions. This study was supported by research funds from Dong-A University.

References

- [1] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley & Sons, New York, NY, USA, 1966.
- [2] E. Babolian and J. Biazar, "On the order of convergence of Adomian method," *Applied Mathematics and Computation*, vol. 130, no. 2-3, pp. 383–387, 2002.
- [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] E. Babolian and J. Biazar, "Solution of nonlinear equations by modified Adomian decomposition method," *Applied Mathematics and Computation*, vol. 132, no. 1, pp. 167–172, 2002.