The Third-order Iterative Method for Solving Nonlinear Equations

Chalermwut Comemuang*, Naratip Wisesram and Natthapoom Pongsawat

Department of Mathematics Science, Faculty of Science, Buriram Rajabhat University, Buriram 31000, Thailand

* Corresponding author. E-mail address: chalermwut.cm@bru.ac.th

Received: 17 August 2021; Revised: 23 June 2022; Accepted: 1 July 2022; Available online: 28 December 2022

Abstract

In this paper, we present a new iterative method for solving nonlinear equations, which were developed from the concept of Rafiq et al. The new method is based on Newton's method and using Taylor's Series to prove the convergence of the method. This iterative method requires three evaluations of the function, and only use the first derivative. Analysis of its convergence shows that the order of convergence of the new iterative method is third. Numerical comparisons are made with other methods to show the efficiency of the proposed method.

Keywords: Nonlinear equations, Newton's method, Oder of convergence, Iterative method

Introduction

One of the most important problems in numerical analysis is to solve a root-finding problem of nonlinear equations. In this work we are concerned with iterative methods to solve a nonlinear equation $f(x) = 0$ that uses f and f' but not the higher derivatives of f. The classical iterative method for solving a nonlinear equation is Newton's method (Gautschi, 2012) as below

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (1)$$

This is a classical method for finding the approximate root of $f(x) = 0$, which is a one-step iterative method and has quadratic convergence.

Schröder introduced a modified Newton's method for finding roots (Schröder, 1870), which is written as

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)} \qquad (2)$$

It is well known that if α is a root with multiplicity m, then it is also a root of $f'(x) = 0$ with multiplicity $m-1$ of $f''(x) = 0$ with multiplicity $m-2$ and so on. So, if initial approximation guess x_0 is sufficiently close to α, the expressions

$$x_1 = x_0 - m\frac{f(x_0)}{f'(x_0)}$$

$$x_1 = x_0 - (m-1)\frac{f(x_0)}{f'(x_0)} \qquad (3)$$

$$x_1 = x_0 - (m-2)\frac{f(x_0)}{f'(x_0)}$$

$$\vdots$$

will have the same value. The generalized Newton has the following approximation formula

$$x_{n+1} = x_n - 2\frac{f(x_n)}{f'(x_n)} \tag{4}$$

which is a one-step iterative method and has quadratic convergence for a simple zero of the nonlinear equation $f(x) = 0$, and has a pair of double roots in the neighborhood of x_0. It may be noted that for the double root α near to x_0, $f(\alpha) = 0 = f'(\alpha)$. Rafiq has presented and studied two-step and three-step iterative methods with second-order convergence (Kang, Ali, & Rafiq, 2016). Several other iterative methods have also been developed for finding the simple zero of nonlinear equations.

In this paper, we present a new iterative method for solving nonlinear equations with three-step iterations by, implementing the concept of Kang et al. (2016) and including a modified Newton's method (Schröder, 1870). The motivation for this work is the construction of an efficient iterative method to find the approximate roots of nonlinear equations, and most importantly to avoid calculating higher-order derivatives of the function. In the next section, we discuss the proposed iterative method that demonstrates the per iteration characteristic of requiring three evaluations of the function and one of its first derivative, but it does not require to compute the second or higher derivatives. Then, we prove that the order of convergence of the method is third-order, using Taylor's polynomial to prove convergence analysis. Finally, we give the numerical results to show the efficiency of the presented methods and to compare it with other methods.

New iterative methods

In this section, we consider the iterative methods to find a simple root of nonlinear equation. In the following equation, we can rewrite the nonlinear equation $f(x) = 0$ as a coupled system

$$f(\gamma) + (x-\gamma)\frac{f'(\gamma)+f'(x)}{2} = 0 \tag{5}$$

where γ is the initial approximation for a zero of $f(x) = 0$.

From equation (5), we can rewrite in the following form

$$\begin{aligned} x &= \gamma - 2\frac{f(\gamma)}{f'(\gamma)} - (x-\gamma)\frac{f'(x)}{f(\gamma)} \\ &= c + N(x) \end{aligned} \tag{6}$$

where

$$c = \gamma - 2\frac{f(\gamma)}{f'(\gamma)} \tag{7}$$

and

$$N(x) = -(x-\gamma)\frac{f'(x)}{f(\gamma)} \tag{8}$$

while $N\left(x\right)$ is a nonlinear operator.

Chun, Bac, and Neta (2009) introduced two new families of iterative methods for multiple roots of nonlinear equations with third-order convergent for multiple roots as in (Chun et al, 2009), equation (6) has the series form

$$x = \sum_{i=0}^{\infty} x_i \tag{9}$$

The nonlinear operator $N\left(x\right)$ can be decomposed as it has been shown to be in a new one-parameter fourth-order family of iterative methods for solving nonlinear equations (Chun & Ham, 2007). Also, the series $x = \sum_{i=0}^{\infty} x_i$ converges absolutely and uniformly to a unique solution of equation (6) if the nonlinear operator

$$N\left(x\right) = N\left(\sum_{i=0}^{\infty} x_i\right) = N\left(x_0\right) + \sum_{i=0}^{\infty}\left\{N\left(\sum_{j=0}^{i} x_j\right)\right\} \tag{10}$$

is a contraction. Also, from equations (6), (8) and (10), we obtain

$$\sum_{i=0}^{\infty} x_i = c + N\left(x_0\right) + \sum_{i=1}^{\infty}\left\{N\left(\sum_{j=0}^{i} x_j\right)\right\} \tag{11}$$

Therefore, we derive the following iterative scheme

$$\begin{aligned}
x_0 &= c \\
x_1 &= N\left(x_0\right) \\
x_2 &= N\left(x_0 + x_1\right) \\
&\vdots \\
x_{n+1} &= N\left(x_0 + x_1 + \ldots + x_n\right), \qquad n = 1, 2, \ldots
\end{aligned}$$

Then

$$x_1 + x_2 + \ldots + x_{n+1} = N\left(x_0 + x_1 + \ldots + x_n\right), \qquad n = 1, 2, \ldots$$

which implies

$$x = c + \sum_{i=1}^{\infty} x_i \tag{12}$$

From (7), (8) and (12), one can have

$$x_0 = c = \gamma - 2\frac{f\left(\gamma\right)}{f'\left(\gamma\right)} \tag{13}$$

and

$$\begin{aligned}
x_1 &= N\left(x_0\right) \\
x_1 &= -\left(x_0 - \gamma\right)\frac{f'\left(x_0\right)}{f'\left(\gamma\right)} \\
x_1 &= 2\frac{f'\left(x_0\right)}{f'\left(\gamma\right)}
\end{aligned} \tag{14}$$

Again, using (8), (13) and (14), we obtain

$$x_2 = N\left(x_0 + x_1\right)$$

$$x_2 = -\left(x_0 + x_1 - \gamma\right)\frac{f'\left(x_0 + x_1\right)}{f'\left(\gamma\right)}$$

$$x_2 = 2\frac{f\left(\gamma\right)f'\left(x_0\right)}{\left(f'\left(\gamma\right)\right)^2} \tag{15}$$

This enables us to obtain the following iterative method.

Algorithm 2.1 Starting with an initial approximation x_0, the iteration is generated by the following iterative scheme

$$y_n = x_n - \frac{2f\left(x_n\right)}{f'\left(x_n\right)}, \qquad f'\left(x_n\right) \neq 0, \qquad n = 0,1,2,\ldots \tag{16}$$

$$z_n = y_n - \frac{2f\left(y_n\right)}{f'\left(x_n\right)} \tag{17}$$

$$x_{n+1} = z_n - \frac{2f\left(x_n\right)f'\left(y_n\right)}{\left(f'\left(x_n\right)\right)^2} \tag{18}$$

Algorithm 2.1 is the new method for a root-finding nonlinear equation. The characteristic of its per iteration calculation is that it requires three evaluations of the function and one for its first derivative, apparently Algorithm 2.1 is second derivative free. The order of its convergence is third.

Convergence analysis

In this section, we discuss the analysis of convergence in Algorithm 2.1

Theorem 3.1 Assume that the function $f : I \subset \mathbb{R} \to \mathbb{R}$ has a simple root $\alpha \in I$, for an open interval I. If $f\left(x\right)$ is sufficiently smooth in the neighborhood of the root α, then the order of convergence given by Algorithm 2.1 is third.

Proof. Assume α is a solution of the equation $f\left(x\right) = 0$. Let e_n be the error at n - th iteration, then one has $e_n = x_n - \alpha$. Using Taylor polynomial for $f\left(x_n\right)$ expanded about α, we obtain

$$f\left(x_n\right) = c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5) \tag{19}$$

$$f'\left(x_n\right) = 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4O(e_n^5) \tag{20}$$

where $c_k = \dfrac{f^{(k)}\left(\alpha\right)}{k!}$, $k = 2,3,\ldots$

From equations (19) and (20), one can get

$$\frac{f(x_n)}{f'(x_n)}=\frac{1}{2}e_n-\frac{1}{4}\frac{c_3}{c_2}e_n^2+\left(-\frac{1}{2}\frac{c_4}{c_2}+\frac{3}{8}\frac{c_3^2}{c_2^2}\right)e_n^3+\left(-\frac{3}{4}\frac{c_5}{c_2}+\frac{5}{4}\frac{c_3c_4}{c_2^2}-\frac{9}{16}\frac{c_3^3}{c_2^3}\right)e_n^4+O(e_n^5) \tag{21}$$

So, from $y_n=x_n-2\frac{f(x_n)}{f'(x_n)}$, we have

$$y_n=\alpha-\frac{1}{2}\frac{c_3}{c_2}e_n^2+\left(-\frac{c_4}{c_2}+\frac{3}{4}\frac{c_3^2}{c_2^2}\right)e_n^3+\left(-\frac{3}{2}\frac{c_5}{c_2}+\frac{5}{2}\frac{c_3c_4}{c_2^2}-\frac{9}{8}\frac{c_3^3}{c_2^3}\right)e_n^4+O(e_n^5) \tag{22}$$

Using Taylor series for $f(y_n)$ expanded about α, we have

$$f(y_n)=\frac{1}{4}\frac{c_3^2}{c_2}e_n^4+\left(\frac{c_3c_4}{c_2}-\frac{3}{4}\frac{c_3^3}{c_2^2}\right)e_n^5+\left(\frac{3}{2}\frac{c_3c_5}{c_2}-\frac{4c_4c_3^2}{c_2^2}+\frac{29}{16}\frac{c_3^4}{c_2^3}+\frac{c_4^2}{c_2}\right)e_n^6+O(e_n^7) \tag{23}$$

Again, using Taylor series for $f'(y_n)$ expanded about α, we obtain

$$\begin{aligned}f'(y_n)=&\ c_3e_n^2+\left(2c_4-\frac{3}{2}\frac{c_3^2}{c_2}\right)e_n^3+\left(\frac{3c_3^3}{c_2^2}+3c_5-\frac{5c_3c_4}{c_2}\right)e_n^4\\&+\left(\frac{27}{2}\frac{c_4c_3^2}{c_2^2}-\frac{45}{8}\frac{c_3^4}{c_2^3}+4c_6-\frac{7c_3c_5}{c_2}-\frac{4c_4^2}{c_2}\right)e_n^5+O\left(e_n^6\right)\end{aligned} \tag{24}$$

Substituting equations (20), (22) and (23) in to equation (17) in Algorithm 2.1, one can get

$$z_n=\alpha+\frac{1}{2}\frac{c_3}{c_2}e_n^2+\left(\frac{c_4}{c_2}-\frac{c_3^2}{c_2^2}\right)e_n^3+\left(\frac{3}{2}\frac{c_5}{c_2}-\frac{7}{2}\frac{c_3c_4}{c_2^2}+\frac{9}{4}\frac{c_3^3}{c_2^3}\right)e_n^4+O(e_n^5) \tag{25}$$

Therefore, from equation (18) in Algorithm 2.1, equation (21), (24), (25) and $e_n=x_n-\alpha$, we can obtain

$$x_{n+1}=\alpha+\frac{3}{4}\frac{c_3^2}{c_2^2}e_n^3+O(e_n^4) \tag{26}$$

That is

$$e_{n+1}=\frac{3}{4}\frac{c_3^2}{c_2^2}e_n^3+O(e_n^4) \tag{27}$$

From Equation (27) it is shown that the order of convergence of Algorithm 2.1 is third.

Numerical example

Now, we present some numerical examples on some test equations to verify the efficiency of the third-order methods defined by Algorithm 2.1 (abbreviated as Alg. 2.1). Also, we compare their results with the methods espoused in Chun (the Chun method, CM for short) (Chun, 2007), Rostam and Fuad (RFM for short) (Rostam & Fuad, 2011), Nappassanan and Montri (MNM for short) (Nappassanan & Montri, 2017) and the methods of Chun and Kim (KCM1 and KCM2) (Chun & Kim, 2010), which are the same third -order convergence.

Starting with a given initial approximation x_0, where α is the exact root computed with 200 significant digits, all numerical computations are carried out using the Maple program to a maximum computations of 10^4. For $\varepsilon = 10^{-15}$, so that the computer programs for iterative computations are terminated, we choose the stopping criteria

(i) $|x_{n+1} - x_n| < \varepsilon$,

(ii) $|f(x_n)| < \varepsilon$.

The test equations are presented as below

$f_1(x) = (x-2)^2$,

$f_2(x) = (e^x - 4x^2)^2$,

$f_3(x) = x^3 - x^2 - x + 1$,

$f_4(x) = (\sin x - \cos x)^2$,

$f_5(x) = (\sin^2 x - x^2 + 1)^2$,

$f_6(x) = x^3 - 3x + 2$,

$f_7(x) = x^4 - 11x^3 + 36x^2 - 16x - 64$

Table 1 Numerical experiments and comparison of different iterative methods

Equations	x_0	Methods	No. iterations	x_n	$\lvert f(x_n)\rvert$	$\lvert x_n - x_{n-1}\rvert$
$f_1(x)$	1.97	KCM1	14	1.9999999952635289804	2.24×10^{-17}	9.76×10^{-9}
		KCM2	14	1.9999999953449830000	2.16×10^{-17}	9.59×10^{-9}
		RFM	≥ 10000	-	-	-
		MNM	14	1.9999999938225488731	3.81×10^{-17}	1.23×10^{-8}
		CM	16	1.9999999954120197642	2.10×10^{-17}	7.64×10^{-9}
		Alg.2.1	1	2.0000000000000000000	0	3.00×10^{-2}
$f_2(x)$	-0.39	KCM1	15	-0.407776708525538196	1.19×10^{-17}	1.81×10^{-9}
		KCM2	15	-0.407776708516401302	1.21×10^{-17}	1.83×10^{-9}
		RFM	≥ 10000	-	-	-
		MNM	15	-0.407776708175209912	2.33×10^{-17}	2.45×10^{-9}
		CM	17	-0.407776708425631817	1.47×10^{-17}	1.63×10^{-9}
		Alg.2.1	2	-0.407776709404469496	1.80×10^{-27}	1.60×10^{-5}
$f_3(x)$	1.03	KCM1	14	1.0000000048244297172	4.65×10^{-17}	9.94×10^{-9}
		KCM2	14	1.0000000049048930989	4.81×10^{-17}	1.01×10^{-8}
		RFM	16	-0.999999999999999986	5.35×10^{-17}	1.69×10^{-15}
		MNM	14	1.0000000064144057398	8.22×10^{-17}	1.28×10^{-8}
		CM	16	1.0000000046700853076	4.36×10^{-17}	7.78×10^{-9}
		Alg.2.1	2	1.0000000000000000208	8.67×10^{-6}	4.80×10^{-6}

Table 1 (Cont.)

Equations	x_0	Methods	No. iterations	x_n	$\lvert f(x_n)\rvert$	$\lvert x_n - x_{n-1}\rvert$
$f_4(x)$	0.7	KCM1	15	0.7853981590125644165	3.87×10^{-17}	9.03×10^{-9}
		KCM2	15	0.7853981592323155500	3.46×10^{-17}	8.58×10^{-9}
		RFM	≥ 10000	-	-	-
		MNM	15	0.7853981577044005283	6.48×10^{-17}	1.13×10^{-8}
		CM	17	0.7853981585206856420	4.75×10^{-17}	8.12×10^{-9}
		Alg.2.1	2	0.7853981633974483096	1.15×10^{-61}	1.01×10^{-6}
$f_5(x)$	1.45	KCM1	15	1.4044916507720333489	4.02×10^{-17}	5.23×10^{-9}
		KCM2	15	1.4044916508336378850	4.22×10^{-17}	5.39×10^{-9}
		RFM	≥ 10000	-	-	-
		MNM	15	1.4044916515800825572	6.97×10^{-17}	6.72×10^{-9}
		CM	17	1.404491651049827549,	4.95×10^{-17}	4.72×10^{-9}
		Alg.2.1	2	1.4044916482211642267	2.08×10^{-22}	1.46×10^{-4}
$f_6(x)$	1.06	KCM1	15	1.0000000031714040348	3.01×10^{-17}	6.53×10^{-9}
		KCM2	15	1.0000000032759371321	3.21×10^{-17}	6.75×10^{-9}
		RFM	32	-1.9999999999999999999	5.59×10^{-19}	1.35×10^{-16}
		MNM	15	1.0000000043497837726	5.67×10^{-17}	8.69×10^{-9}
		CM	17	1.0000000035229358334	3.72×10^{-17}	5.87×10^{-9}
		Alg.2.1	2	1.0000000000000003953	4.68×10^{-31}	1.68×10^{-5}
$f_7(x)$	4.02	KCM1	15	4.0000015299563080550	1.79×10^{-17}	1.34×10^{-6}
		KCM2	15	4.0000015367599304127	1.81×10^{-17}	1.35×10^{-6}
		RFM	36	-1.0000000000000000000	1.93×10^{-19}	1.01×10^{-12}
		MNM	13	4.000002460926661399,	7.45×10^{-17}	2.46×10^{-6}
		CM	17	4.000001601151262750,	2.05×10^{-17}	1.18×10^{-6}
		Alg.2.1	7	4.0000007847556940356	2.41×10^{-18}	2.56×10^{-6}

Table 1 displays the different test equations $f_i = 0,\ i = 1, 2, \ldots, 7$, the initial approximation x_0, the number of iterations n to approximate the solution, the approximate solution x_n, the values $\lvert x_n - x_{n-1}\rvert$ and $\lvert f(x_n)\rvert$. From these numerical results, the proposed method appears to be more robust and thus more competitive than the other methods compared and we can see that there are some equations that cannot be root-finding. As shown in Table 1 the proposed method is preferable to the other known method of the same order with third-order convergence.

Conclusion

The objective for introducing a new three-step algorithm was to develop several simple yet powerful mathematical tips for improving the efficiency of solving nonlinear equations. The new algorithm was developed from a concept of Kang et al. (2016) including a modified Newton's method, where it requires three evaluations of the function per iteration so that it does not need to compute the second or higher derivatives. We showed the numerical results to show the efficiency of the presented method and compare it to other third-order methods. So, we have proved that the order of convergence of the method suggested by Algorithm 2.1 is three. In our future work we will apply our methods for more complex examples.

References

Chun, C. (2007). A method for obtaining iterative formulas of order three. *Applied Mathematics Letters*, *20*, 1103-1109. https://doi.org/10.1016/j.aml.2006.11.010

Chun, C., Bac, H. J., & Neta, B. (2009). New families of nonlinear third-order solvers for finding multiple roots. *Computers & Mathematics with Applications*, *57*, 1574-1582. https://doi.org/10.1016/j.camwa.2008.10.070

Chun, C., & Ham, Y. (2007). A one-parameter fourth-order family of iterative method for nonlinear equations. *Applied Mathematics and Computation*, *189*, 610-614. https://doi.10.1016/j.amc.2006.11.113

Chun, C., & Kim, Y. (2010). Several New Third-Order Iterative Method for Solving Nonlinear Equations. *Acta Applicandae Mathematicae*, *109*, 1053-1063. https://doi.org/10.1007/s10440-008-9359-3

Gautschi, W. (2012). *Numerical analysis* (2nd ed.). Boston: Birkhauser,

Kang, S. M., Ali, F., & Rafiq, A. (2016). Iterative method for solving scalar equations. *Journal of Nonlinear Sciences and Applications*, *9*, 1035-1042.

Nappassanan, S., & Montri, T. (2017). A new third-order Iterative method for solving nonlinear equations. *Journal of Science and Technology, Ubon Ratchathani University*, *19*(2), 173-177.

Rostam, K. S., & Fuad, W. K. (2011). New Third-order Method for Solving Nonlinear Equations. *Journal of Applied Sciences Research*, 7(6), 916-921.

Schröder, E. (1870). Ueber unendlich viele Algorithmen zur Auflösung der Gleichungen, *Mathematische Annalen*, *2*, 317-365. https://doi.org/10.1007/BF01444024