Journal of Mathematics Research; Vol. 4, No. 3; 2012
ISSN 1916-9795 E-ISSN 1916-9809
Published by Canadian Center of Science and Education

New Two-step Iterative Methods for Solving Nonlinear Equations

Mohamed S. M. Bahgat (Corresponding author)
Department of Mathematics, Faculty of Science and Arts
Najran University, Najran 1988, Saudi Arabia
&
Department of Mathematics, Faculty of Science, Minia University, Minia 61111, Egypt
Tel: 966-50-891-7215 E-mail: msmbahgat66@hotmail.com

Received: April 5, 2012 Accepted: April 19, 2012 Online Published: May 29, 2012
doi:10.5539/jmr.v4n3p128 URL: http://dx.doi.org/10.5539/jmr.v4n3p128

Abstract

In this paper, we have suggested and analyzed a new two-step type iterative methods for solving nonlinear equations of the type. We show that this new two-step method is cubic convergence method. It is proved that this method is better than the Newton method and all results in (Soheili et al., 2008). Several examples are given to illustrate the efficiency of this new method and its comparison with other methods. This method can se considered as a significant improvement of the Newton method and its variant forms.

Keywords: Nonlinear equations, Convergence, Two-step method, Numerical result

1. Introduction

Iterative methods for finding the roots of nonlinear equations $f(x) = 0$ are common yet important problem in science and engineering. Analytical methods for solving such equations are difficult or almost non-existent. It has been shown in (Noor, 2006b) that these quadrature formulas can be used to develop some iterative methods for solving nonlinear equations, we have suggested and analyzed new iterative method by using the Newton and the Halley (Noor & Inayat Noor, 2007) methods and some newly developed method by (Noor, 2006a; Noor & Inayat Noor, 2007; Noor & Ahmad, 2006a) as predictor method and then use this new method as a corrector method. We prove that this new method has a cubic convergence. Several examples are given to illustrate the efficiency and performance of this new method and its comparison with other methods. All test problems reveals a good accuracy and fast convergence of the new method.

2. Iterative Method and Convergence

Consider the nonlinear equation of the type $f(x) = 0$. For simplicity, assume that r is a simple root zero and γ is an initial guess sufficiently close to r. Using the Taylor's series expansion of the function $f(x)$, we have

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

from which we have

$$x = \gamma - \frac{f(\gamma)}{f'(\gamma)} - \frac{(x-\gamma)^2 f''(\gamma)}{2f'(\gamma)}$$

This formulation allows us to suggest the following iterative methods for solving the nonlinear equations.

Algorithm 2.1 For a given x_0 find the approximate solution x_{n+1} by the iterative scheme

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

which is an implicit method, since x_{n+1} occurs on both sides of the equation, which it itself a difficult problem. We remark that $f''(x_n) = 0$, then algorithm 2.1 reduces to the well known Newton Method, that is.

Algorithm 2.2 For a given x_0, find the approximate solution x_{n+1} by the iterative scheme

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

It is well known that algorithm 2.2 has a quadratic convergence, see (Burden & Douglas Faires, 2001). Using this algorithm as a predictor, in (Noor, 2006a) has suggested and analyzed the following two-step iterative method.

Algorithm 2.3 For a given x_0, find the approximate solution x_{n+1} by the iterative schemes

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

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

It has been shown (Noor, 2006a) that algorithm has a cubic convergence and is called the two-step iterative method. Now using algorithm 2.3 to suggest and analyze a new two-step iterative method for solving the nonlinear equation, which is the tain motivation of this paper. We use in the following algorithms as predictor in (Nooc & Inayat Noor, 2007).

Algorithm 2.4 For a given x_0, find the approximate solution x_{n+1} by the iterative schemes

$$y_n = x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)}, \quad n = 0, 1, 2, ...$$

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

We point out that algorithm 2.4 can be written in the following equivalent form.

Algorithm 2.5 For a given x_0 find the approximate solution x_{n+1} by the iterative scheme

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{2f^2(x_n)f'(x_n)f''(x_n)}{[2(f'(x_n))^2 - f(x_n)f''(x_n)]^2}$$

Algorithm 2.6 For a given x_0 find the approximate solution x_{n+1} by the iterative scheme

$$y_n = x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)}$$

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

We now study the convergence of algorithm 2.4. In similar way, one can prove the convergence of other two-step algorithm 2.6.

Theorem 1 *Let $r \in I$ be a simple zero of sufficiently differentiable function $f : \subseteq R \to R$ for an open interval I. If x_0 is sufficiently close to r, then the two-step method defined by the algorithm 2.4 has third-order convergence.*

Proof. Consider to

$$y_n = x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)} \tag{1}$$

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

Let r be a simple zero of f. Since f is sufficiently differentiable, by expanding $f(x_n)$ and $f'(x_n)$ about r, I can get

$$f(x_n) = f(r) + (x_n - r)f'(r) + \frac{(x_n - r)^2}{2!}f^{(2)}(r) + \frac{(x_n - r)^3}{3!}f^{(3)}(r) + \frac{(x_n - r)^4}{4!}f^{(4)}(r) + ...$$

$$f(x_n) = f'(r)(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + ...) \tag{3}$$

$$f'(x_n) = f'(r)(1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + ...) \tag{4}$$

$$f''(x_n) = f'(r)(2c_2 + 6c_3e_n + 12c_4e_n^2 + 20c_5e_n^3 + 30c_6e_n^4 + ...) \tag{5}$$

where $c_k = \frac{1}{k!}\frac{f^{(k)}(r)}{f'(r)}$, $k = 1, 2, 3, ...,$ and $e_n = x_n - r$ Now from (4) and (5), we have

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2e_n^2 + 2(c_2^2 - c_3)e_n^3 + (7c_2c_3 - 4c_2^3 - 3c_4)e_n^4 + ... \tag{6}$$

Using (3), (4) and (5) in (1) we have:

$$y_n = r + (c_2^2 - c_3)e_n^3 + (-3c_2^3 + 6c_2c_3 - 3c_4)e_n^4 + (148c_3c_2^2 + 5c_2^4 + 12c_2c_4 + 6c_3^2)e_n^5 + ... \tag{7}$$

From (4), (5) and (7) I can get

$$\frac{(y_n - x_n)^2 f''(x_n)}{2f'(x_n)} = c_2e^2 - (2c_2^2 - 3c_3)e^3 + (2c_2^3 - 37c_2c_3 + 6c_4)e^4 + ... \tag{8}$$

Using (6) and (8) in (2), we have:

$$x_{n+1} = x_n-(e_n-c_2e_n^2+2(c_2^2-c_3)e_n^3+(7c_2c_3-4c_2^3-3c_4)e_n^4+...)-(c_2e_n^2-(2c_2^2-3c_3)e_n^3+(37c_2c_3+2c_2^3+6c_4)e_n^4+...) \tag{9}$$

From (9), $e_{n+1} = x_{n+1} - r$ and $e_n = x_n - r$ we have

$$e_{n+1} = e_n-(e_n-c_2e_n^2+2(c_2^2-c_3)e_n^3+(7c_2c_3-4c_2^3-3c_4)e_n^4+...)-(c_2e_n^2-(2c_2^2-3c_3)e_n^3+(-37c_2c_3+2c_2^3+6c_4)e_n^4+...)$$

$$e_{n+1} = -c_3e_n^3 + O(e^4)$$

Which shows that the algorithm 2.4 has third-order convergence. Since asymptotic convergence of Newton method is c_2 and from Theorem 1, We deduced that the convergence rate of algorithm 2.4 is better than the Newton method and the method of Soheili (SM) (Soheili et al., 2008). We know that the cubic convergent method is vastly superior to the linear and the quadratically convergent methods.

3. Numerical Results

We present some examples to illustrate the efficiency of the new developed two-step methods, compare the Newton method (NM), the method of Soheili (SM), and the methods (BM), introduced in this present paper (algorithms 2.4 and 2.6). We take $\varepsilon = 10^{-15}$ as tolerance. The following criteria is used for estimating the zero:

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

(ii) $|f(x_{n+1})| < \varepsilon$.

Table 1 presents iteration number comparison of algorithms 2.4, and 2.6 with (NM) and (SM), in given precision. In Table 2, the CPU time (per second) of the new algorithms, (NM) and (SM) are compared. The numerical computations listed in tables are performed using Fortran programs with double precision.

4. Conclusions

The present two-step methods is generalized and applied for solving the nonlinear algebraic equations. The numerical results in the tables 1,2 show that the new method is very effective and provide highly accurate results in a less number of iterations as compared with Newton method (NM) and (SM) .

References

Burden, R. L., & Douglas Faires, J. (2001). *Numerical analysis.* Boston: PWS publishing company.

Noor, M. A. (2006a). Nunerical Analysis and optimization. Lecture Notes, Mathematics Department, COMSATS, Institute of information Technology, Islamabad, Pakistan.

Noor, M. A. (2006b). New iterative methods for nonlinear equations. *Journal of applied mathematical and computation*.

Noor, M. A., & Ahmad, F. (2006). On a predictor-corrector method for solving nonlinear equations. *Journal of applied mathematical and computation, 183*, 128-133.

Noor, M. A., & Inayat Noor, K. (2007). Predictor-corrector Halley method for nonlinear equations. *Journal of applied mathematical and computation, 188,* 1587-1591. http://dx.doi.org/10.1016/j.amc.2006.11.023

Soheili, A. R., Ahmadian, S. A., & Naghipoor, J. (2008). A Family of Predictor-Corrector Methods Based on Weight Combination of Quadratures for Solving Nonlinear equations. *International journal of nonlinear science, 6*, 29-33.

Table 1. Examples and comparison between other methods

Equation	x_0	Other methods		Present method (BM)		x_n
		(NM)	(SM)	Algorithm 2.4	Algorithm 2.6	
$e^{x^2+7x-30} - 1 = 0$	4	20	13	6	8	3.000000000000000
$x^3 - 10 = 0$	1.5	7	5	4	4	2.154434690031884
$x^2 - e^x - 3x + 2 = 0$	2	6	4	3	4	0.257530285439861
$sin^2(x) - x^2 + 1 = 0$	-1	7	5	3	4	-1.404491648215341
$x^{10} - 1 = 0$	1.5	10	7	4	5	1.000000000000000
$11x^{11} - 1 = 0$	0.7	8	6	4	5	0.804133097503664
$sin(1/x) - x = 0$	2	6	4	4	4	0.897539461280487

Table 2. The CPU time (per second) of aliorgthms

Equation	Other methods		Present method (BM)	
	(NM)	(SM)	Algorithm 2.4	Algorithm 2.6
$e^{x^2+7x-30} - 1 = 0$	0.171875	0.109375	0.052594	0.062500
$x^3 - 10 = 0$	0.078125	0.031250	0.025000	0.025000
$x^2 - e^x - 3x + 2 = 0$	0.046875	0.031250	0.023438	0.025000
$sin^2(x) - x^2 + 1 = 0$	0.046875	0.046875	0.023438	0.025000
$x^{10} - 1 = 0$	0.078125	0.046875	0.025000	0.039062
$11x^{11} - 1 = 0$	0.062500	0.031250	0.025000	0.039062
$sin(1/x) - x = 0$	0.046875	0.031250	0.025000	0.025000