

**FIFTH AND SIXTH-ORDER ITERATIVE ALGORITHMS
WITHOUT DERIVATIVES FOR SOLVING
NON-LINEAR EQUATIONS**

G. Fernández-Torres^{1 §}, F.R. Castillo-Soria², I. Algreto-Badillo³

¹Petroleum Engineering Department
Universidad del Istmo
Tehuantepec, Oaxaca, 70760, MÉXICO

²Computer Engineering Department
Universidad del Istmo
Tehuantepec, Oaxaca, 70760, MÉXICO

Abstract: In this paper we will present two derivative free iterative methods for finding the root of a nonlinear equation $f(x) = 0$. The new methods are based on direct and inverse polynomial interpolation. We will prove that one of the methods has fifth-order convergence and the other method sixth-order convergence. Several examples will show that convergence and the efficiency of the new methods to be better than the classic Newton's method and others derivative free methods with high order of convergence previously presented.

AMS Subject Classification: 65D05, 65D07, 65D15

Key Words: nonlinear equations, iterative methods, roots finding method, derivative free, interpolation

1. Introduction

There are several different methods in the literature to compute the root r of a

Received: November 15, 2012

© 2013 Academic Publications, Ltd.
url: www.acadpubl.eu

[§]Correspondence author

nonlinear equation $f(x) = 0$. The most famous of these methods is the classical Newton's method

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

starting from some initial value x_0 with convergence order 2. If the derivative $f'(x_n)$ is replaced by the forward difference approximation

$$f'(x_n) \approx \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)} \quad (2)$$

we obtain the classical Steffensen's method

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n + f(x_n)) - f(x_n)}. \quad (3)$$

Both the above methods have an order of convergence 2 and require two functional evaluations per iteration. However Steffensen's method is free from any derivative of the function. This is important because sometimes, in science and engineering, the applications of the iterative methods which depend upon derivatives are restricted. In recent years several researchers have developed iterative methods for solving nonlinear equations. These methods have been developed using error analysis techniques, quadrature rules and other techniques, see [2, 6, 8, 13]. Most recently, a derivative free Steffensen-secant method was proposed by Jain [4]. It uses three functional evaluations per step and has cubic convergence. In [14] a family of Steffensen like methods was derived by Zheng by applying forward difference approximation to the Weerakoon and Fernando method [13], which has cubic convergence and uses four functional evaluations per iteration. In a similar way Cordero [3] used central difference approximation

$$f'(x_n) \approx \frac{f(x_n + f(x_n)) - f(x_n - f(x_n))}{2f(x_n)} \quad (4)$$

in the Ostrowski's fourth-order method [9], obtaining a new method. This new method, free from derivatives, has fourth order convergence and requires four functional evaluations per iteration.

2. Develop of the Methods

Consider a function sufficiently differentiable $f(x)$ with $f(x) \neq 0$ on an open interval I and $r \in I$ a simple zero of $f(x)$. Since r and $f(x) \neq 0$ is a simple

zero we can consider $g = f^{-1}$ on I . Suppose that the process starting with $x_0 \in I$ close to r and x_n has been chosen. Define

$$t_1 = x_n + f(x_n), t_2 = x_n - f(x_n), \tag{5}$$

$$L_0 = f(x_n), L_1 = f(t_1), L_2 = f(t_2). \tag{6}$$

Using the notation of divided differences, consider the following interpolating polynomial for $f(x)$

$$p(x) = A(x - x_n)(x - t_1) + B(x - x_n) + C \tag{7}$$

where

$$A = f[x_n, t_1, t_2], B = f[x_n, t_1], C = f[x_n], \tag{8}$$

and verify $p(x_n) = L_0, p(t_1) = L_1, p(t_2) = L_2$. Solving the polynomial

$$p_1(x) = [-AL_0 + B](x - x_n) + C = 0 \tag{9}$$

we obtain the following two-step derivative free method (FCAM1) as follows

$$y_n = x_n - \frac{f[x_n]}{f[x_n, t_1] - f[x_n, t_1, t_2]f(x_n)}, L_3 = f(y_n) \tag{10}$$

$$x_{n+1} = x_n + \sum_{i=1}^3 (-1)^i g[L_0, \dots, L_i]L_0 \cdots L_{i-1}. \tag{11}$$

Now, consider the following interpolating polynomial for $g(y)$

$$s(y) = A(y - L_0)(y - L_1) + B(y - L_0) + C \tag{12}$$

where

$$A = g[L_0, L_1, L_2], B = g[L_0, L_1], C = g[L_0], \tag{13}$$

with $s(L_0) = g(L_0) = x_n, s(L_1) = g(L_1) = t_1, s(L_2) = g(L_2) = t_2$. Defining $y_n = s(0)$ we obtain the following new two-step derivative free method (FCAM2) as follows

$$y_n = x_n + \sum_{i=1}^2 (-1)^i g[L_0, \dots, L_i]L_0 \cdots L_{i-1}, L_3 = f(y_n), \tag{14}$$

$$x_{n+1} = y_n - g[L_0, \dots, L_3]L_0 \cdots L_2. \tag{15}$$

Observe that in (11) and (15) we have used inverse interpolation on the points L_i , for $1 \leq i \leq 3$, with the polynomial

$$q(y) = g[L_0] + \sum_{i=1}^3 g[L_0, \dots, L_i](y - L_0) \cdots (y - L_{i-1}). \quad (16)$$

The computational efficiency of an iterative method of order p , requiring N function evaluations per iteration, is most frequently calculated by Ostrowski-Traub's efficiency index [9, 12]

$$E = p^{\frac{1}{N}}. \quad (17)$$

It can be seen that the methods (FCAM1), (FCAM2) require four function evaluations per iteration with efficiency indexes $(5)^{\frac{1}{4}} \sim 1.495$ and $(6)^{\frac{1}{4}} \sim 1.565$, respectively. These efficiency indexes are clearly superior to the efficiency indexes of Steffensen's and Newton's methods which are equal to 1.414.

3. Order of Convergence

Theorem 1. *Let $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ be a sufficiently differentiable function and let $r \in I$ be a simple zero of $f(x) = 0$ in an open interval I , with $f(x) \neq 0$ on I . Define $g = f^{-1}$ on I . If $x_0 \in I$ is sufficiently close to r , then the method free from derivatives (FCAM1) defined by (10) and (11) has order of convergence 5.*

Proof. First, from (7) and (9) we have $p(x) = A(x - x_n)^2 + p_1(x)$ then, the error for the polynomial $p_1(x)$ can be written as

$$f(x) - p_1(x) = \frac{f'(\xi_1)}{2}(x - x_n)^2 + \frac{f'(\xi_2)}{6}(x - x_n)(x - t_1)(x - t_2), \quad (18)$$

for some $\xi_1, \xi_2 \in I$. Substituting t_1, t_2 , evaluating in $x = r$ and taking $\epsilon_n = x_n - r$ we have

$$p_1(r) = -\frac{f'(\xi_1)}{2}\epsilon_n^2 + \frac{f'(\xi_2)}{6}\epsilon_n(\epsilon_n + f(x_n))(\epsilon_n - f(x_n)). \quad (19)$$

Note that if we expand $f(x_n)$ with Taylor's polynomial of first degree around r , the second term in the right side can be written, for some $\xi_3 \in I$, as

$$\frac{f'(\xi_2)}{6}\epsilon_n(\epsilon_n^2 - f^2(x_n)) = \frac{f'(\xi_2)}{6}\epsilon_n^3(1 - (f'(\xi_3))^2) = O(\epsilon_n^3). \quad (20)$$

Now, since $p_1 = -AL_0 + B = -\frac{f''(\xi_1)}{2}f(\xi_3)\epsilon_n + f(\xi_4)$, for some $\xi_4 \in I$, expanding $p_1(r)$ around y_n with $\epsilon_{n+1} = r - y_n$, we have, from (19), the following asymptotic equation

$$y_n - r \sim -\frac{f(\xi_1)}{2f(\xi_4)}\epsilon_n^2 + O(\epsilon_n^3). \tag{21}$$

In the same way, the error for the polynomial $q(y)$ given by (16) is

$$g(y) - q(y) = \frac{g^{(4)}(\eta)}{24}(y - L_0)(y - L_1)(y - L_2)(y - L_3), \tag{22}$$

for some $\eta \in f(I)$. Expanding, $f(x_n + f(x_n))$, $f(x_n - f(x_n))$ and $f(y_n)$ with Taylor's polynomial of first degree around r , we have, for some $\xi_5, \xi_6, \xi_7 \in I$,

$$f(x_n + f(x_n)) = f(\xi_5)(x_n - r + (x_n - r)f(\xi_3)), \tag{23}$$

$$f(x_n - f(x_n)) = f(\xi_6)(x_n - r - (x_n - r)f(\xi_3)), \tag{24}$$

$$f(y_n) = (y_n - r)f(\xi_7). \tag{25}$$

Thus, substituting $y = 0$ in (22) with $\epsilon_{n+1} = x_{n+1} - r$, and using (21) we have the following asymptotic equation

$$\epsilon_{n+1} \sim -\frac{g^{(4)}(\eta)}{24}\epsilon_n^5 f(\xi_3)f(\xi_5)f(\xi_6)f(\xi_7)(1 - (f(\xi_3))^2)\frac{f(\xi_1)}{2f(\xi_4)} + O(\epsilon_n^6), \tag{26}$$

and finally

$$\epsilon_{n+1} \sim k\epsilon_n^5 \tag{27}$$

with k as a suitable constant. □

Theorem 2. *Let $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ be a sufficiently differentiable function and let $r \in I$ be a simple zero of $f(x) = 0$ in an open interval I , with $f(x) \neq 0$ on I . Define $g = f^{-1}$ on I . If $x_0 \in I$ is sufficiently close to r , then the method free from derivatives (FCAM2) defined by (14) and (15) has an order of convergence of 6.*

Proof. First, from (12) the error for the polynomial $s(y)$ can be written as

$$g(y) - s(y) = \frac{g(\eta_1)}{6}(y - L_0)(y - L_1)(y - L_2). \tag{28}$$

Evaluating in $y = 0$, and expanding L_0, L_1, L_2 as in (23), (24) and (25) we obtain

$$y_n - r = \frac{g(\eta_1)}{6}\epsilon_n^3 f(\xi_3)f(\xi_5)f(\xi_6)(1 - (f(\xi_3))^2). \tag{29}$$

Now, for the polynomial in (16) the error can be written as

$$g(y) - q(y) = \frac{g(\eta_2)}{6}(y - L_0)(y - L_1)(y - L_2)(y - L_3). \quad (30)$$

Evaluating in $y = 0$, and expanding L_0, L_1, L_2, L_3 as before we obtain

$$\epsilon_{n+1} = -\frac{g^{IV}(\eta_1)}{6}\epsilon_n^6 f^2(\xi_3)f^2(\xi_5)f^2(\xi_6)(1 - (f(\xi_3))^2)^2. \quad (31)$$

and finally

$$\epsilon_{n+1} \sim k\epsilon_n^6 \quad (32)$$

with k as a suitable constant. \square

4. Numerical Examples

In this section we use numerical examples to compare the methods introduced in this paper. (FCAM1) and (FCAM2) are compared with methods that do not use derivatives such as Steffensens (SM), Cordero's (CM), Jain's (JM) [4] and Zheng's methods (ZM) [14]. We also compare methods that use derivatives such as Newton's (NM), Chun's (CHM) [2], Kou's (KM) [5] and Noor's methods (NOM) [7]. All computations were carried out with double arithmetic precision on MATLAB 10. Table 1 and Table 2 show the number of iterations and the number of functional evaluations required such that $|f(x_n)| < 10^{-17}$. Numerical results show that the new derivative free methods (FCAM1) and (FCAM2) can compete with Newton's method and Steffensens method and converge faster than (JM) and (ZM), in many cases. We use the following functions, some of which are the same as in the derivative free methods [2, 3, 4, 5, 6, 7, 8, 13, 14] respectively.

Modern applications have high computational requirements in a many variety of fields, including financial analysis, data mining, medical imaging, and scientific computation. In many applications, implementations on FPGA report better results and computation times than software implementation methods, such as Newton-Raphson, Non-Restoring Square Root [10], Bisection, Lanczos [11] and secant Volterra methods [1], among others. Future work will be needed to implement the proposed methods on high-performance FPGA-based systems for obtaining implementation results.

$$\begin{aligned}
 f_1(x) &= \cos(x) - x, & r &= 0.7390851332151606. \\
 f_2(x) &= (x - 1)^3 - 2, & r &= 2.2599210498948731. \\
 f_3(x) &= (x - 1)^2 - 1, & r &= 2. \\
 f_4(x) &= x^3 + 4x^2 - 10, & r &= 1.3652300134140968. \\
 f_5(x) &= \sin(x) - \frac{1}{2}x, & r &= 1.8954942670339809. \\
 f_6(x) &= \sin^2(x) - x^2 + 1, & r &= 1.4044916482153412. \\
 f_7(x) &= x^2 - e^x - 3^x + 2, & r &= 0.2575302854398607. \\
 f_8(x) &= xe^{x^2} - \sin^2(x) + 3\cos(x) + 5, & r &= 1.2076478271309189.
 \end{aligned}$$

$f(x)$	x_0	Iterations - NFE					
		SM	JM	ZM	CM	FCAM1	FCAM2
$f_1(x)$	0.5	4 - 8	3 - 9	3 - 12	2 - 8	2 - 8	2 - 8
$f_2(x)$	1.85	6 - 12	6 - 18	6 - 24	4 - 16	3 - 12	3 - 12
$f_3(x)$	3.5	6 - 12	4 - 12	4 - 16	3 - 12	3 - 12	3 - 12
$f_4(x)$	1	5 - 10	4 - 16	4 - 12	3 - 12	3 - 12	2 - 8
$f_5(x)$	2	4 - 8	3 - 12	3 - 9	2 - 8	2 - 8	2 - 8
$f_6(x)$	1.5	4 - 8	3 - 12	3 - 9	2 - 8	2 - 8	2 - 8
$f_7(x)$	3	6 - 12	4 - 16	4 - 12	3 - 12	3 - 12	3 - 12
$f_8(x)$	-3	14 - 28	10 - 40	10 - 30	8 - 32	7 - 28	6 - 24

Table 1: Comparison with derivative free methods

5. Conclusions

The problem of locating roots of nonlinear equations (or zeros of functions) occurs frequently in scientific work. In this paper, we have introduced two new techniques for solving nonlinear equations. The introduced methods can be used for solving nonlinear equations without computing derivatives. We have proven that one method has fifth-order convergence and the other has sixth-order convergence. The efficiency indexes of the methods are 1.495 and 1.565 greater than the efficiency index of Newton's and Steffensen's methods. The numerical examples show the greater efficiency of the new methods, especially (FCAM2).

$f(x)$	x_0	Iterations - NFE					
		NM	CHM	KM	NOM	FCAM1	FCAM2
$f_2(x)$	3	7 - 14	4 - 12	6 - 24	4 - 16	3 - 12	3 - 12
$f_4(x)$	1	6 - 12	4 - 12	4 - 12	3 - 12	3 - 12	2 - 8
$f_6(x)$	1	7 - 14	4 - 12	5 - 15	3 - 12	4 - 16	3 - 12
	2	6 - 12	4 - 12	4 - 12	3 - 12	3 - 12	3 - 12
$f_7(x)$	0	5 - 10	4 - 12	3 - 9	3 - 12	2 - 8	2 - 8
	1	5 - 10	4 - 12	3 - 9	3 - 12	3 - 12	2 - 8
$f_8(x)$	-2	9 - 18	5 - 15	6 - 18	4 - 16	5 - 20	4 - 16

Table 2: Comparison with methods that use derivatives

Acknowledgments

The authors wish to acknowledge the valuable participation of Professors Ted Fondulas and Alex Yealland in the proof-reading of this paper.

The third author has been financially supported by PROMEP (UNISTMO-NPTC-056 Ignacio Algreto Badillo).

References

- [1] E. Aschbacher et. al., Prototype implementation of two efficient low-complexity digital predistortion algorithms, *EURASIP Journal on Advances in Signal Processing*, Vol. 2008 (2008).
- [2] C. Chun, A family of composite fourth-order iterative methods for solving nonlinear equations, *Applied Mathematics and Computation*, **187** (2007), 951-956.
- [3] A. Cordero et. al., Steffensen type methods for solving nonlinear equations, *Journal of Computational and Applied Mathematics*, **236** (2012), 3058-3064.
- [4] P. Jain, Steffensen type methods for solving nonlinear equations, *Applied Mathematics and Computation*, **194** (2007), 527-533.

- [5] J. Kou, Y. Li and X. Wang, A composite fourth-order iterative method for solving non-linear equations, *Applied Mathematics and Computation*, **184** (2007), 471-475.
- [6] J. Kou, Y. Li and X. Wang, A modification of Newton method with third-order convergence, *Applied Mathematics and Computation*, **181** (2006), 1106-1111.
- [7] M. A. Noor, W. A. Khan and A. Hussain, A new modified Halley method without second derivatives for nonlinear equation, *Applied Mathematics and Computation*, **189** (2007), 1268-1273.
- [8] K. I. Noor, M. A. Noor, Predictor-corrector Halley method for nonlinear equations, *Applied Mathematics and Computation*, **188** (2007), 1587-1591.
- [9] A. M. Ostrowski, *Solution of equations and systems of equations*, Academic Press, New York (1960).
- [10] K. Piromsopa, C. Aporntewan, P. Chongsatitvatana, An FPGA implementation of a fixed-point square root operation, *2001 International Symposium on Communications and Information Technology (ISCIT'2001)*, Thailand (2001).
- [11] A. Rafique, N. Kapre, and G. A. Constantinides, A high throughput FPGA-based implementation of the Lanczos method for the symmetric extremal eigenvalue problem, *ARC* (2012), 239-250.
- [12] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York (1964).
- [13] S. Weerakoon, T.G.I. Fernando, A variant of Newton's method with accelerated third-order convergence, *Applied Mathematics Letters*, **13** (2000), 87-93.
- [14] S. Zheng et. al., A Steffensen-like method and its higher-order variants, *Applied Mathematics and Computation*, **214** (2009), 10-16.

