

Root Finding With Some Engineering Applications

Tekle Gemechu

Department of mathematics, Adama Science and Technology University, Adama, Ethiopia

Abstract

In this article, derivative estimations up to the third order (in root finding, some new initiatives) are applied in Taylor’s approximation of a nonlinear function / equation to achieve efficient iterative methods. Competent methods of higher orders for solving simple roots of nonlinear equations, which improve convergence of some basic existing methods, are investigated. We shall offer several examples for test of efficiency and convergence analyses, in C++. We also provide some examples of engineering applications of root finding. Graphical demonstrations are using matlab basic tools.

Keywords: *Derivative estimations, applications, Taylor’s approximation, iterative methods, simple root*

1. Introduction

Nonlinear equations emerge in most science and engineering models. For example, in circuit analysis, analysis of state equations for a real gas, mechanical motions / oscillations, weather forecasting, in optimization and many other fields of engineering. Nonlinear problems are difficult to solve but they occur naturally.

Some existing iterative root finding methods such as the secant method needs more than one initial guesses. And others may contain higher derivatives [1, 2, 3, 10]. In this research, we substitute the higher derivative f''' in terms of f, f' and f'' to minimize the number of function evaluations which can also help us develop new one-point fixed point iterative algorithms / models from the third order Taylor’s approximation / interpolation of a nonlinear model / equation $f(x)$. These derivative estimations are not the actual function evaluation but a kind of replacement (to reduce algorithm complexity), which we are to propose only in root finding. This is our main success.

This manuscript consists of 1) introduction and starting reviews 2) construction methods 3) Analysis 4) test equations and results 5) applications 6) conclusion and recommendation 7) references.

1.2. Some starting / Review points

Let us begin with the Taylor’s approximation of $f(x)$ about an approximate root $r = x_o + h$, with x_o an initial guess for a root r of $f(x)$ and $\|h\|$ is too small.

$$f(x_o + h) = f(x_o) + hf'(x_o) + 1/2h^2 f''(x_o) + 1/6h^3 f'''(x_o) + \dots \quad (1)$$

The linear approximation $f(x_o) + hf'(x_o) \approx 0$ gives the Newton’s method [2, 3, 4]

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} : k = 0, 1, \dots, n \quad (2)$$

If we consider quadratic interpolation (approximation) of $f(x)$ as

$$f(x_o + h) = f(x_o) + hf'(x_o) + 1/2h^2 f''(x_o) \approx 0$$

, then we obtain Halley’s method (3), Chebyshev’s method (4), the extended Newton’s method (5) and Euler’s method (6) below, [2, 3, 5, 6, 14].

$$x_{k+1} = x_k - \frac{f(x_k)f'(x_k)}{[f'(x_k)]^2 - 1/2f(x_k)f''(x_k)} \quad (3)$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - 1/2 \frac{[f(x_k)]^2 f''(x_k)}{[f'(x_k)]^3} \quad (4)$$

$$\phi(x) = x - \left[\frac{f'(x)}{f''(x)} \pm \frac{\sqrt{[f'(x)]^2 - 2f(x)f''(x)}}{f''(x)} \right]. \quad (5)$$

$$\varphi(x) = x - \frac{2G(x)}{1 + \sqrt{1 - 2p(x)}}, \quad [7] \quad (6)$$

$$p(x) = \frac{G(x)f''(x)}{f'(x)}, G(x) = \frac{f(x)}{f'(x)}$$

In addition, Halley's method (3) and Chebyshev's method (4) can be obtained applying Newton's method for solving simple roots of $u(x) = \frac{f(x)}{[f'(x)]^{1/2}}$ [3]. And one can also obtain (7) if $u(x) = \frac{f(x)}{[f'(x)]}$ and then Newton's method is applied [4].

$$\phi(x_k) = x_k - \frac{f(x_k)f'(x_k)}{[f'(x_k)]^2 - f(x_k)f''(x_k)} \quad (7)$$

Note that (2) & (7) are second order and (3), (4), (5) & (6) are cubic order convergent. The following, our finding, is based on some ideas used in the above discussions.

2. New Methods by Derivative Estimations

In this section, we apply derivative estimation (substitution) technique (a very new idea) in the third order Taylor's (interpolation) approximation to construct new one-point iterative models.

We shall apply a technique of replacement of a higher derivative (here $f''' = F(f, f', f'')$) by lower derivatives (in terms of f, f' and f''). This will reduce the cost of function evaluations and may increase efficiency index ($e = p^{1/f}$). The higher derivative estimations we are proposing is $f''' = F(f, f', f'')$.

Some of which we shall prove in our future works. Now to derive our new algorithms, let

$$f(x_0 + h) = f(x_0) + hf'(x_0) + 1/2h^2 f''(x_0) + 1/6h^3 f'''(x_0) \approx 0. \quad (8)$$

$$(8) \Rightarrow hf' + 1/6h^3 f''' = h(f' + 1/6h^2 f''') \quad (9)$$

$$= -(f + 1/2h^2 f'')$$

Using $h^2 = \left(-\frac{f}{f'}\right)^2$ in (9), we obtain an iterative

algorithm (10)

$$x_{n+1} = \varphi(x_n) = x_n - 3 \frac{2f(x_n)[f'(x_n)]^2 + [f(x_n)]^2 f''(x_n)}{6[f'(x_n)]^3 + [f(x_n)]^2 f'''(x_n)} \quad (10)$$

Inserting $f''' = \frac{(f'')^2}{f'}$, $f''' = \frac{f' f''}{f}$ and $f''' = -f''$

into (10), algorithms (11), (13) and (15) below can be obtained.

$$x_{n+1} = \xi(x_n) = x_n - 3 \frac{2f(x_n)[f'(x_n)]^2 + [f(x_n)]^2 f''(x_n)}{6[f'(x_n)]^3 + \frac{[f(x_n)]^2 f''(x_n)}{f'(x_n)}} \quad (11)$$

$$= x_n - \frac{2f(x_n)f'(x_n)}{(2[f'(x_n)]^2 - f(x_n)f''(x_n))(1 + 1/6 \frac{[f(x_n)]^2 f''(x_n)}{[f'(x_n)]^4})} \quad (12)$$

$$x_{n+1} = \theta(x_n) = x_n - 3 \frac{2f(x_n)[f'(x_n)]^2 + [f(x_n)]^2 f''(x_n)}{6[f'(x_n)]^3 + f(x_n)f'(x_n)f''(x_n)} \quad (13)$$

$$= x_n - \frac{2f(x_n)f'(x_n)}{(2[f'(x_n)]^2 - f(x_n)f''(x_n))(1 + 1/6 \frac{f(x_n)f''(x_n)}{[f'(x_n)]^2})} \quad (14)$$

$$\phi(x_n) = x_n - 3 \frac{2f(x_n)[f'(x_n)]^2 + [f(x_n)]^2 f''(x_n)}{6[f'(x_n)]^3 - [f(x_n)]^2 f''(x_n)} \quad (15)$$

$$= x_n - \frac{2f(x_n)f'(x_n)}{(2[f'(x_n)]^2 - f(x_n)f''(x_n))(1 - 1/6 \frac{[f(x_n)]^2 f''(x_n)}{[f'(x_n)]^3})} \quad (16)$$

Note that the estimations used in (12), (14) and (16) above are based on series approximation concepts.

Now let

$$f(x_0) + hf'(x_0) + 1/2h^2 f''(x_0) + 1/6h^3 f'''(x_0) \approx 0. \quad (17)$$

(17) gives

$$h = \frac{-f}{f' + h/2 f'' + 1/6 h^2 f'''} \quad (18)$$

If $h = -\frac{f}{f'}$ in the right part of (18), then one will get

$$h = \frac{-6f(f')^2}{6(f')^3 - 3ff'f'' + f^2f'''} \quad (19)$$

From which

$$\phi(x_{k+1}) = x_k - \frac{6f(x_k)[f'(x_k)]^2}{6[f'(x_k)]^3 - 3f(x_k)f'(x_k)f''(x_k) + [f(x_k)]^2f'''(x_k)} \quad (20)$$

Note also that (20) is an extension of (3).

If we use $f''' = \frac{(f'')^2}{f'}$ and $f'' = f''$ in (20), then we

obtain algorithms (21) and (22) respectively.

$$\phi(x_{k+1}) = x_k - \frac{6f(x_k)[f'(x_k)]^2}{6[f'(x_k)]^3 - 3f(x_k)f'(x_k)f''(x_k) + [f(x_k)]^2 \frac{[f''(x_k)]^2}{f'(x_k)}} \quad (21)$$

$$\phi(x_{k+1}) = x_k - \frac{6f(x_k)[f'(x_k)]^2}{6[f'(x_k)]^3 - 3f(x_k)f'(x_k)f''(x_k) + [f(x_k)]^2 f'''(x_k)} \quad (22)$$

Suppose also

$$f(x_0) + hf'(x_0) + 1/2h^2 f''(x_0) + 1/6h^3 f'''(x_0) \approx 0.$$

$$\Rightarrow hf'(x_0) = -(f(x_0) + 1/2h^2 f''(x_0) + 1/6h^3 f'''(x_0)) \quad (23)$$

Using $h = -\frac{f}{f'}$ in the right part of (23) gives

$$h = -\frac{1}{f'} [f + 1/2 \left(\frac{f}{f'}\right)^2 f'' - 1/6 \left(\frac{f}{f'}\right)^3 f''']. \quad (24)$$

From which

$$\phi(x_{k+1}) = x_k - \frac{f(x_k)}{f'(x_k)} \left[1 + 1/2 \frac{f(x_k)f''(x_k)}{[f'(x_k)]^2} - 1/6 \frac{[f(x_k)]^2 f'''(x_k)}{[f'(x_k)]^3} \right] \quad (25)$$

One can note that (25) is the obvious extension of (4).

In (25), if we use $f''' = \frac{f'f''}{f}$, $f'' = \frac{(f'')^2}{f'}$,

then we obtain the algorithms below respectively.

$$\phi(x_{k+1}) = x_k - \frac{f(x_k)}{f'(x_k)} \left[1 + 1/3 \frac{f(x_k)f''(x_k)}{[f'(x_k)]^2} \right], \quad (26)$$

$$\omega(x_{k+1}) = x_k - \frac{f(x_k)}{f'(x_k)} \left[1 + 1/2 \frac{f(x_k)f''(x_k)}{[f'(x_k)]^2} - 1/6 \frac{[f(x_k)]^2 \frac{[f''(x_k)]^2}{f'(x_k)}}{[f'(x_k)]^3} \right]. \quad (27)$$

3. Convergence Analysis

We shall use the following important definition and theorem.

Definition 3.1 A sequence (x_n) generated by an iterative method is said to converge to a root r with order $p \geq 1$ if there exists $c > 0$ such that $e_{n+1} \leq ce_n^p$, $\forall n \geq n_0$, for some integer $n_0 \geq 0$ and $e_n = |r - x_n|$ [1, 2,3].

Theorem 3.1 (Order of Convergence) Assume that $\phi(x)$ has sufficiently many derivatives at a root r of $f(x)$. The order of any one-point iteration function $\phi(x)$ is a positive integer p , more especially $\phi(x)$ has order p if and only if $\phi(r) = r$ and $\phi^{(j)}(r) = 0$ for $0 < j < p$, $\phi^{(p)}(r) \neq 0$ [2, 5].

All the algorithms we presented need an appropriate choice of only one suitable initial guess x_0 in an interval $I_0 = [a, b]$ [8]. Random choice of x_0 leads to redundant works, we do not do it.

1. Proof order of convergence of algorithm in (12)

$$\phi(x_n) = x_n - \frac{2f(x_n)f'(x_n)}{(2[f'(x_n)]^2 - f(x_n)f''(x_n))(1 + 1/6 \frac{[f(x_n)f''(x_n)]^2}{[f'(x_n)]^4})} \quad (28)$$

We can write (28) as $\phi(x) = x - (x - \phi(x))H$. With

$$H = x_n - \frac{1}{(1+1/6) \frac{[f(x_n)f''(x_n)]^2}{[f'(x_n)]^4}}$$

And $x = \phi(x)$ is Halley's iteration function of order 3.

Let r be a simple root of $f(x) = 0$. We have $\phi(r) = r$,

$\phi(x) = x$ and $\phi'(x) = 0$.

And $\phi'(r) = \phi''(r) = 0$ but $\phi'''(r) \neq 0$.

Differentiating $\phi(x) = x - (x - \phi(x))H$, we find that

$\phi'(r) = \phi''(r) = 0$ but $\phi'''(r) \neq 0$.

So $p \geq 3$. Conversely, if $p = 3$, then we can show

that $\phi'(r) = \phi''(r) = 0$ but $\phi'''(r) \neq 0$.

Hence, (12) is third order convergent method by the theorem 3.1 above.

2. Similarly (26) and (27) are of orders 2 and 3 respectively. Other super cubic and fourth order methods in this study shall be analyzed in detail in our future works.

3. Test Equations and Numerical Results

We selected the following 7 equations from many examples we have done.

- $f_1(x) = 3x^6 - 3x - 3 = 0$, a root $r \approx 1.134724$,
- $f_2(x) = 3x^3 - 3x - 3 = 0$, $r \approx 1.324718$.
- $f_3(x) = 3x - \cos x - 1 = 0$, $r \approx 0$.
- $f_4(x) = x^4 - x - e^x = 0$, $r \approx -0.52065$.
- $f_5(x) = \cos x + x^3 - e^x = 0$, $r \approx -0.649565$.
- $f_6(x) = e^{x-1} - \frac{1}{x} = 0$, $r = 1.000000$.
- $f_7(x) = \log_{10}(x) - 2x + 2$, $r = 1.000000$.

Comparisons were made relative to Newton method (NM), Chebyshev's method (CM), and the algorithms in equations (10), (12), (14), (16), and (26). C++ implementation was done for each algorithms and the number of iterations taken to converge to a root r to six decimal places was recorded and written in the body cells of the next **table-1** under each method. The stopping criteria were using the residual error $E_i = f(x_i)$ such that $f(x_i) \leq \epsilon$, for chosen $\epsilon = 10^{-6}$. We also checked this by other stopping criteria in the literature. The triplets of numbers in each cells of table-1 correspond to the

number of iterations needed for convergence for each of the three initial guesses of a root r. In the first column, 'Functions (f)' refers to the number of functional evaluations, 'Efficiency (e)' represents the computational efficiency index calculated by $e = p^{1/f}$. The average number of iteration **Nar** is estimated.

Table 1. Summary of comparison results

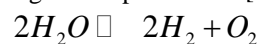
show	The number iterations taken for all triplets of initial guesses						
	fi	NM	CH	(10)	(12)	(16)	(26)
f_1	5, 7, 9	4, 5, 6	4, 5, 7	4, 5, 7	3, 4, 5	3, 4, 5	5, 7, 9
f_2	5, 5, 6	4, 3, 4	4, 3, 4	4, 3, 4	4, 3, 4	3, 3, 4	5, 5, 6
f_3	3, 3, 3	2, 2, 3	2, 2, 2	2, 2, 2	2, 2, 3	2, 2, 3	3, 3, 3
f_4	6, 4, 3	4, 3, 3	4, 3, 3	4, 3, 3	3, 3, 3	3, 3, 3	6, 4, 3
f_5	7, 5, 4	5, 3, 4	5, 3, 4	5, 3, 4	4, 3, 3	4, 3, 3	6, 5, 4
f_6	4, 4, 4	3, 3, 4	3, 3, 4	3, 3, 4	3, 3, 4	3, 3, 3	4, 4, 4
f_7	3, 3, 3	3, 2, 3	3, 2, 2	3, 2, 2	3, 2, 3	3, 2, 3	3, 3, 3
Nar	≈ 4	≈ 3	≈ 3	≈ 3	≈ 3	≈ 3	≈ 4
(p)	2	3	3	3	3	3	2
(f)	2	3	4	3	3	3	3
(e)	1.414	1.442	1.316	1.442	1.442	1.442	1.260

5. Some Areas of Applications

a) Applications in Engineering Models

Here, we shall offer some realistic applications of iterative root finding. For more examples, refer to [1, 9, 10, 12].

1. Water vapor splits into Oxygen and Hydrogen at higher temperature as [12]



The model for the reaction rate r and mole fraction f of H₂O is given by

$r = R_2/R_1$, with concentrations $R_1 = [H_2O]^2$, $R_2 = [H_2]^2[O_2]$ and

$$k = \frac{f}{1-f} \sqrt{\frac{2p_t}{2+f}} \quad (29)$$

Where $k = 0.04$ is a given reaction equilibrium constant and $p_t = 3.5\text{atm}$ is the total pressure at some temperature T . (29) is an intricate equation to determine f not easily and hence the importance of root finding. The equation to be solved for the mole fraction of water vapor in (29) becomes

$$F = F(f) = 0.04 - \frac{f}{1-f} \sqrt{\frac{7}{2+f}} = 0.$$

$$F = \frac{1}{25} - \frac{f(1-f)^{1/2}}{7(2+f)^{1/2}}$$

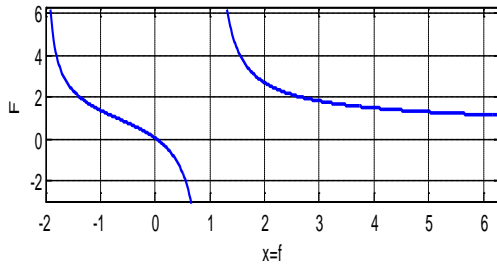


Fig-1: Graph of F indicating a root

The Newton's method for F is

$$f_{i+1} = f_i - F(f_i)/F'(f_i), i=0,1,2... \quad (3.1)$$

With an initial guess of $f_0 = 0.1$, the approximate mole fraction is $f = 0.0210$ in (0, 1).

- b) The model below can be used to estimate Oxygen level c (mg/l) in a river downstream from a sewage discharge [12]:

$$c = 10 - 20(e^{-0.15x} - e^{-0.5x}) \quad (30)$$

Where x is the distance downstream in kilometers. Then we may need the value of x for which Oxygen is minimum.

We mean the zeros of the first derivative of (30), which is

$$p = 3e^{(-3/20*x)} - 10 * e^{(-1/2*x)} \quad (31)$$

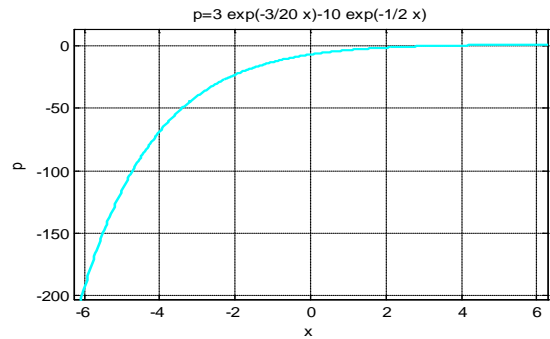


Fig 2: Root location for p

Taking an initial guess $x_0 = 4$ and applying Newton's method we get $x = 3.4399$ with minimum Oxygen level $c = 1.6433$.

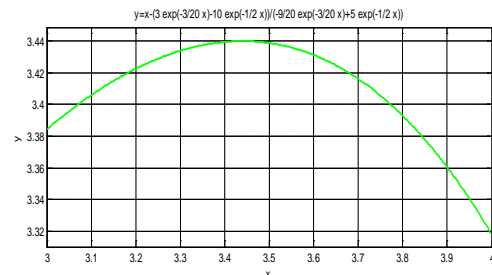


Fig 3 Newton fixed point for p on [3, 4]

6. Conclusions

In this work, we have applied derivative estimation (a very new concept in root finding) combined with Taylor's third order approximation to obtain new iterative methods for estimating simple roots of nonlinear equations, which is our main success. We investigated algorithms which are more efficient than some existing methods. We have observed that estimation of f''' in terms of lower derivatives could result in fast convergent or efficient methods, important in science and engineering. We also presented two examples of engineering applications.

In the future, we will make further analyses of these algorithms with possible extensions and other higher order iterative algorithms for multiple roots also. We hope that this result will be more precious and activate one to perform further research in the field.

Acknowledgments

I would like to thank all the mathematics staff of ASTU.

References

- [1] Alfio Quarteroni, Riccardo Sacco, Fausto Saleri., Numerical mathematics (Texts in applied mathematics; 37), Springer-Verlag New York, Inc., USA, 2000
- [2] Anthony Ralston, Philip Rabinowitz, A first course in numerical analysis (2nd ed.), McGraw-Hill Book Company, New York, 1978
- [3] Germund Dahlquist, and Ake Bjorck, Numerical methods in scientific computing Volume I, Siam - Society for industrial and applied mathematics. Philadelphia, USA, 2008
- [4] Jorgen Gerlach, Accelerated Convergence in Newton's Method, Society for industrial and applied mathematics, (1994), Siam Review 36, 272-276.
- [5] J.Stoer, R. Bulirsch, Texts in Applied mathematics 12, Introduction to Numerical Analysis (2nd ed.), springer-Verlag New York, Inc., USA, 1993.
- [6] Kou Jisheng, Li Yitian, Wang Xiuhua, A uniparametric Chebyshev-type method free from second derivative, Applied Mathematics and Computational, (2006), 179, 296-300.
- [7] Ljiljana D.Petkovic, and Miodrag S.Petkovic, on the fourth order root finding methods of Euler type, NOVI SAD J.MATH.,(2006), 36, 157-165.
- [8] M.M. Hussein, A note on one-step iteration methods for solving nonlinear equations, World Applied Sciences Journal 7 (special issue for applied math), IDOSI, Publications, (2009), 90-95.
- [9] Malvin R.Spencer (PhD Dissertation, 1994), Polynomial Real Root Finding in Bernstein Form, Department of engineering, Brigham Young University.
- [10] Pakdemirli, H. Boyacl and H. A. Yurtsever., A root finding algorithm with fifth order derivatives, Mathematical and Computational Applications, (2008), Vol.13, 123-128.
- [11] Philipp Rostalski (PHD.Thesis, 2009), Algebraic Moments – Real Root Finding and Related Topics, ETH Zurich (TU Hamburg-Harburg, German), Dissertation ETH Zurich No. 18371
- [12] Steven C Chapra, Raymond P Canale (2007), Numerical methods for engineers (5thed.), Tata McGraw-Hill, New york.
- [13] Shengfeng Li and Rujing Wang, Two fourth order *iterative methods based on Continued Fraction for root finding problems*, World Academy of Science, Engineering and Technology 60, 2011
- [14] YI JIN AND BAHMAN KALANTARI (communicated by Peter A. Clarkson), A combinatorial construction of high order algorithms for finding polynomial roots of known multiplicity, American Mathematical Society, (2010), 138, 1897-1906.

Author's Biography

Name: Tekle Gemechu (PHD)
BSC in applied mathematics (1997)
MSC in numerical analysis (2004)
PHD in applied maths (computational maths) (2014)
Title (Rank): Assistant professor
Address: Adama Science and Technology University (ASTU), Ethiopia

In mathematics and computing, a root-finding algorithm is an algorithm for finding zeroes, also called "roots", of continuous functions. A zero of a function f , from the real numbers to real numbers or from the complex numbers to the complex numbers, is a number x such that $f(x) = 0$. As, generally, the zeroes of a function cannot be computed exactly nor expressed in closed form, root-finding algorithms provide approximations to zeroes, expressed either as floating point numbers or as small isolating This is the semester project for the course of "Numerical Analysis and Computer Applications" it determines roots of equations by several methods. python root-finding appjar. 62 commits. This is the semester project for the course of "Numerical Analysis and Computer Applications" - CS213 for the 2 nd year of Computer and Systems Engineering Department to be delivered to Dr. Wafaa ElHaweet and the teaching assistant Eng. Omar Salaheldine. Our project is developed in Python programming language. We used appJar library to create the interactive GUI. Specifications: The pupose of the app is to find roots of any Equation based on some methods: Bracketing Methods. Open Methods. Looking inside our applications (reverse engineering). The applications which we use on our devices are actually properly packed .apk files containing all the data required to operate correctly. If we want to view their content and see what is hidden there, we have to make some necessary preparations. First of all, we need a tool that will let us communicate with the mobile device using our computer " Android Debug Bridge is what we are looking for. Secondly if we want to disassemble our .apk files into separated source code files (.smali classes), Apktool seems to be the perfect choice. First of all, we create a keystore.properties file in our root project directory, which contains the definition of our SECRET_TOKEN. Then we include this file in our app build.gradle.