Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.8, No.7, 2018

ROOT FINDING FOR NONLINEAR EQUATIONS

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

Abstract –Nonlinear equations /systems appear in most science and engineering models. For example, when solving eigen value problems, optimization problems, differential equations, in circuit analysis, analysis of state equations for a real gas, in mechanical motions /oscillations, weather forecasting, integral equations, image processing and many other fields of engineering designing processes. Nonlinear systems /problems are difficult to solve manually but they occur naturally in fluid motions, heat transfer, wave motions, chemical reactions, etc. This study deals with construction of iterative methods for nonlinear root finding, applying Taylor's series approximation of a nonlinear function f(x) combined with a new correction term in a quadratic or cubic model. Competent iterative algorithms of higher order were investigated. For test of convergence and efficiency, we applied basic theorems and solved some equations in C++.

Keywords – nonlinear equations, Taylor's approximation, iterative algorithms for roots, error correction

1. Introduction

Nonlinear problem solving with root finding is very common in science and engineering model applications. For example, in chemical and electrical engineering, environmental engineering, in physics, etc. The method can be direct /symbolic, graphical or numerical iterative. The iterative ones are derived using interpolations, perturbation method, variational technique, fixed point methods and so many others [1, 2].

There are several existing root solving methods such as Bisection method, Secant method, Regula Falsi, Newton's method and its variants /accelerators (Chebyshev's method, Halley's method, Super Halley's method….) [1, 2, 4, 7, 8, 9, 10,11, 12, 13]. However, choice of initial guesses, interval selections, existences of derivatives and acceleration convergence are some common drawbacks connected with algorithmic complexities.

In this article, we apply Taylor's approximation of $f(x)$ by quadratic and cubic model to derive new iterative algorithms, using a new correction term. We also discuss extension to higher dimensions for solving nonlinear system.

The article is organized as: introduction, basic methods based on Taylor's series, extensions to 2D, convergence analysis, procedures for computer codes, test problems, result and discussion, conclusion and references.

1.2 Construction methods based on Taylor's expansion

Consider the Taylor's approximation of a nonlinear function $f(x)$ about an approximate root $r = x_o + h$ in 1D.

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

If f = f(x,y) then the Taylor's series expansion in 2D is expressed as

$$f(x+h, y+k) = f(x, y)+hf_x+kf_y+1/2(h^2f_{xx}+2hkf_{xy}+k^2f_{yy})+\dots \quad (1a)$$

And if f = f(x, y, z) in 3D, then

$$f(x+h, y+k, z+l) = f(x, y, z)+hf_x+kf_y+lf_z+1/2(h^2f_{xx}+2hkf_{xy}+2hlf_{xz}+ 2klf_{yz} + h^2f_{yy} + k^2f_{zz}) \quad (1b)$$

The linear approximation from (1), $f(x_o) + hf'(x_o) \approx 0$ yields Newton's method

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

From a modified Newton's method for multiple root [12] one gets for simple roots

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

From the linear estimation of f(x) in (1), we can also derive a new method

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

Assuming that a combination (2c) of (2) & (2a) is of higher order, one has

$$x_{k+1} = x_k - w1\frac{f(x_k)}{f'(x_k)} + w2\frac{f(x_k)}{f'(x_k) - f(x_k)} : k = 0,1,...,n \quad (2c)$$

This satisfies

$$w1 - w2 = 1, w2 = \frac{-f'^3 f''}{f'^3\, f'' + (f'\,f'' - f'^2)\,f'^2 - 2f'^3\, f'' + 2f'^4}.$$

If we think about the quadratic interpolation approximation of (1)
$f(x_o + h) = f(x_o) + hf'(x_o) + 1/2h^2 f''(x_o) \approx 0$, we obtain

$$h = \frac{-f(x_0)}{f'(x_0) + 1/2hf''(x_0)} \quad (2d)$$

Then with a new (error) correction term h = f/(f'-f) from (2a) in the right part of (2d), one gets the two methods

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

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

There are also Halley's method (4a) and Chebyshev's method (4b)

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

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

And extended Newton (5) and Euler method (5b), see [1, 2, 4, 6, 8,9,10, 17, 19].

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

$$\varphi(x) = x - \frac{2H(x)}{1 + \sqrt{1 - 2p(x)}}, Q(x) = \frac{H(x)f''(x)}{f'(x)}, H(x) = \frac{f(x)}{f'(x)}. \quad (5a)$$

Suppose one ponders now the cubic model of f(x) as

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

$$(6) \Rightarrow hf' + 1/6h^3 f''' = h(f' + 1/6h^2 f''') = -(f + 1/2h^2 f''). \quad (7)$$

$$h = -\frac{f + 0.5h^2 f''}{f' + 1/6h^2 f'''}$$

Using a correction term h=f/(f'-f) in the right part, we obtain

$$h = -3\frac{2ff'^2 - 4f'f^2 + 2f^3 + f^2 f''}{6f'^3 - 12f'^2\, f + 6f'f^2 + f^2 f'''}$$

This yields an iteration function

$$\psi(x) = x - 3\frac{2ff'^2 - 4f'f^2 + 2f^3 + f^2 f''}{6f'^3 - 12f'^2\, f + 6f'f^2 + f^2 f'''} \quad (7a)$$

Replacing the higher order derivative by $\frac{f'f''}{f}$, we get another algorithm

$$\psi(x) = x - 3\frac{2ff'^2 - 4f'f^2 + 2f^3 + f^2 f''}{6f'^3 - 12f'^2 f + 6f'f^2 + ff'f''} \quad (7b)$$

However some of these methods need high memory computer.

1.3. Extensions to Higher Dimensions for Nonlinear Systems

Let us start with Newton's method to solve systems of nonlinear equations in 2D.

Assume a nonlinear system of equations [1, 4, 5, 7,15],

$$F = f(x,y) = \begin{Bmatrix} f1 = g(x,y) = 0 \\ f2 = h(x,y) = 0 \end{Bmatrix} \quad (8)$$

We hope to get X = (x, y) that satisfies f. If X_0 = (x0, y0) is an initial guess and X_1 = (x1,y1) is an enhanced approximation , then one can apply Taylor's linear estimation as

$$F(X1) = F(X0) + F'(X0)(X1 - X0) = 0 \quad (8a)$$

Where the Jacobean matrix of F is

$$J = F' = F'(X = (x,y)) = \begin{bmatrix} f1_x & f1_y \\ f2_x & f2_y \end{bmatrix} \quad (9)$$

The linear system (8a) can be solved by elimination. Or by Newton's method

$$X1 = X0 - J^{-1}F(X0). \quad (9a)$$

Provided that the inverse $J^{-1} = F'(X0)^{-1}$ exists. And the iteration process repeats until convergence.

The convex acceleration of Newton's method (9a) in 2D to solve F is [5]

$$X_{k+1} = X_k - \{I + 1/2L(X_k)[I - L(X_k)]^{-1}\}J^{-1}(X_k)F(X_k) \quad (10)$$

With I is identity operator/matrix and $L = F'(X_k)^{-1} F''(X_k) J^{-1}(X_k) F(x_k)$ is called the logarithmic degree of convexity of F. And the second partial derivative of F , F'' is a tensor whose elements are the partial derivatives $(F''(x))_{jik} = \frac{\partial^2 f_i(x)}{\partial x_k \partial x_i}$ [16]. Equation (10) is super-Halley's method to solve (8). And Halley's method in 2D is

$$X_{k+1} = X_k - \{I + 1/2L(X_k)[I - 0.5L(X_k)]^{-1}\}J^{-1}(X_k)F(X_k) \quad (11)$$

The Chebyshev's method in 2D is

$$X_{k+1} = X_k - \{I + 1/2L(X_k)\}J^{-1}(X_k)F(X_k) \quad (12)$$

Note that if $x = \phi(x)$ is Newton's iteration function in (2) then $\phi'(x) = L_f = L$ in 1D.

To extend the new methods in (3) and (4) , we first write (3) as below

$$x_{k+1} = x_k - (x_k - \phi(x_k))[1 - 0.5ff''f'^{-1}[f' - f]^{-1}]^{-1} \quad (13)$$

And (4) as

$$x_{k+1} = x_k - (x_k - \phi(x_k))[1 - 0.5ff''f'(f' - f]^{-1})^3] \quad (14)$$

Where $\phi(x)$ is Newton's iteration function to solve $f(x) = 0$ in 1D .

3. CONVERGENCE ANALYSIS

We shall use the following important definition and theorem.

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

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, $\phi^{(p)}(r) \neq 0$ [8], see also [1, 2, 4,8,17].

All the algorithms we presented need an appropriate choice of only one initial guess x_o in an interval Io = [a, b]. And random choices of x_o may lead us to unnecessary works. Notice that from Theorem 1.1 above and convergence of fixed point iteration method $x = \phi(x)$ [1, 2, 8, 18], $|\phi'(x)| < 1$ for all x in [a, b]. From which $|\phi'(x_o)| < 1$. The case $|\phi'(x_o)|$ >1 is divergence. And $|\phi'(x_o)| = 1$ needs especial treatment (reformulations or need for alternative methods) [1, 2].

Proof of the order of convergence (p)

The proof can be done applying theorem 3.1 and definition 3.1 or methods of proofs in [1, 2, 5, 6, 8].

1) Proof of order of convergence of algorithm in (3)

$$x_{k+1} = x_k - (x_k - \phi(x_k))[1 - 0.5ff''f'^{-1}[f' - f]^{-1}]^{-1} \quad (15)$$

We can write (15) as

$\phi(x) = x - (x - \varphi(x))H$. Where H(x) = $[1 - 0.5ff''f'^{-1}[f' - f]^{-1}]^{-1}$

And $x = \varphi(x)$ is Newton's iteration function.

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

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

Differentiating $\phi(x) = x - (x - \varphi(x))H$, 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, (3) or (15) is third order convergent method.

2) To prove order of convergence of algorithm in equation (4). We can write (4) as

$\phi(x) = x - (x - \theta(x))T$. Where T=$[1 - 0.5ff''f'(f' - f]^{-1})^3]$.

And $x=\theta(x)$ is Newton's iteration function. From which $\phi(x)=x$ and $\theta'(r)=0$ but $\theta''(r)\neq 0$. Differentiating $\phi(x)=x-(x-\theta(x))T$, we find that $\phi'(r)=\phi''(r)=0$ but $\phi'''(r)\neq 0$. So $p\geq 3$. On the other hand, if p = 3, then we can show that $\phi'(r)=\phi''(r)=0$ but $\phi'''(r)\neq 0$. Hence, (4) is third order convergent method. Similarly (2a) and (2b) are quadratic and (2c) is cubic convergent. The proofs for orders of the other algorithms can be done in a similar way. We shall make a detailed analysis in the future work. Related concepts are in [1, 2, 4, 5, 6, 7, 9, 10, 11, 17, 18, 19].

3.2. Procedure setting up for computer codes

1. Define an equation to solve, f(x)
2. Define derivatives, df, ddf, ….
3. Enter Inputs; x0, tolerance, number (Iter=n)
4. For i= 1, I++
   F = f(x0);
Df = df(x0), DDf=ddf(x0)
X = [define the method]

5. output x
6. Set error = c [c is correction term]
7. if error < = tol
   output x, f(x)
   end
8. else
   x0 = x
   i = i+1
   output" no solution /root" , end

4. TEST EQUATIONS, RESULTS &DISCUSSIONS

We have chosen five equations for test of efficiency.

$f_1(x)=2x^3-2x-2=0$, with x_o = 1, 2, 3 and root r ≈ 1.324718 in (1, 2),

$f_2(x)=6x-2\cos x-2=0$, x_o = 0, 1, 2, r ≈ 0.607102,

$f_3(x)=\cos x+x^3-e^x=0$, with x_o = -2.5, -1, -0.5, r ≈ -0.649565 in (-1, 0).

$f_4(x)=2x^6-2x-2=0$, x_o = 1, 2, 3, r ≈ 1.134724 in (1, 2)

$f_5(x)=\log_{10}(x)-2x+2$, x_o = 0.5, 1.5, 2.2, r =1.000000 in [1, 2).

Comparisons were relative to Newton method (NM), Chebyshev's method (CM), Halley's method (HM), (3), (4), (2a), (2b). 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 of the next **table-1** under each method. The stopping criteria were using the residual error such that $f(x_i)\leq\varepsilon$, for chosen $\varepsilon=10^{-8}$. We also checked this by other stopping criteria in the literature.

Hint: The triplets of numbers in each cells of table-1 correspond to the number of iterations needed for convergence with each of the three initial guesses of a root r of $f_i(x)=0, i=1,2,...,5.$ From the table, an algorithm (HM) for solving $f_1(x)=2x^3-2x-2=0$ converges at steps 4, 3, 4 for the initial guesses taken at $x_o=1,2,3$ respectively and being $\varepsilon=10^{-8}$ given. And NM for solving $f_2(x)=2x^3-2x-2=0$ converges at steps 5, 5, 6 taking the same initial guesses x_o = 1, 2, 3. In the first column, ´**Functions** (f) ´´ refers to the number of functional evaluations up to derivatives, and ´**Efficiency** (e) ´´ represents the computational efficiency index. "–"indicates slowness at the point. Algorithms are compared (may be ranked as fast, faster or very fast) relatively depending on their **Nar** values in the table being the uppermost, intermediate or the lowest respectively, see [3]. An algorithm with the least average number of iterations (**Nar**) to converge to a root r would be ranked very fast convergent. The higher the Nar value, the slower is an algorithm to converge. The lesser the Nar value, the faster is an algorithm to converge. Taking more initial guesses or more examples gives good ranking measure. In the table, the highest value of efficiency index is 1.442 (for third order). We can observe that all methods presented in the table are better competent with 3 to 5 average number of iterations to converge from both directions at x_o when an appropriate initial guess x_o is used. If x_o is not suitably chosen, then one can expect slow convergence and even divergence from a root.

Table 1 - Summary of comparison results

f, x0	NM	CH	(3)	(4)	(2a)	(2b)	HM
f_1: 1, 2, 3	5, 5, 6	4, 3, 4	2, 2, -	2, 4, -	3, 3,-	3, 4,6	3,3,4
f_2 :0, 1, 2	3, 3, 3	3, 3, 3	2, 3, 3	2, 3, 3	5, 5, -	4, 3,3	2,2,3
f_3 :-2.5,-1,-.5	7, 5, 4	4, 3, 4	4, 3, 3	4, 3, 3	6, 5, 4	6, 4,4	4,3,3
f_4 ,1,2,3	5, 7,9	3, 4, 4	2, 2, -	2, 2, 3	3, 5, 6	3,6, 9	3,4,4
f_5 :.5,1.5,2.2	3, 3, 3	3, 2, 3	3, 2, 3	3, 2, 2	4, 5, 6	4,5,6	3, 2,3
Nar, average iter	≈ 4	≈ 3	≈ 3	≈ 3	≈ 4	≈ 4	≈ 3
Order (p)	2	3	3	3	2	2	3
Functions (f)	2	3	3	3	2	2	3
Efficiency(e)	1.414	1.442	1.442	1.442	1.414	1.414	1.442

The new (error) correction might be fast convergence when it converges but does not affect the number of functional evaluations.

6. Conclusions

In this work, we have applied a new correction term in Taylor second and third order approximation to obtain some iterative methods for estimating simple roots of nonlinear equations. The correction technique does not affect the number of functional evaluations but convergence. We have shown possible extensions for solving 2D nonlinear systems. Competent methods were investigated. In the future, we will present further analyses of these algorithms and other higher order iterative algorithms with applications. We hope that this result will be very slyness and bring about one to perform further research.

ACKNOWLEDGEMENTS

The author would like to thank all the staff members of ASTU.

REFERENCES

[1] Quarteroni, A. et al.(2000), *Numerical mathematics (Texts in applied mathematics; 37)*, Springer-Verlag New York, Inc., USA.

[2] Ralston, A. (1978), Rabinowitz, P. A first course in numerical analysis (2nd ed.), McGraw-Hill Book Company, New York.

[3] Chun,C. and Neta,B.(2009), A third-order modification of Newton's method for multiple roots, *Applied Mathematics and Computation 211,p 474–479.*

[4] Dahlquist, G.(2008), Bjorck, A., *Numerical methods in scientific computing* Volume I Siam - Society for industrial and applied mathematics. Philadelphia, USA.

[5] Albeanu, A.(2008), ON THE GENERALIZED HALLEY METHOD FOR SOLVING NONLINEAR EQUATIONS, *ROMAI J., 4, 2(2008), 1-6.*

[6] Jain, M.K.,et al.(2007), *Numerical Methods for Scientific and Engineering Computation. Fifth Edition.* New Age International (P) Limited Publishers.

[7] J.Stoer, R.Bulirsch (1993), *Texts in Applied mathematics 12, Introduction to Numerical Analysis (2nd ed.)*, springer-Verlag New York, Inc., USA.

[8] Gerlach, J.(1994), Accelerated Convergence in Newton's Method, *Society for industrial and applied mathematics,Siam Review 36, 272-276.*

[9] Jisheng, K., et al.(2006), A uniparametric Chebyshev-type method free from second derivative, *Applied Mathematics and Computational 179, 296-300.*

[10] D.Petkovic, L.,et al.(2006), on the fourth order root finding methods of Euler type, *NOVI SAD J.MATH., 36, 157-165.*

[11] M.M. Hussein (2009), A note on one-step iteration methods for solving nonlinear equations, *World Applied Sciences Journal 7 (special issue for applied math), 90-95, IDOSI Publications.*

[12] Noor, M.A., et al.(2011), variation iteration technique for finding multiple roots of nonlinear equations, *Scientific Research and Essays Vol.6(6),pp.1344-1350.*

[13] Pakdemirli, H. et al.(2008), A root finding algorithm with fifth order derivatives, *Mathematical and Computational Applications,Vol.13, 123-128.*

[14] Shengfeng Li, S., and Wang, R.,(2011), Two fourth order iterative methods based on Continued Fraction forroot finding problems, *World Academy of Science, Engineering and Technology 60.*

[15] Chapra, S.(2012), *Applied Numerical methods with Matlab for Engineers and Scientists (3rd ed.)*, McGraw Hill Education (India), New Delhi.

[16] Khalil, S.A., et al.(2017), Types of Derivatives: Concepts and Applications (II), *Journal of Mathematics Research; Vol. 9, No. 1.*

[17] Gemechu, T.(2017) , Some Root Finding With Extensions to Higher Dimensions, *IISTE, ISSN 2224-5804, Vol.7, No.4.*

[18] Gemechu,T.(2017), Some Multiple and Simple Real Root Finding Methods, *IISTE, ISSN 2224-5804, Vol.7, No.10.*

[19] JIN Y. and KALANTARI, B. (communicated by Peter A. Clarkson)(2010), A combinatorial construction of high order algorithms for finding polynomial roots of known multiplicity, *American Mathematical Society 138, 1897-1906*.