A Modification of Newton Method for Solving Non-linear Equations

Louis Asiedu[1*], Charlotte Chapman-Wardy[1], K. Doku-Amponsah[1]

[1] Department of Statistics & Actuarial Science, University of Ghana, Legon, Ghana

*** Corresponding author:** lasiedu@ug.edu.gh

ABSTRACT

A large number of complex problems in Mathematics and its related fields require the solution of non-linear equations. The Newton method and its early modifications belong to the simplest but not sufficiently efficient techniques for solving non-linear equations. A desired characteristic of an efficient method of solving non-linear equations is to obtain a root with minimum error (usually lower than the precision limit) and lower number of iterations. In this study, we propose two methods of solving non-linear equations (Proposed methods 1 and 2) through a modification of the Newton Raphson's method with the forward and central difference approximations of the first derivative. The performance of the proposed methods are assessed along with an existing method (Secant Method) using three illustrations. The proposed method 2 outperformed the existing method (Secant method) and proposed method 1, yielding the lowest absolute relative approximate error and the least number of iterations when used to find the roots of the non-linear equations under consideration. The proposed methods 1 and 2 were found to be suitable alternatives for solving non-linear equations.

Keyword: Central difference approximation, Forward difference approximation, Secant method, Non-linear equations, Absolute relative approximate error

1. Introduction

Non-linear equations are used to represent occurrences in most aspects of our life. According to Maheshwari (2009), many complex problems in Science and Engineering contain functions of nonlinear and transcendental nature in the equation of the form $f(x) = 0$. Their solutions are therefore very important in providing answers to many questions confronting us. The large number of variables and its corresponding parameters in nonlinear equations result in complex situations which are mainly solved by numerical methods. Various kinds of numerical methods have therefore been propounded over the years to help provide an effective method of solving these equations. These methods include the Bisection, Newton Raphson, and Secant Methods. Newton's method has distinction of being most frequently used in the construction of multipoint methods (Petkovic, 2012).

According to Kaw (2009), one of the first numerical methods developed to find the root of a nonlinear equation was the bisection method (also called binary-search method). The method is based on a theorem which states that, "an equation $f(x) = 0$, where $f(x)$ is a real continuous function has at least one root between x_l and x_u if $f(x_l)f(x_u) < 0$. x_l and x_u are the two initial guesses of the root of the equation."

The advantages of this method as listed by Kaw (2009), are as follows: Since the method brackets the root, the method is guaranteed to converge. That is, they are always convergent since they are based on reducing the interval between the two guesses to obtain the root of the equation.

Also, the interval gets halved as iterations are conducted. So, one can guarantee the error in the solution of the equation.

According to Chhabra (2014) and Kaw (2009), some drawbacks of the bisection method are;

- The convergence of the bisection method is slow as it is simply based on halving the interval.
- If one of the initial guesses is closer to the root, it will take larger number of iterations to reach the root which consequently decrease the computational efficiency of the method.
- For a function $f(x)$ where there is a singularity and it reverses sign at the singularity, the bisection method may converge on the singularity.

The Newton Raphson's method of solving non-linear equations was proposed to address the drawbacks of the bisection method listed above. The Newton Raphson's method is an open method. This means, only one initial guess is required to get the iterative process started in finding the root of the equation.

2. Material and Methods

Let f be a real single-valued function of a real variable. If $f(\alpha) = 0$, then α is said to be a zero of f or equivalently a root of the equation (Petkovic, 2012).

2.1 Derivation of the Newton Raphson Method from Taylor series

According to Cirnu (2012), the Newton Raphson Method is derived by the first order of the Taylor expansion. It assumes that if the initial guess of the root of $f(x) = 0$ is at x_i, then when the tangent of the curve is drawn at $f(x_i)$, the point at which the tangent line crosses the $x - axis$, x_{i+1}, is an improved estimate of the root of the function $f(x_i)$.

Let x_i, $i = 1, 2, \dots$, be an initial guess of a root or solution to a non-linear equation $f(x) = 0$ and define $x_{i+1} = x_i + \delta x$, where δx is a small change in solution. Now a Taylor series expansion of a general function $f(x)$ is given as;

$$f(x_{i+1}) = \sum_{i=0}^{\infty} f^i(x_i) \left(\frac{x_{i+1}-x_i}{i!}\right)^i \quad (1)$$

where $f^0(x_i) = f(x_i)$, $f^1(x_i) = f'(x_i)$, $f^2(x_i) = f''(x_i)$,, with $f'(x)$ and $f''(x)$ being the first and second derivative of the function with respect to respectively.

$$\begin{aligned} f(x_{i+1}) = {} & f(x_i) + f'(x_i)\frac{(x_{i+1}-x_i)}{1!} + f''(x_i)\frac{(x_{i+1}-x_i)^2}{2!} + f'''(x_i)\frac{(x_{i+1}-x_i)^3}{3!} \\ & + \cdots \qquad (2) \\ = {} & f(x_i) + f'(x_i)\frac{(x_{i+1}-x_i)}{1!} + O(\delta x), \end{aligned}$$

where $O(\delta x)$ is the error due the truncation the Taylor series at the second term.

$$\approx f(x_i) + f'(x_i)(x_{i+1} - x_i) \quad (3)$$

Assuming x_{i+1} is a root of the equation then $(x_{i+1}) = 0$. Now, from equation (3) we have;

$$x_{i+1} \approx x_i - \frac{f(x_i)}{f'(x_i)} \quad (4)$$

Equation (4) is known as the Newton Raphson's Method for solving non-linear equations.

Generally, convergence in an open method is not guaranteed but if an iterative process of solving non-linear equation using an open method does converge, it is faster than a bracket method.

Some drawbacks of the Newton Raphson's method as stated by Kaw (2009) are that; the method diverges from root at inflection points, the method fails when the first derivative at a point is zero (division by zero) and results obtained from the Newton Raphson method may oscillate about the local maximum or minimum without converging on a root but converging on the local maximum or minimum. Eventually, it may lead to division by a number close to zero and may diverge. There are many methods developed on the improvement of quadratically convergent Newton's method so as to get a superior convergence order than the Newton method (Maheshwari, 2009). You may refer to Kou *et. al.*, (2006), Jisheng *et. al.*, (2006), Golbabai *et. al.*, (2007) and Abbasbandy (2003) for some modifications of the Newton's Methods to find the roots of non-linear and transcendental equations. In this study, we propose two modifications of the Newton's method (Proposed Method 1 and 2) to find the root of a nonlinear equation.

2.2 Forward Difference Approximation of the First Derivative

From equation (3), the forward difference approximation of the first derivative of the function is given by;

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

$$= \frac{f(x_i + \delta x) - f(x_i)}{\delta x} \qquad (5)$$

2.3 Backward Difference Approximation of the First Derivative

Let $x_{i-1} = x_i - \delta x, i = 1, 2, \dots$, where δx is a small change in solution; then a Taylor series expansion of a general function is given as:

$$f(x_{i-1}) = f(x_i) - f'(x_i)\frac{(x_i - x_{i-i})}{1!} + f''(x_i)\frac{(x_i - x_{i-1})^2}{2!} - f'''(x_i)\frac{(x_i - x_{i-1})^3}{3!} + \cdots (6)$$

$$= f(x_i) - f'(x_i)\frac{(x_i - x_{i-i})}{1!} + O(\delta x)$$

$$\approx f(x_i) - f'(x_i)(x_i - x_{i-1}) \qquad (7)$$

From equation (7), the backward difference approximation of first derivative of the function f is given as:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \qquad (8)$$

$$= \frac{f(x_i) - f(x_i - \delta x)}{\delta x} \qquad (9)$$

2.4 Central Difference Approximation of the First Derivative

Given that $\delta x = x_{i+1} - x_i$, we can rewrite equation (2) as:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\frac{\delta x}{1!} + f''(x_i)\frac{(\delta x)^2}{2!} + f'''(x_i)\frac{(\delta x)^3}{3!} + \cdots \qquad (10)$$

Also, given that $\delta x = x_i - x_{i-1}$ we can rewrite equation (6) as:

$$f(x_{i-1}) = f(x_i) - f'(x_i)\frac{\delta x}{1!} + f''(x_i)\frac{(\delta x)^2}{2!} - f'''(x_i)\frac{(\delta x)^3}{3!} + \cdots \qquad (11)$$

Subtracting (11) from (10) gives:

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)(\delta x) + 2f'''(x_i)\frac{(\delta x)^3}{3!}$$

$$= 2f'(x_i)(\delta x) + O(\delta x)^2,$$

where $O(\delta x)^2$ is the error due to the truncation of the Taylor series at the third term.

$$\approx 2f'(x_i)(\delta x) \qquad (12)$$

From equation (12), the central difference approximation of the first derivative of the function f is given as:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2\delta x}$$

$$= \frac{f(x_i + \delta x) - f(x_i - \delta x)}{2\delta x} \qquad (13)$$

2.5 Secant Method of Solving Non-linear Equations

Substituting (8) into (4) gives:

$$x_{i+1} \approx x_i - f(x_i)\left[\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}\right] \qquad (14)$$

which is also known as the Secant method of solving non-linear equations.

2.6. Proposed Method 1 (PM1): Using the Forward difference Approximation of the First Derivative

By substituting (5) into (4), we get:

$$x_{i+1} \approx x_i - f(x_i)\left[\frac{\delta x}{f(x_i + \delta x) - f(x_i)}\right]$$

$$= x_i - f(x_i)\left[\frac{x_i - x_{i-1}}{f(2x_i - x_{i-1}) - f(x_i)}\right] \qquad (15)$$

where $\delta x = x_i - x_{i-1}$.

2.7 Proposed Method 2 (PM2): Using the Central Difference Approximation of the First Derivative

By substituting equation (13) into (4), we get

$$x_{i+1} \approx x_i - f(x_i)\left[\frac{2\delta x}{f(x_i + \delta x) - f(x_i - \delta x)}\right]$$

$$= x_i - f(x_i)\left[\frac{2(x_i - x_{i-1})}{f(2x_i - x_{i-1}) - f(x_{i-1})}\right] \quad (16)$$

where $\delta x = x_i - x_{i-1}$.

2.8 The Algorithm

The following are the steps for the Secant and the proposed methods in finding the root of an equation $f(x) = 0$.

1. Use two initial guesses, x_i and x_{i-1} to estimate the new value of the root x_{i+1} using either equation (14), (15) or (16).
2. Find the absolute relative approximate error, $|\varepsilon_a|$, which is the absolute value of the relative approximate error, ε_a. The relative approximate error ε_a is defined as the ratio between the approximate error $(x_{i+1} - x_i)$ and the present approximation (x_{i+1}).

   Mathematically, the absolute relative approximate error is given as;

   $$|\varepsilon_a| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \quad (17)$$

   The absolute relative approximate error, $|\varepsilon_a|$ is usually expressed as percentage.
3. According to Petkovic (2012), iterating any root-finding method based on the evaluation of a function and its derivative makes sense only when the absolute value of the function do not exceed the precision limit ε_{tol} of the employed computer arithmetic. Also the number of iterations must be finite. Here, we compare the absolute relative approximate error, $|\varepsilon_a|$ to a pre-specified error tolerance ε_{tol}.

a. If $|\varepsilon_a| > |\varepsilon_{tol}|$, proceed to step 1 and go through the algorithm; else, terminate the algorithm.

b. The algorithm may also be terminated if the number of iterations exceeds the maximum number of iterations allowed; in which case the user must be notified accordingly.

4. Equivalently, suppose that a zero α lies in an interval of unit width (if α is real) or in the unit disk (if α is complex). Starting with an initial approximation x_0 to α, a stopping criterion may be given by

   $$|x_n - \alpha| < \varepsilon_{tol} = 10^{-m} \quad (18)$$

   where n is the iteration index, ε_{tol} is the required accuracy (precision limit) and m is the number of significant decimal digits of the approximation x_n.

2.9 Evaluation of the Numerical Methods

In this study, the absolute relative approximate error $|\varepsilon_a|$, the number of numerical iteration required n, and the number of significant digits correct in an answer, m are main performance metrics used to assess the numerical methods.

It is expected that;

$$|\varepsilon_a| \leq 0.5 \times 10^{2-m} \quad (19)$$

It follows from (18) that, the number of significant digits correct in an answer, m is given as;

$$m \leq 2 - log_{10} 2|\varepsilon_a| \quad (20)$$

3. Results and Discussion

In this section, we compare solutions of some non-linear equations using the Secant method, Proposed method 1 (Using the Forward difference approximation of the first derivative) and Proposed Method 2 (Using the Central difference approximation of the first derivative).

3.1 Illustration 1

The equation that gives the depth x to which the ball with radius m is submerged under water is given by

$$x^3 - 0.165x^2 + 0.0003993 = 0$$

Use the secant method, proposed method 1 and proposed Method 2 of finding roots of equations to find the depth x to which the ball is submerged under water.

3.1.1 Solutions to illustration 1

For all three methods, we set the two initial guesses to 0 and 0.18 since the diameter of the ball is 0.18m.

The error tolerance is also set at 10^{-3} and the maximum number of iterations at $n = 10$. Figure 1 is a graph of the function $f(x) = x^3 - 0.165x^2 + 0.0003993$.

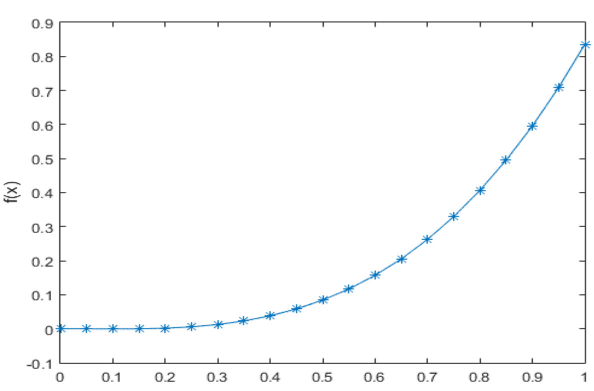

***Figure 1**: A sketch of the function in illustration 1*

From Table 1, a root of the equation is 0.146360 attained at the 7th iteration $(n = 7)$ with an error less than or equal to the pre-specified tolerance.

Table 1: Solution from Secant Method

iteration	root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	-0.14789	221.7130	-0.006443932	-5
2	0.140394	205.3383	-0.000085693	-5
3	0.144280	2.692927	-0.000032029	0
4	0.146598	1.581830	0.000003831	0
5	0.146351	0.169274	-0.000000140	3
6	0.146359	0.005955	-0.000000001	6
7	0.146360	0.000024	0.000000000	11

From Table 2 the Proposed Method 1, gave a root of 0.146360 at the 7th iteration with $|\varepsilon_a| \leq 10^{-3}$.

Table 2: Solution from Proposed Method 1

iteration	root	$\lvert\varepsilon_a\rvert$	$f(x)$	M
1	0.173571	3.704076	0.000657511	-1
2	0.152262	13.99473	0.000103988	-2
3	0.144583	5.311406	-0.000027510	-1
4	0.146699	1.442798	0.000005457	0
5	0.146373	0.222728	0.000000220	2
6	0.146359	0.009485	-0.000000001	5
7	0.146360	0.000051	0.000000000	11

It is evident from Table 3 that the root of equation (0.146360) is found on the 6th iteration with $|\varepsilon_a| \leq 10^{-3}$ when the Proposed method 2 is used to solve the equation.

Table 3: Solution from Proposed Method 2

iteration	Root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	0.167389	7.5340190	0.000466234	-1
2	0.15130	10.634103	0.000085675	-2
3	0.146791	3.0711020	0.000006947	0
4	0.146363	0.2925720	0.000000059	2
5	0.14636	0.0025300	0.000000000	7
6	0.14636	0.0000000	0.000000000	16

3.2: Illustration 2

Find a root of the non-linear equation, $cos(x) - xe^x = 0$, using the Secant method, Proposed methods 1 and 2. Take the maximum number of iterations $n = 10$, with initial guesses, $x_{i-1} = 0, x_i = 1$, and the error tolerance to be 10^{-3}. Figure 2 is the graph of the function $f(x) = cos(x) - xe^x$.

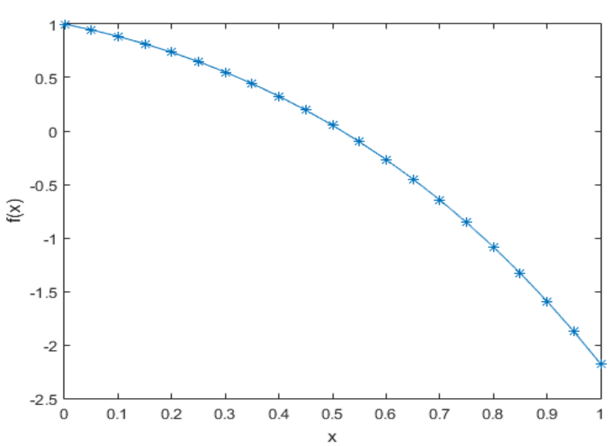

***Figure 2**: A sketch of the function in illustration 2*

From Table 4, the secant method provides a root, 0.517757 for the equation under consideration $n = 7$ iterations with $\lvert\varepsilon_a\rvert = 10^{-6}$ which is less than the pre-specified tolerance of 10^{-3}.

Table 4: Solution from Secant Method

Iteration	Root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	0.314665	217.797952	0.519871174	-5
2	0.446728	29.562231	0.203544778	-3
3	0.531706	15.982091	-0.042931093	-2
4	0.516904	2.863468	0.002592763	0
5	0.517747	0.162820	0.000030112	3
6	0.517757	0.001913	-0.000000022	7
7	0.517757	0.000001	0.000000000	14

From Tables 5 and 6 the Proposed methods 1 and 2 provided a root (0.517757) to the equation at 6th and 5th iterations (n=5 and n=6) respectively. The roots were obtained at precision limit less than the pre-specified tolerance of .

Table 5: Solution from Proposed Method 1

iteration	Root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	0.832673	20.09521	-1.241793162	-2
2	0.549698	51.47814	-0.099796443	-3
3	0.510267	7.727561	0.022643969	-1
4	0.518060	1.504182	-0.000919664	0
5	0.517759	0.057979	-0.000006207	4
6	0.517757	0.000394	0.000000002	9

Table 6: Solution from Proposed Method 2

iteration	Root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	0.731018	36.795478	-0.773972497	-3
2	0.553063	32.176402	-0.110618158	-3
3	0.519082	6.546315	-0.004033931	-1
4	0.517759	0.255454	-0.000005839	2
5	0.517757	0.000371	0.000000000	9

3.3 Illustration 3

Find a root of the nonlinear equation, $x^3 - e^{-x} = 0$, using the Secant, Proposed Methods 1 and 2 given the initial guesses $(0, 2)$ maximum number of iterations, $n = 10$ and pre-specified tolerance of 10^{-3}. Figure 3 is a graph of the function, $f(x) = x^3 - e^{-x} = 0$.

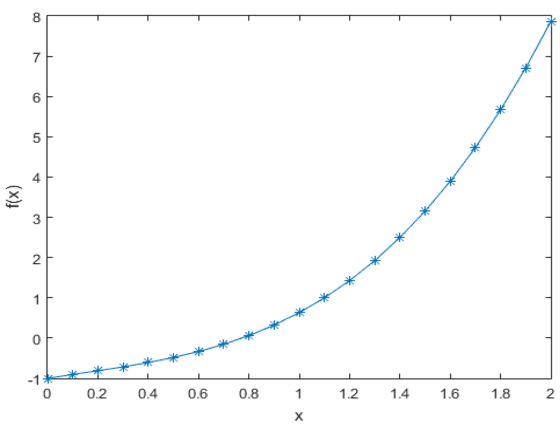

***Figure 3**: A sketch of the function in illustration 3*

It can be seen from Table 7 that, the Secant method of solving non-linear equation provided a root of 0.7728834 with $|\varepsilon_a| < 10^{-3}$ at $n = 9$ iterations.

Table 7: Solution from Secant Method

iteration	root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	0.225615	786.466472	-0.786541141	-6
2	0.386937	41.692024	-0.621202127	-3
3	0.993045	61.035345	0.608832428	-3
4	0.693038	43.288595	-0.16718665	-3
5	0.757672	8.530581	-0.033801676	-1
6	0.774051	2.116029	0.002635996	0
7	0.772866	0.153314	-0.000037316	3
8	0.772883	0.00214	-0.00000004	7
9	0.772883	0.000002	0.000000000	14

From Tables 8 and 9, the Proposed methods 1 and 2 gave a root (0.772883) to the non-linear equation, $f(x) = x^3 - e^{-x}$ at $n = 8$ and $n = 7$ iterations respectively. This was achieved at an absolute relative approximate error, $|\varepsilon_a|$ less than the precision limit, 10^{-3}.

Table 8: Solution from Proposed Method 1

iteration	root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	1.719705	16.299028	4.906709767	-2
2	1.083411	58.730648	0.933245222	-3
3	0.683083	58.606065	-0.186329777	-3
4	0.820009	16.698124	0.110958135	-2
5	0.78016	5.107848	0.016510283	-1
6	0.772661	0.970458	-0.000499788	1
7	0.772885	0.028904	0.000003587	4
8	0.772883	0.000206	0.000000001	9

Table 9: Solution from Proposed Method 2

iteration	root	$\lvert\varepsilon_a\rvert$	$f(x)$	m
1	1.515884	31.936198	3.263743963	-3
2	1.072222	41.377791	0.890444758	-3
3	0.849578	26.206468	0.185615279	-2
4	0.77943	8.9999	0.01484507	-1
5	0.772938	0.839976	0.000122994	1
6	0.772883	0.00706	0.000000009	6
7	0.772883	0.000000	0.00000000	15

Table 10 contains summary results of the illustrations. Proposed Method 2 (PM2) obtained the roots for all three non-linear equations with the smallest absolute relative approximate error, $|\varepsilon_a|$ and least number of iterations, n. This makes Proposed Method 2 (PM2) the preferred method among the three methods considered. Proposed Method 1 (PM1) also obtained the roots of the non-linear equations with lower number of iterations when compared to the Secant method. Although the $|\varepsilon_a|$ of PM1 were higher than those of the Secant method, they were less than the pre-specified precision limit of 10^{-3}.

Table 10: Comparison of the results obtained from the different methods

Non-Linear Equation	$\lvert\varepsilon_a\rvert$	Method	n	root (x)	$f(x)$
$x^3 - 0.165x^2 + 0.0003993 = 0$	0.000024	Secant	7	0.146360	0
	0.000051	PM1	7	0.146360	0
	0.000000	PM2	6	0.146360	0
$cos(x) - xe^x = 0$	0.000001	Secant	7	0.517757	0
	0.000394	PM1	6	0.517757	2×10^{-9}
	0.000371	PM2	5	0.517757	0
$x^3 - e^{-x} = 0$	0.000002	Secant	9	0.772883	0
	0.000206	PM1	8	0.772883	10^{-9}
	0.000000	PM2	7	0.772883	0

4. Conclusion and Recommendation

The study proposed a modification of the Newton's method with forward and central difference approximations of the first derivative. The proposed method 2, which was obtained through a modification of the Newton Raphson's method of solving non-linear equations with the central difference approximation of the first derivative, outperformed the Secant method and Proposed Method 1 (modification with forward difference approximation of the first derivative).

That is, the Proposed method 2 had the lowest absolute relative approximate error and the least number of iterations in finding the roots of the non-linear equations considered. This can be attributed to the fact that, the central difference approximation of the first derivative gives a better approximation when compared to the forward and backward difference approximations of the first derivative.

With a pre-specified precision limit of 10^{-3} Proposed Method 1 (PM1) also obtained the roots of the non-linear equations with lower number of iterations when compared to the Secant method.

The Proposed methods 1 and 2 are therefore recommended as viable alternatives for solving non-linear equations.

Conflict of Interest Statement

The authors declare that there is no conflict of interest.

References

Abbasbandy, S. (2003). Improving newton–raphson method for nonlinear equations by modified adomian decomposition method. *Applied mathematics and computation*, 145, 887–893.

Chhabra, C. (2014). Improvements in the bisection method of finding roots of an equation. In 2014 *IEEE International Advance Computing Conference (IACC)* (pp. 11–16). IEEE.

Cirnu, M. (2012). Newton-raphson type methods. *International Journal of Open Problems in Computer Science and Mathematics*, 5, 95–104. doi:10. 185 12816/0006108.

Golbabai, A., & Javidi, M. (2007). Newton-like iterative methods for solving system of non-linear equations. *Applied Mathematics and Computation*, 192, 546–551.

Jisheng, K., Yitian, L., & Xiuhua, W. (2006). A uniparametric chebyshev-type method free from second derivatives. *Applied Mathematics and Computation*, 179, 296–300

Kaw, A. (2009). Newton-raphson method of solving non-linear equations. General Engineering,

http://numerical methods. eng. usf. edu. Downloaded, 26.

Kou, J., Li, Y., & Wang, X. (2006). A modification of newton method with 195 third-order convergence. *Applied Mathematics and Computation*, 181, 1106– 1111.

Maheshwari, A. K. (2009). A fourth order iterative method for solving nonlinear equations. *Applied Mathematics and Computation*, 211, 383–391. URL: https://www.sciencedirect.com/science/article/pii/ 200 S0096300309000861 doi: https://doi.org/10.1016/j.amc.2009.01.047

Petkovic, M., Neta, B., Petkovic, L., & Dzunic, J. (2012). Multipoint methods for solving nonlinear equations. Academic press