International Multilingual Journal of Science and Technology (IMJST)
ISSN: 2528-9810
Vol. 1 Issue 4, November - 2016

Development Of Single To Dual Initial Root Values Mechanism Suitable For Bisection And Regular Falsi Iteration Methods

Ozuomba Simeon
Department of Electrical/Electronic and Computer Engineering
University of Uyo
Akwa Ibom State, Nigeria
simeonoz@yahoo.com

***Abstract*— In this paper, development of single to dual initial root values mechanism suitable for Bisection and Regular Falsi iteration methods is presented. Generally , to find the root of function, f(x) using the Bisection and Regular Falsi iteration methods, two initial guess root values (X_L and X_U) are required such that f(X_L)*f(X_U) < 0. As such, users of such method will continue to try two different sets of initial values until the required condition is met before they can proceed to use the Bisection or Regular Falsi iteration method to find the root of the function. In this paper, a procedure that can enable the user to provide only one initial guess root value and then the procedure will generate the required two initial root values that satisfy the Bisection or Regular Falsi iteration method is presented. The relevant mathematical expressions, algorithm and flowchart are presented along with numerical examples based on three different functions. The results show that for the different functions considered in the study, different arbitrary initial guess root values were selected and the procedure presented in this paper was able to generate the initial two root values that are suitable for Bisection and Regular Falsi iteration methods for finding the root of the functions. The results for the initial values for the first function, $f(x) = x^{3.5} - 80 = 0$ simulated with a single initial guess root value (X_0=2.7978856000) which is about 80% of the actual root (X = 3.4973572432) shows that it took two iterations to arrive at the two initial root values ($X_L = 2.7978856000$ and X_U=3.7348194090) such that f(X_L)*f(X_U) < 0 as required by the Regular Falsi or Bisection iteration method. Again, initial values for three different functions were simulated with a single initial guess root value that is about 80%, 40% and 20% of the actual root of the functions the results showed that for the different functions considered in the study and for the different arbitrary initial guess root values selected, the procedure presented in this paper was able to generate the initial two root values that are suitable for Bisection and Regular Falsi iteration methods for finding the root of the functions.**

Keywords*— *Bisection Method, Regular Falsi Method, Numerical Iteration Method, Root Of Function, Initial Root, Convergence Cycle

I. INTRODUCTION

Bisection and Regular Falsi methods are among the two most popular bracketing numerical iteration methods for finding the root of function, f(x). Usually, the bracketing iteration methods require two initial guess root values, x_L and x_U , such that $f(x_L) * f(x_U) < 0$ [1,2,3,4,5,6,7,8,9,10]. In essence, the two initial roots must bracket one of the roots (x) of the function, such that $x_L \leq$ x $\leq x_U$.

In practice, users of the Bisection and Regular Falsi methods are required to provide the two initial guess root values before the Bisection or Regular Falsi iteration can start [11,12,13,14,15]. However, in this paper, a method that enable the user to provide only one initial guess root value and then an algorithm developed in this paper will be used to generate the required two initial root values that satisfy the Bisection or Regular Falsi iteration method is presented. The mathematical expressions associated with the method are presented along with numerical examples based on three different functions. Matlab software was used to perform the simulation computations.

II. METHODOLOGY

Regular Falsi or Bisection iteration methods require that given two initial roots , x_L and x_U , then the condition $f(x_L) * f(x_U) < 0$ must be met before the iteration can proceed [16,17,18,19,20,21,22,23,24,25]. In this paper, the focus is on the use of single initial guess root value, x_0 to analytically and automatically determine the two initial roots x_L and x_U , which are suitable for Regular Falsi or Bisection iteration methods. Now, let the function, f(x) be such that;

$$f(x) = 0 \tag{1}$$

Also, when a single initial guess root value, x_0 is provided, then, the second initial guess root value, x_1 can be calculated as;

$$x_1 = 1 + \frac{(x_0)f(x_0)}{f(x_0) - 1} \tag{2}$$

In some cases, $f(x_0) * f(x_1) > 0$, as such , the two initial roots, x_0 and x_1 are not suitable for Regular Falsi or Bisection iteration method. However, from the available initial guess roots, x_0 and x_1 the suitable root values ,

x_L and x_U can be determined such that $f(x_L) * f(x_U) < 0$. The analytical expression used to realize x_L and x_U is given as;

$$x_k = 1 + \left(\frac{(K)f(x_{k-1})}{f(x_{k-1})-1}\right)(x_{k-1}) \qquad (3)$$

Where k is incremented at each iteration until x_{k-1} and x_K is reached at which $f(x_{K-1}) * f(x_K) < 0$. The flowchart for single guess root value to dual initial root values mechanism suitable for Bisection and Regular Falsi iteration methods is given in Figure 1. The algorithm is summarized as follows;

Step 1: Choose a single initial value, x_0
Step 2: Compute k =1
Step 3: Compute $x_k = 1 + \left(\frac{(K)f(x_{k-1})}{f(x_{k-1})-1}\right)(x_{k-1})$
Step 4: If $f(x_{k-1}) * f(x_k) < 0$ Then
Step 4.1: Goto Step 5
Step 4.2: Else
Step 4.2.1: k = k +1
Step 4.2.2: $x_{k-1} = x_k$
Step 4.2.3: Goto Step 3
Step 4.3: Endif
Step 5: Return The Initial Roots
Step 5.1: $x_L = x_{k-1}$
Step 5.2: $x_U = x_k$
Step 5.3: Return The Initial Roots x_L and x_U
// run the normal Bisection and Regular Falsi iteration method with the two initial values, x_L and x_U
Step 5.4: End

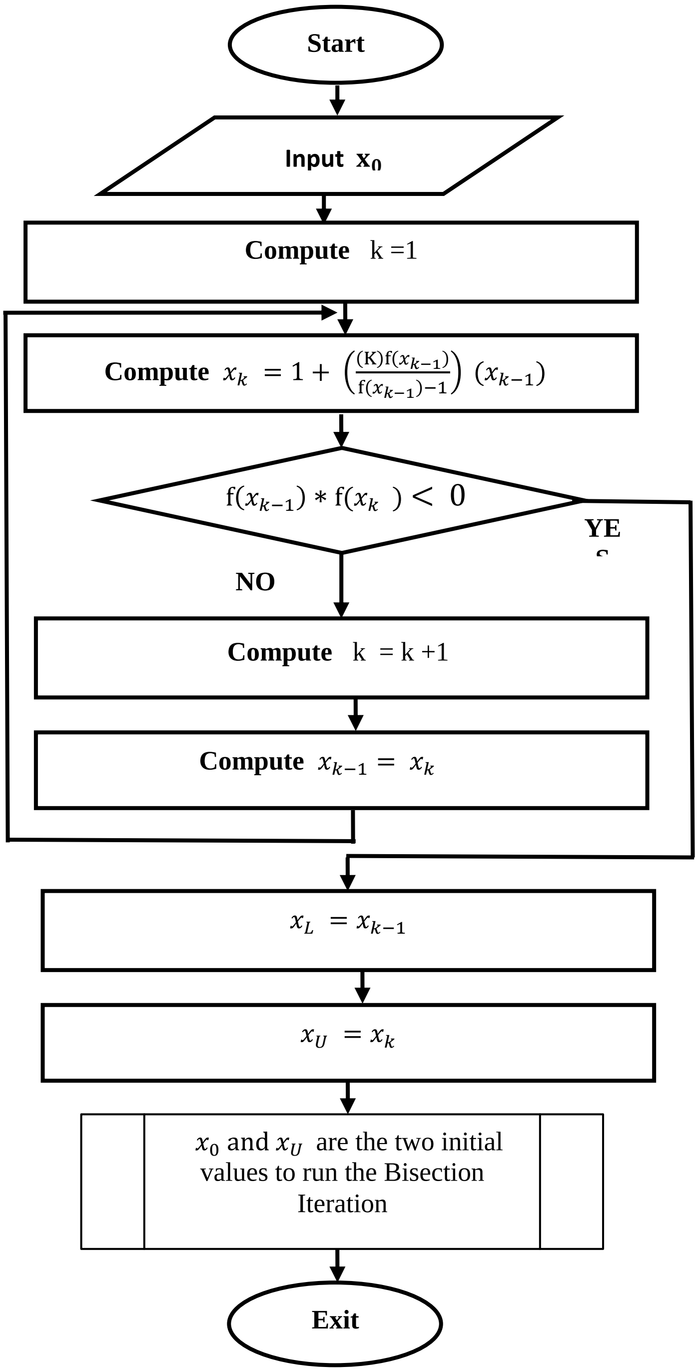

Figure 1: The flowchart for single to dual initial root values mechanism suitable for Bisection and Regular Falsi iteration methods

IV. NUMERICAL EXAMPLES, RESULTS AND DISCUSSIONS

The single guess root value to dual initial root values mechanism suitable for Bisection and Regular Falsi iteration methods is demonstrated using the following functions;

1) $\mathbf{f(x) = x^{3.5} - 80 = 0}$
2) $\mathbf{f(x) = \cos(x) - xe^{x} = 0}$
3) $\mathbf{f(x) = x^2 - \sin(x) - 0.5 = 0}$

The two initial root values mechanism was implemented in Matlab software. For each of the listed functions, a single guess root value was provided and the effective initial root values, x_L and x_U suitable for Bisection and Regular Falsi iteration methods were determined using the mechanism presented in this paper. The process was repeated for different initial single guess root values.

The results for the initial values for the first function, $f(x) = x^{3.5} - 80 = 0$ simulated with a single initial guess root value (X_0=**2.7978856000**) which is about 80% of the actual root (X = **3.4973572432**) is shown in Table 1.

The results show that it took 2 iterations to arrive at the 2 initial root values ($X_L = \mathbf{2.7978856000}$ and X_U=**3.7348194090**) such that $f(X_L)*f(X_U) < 0$ as required by the Regular Falsi or Bisection iteration method.

Again the results for the initial values for the first function, $f(x) = x^{3.5} - 80 = 0$ simulated with a single initial guess root value which is about 40% of the actual root is shown in Table2 while that with a single initial guess root value that is about 20% of the actual root is shown in Table 3.

Table 1 **The** results for the initial values for the first function , $f(x) = x^{3.5} - 80 = 0$ simulated with a single initial guess root value that is about 80% of the actual root

k		x_k	$f(x_k)$	$f(x_{k-1})*f(x_k)$	Actual Root ,X = **3.4973572432**	
0	$f(x_0)$=f(x)	**2.7978857946**	**-43.3642622559**		X_0=**2.7978856000**	X_1=**3.7348194090**
0	$f(x_1)$= = f(x+δ)	**3.7348195865**	**20.6801136174**	**-896.7778703869**	**(Xo/X)100%**	**80.%**
1	$f(x_2)$== f(1-(Δ*x))	**8.3012685342**	**1568.1837202834**	**32430.2175084084**	X_L=**2.7978856000**	X_U=**3.7348194090**
2	$f(x_3)$== f(1-(2*Δ)*x)	**25.3424572485**	**81855.2020236826**	**128363995.2340510000**	$f(X_L)*f(X_U)$ =-**896.7778703869**	

Table 2 The results for the initial values for the first function , $f(x) = x^{3.5} - 80 = 0$ simulated with a single initial guess root value that is about 40% of the actual root

k		x_k	$f(x_k)$	$f(x_{k-1})*f(x_k)$	Actual Root ,X = **3.4973572432**	
0	$f(x_0)$=f(x)	**1.3989428973**	**-76.7618276759**		X_0=**1.3989428973**	X_1=**2.3809527993**
0	$f(x_1)$= f(x+δ)	**2.3809527993**	**-59.1729504293**	**4542.223823932**	**(Xo/x)100%**	**40.00**
1	$f(x_2)$= f(1-(Δ*x))	**5.7006685400**	**362.3237565874**	**-21439.765687911**	X_L=**2.3809527993**	X_U=**5.7006685400**
2	$f(x_3)$= f(1-(2*Δ)*x)	**17.8820775895**	**24100.3683437856**	**8732135.99345926**	$f(X_L)*f(X_U)$ = **-21439.7656879110**	

Table 3 The results for the initial values for the first function , $f(x) = x^{3.5} - 80 = 0$ simulated with a single initial guess root value that is about 20% of the actual

j		xJ	f(xj)	f(xj-1)*f(xj)	XL Actual Root ,X = **3.4973572432**	
0	f(xo)=f(x)	**0.6994714486**	**-79.7137832989**	**1.0000000000**	X_0=**0.6994714486**	X_1=**1.6908053767**
1	f(x1) = f(x+δ)	**1.6908053767**	**-73.7146759877**	**5876.0757076318**	**(Xo/x)100%**	**20.00**
2	f(x2) = f(1-(Δ*x))	**4.3397144303**	**90.2605503147**	**-6653.5272209208**	**1.6908053767**	X_U=**4.3397144303**
3	f(x3) = f(1-(2*Δ)*x)	**13.8578431664**	**9826.8308280915**	**886975.1583929230**	$f(X_L)*f(X_U)$ = **-6653.5272209208**	

The results for the initial values for the second function, $f(x) = \cos(x) - xe^x = 0$ simulated with a single initial guess root value (X_0=**0.4142056000**) that is about 80% of the actual root (X = **0.5177573637**) is shown in Table 4. It took 2 iterations to arrive at the 2 initial root values (X_L =**0.4142056000** and X_U =**0.8319052806**) such that $f(X_L)*f(X_U) < 0$ as required by the Regular Falsi or Bisection iteration method. Again the results for the initial values for the second function, $f(x) = \cos(x) - xe^x = 0$ simulated with a single initial guess root value that is about 40% of the actual root is shown in Table 5 while that with a single initial guess root value that is about 20% of the actual root is shown in Table 6.

Table 4 The results for the initial values for the first function , $f(x) = \cos(x) - xe^{x} = 0$ simulated with a single initial guess root value that is about 80% of the actual root

k		x_k	$f(x_k)$	$f(x_{k-1})$* $f(x_k)$	Actual Root ,X = **0.5177573637**	
0	$f(x_0)$=f(x)	**0.4142056000**	**0.2886735827**		X_0=**0.4142056000**	X_1=**0.8319052806**
0	$f(x_1)$= f(x+δ)	**0.8319052806**	**-1.2379938055**	**-0.3573761072**	**(Xo/x)100%**	**80.00**
1	$f(x_2)$= f(1-(Δ*x))	**0.3247851565**	**0.4983032606**	**-0.6168963499**	X_L=**0.4142056000**	X_U=**0.8319052806**
2	$f(x_3)$= f(1-(2*Δ)*x)	**0.6045828226**	**-0.2839425497**	**-0.1414894983**	$f(X_L)$*$f(X_U)$	**-0.3573761072**

Table 5 The results for the initial values for the first function , $f(x) = \cos(x) - xe^{x} = 0$ simulated with a single initial guess root value that is about 40% of the actual

k		xJ	f(xj)	f(xj-1)*f(xj)	Actual Root ,X = **0.5177573637**	
0	f(xo)=f(x)	**0.2071028000**	**0.7238717368**	**1.0000000000**	X_0=**0.2071028000**	X_1=**0.4570788887**
0	f(x1) = f(x+δ)	**0.4570788887**	**0.1754106529**	**0.1269748140**	**(Xo/x)100%**	**40.00**
1	f(x2) = f(1-(Δ*x))	**-1.3964695620**	**0.5190282082**	**0.0910430769**	X_L=**-1.3964695620**	X_U=**11.9825575486**
2	f(x3) = f(1-(2*Δ)*x)	**11.9825575486**	**-1916496.1749836200**	**-994715.5757877420**	$f(X_L)$*$f(X_U)$	**-994715.5757877420**

In the case of Table 6, the initial guess root , X_0=0.1035514000 is 20% of the actual Root ,X = 0.5177573637. However, after the second iteration, the generated 2 initial root values (X_L = 0.2421004890 and X_U = -2.5438988221) bracketed another root of the function which in this case is X = -1.8639951924. At this point, the initial guess root , X_0=0.1035514000 is -5.56% of the actual root that is enclosed by X_L and X_U.

The results for the initial values for the third function,$f(x) = x^2 - \sin(x) - 0.5$ simulated with a single initial guess root value X_0=**0.9568656266** that is about 80% of the actual root (x = **0.5177573637**) is shown in Table 7. It took 2 iterations to arrive at the 2 initial root values (X_L=**0.9568656266** and X_1=**1.2742668767**) such that $f(X_L)*f(X_U) < 0$ as required by the Regular Falsi or Bisection iteration method.

Again the results for the initial values for the third function,$f(x) = x^2 - \sin(x) - 0.5$ simulated with a single initial guess root value that is about 40% of the actual root is shown in Table 8 while that with a single initial guess root value that is about 20% of the actual root is shown in Table 9.

Table 6 The results for the initial values for the first function , $f(x) = \cos(x) - xe^{x} = 0$ simulated with a single initial guess root value that is about 20% of the actual

k		x_k	$f(x_k)$	$f(x_{k-1})$* $f(x_k)$	Actual Root ,X = **-1.8639951924**	
0	f(xo)=f(x)	**0.1035514000**	**0.8797941953**	**1.0000000000**	X_0=**0.1035514000**	X_1=**0.6624193527**
0	f(x1) = f(x+δ)	**0.2421004890**	**0.6624193527**	**0.5827927013**	**(Xo/x)100%**	**-5.56**
1	f(x2) = f(1-(Δ*x))	**-2.5438988221**	**-0.6267881287**	**-0.4151965865**	X_L = **0.2421004890**	X_U= **-2.5438988221**
2	f(x3) = f(1-(2*Δ)*x)	**56.8568886527**	**-28016642627163800**	**17560499005495100**	$f(X_L)$*$f(X_U)$	**-0.4151965865**

Table 7 The results for the initial values for the first function , $f(x) = x^2 - \sin(x) - 0.5$ simulated with a single initial guess root value that is about 80% of the actual root

k		x_k	$f(x_k)$	$f(x_{k-1})$* $f(x_k)$	Actual Root ,X = **0.5177573637**	
0	f(xo)=f(x)	**0.9568656266**	**-0.4017980940**	**1.0000000000**	X_0=**0.9568656266**	X_1=**1.2742668767**
1	f(x1) = f(x+δ)	**1.2742668767**	**0.1673997218**	**-0.0672608892**	**(Xo/x)100%**	**80.00**
2	f(x2) = f(1-(Δ*x))	**1.5485337534**	**0.8982045864**	**0.1503591979**	X_L=**0.9568656266**	X_U=**1.2742668767**
3	f(x3) = f(1-(2*Δ)*x)	**1.8228006302**	**1.8541875340**	**1.6654397470**	$f(X_L)$*$f(X_U)$	**-0.0672608892**

Table 8 The results for the initial values for the first function , $f(x) = x^2 - \sin(x) - 0.5$ simulated with a single initial guess root value that is about 40% of the actual root

k		x_k	f(x_k)	f(x_{k-1})* f(x_k)	Actual Root ,X =	1.1960820333
0	f(xo)=f(x)	0.4784328133	-0.7314905655	1.0000000000	X_0=0.4784328133	X_1=1.2021201248
0	f(x1) = f(x+δ)	1.2021201248	0.0122875601	-0.0089882343	(Xo/x)100%	40.00
1	f(x2) = f(1-(Δ*x))	1.4042402496	0.4857291067	0.0059684256	X_L=0.4784328133	X_U=1.2021201248
2	f(x3) = f(1-(2*Δ)*x)	1.6063603744	1.0810259865	0.5250857867	f(X_L)*f(X_U)	-0.0089882343

Table 9 The results for the initial values for the first function , $f(x) = x^2 - \sin(x) - 0.5$ simulated with a single initial guess root value that is about 20% of the actual root

kj		x_k	f(x_k)	f(x_{k-1})* f(x_k)	Actual Root ,X =	1.1960808193
0	f(xo)=f(x)	0.2392164067	-0.6797169303	1.0000000000	X_0=0.2392164067	X_1=1.0968016924
0	f(x1) = f(x+δ)	1.0968016924	-0.1867781120	0.1269562449	(Xo/x)100%	20
1	f(x2) = f(1-(Δ*x))	1.1936033848	-0.0050131305	0.0009363431	X_L = 1.1936033848	X_U=1.2904050772
2	f(x3) = f(1-(2*Δ)*x)	1.2904050772	0.2041980226	-0.0010236713	f(X_L)*f(X_U)	-0.0010236713

V CONCLUSION

A procedure to use one initial guess root value for a function to generate two initial root values that are suitable for Bisection and Regular Falsi iteration methods is presented. The relevant mathematical expressions, algorithm and flowchart are presented along with numerical examples based on three different functions. The results show that for the different functions considered in the study, different arbitrary initial guess root values were selected and the procedure presented in this paper was able to generate the initial two root values that are suitable for Bisection and Regular Falsi iteration methods for finding the root of the functions.

REFERENCES

1. Solanki, C., Thapliyal, P., & Tomar, K. (2014). Role of bisection method. *International Journal of Computer Applications Technology and Research*, *3*(8), 535-535.
2. Chun, S., & Kwasinski, A. (2011). Analysis of classical root-finding methods applied to digital maximum power point tracking for sustainable photovoltaic energy generation. *IEEE transactions on power electronics*, *26*(12), 3730-3743.
3. Bahgat, M. S., & Hafiz, M. A. (2014). Three-step iterative method with eighteenth order convergence for solving nonlinear equations. *International Journal of pure and applied Mathematics*, *93*(1), 85-94.
4. Chapra, S. C., & Canale, R. P. (2010). *Numerical methods for engineers*. Boston: McGraw-Hill Higher Education,.
5. McNamee, J. M., & Pan, V. (2013). *Numerical Methods for Roots of Polynomials-Part II*. Newnes.
6. Herceg, D., & Herceg, Đ. (2008). Numerical mathematics with GeoGebra in high school. *Teaching Mathematics and Computer Science*, *6*(2), 363-378.
7. Isaacson, E., & Keller, H. B. (2012). *Analysis of numerical methods*. Courier Corporation.
8. Collatz, L. (2012). *The numerical treatment of differential equations* (Vol. 60). Springer Science & Business Media.
9. Solanki, C., Thapliyal, P., & Tomar, K. (2014). Role of bisection method. *International Journal of Computer Applications Technology and Research*, *3*(8), 535-535.
10. Szymkiewicz, R. (2010). *Numerical modeling in open channel hydraulics* (Vol. 83). Springer Science & Business Media.
11. Yun, B. I. (2008). A non-iterative method for solving non-linear equations. *Applied Mathematics and Computation*, *198*(2), 691-699.
12. Suckow, S., Pletzer, T. M., & Kurz, H. (2014). Fast and reliable calculation of the two-diode model without simplifications. *Progress in photovoltaics: research and applications*, *22*(4), 494-501.
13. McNamee, J. M., & Pan, V. Y. (2012). Efficient polynomial root-refiners: A survey and new record efficiency estimates. *Computers & Mathematics with Applications*, *63*(1), 239-254.
14. Engeln-Müllges, G., & Uhlig, F. (2013). *Numerical algorithms with C*. Springer Science & Business Media.

15. Cohen, H. (2011). *Numerical approximation methods* (p. 485). New York: Springer.
16. Bermel, P. (2013). ECE 595, Section 10 Numerical Simulations Lecture 6: Finding Special Values.
17. Walther, A., Bause, F., & Henning, B. (2012). Computing roots for the modelling of waves in acoustic waveguides. *Pamm*, *12*(1), 763-764.
18. Ning, S. A. (2014). A simple solution method for the blade element momentum equations with guaranteed convergence. *Wind Energy*, *17*(9), 1327-1345.
19. Färber, M., & Döring, U. (2011). Numerical Root Finding from an Engineer's. *Mechanisms, Transmissions and Applications*, *3*, 91.
20. Färber, M., & Döring, U. (2012). Numerical Root Finding from an Engineer's Point of View. In *Mechanisms, Transmissions and Applications* (pp. 91-102). Springer, Dordrecht.
21. Chalvatzis, C. K. (2013). *Utility-based Shortfall Risk: Implementation using stochastic, deterministic root-finding aglorithms and Fourier Transform* (Bachelor's thesis).
22. Bachrathy, D., & Stépán, G. (2012). Bisection method in higher dimensions and the efficiency number. *Periodica Polytechnica Mechanical Engineering*, *56*(2), 81-86.
23. LING, SAM TEEK. "USING MS EXCEL IN SOLVING NONLINEAR EQUATIONS." *No. Paper ID* (2010): 684.
24. Kwon, M. S. (2009). A numerically stable analysis method for complex multilayer waveguides based on modified transfer-matrix equations. *Journal of lightwave technology*, *27*(20), 4407-4414.
25. Adebile, E. A., & Idoko, V. I. (2010). The Exact Root Algorithm for Computing the Real Roots of an Nth Degree Polynomial. *Journal of Mathematics and Statistics*, *6*(3), 226-232.