Strict Complementary Root-Based Seeded Secant Iteration Method For Finding The Roots Of Nonlinear Functions

Olutimo Akinwale Lewis[1]
Department of mathematics, Lagos State University, Nigeria
aolutimo@yahoo.com

Uwakwe Chikwado[2]
Department of Electrical/Electronic Engineering Imo State Polytechnic, Umuagwo, Owerri, Nigeria

Afolayan Jimoh Jacob[3]
Department of Electrical/Electronic and Computer Engineering University of Uyo

***Abstract*— In this paper, a strict complementary root-based seeded secant iteration (SCSSI) method for finding the roots of nonlinear functions is presented. The SCSSI method requires only one initial guess root value making it easier to implement than the classical secant iteration method. The mathematical and algorithmic procedure required to implement the SCSSI method are presented along with numerical examples that were implemented in Mathlab software. The result of the SCSSI method applied for the roots of a case study function, $\mathrm{f(x)} = \mathrm{x^6 - 2x - 1} = 0$ with initial $x_0 = 1$ and error tolerance of $\varepsilon = 10^{-4}$ showed that the SCSSI algorithm converged at the 3rd cycle with actual root value of -0.49283556 whereas the CSSI algorithm converged at the 5th cycle with actual root value of -0.49283556. Also, the results for the case of $x_0 = 5$ show that the SCSSI algorithm converged at the 5th cycle with actual root value of 1.22981 whereas the CSSI algorithm converged at the 17th cycle. However, the results for the case of $x_0 = 10$ show that the SCSSI algorithm converged at the 6th cycle with actual root value of 1.22981 whereas the CSSI algorithm did not converge. The later result show the limitation of the CSSI algorithm which is that when the initial guess root is far above or far below the actual root, the CSSI may never converge. In essence, the CSSI is suitable for cases where the approximate value of the actual root can be estimated in advance. However, the SCSSI algorithm solved that problem by using a binary back-off mechanism.**

Keywords*— *Classical Secant Method, Strict Complementary Root-Based Mechanism, Error Tolerance¸ Seeded Secant Iteration, Convergence Cycle

1. INTRODUCTION

Over the years, researchers have developed different numerical iteration approaches that are applied in finding the roots of nonlinear equations, as well as in solving transcendental equations [1,2,3,4,5,6,7,8,9,10]. Among the numerous numerical iteration approaches, the classical secant approach has been widely applied [11,12,13,14,15,16,17]. However, because secant method required two initial guess root values, researchers have gone ahead to develop modified versions of the secant approach such that only one initial guess root is required [18,18,19,20,21,22]. One of such method is the complementary root-based seeded secant method which uses a fixed point iteration-like method [23,24,25,26,27,28] once to generate the required two initial guess roots from a single initial guess root value.

Evaluation result on the complementary root approach shows that the method works best when the single initial guess root, x_0 is close to the actual root, x_{act} . However, when the initial guess root, x_0 is much greater than x_{act} , then , the complementary root, $\mathrm{g}(x_0)$ (denoted here as $x_1 = \mathrm{g}(x_0)$ obtained is very high compared to the value of x_0 and hence, the value of x_2 obtained using the secant iteration method is almost equal to x_0 . In that case, the best approach is to reduce the absolute value of x_0 and then repeat the computation of $\mathrm{g}(x_0)$ and x_2 in the next iteration. A simple approach to reduce x_0 is to use half of the current value of x_0, (that is, in the next iteration, $x_0 = \frac{x_0}{2}$). This is a binary back-off approach. The binary back-off procedure is repeated until a point at which the absolute value of x_1 is relatively close to the value of x_0 . At this point, the secant iteration continues with $x_0 = x_2$ and $x_1 = \mathrm{g}(x_0)$ and then both x_0 and x_1 are used to compute the next value of x_2.
Specifically, in this paper, the development of the Complementary Root-Based Seeded Secant Iteration (SCSSI) method for finding the roots of nonlinear equations and for solving transcendental equations is presented. The SCSSI algorithm is also presented. Also, some numerical examples are used to compare the convergence of the SCSSI method with that of the Onetime Complementary Root Seeded Secant (CSSI) method.

2. METHODOLOGY

2.1 Development of the Strict Complementary Root-Based Seeded Secant Iteration Method

The complete procedure for the strict complementary root-based seeded secant iteration method is presented in respect of a function $f(x)$ which has been re-arranged in its complementary root form, denoted as $\hat{f}(x)$ where,

$$\hat{f}(x) = x - g(x) \qquad (1)$$

The complementary root form, $\hat{f}(x)$ is such that the value of x at which $g(x)$ = x gives the root of the function f(x). Now , let x_0= 1 be the initial guess root of function f(x) and $x_1 = \mathrm{g}(x_0)$, then, then applying secant approach for the next guess root, x_2 gives;

$$x_2 = \frac{[(x_0)\mathrm{f}(x_1)] - [(x_1)\mathrm{f}(x_0)]}{\mathrm{f}(x_1) - \mathrm{f}(x_0)} \qquad (2)$$

The procedure for computing the actual root, x_{ac} of the function f(x) is given as follows;

Step 1 Get x_0
Step 2 Compute $x_1 = g(x_0)$
Step 3 Compute $x_2 = \frac{[(x_0)f(x_1)]-[(x_1)f(x_0)]}{f(x_1)-f(x_0)}$
Step 4 Compute $f(x_2)$
Step 5 if $(|f(x_2)| \leq \in)$ then $x_{ac} = x_2$; goto step 7
endif
Step 6 If $\left(\left(\left|\frac{max(x_0,x_1)}{min(x_0,x_1)}\right| > 2\right) \; and \; (|x_0| - |x_2| < 0.01|x_0|)\right)$ then

$x_0 = \frac{x_0}{2}$; *goto step* 2
else
$x_0 = x_2$; *goto step* 2
endif
Step 7 Output x_{ac}
Step 8 End

2.2 Sample function and it complementary root form

Consider the nonlinear function of x given as;

$$f(x) = x^6 - 2x - 1 = 0 \quad (3)$$

It can be arranged in the complementary root form in different ways. One, by making x the subject of the formula using 2x as the basis for the x as follows;

$$x = g(x) = \frac{x^6-1}{2} \quad (4)$$

Hence, the complementary root form of f(x) is

$$\hat{f}(x) = x - g(x) = x - \frac{x^6-1}{2} \quad (5)$$

Also, by making x the subject of the formula using x^6 as the basis for the x as follows;

$$x = g(x) = \sqrt[6]{(2x+1)} = (2x+1)^{\frac{1}{6}} \quad (6)$$

Hence, the complementary root form of f(x) is

$$\hat{f}(x) = x - g(x) = x - (2x+1)^{\frac{1}{6}} \quad (7)$$

Numerical example is given in respect of the nonlinear function, $f(x) = x^6 - 2x - 1 = 0$. For the given function, the complementary root form used is,

$$\hat{f}(x) = x - g(x) = x - \frac{x^6-1}{2} \quad (8)$$

Where,

$$g(x) = \frac{x^6-1}{2} \quad (9)$$

The iterative solution for the Strict Complementary Root-Based Seeded Secant Iteration (SCSSI) method is implemented in Mathlab software for different initial guess roots and the convergence performance is compare with that of the Onetime Complementary Root Seeded Secant (CSSI) method.

3. RESULTS AND DISCUSSION

The result of the SCSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 1$ which gives $x_1 = g(x_0) = 0$ is presented in Table 1. The SCSSI was simulated with error tolerance of $\varepsilon = 10^{-4}$ and the results show that the SCSSI algorithm converged at the 3rd cycle with actual root value of -0.49283556 whereas the CSSI algorithm (in Table 2) converged at the 5th cycle with actual root value of -0.49283556. Also, the results for the case of $x_0 = 5$ show that the SCSSI algorithm (in Table 3) converged at the 5th cycle with actual root value of 1.22981 whereas the CSSI algorithm (in Table 4) converged at the 17th cycle with actual root value of 1.22981.

Again, the results for the case of $x_0 = 10$ show that the SCSSI algorithm (in Table 5) converged at the 6th cycle with actual root value of 1.22981 whereas the CSSI algorithm (in Table 6) did not converge. Similarly, the results for the case of $x_0 = 100$ show that the SCSSI algorithm (in Table 7) converged at the 12th cycle with actual root value of -0.49281 whereas the CSSI algorithm (in Table 8) did not converge. The results in Table 6 and Table 8 show the limitation of the CSSI algorithm. When the initial guess root is far above or far below the actual root, the CSSI may never converge. In essence, the CSSI is suitable for cases where the approximate value of the actual root can be estimated in advance. However, the SCSSI algorithm solved that problem by using the binary back-off mechanism.

Table 1 The result of the SCSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 1$ which gives $g(x_0) = 0$.

j	x_0	x_1	$f(x_0)$	$f(x_1)$	x_2	$f(x_2)$	x_2 -$g(x_2)$
0	1	0	-2	-1	-1	2.000E+00	1.000E+00
1	-1	0	2	-1	-0.33333	-3.320E-01	-1.000E+00
2	-0.33333333	-0.49931	-0.33196	0.014125	-0.49254	-6.430E-04	1.660E-01
3	-0.49253983	-0.49286	-0.00064	5.6E-05	-0.49284	-6.736E-09	3.215E-04
4	-0.49283556	-0.49284	-6.7E-09	5.88E-10	-0.49284	0.000E+00	3.368E-09

Table 2 The result of the CSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 1$ which gives $g(x_0) = 0$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	1	0	-2	-1	-1.00000000	2.000E+00	1.000E+00
1	0	-1	-1	2	-0.33333333	-3.320E-01	1.000E+00
2	-1	-0.333333333	2	-0.33196159	-0.42823529	-1.374E-01	-6.667E-01
3	-0.33333333	-0.42823529	-0.33196159	-0.13736212	-0.49522383	5.198E-03	9.490E-02
4	-0.42823529	-0.49522383	-0.13736212	0.00519825	-0.49278120	-1.182E-04	6.699E-02
5	-0.49522383	-0.49278120	0.00519825	-0.00011821	-0.49283551	-1.155E-07	-2.443E-03
6	-0.49278120	-0.49283551	-0.00011821	-0.00000012	-0.49283556	2.555E-12	5.431E-05

Table 3 The result of the SCSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 5$ which gives $g(x_0) = 7812$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	5	7812	15614	2.27E+23	5	1.561E+04	-7.807E+03
1	2.5	121.5703	238.1406	3.23E+12	2.5	2.381E+02	-1.191E+02
2	1.25	1.407349	0.314697	3.955109	1.236398	9.952E-02	-1.573E-01
3	1.236397915	1.286157	0.099518	0.954197	1.230604	1.183E-02	-4.976E-02
4	1.230604048	1.236521	0.011834	0.101406	1.229822	1.811E-04	-5.917E-03
5	1.229822323	1.229913	0.000181	0.001529	1.22981	4.291E-08	-9.057E-05
6	1.229810152	1.22981	4.29E-08	3.62E-07	1.22981	2.665E-15	-2.146E-08
7	1.229810149	1.22981	2.66E-15	2.31E-14	1.22981	0.000E+00	0.000E+00

Table 4 The result of the CSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 1$ which gives $g(x_0) = 0$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	5	7812	15614	2.273E+23	5	1.561E+04	-7.807E+03
1	7812	5	2.273E+23	15614	5	1.561E+04	7.807E+03
2	5	5	15614	15614	4.6666667	1.032E+04	0.000E+00
3	5	4.6666667	15614	10318.248	4.0171996	4.194E+03	3.333E-01
4	4.6666667	4.0171996	10318.248	4193.7827	3.5724713	2.071E+03	6.495E-01
5	4.0171996	3.5724713	4193.7827	2070.6539	3.1387347	9.489E+02	4.447E-01
6	3.5724713	3.1387347	2070.6539	948.87616	2.7718508	4.470E+02	4.337E-01
7	3.1387347	2.7718508	948.87616	446.99993	2.4450828	2.078E+02	3.669E-01
8	2.7718508	2.4450828	446.99993	207.7886	2.1612389	9.659E+01	3.268E-01
9	2.4450828	2.1612389	207.7886	96.587481	1.9146967	4.444E+01	2.838E-01
10	2.1612389	1.9146967	96.587481	44.442579	1.7045713	2.012E+01	2.465E-01
11	1.9146967	1.7045713	44.442579	20.120488	1.5307447	8.804E+00	2.101E-01
12	1.7045713	1.5307447	20.120488	8.8037118	1.395519	3.595E+00	1.738E-01
13	1.5307447	1.395519	8.8037118	3.5950502	1.3021854	1.271E+00	1.352E-01
14	1.395519	1.3021854	3.5950502	1.2713279	1.2511218	3.330E-01	9.333E-02
15	1.3021854	1.2511218	1.2713279	0.3330399	1.232997	4.777E-02	5.106E-02
16	1.2511218	1.232997	0.3330399	0.0477666	1.2299622	2.263E-03	1.812E-02
17	1.232997	1.2299622	0.0477666	0.002263	1.2298113	1.656E-05	3.035E-03
18	1.2299622	1.2298113	0.002263	1.656E-05	1.2298101	5.805E-09	1.509E-04
19	1.2298113	1.2298101	1.656E-05	5.805E-09	1.2298101	1.599E-14	1.112E-06

Table 5 The result of the SCSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 10$ which gives $g(x_0) = 499999.5$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	10	499999.5	999979	1.56E+34	10	1.000E+06	-5.000E+05
1	5	7812	15614	2.27E+23	5	1.561E+04	-7.807E+03
2	2.5	121.5703	238.1406	3.23E+12	2.5	2.381E+02	-1.191E+02
3	1.25	1.407349	0.314697	3.955109	1.236398	9.952E-02	-1.573E-01
4	1.236397915	1.286157	0.099518	0.954197	1.230604	1.183E-02	-4.976E-02
5	1.230604048	1.236521	0.011834	0.101406	1.229822	1.811E-04	-5.917E-03
6	1.229822323	1.229913	0.000181	0.001529	1.22981	4.291E-08	-9.057E-05
7	1.229810152	1.22981	4.29E-08	3.62E-07	1.22981	2.665E-15	-2.146E-08
8	1.229810149	1.22981	2.66E-15	2.31E-14	1.22981	0.000E+00	0.000E+00

Table 6 The result of the CSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 10$ which gives $g(x_0) = 0$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	10	499999.5	999979	1.562E+34	10	1.000E+06	-5.000E+05
1	499999.5	10	1.562E+34	999979	10	1.000E+06	5.000E+05
2	10	10	999979	999979	#DIV/0!	#DIV/0!	0.000E+00

Table 7 The result of the SCSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 1000$ which gives $g(x_0) = 5E+17$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	1000	5E+17	1E+18	1.6E+106	1000	1.000E+18	-5.000E+17
1	500	7.81E+15	1.56E+16	2.27E+95	500	1.562E+16	-7.812E+15
2	250	1.22E+14	2.44E+14	3.31E+84	250	2.441E+14	-1.221E+14
3	125	1.91E+12	3.81E+12	4.81E+73	125	3.815E+12	-1.907E+12
4	62.5	2.98E+10	5.96E+10	7.01E+62	62.5	5.960E+10	-2.980E+10
5	31.25	4.66E+08	9.31E+08	1.02E+52	31.25	9.313E+08	-4.657E+08
6	15.625	7275957	14551883	1.48E+41	15.625	1.455E+07	-7.276E+06
7	7.8125	113686.3	227357.1	2.16E+30	7.8125	2.274E+05	-1.137E+05
8	3.90625	1775.857	3543.901	3.14E+19	3.90625	3.544E+03	-1.772E+03
9	1.953125	27.25558	50.6049	4.1E+08	1.953122	5.060E+01	-2.530E+01
10	0.9765625	-0.06632	-2.08576	-0.86736	-0.80873	8.972E-01	1.043E+00
11	-0.80873077	-0.36011	0.897246	-0.2776	-0.46611	-5.752E-02	-4.486E-01
12	-0.46611231	-0.49487	-0.05752	0.004433	-0.49281	-4.550E-05	2.876E-02
13	-0.49281463	-0.49284	-4.6E-05	3.97E-06	-0.49284	-3.380E-11	2.275E-05
14	-0.49283556	-0.49284	-3.4E-11	2.95E-12	-0.49284	0.000E+00	1.690E-11

Table 8 The result of the CSSI method for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial $x_0 = 10$ which gives $g(x_0) = 5E+17$.

j	x_0	x_1	f(x_0)	f(x_1)	x_2	f(x_2)	x_2 -g(x_2)
0	1000	5E+17	1E+18	1.56E+106	1000	1.000E+18	-5.000E+17
1	5E+17	1000	1.56E+106	1E+18	1000	1.000E+18	5.000E+17
2	1000	1000	1E+18	1E+18	#DIV/0!	#DIV/0!	0.000E+00

4. CONCLUSION

A form of secant iteration scheme that combines the classical secant algorithm and a complementary root method is presented for finding the roots of nonlinear functions. The new method referred here as strict complementary root-based seeded secant iteration (SCSSI) method requires only one initial guess root value which makes it easier to implement than the classical secant iteration method. More so, the SCSSI method is suitable for arbitrary initial guess root value. In essence, it can accommodate arbitrary initial root guess value and still converge to the actual root as long as the function is defined in the range of values the arbitrary initial guess root is selected. The SCSSI used binary back-off mechanism to reduce the initial guess value iteratively until a value that is close to the actual root value is obtained. Sample numerical computations are presented to demonstrate the applicability of the new method.

REFERENCES

1. Thota, S. (2021). A numerical algorithm to find a root of non-linear equations using householder's method. *Int J Adv Appl Sci*, *10*(2), 141-148.
2. Thota, S., & Srivastav, V. K. (2018). Quadratically convergent algorithm for computing real root of non-linear transcendental equations. *BMC research notes*, *11*(1), 1-6.
3. Srivastav, V. K., Thota, S., & Kumar, M. (2019). A new trigonometrical algorithm for computing real root of non-linear transcendental equations. *International Journal of Applied and Computational Mathematics*, *5*(2), 1-8.
4. Qureshi, U. K., Bozdar, I. A., Pirzada, A., & Arain, M. B. (2019). Quadrature Rule Based Iterative Method for the Solution of Non-Linear Equations: Quadrature Rule for the Solution of Non-Linear Equations. *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, *56*(1), 39-43.
5. Moheuddin, M. M., Uddin, M. J., & Kowsher, M. ANew STUDY TO FIND OUT THE BEST COMPUTATIONAL METHOD FOR SOLVING THE NONLINEAR EQUATION.
6. González-Gaxiola, O., & Hernández-Linares, S. (2021). An Efficient Iterative Method for Solving the Elliptical Kepler's Equation. *International Journal of Applied and Computational Mathematics*, *7*(2), 1-14.
7. Qureshi, U. K., Shaikh, A. A., & SOLANGI, M. (2017). Modified Free Derivative Open Method for Solving Non-Linear Equations. *Sindh University Research Journal-SURJ (Science Series)*, *49*(4), 821-824.
8. Jeswal, S. K. (2020). *Connectionist Models for Solving Linear and Nonlinear Equations* (Doctoral dissertation).
9. Qureshi, U. K., Kalhoro, Z. A., Malookani, R. A., Dehraj, S., Siyal, S. H., & Buriro, E. A. (2020). Quadratic Convergence Iterative Algorithms of Taylor Series for Solving Non-linear Equations. *Quaid-E-Awam University Research Journal of Engineering, Science & Technology, Nawabshah.*, *18*(2), 150-156.
10. Rehman, M. A., Naseem, A., & Abdeljawad, T. (2021). Some Novel Sixth-Order Iteration Schemes for Computing Zeros of Nonlinear Scalar Equations and Their Applications in Engineering. *Journal of Function Spaces*, *2021*.
11. Monsalve, M., & Raydan, M. (2011). Newton's method and secant methods: A longstanding relationship from vectors to matrices. *Portugaliae Mathematica*, *68*(4), 431-475.
12. Kelley, C. T. (2003). *Solving nonlinear equations with Newton's method*. Society for Industrial and Applied Mathematics.
13. Birgin, E. G., & Martínez, J. M. (2020). Secant acceleration of sequential residual methods for solving large-scale nonlinear systems of equations. *arXiv preprint arXiv:2012.13251*.
14. Dieci, L., & Russell, R. D. (2020). Some aspects of invariant subspaces computation. In *Asymptotic and Computational Analysis* (pp. 565-585). CRC Press.
15. Liu, H., Zhang, Q., Xu, C., & Ye, Z. A novel FastICA algorithm based on improved secant method for Intelligent drive. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-14.
16. Ehiwario, J. C., & Aghamie, S. O. (2014). Comparative study of bisection, Newton-Raphson and secant methods of root-finding problems. *IOSR Journal of Engineering*, *4*(04), 01-07.
17. Gmati, N., & Zrelli, N. (2006). Numerical study of some iterative solvers for acoustics in unbounded domains. *Revue Africaine de*

la Recherche en Informatique et Mathématiques Appliquées, *4*, 1-23.

18. Simeon, O. (2015) Analysis Of Perturbance Coefficient-Based Seeded Secant Iteration Method. *Journal of Multidisciplinary Engineering Science and Technology* (JMEST) Vol. 2 Issue 1, January – 2015
19. Kumar, S., Kanwar, V., & Singh, S. (2012). On some modified families of multipoint iterative methods for multiple roots of nonlinear equations. *Applied Mathematics and Computation*, *218*(14), 7382-7394.
20. Petkovic, M., Neta, B., Petkovic, L., & Dzunic, J. (2012). Multipoint methods for solving nonlinear equations.
21. Oliveira, I. F., & Takahashi, R. H. (2020). An Enhancement of the Bisection Method Average Performance Preserving Minmax Optimality. *ACM Transactions on Mathematical Software (TOMS)*, *47*(1), 1-24.
22. Pourjafari, E., & Mojallali, H. (2012). Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. *Swarm and Evolutionary Computation*, *4*, 33-43.
23. Graichen, K. (2012). A fixed-point iteration scheme for real-time model predictive control. *Automatica*, *48*(7), 1300-1305.
24. Nilsrakoo, W., & Saejung, S. (2006). A new three-step fixed point iteration scheme for asymptotically nonexpansive mappings. *Applied Mathematics and Computation*, *181*(2), 1026-1034.
25. Abushammala, M., Khuri, S. A., & Sayfy, A. (2015). A novel fixed point iteration method for the solution of third order boundary value problems. *Applied Mathematics and Computation*, *271*, 131-141.
26. Khuri, S. A., & Sayfy, A. (2015). A novel fixed point scheme: Proper setting of variational iteration method for BVPs. *Applied Mathematics Letters*, *48*, 75-84.
27. Khuri, S. A., & Louhichi, I. (2021). A new fixed point iteration method for nonlinear third-order BVPs. *International Journal of Computer Mathematics*, 1-13.
28. Akewe, H. (2016). The stability of a modified Jungck-Mann hybrid fixed point iteration procedure. *International Journal of Mathematical Sciences and Optimization: Theory and Applications*, *2016*, 95-104.