Science and Technology Publishing (SCI & TECH)
ISSN: 2632-1017
Vol. 5 Issue 5, May - 2021

EVALUATION OF SOLUTION TO NONLINEAR EQUATION USING ONETIME COMPLEMENTARY ROOT-BASED SEEDED SECANT ITERATION METHOD

Stephen, Bliss Utibe-Abasi[1]
Department of Electrical/Electronic and Computer Engineering University of Uyo

Olutimo Akinwale Lewis[2]
Department of mathematics, Lagos State University, Nigeria
aolutimo@yahoo.com

Ezuruike Okafor S.F.[3]
Department of Electrical/Electronic Engineering, Imo State Polytechnic, Umuagwo, Nigeria

***Abstract*—In this paper, evaluation of solution to nonlinear equation using onetime complementary root-based seeded secant iteration method is presented. The onetime complementary root-based secant iteration relies on using one initial guess root in a complementary root form of a function to generate the two initial guess roots required by the secant method. By this method, a single initial guess root is required to apply the secant method rather than then two initial guess roots as required by the conventional secant iteration method. A numerical example for the root of a case study nonlinear function, $f(x) = x^6 - 2x - 1 = 0$ was solved using the onetime complementary root-based seeded secant algorithm implemented in Mathlab software with $\epsilon = 10^{-4} = 0.0001$. The result of the iterative solution for with initial guess root value, $x_0 = 1$ gave complementary root, $g(x_0) = 0$ and it took 5 cycles for the algorithm to converge to the actual root of -0.49283556. Generally, the results show that the closer the initial root is to the actual root, the smaller is the convergence cycle. For instance, x_0 =4 and x_0 =2 gave the same root value of 1.229810149 but it took 14 cycles for the case of x_0 =4 to converge whereas it took 9 cycles for the case of x_0 =2 to converge. Similarly, x_0 =1 and x_0 =0.75 gave the same root value of -0.49283556 but it took 5 cycles for the case of x_0 =1 to converge whereas it took 2 cycles for the case of x_0 =0.75 to converge.**

Keywords— Seeded Secant Iteration Method, Convergence Cycle, Nonlinear Equation, Onetime Complementary Root-Based Method, Mathlab Software, Initial Guess Roots

1. INTRODUCTION

Iterative root finding methods are numerous and they have been widely used in the solutions to nonlinear and transcendental equations [1,2,3,4,5,6,7,8,9]. Each of the different iterative methods has its merits and demerits [10,11,12,131,14,15,16,17,18]. However, researchers continue to develop more methods that that are targeted to different kinds of equations or methods that satisfy different performance measures. In this paper, complementary root-based seeded secant approach is presented. The complementary root is used to use one initial guess root to generate the second initial root required for secant iteration. Hence, in the complementary root-based seeded secant, the classical secant method [19,20,21,22,23] is adapted to operate with a single initial guess root. Specifically, the complementary root mechanism is implemented first; it takes a single initial guess root value, x_0 and generates the second guess root, x_1 which in this case is a complementary root of the guess root, $g(x_0)$, that is $x_1 = g(x_0)$. The two roots, x_0 and x_1 are then used iteratively in the classical secant iteration method to determine the actual root of the function , f(x). In practice, the complementary root mechanism is adapted from the fixed point iteration method [24,25,26,27,28]. Basically, the onetime complementary root-based seeded secant iteration method is a form of secant iteration method that uses algorithm similar to the fixed point iteration to find the initial two guess roots required by the secant method. After the application of the fixed-point-like mechanism once, the secant method is then repeatedly applied until the actual root is determined. The convergence cycle performance of the iteration scheme is examined in this paper by using some numerical examples. Also, the relevant mathematical procedure and algorithm for the iteration scheme are presented.

2 . METHODOLOGY: DEVELOPMENT OF THE ALGORITHM FOR THE ONETIME COMPLEMENTARY ROOT-BASED SEEDED SECANT

The onetime complementary root-based secant iteration relies on using one initial guess root in a complementary root form of a function to generate the two initial guess roots required by the secant method. By this method, a single initial guess root is required to apply the secant method rather than then two initial guess roots as required by the conventional secant iteration method. The method is explained in this paper using a nonlinear function of x. Given a nonlinear function , f(x) where;

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

Express the function in the complementary root form as follows;

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

Hence

$$x = \frac{x^6-1}{2} \quad (3)$$

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

The complementary root form of f(x) is denoted as $\hat{f}(x)$such that

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

A single value of x is selected to compute g(x) and $\hat{f}(x)$. If $\hat{f}(x) \neq 0$ then the root of f(x) has not been found. At this point, if the present iteration cycle number is j, then the secant iteration formula for the next guess root, x_{j+1} is computed from the values of x and g(x)as follows:

$$x_{j-1} = x \quad (6)$$

$$x_j = g(x) \quad (7)$$

$$f(x_{j-1}) = (x_{j-1})^6 - 2(x_{j-1}) - 1 \quad (8)$$

$$f(x_j) = (x_j)^6 - 2(x_j) - 1 \quad (9)$$

$$x_{j+1} = \frac{(x_{j-1})f(x_j) - (x_j)f(x_{j-1})}{f(x_j) - f(x_{j-1})} \quad (10)$$

Then compute the following;

$$f(x_{j+1}) = (x_{j+1})^6 - 2(x_{j+1}) - 1 \quad (11)$$

$$\hat{f}(x_{j+1}) = x_{j+1} - \left(\frac{(x_{j-1})^6-1}{2}\right) \quad (12)$$

Assuming a tolerance error of ϵ is specified, then if $[f(x_{j+1}) \le \epsilon]$ the root is x_{j+1} otherwise j =j+1 , $x_{j-1} = x_j$, $x_j = x_{j+1}$ and the secant iteration is repeated until $[f(x_{j+1}) \le \epsilon]$. Also, $[\hat{f}(x_{j+1}) \le \epsilon]$ can be used to determine the termination point for the iteration.

A numerical example for finding the root of the function $f(x) = x^6 - 2x - 1 = 0$ with the value of $\epsilon = 10^{-4} = 0.0001$ is presented next. Let the initial guess root value, $x_0 = 1$ and $g(x_0) = \frac{1^6-1}{2} = 0$. So, for j =1

$$x_{j-1} = x_0 = 1$$

$$x_j = x_1 = g(x_0) = 0$$

$$f(x_0) = f(1) = 1^6 - 2(1) - 1 = -2$$

$$f(x_1) = f(0) = 0^6 - 2(0) - 1 = -1$$

$$x_{j+1} = x_2 = \frac{[(x_0)f(x_1)] - [(x_1)f(x_0)]}{f(x_1) - f(x_0)}$$

$$= \frac{[(1)(-1)] - [(0)(-2)]}{-1-(-2)} = -1$$

Then compute,

$$f(x_2) = f(-1) = (-1)^6 - 2(-1) - 1$$

$$= 1 + 2 - 1 = 2$$

Since $f(x_2) = 2 > 0.0001$, the secant iteration is repeated with the following values; j =j+1, $x_{j-1} = x_j$, $x_j = x_{j+1}$ until $f(x_2) \le \epsilon$. The detailed algorithm for iterative computation of the actual root, x_{ac} of the function f(x) based on the onetime complementary root-based seeded secant approach is given as follows;

Step 1 Input initial guess root, x_0
Step 2 Compute the complementary root of x_0, $x_1 = g(x_0)$
Step 3 Initialize cycle counter to 1; j =1
Step 4 Compute $f(x_{0(j)}) = (x_{0(j)})^6 - 2(x_{0(j)}) - 1$
Step 5 Compute $f(x_{1(j)}) = (x_{1(j)})^6 - 2(x_{1(j)}) - 1$
Step 6 Compute by secant approach the next guess root, x_2 where

$$\boldsymbol{x}_{2(j)} = \frac{[(\boldsymbol{x}_{0(j)})\mathbf{f}(\boldsymbol{x}_{1(j)})] - [(\boldsymbol{x}_{1(j)})\mathbf{f}(\boldsymbol{x}_{0(j)})]}{\mathbf{f}(\boldsymbol{x}_{1(j)}) - \mathbf{f}(\boldsymbol{x}_{0(j)})}$$

Step 7 Compute $\boldsymbol{f}(\boldsymbol{x}_{2(j)})$

Step 8 If $(|\boldsymbol{f}(\boldsymbol{x}_{2(j)})| \leq \in)$ then $\boldsymbol{x}_{ac} = \boldsymbol{x}_{2(j)}$; goto step 10

Step 9 If $(|\boldsymbol{f}(\boldsymbol{x}_{2(j)})| > \in)$ then

$\mathbf{j} = \mathbf{1} + \mathbf{1}$

$\boldsymbol{x}_{0(j)} = \boldsymbol{x}_{1(j-1)}$

$\boldsymbol{x}_{1(j)} = \boldsymbol{x}_{2(j-1)}$

Goto step 4

endif

Step 10 Output $\boldsymbol{x}_{ac}$

Step 11 End

3. RESULTS AND DISCUSSION

The numerical example for the root of the nonlinear function $f(x) = x^6 - 2x - 1 = 0$ is solved using the onetime complementary root-based seeded secant algorithm implemented in Mathlab software with $\epsilon = 10^{-4} = 0.0001$. The result of the iterative solution for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial guess root value, $x_0 = 1$ which gives $g(x_0) = 0$ is presented in Table 1. The results showed that it took 5 cycles for $f(x_{2(j)}) < \epsilon = 10^{-4}$. The results for $x_0 = 4$ in Table 2 showed that it took 15 cycles for $f(x_{2(j)}) < \epsilon = 10^{-4}$.

The simulation results for other values of x_0 are shown in Table 3, Table 4, and Table 5 while Table 6 shows the summary of the results for all the five test cases in Table 1, Table 2, Table 3, Table 4, and Table 5. The results in Table 6 show that different initial guess roots gave different convergence cycles and in some cases different actual root of the function. However, the results show that the closer the initial root is to the actual root, the smaller is the convergence cycle. For instance, $x_0 = 4$ and $x_0 = 2$ gave the same root value of 1.229810149 but it took 14 cycles for the case of $x_0 = 4$ to converge whereas it took 9 cycles for the case of $x_0 = 2$ to converge. Similarly, $x_0 = 1$ and $x_0 = 0.75$ gave the same root value of -0.49283556 but it took 5 cycles for the case of $x_0 = 1$ to converge whereas it took 2 cycles for the case of $x_0 = 0.75$ to converge.

Table 1 The result of the iterative solution for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial guess root value, $x_0 = 1$ which gives $g(x_0) = 0$.

j	$x_{0(j)}$	$x_{1(j)}$	$f(x_{0(j)})$	$f(x_{1(j)})$	$x_{2(j)}$	$f(x_{2(j)})$	$x_{2(j)}$ - $g(x_{2(j)})$
0	1	0	-2	-1	-1	2.000E+00	1.000E+00
1	0	-1	-1	2	-0.333333	-3.320E-01	1.000E+00
2	-1	-0.333333	2	-0.331962	-0.428235	-1.374E-01	-6.667E-01
3	-0.333333	-0.428235	-0.331962	-0.137362	-0.495224	5.198E-03	9.490E-02
4	-0.428235	-0.495224	-0.137362	0.0051982	-0.492781	-1.182E-04	6.699E-02
5	-0.495224	-0.492781	0.0051982	-0.000118	-0.492836	-1.155E-07	-2.443E-03
6	-0.492781	-0.492836	-0.000118	-1.16E-07	-0.492836	2.555E-12	5.431E-05
7	-0.492836	-0.492836	-1.16E-07	2.555E-12	-0.492836	0.000E+00	5.312E-08

Table 2 The result of the iterative solution for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial guess root value, $x_0 = 4$ which gives $g(x_0) = 2047.5$.

j	$x_{0(j)}$	$x_{1(j)}$	$f(x_{0(j)})$	$f(x_{1(j)})$	$x_{2(j)}$	$f(x_{2(j)})$	$x_{2(j)}$ - $g(x_{2(j)})$
0	4	2047.5	4087	7.368E+19	4	4.087E+03	-2.044E+03
1	2047.5	4	7.368E+19	4087	4	4.087E+03	2.044E+03
2	4	4	4087	4087	3.332899	1.363E+03	1.137E-13
3	4	3.332899	4087	1363.0043	2.9991021	7.207E+02	6.671E-01
4	3.332899	2.9991021	1363.0043	720.69358	2.624571	3.206E+02	3.338E-01
5	2.9991021	2.624571	720.69358	320.60194	2.3244512	1.521E+02	3.745E-01
6	2.624571	2.3244512	320.60194	152.08413	2.0535989	6.990E+01	3.001E-01
7	2.3244512	2.0535989	152.08413	69.898408	1.8232408	3.209E+01	2.709E-01
8	2.0535989	1.8232408	69.898408	32.087172	1.6277555	1.435E+01	2.304E-01
9	1.8232408	1.6277555	32.087172	14.345432	1.469692	6.138E+00	1.955E-01
10	1.6277555	1.469692	14.345432	6.1382353	1.3514749	2.390E+00	1.581E-01
11	1.469692	1.3514749	6.1382353	2.390285	1.276081	7.657E-01	1.182E-01
12	1.3514749	1.276081	2.390285	0.7657074	1.2405459	1.637E-01	7.539E-02
13	1.276081	1.2405459	0.7657074	0.1637353	1.2308803	1.596E-02	3.554E-02
14	1.2405459	1.2308803	0.1637353	0.0159626	1.2298363	3.886E-04	9.666E-03
15	1.2308803	1.2298363	0.0159626	0.0003886	1.2298102	9.578E-07	1.044E-03
16	1.2298363	1.2298102	0.0003886	9.578E-07	1.2298101	5.768E-11	2.605E-05

Table 3 The result of the iterative solution for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial guess root value, $x_0 = 2$ which gives $g(x_0) = 31.5$.

j	$x_{0(j)}$	$x_{1(j)}$	$f(x_{0(j)})$	$f(x_{1(j)})$	$x_{2(j)}$	$f(x_{2(j)})$	$x_{2(j)}$ - $g(x_{2(j)})$
0	2	31.5	59	976929658	1.9999982	5.900E+01	-2.950E+01
1	31.5	1.9999982	976929658	58.999661	1.9999964	5.900E+01	2.950E+01
2	1.9999982	1.9999964	58.999661	58.999323	1.6894716	1.888E+01	1.782E-06
3	1.9999964	1.6894716	58.999323	18.875468	1.5433914	9.429E+00	3.105E-01
4	1.6894716	1.5433914	18.875468	9.4294724	1.3975667	3.656E+00	1.461E-01
5	1.5433914	1.3975667	9.4294724	3.6562205	1.3052154	1.334E+00	1.458E-01
6	1.3975667	1.3052154	3.6562205	1.3337356	1.2521807	3.504E-01	9.235E-02
7	1.3052154	1.2521807	1.3337356	0.3504403	1.2332795	5.203E-02	5.303E-02
8	1.2521807	1.2332795	0.3504403	0.0520341	1.2299836	2.582E-03	1.890E-02
9	1.2332795	1.2299836	0.0520341	0.002582	1.2298115	2.056E-05	3.296E-03
10	1.2299836	1.2298115	0.002582	2.056E-05	1.2298101	8.222E-09	1.721E-04
11	1.2298115	1.2298101	2.056E-05	8.222E-09	1.2298101	2.576E-14	1.381E-06

Table 4 The result of the iterative solution for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial guess root value, $x_0 = 0.75$ which gives $g(x_0) = \frac{1^6 - 1}{2}$ = -0.411011

j	$x_{0(j)}$	$x_{1(j)}$	$f(x_{0(j)})$	$f(x_{1(j)})$	$x_{2(j)}$	$f(x_{2(j)})$	$x_{2(j)}$ - $g(x_{2(j)})$
0	0.75	-0.411011	-2.322021	-0.173158	-0.504566	2.563E-02	1.161E+00
1	-0.411011	-0.504566	-0.173158	0.0256334	-0.492503	-7.239E-04	9.356E-02
2	-0.504566	-0.492503	0.0256334	-0.000724	-0.492834	-3.548E-06	-1.206E-02
3	-0.492503	-0.492834	-0.000724	-3.55E-06	-0.492836	4.804E-10	3.313E-04
4	-0.492834	-0.492836	-3.55E-06	4.804E-10	-0.492836	0.000E+00	1.632E-06
5	-0.492836	-0.492836	4.804E-10	0	-0.492836	0.000E+00	-2.209E-10

j	$x_{0(j)}$	$x_{1(j)}$	$f(x_{0(j)})$	$f(x_{1(j)})$	$x_{2(j)}$	$f(x_{2(j)})$	$x_{2(j)}$ - $g(x_{2(j)})$
1	8	131071.5	262127	5.07E+30	8	262127	-1.311E+05
2	131071.5	8	5.07E+30	262127	8	262127	1.311E+05
3	8	8	262127	262127	#DIV/0!	#DIV/0!	0.000E+00

Table 5 The result of the iterative solution for the root of $f(x) = x^6 - 2x - 1 = 0$ with initial guess root value, x_0 = -1 which gives $g(x_0) = 0$

j	Xj-1	Xj	f(Xj-1)	f(Xj)	Xj+1	f(Xj+1)	Xj+1-g(Xj+1)
0	-1	0	2	-1	-0.333333	-3.320E-01	-1.000E+00
1	0	-0.333333	-1	-0.331962	-0.498973	1.338E-02	3.333E-01
2	-0.333333	-0.498973	-0.331962	0.0133801	-0.492556	-6.085E-04	1.656E-01
3	-0.498973	-0.492556	0.0133801	-0.000609	-0.492835	-1.541E-06	-6.418E-03
4	-0.492556	-0.492835	-0.000609	-1.54E-06	-0.492836	1.754E-10	2.792E-04
5	-0.492835	-0.492836	-1.54E-06	1.754E-10	-0.492836	0.000E+00	7.087E-07

Table 6 The summary of the results for all the five test cases in Table 1, Table 2, Table 3, Table 4, and Table 5

S/N	Result Table Number	x_0	$g(x_0)$	Convergence Cycle	$f(x_{2(j)})$	Actual Root, x_{ac}
1	Table 1	1	0	5	-1.155E-07	-0.49283556
2	Table 2	4	2047.5	14	9.578E-07	1.229810149
3	Table 3	2	31.5.	9	2.056E-05	1.229810149
4	Table 4	0.75	-0.411011	2	-3.548E-06	-0.49283556
5	Table 5	-1	0	3	-0.49283556	-0.49283556

4. CONCLUSION

The solution of nonlinear equation using a onetime seeded secant method is presented using a numerical example. The detailed algorithm for the method is presented. Mathlab software was used to implement the algorithm for the sample nonlinear function. The solution was evaluated with different single initial guess root values and the convergence cycles were compare. The results show that different initial guess roots gave different convergence cycles and in some cases different actual root of the function. However, the results show that the closer the initial root is to the actual root, the smaller is the convergence cycle.

REFERENCES

1. 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.
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. Boyd, J. P. (2014). *Solving Transcendental Equations: The Chebyshev Polynomial Proxy and Other Numerical Rootfinders, Perturbation Series, and Oracles*. Society for Industrial and Applied Mathematics.
4. Gemechu, T., & Thota, S. (2020). On New Root Finding Algorithms for Solving Nonlinear Transcendental Equations. *International Journal of Chemistry, Mathematics and Physics*, *4*(2), 18-24.
5. Singh, R., & Kumar, J. (2014). An efficient numerical technique for the solution of nonlinear singular boundary value problems. *Computer Physics Communications*, *185*(4), 1282-1289.
6. Raja, M. A. Z., Sabir, Z., Mehmood, N., Al-Aidarous, E. S., & Khan, J. A. (2015). Design of stochastic solvers based on genetic algorithms for solving nonlinear equations. *Neural Computing and Applications*, *26*(1), 1-23.
7. Basto, M., Abreu, T., Semiao, V., & Calheiros, F. L. (2017). Convergence and dynamics of structurally identical root finding methods. *Applied Numerical Mathematics*, *120*, 257-269.
8. 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.
9. Mahesh, G., Swapna, G., & Venkateshwarlu, K. (2020). An iterative method for solving non-linear transcendental equations. *J. Math. Comput. Sci.*, *10*(5), 1633-1642.
10. Soleymani, F., Lotfi, T., & Bakhtiari, P. (2014). A multi-step class of iterative methods for nonlinear systems. *Optimization Letters*, *8*(3), 1001-1015.
11. Patade, J., & Bhalekar, S. (2015). Approximate analytical solutions of Newell-Whitehead-Segel equation using a new iterative method. *World Journal of Modelling and Simulation*, *11*(2), 94-103.
12. Fazli, A., & Arezoo, B. (2012). A comparison of numerical iteration based algorithms in blank optimization. *Finite elements in analysis and design*, *50*, 207-216.
13. Minka, T. P. (2003). A comparison of numerical optimizers for logistic regression. *Unpublished draft*, 1-18.
14. Axelsson, O., Farouq, S., & Neytcheva, M. (2016). Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems. *Numerical Algorithms*, *73*(3), 631-663.
15. Varona, J. L. (2002). Graphic and numerical comparison between iterative methods. *The Mathematical Intelligencer*, *24*(1), 37-46.
16. Boyle, M., Brown, D. A., Kidder, L. E., Mroue, A. H., Pfeiffer, H. P., Scheel, M. A., ... & Teukolsky, S. A. (2007). High-accuracy comparison of numerical relativity simulations with post-Newtonian expansions. *Physical Review D*, *76*(12), 124038.
17. Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P. J., & Vachaud, G. (1977). A comparison of numerical simulation models for one-dimensional infiltration. *Soil Science Society of America Journal*, *41*(2), 285-294.
18. Biegel, B. A., & Plummer, J. D. (1996). Comparison of self-consistency iteration options for the Wigner function method of quantum device simulation. *Physical Review B*, *54*(11), 8070.
19. Amat, S., & Busquier, S. (2003). A modified secant method for semismooth equations. *Applied mathematics letters*, *16*(6), 877-881.
20. Kahya, E., & Chen, J. (2007). A modified Secant method for unconstrained optimization. *Applied mathematics and computation*, *186*(2), 1000-1004.
21. Kahya, E. (2006). Modified Secant-type methods for unconstrained optimization. *Applied mathematics and computation*, *181*(2), 1349-1356.
22. Crittin, F., & Bierlaire, M. (2003). A generalization of secant methods for solving nonlinear systems of equations. In *Swiss Transport Research Conference* (No. CONF).
23. Grau-Sánchez, M., & Noguera, M. (2012). A technique to choose the most efficient method between secant method and some variants. *Applied Mathematics and Computation*, *218*(11), 6415-6426.

24. Martinez-Yanes, C., & Xu, H. K. (2006). Strong convergence of the CQ method for fixed point iteration processes. *Nonlinear Analysis: Theory, Methods & Applications*, *64*(11), 2400-2411.
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. Popova, E. D. (2004, December). Generalization of a Parametric Fixed-Point Iteration. In *PAMM: Proceedings in Applied Mathematics and Mechanics* (Vol. 4, No. 1, pp. 680-681). Berlin: WILEY-VCH Verlag.
27. Tian, Z., Zarepisheh, M., Jia, X., & Jiang, S. B. (2013). The fixed-point iteration method for IMRT optimization with truncated dose deposition coefficient matrix. *arXiv preprint arXiv:1303.3504*.
28. Sung, W., & Lee, J. (2018). Improved capacity estimation technique for the battery management systems of electric vehicles using the fixed-point iteration method. *Computers & Chemical Engineering*, *117*, 283-290.