

Quadratic Convergence Iterative Algorithms of Taylor Series for Solving Non-linear Equations

Umair Khalid Qureshi^{1,*}, Zubair Ahmed kalhoro², Rajab Ali Malookani³, Sanaullah Dehraj³, Shahid Hussain Siyal⁴, Ehsan Ali Buriro⁵

¹Department of Business Administration, SBBU, Sanghar, Pakistan

²Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Pakistan

³Department of Mathematics and Statistics, QUEST, Nawabshah, Pakistan

⁴Department of Energy & Environment Engineering, QUEST, Nawabshah, Pakistan

⁵Department of Electronic Engineering, QUEST, Nawabshah, Pakistan

*Corresponding author: umair.khalid_sng@sbbusba.edu.pk

Abstract

Solving the root of algebraic and transcendental nonlinear equation $f'(x) = 0$ is a classical problem which has many interesting applications in computational mathematics and various branches of science and engineering. This paper examines the quadratic convergence iterative algorithms for solving a single root nonlinear equation which depends on the Taylor's series and backward difference method. It is shown that the proposed iterative algorithms converge quadratically. In order to justify the results and graphs of quadratic convergence iterative algorithms, C++/MATLAB and EXCELL are used. The efficiency of the proposed iterative algorithms in comparison with Newton Raphson method and Steffensen method is illustrated via examples. Newton Raphson method fails if $f'(x) = 0$, whereas Steffensen method fails if the initial guess is not close enough to the actual solution. Furthermore, there are several other numerical methods which contain drawbacks and possess large number of evolution; however, the developed iterated algorithms are good in these conditions. It is found out that the quadratic convergence iterative algorithms are good achievement in the field of research for computing a single root of nonlinear equations.

Keywords—Nonlinear equations, Taylor series, Backward difference, Second order Convergence, Newton method, Steffensen method.



1 Introduction

THE estimation of a single root of nonlinear equations leads to wide range of applications in the field of numerical analysis, which emerges in applied sciences and engineering, for instance: distance, rate, time problems, population change and trajectory of a ball [1][2][3]. We consider the following nonlinear equation.

$$f(x) = 0 \tag{1.1}$$

To estimate the roots of Equation 1.1, several approaches have been established, see for instance [4][5].

Among them, the Taylor series is an important and basic concept:

$$f(x) = f(x_0) + hf'(x_0) + \frac{h^2}{2} f''(x_0) + \dots \tag{1.2}$$

By using Taylor series, one of the most important technique Newton Raphson's method has been developed [6][7][8] which is given as follows.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{1.3}$$

By using Newton forward difference, we get Steffensen method [9]:

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n + f(x_n)) + f(x_n)} \tag{1.4}$$

Moreover, to improve the efficiency and order of convergence, several approaches have been developed, see for instance [10] [11], given as follows.

$$x_{n+1} = y_n - \frac{2f(y_n)}{3f'(y_n) - f'(x_n)} \quad (1.5)$$

where, $y_n = x_n - \frac{fx_n}{f_n}$ and

$$x_{n+1} = y_n - \frac{f'(y_n) \pm \sqrt{f'(y_n)^2 - 2f''(y_n)f(y_n)}}{f''(y_n)} \quad (1.6)$$

where, $y_n = x_n - \frac{fx_n}{f'_n}$

In addition, this study suggests Algorithm 3.1 and Algorithm 3.2 of Taylor series and backward difference method for computing a root of nonlinear equations. The proposed algorithms have second order of convergence. Some illustrations are given to express the applicability of the proposed iterative algorithms. Quadratic convergence iterative algorithms are compared with Newton Raphson method and Steffensen method. In literature, it is observed that there are various numerical techniques which have crucial weakness, for example, if the initial guess is chosen far from the actual solution. Additionally, such techniques also take large number of evolutions.

This study aims to work on these types of conditions and ascertain that the proposed iterated algorithms are decent attainment in these conditions in comparison with Newton Raphson method and Steffensen method for estimating a root of nonlinear equations.

2 Iterated Method

We consider the following nonlinear equation.

$$f(x) = 0 \quad (2.1)$$

Assuming that x is zero if $f(x) = 0$. (a, b) is an interval from which the condition of $f(a)f(b) < 0$ is selected. The initial guess can be chosen if $|f(a)| < |f(b)|$. According to this condition, the initial guess would be better if $x_0 = b$ and x_o is an initial guess sufficiently nearest to the actual solution. Using the Taylor's series about the point x_o for Equation 2.1, we have:

$$f(x_r) = f(x_o) + hf'(x_o) + \frac{h^2}{2} f''(x_o) + \dots \quad (2.2)$$

To generalize Equation 2.2 with $f(x_r) = 0$ and taking only three terms, we obtain:

$$f(x_n) + hf'(x_n) + \frac{h^2}{2} f''(x_n) = 0 \quad (2.3)$$

By using backward difference operator, we obtain

$$f''(x_n) = \frac{f'(x_n) - f'(x_{n+1})}{h} \quad (2.4)$$

By substituting Equation 2.3 into equation (2.4), we get

$$f(x_n) + hf'(x_n) + \frac{h^2}{2} \left(\frac{f'(x_n) - f'(x_{n+1})}{h} \right) = 0$$

or

$$f(x_n) + \frac{h}{2} (2f'(x_n) + f'(x_n) - f'(x_{n+1})) = 0 \quad (2.5)$$

where $h = x - x_n$, then Equation 2.5 becomes

$$2f(x_n) + (x - x_n)(3f'(x_n) - f'(x_{n+1})) = 0$$

This implies that

$$x = x_n - \frac{2f(x_n)}{3f'(x_n) - f'(x_{n+1})} \quad (2.6)$$

This equality proposes the iterative methods mentioned below in order to solve the nonlinear equations under certain conditions.

3 New Iterative Methods

Suppose $f : X \rightarrow R$ is a scalar function of Taylor series expansion, we get Equation 2.6.

$$x = x_n - \frac{2f(x_n)}{3f'(x_n) - f'(x_{n+1})}$$

3.1 Algorithm 3.1

For certain x_0 , the following iterative method is considered to compute the approximate solution x_{n+1} ,

$$x = x_n - \frac{2f(x_n)}{3f'(x_n) - f'(x_{n+1})}$$

where, $x_{n+1} = x_{n+h}$ and $h = f(x_n)$, hence $x_{n+1} = x_n + f(x_n)$

Finally, we get

$$x_{n+1} = x_n - \frac{2f(x_n)}{3f'(x_n) - f'(x_n + f(x_n))} \quad (3.1)$$

3.2 Algorithm 3.2

For a given x_0 , the iterative schemes given below are used to approximate the solution x_{n+1} :

$$x = x_n - \frac{2f(x_n)}{3f'(x_n) - f'(x_{n+1})}$$

By Bisection method we have $x_{n+1} = \frac{x_n + y_n}{2}$ where $y_n = x_n - \frac{f(x_n)}{f'(x_n)}$, thus, we have $x_{n+1} = x_n - \frac{f(x_n)}{2f'(x_n)}$ By using above condition in Equation 2.6, we get

$$x_{n+1} = x_n - \frac{2f(x_n)}{3f'(x_n) - f'\left(x_n - \frac{f(x_n)}{2f'(x_n)}\right)} \quad (3.2)$$

Equation 3.2 is Algorithm 3.2 which needs three functional evaluations to solve the nonlinear equations. Hence, Algorithm 3.1 and Algorithm 3.2 are proposed iterated methods for computing a root of nonlinear equations.

4 Rate of Convergence

In this section, the order of convergence for Algorithm 3.1 and Algorithm 3.2 will be proved as quadratic with the help of theorems.

4.1 Theorem 4.1

Assume x is a root of $f(x) = 0$. If $f(x)$ is neighborhood of x , then the convergence of Algorithm 3.1 is of order two.

4.1.1 Proof of the concept

To examine the convergence of Algorithm 3.1, we suppose x is a root of $f(x) = 0$ and e_n be the error at the n^{th} iteration, then $e_n = x_n - a$. By expanding $f(x_n)$, $f'(x_n)$ and $f'(x_n + f(x_n))$ from Taylor's Series about ' a ', we obtain

$$f(x_n) = f^{(a)}(e_n + ce_n^2) \tag{4.1}$$

$$f'(x_n) = f'(a)(1 + 2ce_n + 3ce_n^2) \tag{4.2}$$

$$f(x_n + f(x_n)) = f^{(a)}[(e_n + f(x_n)) + (e_n + f(x_n))^2 c] \tag{4.3}$$

By putting Equation 4.1 and Equation 4.2 into Equation 4.3, we get

$$\begin{aligned} f'(x_n + f(x_n)) &= f'(a)(1 + f'(a)(1 + 2ce_n)) [1 + 2ce_n(1 + f'(a))] \\ f'(x_n + f(x_n)) &= f'(a) [1 + f'(a) + 2ce_n(6ce_n f'(a) + 2ce_n f'^2(a))] \end{aligned} \tag{4.4}$$

By using Equation 4.1, Equation 4.2 and Equation 4.4 in Algorithm 3.1, we get

$$\begin{aligned} e_{n+1} &= e_n - (2e_n f'(a) + ce_n) / f'(a) [3 + 6c_2 e_n - 1 - f'(a) - 2ce_n - 6ce_n f'(a) - 2ce_n f'^2(a)] \\ e_{n+1} &= e_n - \frac{2e_n f'(a)(1+ce_n)}{f'(a)[2-f'(a)+4ce_n-6ce_n f'(a)-2ce_n f'^2(a)]} \\ e_{n+1} &= e_n - \frac{e_n(1+ce_n)}{[1-0.5f'(a)+2ce_n-3ce_n f'(a)-ce_n f'^2(a)]} \\ e_{n+1} &= e_n - e_n(1 + ce_n) [1 - 0.5f'(a) + 2ce_n - 3ce_n f'(a) - ce_n f'^2(a)]^{-1} \end{aligned}$$

$$\begin{aligned} e_{n+1} &= e_n - e_n(1 + ce_n)(1 + 0.5f'(a) - 2ce_n + 3ce_n f'(a) + ce_n f'^2(a)) \\ e_{n+1} &= e_n - e_n(1 + 0.5f'(a) - 2ce_n + 3ce_n f'(a) + ce_n f'^2(a) + ce_n + 0.5ce_n f'(a)) \\ e_{n+1} &= e_n - e_n - 0.5e_n f'(a) - e_n^2(c - 3.5cf'(a) - cf'^2(a)) \end{aligned}$$

$$e_{n+1} = -0.5e_n f'(a) - e_n^2(c - 3.5cf'(a) - cf'^2(a)) \tag{4.5}$$

By putting equation (2.1) in equation (4.5) then plugging the resulting equation in equation (4.5) we obtain;

$$\begin{aligned} e_{n+1} &= 0.5 e_n^2 f''(a) - e_n^2(c - 3.5cf'(a) - cf'^2(a)) \\ e_{n+1} &= e_n^2 [0.5f''(a) - c + 3.5cf'(a) + cf'^2(a)] + o(e_n^3) \end{aligned} \tag{4.6}$$

Equation 4.6 shows that the Algorithm 3.1 converges quadratically.

4.2 Theorem 4.2.

Assume x is a root of $f(x) = 0$. If $f(x)$ is neighborhood of x , then the convergence of Algorithm 3.1 is of order two.

4.2.1 Proof of the Concept

To investigate the convergence of Algorithm 3.1, suppose x is a root of $f(x) = 0$, and e_n be the error at n^{th} iteration, then $e_n = x_n - a$, then by expanding $f(x_n)$, $f'(x_n)$, $f'(x_n + f(x_n))$, $f(x_n - \frac{f(x_n)}{2f'(x_n)})$ and $f'(x_n - \frac{f(x_n)}{2f'(x_n)})$ from Taylor's Series about a point a , we have

$$f(x_n) = f'(a)(e_n + ce_n^2) \tag{4.7}$$

$$f'(x_n) = f'(a)(1 + 2ce_n + 3ce_n^2) \tag{4.8}$$

From equation (4.7) and equation (4.8), we have

$$\frac{f(x_n)}{f'(x_n)} = e_n - ce_n^2 \tag{4.9}$$

From equation (4.9), we get

$$x_n - \frac{f(x_n)}{f'(x_n)} = ce_n^2 \tag{4.10}$$

Now, by expanding $f(x_n - \frac{f(x_n)}{f'(x_n)})$, $f'(x_n - \frac{f(x_n)}{f'(x_n)})$ and $f'(x_n - \frac{f(x_n)}{2f'(x_n)})$ and again using Taylor's series about a with the help of Equation 4.10, we have

$$\begin{aligned} f(x_n - \frac{f(x_n)}{f'(x_n)}) &= f'(a) [ce_n^2 + 2ce_n^3] \\ f'(x_n - \frac{f(x_n)}{f'(x_n)}) &= f'(a) [1 + 4ce_n^3] \end{aligned} \tag{4.11}$$

$$f' \left(x_n - \frac{f(x_n)}{2f'(x_n)} \right) = f'(a) \left[1 + ce_n + \frac{3}{4}ce_n^2 \right] \quad (4.12)$$

Putting Equation 4.7, Equation 4.8 and Equation 4.12 in Algorithm 3.2, we have

$$\begin{aligned} e_{n+1} &= e_n - \frac{2f'(a)(e_n+ce_n^2)}{f'(a)[3+6ce_n-1-ce_n-\frac{3}{4}ce_n^2]} \\ e_{n+1} &= e_n - \frac{e_n(1+ce_n)}{[1+2ce_n-\frac{3}{2}ce_n^2]} \\ e_{n+1} &= e_n - e_n(1 + ce_n) [1 + 2ce_n - \frac{3}{2}ce_n^2]^{-1} \\ e_{n+1} &= e_n - e_n(1 + ce_n)[1 - 2ce_n] \\ e_{n+1} &= e_n - e_n(1 + ce_n - 2ce_n) \\ e_{n+1} &= e_n - e_n(1 - ce_n) \\ e_{n+1} &= ce_n^2 + o(e_n^3) \end{aligned} \quad (4.13)$$

Equation 4.13 shows that the Algorithm 3.2 is quadratically convergent. Hence, it is proved that the convergence order for Algorithm 3.1 and Algorithm 3.2 is 2.

5 Numerical Results

This section discusses the following nonlinear equations to demonstrate the efficiency of proposed Algorithm 3.1 and Algorithm 3.2 by equating with the Newton Raphson method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

and Steffensen method

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n+f(x_n))+f(x_n)}$$

Following tables and graphs depict the efficiency of proposed Algorithms.

The results are obtained with the help of C++/MATLAB and EXCEL. All tables and Figure 1-4 exhibit the numerical comparisons of the proposed Algorithm 3.1 and 3.2 with Newton Raphson method and Steffensen method.

6 Conclusion

Root calculating problem of algebraic and transcendental nonlinear equations occurs frequently in applied mathematics and scientific computing. In this study, the new methods are established for solving algebraic and transcendental nonlinear equations. The proposed methods are based on the Taylor’s series and backward difference method. It turned out that the proposed numerical techniques are quadratically convergent. The efficiency of the proposed methods is illustrated by a few examples of nonlinear equations and is compared with the existing numerical methods such as

Newton Raphson method and Steffensen method. The well-known software C++/MATLAB and EXCEL are used to compute the numerical results. The numerical outcomes have shown that the proposed algorithms perform well and are easily applicable to the nonlinear equations than the existing numerical techniques..

References

- [1] C. N. Iwetan, I. A. Fuwape, M. S. Olajide, and R. A. Adenodi, "Comparative Study of the Bisection and Newton Methods in solving for Zero and Extremes of a Single-Variable Function". J. of NAMP Vol.21, pp.173-176, 2012.
- [2] N. D. Biswa, "Lecture Notes on Numerical Solution of root Finding Problems", 2012.
- [3] Golbabai, A., M. Javidi, "A Third-Order Newton Type Method for Nonlinear Equations Based on Modified Homotopy Perturbation Method", Appl. Math. And Comput., Vol. 191, pp.199–205, 2007.
- [4] A. A. Mallah, A. A. Shaikh, S. Qureshi, "Comparison of Proposed and Existing Fourth Order Schemes for Solving Non-linear Equations, Asian Research Journal of Mathematics, Vol. 15, No. 2, pp.1-7, 2020.
- [5] k. Rajput, A. A. Shaikh and S. Qureshi, "Comparison of Proposed and Existing Fourth Order Schemes for Solving Non-linear Equations, Asian Research Journal of Mathematics, Vol. 15, No. 2, pp.1-7, 2019.
- [6] A. Akram, and Q. U. Ann, "Newton Raphson Method, International Journal of Scientific & Engineering Research", Vol. 6, pp. 456-462, 2015.
- [7] R. Soram, S. Roy, S. R. Singh, M. Khomdram, S. Yaikhom, S. Takhellambam, "On the Rate of Convergence of Newton-Raphson Method", The International Journal of Engineering and Science, Vol. 2, No. 11, pp.5-12, 2013.
- [8] T. J. Ypmat, "Historical development of the newton-raphson method", Society for Industrial and Applied Mathematics, Vol. 37, No. 4, pp. 531-551, 1995.
- [9] Z. Liu, H. Zhang, "Steffensen-Type Method of super third-order convergence for solving nonlinear equations", Journal of Applied Mathematics and Physics Vol.2, pp.581-586, 2014.
- [10] U. K. Qureshi, N. Solanki, M. Y. Ansari, "Algorithm of Difference Operator for Computing a Single Root of Nonlinear Equations", Punjab University Journal of Mathematics, Vol.51, No.4, pp.51-56, 2018.
- [11] W. Nazeera, Amir N., Shin M. K., Young C. K, "Generalized Newton Raphson’s method free from second derivative", J. Nonlinear Sci. Appl. Vol. 9, pp.2823–2831, 2016.

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
1	$\sin x - x + 1 = 0$ $x_0 = 3$	1	1.06086	2.02963	2.30981	2.10406
		2	1.26018	1.93335	2.01359	1.94672
		3	2.25674	1.93456	1.93921	1.93464
		4	1.96109	1.93456	1.93458	1.93456
		5	1.9348	1.93456	1.93456	1.93456
		6	1.93456	-	-	-
		7	1.93456	-	-	-
		8	1.93456	-	-	-

TABLE 1: Results table for the function: $\sin x - x + 1 = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
1	$\sin x - x + 1 = 0$ $x_0 = 3$	1	1.93914	5.02963	0.690193	0.895941
		2	2.32104	0.0962742	0.296213	0.157342
		3	0.996557	0.0012114	0.0743817	0.0120779
		4	0.29565	1.80E-07	0.0046324	7.52E-05
		5	0.0262884	4.00E-15	1.75E-05	2.92E-09
		6	0.0002365	-	-	-
		7	1.93E-08	-	-	-
		8	2.22E-16	-	-	-

TABLE 2: Error table for the function: $\sin x - x + 1 = 0$

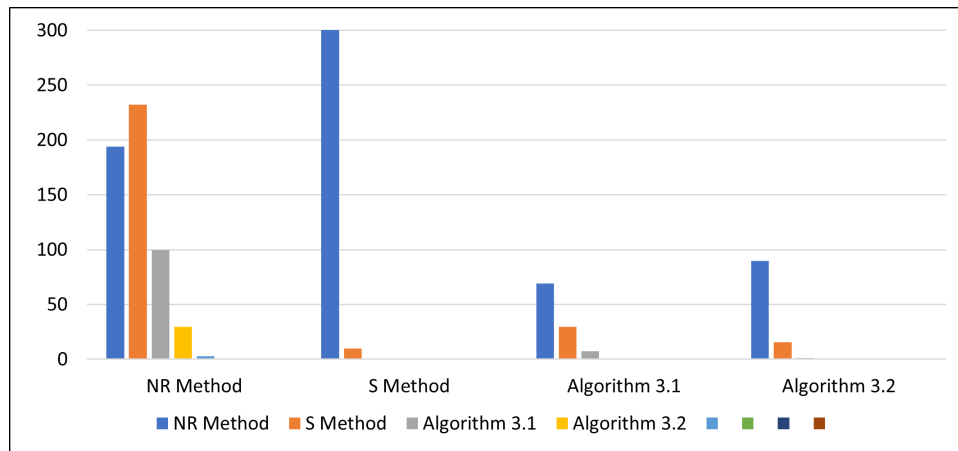


Fig. 1: Comparison of accuracy analysis of $\sin x - x + 1 = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
2	$2x^2 - 5x - 2 = 0$ $x_0 = 0$	1	0.4	0.222222	0.5	0.408163
		2	0.351515	0.328007	0.366667	0.351894
		3	0.350781	0.349939	0.350983	0.350781
		4	0.350781	0.35078	0.350781	0.350781
		5	0.350781	0.350781	0.350781	0.350781
		6	0.350781	0.350781	0.350781	0.350781

TABLE 3: Results table for the function $2x^2 - 5x - 2 = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
2	$x^2 - 5x - 2 = 0$ $x_0 = 0$	1	0.4	0.222222	0.5	0.408163
		2	0.0484848	0.105785	0.133333	0.0562697
		3	0.0007339	0.0219319	0.0156836	0.001112
		4	1.68E-07	0.0008406	0.000202	4.35E-07
		5	8.88E-15	1.19E-06	3.31E-08	6.64E-14
		6	5.55E-17	2.41E-12	8.88E-16	5.55E-17

TABLE 4: Error table for the function $2x^2 - 5x - 2 = 0$

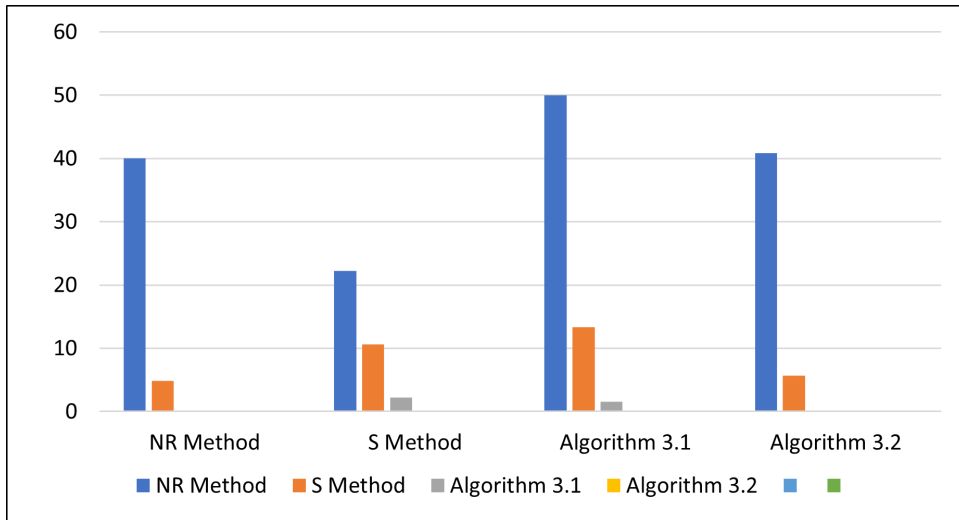


Fig. 2: Comparison of accuracy analysis of $2x^2 - 5x - 2 = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
3	$x^2 - e^x = 0$	1	1	0.6127	1.2254	1.05626
		2	0.733044	0.700737	0.823266	0.750182
		3	0.703808	0.703465	0.711988	0.704447
	$x_0 = 0$	4	0.703467	0.703467	0.703514	0.703468
		5	0.703467	0.703467	0.703467	0.703467
		6	0.703467	-	-	-

TABLE 5: Results table for the function $x^2 - e^x = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
3	$x^2 - e^x = 0$	1	1	0.6127	1.2254	1.05626
		2	0.266956	0.0880376	0.402134	0.30608
		3	0.0292358	0.0027273	0.111278	0.0457352
	$x_0 = 0$	4	0.0003403	2.65E-06	0.0084737	0.0009793
		5	4.58E-08	2.51E-12	4.68E-05	4.43E-07
		6	7.77E-16	-	-	-

TABLE 6: Error table for the function $x^2 - e^x = 0$

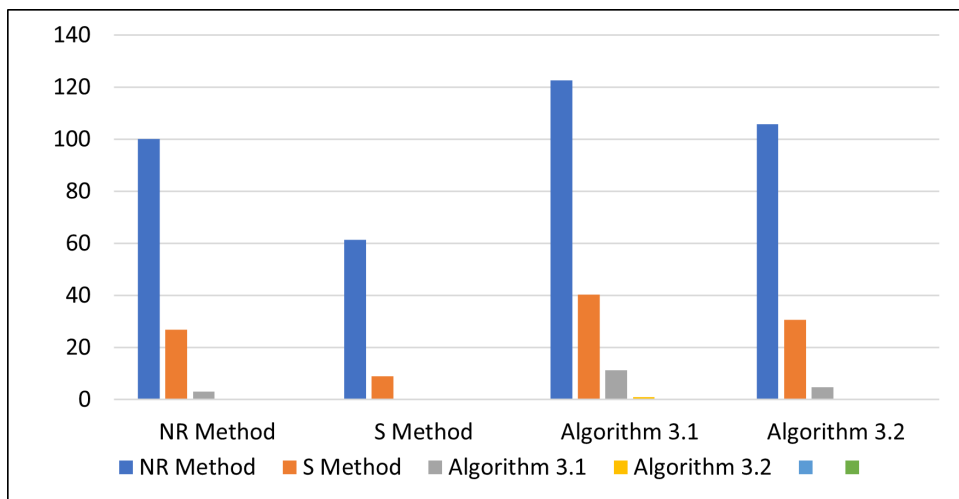


Fig. 3: Comparison of accuracy analysis of $x^2 - e^x = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
4	$e^{-x} - \cos x = 0$ $x_0 = 0$	1	4.71323	4.72127	4.70726	4.70874
		2	4.72129	4.72129	4.72129	4.72129
		3	4.72129	4.72129	4.72129	4.72129
		4	4.72129	-	-	-

TABLE 7: Results table for the function: $e^{-x} - \cos x = 0$

S#	Functions	I	NR Method	S Method	Algorithm 3.1	Algorithm 3.2
4	$e^{-x} - \cos x = 0$ $x_0 = 0$	1	0.286771	0.278735	0.292742	0.291264
		2	0.0080637	2.75E-05	0.0140343	0.0125569
		3	4.02E-07	5.95E-14	1.29E-07	3.70E-07
		4	8.88E-16	-	-	-

TABLE 8: Error table for the function: $e^{-x} - \cos x = 0$

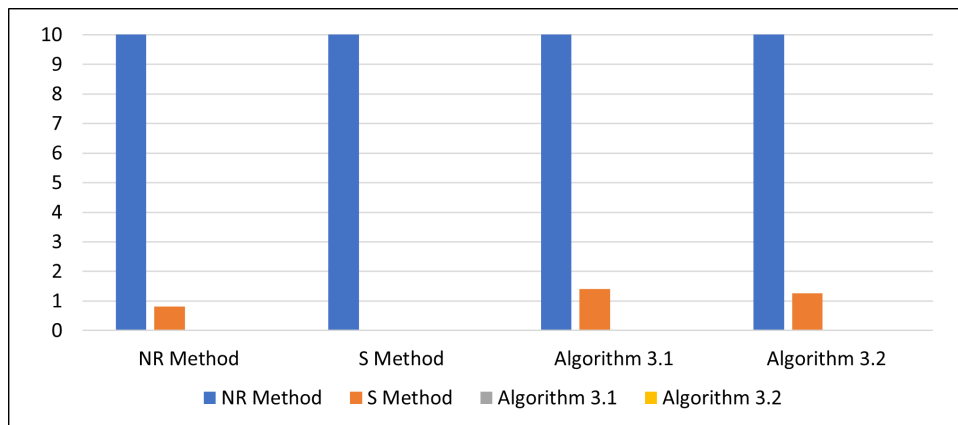


Fig. 4: Comparison of accuracy analysis of $e^{-x} - \cos x = 0$