Available online at http://scik.org
J. Math. Comput. Sci. 10 (2020), No. 5, 1633-1642
https://doi.org/10.28919/jmcs/4723
ISSN: 1927-5307

AN ITERATIVE METHOD FOR SOLVING NON-LINEAR TRANSCENDENTAL EQUATIONS

G. MAHESH[1], G. SWAPNA[2], K. VENKATESHWARLU[1,*]

[1]Department of Freshman Engineering, Geethanjali College of Engineering and Technology, Keesara, Telangana, India

[2]Department of Science and Humanities, Geethanjali College of Pharmacy, Keesara, Telangana, India

Abstract: In this paper, we introduced a new method to compute a non-zero real root of the transcendental equations. The proposed method results in better approximate root than the existing methods such as bisection method, regula-falsi method and secant method. The implementation of the proposed method in MATLAB is applied on different problems to demonstrate the applicability of the method. The proposed method is better in reducing error rapidly, hence converges faster as compared to the existing methods. This method will help to employ in the commercial package for finding a non-zero real root of a given nonlinear equations (transcendental, algebraic and exponential).

Keywords: transcendental equations; secant method; non-linear equation; iteration method.

2010 AMS Subject Classification: 65H04, 65H05.

1. INTRODUCTION

Most of the engineering and scientific problems are expressed as nonlinear transcendental equations for which the evaluations of roots are more complicated. Such nonlinear equations

[*]Corresponding author
E-mail address: venkat1812@gmail.com
Received May 23, 2020

involves in various physical problems like van der waal equation, decay equation, charlesrichter magnitude of earthquake, surface-wave formula and many more [1-2]. These equations can be solved numerically (iterative) and analytically. But it is very difficult to find analytic solution and many times even solution does not exist. The only alternative way to solve those equations is numerical methods. There are a number of roots finding methods are provided for any type of non-linear equations in the literature. The aim of these methods is to find the roots of the given equation with less number of iterations, which reflect through the convergence of the method. The convergence of the method implies the fastness of approaching the root.

In the recent years many methods have been developed for solving nonlinear equations. These methods are developed using Taylor interpolating polynomials, quadrature or other techniques, [3-4]. All the existing and new methods can be classified into two types namely: one-step methods and two-step (predictor–corrector) methods. Abbasbandy [5] and Chun [6] proposed and studied several one-step and two-step iterative methods with higher order convergence by using the decomposition technique. M A Noor [7] introduced a two-step iterative method for finding roots of nonlinear equations these methods perform better than one-step iterative methods including Newton method. He also suggested and analyzed a new family of iterative methods for solving nonlinear equations using the system of coupled equations coupled with decomposition technique [8].

Alojz Suhadolnik [9] proposed new method based on combination of bisection, regula falsi, and parabolic interpolation for solving nonlinear equations. The results show that a switching mechanism between the bisection and regula falsi improves the rate of convergence. In [10], Traub discussed about the various methods to find roots of one-point and multipoint iteration functions. Basto [11] and his team proposed a method based on Abbasbandy on improving the order of accuracy of Newton Raphson method and improving the order of accuracy of Adomian's decomposition method. Hafiz [12] proposed an improved method called BRFC, by the combination of the Bisection, Regula Falsi and parabolic interpolation. Recently Srivastav et al.[13] proposed a new algorithm for computing real root of non-linear transcendental equations where they used inverse sine function in the iterative process.

2. THE PROPOSED METHOD:

The new iterative methods under consideration is proposed as

$$x_{n+1} = x_n - \frac{1}{2}\frac{f(x_n)}{f'(x_n)} + \sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{2nx_n^{n-1}}\left(\frac{f(x_n)}{f'(x_n)}\right)^n, \quad \text{for } n = 0,1,2\ldots \tag{1}$$

By expanding the above iterative formula, one can obtain the standard secant method as in first two terms by replacing $f'(x) = \frac{f(x)-f(x-h)}{h}$.

Theorem 2.1. Suppose α≠ 0 is a real exact root of $f(x)$ and θ is a sufficiently small neighbourhood of α. Let $f''(x)$ exists and $f'(x) \neq 0$ in θ. Then the iterative formula given in Eq. (1) produces a sequence of iterations {x_n : n = 0, 1, 2 . . .} with quadratically convergent..

Proof: The proposed method in Eq. (1) can be expressed in the form

$$x_{n+1} = x_n - \frac{1}{2}\frac{f(x_n)}{f'(x_n)} + \sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{2nx_n^{n-1}}\left(\frac{f(x_n)}{f'(x_n)}\right)^n$$

Since $\lim\limits_{x_n\to\infty}\left(\frac{1}{2nx_n^{n-1}}\left(\frac{f(x_n)}{f'(x_n)}\right)^n\right) = 0$ and hence $x_{n+1} = \alpha$.

$$x_{n+1} = x_n - \frac{1}{2}\frac{f(x_n)}{f'(x_n)} - \frac{1}{2.1}\frac{f(x_n)}{f'(x_n)} + \frac{1}{2.2x_n}\left(\frac{f(x_n)}{f'(x_n)}\right)^2 - \frac{1}{2.3x_n^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^3$$

$$= x_n - \frac{f(x_n)}{f'(x_n)} + \frac{1}{2.2x_n}\left(\frac{f(x_n)}{f'(x_n)}\right)^2 - \frac{1}{2.3x_n^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^3$$

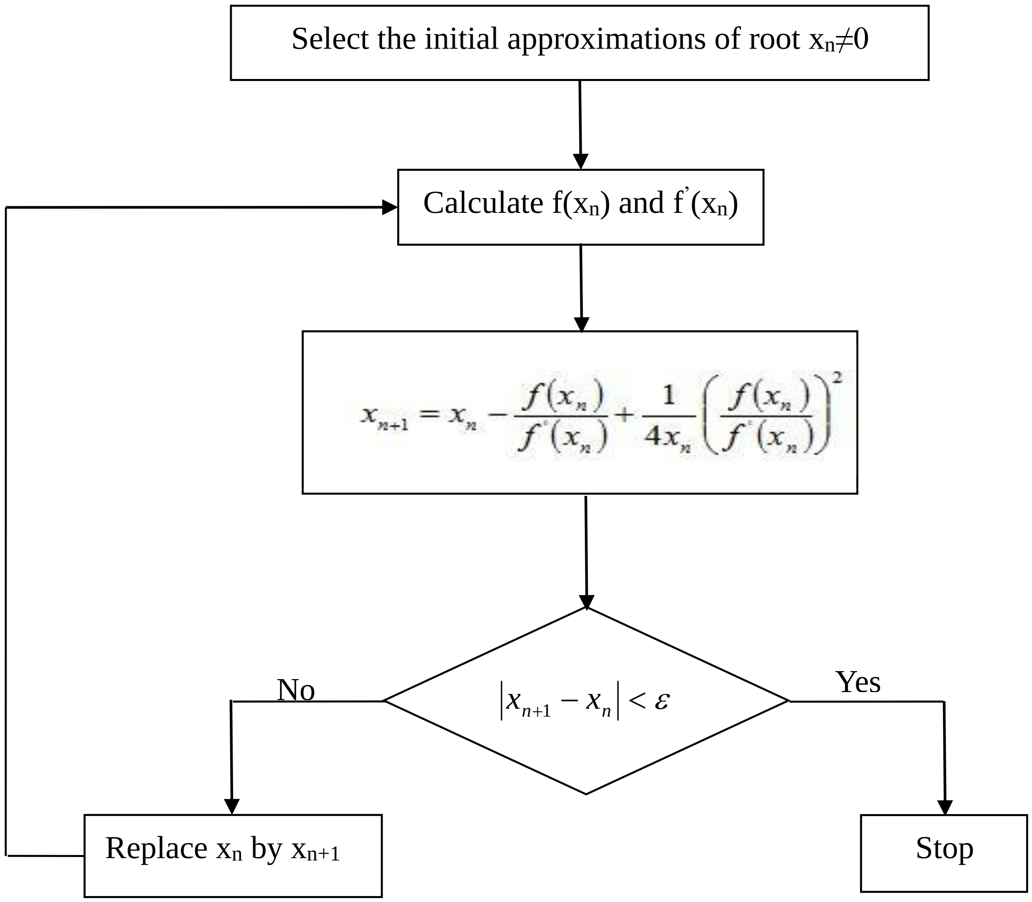
By neglecting $O\left(\frac{1}{2.3x_n^2}\left(\frac{f(x_n)}{f'(x_n)}\right)^3\right)$ and higher order terms the above equation reduces to standard Newton-Raphson method having quadratic convergence.

Therefore, the order of convergence of proposed algorithm is quadratic.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} + \frac{1}{2.2x_n}\left(\frac{f(x_n)}{f'(x_n)}\right)^2 \tag{2}$$

2.2 Proposed method for computing real root of the transcendental equation

- **Step1:** Identify the initial approximations x_0 and $x_1 \neq 0$.
- **Step2**: Apply the proposed method for next approximations using Eq. 2
- **Step3**: Repeat the Step2 until we get the desired approximate root.



2.3 Implementation of Proposed Method in Matlab

In this section present MATLAB implementation of the proposed method as follows.

```
a=input('Given Function:', 's');
b=input('Given derivative Function:','sprime');
f=inline (a);
x(1)=input(' Enter x0 value:');% x0 & x1 are approximate roots
x(2)=input('Enter x1 value:');% such that f(x0)*f(x1)<0
n=input('Enter allowed Error:');
itr=0;
  for i=3:100
        t1=s(x(i-1));
        t2=sprime(x(i-1));
        t=s(x(i-1))/ sprime(x(i-1));
```

```
    x(i)=x(i-1)-t+(0.25*t^2)/x(i-1);
     disp(x(i));
    if abs((x(i)-x(i-1))/x(i))*100<n
        root=x(i);
        disp(itr);
          disp(x(i));
        break % breaking if abs error ≥ n
    end
       itr=itr+1;
end
```

3. COMPARING WITH EXISTING RESULTS

We compared our proposed method with existing results available in the literature [13-14]. Our proposed method shown better convergence than the existing methods.

Table 1: Comparing no. of iterations of proposed method with Srivastav et al [13]

Function	Initial approximation	Exact root	Srivastav et al. [13]	Proposed method
$f(x) = xe^{-x} - 0.1$	0.1	0.11183	3	2
$f(x) = x - e^{\sin(x)} + 1$	4	1.69681	23	6

Table 2: Comparing no. of iterations of proposed method with Srinivasarao Thota [14]

Function	Initial approximation	Exact root	Srinivasarao Thota [14]	Proposed method
$f(x) = xe^{-x} - 0.1$	$x_0 = -0.9$ & $x_1 = 0.9$	0.11183256	11	7
$f(x) = e^{x} - x - 2$	$x_0 = 1$ & $x_1 = 2$	1.146193221	6	4
$f(x) = 8 - 4.5*(x - \sin(x))$	$x_0 = 2$ & $x_1 = 3$	2.43046574	6	4

4. NUMERICAL EXAMPLES

In this section the numerical experiments were performed with the following algebraic/transcendental functions with initial conditions with convergence to the associated root in each case. We compare the number of iterations with various standard methods, required to get approximation root with accuracy of $\varepsilon = 10^{-10}$. The numerical results obtained are presented in the tables 3-8.

Example 1. $f(x) = xe^{-x} - 0.1,\ with\ \ x_0 = -0.9\ \&\ x_1 = 0.9$

Example 2. $f(x) = 8 - 4.5(x - \sin x),\ with\ \ x_0 = 2\ \&\ x_1 = 3$

Example 3. $f(x) = x^6 - x - 1,\ with\ \ x_0 = 1\ \&\ x_1 = 1.5$

Example 4. $f(x) = e^{(x^2+7x-30)} - 1,\ with\ x_0 = 2.5\ \&\ x_1 = 3.1$

Example 5. $f(x) = 11x^{11} - 1,\ with\ x_0 = 0.5\ \&\ x_1 = 1$

Table 3. Comparing No. of iterations by different methods

Function	Initial approximation	Exact Root	Bisection Method	Regula-Falsi method	Secant method	Proposed method
$f(x) = xe^{-x} - 0.1$	x_0=-0.9 & x_1=0.9	0.1118326	30	43	44	7
$f(x) = 8 - 4.5(x - \sin x)$	x_0=2 & x_1=3	2.4304658	34	11	7	4
$f(x) = x^6 - x - 1$	x_0=1 & x_1=1.5	1.1347242	32	40	12	6
$f(x) = e^{(x^2+7x-30)} - 1$	x_0=2.5 & x_1=3.1	3.0000000	15	54	55	5
$f(x) = 11x^{11} - 1$	x_0=0.5 & x_1=1	0.8041307	15	40	41	6

Table 4. Comparing approximate root using various existing methods

$f(x) = xe^{-x} - 0.1, x_0 = -0.9 \ \& \ x_1 = 0.9$

Iteration No.	Bisection method	Iteration No.	Regula-Falsi method	Iteration No.	Secant method	Iteration No.	Proposed method
1	0.4500000	1	0.7144476	1	0.7144476	1	0.6502037
2	0.2250000	2	0.5571840	2	0.5571840	2	0.6002327
3	0.1125000	3	0.4310926	3	0.4310926	3	0.0099950
4	0.0562500	4	0.3349485	4	0.3349485	4	0.3132979
5	0.0843750	5	0.2646681	5	0.2646681	5	0.1089831
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
30	0.1118326	43	0.1118326	43	0.1118326	7	0.1118326

Table 5. Comparing approximate root using various existing methods

$f(x) = 8 - 4.5(x - \sin x) \ x_0 = 2 \ \& \ x_1 = 3$

Iteration No.	Bisection method	Iteration No.	Regula-Falsi method	Iteration No.	Secant method	Iteration No.	Proposed method
1	2.2500000	1	2.4272726	1	2.3885783	1	2.4813257
2	2.3750000	2	2.4302288	2	2.4272726	2	2.4311746
3	2.4375000	3	2.4304482	3	2.4302288	3	2.4304659
4	2.4062500	4	2.4304645	4	2.4304482	4	2.4304658
5	2.4218750	5	2.4304657	5	2.4304645		
.	.	.	.	.	.		
.	.	.	.	.	.		
.	.	.	.	.	.		
34	2.4304658	11	2.4304658	7	2.4304658		

Table 6. Comparing approximate root using various existing methods

$f(x) = x^6 - x - 1, x_0 = 1 \& x_1 = 1.5$

Iteration No.	Bisection method	Iteration No.	Regula-Falsi method	Iteration No.	Secant method	Iteration No.	Proposed method
1	1.2500000	1	1.0505530	1	1.1119831	1	1.3065367
2	1.1250000	2	1.0836271	2	1.1313292	2	1.1870824
3	1.1875000	3	1.1043011	3	1.1342280	3	1.1410268
4	1.1562500	4	1.1168327	4	1.1346519	4	1.1348271
5	1.1406250	5	1.1242817	5	1.1347137	5	1.1347242
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
32	1.1347242	40	1.1347242	12	1.1347242	6	1.1347242

Table 7. Comparing approximate root using various existing methods

$f(x) = e^{(x^2+7x-30)} - 1, \ with \ x_0 = 2.5 \ \& x_1 = 3.1$

Iteration No.	Bisection method	Iteration No.	Regula-Falsi method	Iteration No.	Secant method	Iteration No.	Proposed method
1	2.8000000	1	2.6616638	1	2.6616638	1	3.0449302
2	2.9500000	2	2.7787404	2	2.7787404	2	3.0111423
3	3.0250000	3	2.8616172	3	2.8616172	3	3.0007878
4	2.9875000	4	2.9176384	4	2.9176384	4	3.0000042
5	3.0062500	5	2.9531713	5	2.9531713	5	3.0000000
6	2.9968750	6	2.9742965	6	2.9742965		
.	.	.	.	.	.		
.	.	.	.	.	.		
.	.	.	.	.	.		
15	3.0000060	54	3.0000000	55	3.0000000		

Table 8. Comparing approximate root using various existing methods

$f(x) = 11x^{11} - 1, x_0 = 0.5, \& x_1 = 1$

Iteration No.	Bisection method	Iteration No.	Regula-Falsi method	Iteration No.	Secant method	Iteration No.	Proposed method
1	0.8750000	1	0.5860510	1	0.5452325	1	0.9190630
2	0.8125000	2	0.6226258	2	0.5860510	2	0.8558579
3	0.7812500	3	0.6550521	3	0.6226258	3	0.8176802
4	0.7968750	4	0.6833942	4	0.6550521	4	0.8052490
5	0.8046875	5	0.7077389	5	0.6833942	5	0.8041412
6	0.8007813	6	0.7282437	6	0.7077389	6	0.8041331
.	.	.	.	.	.		
.	.	.	.	.	.		
.	.	.	.	.	.		
15	0.8041306	40	0.8041307	41	0.8041307		

5. CONCLUSIONS

The present work provides the dominance for finding the approximate root of a given transcendental function better than previous existing methods (Bisection, Regula-Falsi and secant methods). This is illustrated through numerical examples. The proposed new algorithm was based on an infinite series of ratio of functions and its derivative. The rate of convergence of the proposed method is discussed and found to be quadratic. Implementation of the proposed method in Matlab is also discussed. Overall, the proposed method executes much faster and more accurate convergence to the exact solution than the previously existing other standard methods. This proposed algorithm is useful for solving the complex real life problems such as a chemical equilibrium problem, azeotropic point of a binary solution and volume from van der Waals equation.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] S. Soomro, A. Shaikh, S. Abro. An algorithm for improving convergence rate by decomposition of interval for solving f (x)= 0, Amer. J. Eng. Res. 8 (2019), 10 – 14.

[2] O. Said Solaiman, I. Hashim, Efficacy of Optimal Methods for Nonlinear Equations with Chemical Engineering Applications, Math. Probl. Eng. 2019 (2019), 1728965.

[3] M. Frontini, E. Sormani, Third-order methods from quadrature formulae for solving systems of nonlinear equations, Appl. Math. Comput. 149 (2004), 771–782.

[4] S. Amat, S. Busquier, J. Gutierrez, Geometric constructions of iterative functions to solve nonlinear equations, J. Comput. Appl. Math. 151(2003),197–205.

[5] S. Abbasbandy, Improving newton–raphson method for nonlinear equations by modified adomian decomposition method, Appl. Math. Comput. 145 (2003), 887–893

[6] C. Chun, Iterative methods improving newton's method by the decomposition method, Comput. Math. Appl. 50 (2005), 1559–1568.

[7] M. A. Noor, F. Ahmad, S. Javeed, Two-step iterative methods for nonlinear equations, Appl. Math. Comput. 181 (2006), 1068–1075.

[8] M. A. Noor, New family of iterative methods for nonlinear equations, Appl. Math. Comput. 190 (2007), 553–558.

[9] A. Suhadolnik, Combined bracketing methods for solving nonlinear equations, Appl. Math. Lett. 25 (2012), 1755–1760.

[10] J. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[11] M. Basto, V. Semiao, F.L. Calheiros, A new iterative method to compute nonlinear equations Appl. Math. Comput. 173 (2006), 468–483.

[12] M. Hafiz, A new combined bracketing method for solving nonlinear equations, J. Math. Comput. Sci. 3 (2013), 87–93.

[13] V. K. Srivastav, S. Thota, M. Kumar, A new trigonometrical algorithm for computing real root of nonlinear transcendental equations, Int. J. Appl. Comput. Math. 5 (2019), 44.

[14] T. Srinivasarao, A new root-finding algorithm using exponential series, Ural Math. J. 5 (2019), 83–90.