Southeast-Asian J. of Sciences, Vol. 7, No. 2 (2019) pp. 106-114

A NEW ALGORITHM FOR COMPUTING A ROOT OF TRANSCENDENTAL EQUATIONS USING SERIES EXPANSION

Srinivasarao Thota* and **Tekle Gemechu**

Department of Applied Mathematics
School of Applied Natural Sciences
Adama Science and Technology University
Post Box No. 1888, Adama, Ethiopia
e-mail: srinithota@ymail.com; tekgem@yahoo.com

Abstract

In this paper, we discuss a new algorithm to find a non-zero real root of the transcendental equations using series expansion. This proposed method is based on the inverse series expansion, which gives a good approximate root than some other existing methods. The implementation of this algorithm is presented in Matlab and Maple. Sample numerical examples are presented to illustrate and validate the efficiency of the proposed algorithm. The method will help to implement in the commercial package for finding a real root of a given transcendental equation.

1 Introduction

In science, engineering and computing, solving for root or finding a root of transcendental equations is playing important role. In the literature, we have many well known algorithms for root finding, (for example, *Bisection*,*Muller's*, *Regula-Falsi*, *Secant*, *Newton-Raphson* methods etc.); the modified and hybrid algorithms for finding an approximate root are available in literature, see for example [2, 12, 10, 11, 5, 1, 15, 16, 18, 17, 19, 4, 22, 27, 21, 20, 9, 14, 6, 28, 7, 25, 26]. In the present work, we propose a new method based on series expansion, which finds quick root in comparison with other existing algorithms. The rest

*Corresponding Author. Email: srinivasarao.thota@astu.edu.et
Key words: Root finding algorithms, Transcendental equations, Series expansion, Real root.
2010 AMS Classification: 65Hxx, 65H04.

of paper is arranged as follows: Section 2 describes the proposed method, their mathematical formulation, calculation steps and flow chart; implementation of the proposed algorithm in Maple and matlab is presented in Section 3 with sample computations; and Section 4 discuss some numerical examples to illustrate the algorithm and comparisons are made to show efficiency of the new algorithm.

2 Proposed Algorithm

The new iterative formula using series expansion is proposed as

$$x_{n+1} = x_n \left(\frac{x_n f'(x_n)}{f(x_n) + x_n f'(x_n)} \right), \qquad n = 0, 1, 2, \ldots. \tag{1}$$

By expanding this iterative formula, one can obtain the standard Newton-Raphson method as in first two terms. This is shown in the following theorem.

Theorem 1. *Suppose $\alpha \neq 0$ is a real exact root of $f(x)$ and θ is a sufficiently small neighbourhood of α. Let $f''(x)$ exist and $f'(x) \neq 0$ in θ. Then the iterative formula given in equation* (1) *produces a sequence of iterations $\{x_n : n = 0, 1, 2, \ldots\}$ with order of convergence $p \geq 2$.*

Proof. The iterative formula given in equation (1) can be expressed in the following form

$$x_{n+1} = x_n \left(\frac{x_n f'(x_n)}{f(x_n) + x_n f'(x_n)} \right).$$

Since

$$\lim_{x_n \to \alpha} \left(\frac{x_n f'(x_n)}{f(x_n) + x_n f'(x_n)} \right) = 1,$$

and hence $x_{n+1} = \alpha$. This implies that α is a fixed point of the function $t(x)$.

Using the standard expansion of $(1-x)^{-1}$ as

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \cdots \tag{2}$$

and from equations (1) and (2), we have

$$
\begin{aligned}
x_{n+1} &= x_n \left(\frac{x_n f'(x_n)}{f(x_n) + x_n f'(x_n)} \right) \\
&= x_n \left(\frac{1}{1 - \left(\frac{-f(x_n)}{x_n f'(x_n)} \right)} \right) \\
&= x_n (1 + \left(\frac{-f(x_n)}{x_n f'(x_n)} \right) + \left(\frac{-f(x_n)}{x_n f'(x_n)} \right)^2 + \left(\frac{-f(x_n)}{x_n f'(x_n)} \right)^3 \\
&\quad + \left(\frac{-f(x_n)}{x_n f'(x_n)} \right)^4 + \cdots) \\
&= x_n - \frac{f(x_n)}{f'(x_n)} + \frac{1}{x_n} \left(\frac{f(x_n)}{f'(x_n)} \right)^2 - \frac{1}{x_n^2} \left(\frac{f(x_n)}{f'(x_n)} \right)^3 \\
&\quad + o \left(\frac{1}{x_n^3} \left(\frac{f(x_n)}{f'(x_n)} \right)^4 \right).
\end{aligned}
$$

Since $f(x_n) \approx 0$, when we neglect higher order terms, then the above equation becomes Newton-Rapson method. Indeed, we have the following formulae obtained from first two, three and four terms of the expansion respectively.

$$
\begin{aligned}
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}. \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} + \frac{1}{x_n} \left(\frac{f(x_n)}{f'(x_n)} \right)^2. \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} + \frac{1}{x_n} \left(\frac{f(x_n)}{f'(x_n)} \right)^2 - \frac{1}{x_n^2} \left(\frac{f(x_n)}{f'(x_n)} \right)^3.
\end{aligned}
$$

In the above equations, we obtained Newton-Rapson method having quadratic convergence in first two terms. Therefore, the order of convergence of proposed methods (1) is at least $p \geq 2$. □

2.1 Steps for Computing Root

I Select an approximation $x_n \neq 0$.

II Apply the iterative formula given in equation (1).

III Repeat Step II until we get desired approximate root, for $n = 0, 1, 2, \ldots$.

Flow chat of the proposed algorithm is presented in Figure 1

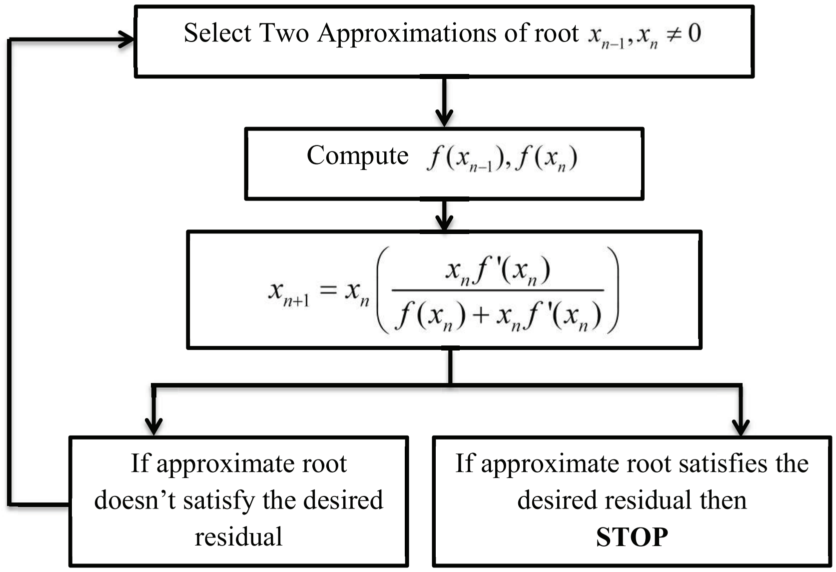

Figure 1: Flow chart for proposed algorithm

3 Implementation of Proposed Algorithm

In this section, we discuss the MATLAB and Maple implementation of the proposed algorithm. The data type `SeriesNewton(f,x0,esp,n)` describes the implementation in MATLAB, where `f` is given non-linear transcendental function, `x0` is the initial approximation of the root, `esp` is the relative error and `n` is the number of iterations required.

```
function root = SeriesNewton(f,x0,esp,n)
iter = 0; ea = 0; xn = x0;
fd = inline(char(diff(formula(f))),'x');
disp(' ----------------------------');
disp(' No     Root     f(Root)     %error ');
disp(' ---------------------------');
while (1)
  xnold = xn;
  xn =x0*(x(0)*fd(x0)/(f(x0)+x0*fd(x0)));
  xnnew = xn;
  disp(sprintf('%4d %10.4f %10.2f %8.2f',iter+1,xn,f(xn),ea));
```

```
    iter = iter + 1;
    if xn  = 0, ea = abs((xn - xnold)/xn) * 100; end
    x0 = xn;
    if ea <= esp | iter >= n, break, end
  end
  disp(' -------------------------------------');
  disp(['Given function f(x) = ' char(f)]);
  disp(sprintf('Approximate root = %10.10f',xn));
```

The following a data type `SeriesNewton(f,x0,n)` gives the implementation in Maple, where `f` is given non-linear transcendental function, `x0` is the initial approximation of the root, and `n` is the number of iterations required.

```
  SeriesNewton:=proc(f,x0,n)
  local iten, fx0;
  for iten from 1 by 1 while iten < n+1
    do
      printf("Iteration %g :  ", iten);
      x0:=(x0*(x0*subs(x=x0,diff(f,x))/
       (subs(x=x0,f)+x0*subs(x=x0,diff(f,x)))));
      fx0:=subs(x = x0, f);
    end do;
  return x0,fx0;
 end proc:
```

Sample computations using the implementation of the proposed algorithm are presented in Section 4.

4 Numerical Examples

This section provides some numerical examples to discuss the algorithm presented in Section 2 and comparisons are taken into account to conform that the algorithm is more efficient than other existing methods.

Example 4.1. *Consider the following transcendental equations [11]. We compare the number of iterations required to get approximation root with accuracy of* 10^{-15}. *The numerical results are provided in Table 1.*

a. $f(x) = \ln(x)$, *with initial approximation* 0.5.

b. $f(x) = x - e^{\sin(x)} + 1$, *with initial approximation* 1.5.

c. $f(x) = xe^{-x} - 0.1$, *with initial approximation* 0.1.

Table 1: Comparing No. of iterations by different methods

Fun.	Exact Root	Regula Falsi method	Newton Raphson method	Steffen method	Proposed method
a.	1.00000	27	Divergent	Failure	3
b.	1.69681 & 0	32	Not Convergent	Failure	6
c.	0.11183	15	Failure	Failure	3

The numerical results given in Table 1 shows that the proposed method is more efficient than other methods.

Example 4.2. *Consider a transcendental equations of the following type. We find approximate root using formulae different existing methods to show the convergence of the proposed algorithm gives a root faster than other methods, see Table 2. To find approximate root, we start with an initial approximation as 1.5 for Newton-Raphson and Proposed methods;* $a = 0, b = 1.5$ *for bisection and regula-falsi methods. We have an exact real root is 0.5.*

$$f(x) = 2x^3 + 11x^2 + 12x - 9. \tag{3}$$

Table 2: Comparing approximate root using proposed algorithm with other existing methods

Iteration No.	Newton-Raphson method	Bisection methods	Regula-Falsi method	Proposed method
1	0.8076923077	0.75	0.2727273	1.026315789
2	0.5428093643	0.375	0.4044266	0.7296759182
3	0.5010101572	0.5625	0.4612480	0.5699486582
4	0.5001005827	0.46875	0.4845290	0.5097474998
5	0.5000105835	0.515625	0.4938624	0.5002347438
6	0.5000005827	0.4921875	0.4975712	0.5000001415
7	0.5000000009	0.5039063	0.4990399	0.4999999998

Example 4.3. *This example gives the sample computation using MatLab and Maple implementation as described in Section 3. Consider a transcendental equation of the form*

$$f(x) = 2x^3 + 11x^2 + 12x - 9$$

with initial approximation of the root as 1.5.

Using MatLab implementation, we have the following computations.

```
f=inline('2*x^3+11*x^2+12*x-9','x');
function root = SeriesNewton(f,1.5,0.00001,9)
------------------------------------------------------
 No      Root              f(Root)          %error
------------------------------------------------------
 1   1.026315789    17.0644     ---
 2   0.7296759182    6.38981    10.60666586
 3   0.5699486582    1.78293    0.4217779832
 4   0.5097474998    0.240146     0.0007051480758
 5   0.5002347438     0.005752    3.526445668*10^-8
 6   0.5000001415     3.467e-06         0
 7   0.4999999998     -4e-09            0
 8   0.4999999996     -7e-09            0
 9   0.4999999998     -4e-09            0
-----------------------------------------------------
Given function f(x) = 2*x^3+11*x^2+12*x-9
Approximate root = 0.4999999998
```

Using Maple implementation, we have the following computations.

```
> f:= 2*x^3+11*x^2+12*x-9:
> ExpNewton(f,1.5,9);
```

Iteration 1 :	1.026315789
Iteration 2 :	0.7296759182
Iteration 3 :	0.5699486582
Iteration 4 :	0.5097474998
Iteration 5 :	0.5002347438
Iteration 6 :	0.5000001415
Iteration 7 :	0.4999999998
Iteration 8 :	0.4999999996
Iteration 9 :	0.4999999998

One can use the implementation of the proposed algorithm to speed up the manual calculations.

5 Conclusion

In this present work, we presented a new algorithm to compute an approximate real root of a given (nonlinear) transcendental equation. The proposed new algorithm was based on series expansion. Illustrative examples were given to prove its better convergence and efficiency over some existing methods. The proposed algorithm is useful for solving some complex real life problems. Implementation of the proposed new algorithm was using Matlab and Maple.

References

[1] C. Gutierrez, F. Gutierrez, M.-C. Rivara: Complexity of the bisection method, Theoretical Computer Science Vol. 382 (2007) 131–138.

[2] D. Bachrathy, G. Stpn: Bisection method in higher dimensions and the efficiency number, Mechanical Engineering, Vol. 56 (2) (2012) 81–86.

[3] Dowell-Jarrat: A modified Regula-Falsi method for computing the real root of an equation. BIT Numerical Mathematics. Vol.11 (1971), 168–174.

[4] E. Novak, K. Ritter, And H. Wozniakowski: Mathematics of computation, Vol. 64, (212), 1995, 1517–1539.

[5] Eiger, K. Sikorski, F. Stenger: A Bisection Method for Systems of Nonlinear Equations, ACM Transactions on Mathematical Softwares, Vol. 10 (4) 1984, 367–377.

[6] G.R. Wood: The bisection method in higher dimensions, Mathematical Programming Vol. 55 (1992) 319–337.

[7] J.C.Yakoubsohn: Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions, Journal of Complexity Vol. 21 (2005) 652–690.

[8] Johan and Ronald. The Newton-Raphson Method. International Journal of Mathematical education in Science and Technology. Vol.26 (1995), No.2, 177–193.

[9] J. R. SHARMA and R. K. GOYAL: Fourth-order derivative-free methods for solving nonlinear equations, International Journal of Computer Mathematics, Vol. 83, No. 1, 2006, 101–106.

[10] J. Chen, W. Li: An exponential regula falsi method for solving nonlinear equations, Numerical Algorithms (2006) 41: 327–338.

[11] J. Chen, W. Li: An improved exponential regula falsi methods with quadratic convergence of both diameter and point for solving nonlinear equations, Applied Numerical Mathematics, 57 (2007) 80–88.

[12] Jinhai Chen: New modified regula falsi method for nonlinear equations, Applied Mathematics and Computation, 184 (2007) 965–971.

[13] John Abbott: Quadratic interval refinement for real roots. Vol.48 (2014), No.1. ACM communications in computer algebra.

[14] M.N. Vrahatis, K.I. Iordanidis: A Rapid Generalized Method of Bisection for Solving System of Non-linear Equation, Numer. Math, Vol 49, 1986, 123–138.

[15] Mamta, V. Kanwar, V.K. Kukreja, Sukhjit Singh: On a class of quadratically convergent iteration formulae, Applied Mathematics and Computation, Vol. 166 (2005) 633–637.

[16] Mamta, V. Kanwar, V.K. Kukreja, Sukhjit Singh: On some third-order iterative methods for solving nonlinear equations, Applied Mathematics and Computation, Vol. 171 (2005) 272–280.

[17] M. A. Noor, K. I. Noor, W. A. Khan, F. Ahmad: On iterative methods for nonlinear equations, Applied Mathematics and Computation, Vol. 183 (2006) 128–133.

[18] M. Aslam Noor, F. Ahmad: Numerical comparison of iterative methods for solving nonlinear equations, Applied Mathematics and Computation, Vol. 180 (2006) 167–172.

[19] M. A. Noor, K. I. Noor: Three-step iterative methods for nonlinear equations, Applied Mathematics and Computation, Vol 183 (2006) 322–327.

[20] S. Hussain, V. K. Srivastav, S. Thota: Assessment of Interpolation Methods for Solving the Real Life Problem, International Journal of Mathematical Sciences and Applications, 5 (1) (2015), 91–95.

[21] S. Thota, V. K. Srivastav: Quadratically Convergent Algorithm for Computing Real Root of Non-Linear Transcendental Equations, BMC Research Notes, (2018) 11:909.

[22] S. Thota, V. K. Srivastav: Interpolation based Hybrid Algorithm for Computing Real Root of Non-Linear Transcendental Functions, International Journal of Mathematics and Computer Research, 2 (11) (2014), 729–735.

[23] Saied and LIAO: A new modification of False-Position method based on homotopy analysis method. Applied Mathematics and Mechanics. Vol.29.No.2, 223–228.

[24] Sagraloff and Mehlhorn: Computing real roots of real polynomials. 2013.

[25] T. Gemechu: Some Root Finding With Extensions to Higher Dimensions, Mathematical Theory and Modeling, 7 (4) (2017), 1–13.

[26] T. Parveen, S. Singh, S. Thota, V. K. Srivastav: A New Hydride Root-Finding Algorithm for Transcendental Equations using Bisection, Regula-Falsi and Newton-Raphson methods, National Conference on Sustainable & Recent Innovation in Science and Engineering, 2019.

[27] V. K. Srivastav, S. Thota, M. Kumar: A New Trigonometrical Algorithm for Computing Real Root of Non-linear Transcendental Equations, International Journal of Applied and Computational Mathematics, (2019) 5:44.

[28] X. Wu: Improved Muller method and Bisection method with global and asymptotic superlinear convergence of both point and interval for solving nonlinear equations, Applied Mathematics and Computation, Vol. 166 (2005) 299–311.