International Journal of Advances in Applied Sciences (IJAAS)
Vol. 10, No. 2, June 2021, pp. 141∼148
ISSN: 2252-8814, DOI: 10.11591/ijaas.v10.i2.pp141-148

A numerical algorithm to find a root of non-linear equations using householder's method

Srinivasarao Thota
Department of Applied Mathematics, School of Applied Natural Sciences, Adama Science and Technology University, Adama, Ethiopia

Article Info

Article history:

Received Nov 24, 2020
Revised Feb 2, 2021
Accepted Apr 15, 2021

Keywords:

Non-linear equations
Iterative method
Householder's method
Approximate root
Maple implementation

ABSTRACT

In this paper, we propose a three step algorithm using Householder's method for finding an approximate root of the given non-linear equations in one variable. Several numerical examples are presented to illustrate and validation of the proposed method. Implementation of the proposed algorithm in Maple is also discussed with sample computations.

Corresponding Author:

Srinivasarao Thota
Department of Applied Mathematics
School of Applied Natural Sciences
Adama Science and Technology University
Post Box No. 1888, Adama, Ethiopia
Email: srinithota@ymail.com, srinivasarao.thota@astu.edu.et

1. INTRODUCTION

Finding an approximate solution of a given non-linear equation is one of the important topics in numerical analysis. Many mathematical models of physics, economics, engineering, science and any other disciplines come up with non-linear equations of the type . In resent time, several scientists and engineers have been focused on to solve the non-linear equations both numerically and analytically. In the literature, there are several iterative methods/algorithms available using different techniques such as homotopy, interpolation, Taylor's series, quadrature formulas, decomposition etc. with modifications and improvements, and different hybrid iterative methods, see, for example [1]-[17]. There are several methods available to find an approximate solution of non-linear systems also, see for example, [18]-[23]. In generally, the roots of non-linear or transcendental equations cannot be expressed in closed form or cannot be computed analytically. The root-finding algorithms provide us to compute approximations to the roots; these approximations are expressed either as small isolating intervals or as isolating point numbers. In this paper, we use the Householder's technique to create a three-step iterative algorithm for solving the given non-linear equation of single variable with converging order more than or equal to four. Indeed, we focus on creating a tree-step algorithm to find a root of a given non-linear equation using the well-known techniques namely, Householder's technique and modified Newton's method. We also focus on its Maple implementation with sample computations. The proposed algorithm is simple to understand and efficient than some existing methods.

Since, we focus on creating an efficient and simple three-step algorithm using Householder's method and modified Newton's method, we recall basic concepts related to the existing methods. In this paper, we consider the non-linear equations of the following type (1).

$$f(x) = 0, \tag{1}$$

where $f : I \to \mathbb{R}$ is a scalar function in single variable for an open interval $I \subseteq \mathbb{R}$ and continuously differentiable function. In numerical analysis, the order of convergence of a convergent sequence are quantities that represent how quickly the sequence approaches its limit. The order of converges of any iterative method is defined [13] as

$$|e_{n+1}| \leq c|e_n|^p,$$

where p is the order of convergence and c is a positive finite constant. Householder's method is one of the famous methods in producing a sequence of approximation roots of (1) with initial point x_0. The Householder's method has cubic order of convergence [1]-[2] and is given as (2)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3}, \quad n = 0, 1, 2, \ldots. \tag{2}$$

In the literature, there are several modifications of Householder's method have been developed and analyzed by various techniques, for example, see [1]-[3], [24]. The modified Householder's methods give better performance than the Newton's method.

Recall the predictor-corrector method developed by Traub [4] in 1864 as follows, for $f'(x_n) \neq 0$,

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_n &= y_n - \frac{f(y_n)}{f'(y_n)}. \end{aligned} \tag{3}$$

A modified Newton's method developed by McDougall and Wotherspoon [25] is converged faster than the Newton's method with a convergence order $1 + \sqrt{2}$, and it is given as follows: Let $x_0 = y_0$, for $n = 0$ (initially), compute

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

and for $n \geq 1$,

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'\left(\frac{x_{n-1}+y_{n-1}}{2}\right)}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'\left(\frac{x_n+y_n}{2}\right)}. \end{aligned} \tag{4}$$

If $x_n = y_n$, for all n in equation (4), then the modified Newton's method developed by McDougall and Wotherspoon becomes Newton's predictor-corrector method as given in equation (3). The rest of paper is organized as follows: Section 2. presents the methodology and steps involving in the proposed algorithm and the flow chart; Section 3. presents couple of numerical examples to illustrate and validate the proposed algorithm; and Section 4. presents the Maple implementation of the proposed algorithm with sample computations.

2. PROPOSED ITERATIVE ALGORITHM

In this section, we present a three-step iterative algorithm using the well-known Householder's method given in (2) and a modified newton's method given in equation (4). The proposed iterative formula under consideration is given as: set $x_0 = y_0$, for $n = 0$ (initially), compute

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3},$$

and for $n \geq 1$,

$$y_n = x_n - \frac{f(x_n)}{f'\left(\frac{x_{n-1}+y_{n-1}}{2}\right)} - \frac{(f(x_n))^2 f''(x_n)}{2\left(f'\left(\frac{x_{n-1}+y_{n-1}}{2}\right)\right)^3},$$
$$x_{n+1} = x_n - \frac{f(x_n)}{f'\left(\frac{x_n+y_n}{2}\right)} - \frac{(f(x_n))^2 f''(x_n)}{2\left(f'\left(\frac{x_n+y_n}{2}\right)\right)^3}. \tag{5}$$

If $x_n = y_n$, for all n and $f''(x_n) \neq 0$ in equation (4), then the proposed algorithm becomes Householder's predictor-corrector method and it has fourth-order convergence [3, 24]. If $x_n = y_n$ for all n and $f''(x_n) = 0$ in (4), then the proposed algorithm becomes Newton's predictor-corrector method as given in (3).

2.1. Steps for calculating root

The following steps are involved to compute an approximate root of a given using the proposed algorithm.

1. Choose an initial approximation $x_0 \in \mathbb{R}$ with $x_0 = y_0$ for $n = 0$.
2. Compute x_1 using formula given in (5). i.e.,
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3},$$
3. For $n \geq 1$, compute y_n and x_{n+1} using the proposed algorithm. i.e.,
$$y_n = x_n - \frac{f(x_n)}{f'\left(\frac{x_{n-1}+y_{n-1}}{2}\right)} - \frac{(f(x_n))^2 f''(x_n)}{2\left(f'\left(\frac{x_{n-1}+y_{n-1}}{2}\right)\right)^3},$$
$$x_{n+1} = x_n - \frac{f(x_n)}{f'\left(\frac{x_n+y_n}{2}\right)} - \frac{(f(x_n))^2 f''(x_n)}{2\left(f'\left(\frac{x_n+y_n}{2}\right)\right)^3}.$$
4. Repeat Step 3 until we get desired approximate root, for $n = 1, 2, 3 \ldots$.

Flow chat of the proposed algorithm is presented in Figure 1.

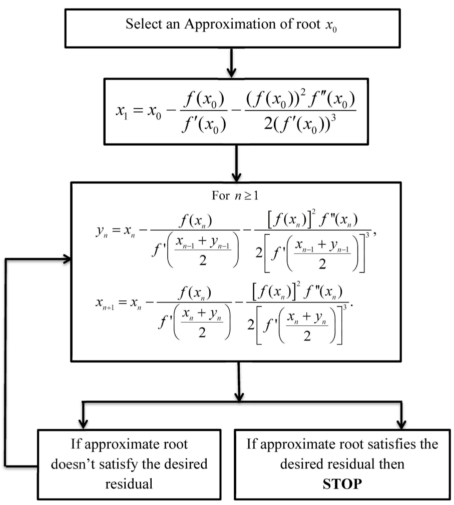

Figure 1. Flow chat of the proposed algorithm

3. NUMERICAL EXAMPLES

In this section, we present couple of numerical examples to illustrate the proposed algorithm and comparisons are made to confirm that the algorithm gives solution faster than existing methods.

Consider the following nonlinear equation to compute an approximate root using the proposed method.

$$f(x) = xe^x - 1. \tag{6}$$

The exact root of the equation is $x = 0.5671432904$. Suppose the initial approximation is $x_0 = 3$. The stopping condition is corrected to ten decimal places. Set $x_0 = y_0 = 3$, compute x_1 as follows

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3} = 1.922456285.$$

Now, we compute y_n and x_{n+1}, for $n = 1$, using the proposed algorithm, as follows

$$y_1 = x_1 - \frac{f(x_1)}{f'\left(\frac{x_0+y_0}{2}\right)} - \frac{(f(x_1))^2 f''(x_1)}{2\left(f'\left(\frac{x_0+y_0}{2}\right)\right)^3} = 1.767472904,$$

$$x_2 = x_1 - \frac{f(x_1)}{f'\left(\frac{x_1+y_1}{2}\right)} - \frac{(f(x_1))^2 f''(x_1)}{2\left(f'\left(\frac{x_1+y_1}{2}\right)\right)^3} = 0.9087794052.$$

Similarly, we can compute y_n and x_{n+1}, for $n > 1$, using the proposed algorithm, as follows

$$\begin{aligned} y_2 &= 0.8380960197, \quad x_3 = 0.5661945014, \\ y_3 &= 0.5667781506, \quad x_4 = 0.5671428368, \\ y_4 &= 0.5671432908, \quad x_5 = 0.5671432903, \\ y_5 &= 0.5671432904, \quad x_6 = 0.5671432904. \end{aligned}$$

Hence the required approximate root of the given (6) is 0.5671432904 up to ten decimal places and with the tolerance error 1×10^{-10}. The graphical representations (using Maple) of the iterations values at different iterations is shown in Figure 2, absolute errors at different iterations are shown in Figure 3 and absolute function values at different iterations are shown in Figure 4.

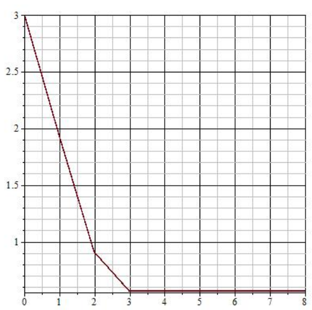

Figure 2. Iterations at different iterations

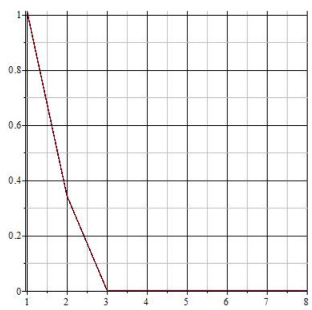

Figure 3. Absolute errors at different iterations

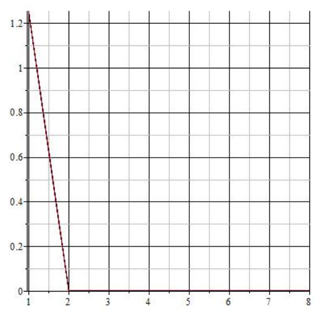

Figure 4. Absolute function values at different iterations

The following non-linear equations with initial approximations to compare the number of iterations taken to obtain the approximate root using existing methods and proposed method. For simplicity, we take tolerance error 1×10^{-8}.

$$f_1(x) = x^2 - (1-x)^5, \qquad x_0 = 5.0,$$
$$f_2(x) = x^3 - e^{-x}, \qquad x_0 = 6.0,$$
$$f_3(x) = -20x^5 - \frac{x}{2} + \frac{1}{2}, \qquad x_0 = 1.5,$$
$$f_4(x) = \cos(x) - x^3, \qquad x_0 = 8.0,$$
$$f_5(x) = (e^x + x - 20)^3, \qquad x_0 = 4.0,$$
$$f_6(x) = x - 3\ln(x), \qquad x_0 = 0.5.$$

We compare the proposed method with various existing methods such as Newton's Method (NM), Kang et al. Method (KM) [26], McDougall-Wotherspoon Method (MWM) [25], Neamvonk Method (ANM) [27]-[29] and Proposed Method (PM) to show the efficiency of the proposed method. Table 1, shows the number of iterations taken by various methods to obtain the required approximate root.

Table 1. Comparison of iterations by different methods for example 3

Function	Approx. Root	NM	KM	MWM	ANM	PM
$f_1(x)$	0.345954815848	12	6	9	6	6
$f_2(x)$	0.772882959149	10	5	7	6	4
$f_3(x)$	0.427677296931	10	5	8	6	4
$f_4(x)$	0.865474033102	10	5	7	6	4
$f_5(x)$	0.865474033102	44	14	33	10	12
$f_6(x)$	0.865474033102	7	4	5	5	2

From Table 1, one can observe that the proposed method (PM) gives the required approximate root faster than the other existing methods (i.e., in lowest iteration number).

4. MAPLE IMPLEMENTATION OF PROPOSED ALGORITHM

In this section, we present the implementation of the proposed algorithm in Maple.

Input: given $f(x)$,
initial approximation $x[0]$,
tolerance ϵ,
correct to decimal places δ,
maximum number of iterations n.
Output: approximate solution.

4.1. Pseudo-code

- Set $x[0] = y[0]$ and compute $x[1]$ using formula in (5).
- for i from 1 to n do
  - compute y[i] and x[i+1] using formulas in (5).
  - if abs($x[i+1] - x[i]$) $< \epsilon$ and abs($f(x[i+1])$) $< \delta$ then break;
- output $x[i+1]$

4.2. Maple programming

We present the maple code of the proposed algorithm as follows.

```
> eps := TYPE;     # epsilon value
> epsabs := TYPE;     # delta value
> f(x):= TYPE;     # given non-linear equation
> fd:=D(f);
> fdd:=D(D(f));
```

```
> x[0] := TYPE;     # initial approximation
> y[0]:=x[0];
> n:= TYPE;     # required number of iterations
> x[1] := x[0]-(f(x[0]))/(fd(x[0]))-
((f(x[0]))^(2)*fdd(x[0]))/ (2*(fd(x[0]))^(3)):
> for i from 1 to n do
> y[i]:=x[i]-(f(x[i]))/(fd((x[i-1]+y[i-1])/(2)))-
(f(x[i]))^(2)fdd(x[i]))/(2*(fd((x[i-1]+y[i-1])/(2)))^(3)):
> x[i+1] := x[i]-(f(x[i]))/(fd((x[i]+y[i])/(2)))-
((f(x[i]))^(2)*fdd(x[i]))/(2*(fd((x[i]+y[i])/(2)))^(3)):
> abserror[i]:=abs(x[i+1]-x[i]);
> absfun[i]:=abs(f(x[i+1]));
> if abs(x[i+1]-x[i]) < eps or abs(f(x[i+1])) < epsabs
> then break;
> end if:
> printf("Iteration No:  %g = %10.10f\\n", i,x[i+1]);
> end do;
```

4.3. Sample computations

Recall the function $f_6(x)$ from Example 3. to compute an approximate solution using the maple implementation.

```
> eps := 0.00000001;
> epsabs := 0.00000001;
> f(x):= x-3 ln(x);
> fd:=D(f);
> fdd:=D(D(f));
> x[0] := 0.5;y[0]:=0.5;
> n:= 15;
> x[1] := x[0]-(f(x[0]))/(fd(x[0]))-
((f(x[0]))^(2)*fdd(x[0]))/ (2*(fd(x[0]))^(3)):
> for i from 1 to n do
> y[i]:=x[i]-(f(x[i]))/(fd((x[i-1]+y[i-1])/(2)))-
((f(x[i]))^(2)fdd(x[i]))/(2*(fd((x[i-1]+y[i-1])/(2)))^(3)):
> x[i+1] := x[i]-(f(x[i])) / (fd((x[i]+y[i])/(2)))
-((f(x[i]))^(2)*fdd(x[i])) /(2*(fd((x[i]+y[i])/(2)))^(3)):
> abserror[i]:=abs(x[i+1]-x[i]);
> absfun[i]:=abs(f(x[i+1]));
> if abs(x[i+1]-x[i]) < eps or abs(f(x[i+1])) < epsabs then
> break;
> end if:
> printf("Iteration No:  %g = %10.10f\\n", i,x[i+1]);
> end do;
```

1.10^{-8}

1.10^{-8}

$x \to x - 3\ln(x)$

$x \to 1 - \frac{3}{x}$

$x \to \frac{3}{x^2}$

0.5

0.5

15

$$1.430307717$$
$$1.850449158$$
$$0.515191954$$
$$0.004163965$$
$$\text{Iteration No:}1 = 1.8504491580$$
$$1.854014254$$
$$1.857200752$$
$$0.006751594$$
$$0.000010394$$
$$\text{Iteration No:}2 = 1.857183861$$
$$1.857183978$$
$$1.857183861$$
$$0.000016891$$
$$1.10^{-9}$$

5. CONCLUSION

In this paper, we proposed a three-step iterative algorithm for finding an approximate root of a given non-linear equation in one variable with the help of Householder's method and modified Newton's method. Maple implementation of the proposed algorithm is also discussed with sample computations. Couple of numerical examples are presented to illustrate and validation of the proposed method. Comparisons of the proposed method with existing methods are discussed and the results showed that the proposed algorithm required less number of iterations (see, for example, Table 1). As future work, one can use this algorithm for finding the approximate solution of the non-linear equations which have multiple solutions as well as for system of non-linear equations.

REFERENCES

[1] A.S. Housholder, The Numerical Treatment of a Single Nonlinear Equation, McGraw-Hill, New York, 1970.

[2] C. Chun, "Iterative methods improving Newton's method by the decomposition method," *Comput. Math. Appl*. vol. 50, no. 10-12, pp. 1559–1568, 2005

[3] J.F. Traub, Iterative Methods for Solution of Equations, Prentice-Hall, Englewood Cliffs, NJ, 1964.

[4] J.F. Traub, Iterative Methods for the Solution of Equations (Englewood Cliffs, NJ: Prentice-Hall, 1964).

[5] J.H. He, "A new iterative method for solving algebraic equations", *Applied Mathematics and Computation*, vol. 135, no. 1, pp. 81–84, 2005.

[6] M.A. Noor, K.I. Noor, W.A. Khan, F. Ahmad, "On iterative methods for nonlinear equations," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 128–133, 2006.

[7] M.A. Noor, F. Ahmad, "Numerical comparison of iterative methods for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 167–172, 2006.

[8] M.A. Noor, K.I. Noor, "Three-step iterative methods for nonlinear equations," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 322–327, 2006.

[9] M. Saqib, M. Iqbal, S. Ali, T. Ismaeel, "New fourth and fifth-order iterative methods for solving nonlinear equations," *Applied Mathematics*, vol. 6, no. 8, pp. 1220–1227, 2015.

[10] Saied, Liao, "A new modification of False-Position method based on homotopy analysis method," *Applied Mathematics and Mechanics*, vol. 29, no. 2, 223–228, 2008.

[11] Sagralo, Mehlhorn, Computing real roots of real polynomials, 2013.

[12] S. Hussain, V. K. Srivastav, S. Thota, "Assessment of Interpolation Methods for Solving the Real Life Problem," *International Journal of Mathematical Sciences and Applications*, vol. 5, no. 1, pp. 91–95, 2015.

[13] S. Thota, V.K. Srivastav, "Quadratically Convergent Algorithm for Computing Real Root of

Non-Linear Transcendental Equations," *BMC Research Notes*, vol. 11, no. 909, pp. 1–6,2018, https://doi.org/10.1186/s13104-018-4008-z.

[14] 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*, vol. 2, no. 11, pp. 729–735, 2014.

[15] S. Thota, "A New Root-Finding Algorithm Using Exponential Series," *Ural Mathematical Journal*, vol. 5, no. 1, pp. 83–90, 2019.

[16] S. Thota, T. Gemechu, "A New Algorithm for Computing a root of Transcendental Equations Using Series Expansion," *Southeast Asian Journal of Sciences*, vol. 7, no. 2, pp. 106–114, 2019.

[17] T. Gemechu, S. Thota, "On New Root Finding Algorithms for Solving Nonlinear Transcendental Equations," *International Journal of Chemistry, Mathematics and Physics*, vol. 4, no. 2, pp. 18–24, 2020.

[18] Sharma, J.R., Kumar, D, "On a reduced cost derivative-free higher-order numerical algorithm for nonlinear systems," *Comp. Appl. Math*. vol. 39, no. 202, 2020. https://doi.org/10.1007/s40314-020-01218-3.

[19] Argyros IK, Hilout S. Numerical Methods in Nonlinear Analysis. World Scientific Publishing Company, New Jersey, 2013.

[20] Constantinides A, Mostoufi N. Numerical Methods for Chemical Engineers with MATLAB Applications. Prentice Hall PTR, New Jersey, 1999.

[21] Moré JJ. A collection of nonlinear model problems. In, "Allgower EL, Georg K (eds) Computational Solution of Nonlinear Systems of Equations Lectures in Applied Mathematics," *American Mathematical Society*, Providence, RI, vol. 26, pp. 723–762, 1990.

[22] Ortega JM, Rheinboldt WC. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York, 1970.

[23] Traub JF. Iterative Methods for the Solution of Equations. Chelsea Publishing Company, New York, 1982.

[24] W.A. Khan, M.A. Noor, A. Rauf, "Higher-order iterative methods by using Householder's method for solving certain nonlinear equations," *Math. Sci. Lett.* vol. 2, no. 2, pp. 107–120, 2013.

[25] T.J. McDougall, S.J. Wotherspoon, "A simple modification of Newton's method to achieve convergence of order," *Journal of Applied and Mathematical letters*, vol. 29, no. 1, pp. 20–25, 2014.

[26] S.M. Kang, A. Rafiq, F. Ali, W Mahmood, "A New Newton's Type Method for Nonlinear Equations," International *Journal of Mathematical Analysis,* vol. 10, no. 4, pp. 151–158, 2016.

[27] A. Neamvonk, "Three-Step Method For Finding Root of Non-Linear Equations," *IOSR Journal of Mathematics*, vol. 15, no. 1, pp. 01–06, 2019.

[28] 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 (SUNRISE-19), (2019) (ISBN No. 978-93-5391-715-9).

[29] 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,*, vol. 5, no. 44, pp. 1-8, 2019.

BIOGRAPHIES OF AUTHORS

Dr. Srinivasarao Thota (Corresponding Author) completed his M.Sc. in Mathematics from Indian Institute of Technology (IIT) Madras and Ph.D. in Mathematics from Motilal Nehru National Institute of Technology (NIT) Allahabad, India. Srinivasarao Thota's areas of research interest are Computer Algebra (symbolic methods for differential equations), Numerical Analysis (root finding algorithms) and Mathematical Modelling (ecology). He has published his research work in several international journals and presented his work in different national and international conferences as oral presenter and expertise/invited presenter in different countries. Presently working at Department of Applied Mathematics, Adama Science and Technology University, Ethiopia. Further info on his homepage: https://srinivasaraothota.webs.com/