*Journal of AppliedMath* 2024, 2(1), 477.
https://doi.org/10.59400/jam.v2i1.477

Article

# A new optimal iterative algorithm for solving nonlinear equations

**Dhyan R. Gorashiya[1], Rajesh C. Shah[2,*]**

[1] Department of Metallurgical and Materials Engineering, Faculty of Technology & Engineering, The Maharaja Sayajirao University of Baroda, Vadodara 390001, Gujarat State, India
[2] Department of Applied Mathematics, Faculty of Technology & Engineering, The Maharaja Sayajirao University of Baroda, Vadodara 390001, Gujarat State, India
*** Corresponding author:** Rajesh C. Shah, dr_rcshah@yahoo.com

**CITATION**

Gorashiya DR, Shah RC. A new optimal iterative algorithm for solving nonlinear equations. Journal of AppliedMath. 2024; 2(1): 477. https://doi.org/10.59400/jam.v2i1.477

**ARTICLE INFO**

Received: 12 December 2023
Accepted: 29 January 2024
Available online: 28 February 2024



**Abstract:** The aim of this paper is to propose a new iterative algorithm (scheme or method) for solving algebraic and transcendental equations, considering a fixed point and an initial guess value on the *x*-axis. The concepts of the slope of a line and the Taylor series are used in the derivation. The algorithm has second-order convergence and requires two function evaluations in each step, which shows that it is optimal with a computational efficiency index of 1.414 and an informational efficiency of 1. The validity of the algorithm is examined by solving some examples and their comparisons with Newton's method.

**Keywords:** iterative algorithm; second-order convergence; nonlinear equations; Newton's method; optimal method

## 1. Introduction

Solving an equation $f(x) = 0$ always receives attraction for investigators due to its various applications in science, technology, and engineering. Many investigators [1–8] developed new iterative methods for better and faster convergence. Among various numerical methods, optimal methods draw significant attention due to their computational and informational efficiency. Any $n$-step iterative method is optimal when it reaches the order of convergence $2^n$ and requires $(n+1)$ function evaluations. As mentioned above, the most important parameters of computational efficiency and informational efficiency are usually expressed [9,10] by

$$E = r^{\frac{1}{c}} \text{ and } I = \frac{r}{c},$$

respectively, where $r$ is the order of convergence and $c$ is the computational cost (number of function evaluations in each step).

In this paper, a new optimal iterative algorithm (scheme or method) for solving an equation $f(x) = 0$ is proposed, wherein a fixed point $x_p$ and an initial guess value $x_0$ on the $x$-axis are considered. The concepts of the slope of a line and Taylor's series are used in the derivation. The algorithm is a single-step and has second-order convergence. The validity of the algorithm is examined by solving some examples and their comparisons with Newton's method. The algorithm works for both when $f'(x) \neq 0$ and $f'(x) = 0$.

## 2. The proposed optimal iterative algorithm

**Figure 1** shows a geometric view of the proposed algorithm for solving $f(x) = 0$. Let $\xi$ be the exact (simple) root in $I \subset R$, where $I$ is some open interval and $R$ is a set of real numbers. Let $f(x)$ and its first and second-order derivatives are continuous nearer to $\xi$. In order to develop a proposed algorithm, consider a fixed point $A(x_p, 0)$ and the initial guess value (if possible, sufficiently close to $\xi$) $B(x_0, 0)$ on the $x$-axis. Draw a perpendicular from $B$ to meet the curve $y = f(x)$ at $C(x_0, f(x_0))$ as shown in figure. Then for $0 < k < 1$, let $D(x_0, kf(x_0))$ be a point on inside the line segment $\overline{BC}$. Draw a line segment $\overline{AD}$. By considering the same slope $\alpha$ of $\overline{AD}$, draw another line segment $\overline{BE}$ giving the first approximation $x_1$ (say $F(x_1, 0)$) by the following procedure, where $x_1 = x_0 + h$.

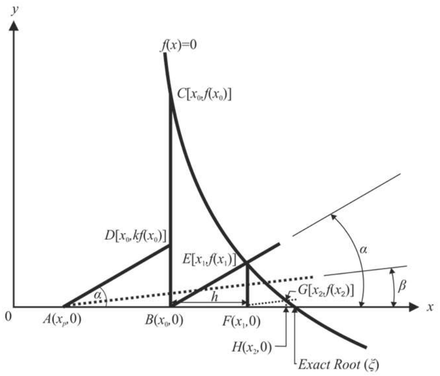


**Figure 1.** The proposed algorithm.

As discussed above, Slope of $\overline{AD}$ = Slope of $\overline{BE}$ implies

$$\frac{kf(x_0)}{x_0 - x_p} = \frac{f(x_1)}{x_1 - x_0} \tag{1}$$

Using $x_1 = x_0 + h$ and Taylor series $f(x_1) = f(x_0 + h) \approx f(x_0) + hf'(x_0)$, Equation (1) becomes

$$x_1 = \frac{(x_0 - x_p)[f(x_0) - x_0 f'(x_0)] + kx_0 f(x_0)}{kf(x_0) - (x_0 - x_p)f'(x_0)}.$$

The repetition of the above procedure gives general algorithm as

$$x_{n+1} = \frac{(x_n - x_p)[f(x_n) - x_n f'(x_n)] + kx_n f(x_n)}{kf(x_n) - (x_n - x_p)f'(x_n)}; 0 < k < 1, n = 0,1,2,\ldots. \tag{2}$$

This algorithm converges to the root $\xi$, provided $kf(x_n) - (x_n - x_p)f'(x_n) \neq 0$. Here, the value of $k$ is restricted to $0 < k < 1$ because of the following reasons.

(a) At $k = 0, \alpha = 0$ and the Algorithm (2) leads to Newton's formula.

(b) At $k = 1$, $\alpha$ takes the maximum possible angle, which may result in more iterations and delay in convergence.

It should be noted here that after the first iteration, due to $kf(x_1)$, there is a decrease in slope (say $\beta$, $\beta < \alpha$), which continues for further iterations as well.

## 3. Convergence analysis

In this article, the generalized convergence rate is considered similar to that of Newton's formula by Taylor series expansion.

Let $\varepsilon_n$ and $\varepsilon_{n+1}$ be the errors in the $n^{\text{th}}$ and $(n+1)^{\text{th}}$ iterations, respectively, then

$$\varepsilon_n = \xi - x_n,\ \varepsilon_{n+1} = \xi - x_{n+1}. \tag{3}$$

Using Equation (3), right-hand side of Algorithm (2) with Taylor's expansion up to $O(\varepsilon_n^2)$ and the fact that $f(\xi) = 0$, gives

$$\frac{(x_n - x_p)[f(x_n) - x_n f'(x_n)] + k x_n f(x_n)}{k f(x_n) - (x_n - x_p) f'(x_n)} = \frac{a + b\varepsilon_n + c\varepsilon_n^2}{d + e\varepsilon_n + g\varepsilon_n^2} = \frac{a + b\varepsilon_n + c\varepsilon_n^2}{d\left(1 + \frac{e}{d}\varepsilon_n + \frac{g}{d}\varepsilon_n^2\right)} = \frac{\frac{a}{d} + \frac{b}{d}\varepsilon_n + \frac{c}{d}\varepsilon_n^2}{1 + \left(\frac{e}{d}\varepsilon_n + \frac{g}{d}\varepsilon_n^2\right)}$$

$$= \left(\frac{a}{d} + \frac{b}{d}\varepsilon_n + \frac{c}{d}\varepsilon_n^2\right)\left[1 - \left(\frac{e}{d}\varepsilon_n + \frac{g}{d}\varepsilon_n^2\right) + \left(\frac{e}{d}\varepsilon_n + \frac{g}{d}\varepsilon_n^2\right)^2\right]$$

$$= \frac{a}{d} + \left(\frac{bd - ae}{d^2}\right)\varepsilon_n + \left(\frac{c}{d} - \frac{be}{d^2} - \frac{ag}{d^2} + \frac{ae^2}{d^3}\right)\varepsilon_n^2,$$

where

$$a = -\xi^2 f'(\xi) + \xi x_p f'(\xi),$$

$$b = \xi^2 f''(\xi) + \xi f'(\xi) - \xi x_p f''(\xi) - k\xi f'(\xi),$$

$$c = -\frac{3}{2}\xi f''(\xi) - \frac{1}{2}\xi^2 f'''(\xi) + \frac{x_p}{2} f''(\xi) + \frac{1}{2}\xi x_p f'''(\xi) + \frac{k\xi}{2} f''(\xi) + k f'(\xi),$$

$$d = -\xi f'(\xi) + x_p f'(\xi),$$

$$e = -k f'(\xi) + \xi f''(\xi) + f'(\xi) - x_p f''(\xi),$$

$$g = \frac{k}{2} f''(\xi) - \frac{\xi}{2} f'''(\xi) - f''(\xi) + \frac{x_p}{2} f'''(\xi).$$

Since

$$\frac{a}{d} = \xi \text{ and } \frac{bd - ae}{d^2} = 0,$$

Algorithm (2) becomes

$$\varepsilon_{n+1} \approx \varepsilon_n^2 \text{ (Finite quantity).}$$

Thus, the algorithm converges quadratically.

## 4. Results and discussion

From the above sections, it is noticed that the proposed algorithm is single-step, has second-order convergence, and uses two function evaluations in each step with the computational efficiency index $E = 2^{1/2} = 1.414$ and informational efficiency $I = \frac{2}{2} = 1$, which is same as Newton's method. Thus, the proposed algorithm for solving an equation $f(x) = 0$ is optimal, as per the definition given in the introduction section.

The comparative study between the proposed Algorithm (2) and Newton's method is given in **Tables 1–6**, where $|x_{n+1} - x_n| < 0.00001$.

**Table 1.** Comparative study of the proposed Algorithm (2) with Newton's method for $x^3 - x^2 - x - 1 = 0$ considering different values of $x_p$, and same values of $x_0$ and $k$.

| **(1) Initial guess value $x_0$** | **(2) Value of $x_p$** | **(3) Value of $k$** | **(4) Number of iterations required by the proposed Algorithm (2)** | **(5) Number of iterations required by Newton's method** | **(6) Solution of the given equation by the proposed Algorithm (2)** | **(7) Solution of the given equation by Newton's method** |
|---|---|---|---|---|---|---|
| 1.5 | −3.0<br>−1.5<br>3.0<br>6.0 | 0.00001 | 5 | 5 | 1.83928675521416 | 1.83928675521416 |

**Table 2.** Comparative study of the proposed Algorithm (2) with Newton's method for different equations considering the same values of $x_0$, $x_p$ and $k$.

| (1) Equations | (2) Initial guess value $x_0$ | (3) Value of $x_p$ | (4) Value of $k$ | (5) Number of iterations required by the proposed Algorithm (2) | (6) Number of iterations required by Newton's method | (7) Solution of the given equations by the proposed Algorithm (2) | (8) Solution of the given equations by Newton's method |
|---|---|---|---|---|---|---|---|
| $x^3 - x^2 - x + 1 = 0$ | 1.5 | 0.5 | 0.00001 | 16 | 16 | 1.000009413559562 | 1.000009413617196 |
| $4x^4 - 4x^2 = 0$ | $\frac{\sqrt{21}}{7}$ | 1.65 | | 19 | 26 | 0.000007441381281 | 0.000007484300647 |
| $x^3 - e^{-x} = 0$ | 0 | −1 | | 5 | 5 | 0.772882959152200 | 0.772882959152202 |
| $\sin x = 0$ | 1.5 | 0.5 | | 4 | 4 | −12.566370614359171 | −12.566370614359172 |
| $x^{10} - 1 = 0$ | 0.5 | −0.5 | | 43 | 43 | 1.000000000000000 | 1.000000000000000 |
| $e^{x^2+7x-30} - 1 = 0$ | 3.5 | 2 | | 11 | 11 | 3.000000000000253 | 3.000000000000253 |
| $x^3 + x^2 - 2 = 0$ | 2.2 | 0 | | 6 | 6 | 1.000000000000000 | 1.000000000000000 |
| $(x-2)^{23} - 1 = 0$ | 3.5 | −4 | | 13 | 13 | 3.000000000022981 | 3.000000000022981 |

**Table 3.** Comparative study of the proposed Algorithm (2) with Newton's method for $xe^{x^2} - \sin^2 x + 3\cos x + 5 = 0$ considering different values of $x_0$ and the same values of $x_p$ and $k$.

| (1) Equation | (2) Initial guess value $x_0$ | (3) Value of $x_p$ | (4) Value of $k$ | (5) Number of iterations required by the proposed Algorithm (2) | (6) Number of iterations required by Newton's method | (7) Solution of the given equation by the proposed Algorithm (2) | (8) Solution of the given equation by Newton's method |
|---|---|---|---|---|---|---|---|
| $xe^{x^2} - \sin^2 x + 3\cos x + 5 = 0$ | −1 | 0 | 0.00001 | 5 | 5 | −1.207647827130919 | −1.207647827130919 |
| | −2 | | | 7 | 7 | −1.207647827173526 | −1.207647827173531 |

**Table 4.** Comparative study of the proposed Algorithm (2) with Newton's method for $\sin x = 0$ considering different values of $k$ and the same values of $x_0$ and $x_p$.

| (1) Initial guess value $x_0$ | (2) Value of $x_p$ | (3) Value of $k$ | (4) Number of iterations required by the proposed Algorithm (2) | (5) Number of iterations required by Newton's method | (6) Solution of the given equation by the proposed Algorithm (2) | (7) Solution of the given equation by Newton's method |
|---|---|---|---|---|---|---|
| 1.5 | 0.5 | 0.5 | 6 | 4 | 3.141592653589793 | −12.566370614359172 |
| | | 0.25 | 5 | | 6.283185307179575 | |
| | | 0.125 | 7 | | 18.849555921538759 | |
| | | 0.0625 | 6 | | −116.238928182822363 | |
| | | 0.03125 | 7 | | −31.415926535897924 | |
| | | 0.015625 | 5 | | −15.707963267948967 | |
| | | 0.0078125 | 5 | | −18.849555921538759 | |
| | | 0.00390625 | 5 | | −12.566370614359171 | |
| | | 0.001953125 | 4 | | −12.566370614359164 | |
| | | 0.0009765 | 4 | | −12.566370614359172 | |
| | | 0.00001 | 4 | | −12.566370614359171 | |

**Table 5.** Comparative study of the proposed Algorithm (2) with Newton's method for $x^{10} - 1 = 0$ considering different values of $k$ and same values of $x_0$ and $x_p$.

| (1) Initial guess value $x_0$ | (2) Value of $x_p$ | (3) Value of $k$ | (4) Number of iterations required by the proposed Algorithm (2) | (5) Number of iterations required by Newton's method | (6) Solution of the given equation by the proposed Algorithm (2) | (7) Solution of the given equation by Newton's method |
|---|---|---|---|---|---|---|
| | | 0.5 | 13 | | 1.000000000005467 | |
| | | 0.25 | 19 | | 1.000000000000000 | |
| | | 0.125 | 24 | | 1.000000000000095 | |
| | | 0.0625 | 29 | | 1.000000000002376 | |
| | | 0.03125 | 34 | | 1.000000000000000 | |
| 0.5 | −0.5 | 0.015625 | 37 | 43 | 1.000000000000537 | 1.000000000000000 |
| | | 0.0078125 | 39 | | 1.000000000226450 | |
| | | 0.00390625 | 41 | | 1.000000000000017 | |
| | | 0.001953125 | 42 | | 1.000000000000000 | |
| | | 0.0009765 | 42 | | 1.000000000002626 | |
| | | 0.00001 | 43 | | 1.000000000000000 | |

**Table 6.** Comparative study of the proposed Algorithm (2) with Newton's method for $x^3 + x^2 - 2 = 0$ with the same values of $x_0$, $x_p$ and $k$.

| (1) Initial guess value $x_0$ | (2) Value of $x_p$ | (3) Value of $k$ | (4) Number of iterations required by the proposed Algorithm (2) | (5) Number of iterations required by Newton's method | (6) Solution of the given equation by the proposed Algorithm (2) | (7) Solution of the given equation by Newton's method |
|---|---|---|---|---|---|---|
| 0 | 2 | 0.00001 | 36 | Fails | 1.000000000020332 | Fails |

**Table 1** shows the result for $x^3 - x^2 - x - 1 = 0$ considering different values of $x_p$ and the same values of $x_0$ and $k$, where the value $x_0 = 1.5$ is chosen by using the intermediate value theorem (IVT). It should be noted here that the use of IVT is not necessary. The table shows that the proposed algorithm gives the same result as Newton's method correct up to at least fourteen decimal places with the same number of iterations when $k = 0.00001$.

**Table 2** shows results for different equations considering the same values of $x_0$, $x_p$ and $k$ ($k = 0.00001$).

**Tables 1** and **2** show that for this value of $k$, the proposed algorithm has almost good agreement of results with Newton's method.

**Table 3** shows results for $xe^{x^2} - \sin^2 x + 3\cos x + 5 = 0$ considering different values of $x_0$ and the same values of $x_p$ and $k$. The table shows that a change in $x_0$ leads to a change in the number of iterations for getting the same result by both proposed and Newton's methods.

**Table 4** shows results for $\sin x = 0$ considering different values of $k$ and the same values of $x_0$ and $x_p$. This table shows the case of multiple roots of an equation. It is observed that the proposed algorithm gives different roots for different values of $k$ with the variation in the number of iterations.

**Table 5** shows results for $x^{10} - 1 = 0$ considering different values of $k$, and the same values of $x_0$ and $x_p$. Here, it is observed that the proposed algorithm gives the same root as that of Newton's method with the variation in the number of iterations.

**Table 6** shows results for $x^3 + x^2 - 2 = 0$ with the same values of $x_0$, $x_p$ and $k$. It is observed that the proposed algorithm works even if $f'(x) = 0$, which is the limitation of the methods suggested by authors [1–7]. Of course, Newton's method is also unable to give a solution, but according to Traub [10], in such cases, it converges linearly.

## 5. Conclusions

This paper proposes a new single-step, second-order optimal iterative algorithm for finding the root of an equation $f(x) = 0$, wherein the concepts of the slope of a line and Taylor's series are used in the derivation. The algorithm has a computational efficiency index of 1.414 and an informational efficiency of 1, and it works for both $f'(x) \neq 0$ and $f'(x) = 0$. Usually, the condition $f'(x) = 0$ is the limitation of the methods suggested by authors [1–7]. The algorithm may be considered as generalization of Newton's method, involving two parameters $k$ and $x_p$. The proposed algorithm is valid if

$$kf(x_n) - (x_n - x_p)f'(x_n) \neq 0.$$

If $kf(x_n) - (x_n - x_p)f'(x_n) = 0$, then

$$kf(x_n) = (x_n - x_p)f'(x_n). \tag{4}$$

As $0 < k < 1$, and $x_n$ and $x_p$ are two different points, so $k \neq 0$ and $(x_n - x_p) \neq 0$. Hence, $kf(x_n) = (x_n - x_p)f'(x_n)$ is satisfied when $f(x_n) = 0$ and $f'(x_n) = 0$ simultaneously.

Again, rearranging Equation (4) implies

$$\frac{k}{x_n - x_p} = \frac{f'(x_n)}{f(x_n)} \tag{5}$$

The satisfaction of the above condition is dependent on $k$ and $x_p$. If, at any stage of iteration, Equation (5) is satisfied, then the proposed algorithm fails, but it happens rarely.

**Author contributions:** Conceptualization, DRG and RCS; methodology, DRG and RCS; software, DRG and RCS; validation, DRG and RCS; formal analysis, DRG and RCS; investigation, DRG and RCS; data curation, DRG and RCS; writing—original draft preparation, RCS; writing—review and editing, DRG and RCS. All authors have read and agreed to the published version of the manuscript.

**Conflict of interest:** The authors declare no conflict of interest.

## References

1. Ujević N. A method for solving nonlinear equations. Applied Mathematics and Computation. 2006, 174(2): 1416-1426. doi: 10.1016/j.amc.2005.05.036
2. Sharma JR. A one-parameter family of second-order iteration methods. Applied Mathematics and Computation. 2007, 186(2): 1402-1406. doi: 10.1016/j.amc.2006.07.140
3. Saeed RK, Aziz KM. An iterative method with quartic convergence for solving nonlinear equations. Applied Mathematics and Computation. 2008, 202(2): 435-440. doi: 10.1016/j.amc.2008.02.037
4. Maheshwari AK. A fourth order iterative method for solving nonlinear equations. Applied Mathematics and Computation. 2009, 211(2): 383-391. doi: 10.1016/j.amc.2009.01.047

5. Singh MK. A Six-order Variant of Newton's Method for Solving Nonlinear Equations. ICHB PAS Poznan Supercomputing and Networking Center. Published online 2009. doi: 10.12921/CMST.2009.15.02.185-193
6. Thukral R. A new eighth-order iterative method for solving nonlinear equations. Applied Mathematics and Computation. 2010, 217(1): 222-229. doi: 10.1016/j.amc.2010.05.048
7. Matinfar M, Aminzadeh M. An iterative method with six-order convergence for solving nonlinear equations. International Journal of Mathematical Modeling and Computations. 2012, 2(1): 45-51.
8. Saheya B, Chen G, Sui Y, et al. A new Newton-like method for solving nonlinear equations. SpringerPlus. 2016, 5(1). doi: 10.1186/s40064-016-2909-7
9. Ostrowski AM. Solution of equations and systems of equations. Academic Press, New York. 1960.
10. Traub JF. Iterative methods for the solution of equations. Prentice-Hall, Englewood Cliffs, New Jersey. 1964.