International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 05 – Issue 12 || December 2020 || PP. 01-12

A Simple IterativeRoot Finding Algorithm for Nonlinear Equations

Chaman Lal Sabharwal
Computer Science Department,
Missouri University of Science and Technology,
Rolla, MO 65409, USA

Abstract: Locating and determining the roots of non-linear and transcendental equations is a fundamental problem in various engineering fields, including social and physical sciences. In general, suchproblems do not have an analytic solution,the researchers resort to numerical techniques for exploring and then exploiting. There is not a single algorithm that works best for every function. We design and implement a hybrid algorithm that is a blend of the Bisection and Regula Falsi algorithms. The hybrid algorithm is a new single pass iterative approach. The method does not use any predictor–corrector technique, but in each iterationpredicts a better estimate closer to the root. It is observed that the new algorithm outperforms the Bisection,Regula Falsi, Newton – Raphson, the quadrature based, undetermined coefficients based, and decompositionbased algorithms. The blended new root finding algorithm is guaranteed to converge to a root. The theoretical and empirical evidence shows that the computational complexity of the blended algorithm is considerably less than that of the classical algorithms. Several functionscited in theliterature are used as benchmark to compare and confirm the simplicity, efficiency and performance of the proposed method.

Keywords: Bisection, Secant, Newton method, Order of convergence, Predictor-Corrector method, Quadrature method

1 Introduction

The modeling of numerical solution to non-linear equationsis required in many branches of science and engineering. Since the non-linear problems do not have analytic solutions, the solution developers resort to numerical methods. More and more efficient methods for solving the nonlinear equations are evolving frequently, and areubiquitously explored and exploited in applications. Thepurpose is to improve the existing methods, such asclassical Bisection, False Position, Newton Raphson and their variant methodsfor efficiency, accuracy and simplicity in the solution ofnon-linear equations.

There are various ways to approach a problem, some functions require only continuous functions while others require differentiability of functions. The classical root-finding algorithms such as Bisection, False Position, Newton–Raphson, and Secant method are ubiquitous. There are improvements of these algorithms for speedup, more elegant and efficient implementations are emerging. The variations of continuity basedBisection and False Position method are due to Dekker[1] , [2] , and Brent [3] [4] and alternate variants of False Position. The variations ofderivative based quadratic orderNewton–Raphsonmethod are 3^{rd}, 4^{th}, 5^{th}, 6^{th} order methods. From these algorithms, the developer has to explore and exploit the algorithm suitable under specified constraints on the function and the domain. There is no single algorithm that works best for every function [5] , [6].

For derivative based methods, several predictor-corrector solutions have been proposedforextending Newton-Raphson methodwith Simpsons quadrature formula[7] , undetermined coefficients[8] ,a *thirdorder* Newton-type method to solve system of nonlinear equations [9] , Newton's method with accelerated convergence [10] using trapezoidal quadrature, *fourth order*method of undetermined coefficients [11] ,one derivative and two function evaluation [12] ,Newton's Method using *fifthorder* quadrature formulas [13] , using Midpoint, Trapezoidal, and Simpson quadrature *sixth* order rule [14] . The hybrid algorithm presented here does not use any variations or quadrature or method of undetermined coefficient, still competes with all these algorithms.

Even though the classical methods have been developed and used for decades, enhancements are progressively made to improve the performance of these methods. A method may outperform other methods on one dataset, and may produce inferior results on another dataset. Different methods have their own strengths/weaknesses. A dynamic new hybrid algorithm (blend of Bisection and False Position) is presented here by taking advantage of the best in the Bisection and False Position methods to locate the roots independent of Dekker and Brent approach. Its computational efficiency is validated by comparing it with the existing methods via complexity analysis and empirical evidence. We present a new hybrid algorithm that outperforms

all the existing algorithms, see Section 4 for empirical outcomes validating the performance of the new algorithm.

This paper is organized as: Section 2 isvery briefdescription of background methods, Section 3 is new algorithm, Section 4 is empirical evidence with experimental results and analysis, Section 5 is future work no end insight, Section 6 is conclusion followed by references.

2 Background

There are several classical methods for finding roots of non-linear equations. For completeness, we briefly describe methods for (1) root approximation, (2) error calculation and (3) termination criteria. There is no single optimal algorithm for root approximation. Normally, when faced with competing choices, the simplest one is the accurate one. This is particularly true in this case. We will provide a simpler new algorithm that outperforms all these methods. For background, we will briefly refer to the equations requiring (1) only continuous functions and (2) those requiringdifferentiable functions along with the desired order of derivative in their formulations.

There are two types of problems: (1) continuity based with no derivative requirement, such as Bisection, False Position and their extensions, (2) derivative based such as Newton-Raphson and its variations. For simulations, we set error Tolerance $\varepsilon = 0.0000001$, iterations terminate when $(|x_k - x_{k-1}| + |f(x_k)|) < \varepsilon$. For function f:[a, b] $\rightarrow$ R, is such that (1) f(x) is continuous on the interval [a, b], where R is the set of all real numbers and

(2) f(a) and f(b) are of opposite signs, i.e., $f(a) \cdot f(b) < 0$, then there exists a root $r \in [a, b]$ such that f(r) = 0

or

(2') the function f(x) is differentiable and |f'(x)<1, then there exists a root $r \in [a, b]$ such that f(r) = 0.

Since the solution is obtained by iterative methods,thedefinition of order of convergence is as follows[10] Letx_n, $a \in R$ $n \geq 0$. Then the sequence $\{x_n\}$ is said to converge to$a$ if

$$\lim_{n \to \infty} |x_n - a| = 0$$

If, in addition, there exist a constant c >0, an integer N≥0 and p≥0 such that for all$n >0$,

$$|x_{n+1} - a| \leq c|x_n - a|^p$$

Thenthen the sequence $\{x_n\}$ is said to converge to$a$ with order p.If p = 2 or 3, the convergence is said to be quadratic or cubic respectively.

2.1 Bisection Method

The Bisection method (1) is based on binary slashing of irrelevant subintervals, (2) is virtualiyabinary search, and (3) is guaranteed to converge to the root. It does not matter what the function is, the rooterror upper bound is fixed at each iteration and can be determined a priori. By specifying the root-error tolerance,the upper bound on the number of iterations can be predetermined quickly.

2.2. False Position (Regula Falsi) Method

The motivation for innovative methods arises from the poor performance of the Bisection method. One suchmethod to find zeros of the function is to use linear interpolation. The False Position is a dynamic method; it takes advantage ofthe location of the root to make a conceivably judicious appropriate selection. Unfortunately, this method does notperform as well as expected, for all functions, see Figure 1. Here False Position method performs poorly as compared to Bisection method.

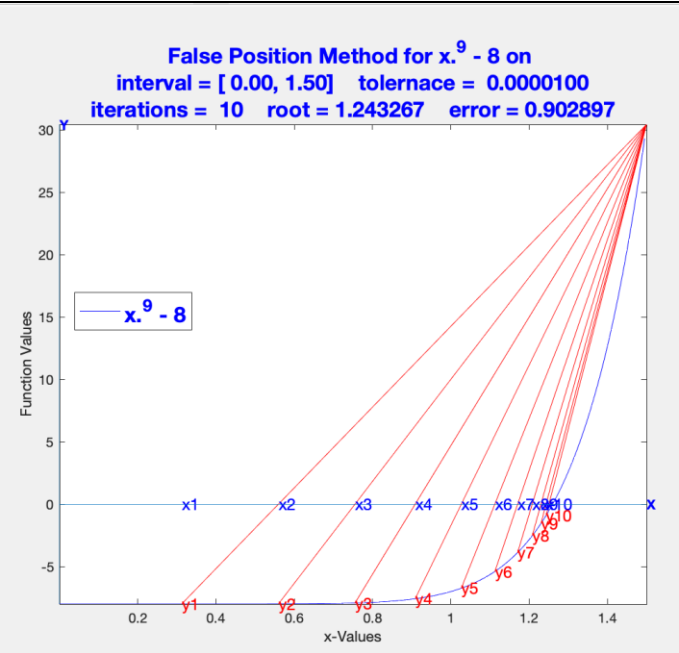

Figure 1. Convex Function Concave down, right end point fixed

2.3 Dekker's Method

Dekker (1969) algorithm was designed to circumvent the shortcomings of slow speed of Bisection anduncontrolled iteration count of False Position method. It is a combination of two methods: Bisection, FalsePosition[2] . Though it is better than the False position method, yet it has some road blocks, handled by Brent next.

2.4 Brent's Algorithm

Brent's[3] algorithm evolved from the failures of Dekker's algorithm, however it is a step towardsimprovement of Dekker's algorithm. It is a slight improvement at the cost of extra test computations. Itis a combination of four methods, Bisection, False Position, Dekker, and reverse (a.k.a.inverse)quadratic interpolation (RQI). Reverse quadratic interpolation is based on the method of Lagrangepolynomials and it leads to extra calcualtions. Thus, it may take more iterations to circumventpathological cases. It uses three estimates to derive the next estimate. This algorithm is more robust and more costly.

The derivative based methods depend on initial start point x_0. The simplest such technique is the Newton-Raphson method which is slower than its variations. These algorithms outperform the conventional Newton-Raphson method. The variations include decomposition based [15] , quadrature based[7], [13], [14], , undetermined coefficients based [8], [11] . These methods are quite complex and detailed. The readermay wish to refer to thefull papers for details. For completeness, we describe the iteration formulasto show how these methods operate to get to the root.

2.5 Newton-Raphson

Let f be differentiable on the open interval (a, b).Thedirect Newton-Raphson method has quadratic order of error , namely, is $O(\epsilon^2)$.

By Taylor series expansion of f retaining first derivative term only, linear approximation of order one

$f(b) \cong f(a)+(b-a)f'(a)$

approximation of b to the root of f further leads to

$0 \cong f(a)+(b-a)f'(a)$

$b = a - \frac{f(a)}{f'(a)}$ which is standard Newton-Raphson method.

Thus for function f, Newton-Raphsonsuccessive estimates for the solution are

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

with quadratic order of error, $O(\epsilon^2)$, where $\epsilon = |x_{n+1} - x_n|$.

This amounts two function evaluations for each iteration. However, if write

$g(x) = x - \frac{f(x)}{f'(x)}$

then $x_{n+1} = g(x_n)$
amounts one function evaluation for each iteration. It is just a matter of how we count the number of function evaluations.

2.6 Oghovese–Emunefe Method(2014)

The Oghovese– Emunefe[7] developed a **sixth order**method that approximates the iteration estimates by using the average of Midpoint and Simpson quadrature with and is shown to have approximation accuracy of order six.

The expression for Oghovese– Emunefe's estimates [7] is

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

where

$u_0 = v_0 = x_o$

$u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}$ (2)

$v_{n+1} = v_n - \frac{12\, f(v_n)}{f'((v_n)+10\, f'\left(\frac{(v_n+u_{n+1})}{2}\right)+f'(u_{n+1})}$ (3)

(based on average of Midpoint and Simpsons 3/8 rule)

Then the iterates, x_{n+1},are defined in terms of $v_n$instead of x_n

$x_{n+1} = v_n - \frac{f(v_n)}{f'(v_n)}$ for n≥0 (4)

instead of Newton-Raphson formula$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

The derivation ofOghovese– Emunefe [7]method uses the average of Midpoint and Simpson quadratures.We briefly describe its derivation and will defer the details of derivation for other methods to the references.The reader may refer to the full paper for details of the order of convergence. If the function is sufficiently differentiable, the order of error is shown to be $O(\epsilon^7)$ where ϵ = (computed – true value) error in the solution. Convergence analysis is quite detailed.By fundamental theorem of integration,

$I = \int_a^b f'(t)\, dt = f(b) - f(a)$

For approximating b as a root, that is, f(b) ≅0,

$$\int_a^b f'(t)\, dt \cong -f(a)$$

Using integration by standard quadrature forms,

By Midpoint rule

$(b-a)f'(\frac{a+b}{2}) \cong -f(a)$ (5)

$b \cong a - \frac{f(a)}{f'(\frac{a+b}{2})}$

By *Trapezoidal* rule

$(b-a)\{f'(a)+f'(b)\}/2 \cong -f(a)$

$b \cong a - \frac{2\, f(a)}{\{f'(a)+f'(b)\}}$

By Simpson quadratic approximation 3/8 rule

$\int_a^b f'(t)\, dt = f(b) - f(a)$

$(\frac{b-a}{6})\{f'(a)+4\, f'(\frac{a+b}{2})+f'(b)\} \cong -f(a)$ (6)

$(b-a) \cong -\frac{6\, f(a)}{f'(a)+4\, f'(\frac{a+b}{2})+f'(b)}$

$b \cong a - \frac{6\, f(a)}{f'(a)+4\, f'(\frac{a+b}{2})+f'(b)}$

By applying the weighted average approximation rule on (5) and (6), using Midpoint and Simpson 3/8 rulefor M and S,

$-f(a) \cong \int_a^b f'(t)\, dt = (1-p)M+pS$ with $0 \le p \le 1$

With $p=1/2 - f(a) \cong 1/2\ (b-a)\ f'(\frac{a+b}{2}) + 1/2\ (\frac{b-a}{6})\{f'(a)+4\, f'(\frac{a+b}{2})+f'(b)\}$

becomes

$b \cong a - \frac{12\, f(a)}{f'(a)+10\, f'\left(\frac{a+b}{2}\right)+f'(b)}$

Now the method proceeds to predict-correct the estimate as follows.

Using the predicted value $\bar{b} \cong a - \frac{f(a)}{f'(a)}$ in the right above, to correct it,

Ogbereyivweet.al have first [7] approximationcomputed as follows

$$\bar{b} \cong a - \frac{f(a)}{f'(a)} \qquad \text{(predicted)}$$

$$\bar{\bar{b}} \cong a - \frac{12\, f(a)}{f'(a) + 10\, f'\left(\frac{a+\bar{b}}{2}\right) + f'\left(\bar{b}\right)}$$

Finally, (to further correct the predicted (corrected) value, we use approximation , $\bar{\bar{b}}$, as the iterate value instead of Newton-Raphson techniques value , a, we have

$$\bar{\bar{\bar{b}}} \cong \bar{\bar{b}} - \frac{f(\bar{\bar{b}})}{f'(\bar{\bar{b}})}$$

Algorithm (Ogbereyivwe2014**)**

Let f, x_0 be given,Newton-Raphson approximationiterates become

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

Modifying it to achieve faster Algorithm

Initialize $u_0=v_0=x_0$. $0 \le p \le 1$

$$u_{n+1} \cong u_n - \frac{f(u_n)}{f'(u_k)} \qquad (7)$$

using this we get

$$v_{n+1} \cong v_n - \frac{12\, f(v_n)}{f'(v_n)+10\, f'\left(\frac{v_n+u_{n+1}}{2}\right)+ f'(u_{n+1})} \qquad (8)$$

and finally

$$x_{n+1} = v_{n+1} - \frac{f(v_{n+1})}{f'(v_{n+1})} \qquad n \ge 0 \qquad (9)$$

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

The results of this algorithm are given inTable5,Table 6

From now on, we will describe the final iteration relations without derivations for simplicity. It is a historical perspective of development of innovations in the improvement of Newton-Raphson method and its variants for solution to non-linear equations [8-15].

2.7Grau-Diaz-Barero(2006)

Grau et.al.[16] introduced a new method by correcting Ostrowski's method. That method useone derivative and two function evaluation [Ostrowski1966]

The method has a **sixth order** convergence with the following iterations:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (10)$$

$$z_n = y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n)-2f(y_n)} \qquad (11)$$

$$x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n)-2f(y_n)} \qquad (12)$$

The method defined in equation (10)-(12) is called the method MG.

2.8 Sharma-Guha (2007)

Sharmaand Guha [17] modified and parameterized Ostrowski's method having four function evaluations and a **sixth order** convergence. Their formula is given as follows:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (13)$$

$$z_n = y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n)-2f(y_n)} \qquad (14)$$

$$x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n)+af(y_n)}{f(x_n)+bf(y_n)} \qquad (15)$$

where a and b are problem dependent parameters, with $b = a - 2$.

The method defined in equation (13-(15) is called the method MSh.

2.9Khattri-Abbasbandy(2011)

Khattri-Abbasbandy [11] introduced an iterative method having a **fourth order** convergence with the following formula:

$$y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)} \quad (16)$$

$$x_{n+1} = y_n - [1 + \frac{21}{8}\frac{f'(y_n)}{f'(x_n)} + \frac{-9}{2}(\frac{f'(y_n)}{f'(x_n)})^2 + \frac{15}{8}(\frac{f'(y_n)}{f'(x_n)})^3]\frac{f(x_n)}{f'(x_n)} \quad (17)$$

2.10 Fang-Chen-Tian-Sun-Chen (2011)

Fanget.al. [18]modified the Newton's method with five evaluation functions and produced a **sixth order** convergence method having the following iterations:

$$y_n = x_n - \frac{f(x_n)}{a_n f(x_n) + f'(x_n)} \quad (18)$$

$$z_n = y_n - \frac{f(y_n)}{b_n f(y_n) + f'(y_n)} \quad (19)$$

$$x_{n+1} = z_n - \frac{f(z_n)}{a_n f(z_n) + f'(z_n)} \quad (20)$$

where is a_n, b_n, $c_n$are real numbers chosen in such a way that $0 \le |a_n|, |b_n|, |c_n| \le 1$, and

$$\text{sign}(a_n f(x_n)) = \text{sign}(f'(x_n)), \text{sign}(b_n f(y_n)) = \text{sign}(f'(y_n)),$$
$$\text{sign}(c_n f(z_n)) = \text{sign}(f'(z_n)),$$

where $n = 1,2, \ldots$, and sign(x) is a sign function.
The method defined in equation (18)-(20) is called the method MF, see Table 6.

2.11 JayaKumar(2013)

Jayakumar [14] Generalized Simpson-Newton's Method for Solving Nonlinear Equations with Cubic or **third order**Convergence.Let a be a root of the function (1)and suppose that$x_{n-1}, x_n, x_{n+1}$are three successive iterations closer to the root a. This is a third order convergence formulation. Recall the Midpoint-Simpson iteration formula

$$x_{n+1} = x_n - \frac{6\, f(x_n)}{f'(x_n) + 4\, f'(\frac{x_n + y_n}{2}) + f'(y_n)} \quad (21)$$

There is no end in sight from the extensions, the arithmetic mean$\frac{f'(x_n)+f'(y_n)}{2}$is implicit equation (21). Harmonic-Simpson-Newton's method (HSN): After dividing numerator and denominator in equation (21) by 2 and using harmonic mean instead of arithmetic mean$\frac{f'(x_n)+f'(y_n)}{2}$, we obtain Harmonic-Simpson-Newton's method

$$x_{n+1} = x_n - \frac{3\, f(x_n)}{\frac{2\, f'(x_n) f'(y)}{f'(x_n) + + f'(y_n)} + 2\, f'(\frac{x_n + y_n}{2})} \quad (22)$$

2.12 Nora- Imran-Syamsudhuha(2018)

This is a very interesting and complex analysis paper[8], the order of convergence is still **six**. In this article, the authors present a combination of the Khattri and Abbasbandy[11]method with Newton's method using the principle of an undetermined coefficient method.

Furthermore, using the **fourth order** method derived by Khattri and Abbasbandy , they propose the following iterative method.

$$y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)} \quad (23)$$

$$z_n = y_n - [1 + \frac{21}{8}\frac{f'(y_n)}{f'(x_n)} + \frac{-9}{2}(\frac{f'(y_n)}{f'(x_n)})^2 + \frac{15}{8}(\frac{f'(y_n)}{f'(x_n)})^3]\frac{f(x_n)}{f'(x_n)} \quad (24)$$

$$x_{n+1} = z_n - \frac{ab(a-b)f(z_n)}{mf'(x_n) - a^3 f'(y_n) + 2b(2a-b)(f(z_n) - f(x_n))'}] \quad (25)$$

with m = a(b^2 – 3ab + a^2), a = $z_n - x_n$, b = $y_n - x_n$. The method defined in equation (23)-(25) is called the method MKA in Table 6.

2.12 Weerakonet.al. (2000)

Weerkoon-Fernando [10] used trapezoidal quadrature to achieve t**hird order** convergence with the following iteration forms

Trapezoid rule is

(b-a){f'(a)+f'(b)}/2≅ -f(a)

$$b \cong a - \frac{2\,f(a)}{\{f'(a)+f'(b)\}} \quad (26)$$

Weerakoon-Fernando formula becomes

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} \quad (27)$$

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

$$x_{n+1} = x_n - \frac{f(x_n)}{(f'(x_n)+f'(y_n))/2} \quad (28)$$

This is also called *Arithmatic mean formula*. If arithmetic mean is replaced with Harmonic mean, another variation called is *Harmonic Trapezoidal Formula*

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{2}{(\frac{1}{f'(x_n)}+\frac{1}{f'(y_n)})}} \quad (29)$$

$$x_{n+1} = x_n - \frac{\frac{1}{2}f(x_n)(f'(x_n)+f'(y_n))}{(f'(x_n)f'(y_n)} \quad (30)$$

This completes ourdiscussion of the 3rd, 4th, 5th, 6th order derivative basedformulas.

3Hybrid Algorithm

The new algorithm competes with the algorithms presented in the previous section. This algorithm guarantees the successful resolution of roots of non-linear equations. Other algorithms like Newton–Raphson and Secant methods have quadratic convergence, but are not guaranteed to converge unless additional constraints are imposed on the functions and the initial guess is close to the root. Thus it has reliability of Bisection method. This algorithm does the get stranded as many cited algorithms into complexity of computations resulting in slow speed. It has the speed of False Position method which works all the time. It differs from all the previous algorithms by tracking the best bracketing interval in addition to the best root approximation. Instead of brute force application of the Bisection or False Position method solely, we selectively apply the most relevant method at each step to redefine the approximate root and bracketing interval. Thus, we construct a new hybrid algorithm that performs better than the Bisection and False Position methods. In addition it also outperforms the Newton–Raphson and its variants for continuous functions as long as the $f(a)\cdot f(b) < 0$ is determined at the end points of the defining interval. At each iteration, the root estimate is computed from both mid point and secant point, and the better of the two is selected for the next approximation, in addition the common enclosing interval is updated to the new smaller interval. This prevents the unnecessary iterations performed in either method. This method does not require differentiability of the function as required by Newton–Raphson and its variants. This the simplest, reliable, fast algorithm

Hybrid Algorithm

Output: root r, k-number of iterations used , bracketing interval $[a_{k+1}, b_{k+1}]$

//initialize

$k = 0$; $a_1 = a$, $b_1 = b$

Initialize bounded interval for bisection and false position

fa is false positiona,ba is bisection method a

$fa_{k+1}=ba_{k+1}=a_1$;$fb_{k+1}=bb_{k+1}=b_1$

repeat

$fa_{k+1}=ba_{k+1}=a_k$;$fb_{k+1}=bb_{k+1}=b_k$

compute the mid point

$m=\frac{a_k+b_k}{2}$, and $\in_m = |f(m)|$

compute the False Position secantline point,

$s = a_k - \frac{f(a_k)\,(b_k-a_k)}{f(b_k)-f(a_k)}$ and $\in_f = |f(s)|$

$r = s$

$\in_a = \in_f$

if $|f(m)| < |f(s)|$,

$f(m_k)$ is closer to zero, Bisection method determines bracketing interval $[ba_{k+1}, bb_{k+1}]$

$r = m$

$\in_a = \in_m$

if $f(a_k)\cdot f(r) > 0$,

$ba_{k+1} = r$; $bb_{k+1} = b_k$;

else
$ba_{k+1} = a_k$; $bb_{k+1} = r$;
else
$f(s_k)$ is closer to zero, False Position method determines bracketing interval $[fa_{k+1}, fb_{k+1}]$
r= s
$\in_a = \in_f$
if $f(a_k)\cdot f(r) > 0$,
$fa_{k+1} = r$; $fb_{k+1} = b_k$;
else
$fa_{k+1} = a_k$; $fb_{k+1} = r$;
endif
endif
Since the root is bracketed by both $[ba_{k+1}, bb_{k+1}]$ and $[fa_{k+1}, fb_{k+1}]$, define
$[a_{k+1}, b_{k+1}] = [ba_{k+1}, bb_{k+1}] \cap [fa_{k+1}, fb_{k+1}]$
$a_{k+1} = \max(ba_{k+1}, fa_{k+1})$;
$b_{k+1} = \min(bb_{k+1}, fb_{k+1})$;
outcome: iteration complexity, root, and error of approximation
iterationCount = k
$r = r_k$
error = $\in_a = |f(r)|+|r_k-r_{k-1}|$
k = k + 1
until$\in_a < \in$ or k >maxIterations

4 Simplicity and Complexity of Hybrid Algorithm

The new hybrid algorithm is a blend of the Bisection algorithm, False Position algorithm, independent of any other algorithm. The new algorithm differs from all the previous algorithms by tracking the best root approximationin addition to the bestbracketing interval. The number of iterations to find a root depends on the criteria used to determine the root accuracy. The complexity of the new algorithm is better that the bisection. In other words, it uses fewer iterations than the bisection algorithm whereas other algorithms use more iterations than Bisection method. If f(x) is used, then complexity depends on the function as well as the method.

If relative error is used, it does not work for every function because it gets stuck for the case where relative error is constant. Most of the time, absolute error, $\in_s$, is used as the stopping criteria. For functions on interval [a, b] with the Bisection method, the upper bound $n_b(\in)$ on the number of iterations can be predetermined from $\frac{b-a}{2^n} < \in_s$ and is lg $((b - a)/\in_s)$. For the False Position method, it depends on the convexity and location of the root near the endpoint of the bracketing interval. The bound $n_f(\in_s)$ for the number of iterations for the False Position method cannot be predetermined, it can be less, $n_f(\in_s) < n_b(\in) = \lg((b - a)/\in_s)$ or can be greater, $n_f(\in_s) > n_b(\in_s) = \lg((b - a)/\in_s)$. The number of iterations,N= $n(\in_s)$, in the new algorithm is less than $\min(n_f(\in_s), n_b(\in_s))$. The new algorithm complexity of order less that N. This is confirmed from the algorithm and is validated with the empirical computations, see Table4, Table 5,Table6.

5 Empirical Results of Simulations

The new hybrid algorithm is a tested technique that outperforms other existing methods by optimizing the *number of iterations* required for approximations and the computation of *function evaluations* at each step.

Table 1 is a listing of functions from literature that are used to test the validity of new algorithm.

For test bench, we have collected a set of eighteen functions actually used in the papers to benchmark our results against their findings published in the papers. There are various types of functions: polynomial, trigonometric, exponential, rational, logarithmic and a heterogeneous combination of these.

Table 1. Collection of functions used in the papers in references

x^3-x^2-x-1	x^2 - x – 2	$8 - x^9$	x^2 - 4	$x^3 + 4x^2$ - I0,	$(x + 3)(x - 1)^2$
x^3-10	$(x - 1)^3$ - 1,	$x^{10} - 1$,	$(x - 2)^{23} - 1$,	x^3-x+3	$4x^3$-$16x^2$+17x-4
x-cos(x)	$\sin^2 x - x^2 + 1$,	tanx-2x;	$x^3 - e^{-x}$	$x^3 - e^{-x} - 3x + 2$,	(1/(x-3))-6
x+log(x);		$x \sin(1/x) - 0.2\, e^{-x}$			

One of the papers[19] used the following nine functions in place of 20 with the roots listed below. We tested our algorithm on all these functions and are comparing with the cited published result.

Table 2. Nine Functions with roots[Singh1027]

Function	Root
$x^3 + 4x^2 - 10$	1.365230013
$\sin^2 x - x^2 + 1$	-1.404491648
$x^3 - 10$	2.15443469
$x^3 - e^{-x}$	0.772882959
$x \sin(1/x) - 0.2e^{-x}$	0.363715709
$(x-1)^3 - 1,$	2
$x^{10} - 1$	1
$x^3 - e^{-x} - 3x + 2$	0.257530285
$(x-2)^{23} - 1$	3

Table 3 One of the papers [18] used these four functions

Function
cos(x)-x
$x^3 + 4x^2 - 10$
$(x-1)^3 - 1,$
sin(x)-x/2

First we compare algorithm with functions that may require only continuity. The algorithm works on continuity based algorithms as well as derivative based algorithms.

Table 4 Comparison of 18 functions for the number of iterations required by each method.

	Number of Iterations				Error Tolerance = 0.00000000001				
	[0.1,1.5]			[-2, 1]			[3.1, 4]		
Intervals	[1, 4]			[0.2, 2]			[0.5, 1]		
		[1, 2]			[0.2, 4]			[0.3, 2]	
Functions →	$8 - x^9$			x^3-x+3			1/(x-3)-6		
	$x^2 - x - 2$			x^3-x^2-x-1			x-cos(x)		
Algorithms ↓		$x^2 - 4$			$4x^3-16x^2+17x-4$			x+log(x)	
Bisection	21	40	19	40	40	40	40	39	40
FalsePosition	30	32	1	20	15	40	34	10	17
Dekker	37	9	2	10	8	10	11	8	7
Brent	17	14	1	11	17	17	14	12	9
Hybrid	8	2	1	9	8	10	9	2	7
Newton	1	6	4	5	18	5	6	5	5
RQI	1	4	3	3	8	3	4	3	3
Halley	2	5	4	4	10	4	2	4	4
Secant	17	9	2	40	9	8	13	6	8
SecantModified	40	7	5	5	24	5	6	6	5

Table5 is a comparison of the algorithms using continuity and derivative of functions.

Table5 Comparisons of iterations for ten algorithms on three functions

Number of Iterations Error Tolerance = 0.00000000001			
Intervals	[1, 4]	[1, 2]	[0.2, 2]
Functions →	$x^2 - x - 2$		
		$x^2 - 4$	
Algorithms ↓			x^3-x^2-x-1
Bisection	40	19	40
FalsePosition	32	1	15
Dekker	9	2	8
Brent	14	1	17
Hybrid	2	1	8
Newton	6	4	18
RQI	4	3	8
Halley	5	4	10
Secant	9	2	9
SecantModified	7	5	24

From the methods defined in section 2, Fang's method is denoted by MF, Sharma's method by MSh, Grau's method by MG) and the Nora Fitriyani, M. Imran, Syamsudhuha's method MKA. They test their algorithms with four functions.If we denote our algorithm as CLS[a,b], where [a,b] is the interval of definition, it is clear in table 6 that our algorithm without using derivatives competes with their findings. It finishes before any of the cited algorithms.

Table6 Comparison of the derivative based algorithms with our algorithm on four functions.

f(x)	x_0	Method	Iterations
$\cos(x) - x$	1.7	MKA	4
		MF	5
		MSh	4
		MG	4
		CLS [.5, 1]	2
$x^3 + 4x^2 - 10$	1.6	MKA	4
		MF	4
		MSh	4
		MG	4
		CLS [1, 4]	3
$(x - 1)^3 - 1$	3.5	MKA	5
		MF	6
		MSh	5
		MG	5
		CLS [0, 4]	1
$\sin(x) - x/2$	2	MKA	4
		MF	4
		MSh	4
		MG	4
		CLS [1, 2]	3

6 Conclusion

We have designed and implemented a new algorithm, a dynamic hybrid of Bisection and Regula Falsi methods. The algorithm was implemented in Matlab R2018B 64 bit (maci64) on MacBook Pro MacOS Mojave2.2GHz intel Core i716 GB2400MHz DDR4 Radeon Pro555X 4GB. The implementation experiments indicate that it outperforms both these algorithms all the time by a considerable margin. It is also observed that the new algorithm outperforms Secant method and Newton-Raphson method and its variants. It is benchmarked using Bisection, False Position, Dekker, Brent algorithms on various functions cited in the literature.The experiments on numerous datasets used in the literature justify that the new algorithm is effective both conceptually and computationally. Thus, we contribute a superior new algorithm to the repertoire of classical algorithms.

We find that some researchers have also used genetic algorithms. New reliable efficient algorithms are emerging, this was an attempt for interval based continuous functions, there is scope to improve point passed algorithms for minimally differentiable functions.In this paper, we observed algorithm efficiency was principal concern thatthe new algorithm uses fewer iterations than their findings on genetic algorithms. In the future, forbetter treatment of genetic algorithms as they pertain to root finding, we will develop our own geneticalgorithms, as the blended algorithm over the bisection and false position methods presented in thispaper.

7 References

[1]. [Dekker1968]T. J. Dekker and W. Hoffmann (part 2), *Algol 60 Procedures in Numerical Algebra, Parts 1 and 2*, Tracts 22 and 23, Mathematisch Centrum Amsterdam, 1968.

[2]. [Dekker1969]Dekker, T. J. (1969), "Finding a zero by means of successive linear interpolation", in Dejon, B.; Henrici, P. (eds.), *Constructive Aspects of the Fundamental Theorem of Algebra*, London: Wiley-Interscience, ISBN978-0-471-20300-1 https://en.wikipedia.org/wiki/Brent%27s_method#Dekker's_method

[3]. [Brent1973]Brent's Method https://en.wikipedia.org/wiki/Brent%27s_method#Dekker's_method (accessed 12 December 12,2019

[4]. [Press2007]Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. (2007). *"Van Wijngaarden–Dekker–Brent Method"*. Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press. *ISBN978-0-521-88068-8*.

[5]. [Sivanandam2008]Sivanandam, S.; Deepa, S. Genetic algorithm implementation using matlab. In *Introduction to Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 211–262, doi: 10.1007/978-3-540-73190-0_8.

[6]. [Moazzam2012]Moazzam, G.; Chakraborty, A.; Bhuiyan, A. A robust method for solving transcendental equations. *Int. J. Comput. Sci. Issues* 2012, *9*, 413–419.

[7]. [Ogbereyivwe 2014] OgbereyivweOghovese, Emunefe O. John , *IOSR Journal of Mathematics (IOSR-JM) e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 10, Issue 5 Ver. IV (Sep-Oct. 2014), PP 44-47 www.iosrjournals.org*

[8]. [Fitriyani2018] Nora Fitriyani, M. Imran, Syamsudhuha , A Three-Step Iterative Method to Solve A Nonlinear Equation via an Undetermined Coefficient Method , Global Journal of Pure and Applied Mathematics. , ISSN 0973-1768 Volume 14, Number 11 (2018), pp. 1425-1435, http://www.ripublication.com/gjpam.htm

[9]. [Darvishi2007] M.T. Darvishi and A. Barati, A third-oredr Newton-type method to solve system of nonlinear equations, Appl. Math. Comput, 187 (2007), 630{635. Math. Comput, 169 (2004), 161{169.

[10]. [Weerakoon2000] S. Weerakoon and T.G.I. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett 13(8)(2000) 87:93.

[11]. [Khattri2011] S. K. Khattri dan S. Abbasbandy, *Optimal Fourth Order Family of Iterative Methods*, Mat. Vesn., 63(2011), 67 - 72.

[12]. [Ostrowski1966] Ostrowski, A. M., (1966). Solutions of Equations and System of Equations, Academic Press, New York-London, (1966).

[13]. [Cordero2007] Cordero, A. and Torregrosa, J. R., (2007). Variants of Newton's Method using fifth-order quadrature formulas. Appl. Math. Comput., 190:686-698.

[14]. [Jayakumar2013] J. Jayakumar , Generalized Simpson-Newton's Method for Solving Nonlinear Equations with Cubic Convergence, *IOSR Journal of Mathematics (IOSR-JM), e-ISSN: 2278-5728,p-ISSN: 2319-765X, Volume 7, Issue 5 (Jul. - Aug. 2013), PP 58-61, www.iosrjournals.org*

[15]. [Chun2005] C. Chun, Iterative method improving Newton's method by the decomposition method, Comput. Math. Appl, 50 (2005), 1559{1568.

[16]. [Grau2006]M.Grau dan J.L. Diaz-Barrero, *An Improvement to Ostrowski Root-Finding Method*, Appl. Math. Comput., 173(2006), 450 – 456.

[17]. [Sharma 2007] J. R. Sharma dan R. K. Guha, *A Family of Modified Ostrowski Methods with Accelerated Sixth Order Convergence*, Appl. Math. Comput., 190(2007), 111 - 115.

[18]. Fang2011]Fang et al. [L. Fang, T. Chen, L. Tian, L. Sun, dan B. Chen, *A Modified Newton-type Method with Sixth-order Convergence for Solving Nonlinear equations*, Procedia Engineering., 15(2011), 3124 - 3128.

[19]. [Singh2017] Manoj Kumar Singh and Arvind K. Singh, A Six-Order Method for Non-linear Equations to Find Roots, International Journal of Advance Engineering and Research Development Volume 4, Issue 9, September -2017 , e-ISSN (O): 2348-4470 p-ISSN (P): 2348-6406.