International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume –05, Issue – 12, December 2020, PP – 05-15

A Geometric Algorithm for Roots of Algebraic and Transcendental Non-Linear Equations

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

Abstract: There are numerous methods for solving non-linear equations whicharise in all areas predominantly engineering field. In general,all these methods are enhancement of three basic methods: Bisection, Regula Falsi, and Newton-Raphson method. Severalvariations have been made forimproving the performance of these methods. Since the non-linear equations do not have analytic solutions, the researchers resort to numerical techniques. We design and implement a threewayhybrid algorithm,that is geometric approach based on only first derivative independent of any existing variation.We compare the performance of our algorithm with the available methods on various equations found in the recent literature as benchmark. Experiments validate thatthe new algorithm converges more quicklyor at least at par with previous algorithms, confirming that the new algorithm is simpler, efficient and accurate.

Keywords: Bisection, False Position, Iteration, Newton-Raphson, Predictor-Corrector, Quadrature

1. Introduction

There are numerous formulas for solving non-linear equations which arise in all areas predominantly the engineering field. Since there are three basic methods: Bisection, Regula Falsi, and Newton-Raphson method,several variations have been made for improving the performance of these methods. Mostly, these variations are incremental enhancements. The non-linear equations do not have an analytic solution, the researchers resort to numerical techniques. We design and implement a three way hybrid algorithm, that is a geometric approach based on only first derivative independent of any existing variation. However, we compare the performance of our algorithm with the available variations on methodsfound in the literature as benchmark. We show that the new algorithm is better or at par with any other existing variants like 2^{nd}-6^{th} order methods.Experiments validate that the new algorithm converges more quickly or at least at par with previous algorithms, Theconstruction of numerical solution for non-linear equationsisessentialin many branches of science and engineering. Most of the time,there is no analytic solution for non-linear equations.In that case,the researchershave no choice except to depend on approximate solutions.

The numerical methods fall into two categories based on: (1) only continuous functions and (2) differentiablefunctions. Continuous function based methods include Bisection and False Position methods and their variants, e.g., Dekker [1], Brent[2],methods, whereas the variants of derivative basedNewton-Raphson methodbelong to categories of 2^{nd}, 3^{rd}, 4^{th} , 5^{th} , and 6^{th} order formulas, method of undetermined coefficients[4],[5][6],[7],[8],[9].For the purpose of improving the performance of the classical methods, Bisection, False Position, Newton Raphson methods, more and more variant methods are evolving.The new algorithm builds on Bisection, False-Position, and Newton-Raphson algorithms collectively and performs better than these and the variant methods.

The paper is divided into sections: Section 2 is brief background of these methods, Section 3 is design of three way hybrid algorithm, Section 4 is theoretical complexity comparison, Section 5 is experimental analysis, Section 6 is conclusion, Section 7 is references.

2. Background

2.1 Definitions

The nonlinear equations encounter two types functions: continuous functions (not necessarily differentiable) and differentiable functions (up to varying orders of differentiability). There are two types of solutions: (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.

Fora function $f:[a, b] \rightarrow R$ 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)·f(b) < 0, then there exists a root r ∈ [a, b] such that f(r) = 0 or

(2') the function f(x) is differentiable with g(x)= x-f(x)/f'(x) and |g'(x)<1, then there exists a root r ∈ [a, b] such that f(r) = 0.

For simulations in search for approximate solution, we use error tolerance $\varepsilon = 10^{-10}$, with termination criteria as $(|x_k - x_{k-1}|+|f(x_k)|) < \varepsilon$, upper bound on the number of iterations as 10^2. Since the solution is obtained by iterative methods, the definition of *order of convergence* is as follows [5] .

Let $x_n, a \in$ R n≥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 >N

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

then then 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.

2.2 Relevant Literature

The continuous function methods are based onBisection (1817) [10] and False Position (1690)[11] methods and their variants, e.g., [1], [2],[3] methods. The derivative based Newton-Raphson(1736) [12] methods belong to category of 2^{nd}-6^{th} order formulas, method of undetermined coefficients. The secant method is overlap between False Position method that ensures bracketing and Newton-Raphson[12] method does not consider bracketing. For continuous functions, there is a smoothincremental transition from Bisection to Dekker to Brent with RQI(reverse quadratic interpolation) to Chandrupatla using exclusively one or the other alternative, not taking advantage of the commonality.

For differentiable functions, there are variousmethods starting from Newton-Raphson method to its various variants. The Newton-Raphson method uses Tylor series expansion of the function.

By Taylor series expansion of f(x), we have

$$f(b) = f(a)+(b-a)f'(a)+ (b-a)^2 f''(a)/2!+\dots$$

For second order formula, we drop the terms from second order derivative terms, f"(a), onward) to get an approximation

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

Further approximatingthe value of b as the root of f(x) = 0, 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 the function f(x), Newton-Raphson successive estimates closer to the solution are

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

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

This amounts two function evaluations for each iteration. To ensure second order convergence, if write

$$g(x)= x - \frac{f(x)}{f'(x)} \qquad (2)$$

then $$x_{n+1} = g(x_n) \qquad (3)$$

amounts to one function evaluation for each iteration. It is just a matter of how we count the number of function evaluations. If | g'(x)|< 1, then the sequence $\{x_n\}$ must converge to a root.

If the multiplicity,m, of the root is known,[13] , then g(x) is optimized to

$$g(x)= x - \frac{m\, f(x)}{f'(x)} \qquad (4)$$

$$\text{or } g(x)= x - \frac{f(x)}{\frac{f'(x)}{m}} \qquad (5)$$

this speeds up the root approximation.

Newton-Raphson uses this function g(x) to go from x_n to x_{n+1}. Modifications are incorporated to insert steps between them to get better reliable estimate of x_{n+1}. Severalvariations have been popularized recently. To ascertain the improvement in the variants, the metrics for evaluation include: Coefficient of Efficiency, Coefficient of Computation, domain of definition, start point, number of iterations, number of function evaluations forthe approximate root. Intermediate steps between x_n and x_{n+1} revolve around the quadrature methods and the order of convergence.Some of the methods are:

Newton-Raphson method is used twice to go fromx_n to predictor y_n, then use y_n to correct x_{n+1}[14]. It is still second order method.

Instead of $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ (6)

Use notation $y_n$for predictor

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} \quad (7)$$

Then use y_n as corrector for x_{n+1}

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} \quad (8)$$

Recently midpoint[15] quadrature method is used to predict x_{n+1}, in order to go from x_n to predictor x_{n+1} using midpoint quadrature method, then$y_n$is used again as intermediate corrector to estimate x_{n+1}. In this case the corrected estimate for x_{n+1}

$$x_{n+1} = x_n - \frac{f(x_n)}{f'\left(\frac{x_n+y_n}{2}\right)} \quad (9)$$

Linear interpolation [16]: Trapezoidal quadrature method is used to predict x_{n+1}, in order to go from x_n to predictor x_{n+1} using trapezoidal quadrature method, the y_n is used again as intermediate predictor to estimate x_{n+1} In this case predictor x_{n+1}

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{f'(x_n)+f'(y_n)}{2}} \quad (10)$$

or

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

This also called Newton – arithmetic mean formula, linear interpolation of derivative. There are various version of this formula by replacing arithmetic mean by Geometric Mean, and Harmonic Mean[17].

Geometric mean[17]:

$$x_{n+1} = x_n - \frac{f(x_n)}{\sqrt[2]{f'(x_n)*f'(y_n)}} \quad (11)$$

Harmonic mean [16]:

$$x_{n+1} = x_n - \frac{f(x_n)}{2/\left\{\frac{1}{f'(x_n)}+\frac{1}{f'(y_n)}\right\}} \quad (12)$$

$$\text{or } x_{n+1} = x_n - \frac{f(x_n)}{2}\left\{\frac{1}{f'(x_n)} + \frac{1}{f'(y_n)}\right\} \quad (13)$$

Trapezoidal-Newton's method [17] : one obtains a new approximation

$$x_{n+1} = x_n - \frac{4f(x_n)}{f'(x_n)+2f'\left(\frac{x_n+y_n}{2}\right)+f'(y_n)}, \quad (14)$$

$$x_{n+1} = x_n - \frac{4f(x_n)}{f'(x_n)+2f'\left(\frac{x_n+y_n}{2}\right)+f'(y_n)}$$

The Simpson 3/8[15] quadrature method yields

$$x_{n+1} = x_n - \frac{6f(x_n)}{f'(x_n)+4f'\left(\frac{x_n+y_n}{2}\right)+f'(y_n)} \quad (15)$$

Most of the time, they are of the form

$$x_{n+1} = x_n - \frac{f(x_n)}{\{Den\,f'(x_n)\}} \quad (16)$$

where $\{Denf'(x_n)\}$ represents variation of f'(x) in the denominator by different quadrature formulas. But this expression is not used all the time [14] . In fact, these single method formulas can further be expressed as the weighted average of different methods to achieve better order to convergence. For example, if M,T, S are Midpoint, Trapezoidal, Simpson methods are used for approximation of f'(x), then mM+tT+sS with m+t+s =1, $0 \leq m, t, s \leq 1$ is a hybrid approximation

$$x_{n+1} = x_n - \frac{f(x_n)}{\{mM+tT+sS\}} \quad (17)$$

[15] method uses the average of Midpoint and Simpson quadrature. JayaKumar [9] proposed a generalization of the Simpson-Newton's method where theyuse multisteptrapezoidal rule.Another interesting approach is to usethe method of undetermined coefficients [18] and in related citations is different in interesting.

For validating the superiority of their findings[18], some may have usedspecial attributes such as start point, start interval, location of the root.All methods work on special cases, but not in general all the time. Probably data used in most of these is hit and trial,with slightingenuity and innovation in combining them, and calculate the related error as justification.

2.3 The Need for New Algorithm

The new algorithm builds on Bisection, False-Position, and Newton-Raphson algorithms collectively.The Bisection method is virtually a binary search. It does not matter what the function is, the approximation error upper bound can be determined a priori. The iterations can be viewed geometrically

For f:[A,B]→ R, the nextapproximation at the midpoint (A+B)/2 = M [see Fig. 1], for next approximation, the interval [A,B] is replaced by the smaller interval [M,B]. For each iteration interval $[A_n,B_n]$, midpoint M_n or approximation $x_n$becomes the end point of next interval $[A_{n+1},B_{n+1}]$, A_{n+1} or B_{n+1} is $x_n$and now the root is bracketed by interval $[A_{n+1}, B_{n+1}]$. The length of interval $[A_{n+1}, B_{n+1}]$ is half the length of $[A_n, B_n]$. Thus this sequence is guaranteed to converge.

If the approximation error tolerance is $O(\in = \frac{1}{2^n})$, then complexity of the number of iterations is O(n) iterations, see section 4.

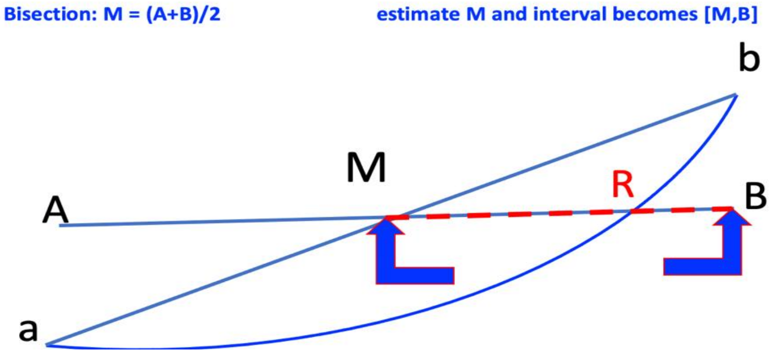

Figure1. Continuous function on [A,B], the iteration estimate is midpoint M, the next interval[M,B] contains the root R.

The bisection method does not take advantage of the fact that B is closer the root R. The False Position is a dynamic method and takes advantage of the location of the root to make a conceivably more accurate outcome.

See Fig. 2 for f:[A,B], the next approximation is S, the intersectionof secant line with the line AB and $S= A - f(A)\frac{(B-A)}{f(B)-f(A)}$ instead of the midpoint M. Clearly Sis closer to the root thanM is to the root [Fig. 2]. Now the root is bracketed in smaller interval [S,B]. Since the intervals are shrinking brackets, it is guaranteed to converge to the root. It is supposed to be faster than Bisection, but unfortunately it is not so all the time. Shrinking intervals may not shrinking faster as expected in some cases [Fig. 3].Thus the number of iterations cannot be predetermined quickly as in the case of Bisection method. In this example, the secant line moves slower than the midpoint towards the root.

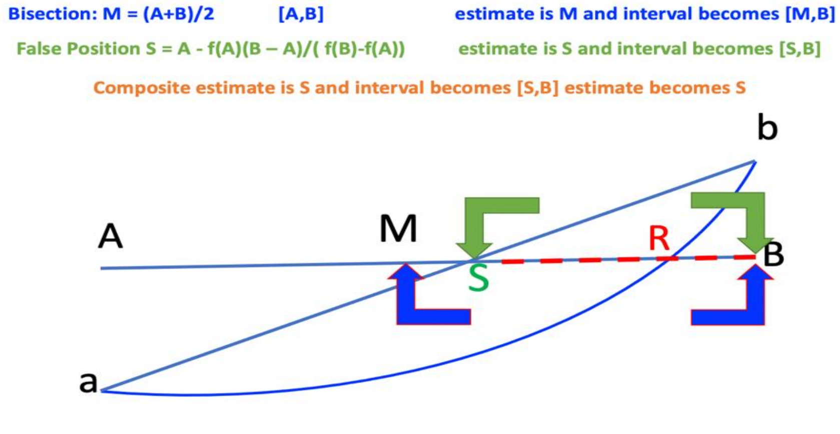

Figure 2. Secant point is closer to the root than the middle point M.

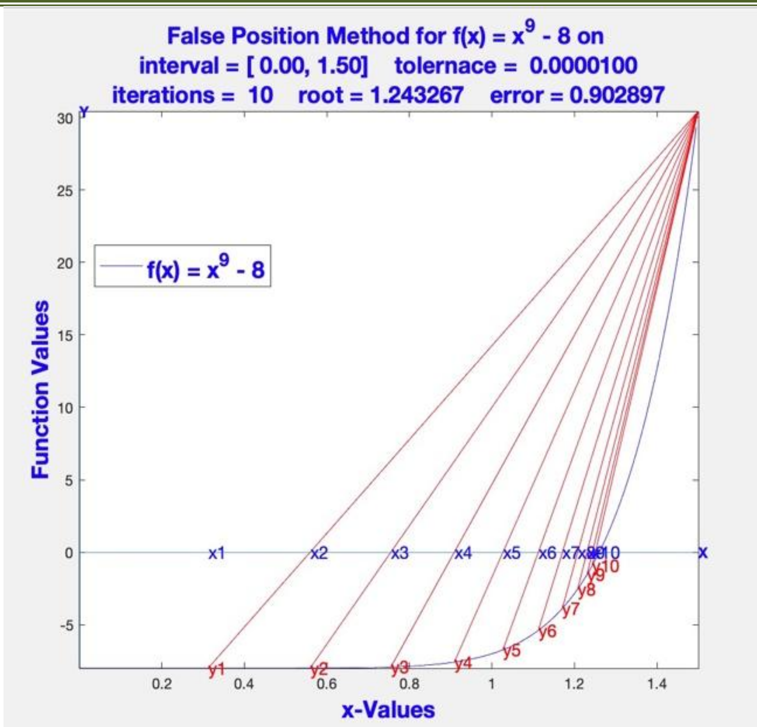

Figure 3. Secant point is father from the root than Mid point is.

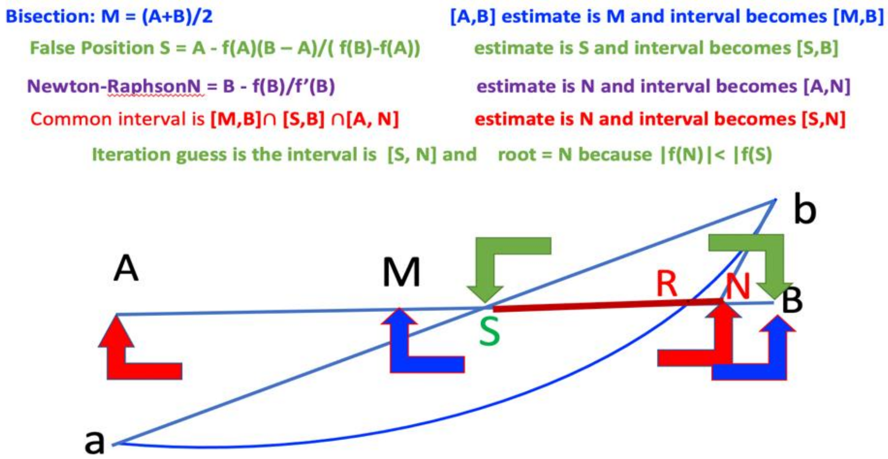

Figure 4. Newton point N is closer to the root than midpoint M, and secant point S.

Newton's method is based on a start point and tangent line. Since here |F(B)| < |F(A)|, the point B is used as the start point [Fig.4]. Here tangent line at b = (B,F(B))intersects AB at N (N for Newton). N is a clear winner, N is closer to the root than M or S.

In the next Section 3, we create a hybrid algorithm that uses all these three methods collectively and has the reliability convergence of Bisection, speed of False position and Newton methods, the computational complexity is better of the three and many more complex order algorithms together.

3. Three Way Hybrid Algorithm

This algorithm determines the roots of algebraic and transcendental non-linear equations quickly. It has the reliability of Bisection method. The algorithms like Newton–Raphson and its variant methods have quadratic, 3^{rd}-6^{th} order convergence, but converge with additional differentiability constraints imposed on the functions and the initial guess is close to the root [15], [18]. The new algorithm differs from all the previous algorithms in tracking the best approximation and bracketing interval. Instead of brute force application of the Bisection or False Position or Newton-Raphson methodsalone, we selectively apply the blend of the three at each step to redefine the next approximate root and bracketing interval. Thus, we construct a new hybridalgorithm that performs better than the Bisection, False Position and Newton-Raphson and many more methods. This prevents the unnecessary iterations performed in either method. This method does not require complex differentiability but utilizes derivative if available as required by Newton–Raphson and its variants. This is the simplest, reliable, fast algorithm.

HybridAlgorithm

Input: function, interval [a,b] containing the root
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 position is bisection method a
$fa_{k+1}=ba_{k+1}=a_1;fb_{k+1}=bb_{k+1}=b_1$
$n_1=a_1 - f(a_1)/df(a_1)$;

repeat
    $fa_{k+1}=ba_{k+1}=a_k;fb_{k+1}=bb_{k+1}=b_k$
    compute theNewton point
    $n_{k+1}= n_k-\frac{f(n_k)}{f'(n_k)}$
    compute the Mid point
    $m=\frac{a_k+b_k}{2}$, and $\in_m = |f(m)|$
    compute the Secant point, False Position secantline point,
    $s_k = a_k- \frac{f(a_k)\,(b_k-a_k)}{f(b_k)-f(a_k)}$ and $\in_f = |f(s)|$
    r= s
    $\in_a = \in_f$
    if $|f(m)| < |f(s_k)|$,
        f(m) 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_k
        $\in_a = \in_f$
        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
        $[a_{k+1}, b_{k+1}] = [ba_{k+1}, bb_{k+1}] \cap [fa_{k+1}, fb_{k+1}]$
        $a_{k+1} = \max(ba_{k+1}, fa_{k+1})$;
        $b_{k+1} = \min(bb_{k+1}, fb_{k+1})$;
    //now if f is differentiable, use Newton step else ignore it,
    //differentiability can be specified in the function call
        //use n_{k+1} if $f(n_{k+1})<\min(f(a_{k+1}),f(b_{k+1}))$
        //replace $a_{k+1.}$ or b_{k+1} by n_{k+1} resulting in further smaller interval, and new root r
        //outcome: iteration complexity, root, and error of approximation

        If $|f(n_{k+1})|< \min(|f(a_{k+1})|,|f(b_{k+1})|)$
            r= n_{k+1}


```
                    if f(r)f(a_{k+1})<0  b_{k+1} = r;
                    else a_{k+1} = r;
                endif
        iterationCount = k
        r_{k+1}= r
        error = ∈_a = |f(r)|+|r_k-r_{k+1}|
        k = k + 1
    until∈_a<∈  or k >maxIterations
```

4. Simplicity and Complexity of Hybrid Algorithm

A twofold algorithm is a blended[19] blend of two algorithms: Bisection algorithm, False Position algorithm) and is independent of any other algorithm. The new hybrid algorithm is an improvement over Newton-Raphson algorithm and the two fold is blended[19]. The new algorithm differs from all the previous algorithms by tracking the best root approximation in addition to the best bracketing interval. The number of iterations to find a root depends on the tolerance criteria used to determine the root accuracy. The complexity of the new algorithm is better than that of the cited algorithms.

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, $n_f(\in_s) > n_b(\in_s) = \lg((b-a)/\in_s)$. For example, if $\in_s = 2^{-n}$, the number of iterations, $n(\in_s) = O(n)$, in the new algorithm is less than $\min(n_f(\in_s), n_b(\in_s))$. The introduction of Newton-Raphson in Hybrid further reduces the complexity of computations resulting fewer iterations, the new algorithm complexity is of order far less thann. This is confirmed from the algorithm and is validated with empirical computations, see Table4, Table 5,Table6.

5. Empirical Results of Simulations

The new hybrid algorithm is a tested technique 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 cited 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. There are six tables here. Table 1. Collection of functions used in the papers in references, Table 2 has Nine Functions used in [20]. Table 3 has four functions for comparison of the hybrid algorithm with four references used in Table 6. We used the same functions for validating the hybrid algorithm by comparing them side by side. Table 4 is a comparison of the algorithms using both continuity and derivative of functions. Table 5 is Comparison of 18 functions for the number of iterations required by each method.

Table 1. Collection of functions used in the papers in references [18].

x^3-x^2-x-1	$x^2 - x - 2$	$8 - x^9$	$x^2 - 4$	$x^3 + 4x^2 - 10$	$(x + 3)(x - 1)^2$
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 authors[20]used the following nine functions in place of 20 listed below. We tested our algorithm on all these functions and will compare with the cited published results side by side.

Table 2. Nine Functions with roots[20]

Function
$x^3 + 4x^2 - 10$
$\sin^2 x - x^2 + 1$
$x^3 - 10$
$x^3 - e^{-x}$
$x \sin(1/x) - 0.2 e^{-x}$
$(x - 1)^3 - 1$,
$x^{10} - 1$
$x^3 - e^{-x} - 3x + 2$
$(x - 2)^{23} - 1$

One of the papers [18] used these four functions for comparison with four references. We also used the same for validating the hybrid algorithm by comparing them side by side in the Table 6.

Table 3 Four functions used for root finding iterations compared in Table 6

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 in Table 6.Table 4 is a comparison of the algorithms using continuity and derivative of functions.

Table 4 Comparisons of iterations for seven algorithms on three functions

Number of iterations	Tolerance 0.0000001		
Interval	[1, 7]	[1, 3]	[0, 2]
Function	$x^2 - x - 2$	$x^2 - 4$	$x^3 - x^2 - x - 1$
Algorithm			
Bisection	23	1	19
FalsePosition	33	10	5
Newton-Rap	5	4	12
Blended	6	1	5
Hybrid	5	1	4
Secant	7	5	11
ModifiedSec	6	5	50

Table 5 Comparison of 18 functions for the number of iterations required by each method

Number of Iterations Error Tolerance = 0.00000000001

Functions → / Algorithms ↓	$8 - x^9$	$x^2 - x - 2$	$x^2 - 4$	x^3-x+3	x^3-x^2-x-1	$4x^3-16x^2+17x-4$	1/(x-3)-6	x-cos(x)	x+log(x)
Intervals	[0.1,1.5]	[1, 4]	[1, 2]	[-2, 1]	[0.2, 2]	[0.2, 4]	[3.1, 4]	[0.7, 1.8]	[0.3, 2]
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
Blended	8	2	1	9	8	10	9	3	7
Hybrid	7	2	1	4	6	3	3	3	3
Newton	8	6	4	5	18	5	6	5	5
RQI	7	4	3	3	8	3	6	3	3
Halley	8	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

One of the papers [15] used these four functions for comparison with four references. Four functions are used for root finding iterations complexity compared with the published results. The hybrid algorithm computes the root with fewer iterations than those identified by acronyms: Fang's [21]method is denoted by MF, Sharma's [22]method by MSh, Grau's [23] method by MG) and the Nora Fitriyani, M. Imran, Syamsudhuha's[18] method MKA. In the table6, there are five methods, MKA compared their findings with the MF, MShe, and MG methods. We added the hybrid findings labelled as CLS and compare the number of iterations side by side to show the superiority of hybrid algorithm They test their algorithms with four functions. The hybrid algorithm is labelled CLS[a,b], where [a,b] is the interval of definition, it is clear in Table 6 that thenew hybrid competes with their findings. It finishes before any of the citedhigher order algorithms.

Table 6 Comparison of new hybridand derivative based algorithms on four selected functions.

f(x)	x_0	Method	Iterations
$\cos(x) - x$	1.7	MKA	4
		MF	5
		MSh	4
		MG	4
		CLS [.7, 1.8]	3
$x^3 + 4x^2 - 10$	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, Regula Falsi and Newton-Raphson 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 number of function evaluations in each iteration is constant and predetermined as compared to variable number of function evaluations for different methods. For number of iterations, the implementation experiments confirm that the new algorithmperforms better or at least as good as any other algorithm. It is benchmarked using variants of Bisection, False Position, Newton-Raphson algorithms on various functions cited in the literature.In this paper, the algorithm simplicity and efficiency were principal concerns that the new algorithm use fewer iterations. The outcomes of numerous datasets justify that the new algorithm is effective cognitively, conceptually and computationally.

7. References

[1]. 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

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

[3]. Chandrupatla, Tirupathi R. (1997). "A new hybrid quadratic/Bisection algorithm for finding the zero of a nonlinear function without using derivatives". *Advances in Engineering Software*. **28** (3): 145–149. doi:10.1016/S0965-9978(96)00051-8

[4]. 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

[5]. 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.

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

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

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

[9]. 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*

[10]. Burden, Richard L.; Faires, J. Douglas (1985), *Numerical Analysis* (3rd ed.), PWS Publishers, ISBN 0-87150-857-5

[11]. Joseph Needham (1 January 1959). *Science and Civilisation in China: Volume 3, Mathematics and the Sciences of the Heavens and the Earth*. Cambridge University Press. pp. 147–. ISBN 978-0-521-05801-8.

[12]. Kendall E. Atkinson, *An Introduction to Numerical Analysis*, (1989) John Wiley & Sons, Inc, ISBN 0-471-62489-6

[13]. Steven C. Chapra, Raymond Canale,2015, Numerical Methods for Engineers, McGraw-Hill Education

[14]. VBKummarVatti, Shouri Dominic, Sahanica V, **Cubic** Convergent Modified Newton's Method, International journal of advanced research(IJAR)pp.48-52, www.journalijar.com, DOI 10.214774/IJAR01/1457

[15]. 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*

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

[17]. [babaje2006] D.K.R. Babajee, M.Z. Dauhoo, An analysis of the properties of the variants of Newton's method with third order convergence, Applied Mathematics and Computation 183 (2006) 659–684.

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

[19]. Chaman Lal Sabharwal, Blended Root Finding Algorithm Outperforms Bisection and Regula Falsi Algorithms, *Mathematics* 2019, *7*, 1118; doi:10.3390/math7111118, www.mdpi.com/journal/mathematics

[20]. 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.

[21]. 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.

[22]. 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.

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

