Math. & Comput. Sci., 6(3) 2025, 77–88

Online: ISSN 2717-2708

Mathematics and Computational Sciences

Journal Homepage: http://mcs.qut.ac.ir/mcs

Non-divergent circular arc root algorithm

Mehmet Pakdemirli[a,*]

[a]Department of Mechanical Engineering, Manisa Celal Bayar University, Yunusemre, Manisa, Turkey

Abstract

A new iteration algorithm for numerically locating the roots of nonlinear algebraic functions is proposed. The algorithm is non-divergent even in the vicinity of local extrema. The algorithm depends of drawing a tangent circular arc to the initial estimation point on the curve. The radius of curvature of the circular arc is equal to the functions radius of curvature at the estimated point. The intersection points of the circular arc with the x axis determine the first iteration for the roots. The iteration equation is derived first. The conditions for which the algorithm works are discussed. It is proven that the convergence rate of the new algorithm is quadratic. Using sample problems, the algorithm is contrasted with the well-known Newton-Raphson algorithm and the parabolic algorithm. It is shown that the algorithm requires less iterations, has a wider convergence interval in general and does not diverge in the vicinity of local extrema as compared to the Newton-Raphson method. For the example considered, the algorithm is better than the parabolic algorithm in terms of the iterations and computational times.

Keywords: Root finding algorithms, Circular arc root algorithm, Newton-Raphson method, Non-divergent algorithm, Nonlinear equations, Convergence rate.

2020 MSC: 65H05, 65D15

1. INTRODUCTION

Determining roots of nonlinear algebraic equations are encountered in many branches of science. They may appear in various physical problems such as wave motion [3], natural frequencies of mechanical vibrations [8], combustion problems [4], molecular attraction forces in chemistry [19, 10], blood flow [10], open channel fluid problems [19, 10], radiation law [19, 10] to mention a few. Polynomiography [10, 20, 9] is a branch of mathematical arts where color graphics are produced for convergence of the polynomial roots.

Newton-Raphson method [7] is one of the most common single-point iteration methods to locate roots of algebraic equations widely used for its simplicity and fast convergence rate. The convergence to the root is maintained by tangent lines and their intersections with the x axis. If the initial guess for the root is not sufficiently close to the root, the algorithm may diverge or converge to another root. For determining systematically the initial guesses for polynomial roots, theorems were developed based on perturbed [12, 13] and non-perturbed approaches [14].

*Corresponding author
Email address: pakdemirli@gmail.com (Mehmet Pakdemirli)

doi: 10.30511/mcs.2025.2028349.1171

Received: 06 May 2024 Accepted: 19 September 2025

A new class of single iteration formulas was proposed by incorporating the perturbation theory and the Taylor series expansions [15]. The algorithms were classified as perturbation-iteration algorithms, i.e. PIA(n,m), where n represents the number of correction terms in the perturbed expansions and m represents the highest order derivatives in the Taylor series expansions. The new class includes many well-known techniques as well as new algorithms as special cases. PIA(1,1) corresponds to the well-known Newton-Raphson method, PIA(1,2) corresponds to the irrational Halley formula, PIA(2,2) corresponds to Householder's or Euler's iteration. Algorithms involving third order derivatives derived by the Adomian Decomposition Method [1] were compared with the PIA(3,3) algorithm given in [15]. New PIA algorithms including fourth order [16] and fifth order [17] derivatives were also proposed. Single-point root finding algorithms were also derived using the Homotopy Perturbation Method [2]. PIA(3,3) which is first derived by the perturbation iteration method [15] was also derived alternatively by the two-parameter Homotopy Method [29].

Convergence rates and the interval of convergence are the two main issues important for root finding techniques. Many higher order convergence rate algorithms than the Newton-Raphson method were proposed [5, 6, 11, 27, 28, 25, 26]. Although these algorithms require lower number of iterations and possess wider range of convergence, the algebra is more involved compared to the Newton-Raphson method. Instead of approaching to the root via the tangent lines, many different tangent functions were employed in locating the roots [18]. The criteria for selecting the suitable functions were also developed in the mentioned study.

Another important issue is to find all roots of a polynomial type equation without searching for the roots one by one [21, 22]. This process is called simultaneous finding of all roots. Convergence rates increase for simultaneous methods [21]. Single root algorithms can be modified to a form suitable for determining all roots [22]. It is shown that the new algorithms presented were more efficient in contrast to the previous methods [21, 22]. Another modification to existing methods is to use fractional derivatives in the iteration algorithms. An efficient single step modified one parameter family of the Caputo-type fractional iterative method has been proposed [23] and applied for solving practical engineering problems. For simultaneous determination of all the roots, a stable Caputo-type inverse numerical fractional scheme has been proposed which performs better than the existing methods [24].

In this work, a new single-point iteration algorithm is proposed. The new algorithm is named as Circular Arc Root Algorithm (CARA). Instead of the tangent lines of the Newton-Raphson method, tangent circular arcs are employed in approaching the roots of the algebraic equations. The circular arcs have a radius of curvature equivalent to that of the function at the tangent point. The iteration formula is derived using the principles of analytical geometry and calculus. One of the deficiencies of the Newton-Raphson method (NR) and many of the more advanced algorithms [15] is that the algorithms diverge in the vicinity of local extrema or at points where the first derivative of the function is extremely small. This new proposed algorithm can locate the root starting from even the exact location of the local extrema. With this property, depending on the specific equation considered, it usually possesses a wider convergence interval. The conditions for which real solutions can be obtained are discussed. The convergence rate is found to be quadratic. Convergence rates can be improved for this algorithm also by implementation of several equations instead of one in a single iteration step as is usually done for other iteration schemes. Numerical simulations revealed that less iterations are required to locate the roots and a much wider convergence interval is achieved in general by CARA compared to the NR method. Furthermore, starting from the same initial guess, it is possible to approach two roots by CARA in contrast to the single root in the case of NR. The well-known geometrical iteration formula, namely the Parabolic iteration algorithm in which the convergence to the root is achieved via tangent parabolas is also employed to compare the results with the new algorithm. For the example considered, the new algorithm performs better than the parabolic algorithm.

In summary the motivation behind the work is:

- To develop a root finding algorithm based purely on geometrical concepts.

- To have an algorithm that does not diverge even in the vicinity of local extrema.
- To reduce the iterations to reach to the root with a given precision.
- To discuss the properties and limitations of this new algorithm.

All the above tasks are accomplished in this pioneering work which can be extended in a number of ways. A theoretical mathematical analysis for the interval of convergence can be presented. The iteration formula may be modified to include fractional derivatives or to determine simultaneous multiple roots.

2. DERIVATION OF THE ALGORITHM

Derivation of the new algorithm is presented using the principles of calculus and analytical geometry. In Figure 1, the function $f(x)$ and the tangent circle at the initial point x_0 is shown. x_1 is the intercept of the point with the x axis and (a, b) are the coordinates of the center of the circle. x_r represents the root of the algebraic equation

$$f(x) = 0. \tag{2.1}$$

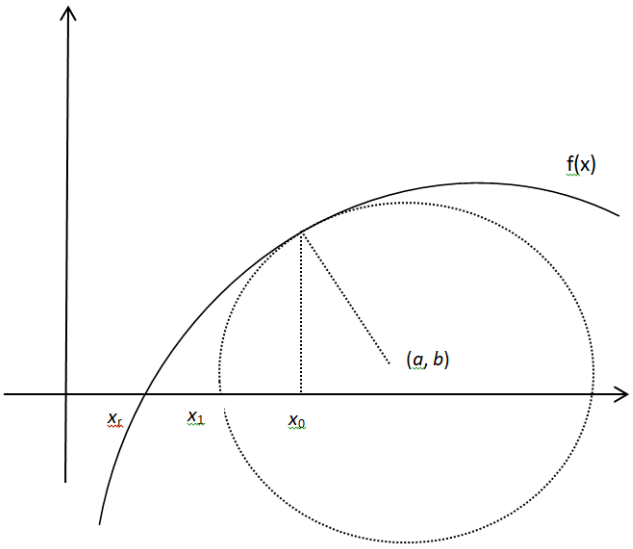

Figure 1: Geometrical sketch of the new algorithm

The equation of the circle is

$$(x-a)^2 + (y-b)^2 = \rho^2, \tag{2.2}$$

where ρ is the radius of curvature of the function at the tangent point

$$\frac{1}{\rho} = \frac{f''}{(1+f'^2)^{3/2}}. \tag{2.3}$$

The intersection points of the circle with the x axis is found by taking $y = 0$ in (2.2)

$$x_1 = a \mp \sqrt{\rho^2 - b^2}. \tag{2.4}$$

The equation of the line connecting the tangent point and the center of the circle is

$$y - f(x_0) = -\frac{1}{f'(x_0)}(x - x_0), \tag{2.5}$$

and the center is on this line, hence

$$b - f(x_0) = -\frac{1}{f'(x_0)}(a - x_0). \tag{2.6}$$

Since $(x_0, f(x_0))$ is a point on the circle, from (2.2)

$$(x_0 - a)^2 + (f(x_0) - b)^2 = \rho^2. \tag{2.7}$$

Solving (2.6) and (2.7) and using (2.3), the center coordinates are

$$a = x_0 - \frac{f'(x_0)\left(1 + f'^2(x_0)\right)}{f''(x_0)}, b = \frac{1 + f'^2(x_0) + f(x_0) f''(x_0)}{f''(x_0)}. \tag{2.8}$$

Substituting (2.8) into (2.4) and employing (2.3),

$$x_1 = x_0 - \frac{f'(x_0)\left(1 + f'^2(x_0)\right)}{f''(x_0)} \mp \frac{1}{|f''(x_0)|}\sqrt{(1 + f'^2(x_0))^3 - (1 + f'^2(x_0) + f(x_0) f''(x_0))^2} \tag{2.9}$$

the first iteration points are determined. Hence the general iteration formula is

$$x_{n+1} = x_n - \frac{f'(x_n)\left(1 + f'^2(x_n)\right)}{f''(x_n)} \mp \frac{1}{|f''(x_n)|}\sqrt{(1 + f'^2(x_n))^3 - (1 + f'^2(x_n) + f(x_n) f''(x_n))^2}$$

$$n = 0, 1, 2, \ldots. \tag{2.10}$$

If $x_{n+1}^{(1)}$corresponds to the negative sign of the last term and $x_{n+1}^{(2)}$the positive sign of the last term, then choose the closest one to the previous iteration

$$x_{n+1} = x_{n+1}^{(1)}, \quad \text{if} \quad \left|x_{n+1}^{(1)} - x_n\right| < \left|x_{n+1}^{(2)} - x_n\right|, \quad \text{else}, \quad x_{n+1} = x_{n+1}^{(2)}. \tag{2.11}$$

The algorithm is summarized in Figure 2.

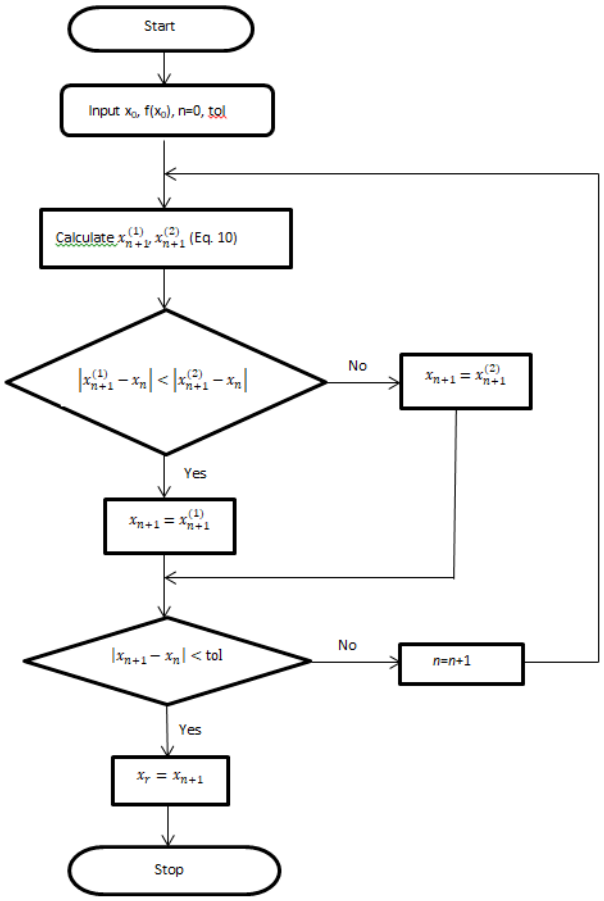

Figure 2: The iteration algorithm

3. PROPERTIES OF THE ALGORITHM

One of the major advantages of the algorithm is that, it works even when the initial point is selected as the local extrema point. It is well-known that the Newton-Raphson algorithm diverges for such points. For local extrema, $f'(x_0) = 0$, and Equation (2.9) reduces to

$$x_1 = x_0 \mp \frac{1}{|f''(x_0)|}\sqrt{1 - (1 + f(x_0)\, f''(x_0))^2} \tag{3.1}$$

For real valued solutions, then

$$\left(1 + f(x_0)\, f''(x_0)\right)^2 < 1. \tag{3.2}$$

This condition cannot be satisfied if $f(x_0)$ and $f''(x_0)$ possess the same sign. Having different signs is a necessary condition not a sufficient condition though. Figure 3 depicts two cases that have the same signs. There are no real valued solutions for the intersection points with the x-axis as also viewed from the figure. In such configurations, other iteration algorithms may also fail because a root may not exist at all or the root is far away from the extremum point.

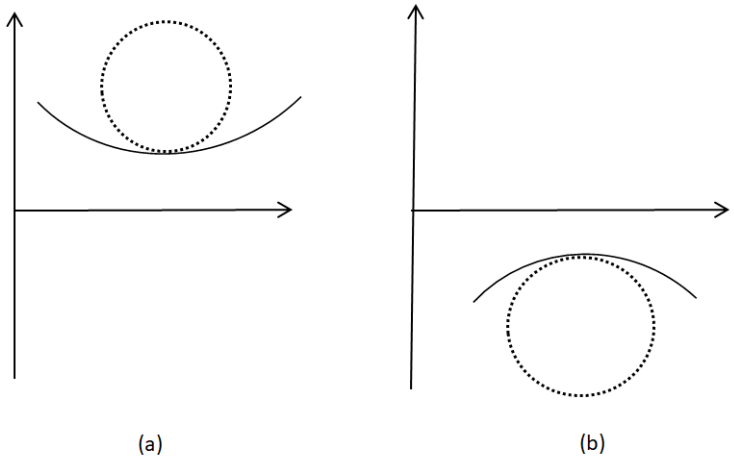

Figure 3: The initial point being a local extrema with no real roots, (a) $f(x_0) > 0$ $f''(x_0) > 0$, (b) $f(x_0) < 0$ $f''(x_0) < 0$

Figure 4 depicts two cases where the function and its derivative have opposite signs.

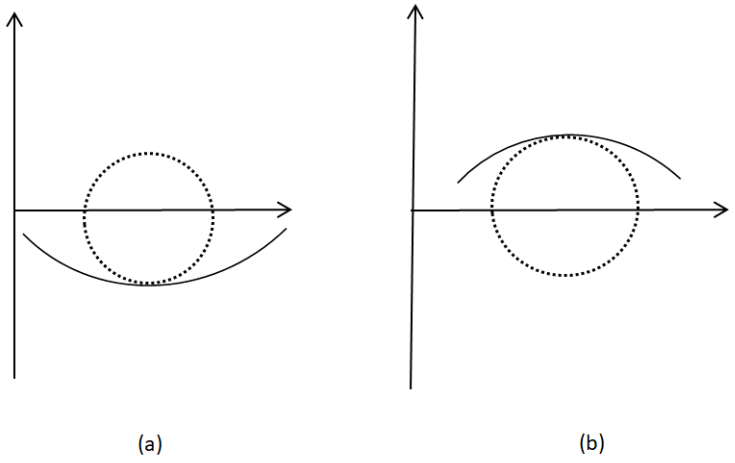

Figure 4: The initial point being a local extrema with real roots, (a) $f(x_0) < 0$ $f''(x_0) > 0$, (b) $f(x_0) > 0$ $f''(x_0) < 0$

The circular arcs may cross the x-axis depending on the numerical values of the functions and their derivatives. In case of opposite signs, if the function or its second derivative is high, condition (3.2) cannot be satisfied. Hence, there may be no intersection points as depicted in Figure 5.

For the special case of satisfaction of Equation (3.2) with a large value of the second derivative, it is probable that the error increases in the calculations. This is also true for the large values of the function itself. Generally speaking, in addition to the necessary condition of opposite signs of the function and its second derivative, $|f(x_0)\, f''(x_0)| \ll 1$ may decrease the error in the calculations.
In general, to have two real roots, the condition for any arbitrary point is

$$\left(1 + f'^2(x_n)\right)^3 > \left(1 + f'^2(x_n) + f(x_n)\, f''(x_n)\right)^2 \tag{3.3}$$

as can be seen from the square root term in (2.10). The tangency condition for the circular arc to the x-axis is

$$\left(1 + f'^2(x_n)\right)^3 = \left(1 + f'^2(x_n) + f(x_n)\, f''(x_n)\right)^2 \tag{3.4}$$

in which case, there is only one calculated root.

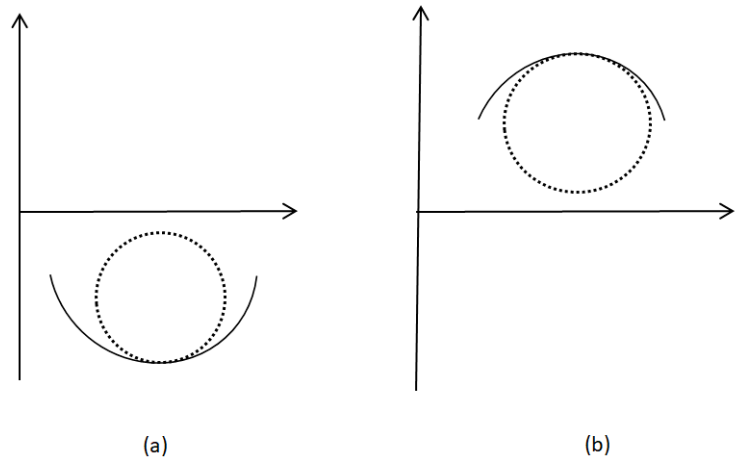

Figure 5: The initial point being a local extrema with no real roots, (a) $f(x_0) < 0$ $f''(x_0) > 0$, (b) $f(x_0) > 0$ $f''(x_0) < 0$

To summarize, the algorithm may fail to converge due to two specific reasons:

1. There is no local root nearby a local extrama as shown in Figure 3.
2. Criterion (3.3) is not satisfied and circular arcs do not cross the x-axis as depicted in Figure 5.

For highly oscillating functions with high amplitudes and high frequencies, Case 2 usually applies. Locating the roots for initial guesses chosen near the peak points is a general problem which leads to divergence or convergence to a distant root with other techniques also.

Another important feature of the algorithm is that it does not diverge even when $f'' \cong 0$. In fact, since for $f'' \cong 0$, the function represents a straight line rather than a curved line, the radius of curvature becomes arbitrarily large, and it is expected that the algorithm reduces to the NR algorithm. To show this, take $f'' = \varepsilon$ where ε is extremely small. From (2.10)

$$x_{n+1} = x_n - \frac{f'\left(1+f'^2\right)}{\varepsilon} + \frac{1}{\varepsilon}\sqrt{(1+f'^2)^3 - (1+f'^2+f\varepsilon))^2}. \tag{3.5}$$

Expand the square term up to $O(\varepsilon)$

$$\left(1+f'^2+f\varepsilon)\right)^2 \cong \left(1+f'^2\right)^2 + 2\varepsilon f(1+f'^2). \tag{3.6}$$

Substitute (3.6) into the square root term and factor out $f'\left(1+f'^2\right)$ from the square root term leading to

$$x_{n+1} = x_n - \frac{f'\left(1+f'^2\right)}{\varepsilon} + \frac{f'\left(1+f'^2\right)}{\varepsilon}\sqrt{1-2\varepsilon\frac{f}{f'^2\,(1+f'^2)}}. \tag{3.7}$$

But expanding the square root term in a first order Taylor series for small ε leads to

$$\sqrt{1-2\varepsilon\frac{f}{f'^2((1+f'^2)}} \cong 1 - \varepsilon\frac{f}{f'^2\,(1+f'^2)}. \tag{3.8}$$

Finally substituting (3.8) into (3.7) yields

$$x_{n+1} = x_n - \frac{f}{f'}, \tag{3.9}$$

which is the well-known Newton Raphson algorithm.

Regarding the convergence rate of the algorithm, the following theorem is posed and proven.

Theorem 1
The Circular Arc Root Algorithm (CARA), i.e. Equation (2.10), has a quadratic convergence rate when sufficiently close to the root

Proof
The function is expanded in the vicinity of the real exact root $x = x_r$

$$f(x_n) = f(x_r) + f'(x_r)(x_n - x_r) + \frac{1}{2!}f''(x_r)(x_n - x_r)^2 + \frac{1}{3!}f'''(x_r)(x_n - x_r)^3 + \dots . \tag{3.10}$$

Since $f(x_r) = 0$, defining the n'th term error as $\in_n = x_n - x_r$ and the constants $c_k = \frac{f^{(k)}(x_r)}{k!f'(x_r)}$, equation (3.5) reduces to

$$f(x_n) = f'(x_r)\left(\in_n + c_2\in_n^2 + c_3\in_n^3 + \dots + c_k\in_n^k + \dots\right). \tag{3.11}$$

The derivatives of (3.6) are

$$f'(x_n) = f'(x_r)\left(1 + 2c_2\in_n + 3c_3\in_n^2 + \dots + kc_k\in_n^{k-1} + \dots\right) \tag{3.12}$$

$$f''(x_n) = f'(x_r)\left(2c_2 + 6c_3\in_n + \dots + k(k-1)c_k\in_n^{k-2} + \dots\right). \tag{3.13}$$

Equations (3.11)-(3.13) can be directly substituted into the iteration equation and necessary simplifications can be done. To reduce the tedious algebra, an alternative approach is used. Note that in the vicinity of the root $f \sim O(\in_n)$, f', $f'' \sim O(1)$. Hence the square root term can be approximated for simplicity

$$\sqrt{(1+f'^2)^3 - (1+f'^2+ff'')^2} = \left(1+f'^2\right)\sqrt{1+f'^2 - \left(\frac{1+f'^2+ff''}{1+f'^2}\right)^2}$$

$$= \left(1+f'^2\right)\sqrt{f'^2 - 2\frac{ff''}{1+f'^2} - \frac{f^2f''^2}{(1+f'^2)^2}} \cong \left(1+f'^2\right)|f'|\left(1 - \frac{ff''}{f'^2(1+f'^2)}\right) \tag{3.14}$$

where the last term with f^2 is much smaller than the other terms and neglected. Substituting the approximations into (2.10)

$$x_{n+1} = x_n - \frac{f'\left(1+f'^2\right)}{f''} \mp \left(1+f'^2\right)\left|\frac{f'}{f''}\right|\left(1 - \frac{ff''}{f'^2(1+f'^2)}\right) \tag{3.15}$$

If $\frac{f'}{f''} > 0$, take the positive sign and if $\frac{f'}{f''} < 0$, take the negative sign. In each case, the approximation reduces to

$$x_{n+1} \cong x_n - \frac{f}{f'} \tag{3.16}$$

Substituting (3.6) and (3.7) into (3.11), subtracting x_r from both sides of the iteration, performing the calculations leads finally to the expression

$$\in_{n+1} = c_2\in_n^2 + O(\in_n^3) \tag{3.17}$$

which states that the convergence rate in the vicinity of the root is quadratic

4. NUMERICAL RESULTS

In this section, numerical simulations of the iteration equation will be presented and compared to the Newton-Raphson solutions.

Example 1
First, the algorithm is tested for the function $x^2 + f^2(x) = 1$ which is the equation of the circle with center at origin and radius 1. For the initial guess of $x_0 = 0$, the Newton-Raphson algorithm $x_{n+1} = x_n - \frac{f}{f'}$ diverges since $f'(0) = 0$. The function and its second derivative are $f(0) = 1$, and $f''(0) = -1$ respectively. Substituting into (2.10) yields $x_1 = \mp 1$, the two roots of the function. In fact for any arbitrary initial point x_0, substituting $f(x_0) = \sqrt{1 - x_0^2}$,
$f'(x_0) = -\frac{x_0}{(1-x_0^2)^{1/2}}$, $f''(x_0) = -\frac{1}{(1-x_0^2)^{3/2}}$ into (2.10) yields again the two roots $x_1 = \mp 1$, as expected.

Example 2
Consider the nonlinear equation

$$f(x) = -\frac{1}{4}x^2 + 1 = 0 \tag{4.1}$$

The roots are calculated and presented in Table 1 for two different starting points. For the first initial point $x_0 = 0$, NR diverges whereas CARA locates both roots up to four digit precision in three iterations. When the initial guess is selected as $x_0 = -1$, then NR converges in four iterations whereas employing the CARA, the closer root is located in two iterations and the farther root in three iterations. For a given single initial value, CARA has the capability of locating two of the roots in contrast with the NR being capable of determining only one root. CARA requires less iteration also.

Table 1: Roots of $f(x) = -\frac{1}{4}x^2 + 1 = 0$

x	NR	CARA (Root1)	CARA (Root 2)
x_0	0	0	0
x_1	Diverges	-1.7321	1.7321
x_2		-1.9987	1.9987
x_3		-2.0000	2.0000
x_0	-1	-1	-1
x_1	-2.5000	-1.9294	2.4294
x_2	-2.0500	-2.0000	2.0041
x_3	-2.0006		2.0000
x_4	-2.0000		

Before treating the next example, CARA is also compared with another well-known geometric method which may be called parabolic method since the convergence to the root is maintained by tangent parabolas [15]. It was shown that the iteration algorithm is indeed a PIA(1,2) algorithm [15]. The algorithm is also known as the irrational Halley formula in the literature [27].

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \mp \frac{1}{f''(x_n)}\sqrt{f'^2(x_n) - 2f(x_n)f''(x_n)} \quad , \qquad n = 0, 1, 2 \ldots \tag{4.2}$$

Note that, since there are two results in the iteration, an iteration algorithm inheriting "logical if" similar to the one given in Figure 2 is required for computations.

Example 3

Consider the nonlinear equation

$$f(x) = e^{-x} - x = 0 \tag{4.3}$$

which has only one real root. The roots are calculated and presented in Table 2 for three different initial guesses. For the two initial values relatively close to the root, CARA and Parabolic algorithms require two iterations whereas NR requires three iterations. However, compared to Parabolic, CARA gets closer to the root after the first iteration. When the initial guess is away from the root, then CARA is the best (two iterations) followed by the Parabolic (three iterations) and then the NR (four iterations).

Table 2: Roots of $f(x) = e^{-x} - x = 0$

x	NR	CARA	Parabolic
x_0	0	0	0
x_1	0.5000	0.5660	0.5858
x_2	0.5663	0.5671	0.5671
x_3	0.5671		
x_0	1	1	1
x_1	0.5379	0.5654	0.5635
x_2	0.5670	0.5671	0.5671
x_3	0.5671		
x_0	5	5	5
x_1	0.0402	0.1212	0.1199
x_2	0.5096	0.5671	0.5751
x_3	0.5665		0.5671
x_4	0.5671		

Regarding the computational times, for x_0=5, the elapsed computational time is 0.000249 seconds for NR and 0.000769 seconds for CARA and 0.000940 seconds for Parabolic for the programs written in MATLAB. The computational time is least for NR. The second lowest time is due to CARA and the longest CPU time is due to Parabolic. The increase in computational time in the latter two stems from the complexities in the iteration equations and the "logical if" in the algorithm as given in Figure 2. (A similar algorithm as in Figure 2 can be employed for Parabolic Iteration Algorithm).

Example 4

Consider the nonlinear equation

$$f(x) = \tan x - \tanh x = 0 \tag{4.4}$$

which has infinitely many roots. The roots are calculated and presented in Table 3 for three different initial guesses. In the first two cases, CARA converged to the nearest root and a distant root with less iteration compared to the NR. When the initial guess is $x_0 = 3$, CARA converged to two of the other distant roots whereas NR converged to the nearest one.

5. CONCLUDING REMARKS

The following conclusions can be derived from the analysis:

- A new non-divergent algorithm named Circular Arc Root Algorithm (CARA) is proposed for the first time to locate the roots of nonlinear algebraic equations.
- The algorithm uses circles and their intersections with the x-axis, having radius of curvatures equal to the radius of curvature of the function at the tangent point.

Table 3: Roots of $f(x) = \tan x - \tanh x = 0$

x	NR	CARA (Root 1)	CARA (Root 2)
x_0	3.5	3.5	3.5
x_1	4.0486	-2.9986	3.9507
x_2	3.9426	-16.5662	3.9266
x_3	3.9269	-16.4935	
x_4	3.9266	-16.4934	
x_0	4.5	4.5	4.5
x_1	4.3384	-104.6401	4.2167
x_2	4.1317	-104.4627	3.9531
x_3	3.9738	-104.4580	3.9266
x_4	3.9289		
x_5	3.9266		
x_0	3.0	3.0	3.0
x_1	4.1258	16.7281	4.3291
x_2	3.9710	16.5055	-16.3667
x_3	3.9286	16.4934	-16.4934
x_4	3.9266		

- The conditions for the algorithm to work are discussed.
- The algorithm has quadratic convergence rate when sufficiently close to the root.
- The algorithm successfully locates the root when the initial guess is in the exact location or in the vicinity of a local extrema whereas Newton-Raphson (NR) algorithm fails to locate the root for such initial guesses.
- For a given initial guess, two of the roots can be determined by CARA in contrast with locating only one root by employing NR.
- The algorithm performed better than the well-known parabolic algorithm (Irrational Halley Iteration) for the example considered regarding the number of iterations and computational times.
- For the sample problems considered, CARA requires less iterations compared to the NR with a higher computational time due to the complexities in the iterations.

References

[1] S. Abbasbandy, Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method, Applied Mathematics and Computation, 145, 887-893, 2003. 1
[2] S. Abbasbandy, Modified Homotopy Perturbation Method for nonlinear equations and comparison with Adomian Decomposition Method, Applied Mathematics and Computation, 172, 431-438, 2006. 1
[3] A. Constantin and R. S. Johnson, The dynamics of waves interacting with the Equatorial undercurrent, Geophysical and Astrophysical Fluid Dynamics, 109(4), 311-358, 2015. 1
[4] Y. Ding, C. Sui and J. Li, An experimental investigation into combustion fitting in a direct injection marine diesel engine, Applied Sciences, 8, 2489, 2018. 1
[5] B. Ghanbari, A new general fourth order family of methods for finding simple roots of nonlinear equations, Journal of King Saud University, 23, 395-398, 2011. 1
[6] T. Gemechu and S. Thota, On new root finding algorithms for solving nonlinear transcendental equations, International Journal of Chemistry, Mathematics and Physics, 4(2), 18-24, 2020. 1

[7] A. S. Householder, The numerical treatment of a single nonlinear equation, McGraw-Hill, New York, 1970. 1
[8] N. Jaksic, Numerical algorithm for natural frequencies computation of an axially moving beam model, Meccanica, 44, 687-695, 2009. 1
[9] S. M. Kang, A. Naseem, W. Nazeer and A. Jan, Modification of Abbasbandy's method and polynomiography, International Journal of Mathematical Analysis, 10, 1197-1210, 2016. 1
[10] A. Naseem, M. A. Rehman and Jihad Younis, A new root finding algorithm for solving real world problems and its complex dynamics via computer technology, Complexity, 2021, Article ID 6369466, 2021. 1
[11] A. Özyapıcı, Z. B. Şensoy and T. Karanfiller, Effective root finding methods for nonlinear equations based on multiplicative calculi, Journal of Mathematics, 2016. 1
[12] M. Pakdemirli and H. A. Yurtsever, Estimating roots of polynomials using perturbation theory, Applied Mathematics and Computation, 188, 2025-2028, 2007. 1
[13] M. Pakdemirli and G. Sarı, A comprehensive perturbation theorem for estimating magnitudes of roots of polynomials, LMS Journal of Computation and Mathematics 16,1-8, 2013. 1
[14] M. Pakdemirli, G. Sarı and N. Elmas, Approximate determination of polynomial roots, Applied and Computational Mathematics 15(1), 67-77, 2016. 1
[15] M. Pakdemirli and H. Boyacı, Generation of Root Finding Algorithms via Perturbation Theory and Some Formulas, Applied Mathematics and Computation, 184, 783-788, 2007. 1, 4
[16] M. Pakdemirli, H. Boyacı and H. A. Yurtsever, Perturbative derivation and comparisons of root-finding algorithms with fourth order derivatives, Mathematical and Computational Applications, 12(2), 117-124, 2007. 1
[17] M. Pakdemirli, H. Boyacı and H. A. Yurtsever, A root finding algorithm with fifth order derivatives, Mathematical and Computational Applications, 13(2), 123-128, 2008. 1
[18] M. Pakdemirli and İ. T. Dolapci, Functional root algorithms for transcendental equations, Applied and Computational Mathematics, 23(1), 99-109, 2024. 1
[19] S. Qureshi, H. Ramos and A. K. Soamro, A new nonlinear ninth order root finding method with error analysis and basins of attraction, Mathematics, 9, 1996, 2021. 1
[20] H. Susanto and N. Karjanto, Newton's methods' basins of attraction revisited, Applied Mathematics and Computation, 215, 1084-1090, 2009. 1
[21] M. Shams, N. Rafiq, N. Kausar, P. Agarwal, N. A. Mir, and Y. M. Li, On highly efficient simultaneous schemes for finding all polynomial roots, Fractals, 30(10), p.2240198, 2022. 1
[22] M. Shams, N. Rafiq, N. Kausar, P. Agarwal, C. Park and N. A. Mir, On iterative techniques for estimating all roots of nonlinear equation and its system with application in differential equation, Advances in Difference Equations, 2021, 1-18, 2021. 1
[23] M. Shams, N. Kausar, P. Agarwal, S. Jain, M. A. Salman and M. A. Shah, M.A., On family of the Caputo-type fractional numerical scheme for solving polynomial equations, Applied Mathematics in Science and Engineering, 31(1), p.2181959, 2023. 1
[24] M. Shams and B. Carpentieri, An efficient and stable caputo-type inverse fractional parallel scheme for solving nonlinear equations, Axioms, 13(10), p.671, 2024. 1
[25] C. L. Sabharwal, An iterative hybrid algorithm for roots of nonlinear equations, Engineering, 2, 80-98, 2021. 1
[26] M. Türkyılmazoğlu, A simple algorithm for high order Newton iteration formula and some new variants, Hacettepe Journal of Mathematics and Statistics, 49(1), 425-438, 2020. 1
[27] C. Vanhille, A note on the convergence of the irrational Halley's iterative algorithm for solving nonlinear equations, International Journal of Computer Mathematics, 97(9), 1840-1848, 2020. 1, 4
[28] C. Vanhille, An improved secant algorithm of variable order to solve nonlinear equations based on the disassociation of numerical approximations and iterative progression, Journal of Mathematics Research, 12(6), 50-65, 2020. 1
[29] Y. Wu and K. F. Cheung, Two parameter homotopy method for nonlinear equations, Numerical Algorithms, 53, 555-572, 2010. 1