**Journal of Informatics and Mathematical Sciences**
Vol. 17, No. 4, pp. 365–374, 2025
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications
DOI: 10.26713/jims.v17i4.3334

http://www.rgnpublications.com

Research Article

# A New Non-Divergent Root Finding Algorithm

Mehmet Pakdemirli

*Department of Mechanical Engineering, Manisa Celal Bayar University, Manisa, Turkey*
pakdemirli@gmail.com

**Received:** August 24, 2025 **Revised:** September 26, 2025
**Accepted:** October 8, 2025 **Published:** December 30, 2025

**Abstract.** A new iteration algorithm is proposed. The algorithm does not diverge when the first derivative is zero or nearly zero as in the case of Newton-Raphson method. The convergence rate of the new algorithm is linear compared to the quadratic convergence of the Newton-Raphson method (The Newton-Raphson method is faster). The algorithm significantly increases the interval of convergence for roots. A hybrid algorithm combining the increase in the range of convergence of the new method and the faster rate of convergence of the Newton-Raphson method is suggested. The criterion to select the best choice during the running of the algorithm is given. Numerical examples are treated and the three methods (Non-Divergent Algorithm, Newton-Raphson method and the Hybrid method) are contrasted with each other. The hybrid method is recommended since it decreases the number of iterations and increases the range of convergence.

**Keywords.** Root finding algorithms, Newton-Raphson method, Non-divergent algorithm, Nonlinear equations

**Mathematics Subject Classification (2020).** 65H05, 65D15



## 1. Introduction

Root finding algorithms are of special importance in many branches of science. Problems involving wave motion (Constantin and Johnson [2]), natural frequencies of vibration (Jaksic [7]), combustion in diesel engines (Ding *et al*. [3]), Van der Walls equation in chemistry (Naseem *et al*. [10], Qureshi *et al*. [15]), blood rheology [10], open channel flow [10, 15], Plank's radiation law [10, 15] all require solutions of transcendental equations employing such algorithms.

Polynomiography (Kalantari [8], Kang *et al*. [9], Naseem *et al*. [10], Susanto and Karjanto [18]) is another mathematical arts branch producing excellent color graphics based on the convergence of the polynomial roots.

Newton-Raphson method (Householder [6]) is one of the most common root-finding techniques preferred for its simplicity and fast convergence rate of quadratic nature. However, the initial guess must be sufficiently close to the root. Otherwise, the algorithm may either diverge or converge to another root.

A systematic way of producing many single iteration formulas were depicted by combining the perturbation concepts and Taylor series expansions (Pakdemirli and Boyacı [13]). The algorithms were called perturbation-iteration algorithms, PIA$(n,m)$, where $n$ denotes the number of correction terms in the perturbation expansions and m denotes the highest order derivatives in the Taylor series expansions. Within the formalism, many well-known techniques in addition to new algorithms have already been developed. Newton-Raphson method is essentially PIA$(1,1)$, irrational Halley formula is PIA$(1,2)$, Householder's or Euler's iteration is PIA$(2,2)$. Algorithms derived by the Adomian Decomposition Method (Abbasbandy [1]) were contrasted with the PIA$(3,3)$ algorithm given in [13] containing third order derivatives. Extensions of the PIA algorithms to fifth order (Pakdemirli *et al*. [14]) derivatives were also made. Two-parameter Homotopy Method (Wu and Cheung [21]) produced the same algorithm of PIA$(3,3)$ given in [13].

As far as the root finding techniques are considered, two major properties, namely the convergence rate and the interval of convergence are of high importance. Higher order convergence rate algorithms were proposed as extensions to the Newton-Raphson method [4, 5, 11, 12, 16, 17, 19, 20]. Despite their lower number of iterations and possibly wider range of convergence, those methods require extensive algebraic calculations compared to the simple algorithm of the Newton-Raphson.

The problem of finding a simple algorithm requiring fewer computations with an enlarged interval of convergency is most desirable which is addressed in this study. First, a non-divergent simple algorithm involving only first order derivatives is derived. The algorithm increases the range of convergence of the Newton-Raphson method with a deficiency of linear convergence rate. A hybrid algorithm is then suggested combining the non-divergence property of the new algorithm with the faster convergence rate of the Newton-Rapson method. For a far-away guess of the real root, the non-divergent algorithm first brings the iterations to sufficiently close intervals so that the Newton-Raphson method determines the roots with a faster convergence rate. Three simple numerical cases are considered to show the efficiency of the hybrid algorithm.

## 2. The Non-Divergent Algorithm

A geometric representation of the new algorithm is given in Figure 1. A tangent line is drawn from the initial guess functional point. In the Newton-Raphson method, the first iteration point is determined by crossing the tangent line with the x-axis. On the contrary, in the proposed method, the tangent line and a vertical line drawn from the point $(x_0, 0)$ intersects the tangent perpendicularly with the intersection point being the first iteration. Repeating the process, finally the root is determined.

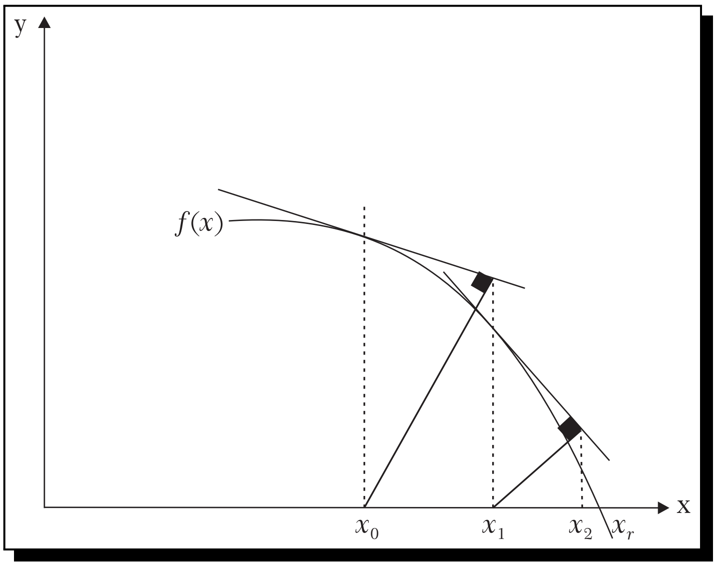


**Figure 1.** Geometrical representation of the new algorithm

For the nonlinear equation to be solved

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

the tangent line to the function at point $x=x_0$ is

$$y=f(x_0)+f'(x_0)(x-x_0). \tag{2.2}$$

The perpendicular line to this tangent from point $(x_0,0)$ is

$$y=-\frac{1}{f'(x_0)}(x-x_0). \tag{2.3}$$

The two lines cross each other at point $x=x_1$ which is found by equating (2.2) and (2.3) and solving for $x$

$$x_1=x_0-\frac{f(x_0)f'(x_0)}{1+f'^2(x_0)}. \tag{2.4}$$

The iteration equation is then

$$x_{n+1}=x_n-\frac{f(x_n)f'(x_n)}{1+f'^2(x_n)}, \quad n=0,1,2,\dots \tag{2.5}$$

which will be called the *Non-Divergent Algorithm* (NDA) which does not diverge for the case of $f'(x_n)\cong 0$ because the denominator of the correction term is always greater or equal to 1. In the case of Newton-Raphson algorithm, however

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}, \quad n=0,1,2,\dots, \tag{2.6}$$

$f'(x_n)\cong 0$ makes the correction term very large causing divergence. Taking limits of the correction term of NDA for $f'(x_n)$,

$$\lim_{f'\to 0}\frac{ff'}{1+f'^2}=0, \quad \lim_{f'\to\infty}\frac{ff'}{1+f'^2}=0. \tag{2.7}$$

It is readily observed that the correction term vanishes in the limiting cases of the derivative.

The non-divergence advantage introduces a disadvantage, however, that is, a slower convergence to the root. Comparing the correction terms of both methods, the absolute value of the NR correction term is always larger than the NDA correction term

$$\left|\frac{f(x_n)}{f'(x_n)}\right| > \left|\frac{f(x_n)f'(x_n)}{1+f'^2(x_n)}\right| \tag{2.8}$$

which can be seen by equating the denominators and comparing the numerators

$$|f(x_n)|\ |1+f'^2(x_n)| > |f(x_n)f'^2(x_n)|. \tag{2.9}$$

This fact can be geometrically observed from Figure 1 also. The following theorem is posed for the convergence of the algorithm

**Theorem 2.1.** *The Non-Divergent algorithm* (2.5) *has a linear convergence rate when sufficiently close to the root.*

*Proof.* Expand the function and its derivative in the vicinity of the real root $x = x_r$,

$$f(x_n) = f(x_r) + f'(x_r)(x_n - x_r) + \frac{1}{2!}f''(x_r)(x_n - x_r)^2 + \frac{1}{3!}f'''(x_r)(x_n - x_r)^3 + \dots. \tag{2.10}$$

Since $f(x_r) = 0$, defining the $n$th term error as $\varepsilon_n = x_n - x_r$ and the constants $c_k = \frac{f^{(k)}(x_r)}{k!f'(x_r)}$. Equation (2.10) reduces to

$$f(x_n) = f'(x_r)(\varepsilon_n + c_2\varepsilon_n^2 + c_3\varepsilon_n^3 + \dots). \tag{2.11}$$

The derivative can be expressed similarly

$$f'(x_n) = f'(x_r)(1 + 2c_2\varepsilon_n + 3c_3\varepsilon_n^2 + \dots). \tag{2.12}$$

Substituting the expressions to the iteration equation, subtracting $x_r$ from both sides of the iteration, performing the calculations leads finally to the expression

$$\varepsilon_{n+1} = \frac{1}{1+f'^2(x_r)}\varepsilon_n + O(\varepsilon_n^2) \tag{2.13}$$

which states that the convergence rate is linear.

This is definitely a slower convergence rate compared to the faster quadratic convergence rate of the Newton-Raphson method. As $|f'(x_r)| \to \infty$, the convergence rate becomes quadratic, similar to the NR case.

Another important feature of the algorithm is that it may locate the local extrema points. The stable local extrema (max or min points) with respect to the NDA is shown in Figure 2. According to the figure, the branches of the function in the vicinity of the extrema diverge from the x-axis with no root in the vicinity. If the iteration is in the vicinity of such an extremum, the algorithm converges to the extremum point. On the contrary, the unstable extrema with respect to NDA are given in Figure 3. A small disturbance from the point results in a divergence from the extremum and the algorithm converges to the root. The range of convergence of the algorithm is therefore much wider than the corresponding Newton-Raphson algorithm. For Figure 3, the range of convergence of the second root starts from the local maximum point, excluding the maximum point and ends up at the local minimum point, excluding the minimum point itself. □

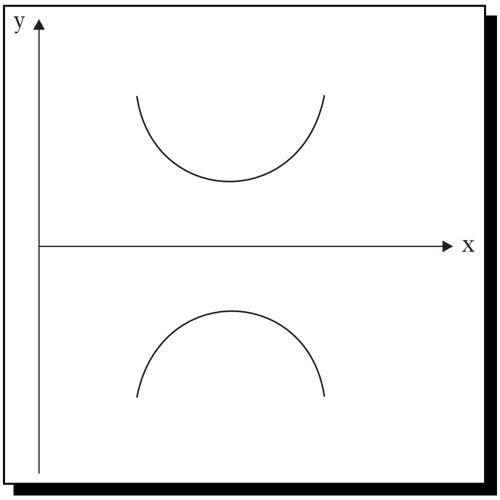


**Figure 2.** Stable local extrema

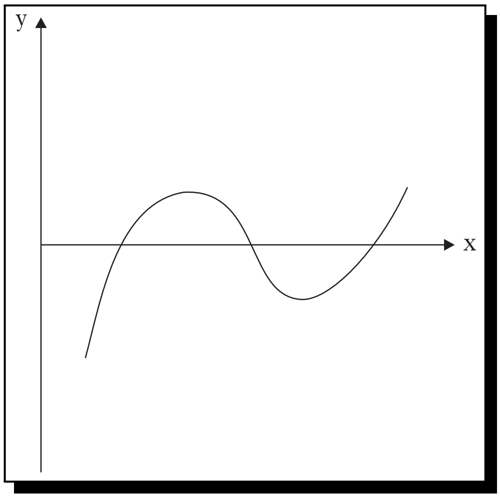


**Figure 3.** Unstable local extrema

## 3. A Hybrid Algorithm Combining NR and NDA

In general, when iterate is far from the root and $f'(x_n) \cong 0$, while the Newton-Raphson algorithm diverges or converges to another root, the Non-Divergent algorithm converges to the nearest root. When the iteration gets closer to the root, NR converges faster than the NDA as the former has a quadratic convergence, but the latter has a linear convergence. A combination of both methods would then produce an algorithm which possesses non-divergence and faster convergence properties in addition to being simple. One has to develop a criterion of switching from one algorithm to another.

A simple reasoning might be to use NDA when $f'(x_n) < 1$ and NR when $f'(x_n) \geq 1$. Using this criterion, the best algorithm is not obtained always. Instead, a better criterion can be derived using the concept of perturbations. In [13], the Newton-Raphson algorithm is derived as the Perturbation Iteration Algorithm of PIA(1,1). It is pointed out there that the correction term should be much smaller than the leading term for a consistent perturbative solution. In the case of Newton-Raphson formula, this corresponds to

$$Cr = \left| \frac{f(x_n)}{x_n f'(x_n)} \right| \ll 1. \tag{3.1}$$

The ‘much smaller condition can be interpreted in several ways and the criterion can be taken different without violating (3.1) of course. One may choose this number to be as large as 0.25 and the hybrid algorithm would then be

$$\left\{ \begin{array}{l} \text{If } Cr > 0.25 \text{ use NDA} \\ \text{If } Cr \leq 0.25 \text{ use NR} \end{array} \right\}. \tag{3.2}$$

The numerical simulations and comparisons will be given in the next section.

## 4. Numerical Simulations

Three sample problems will be solved by the algorithms NR, NDA and the Hybrid. The starting points are intentionally chosen to be points which are away from the roots and for which the first derivatives are extremely small.

**Example 4.1.** Consider the nonlinear equation

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

The roots are calculated and presented in Table 1 starting from a very close point to the maximum point. If the algorithm starts from the exact location of the maximum point $x_0 = 0$, then NR diverges and NDA and Hybrid algorithms cannot advance at all. A small deviation, albeit infinitesimal, is necessary for NDA to progress for such critical points.

**Table 1.** Roots of Example 4.1

| $x$ | NR | NDA | Hybrid | $Cr$ |
|---|---|---|---|---|
| $x_0$ | 0.0100 | 0.0100 | 0.0100 (NDA) | 5000 |
| $x_1$ | 50.0050 | 0.0300 | 0.0300 (NDA) | 555 |
| $x_2$ | 25.0125 | 0.0897 | 0.0897 (NDA) | 61.6 |
| $x_3$ | 12.5262 | 0.2621 | 0.2621 (NDA) | 6.78 |
| $x_4$ | 6.3030 | 0.6451 | 0.6451 (NDA) | 0.7 |
| $x_5$ | 3.2308 | 0.9278 | 0.9278 (NDA) | 0.08 |
| $x_6$ | 1.7702 | 0.9859 | 1.0028 (NR) | 0.003 |
| $x_7$ | 1.1675 | 0.9972 | 1.0000 (NR) | |
| $x_8$ | 1.0120 | 0.9994 | | |
| $x_9$ | 1.0000 | 0.9999 | | |
| $x_{10}$ | | 1.0000 | | |

Since the criterion value ($Cr$) is very large, NR falls far away from the root in the first iteration, and it takes a number of iterations to converge to the root. On the contrary, NDA approaches to the root decreasing the distance at each step. When the iterations are near to the root, it is obvious that NR performs better that the NDA since the former has a quadratic convergence and the latter a linear convergence. Based on the criterion given in (3.1) and (3.2), the Hybrid algorithm starts with NDA, gradually approaches to the root, and when sufficiently close, switches to NR for faster convergence. Overall, the number of iterations is smallest for the hybrid algorithm.

**Example 4.2.** Consider the nonlinear equation

$$f(x) = x^3 + 3x^2 - 4 = 0. \tag{4.2}$$

The initial guess starts from a very close point to the minimum point of the function. If one starts from the point of $x_0 = 0$, then NR diverges and NDA and Hybrid algorithms cannot advance at all. The iterations are given in Table 2.

Because the initial point is far from the root and has a vanishing first derivative, the first iteration of NR appears to be far away from the root. The criterion indicates that the correction term is much higher than the leading term. Although NDA has a first order convergence, since the iterations do not throw the estimates far away from the root, the number of iterations are much less than NR. The hybrid algorithm starts with NDA again and switches to NR when the criterion is met. Hybrid algorithm performs slightly better than the NDA in this specific case.

**Table 2.** Roots of Example 4.2

| $x$ | NR | NDA | Hybrid | $Cr$ |
|---|---|---|---|---|
| $x_0$ | 0.0100 | 0.0100 | 0.0100 (NDA) | 6.63 |
| $x_1$ | 66.3400 | 0.2503 | 0.2503 (NDA) | 8.98 |
| $x_2$ | 43.9034 | 1.9142 | 1.9142 (NDA) | 0.33 |
| $x_3$ | 28.9508 | 1.2923 | 1.2923 (NDA) | 0.19 |
| $x_4$ | 18.9902 | 1.0456 | 1.0441 (NR) | 0.04 |
| $x_5$ | 12.3619 | 1.0018 | 1.0012 (NR) | 0.001 |
| $x_6$ | 7.9619 | 1.0000 | 1.0000 (NR) | |
| $x_7$ | 5.0583 | | | |
| $x_8$ | 3.1707 | | | |
| $x_9$ | 1.9907 | | | |
| $x_{10}$ | 1.3287 | | | |
| $x_{11}$ | 1.0542 | | | |
| $x_{12}$ | 1.0019 | | | |
| $x_{13}$ | 1.0000 | | | |

**Example 4.3.** Consider the first positive root of the cosine function

$$f(x) = \cos x = 0. \tag{4.3}$$

The iterations of the three methods are contrasted in Table 3.

**Table 3.** Roots of Example 4.3

| $x$ | NR | NDA | Hybrid | Cr |
|---|---|---|---|---|
| $x_0$ | 0.1000 | 0.1000 | 0.1000 (NDA) | 99.67 |
| $x_1$ | 10.0666 | 0.1984 | 0.1984 (NDA) | 25.07 |
| $x_2$ | 11.4045 | 0.3843 | 0.3843 (NDA) | 6.43 |
| $x_3$ | 10.9711 | 0.6891 | 0.6891 (NDA) | 1.76 |
| $x_4$ | 10.9956 | 1.0385 | 1.0385 (NDA) | 0.57 |
| $x_5$ | Converged to another root | 1.2895 | 1.2895 (NDA) | 0.22 |
| $x_6$ | | 1.4282 | 1.5785 (NR) | 0.005 |
| $x_7$ | | 1.4992 | 1.5708 (NR) | |
| $x_8$ | | 1.5350 | | |
| $x_9$ | | 1.5529 | | |
| $x_{10}$ | | 1.5618 | | |
| $x_{11}$ | | 1.5663 | | |
| $x_{12}$ | | 1.5686 | | |
| $x_{13}$ | | 1.5697 | | |
| $x_{14}$ | | 1.5702 | | |
| $x_{15}$ | | 1.5705 | | |
| $x_{16}$ | | 1.5707 | | |
| $x_{17}$ | | 1.5707 | | |
| $x_{18}$ | | 1.5708 | | |

The Newton Raphson method did not converge to the desired (or closest) root, but converged to a root farther away from the initial guess. The slow convergence of NDA is best seen from this example since 18 iterations are needed to locate the root with 4 digit precision. The hybrid algorithm however cuts the number of iterations by more than half. As can be seen, the Hybrid algorithm approaches the root safely without any divergence with NDA and when it gets closer, (The criterion is satisfied) switches to the NR for a fast convergence.

Finally, in Table 4, the convergence intervals of all methods are given for the first positive root.

**Table 4.** Interval of convergence for the first positive root

| $x$ | NR | NDA | Hybrid |
|---|---|---|---|
| Interval of Convergence | $[0.41, \pi - 0.41]$ | $(0, \pi)$ | $(0, \pi)$ |

NDA and Hybrid method has an enlarged interval of convergence compared to the NR.

**Example 4.4.** Consider the positive root of the function

$$f(x) = 2e^{-x} + x^2 - 4 = 0. \tag{4.4}$$

The iterations of the three methods are contrasted in Table 5.

**Table 5.** Roots of Example 4.4

| $x$ | NR | NDA | Hybrid | $Cr$ |
|---|---|---|---|---|
| $x_0$ | 0.6 | 0.6 | 0.6 (NDA) | 41.39 |
| $x_1$ | 25.4335 | 0.8576 | 0.8576 (NDA) | 0.50 |
| $x_2$ | 12.7954 | 2.0534 | 2.0534 (NDA) | 0.06 |
| $x_3$ | 6.5540 | 1.9383 | 1.9305 (NR) | 0.003 |
| $x_4$ | 3.5813 | 1.9267 | 1.9257 (NR) | |
| $x_5$ | 2.3316 | 1.9258 | | |
| $x_6$ | 1.9667 | 1.9257 | | |
| $x_7$ | 1.9263 | | | |
| $x_8$ | 1.9257 | | | |

The starting point is selected as a specific point where the derivative of the function is near to zero. The Newton-Raphson algorithm converged after 8 iterations. The non-divergent algorithm converged after 6 iterations. The hybrid algorithm converged after 4 iterations. The hybrid algorithm started with NDA which works well for small derivatives and when the iterations come closer to the root, a switch to NR with a quadratic convergence rate reduced the iterations.

## 5. Concluding Remarks

The following conclusions can be derived from the analysis

(i) A new non-divergent algorithm is proposed.

(ii) The algorithm has linear convergence rate.

(iii) For initial guesses far from the root, the new algorithm performs better than Newton-Raphson method and when the iterations get closer to the root, Newton-Raphson performs better.

(iv) A new hybrid algorithm combining the advantages of both methods is proposed.

(v) The criterion for selecting which of the algorithms to employ is given.

(vi) The hybrid algorithm is simple, possesses an increased range of convergence compared to the Newton-Raphson method, and converges to the root more quickly when the iterate is close to the root than the non-divergent root algorithm.

The algorithm has been developed for single variable nonlinear equations. Extension of the algorithm for multivariable case is a future research topic. Using the same geometric approach, higher order root algorithms with better convergence rates can also be developed.

## Competing Interests

The author declares that he has no competing interests.

## Authors' Contributions

The author wrote, read and approved the final manuscript.

## References

**[1]** S. Abbasbandy, Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method, *Applied Mathematics and Computation* **145**(2-5) (2003), 887 – 893, DOI: 10.1016/S0096-3003(03)00282-0.

**[2]** A. Constantin and R. S. Johnson, The dynamics of waves interacting with the Equatorial undercurrent, *Geophysical and Astrophysical Fluid Dynamics* **109**(4) (2015), 311 – 358, DOI: 10.1080/03091929.2015.1066785.

**[3]** Y. Ding, C. Sui and J. Li, An experimental investigation into combustion fitting in a direct injection marine diesel engine, *Applied Sciences* **8**(12) (2018), 2489, DOI: 10.3390/app8122489.

**[4]** T. Gemechu and S. Thota, On new root finding algorithms for solving nonlinear transcendental equations, *International Journal of Chemistry, Mathematics and Physics* **4**(2) (2020), 18 – 24, DOI: 10.22161/ijcmp.4.2.1.

**[5]** B. Ghanbari, A new general fourth-order family of methods for finding simple roots of nonlinear equations, *Journal of King Saud University* **23**(4) (2010), 395 – 398, DOI: 10.1016/j.jksus.2010.07.018.

**[6]** A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, (1970).

**[7]** N. Jaksic, Numerical algorithm for natural frequencies computation of an axially moving beam model, *Meccanica* **44** (2009), 687 – 695, DOI: 10.1007/s11012-009-9203-5.

**[8]** B. Kalantari, Polynomiography and application in art, education, and science, *Computers & Graphics* **28**(3) (2004), 417 – 430, DOI: 10.1016/j.cag.2004.03.009.

**[9]** S. M. Kang, A. Naseem, W. Nazeer and A. Jan, Modification of Abbasbandy's method and polynomiography, *International Journal of Mathematical Analysis* **10**(24) (2016), 1197 – 1210, DOI: 10.12988/ijma.2016.6898.

**[10]** A. Naseem, M. A. Rehman and J. Younis, A new root-finding algorithm for solving real-world problems and its complex dynamics via computer technology, *Complexity* **2021**(1) (2021), 6369466, DOI: 10.1155/2021/6369466.

**[11]** A. Özyapıcı, Effective numerical methods for non-linear equations, *International Journal of Applied and Computational Mathematics* **6** (2020), article number 35, DOI: 10.1007/s40819-020-0785-x.

**[12]** A. Özyapıcı, Z. B. Sensoy and T. Karanfiller, Effective root-finding methods for nonlinear equations based on multiplicative calculi, *Journal of Mathematics* **2016**(1) (2016), 8174610, DOI: 10.1155/2016/8174610.

**[13]** M. Pakdemirli and H. Boyacı, Generation of root finding algorithms via perturbation theory and some formulas, *Applied Mathematics and Computation* **184**(2) (2007), 783 – 788, DOI: 10.1016/j.amc.2006.05.207.

**[14]** M. Pakdemirli, H. Boyacı and H. A. Yurtsever, A root-finding algorithm with fifth order derivatives, *Mathematical and Computational Applications* **13**(2) (2008), 123 – 128, DOI: 10.3390/mca13020123.

**[15]** 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**(16) (2021), 1996, DOI: 10.3390/math9161996.

**[16]** N. H. A. Rahman, A. Ibrahim and M. I. Jayes, Higher order homotopy Taylor perturbation via start-system, *Applied Mathematics and Computational Intelligence* **2** (2013), 85 – 94.

**[17]** C. L. Sabharwal, An iterative hybrid algorithm for roots of non-linear equations, *Engineering* **2**(1) (2021), 80 – 98, DOI: 10.3390/eng2010007.

**[18]** H. Susanto and N. Karjanto, Newton's method's basins of attraction revisited, *Applied Mathematics and Computation* **215**(3) (2009), 1084 – 1090, DOI: 10.1016/j.amc.2009.06.041.

**[19]** M. Turkyilmazoglu, A simple algorithm for high order Newton iteration formula and some new variants, *Hacettepe Journal of Mathematics and Statistics* **49**(1) (2020), 425 – 438, DOI: 10.15672/hujms.459810.

**[20]** 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) (2020), 50 – 65, DOI: 10.5539/jmr.v12n6p50.

**[21]** Y. Wu and K. F. Cheung, Two-parameter homotopy method for nonlinear equations, *Numerical Algorithms* **53** (2010), 555 – 572, DOI: 10.1007/s11075-009-9319-7.