Ceylon Journal of Science 55 (4) 2026: 1735-1743

RESEARCH ARTICLE

An Anti-Gaussian Quadrature Rule Based Iterative Method for Nonlinear Equations

T. Singh, D. K. Behera, R. B. Dash and R. K. Saeed

Highlights

- Developed a third-order iterative algorithm for solving nonlinear equations
- Method combines anti-Gaussian technique with a Haar wavelet-based quadrature rule
- Achieves faster convergence, fewer iterations, and lower errors than Newton's method
- Demonstrates superior accuracy and efficiency across diverse test functions

Ceylon Journal of Science 55 (4) 2026: 1735-1743
DOI: https://doi.org/10.4038/cjs.v55i4.9149

RESEARCH ARTICLE

An Anti-Gaussian Quadrature Rule Based Iterative Method for Nonlinear Equations

Tusar Singh[1], Dwiti Krushna Behera[2*], Rajani Ballav Dash[2] and Rostam Kareem Saeed[3]

[1]*Department of Mathematics, Siksha O Anusandhan (Deemed to be University), Bhubaneswar, India*
[2*]*Department of Mathematics, Ravenshaw University, Kataka, Odisha, India*
[3]*Department of Mathematics, College of Science, Salahaddin University, Erbil, Iraq*

Received: 28.08.2025 *Accepted: 15.10.2026*

Abstract

In this study, a groundbreaking third-order algorithm is unveiled for tackling nonlinear equations. This innovative algorithm builds upon Newton's method, enhanced by an anti-Gaussian technique applied to the Haar wavelet-based quadrature rule. Through this advancement, the convergence order of Newton's method has been improved significantly. The new iterative approach boasts a convergence order of three, surpassing its predecessor. Various functions are tested to showcase the remarkable efficiency of this new method. The number of iterations, approximate roots, convergence order, runtime, and absolute error of the roots are computed in the numerical computation section. The analysis reveals that the newly developed algorithm not only requires fewer iterations but also achieves shorter runtimes, lower absolute errors, and a higher convergence order for each function as compared to other existing methods. This stands as a testament to the power of innovation in mathematical solutions.

Keywords: Anti Gaussian technique; Newton-Raphson Method; Order of convergence; Taylor's Series

INTRODUCTION

Iterative methods represent a powerful class of algorithms that tackle mathematical problems by continuously refining an approximate solution. These methods shine in addressing large-scale equations, optimization challenges, and differential equations, particularly when direct solutions may be impractical. Starting with an initial guess, the iterative process steadily enhances the solution, with each refinement drawing nearer to the exact solution under the right conditions. By demanding less memory and computational power than direct methods, iterative methods reveal their strength in tackling substantial challenges. One of the most celebrated method among them is Newton's method, a prime example of this transformative approach. It is given by (Atkinson, 2018)

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

for solving a nonlinear equation

$$f(x) = 0 \tag{2}$$

This method begins with a fitting initial guess, paving the way for discovery. The order of convergence is a powerful concept in numerical analysis that illustrates the remarkable speed at which an iterative method draws closer to the exact solution of a problem. It embodies the journey of reducing error in successive iterations of an algorithm, showcasing the potential for precision and innovation. The definition of order of convergence is as follows.

Definition (Burden & Faires, 2011)

Suppose (β_n) is a sequence known to converge to zero, and (α_n) converges to a number α. If a positive constant K exists such that $|\alpha_n - \alpha| \leq K|\beta_n|$, for large n. Then we say that (α_n) converges to α with order of convergence $O(\beta_n)$.

The order of convergence of the method defined in equation (1) is two.

The wavelet-based quadrature rule (Aziz et al., 2011) is laid out as follows:

$$\int_a^b f\left(x\right)dx \cong \frac{b-a}{n}\sum_{i=1}^{n} f\left(a + \frac{(b-a)(2i-1)}{2n}\right)$$

In our study, we have taken n = 2. Under this assumption,

**Corresponding Author's Email:dkb_math@ravenshawuniversity.ac.in*
https://orcid.org/0000-0001-6125-5224

This article is published under the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the above rule reduces to the following quadrature rule

$$I\left(f\right) = \int_a^b f\left(x\right)dx \cong \frac{b-a}{2}\left[f\left(\frac{3a+b}{4}\right) + f\left(\frac{a+3b}{4}\right)\right] \quad (3)$$

In our study, Taylor's series (Ma, 2023) is used as a cornerstone of our paper. The anti-Gaussian quadrature rule (Laurie, 1996) has been used to uncover the complement of this innovative rule (3). The anti-Gaussian formula is a remarkable quadrature tool that integrates all polynomials up to degree (2n + 1) and delivers an error that is equal in magnitude yet opposite in sign to that of the 'n'-point Gaussian formula.

If

$Q_{aG_{n+1}}(f) = \sum_{i=1}^{n+1} \lambda_i f(x_i)$, where λ_i's are constant parameters.

be (n + 1) point anti Gaussian quadrature formula and $Q(G_n)(f)$ the n point Gaussian formula, then

$$I(f) - Q_{aG_{n+1}}(f) = -(I(f) - Q_{G_n}(f)) \quad (4)$$

where f is a polynomial of degree ≤2n+1 and I(f) is the exact value of the integral.

In pursuit of knowledge, researchers and scientists have elevated Newton's method by enriching its order of convergence through innovative techniques. For that purpose, anti-Gaussian quadrature rules are implemented to enhance Newton's method and its order of convergence. Through the application of a wavelet-based quadrature rule, the capabilities of Newton's method have been improved.

In this paper, an anti-wavelet-based quadrature rule has been developed in light of the anti-Gaussian technique as described above is implemented to create the final iterative method. Various numerical tests are conducted to inspire confidence in the performance of the iterative procedure, highlighting the number of iterations, absolute error, computation of the order of convergence (COC), and runtime. Several iterations have been performed using both the new method and other existing methods. In conducting these iterations, a good initial guess is selected that is close to the actual solution; otherwise, the method converges slowly or fails to converge. This choice is crucial for the numerical computation of the iterative method, as it significantly impacts its efficiency. The runtime of the iterative method is defined as the time taken to achieve convergence. This measure is essential for comparing different iterative methods and identifying the most effective ones. Additionally, the error in an iterative method quantifies the difference between the approximate solution obtained at a given iteration and the actual solution.

Definition (Computation of order of convergence, COC) (Sánchez et al., 2010)

The computation order of convergence (COC) of a sequence (x_n) is defined by

$$\tilde{\rho}_n = \frac{\ln\left|\frac{e_{n+1}}{e_n}\right|}{\ln\left|\frac{e_n}{e_{n-1}}\right|}$$

Where $x_{(n-1)}$, x_n and $x_{(n+1)}$ are three iterations near the root α, and $e_n = x_n - \alpha$.

In this work, a third-order iterative method is developed using an anti-Gaussian Haar wavelet-based quadrature rule. A new three-point quadrature rule is derived using the anti-Gaussian technique. This new quadrature rule is used to improve the order of convergence of Newton-Raphson methods. As a result, a new two-step third-order iterative method has been obtained. Derivation of the order of convergence of the newly developed iterative method has been completed using Taylor's Series. Objective of the work is to find solutions of higher-degree nonlinear equations with less number of iterations and small amount of runtime as compared to Newton method. For the computation of the number of iterations, COC, approximate root, and runtime of the newly constructed iterative method, a suitable algorithm is provided. After computation, it is observed that the newly developed iterative method is more efficient as compared to others.

In recent decades, quadrature rules have become a powerful tool for deriving high-order methods. The utility of quadrature rules in high-precision computing has been explored (Khattri, 2012), while novel quadrature-based techniques grounded in classical rules such as Bernstein, trapezoidal, and Simpson formulas have been discovered (Sehrish, 2020; Saeed & Hassan, 2021; Twum & Remembrance, 2023). More recently, quadrature-inspired algorithms for nonlinear equations, enhancing stability and addressing cases with unknown multiplicity, have been developed by researchers (Singh & Behera, 2022; Shah & Waseem, 2024; Singh et al., 2025). Alongside quadrature-based approaches, researchers have advanced multipoint and derivative-free schemes. Three-step methods for nonlinear systems have been analyzed (Dehghan & Shirilord, 2020), while modifications of fixed-point schemes tailored for engineering applications have been introduced (Shams et al., 2022; Sana et al., 2022). A fifth-order method eliminating second derivatives has been proposed (Hassan et al., 2022). An optimal fourth-order class for nonlinear systems has been constructed (Cordero et al., 2024). Collectively, these contributions highlight a strong trend toward designing iterative algorithms that balance efficiency, accuracy, and applicability across mathematical and engineering contexts.

This paper has six sections, starting with an inspiring introduction. The second section delves into the mathematical formulation, featuring two enlightening subsections: the creation of the Anti-Gaussian quadrature rule and the development of the iterative method. The third section offers a profound analysis of convergence. In the fourth section, all necessary numerical calculations are carried out, leading us to the fifth section where results and meaningful discussions are presented. Finally, the

last section draws everything together with a powerful conclusion.

MATERIALS AND METHODS

Formation of anti-Wavelet Based Quadrature Rule

In light of Equation (3), we have,

$$I(f) = \int_{-1}^{1} f(x)\ dx \cong \left[f\left(-\frac{1}{2}\right) + f\left(\frac{1}{2}\right)\right] = Q_W(f) \quad (5)$$

The above quadrature rule $Q_w(f)$ is a two-point rule. Now the anti-Gaussian technique defined in Equation (4) is implemented to the quadrature rule $Q_w(f)$ defined in Equation (5). The goal is to construct a three-point anti-Gaussian wavelet-based quadrature rule $Q_{aw}(f,)$ which is defined as follows

$$I(f) - Q_{aW}(f) = -(I(f) - Q_W(f))$$
$$\Rightarrow Q_{aW}(f) = 2I(f) - Q_W(f) \quad (6)$$

Where $Q_{aW}(f) = \sum_{i=1}^{3} \lambda_i f(x_i)$

Here f is a function of degree less than or equal to 5. λ_i's are to be determined later.

Here $I(f) = \int_{-1}^{1} f(x)\ dx, \quad x_1 = -1, \ x_3 = 1$

Let f (x) = 1

From Equation (6) , it follows that

$$\lambda_1 f(-1) + \lambda_2 f(x_2) + \lambda_3 f(x_3)$$
$$= 2\int_{-1}^{1} f(x)\ dx - \left[f\left(-\frac{1}{2}\right) + f\left(\frac{1}{2}\right)\right]$$
$$\Rightarrow \lambda_1 f(-1) + \lambda_2 f(x_2) + \lambda_3 f(1) = 2\int_{-1}^{1} 1\ dx - 2 \quad (7)$$
$$\Rightarrow \lambda_1 + \lambda_2 + \lambda_3 = 2$$

Assuming f (x) = 1

Using Equation (6), it yields

$$\lambda_1 f(-1) + \lambda_2 f(x_2) + \lambda_3 f(1)$$
$$= 2\int_{-1}^{1} x\ dx - \left[f\left(-\frac{1}{2}\right) + f\left(\frac{1}{2}\right)\right]$$
$$\Rightarrow -\lambda_1 + \lambda_2 x_2 + \lambda_3 = 0 \quad (8)$$

Taking f (x) = x^2

By virtue of Equation (6),

$$\lambda_1 f(-1) + \lambda_2 f(x_2) + \lambda_3 f(1)$$
$$= 2\int_{-1}^{1} x^2\ dx - \left[f\left(-\frac{1}{2}\right) + f\left(\frac{1}{2}\right)\right]$$
$$\Rightarrow \lambda_1 + \lambda_2 {x_2}^2 + \lambda_3 = \frac{5}{6} \quad (9)$$

Assuming f (x) = x^3

$$\lambda_1 f(-1) + \lambda_2 f(x_2) + \lambda_3 f(1)$$
$$= 2\int_{-1}^{1} x^2\ dx - \left[f\left(-\frac{1}{2}\right) + f\left(\frac{1}{2}\right)\right]$$
$$\Rightarrow \lambda_1 + \lambda_2 {x_2}^2 + \lambda_3 = \frac{5}{6} \quad (10)$$

Adding Equation (7) with (8) results

$$\lambda_2(1 + x_2) + 2\lambda_3 = 2 \quad (11)$$

Again adding Equation (8) with (9) yields

$$\lambda_2\left(x_2 + {x_2}^2\right) + 2\lambda_3 = \frac{5}{6} \quad (12)$$

Adding Equation (9) and (10) gives

$$\lambda_2\left({x_2}^2 + {x_2}^3\right) + 2\lambda_3 = \frac{5}{6} \quad (13)$$

Subtracting Equation (12) from Equation (11) , it follows that

$$\lambda_2\left(1 - {x_2}^2\right) = \frac{7}{6} \quad (14)$$

Subtracting Equation (13) from Equation (12) results

$$\lambda_2\left(x_2 - {x_2}^3\right) = 0$$
$$\Rightarrow \quad \lambda_2 = 0 \ \text{ or } \ x_2 - {x_2}^3 = 0$$

$\lambda_2 = 0$ violates the existence of Equation (14). Therefore, it is unaccepted.

Hence $x_2 - {x_2}^3 = 0$
$\Rightarrow x_2 = 0, \ +1, \ -1$

x_2 = +1, -1 violates the existence of Equation (14). Therefore, it is impossible to accept these values.

Therefore $\lambda_2 = \frac{7}{6}$. (from Equation (14))

From Equation (11)

$$\lambda_3 = \frac{5}{12}$$

From Equation (7)

$$\lambda_1 = \frac{5}{12}$$

Substituting values of λ_1, λ_2, λ_3, x_2 in Equation (6), it follows that

$$Q_{aW}(f) = \sum_{i=1}^{3} \lambda_i f(x_i)$$
$$\Rightarrow Q_{aW}(f) = \frac{5}{12}f(-1) + \frac{7}{6}f(0) + \frac{5}{12}f(1)$$

Therefore newly developed three point quadrature rule is

(15)

$$\int_{-1}^{1} f(x)\ dx \cong \frac{5}{12}f(-1) + \frac{7}{6}f(0) + \frac{5}{12}f(1) = Q_{aW}(f)$$

Formulation of Iterative Method Based on $Q_{aW}(f)$

We know that

$$f(x) = f(x_n) + \int_{x_n}^{x} f'(t)\ dt \quad (16)$$

$$\Rightarrow 0 = f(x_n) + \int_{x_n}^{x} f'(t)\ dt$$

Using the transformation $v = \frac{2t-(x+x_n)}{x-x_n}$, the integral present on the right hand side of above equation is converted as follows

$$\int_{x_n}^{x} f'(t)\ dt = \left(\frac{x-x_n}{2}\right)\int_{-1}^{1} f\left(\frac{v(x-x_n)+(x+x_n)}{2}\right)\ dv$$

Using the above expression in Equation (16)

$$0 = f(x_n) + \left(\frac{x-x_n}{2}\right)\int_{-1}^{1} f\left(\frac{v(x-x_n)+(x+x_n)}{2}\right)\ dv$$

$$\Rightarrow 0 = f(x_n) + \left(\frac{x-x_n}{2}\right)\left[\frac{5}{12}f'(x_n) + \frac{7}{6}f'\left(\frac{(x+x_n)}{2}\right) + \frac{5}{12}f'(x)\right]$$

(by using Equation 15)

$$\Rightarrow x = x_n - \frac{2f(x_n)}{\left[\frac{5}{12}f'(x_n) + \frac{7}{6}f'\left(\frac{(x+x_n)}{2}\right) + \frac{5}{12}f'(x)\right]}$$

$$\Rightarrow x = x_n - \frac{24f(x_n)}{\left[5f'(x_n) + 14f'\left(\frac{(x+x_n)}{2}\right) + 5f'(x)\right]}$$

Fixed point form of the above equation can be written as

$$\Rightarrow x_{n+1} = x_n - \frac{24f(x_n)}{\left[5f'(x_n) + 14f'\left(\frac{(x_{n+1}{}^{*}+x_n)}{2}\right) + 5f'(x_{n+1}{}^{*})\right]}$$

Replacing $y_n = x_n - \frac{f(x_n)}{f'(x_n)}$ in place of $x_{n+1}{}^{*}$ in right hand side of above expression we have the following algorithm.

Algorithm:

$$x_{n+1} = x_n - \frac{24f(x_n)}{H(x_n, y_n)} \quad (17)$$

Where $H(x_n, y_n) =$

$$\left[5f'(x_n) + 14f'\left(\frac{(y_n+x_n)}{2}\right) + 5f'(y_n)\right]$$

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

The iterative method defined in Equation (17) is the final iterative method.

Convergence Analysis

In this section, the rate of convergence of the new two-step method is presented in terms of a theorem.

Theorem:

Assuming that the function $f:I \subset R \to R$ for an open interval I possesses a simple root $\delta \in I$, and given that it is a sufficiently differentiable function in close proximity, it can be inferred that the iterative method defined in Equation (17) exhibits third-order convergence.

$$e_{n+1} = {e_n}^3\left({c_2}^2 + \frac{1}{16}c_3\right) + O\left({e_n}^4\right)$$

Where $c_j = \frac{f^j(\mu)}{j!\ .\ f'(\mu)}$ and $e_j = x_j - \delta, j = 1,2,3,\ldots.$

Proof:

Applying Taylor's series [4] to the function $f(x_n)$ about δ with the help of the expression $c_j = \frac{f^j(\mu)}{j!\ .\ f'(\mu)}$ and $e_j = x_j - \delta$ we have

$$f(x_n) = f'(\delta)\left[e_n + c_2{e_n}^2 + c_3{e_n}^3 + c_4{e_n}^4 + O\left({e_n}^5\right)\right] \quad (18)$$

$$f'(x_n) = f'(\delta)\left[1 + 2c_2e_n + 3c_3{e_n}^2 + 4c_4{e_n}^3 + O\left({e_n}^4\right)\right] \quad (19)$$

From [7]

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)} = \delta + c_2{e_n}^2 + 2\left(c_3 - {c_2}^2\right){e_n}^3 + O\left({e_n}^4\right) \quad (20)$$

$$\text{let}\quad h_n = \frac{(y_n+x_n)}{2} = \delta + \frac{e_n}{2} + \frac{c_2}{2}{e_n}^2 + \left(c_3 - {c_2}^2\right){e_n}^3 + O\left({e_n}^4\right) \quad (21)$$

In light of Equation (19) , it can be written as follows

$$f'(h_n) = f'(\delta)$$

$$\left[1 + 2c_2(h_n-\delta) + 3c_3(h_n-\delta)^2 + 4c_4(h_n-\delta)^3 + O((h_n-\delta)^4)\right]$$

From Equation (21) , it follows that

$$h_n - \delta = \frac{e_n}{2} + \frac{c_2}{2}{e_n}^2 + \left(c_3 - {c_2}^2\right){e_n}^3 + O\left({e_n}^4\right)$$

$$\Rightarrow (h_n-\delta)^2 = \frac{1}{4}{e_n}^2 + O\left({e_n}^3\right)$$

Therefore $f'(h_n) = f'(\delta)\left[1 + c_2e_n + \left({c_2}^2 + \frac{3c_3}{4}\right){e_n}^2 + O\left({e_n}^3\right)\right]$

$$H(x_n, y_n) = 5f'(x_n) + 14f'(h_n) + 5f'(y_n)$$

$$= f'(\delta)\left[24 + 24c_2e_n + {e_n}^2\left(\frac{51}{2}c_3 + 24{c_2}^2\right) + O\left({e_n}^3\right)\right]$$

$$\frac{24f(x_n)}{H(x_n,y_n)} = \frac{\left[e_n + c_2e_n^{\,2} + c_3e_n^{\,3} + c_4e_n^{\,4} + O(e_n^{\,5})\right]}{\left[1 + c_2e_n + e_n^{\,2}\left(\frac{51}{48}c_3 + c_2^{\,2}\right) + O(e_n^{\,3})\right]}$$

$$= \left[e_n + c_2e_n^{\,2} + c_3e_n^{\,3} + c_4e_n^{\,4} + O(e_n^{\,5})\right]$$

$$\times\left[1 + c_2e_n + e_n^{\,2}\left(\frac{51}{48}c_3 + c_2^{\,2}\right) + O(e_n^{\,3})\right]^{-1}$$

$$= e_n - e_n^{\,3}\left(\frac{1}{16}c_3 + c_2^{\,2}\right) + O(e_n^{\,4})$$

$$x_{n+1} = x_n - \frac{24f(x_n)}{H(x_n,y_n)}$$

$$= \delta + e_n^{\,3}\left(\frac{1}{16}c_3 + c_2^{\,2}\right) + O(e_n^{\,4})$$

This proves the above theorem.

RESULTS AND DISCUSSION

In this section, the number of iterations, runtime, absolute error, and COC of the newly developed algorithm (AGNM) and Newton method (NM) are computed with the help of Python Software. The tolerance is set as $\in = 10^{-6}$. The numerical algorithm of the iterative method AGNM is written as follows.

Algorithm

Input:

- f (x) : function whose root is sought
- f'(x) : derivative of f (x)
- x_0 : starting value for the iteration
- $\in$: set desired accuracy
- N_{max}: set maximum iterations

Steps:

- Initialize:
  - Set n = 0 (iteration counter)
  - Set $x_n = x_0$
- Iterative Process:
  - Repeat while fxn > $\in$ and < Nmax
  - compute y_n :

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

  - compute H (x_n, y_n) :

$$H(x_n,y_n) = \left[5f'(x_n) + 14f'\left(\frac{(y_n + x_n)}{2}\right) + 5f'(y_n)\right]$$

- check for division by zero:

If H (x_n, y_n) = 0, stop the algorithm and return an error message.

- compute next approximation x_{n+1}:

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

- update for next iteration:

* Compute absolute error:

$$\text{Absolute} \quad \text{Error} = |x_{n+1} - x_n|$$

* Set $x_n = x_{n+1}$

* Increment the counter:n = n + 1

- Check convergence:

- If If |f (xn)| ≤ $\in$:

- Compute the Convergence Order Constant (COC):

$$\text{COC} = \frac{|x_{n+1} - x_n|}{|x_n - x_{n-1}|^p}$$

$$\text{where} \quad p = 3$$

- Record end time and calculate the run time.

Output:

- Root x_n (if convergent).
- Number of iteration n.
- Absolute error at each iteration.
- COC
- Run time of algorithm
- A failure message if method does not converge

Flow of Method

- Start with x_0.
- Compute y_n.
- Use y_n to evaluate (x_n, y_n).
- Update x_n using the new iterative formula.

• Compute absolute error and COC.

• Repeat until the root is found within the desired tolerance or the maximum iterations are exceeded.

The following examples are taken as test functions with x_0 as their initial guess.

1. $f_1(x) = \cos x - x, x_0 = -0.3$
2. $f_2(x) = x^3 + 4x^2 - 10, x_0 = -0.5$
3. $f_3(x) = e^{(x^2+7x-30)} - 1, x_0 = 3.25$
4. $f_4(x) = x^3 - 10, x_0 = 1.5$

Numerical computations of these methods are reflected in table 1, table 2, table3, table 4 and table 5.

In this section, we have computed the number of iterations, absolute error, computational order of convergence (COC), runtime, and approximate roots of AGNM along with methods like (Shams et al., 2022; Sehrish et al., 2020; Twum et al., 2023). It has been observed that AGNM requires a smaller number of iterations, produces a lower absolute error, and requires less computational time than the other methods for solving nonlinear equations. The order of convergence of all the methods is also evaluated. However, AGNM provides the solution in fewer iterations with a smaller absolute error and lower runtime. These results demonstrate that the newly developed iterative method AGNM is computationally more efficient than (Shams et al., 2022; Sehrish et al., 2020; Twum et al., 2023) methods.

Table 1: Computation of number of iterations, runtime, absolute error, approximate root and COC of AGNM

Functions	Initial value	Number of iterations (AGNM)	Absolute error	COC	Runtime (seconds)	Approximate Root
f_1 (X)	-0.3	3	$2.57371901568 \times 10^{-12}$	3.3328743	0.000088	0.738698675247305
f_2 (X)	-0.5	18	$2.5102364631379714 \times 10^{-11}$	2.97493991	0.000072	1.3656985826950796
f_3 (X)	3.25	6	0	2.96604076	0.000175	3.0000000000000000
f_4 (X)	1.5	3	$4.0953906932372774 \times 10^{-12}$	3.332874	0.000023	2.1544346900237157

Table 2: Computation of number of iterations, runtime, absolute error, approximate root, and COC of NM

Functions	Initial value	Number of iterations (Newton)	Absolute error	COC	Runtime (seconds)	Approximate Root
f_1 (X)	-0.3	4	$1.7291612364189746 \times 10^{-7}$	1.674466272	0.000098	0.7395092566938682
f_2 (X)	-0.5	115	$1.250418417697574 \times 10^{-7}$	1.994434390	0.000178	1.365735139060055
f_3 (X)	3.25	8	0	2.404474182	0.000192	3.000000000000000
f_4 (X)	1.5	5	$2.7407895242248514 \times 10^{-7}$	1.871631	0.000033	2.154434964110871

Table 3: Computation of number of iteration, runtime, absolute error,approximate root and COC (Twum et al., 2023)

Functions	Initial value	Number of iterations (Twum et al., 2023)	Absolute error	COC	Runtime (seconds)	Approximate Root
f_1 (X)	-0.3	4	2.894691×10^{-9}	2.9744662	0.000096	0.7395092566723658
f_2 (X)	-0.5	80	2.250928×10^{-8}	2.9944343	0.00094	1.365735139652798
f_3 (X)	3.25	7	0	2.9044741	0.000189	3.000000000000000
f_4 (X)	1.5	4	$4.5672485 \times 10^{-10}$	2.87163198	0.000039	2.15443496489017

Table 4: Computation of number of iteration, runtime, absolute error,approximate root and COC of (Shams et al., 2022)

Functions	Initial value	Number of iterations (Shams et al., 2022)	Absolute error	COC	Runtime (seconds)	Approximate Root
$f_1(X)$	-0.3	4	2.5980256×10^{-11}	7.9744662	0.000090	0.7395092566932798
$f_2(X)$	-0.5	43	2.7950265×10^{-9}	7.9944343	0.00065	1.3657351390604517
$f_3(X)$	3.25	7	0	7.9044741	0.000152	3.000000000000000
$f_4(X)$	1.5	4	4.59327812×10^{-10}	7.87163198	0.000022	2.1544349641167821

Table 5 : Computation of number of iteration, runtime, absolute error,approximate root and COC of (Sehrish et al., 2020)

Functions	Initial value	Number of iterations (Sehrish et al., 2020)	Absolute error	COC	Runtime (seconds)	Approximate Root
$f_1(X)$	-0.3	4	2.532861913×10^{-9}	2.674466272	0.000098	0.7395092566938682
$f_1(X)$	-0.5	49	2.59132985×10^{-7}	2.994434390	0.000178	1.365735139060055
$f_1(X)$	3.25	9	0	2.404474182	0.000192	3.000000000000000
$f_1(X)$	1.5	5	4.2375648×10^{-7}	2.871631	0.000033	2.154434964110871

CONCLUSIONS

Our work unveils an improved algorithm, AGNM, that surpasses Newton's method and other existing higher-order methods. In the convergence analysis, we reveal an impressive order of convergence for AGNM at 3, a testament to its superior performance. Section 4 showcases various examples that compare runtime, number of iterations, convergence order coefficient (COC), absolute error, and approximate root of both the newly developed method and other existing methods. This section includes a thorough analysis of the results, illuminating the remarkable advantages of our innovative algorithm over the traditional approach.

ACKNOWLEDGMENTS

We sincerely acknowledge Dr. Rajani Ballav Dash, retired Professor, Department of Mathematics, Ravenshaw University, Kataka for his support and guidance.

DECLARATION OF CONFLICT OF INTEREST

There is no conflict of interest.

FUNDING INFORMATION

The authors declare that they did not receive any specific grant or financial support from funding agencies in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTION

Conceptualization: **T.S. & D.K.B.;** Mathematical Formulation**: T.S., D.K.B. & R.B.D.;** Convergence analysis**: T.S., D.K.B., & R.K.S.;** Numerical Formulation**: T.S., D.K.B., R.B.D & R.K.S.;** Conclusion: **T.S. & D.K.B.**; Manuscript editing and reviewing: **D.K.B., R.B.D & R.K.S.**

DATA AVAILABILITY STATEMENT

The datasets generated and/or analyzed during the current study are not publicly available but may be obtained from the corresponding author upon reasonable request.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN

THE MANUSCRIPT PREPARATION PROCESS

The authors declare that no generative AI or AI-assisted technologies were used in the preparation, writing, editing or revision of this manuscript.

REFERENCES

Atkinson, K. E. (2018). *An Introduction to Numerical Analysis*. John Wiley & Sons. https://math.science.cmu.ac.th/docs/qNA2556/ref_na/Katkinson.pdf

Aziz, I., Siraj-Ul-Islam, & Khan, W. (2011). Quadrature rules for numerical integration based on Haar wavelets and hybrid functions. *Computers and Mathematics with Applications*, **61**, 2770–2781. https://doi.org/10.1016/j.camwa.2011.03.043

Burden, R. L., & Faires, J. D. (2011). *Numerical Analysis*. Cengage Learning. https://fac.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf

Cordero, A. A., Rojas-Heiciano, R. V., Torregrosa, J. R., & Vassileva, M. P. (2024). A highly efficient class of optimal fourth-order methods for solving nonlinear systems. *Numerical Algorithms*, **95**, 1879–1909. https://doi.org/10.21203/rs.3.rs-2775587/v1

Dehghan, M., & Shirilord, A. (2020). Three-step iterative methods for numerical solution of systems of nonlinear equations. *Numerical Functional Analysis and Optimization*, **42**, 834–848. https://doi.org/10.1007/s00366-020-01072-1

Hassan, A., Ali, N. Y., & Park, A. H. (2022). A new fifth-order iterative method free from second derivative for solving nonlinear equations. *Journal of Applied Mathematics and Computing*, **68**, 2877–2886. https://doi.org/10.1007/s12190-021-01647-1

Khattri, S. K. (2012). Quadrature-based optimal iterative methods with applications in high-precision computing. *Numerical Mathematics: Theory, Methods and Applications*, **5**(4), 592–601. https://global-sci.com/pdf/article/90678/quadrature-based-optimal-iterative-methods-with-applications-in-high-precision-computing.pdf

Laurie, D. P. (1996). Anti-Gaussian quadrature formulas. *Mathematics of Computation*, **65**(214), 739–747. https://doi.org/10.1090/S0025-5718-96-00713-2

Ma, L. (2023). The Taylor's theorem and its application. *Highlights in Science, Engineering and Technology*, **72**, 1368–1372. https://doi.org/10.54097/98knf072

Saeed, H. J., & Abdul-Hassan, N. Y. (2021). An efficient three-step iterative method based on Bernstein quadrature formula for solving nonlinear equations. *Basrah Journal of Science*, **39**(3), 329–338. https://basjsci.edu.iq/index.php/basjsci/en/article/view/115/97

Sana, G., Noor, M. A., Hassan, M. U., & Hammouch, Z. (2022). Some recent modifications of fixed point iterative schemes for computing zeros of nonlinear equations. *Complexity*, 1–17. https://doi.org/10.1155/2022/4331899

Sánchez, M. G., Noguera, M., & Gutiérrez, J. M. (2010). On some computational orders of convergence. *Applied Mathematics Letters*, **23**, 472–478. https://doi.org/10.1016/j.aml.2009.12.006

Sehrish, U., Mohammed, M. S., & Abdul, W. S. (2020). A new quadrature-based iterative method for scalar nonlinear equations. *Journal of Mechanics of Continua and Mathematical Sciences*, **15**(10), 79–93. https://doi.org/10.26782/jmcms.2020.10.00006

Shah, F. A., & Waseem, M. (2024). Quadrature-based innovative techniques concerning nonlinear equations having unknown multiplicity. *Examples and Counterexamples*, **6**, 1–7. https://doi.org/10.1016/j.exco.2024.100150

Shams, M., Kausar, N., Agarwal, P., & Oros, G. I. (2022). Efficient iterative scheme for solving nonlinear equations with engineering applications. *Applied Mathematics in Science and Engineering*, **30**, 708–735. https://doi.org/10.1080/27690911.2022.2130914

Singh, T., & Behera, D. K. (2022). A new approach of iterative methods for solving non-linear equations using Boole's quadrature. *Southeast Asian Journal of Mathematics and Mathematical Sciences*, **18**(1), 277–286. https://rsmams.org/journals/download.php?articleid=693

Singh, T., Behera, D. K., Ahmed, S. O., & Saeed, R. K. (2025). An iterative method based on the average quadrature formula. *Computational Algorithms and Numerical Dimensions*, **4**(1), 57–64. https://doi.org/10.22105/cand.2024.485552.1155

Twum, S. B., & Remembrance, Y. L. (2023). A new iterative method of solution of nonlinear equations derived from Simpson and trapezoidal quadrature rules. *International Journal of Innovation in Science and Mathematics*, **11**(1), 1–10. https://www.ijism.org/administrator/components/com_jresearch/files/publications/IJISM_985_FINAL.pdf