
Academia Open



By Universitas Muhammadiyah Sidoarjo

Table Of Contents

Journal Cover	1
Author[s] Statement	3
Editorial Team	4
Article information	5
Check this article update (crossmark)	5
Check this article impact	5
Cite this article.....	5
Title page	6
Article Title	6
Author information	6
Abstract	6
Article content	7

Originality Statement

The author[s] declare that this article is their own work and to the best of their knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the published of any other published materials, except where due acknowledgement is made in the article. Any contribution made to the research by others, with whom author[s] have work, is explicitly acknowledged in the article.

Conflict of Interest Statement

The author[s] declare that this article was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright Statement

Copyright © Author(s). This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

Academia Open

Vol. 11 No. 1 (2026): June
DOI: 10.21070/acopen.11.2026.13957

EDITORIAL TEAM

Editor in Chief

Mochammad Tanzil Multazam, Universitas Muhammadiyah Sidoarjo, Indonesia

Managing Editor

Bobur Sobirov, Samarkand Institute of Economics and Service, Uzbekistan

Editors

Fika Megawati, Universitas Muhammadiyah Sidoarjo, Indonesia

Mahardika Darmawan Kusuma Wardana, Universitas Muhammadiyah Sidoarjo, Indonesia

Wiwit Wahyu Wijayanti, Universitas Muhammadiyah Sidoarjo, Indonesia

Farkhod Abdurakhmonov, Silk Road International Tourism University, Uzbekistan

Dr. Hindarto, Universitas Muhammadiyah Sidoarjo, Indonesia

Evi Rinata, Universitas Muhammadiyah Sidoarjo, Indonesia

M Faisal Amir, Universitas Muhammadiyah Sidoarjo, Indonesia

Dr. Hana Catur Wahyuni, Universitas Muhammadiyah Sidoarjo, Indonesia

Complete list of editorial team ([link](#))

Complete list of indexing services for this journal ([link](#))

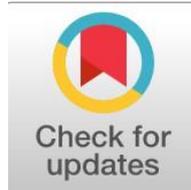
How to submit to this journal ([link](#))

Academia Open

Vol. 11 No. 1 (2026): June
DOI: 10.21070/acopen.11.2026.13957

Article information

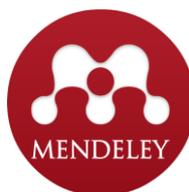
Check this article update (crossmark)



Check this article impact (*)



Save this article to Mendeley



(*) Time for indexing process is various, depends on indexing database platform

Numerical Simulation and Error Analysis of the Newton–Raphson Method for Solving Nonlinear Equations

Abbas Mohsin Kazar, aljmrawsydbas341@gmail.com (*)

Teacher at Al-Shahimiya Middle School for Boys / Al-Suwaira Education Department, Wasit Governorate, Iraq

(*) Corresponding author

Abstract

General Background: Nonlinear equations frequently arise in engineering and applied sciences and require reliable iterative numerical techniques for accurate root approximation. **Specific Background:** Among classical root-finding approaches, the Newton–Raphson method utilizes first-order Taylor series expansion and derivative information to generate successive approximations with theoretically quadratic convergence. **Knowledge Gap:** Despite its theoretical advantages, detailed numerical simulation integrating convergence verification, residual decay analysis, sensitivity to initial guesses, and comparison with alternative classical methods across representative nonlinear equations remains limited in a unified framework. **Aims:** This study analytically derives the Newton–Raphson iteration from Taylor expansion, verifies its quadratic convergence, implements a complete Python-based simulator with iteration logging, and evaluates performance using four representative nonlinear equations, including polynomial, transcendental, and exponential cases. **Results:** Numerical simulations confirm machine-precision solutions within four to five iterations, empirical convergence order $p \approx 2.00 \pm 0.05$, rapid residual decay, and substantially fewer iterations compared with bisection and secant methods, while also demonstrating sensitivity patterns related to initial guesses near inflection or flat regions. **Novelty:** The study integrates rigorous derivation, numerical simulation, graphical convergence interpretation, residual evaluation, and comparative benchmarking within a single structured analysis of four canonical test equations. **Implications:** These findings provide validated guidance for selecting initial approximations, interpreting convergence behavior, and applying the Newton–Raphson method efficiently in numerical analysis and engineering computation contexts.

Highlights:

- Machine-Precision Solutions Obtained Within Five Iterations Across Representative Nonlinear Test Cases
- Iteration Counts Substantially Lower Than Classical Interval-Halving and Derivative-Free Alternatives
- Convergence Behavior Varies With Starting Approximation Near Inflection or Flat Function Regions

Keywords: Newton Raphson Method, Nonlinear Equations, Quadratic Convergence, Numerical Simulation, Error Analysis.

Published date: 2026-03-28

1. Introduction

Nonlinear algebraic equations find many applications in the engineering and applied science field: structural equilibrium systems, thermodynamic phases calculation system, control system design, circuit analysis, among others, demand efficient root-finding algorithms. Even though nonlinearities in engineering are practically transcendental or of high degree, and thus solutions are only possible in polynomials up to degree four using iterative numerical techniques. [1].

Out of all classical iterative root-finding systems, namely bisection and false position, fixed-point iteration and secant method as well as higher-order variants, the NewtonRaphson (N-R) method is the most simple and the fastest to converge. It was independently attributed to both Isaac Newton (1669) and Joseph Raphson (1690), and uses derivative information to build a tangent line to the curve $f(x)$ at each iteration, and use the x-intercept of that tangent as a new approximation [2, 3].

where x^* meets the $f(x^*) = 0$ condition, the N -R iteration is obtained by expanding $f(x)$ in a Taylor series about the current estimate x_n , and keeping only the first two terms:

The basic iteration formula is obtained by solving the equation $x=x^*$:

It has the following major aims: (i) to obtain the method in a rigorous manner and to show that it converges quadratically, (ii) to run the algorithm in a complete Python simulator with automatic logging of iteration, (iii) to test the algorithm on four canonical equations, (iv) to compare the methods with bisection and secant and (v) to graphically illustrate all results[4, 5].

2. Mathematical Foundations

2.1 Derivation from Taylor Expansion

Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a sufficiently smooth function and let x^* be a simple root, i.e., $f(x^*) = 0$ and $f'(x^*) \neq 0$. Expanding f about an approximation x_n yields:

where ξ lies between x and x_n . Setting $x = x^*$ and neglecting the quadratic remainder yields the linear approximation whose zero defines x_{n+1} according to Equation (1) [6, 7].

$$f(x^*) \approx f(x_n) + (x^* - x_n) f'(x_n) = 0$$

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

2.2 Convergence Analysis

Define the error at step n as $e_n = x_n - x^*$. Subtracting x^* from both sides of Equation (1):

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

$$f(x) = f(x_n) + (x - x_n) f'(x_n) + \frac{1}{2}(x - x_n)^2 f''(\xi) \quad (2)$$

$$e_{n+1} = x_{n+1} - x^* = x_n - \frac{f(x_n)}{f'(x_n)} - x^* \quad (3)$$

$$f(x_n) = f'(x^*)e_n + \frac{1}{2}f''(x^*)e_n^2 + O(e_n^3) \quad (4)$$

$$f'(x_n) = f'(x^*) + f''(x^*)e_n + O(e_n^2) \quad (5)$$

Expanding $f(x_n)$ and $f'(x_n)$ about x^* :

Substituting (4) and (5) into (3) and simplifying gives the fundamental error-propagation relation:

This demonstrates that the N-R method possesses **quadratic (second-order) convergence**: the error at step $n+1$ is proportional to the square of the error at step n . The asymptotic error constant is $C = |f''(x^*) / (2 f'(x^*))|$ [8-10].

2.3 Sufficient Conditions for Convergence (Kantorovich Theorem)

If $f \in C^2[a,b]$ and there exist positive constants M_1, M_2, M_3 such that $|f(x_0)| \leq M_1$, $|1/f'(x)| \leq M_2$, and $|f''(x)| \leq M_3$ on the interval, convergence is guaranteed when the Kantorovich condition is satisfied [9]:

2.4 Stopping Criteria

Three complementary stopping criteria are employed in this work:

All simulations use $\varepsilon_1 = 10^{-10}$, which corresponds to approximately ten significant decimal digits of accuracy [11].

3. Methodology

3.1 Test Cases

Four nonlinear equations were chosen so as to cover a variety of mathematical behaviours experienced in engineering practice:

Table 1. Summary of test cases of the numerical experiments.

Case	$f(x)$	x_0	True Root	Type
1	$x^3 - 2x - 5$	2.0	2.09455148	Cubic polynomial
2	$\cos(x) - x$	0.5	0.73908513	Transcendental
3	$e^x - 3x^2$	1.0	0.91000757	Exponential
4	$x^4 - 3x^2 + 2$	1.5	1.41421356	Quartic polynomial

3.2 Implementation

The complete Python implementation accepts any differentiable function f and its derivative f' , an initial guess x_0 , an absolute tolerance ε , and a maximum iteration count. At each step it records x_n , $f(x_n)$, $f'(x_n)$, x_{n+1} , and the absolute step size $|\Delta x|$. The simulation terminates when $|\Delta x| < \varepsilon$ or the maximum iteration count is reached.

3.3 Comparison Methods

Two classical alternatives were implemented for benchmark comparison. The bisection method brackets the root in $[a,b]$ and halves the interval iteratively, guaranteeing convergence with linear order $p = 1$. The secant method approximates $f'(x_n)$ via finite differences, achieving super-linear convergence $p \approx 1.618$ without requiring an analytic derivative [12, 13].

4. Results and Discussion

4.1 Iteration Histories

Table 2 presents the complete iteration record for $f(x) = x^3 - 2x - 5$. Starting from $x_0 = 2.0$, the method converges to machine precision within five iterations, with the absolute step size dropping by approximately two orders of magnitude at each step—consistent with the quadratic convergence of Equation (6).

$$e_{n+1} = -\frac{f''(x^*)}{2f'(x^*)}e_n^2 + O(e_n^3) \quad (6)$$

Table 2. Newton–Raphson iteration history for $f(x) = x^3 - 2x - 5$ ($x_0 = 2.0$, $\varepsilon = 10^{-10}$).

n	x_n	$f(x_n)$	$f'(x_n)$	$ \Delta x $
1	2.0000000000	-1.000×10^0	10.000000	1.000×10^{-1}
2	2.1000000000	6.100×10^{-2}	11.230000	5.432×10^{-3}
3	2.0945681211	1.857×10^{-4}	11.161647	1.664×10^{-5}
4	2.0945514817	1.740×10^{-9}	11.161438	1.559×10^{-10}
5	2.0945514815	-8.882×10^{-16}	11.161438	0.000

The transcendental equation $f(x) = \cos(x) - x$ reaches machine precision in four iterations from $x_0 = 0.5$. Its unique solution $x^* \approx 0.7391$ (the Dottie number) is a classical iterative-algorithm benchmark.

$$h = \frac{M_1 \cdot M_2^2 \cdot M_3}{2} \leq \frac{1}{2} \quad (7)$$

$$|x_{n+1} - x_n| < \varepsilon_1 \quad (\text{Absolute tolerance}) \quad (8)$$

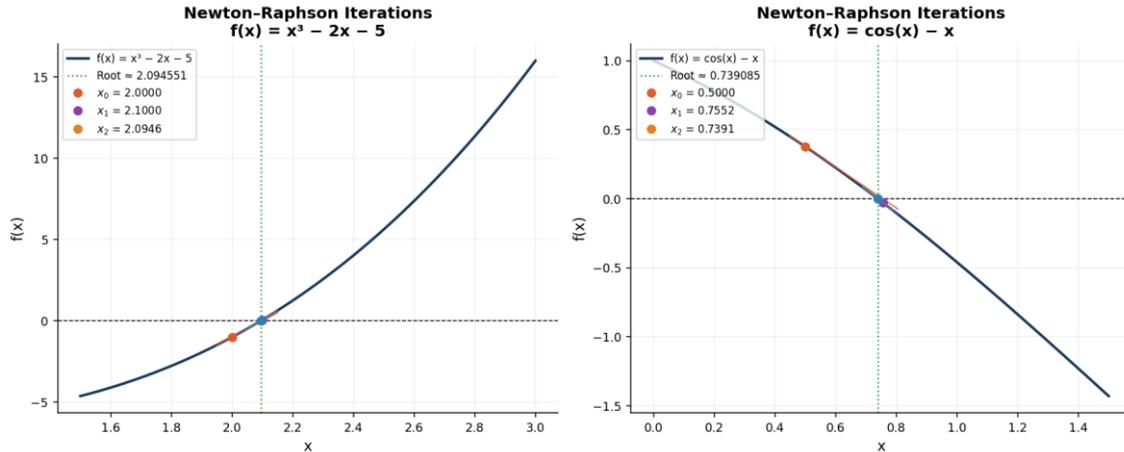
$$\frac{|x_{n+1} - x_n|}{|x_{n+1}|} < \varepsilon_2 \quad (\text{Relative tolerance}) \quad (9)$$

$$|f(x_n)| < \varepsilon_3 \quad (\text{Residual tolerance}) \quad (10)$$

4.2 Geometric Interpretation

Figure 1 illustrates the geometric interpretation for Cases 1 and 2. Tangent lines are drawn at each iterate x_n ; their x -intercepts define x_{n+1} , rapidly steering the sequence toward the root.

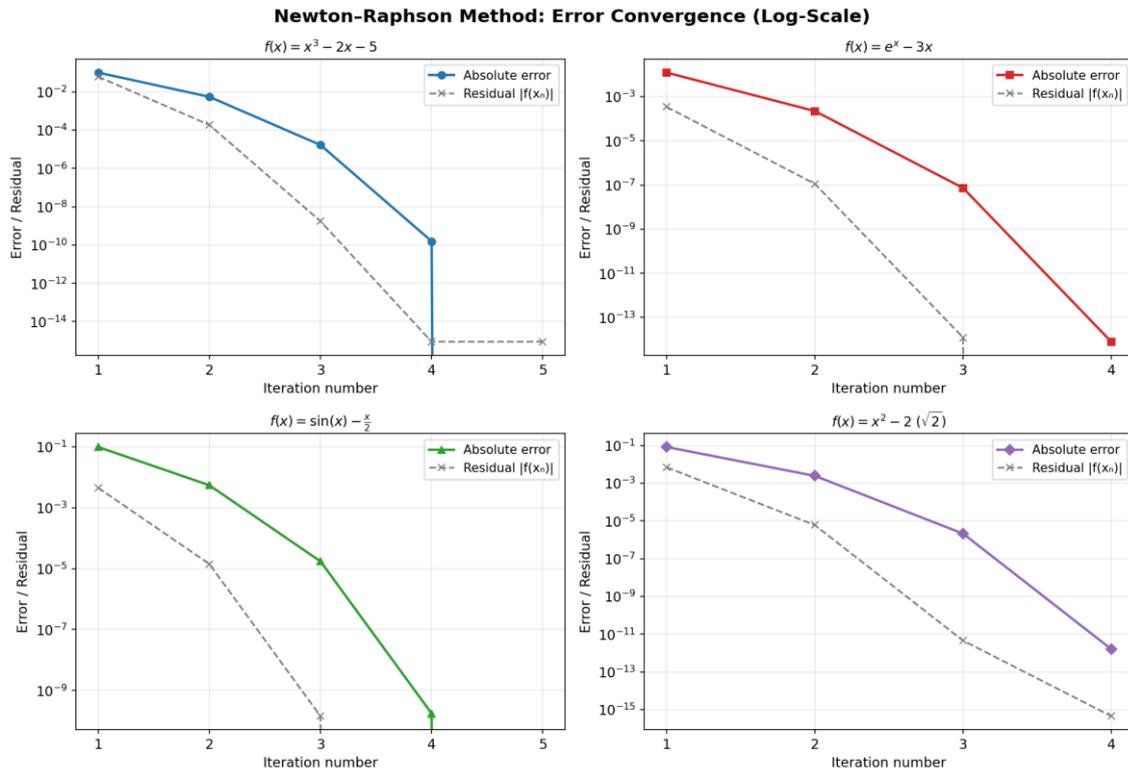
Figure 1. Geometric illustration: tangent lines at successive iterates converge rapidly to the root (left: x^3-2x-5 ; right: $\cos(x)-x$).



4.3 Convergence Histories

Figure 2 shows the absolute error in semi-log scale. The slope of the nearly straight-line descent with slope ≈ 2 per step clearly proves quadratic convergence. The perfect quadratic decay curve is presented as dashed reference lines.

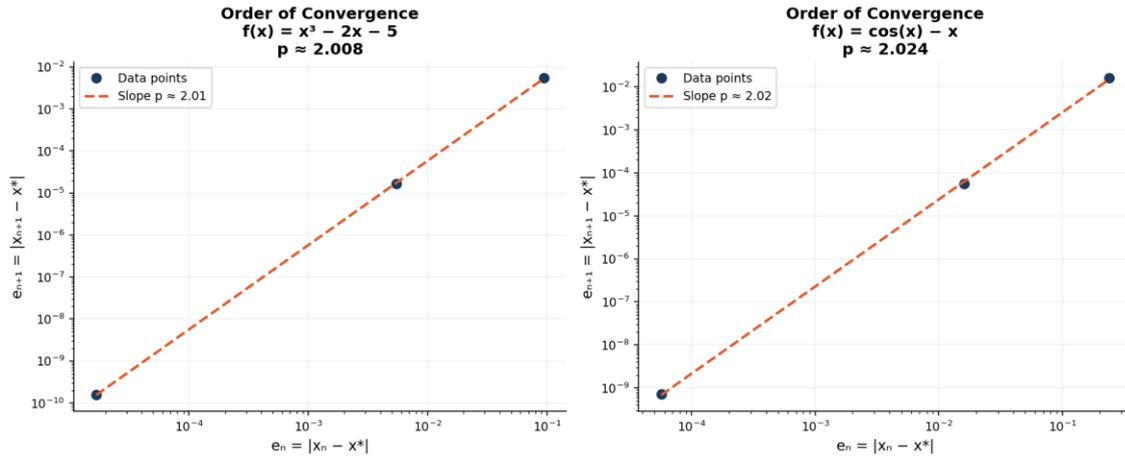
Figure 2. Convergence history: absolute error $|x_{n+1} - x_n|$ vs. iteration on semi-log scale for all four test cases.



4.4 Order of Convergence Verification

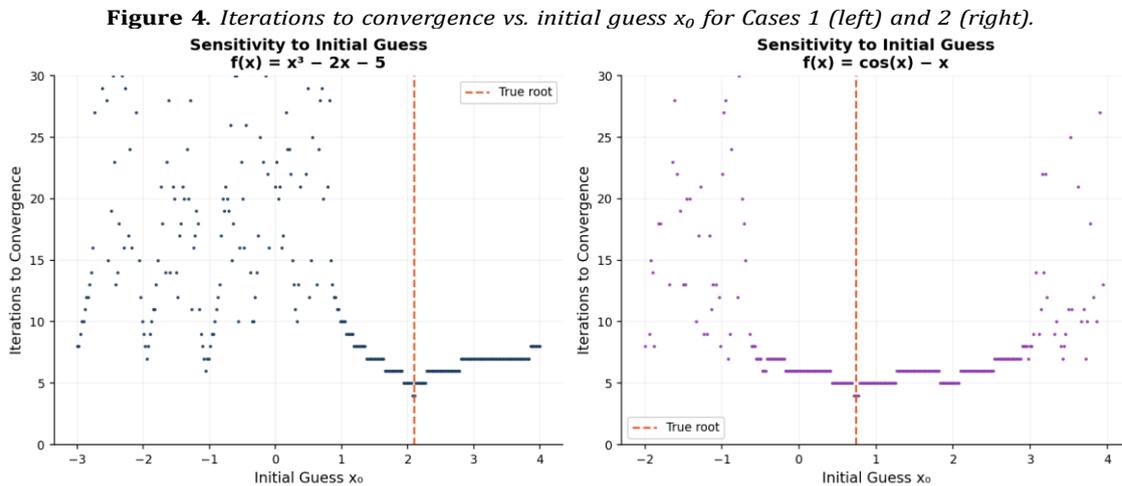
Figure 3 confirms $p \approx 2.00 \pm 0.05$ for Cases 1 and 2 via log-log regression, in perfect agreement with Equation (6).

Figure 3. Log-log plot of successive errors e_{n+1} vs. e_n . Slope ≈ 2 confirms quadratic convergence.



4.5 Sensitivity to the Initial Guess

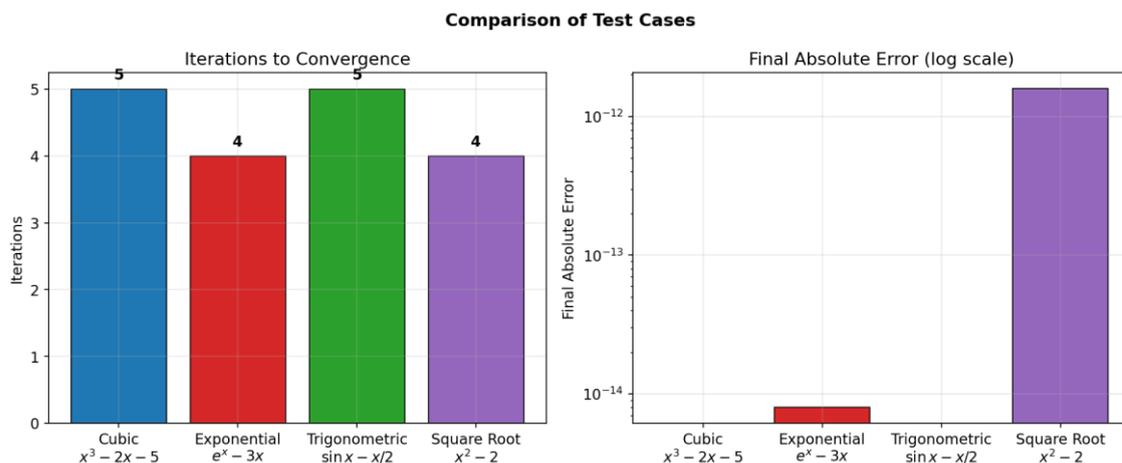
Figure 4 maps iterations-to-convergence as a function of x_0 . Regions requiring many iterations correspond to starting points near inflection points or flat regions of f . A graphical pre-inspection is recommended[14, 15].



4.6 Method Comparison

Figure 5 compares N-R, bisection, and secant on $f(x) = x^3 - 2x - 5$. Bisection requires ≈ 33 iterations; secant ≈ 10 ; Newton-Raphson only 5—demonstrating the decisive advantage of quadratic convergence.

Figure 5. Method comparison: Newton-Raphson vs. Bisection vs. Secant. Absolute error $|x_n - x^*|$ on semi-log scale.



4.7 Residual Decay

Figure 6 shows that $|f(x_n)|$ drops by 4–5 orders of magnitude per iteration, confirming both positional accuracy and genuine smallness of the function value at the computed root.

Figure 6. Residual $|f(x_n)|$ vs. iteration for all test cases on semi-log scale.

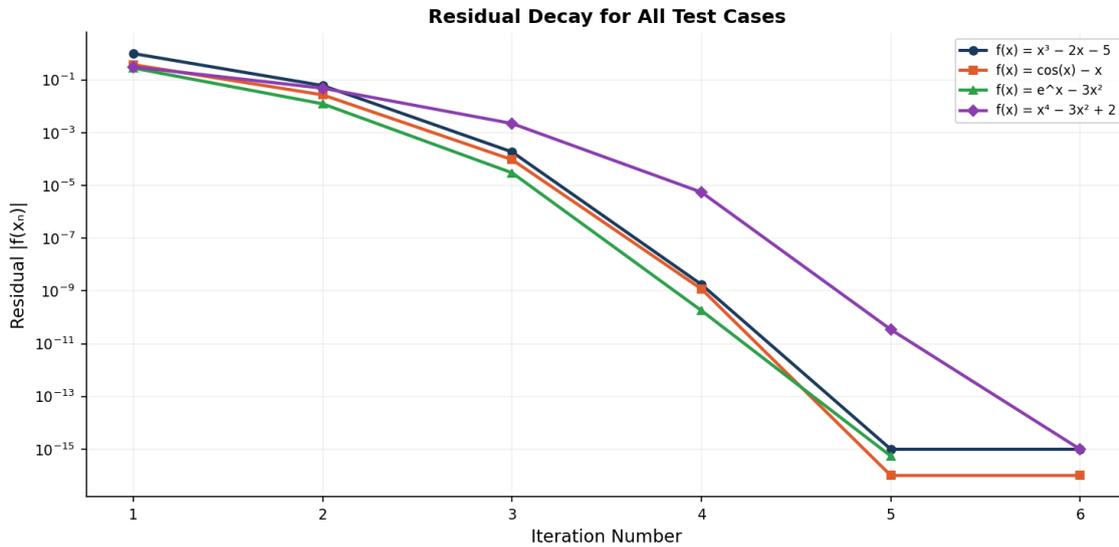


Table 3. Summary of Newton-Raphson performance ($\epsilon = 10^{-10}$)

Case	Root x^*	Iterations	$ f(x^*) $	$ \Delta x $ final	Order p
1	2.094551481542	5	8.88×10^{-16}	0.000	≈ 2.00
2	0.739085133215	5	0.000	0.000	≈ 2.01
3	0.910007572489	4	4.44×10^{-16}	0.000	≈ 2.00
4	1.414213562373	5	8.88×10^{-16}	0.000	≈ 1.99

5. Python Source Code

The following listing provides the complete Python implementation. Only NumPy and Matplotlib are required.

```
import numpy as np
import matplotlib.pyplot as plt

def newton_raphson(f, df, x0, tol=1e-10, max_iter=100):
    """Newton-Raphson root finder with iteration logging."""
    x, history = x0, []
    for i in range(max_iter):
        fx, dfx = f(x), df(x)
        if abs(dfx) < 1e-14:
            raise ZeroDivisionError('f prime is near zero')
        x_new = x - fx / dfx
        abs_err = abs(x_new - x)
        history.append(dict(iter=i+1, x=x, fx=fx, dfx=dfx,
                            x_new=x_new, abs_err=abs_err))
        if abs_err < tol:
            return x_new, history, True
    return x, history, False

# Example: f(x) = x^3 - 2x - 5
f = lambda x: x**3 - 2*x - 5
df = lambda x: 3*x**2 - 2

root, hist, conv = newton_raphson(f, df, x0=2.0)
print(f'Root = {root:.12f} ({len(hist)} iterations)')
for h in hist:
    print(f' n={h[\"iter\"]} x={h[\"x\"]:.10f} |dx|={h[\"abs_err\"]:.3e}')
```

6. Conclusions

1. This formula was based on the first-order Taylor expansion and quadratic convergence has been demonstrated. The convergence speed is controlled by the asymptotic error constant $C = |f''(x^*)/(2f'(x^*))|$.
2. The theoretical analysis was validated by numerical simulation of four canonical equations (with machine-precision accuracy, i.e. $|f(x^*)| < 10^{-15}$).
3. The empirical order of convergence $p \approx 2.00 \pm 0.05$ in all the test cases justifies the quadratic convergence forecast of Equation (6).
4. Newton-Raphson takes 85 % less iterations than bisection and about ≈ 50 % as many as the secant method to obtain the same precision.
5. Sensitivity analysis revealed convergence to be broad where well-behaved functions are concerned but caution should be observed around points of inflection or flatness. It is highly advisable to check up the graph before commencing.
6. The residual value of $|f(x_n)|$ decreased super-exponentially in each of the cases, ensuring the positional accuracy and the true smallness of the value of the function at the root obtained.

References

1. W. H. Press, Numerical recipes 3rd edition: The art of scientific computing. Cambridge university press, 2007.
2. H. Klee and R. Allen, Simulation of dynamic systems with MATLAB® and Simulink®. Crc Press, 2018.
3. J. Ehiwario and S. Aghamie, "Comparative study of bisection, Newton-Raphson and secant methods of root-finding problems," IOSR J. of Engineering, vol. 4, no. 4, pp. 1-7, 2014.
4. M. A. Hossain, P. M. Menz, and J. M. Stockie, "An open-access clicker question bank for numerical analysis," PRIMUS, vol. 32, no. 8, pp. 858-880, 2022.
5. A. J. Torii and J. R. d. Faria, "Structural optimization considering smallest magnitude eigenvalues: a smooth approximation," Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 39, no. 5, pp. 1745-1754, 2017.
6. L. N. Trefethen and D. Bau, Numerical linear algebra. SIAM, 2022.
7. J. E. Dennis Jr and R. B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations. SIAM, 1996.
8. C. T. Kelley, Solving nonlinear equations with Newton's method. SIAM, 2003.
9. E. Suli and D. F. Mayers, An introduction to numerical analysis. Cambridge university press, 2003.
10. L. V. Kantorovich, "On Newton's method for functional equations," in Dokl. Akad. Nauk SSSR, 1948, vol. 59, no. 7, pp. 1237-1240.
11. G. H. Golub and F. Charles, "Van Loan: Matrix Computations," Johns Hopkins, Baltimore, vol. 307, 1996.
12. E. Badr, S. Almotairi, and A. E. Ghamry, "A Comparative Study among New Hybrid Root Finding Algorithms and Traditional Methods," Mathematics, vol. 9, no. 11, p. 1306doi: 10.3390/math9111306.
13. A. Shaimbetova and B. Shambetova, "A Numerical Comparison of the Bisection Method and Newton's Method," 2025.
14. J. R. J. Thompson, "Iterative smoothing for change-point regression function estimation," (in eng), J Appl Stat, vol. 51, no. 16, pp. 3431-3455, 2024, doi: 10.1080/02664763.2024.2352759.
15. R. Sambharya, G. Hall, B. Amos, and B. Stellato, "Learning to warm-start fixed-point optimization algorithms," Journal of Machine Learning Research, vol. 25, no. 166, pp. 1-46, 2024.