

Comparing Bisection and Newton–Raphson Methods for Solving Nonlinear Equations

Deepika, Assistant professor, Matu Ram Institute of Engineering and Management Collage Rohtak Haryana (124001)

Abstract

Numerical root-finding is essential when nonlinear equations cannot be solved conveniently by symbolic techniques. This paper compares the bisection and Newton–Raphson methods with respect to mathematical foundations, convergence order, error control, computational requirements, and practical reliability. Bisection is a bracket-preserving method whose convergence follows from continuity and a change of sign; Newton–Raphson is a derivative-based local method derived from tangent-line linearization. Three representative problems—a cubic equation, a transcendental cosine equation, and an exponential equation—are examined under a common residual tolerance of 10^{-10} , with a bracket-width condition additionally imposed on bisection. For the specified starting values, bisection requires 33–34 iterations, whereas Newton–Raphson requires 3–4. The faster local convergence of Newton–Raphson does not imply unconditional superiority: it can become unstable when derivatives are small, initial guesses are unsuitable, or roots are multiple. The study recommends selecting solvers on the basis of available derivative information, guaranteed bracketing, accuracy requirements, and failure risk. A safeguarded hybrid approach is identified as a useful compromise.

Keywords: nonlinear equations; bisection method; Newton–Raphson method; convergence; numerical error; iterative algorithms

1. Introduction

Equations of the form $f(x) = 0$ arise across mathematics, engineering, economics, physics, and computational science. For example, an unknown parameter may be determined by balancing forces, predicting a chemical equilibrium, or matching a numerical model to measurements. Polynomial equations of high degree and many transcendental equations generally lack practical closed-form solutions. Consequently, numerical methods construct sequences of approximations that converge toward a root. A useful algorithm must balance accuracy, execution cost, interpretability, and resilience to unfavorable starting information.

Among the classical methods, bisection and Newton–Raphson illustrate contrasting strategies. Bisection requires two endpoints that enclose a sign change and systematically contracts the interval. Newton–Raphson begins with a single estimate, incorporates a first derivative, and usually improves rapidly near a simple root. Their distinctive assumptions produce different strengths: one emphasizes reliable enclosure while the other emphasizes high local speed. Comparing iteration counts alone is therefore insufficient; meaningful evaluation also considers stopping tests, derivative evaluation, error guarantees, and the possibility of nonconvergence.

The objectives of this study are to derive both iteration formulas, establish their principal convergence properties, apply reproducible computational tests, and formulate practical selection criteria. The analysis concerns real scalar equations; multidimensional and stochastic root-finding fall outside its scope.

2. Mathematical Preliminaries

Let $f: I \rightarrow \mathbb{R}$ be a continuous real-valued function on an interval I . A root α satisfies $f(\alpha) = 0$. The root is simple if $f'(\alpha) \neq 0$ and multiple if $f'(\alpha) = 0$ with higher-order vanishing. For an approximation x_n , the absolute error and residual are distinct quantities:

$$e_n = |x_n - \alpha|, \quad r_n = |f(x_n)|. \quad (1)$$

The exact error is normally unavailable because α is unknown. A small residual does not necessarily imply a comparably small positional error when $|f'|$ is very small. If $|f'(x)| \geq m > 0$ throughout the relevant segment, the mean value theorem supplies the useful local estimate

$$|x_n - \alpha| \leq |f(x_n)| / m. \quad (2)$$

A convergent sequence has order p when $|x_{n+1} - \alpha| \approx C|x_n - \alpha|^p$ near α , for $C > 0$. Linear convergence corresponds to $p = 1$; quadratic convergence corresponds to $p = 2$. These

asymptotic descriptions must be distinguished from global guarantees and from raw wall-clock runtime.

3. Bisection Method

3.1 Algorithm and mathematical justification

Suppose f is continuous on $[a_0, b_0]$ and $f(a_0)f(b_0) < 0$. The intermediate value theorem guarantees at least one root in the interval. At iteration n , the midpoint is computed as

$$c_n = (a_n + b_n) / 2. \quad (3)$$

If $f(a_n)f(c_n) \leq 0$, retain $[a_n, c_n]$; otherwise retain $[c_n, b_n]$. An exact zero at the midpoint terminates the process. Each step preserves an enclosing interval and halves its length. It does not, however, distinguish between multiple roots inside the original bracket or detect roots of even multiplicity solely through a sign change.

$$b_n - a_n = (b_0 - a_0) / 2^n. \quad (4)$$

When c_n is the midpoint of the interval after n contractions, its distance from some enclosed root is at most half the current interval width:

$$|c_n - \alpha| \leq (b_0 - a_0) / 2^{n+1}. \quad (5)$$

Thus a prescribed interval-based accuracy can be guaranteed in advance. For a target midpoint error ε , any integer n satisfying $n \geq \log_2((b_0 - a_0)/(2\varepsilon))$ is sufficient. Bisection does not require derivatives and works even when the derivative is discontinuous, provided the function itself remains continuous and a valid sign-changing bracket exists.

3.2 Strengths and limitations

The greatest advantage of bisection is its predictable behavior: its enclosing interval shrinks monotonically, its accuracy bound is transparent, and each step requires only one new function evaluation after endpoint values are known. It is effective when the model is nonsmooth or when a guaranteed answer is more important than speed. The principal disadvantages are linear convergence and the prerequisite of a sign-changing interval. A tangent root such as $f(x) = (x - 1)^2$ illustrates a limitation: ordinary endpoints on either side produce positive function values, so the standard sign-change test fails even though a root exists.

4. Newton–Raphson Method

4.1 Derivation from linearization

Newton–Raphson assumes that f is differentiable in the region considered. A first-order Taylor approximation around x_n gives

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n). \quad (6)$$

Setting the approximation to zero yields the x -intercept of the tangent line and therefore the next iterate:

$$x_{n+1} = x_n - f(x_n) / f'(x_n), \quad f'(x_n) \neq 0. \quad (7)$$

At each step, one normally evaluates both f and f' . When derivatives are expensive, obtained numerically, or inaccurate, the per-step cost and stability can differ substantially from the ideal analytical picture.

4.2 Local convergence analysis

Let α be a simple root, let f possess a continuous second derivative near α , and assume the iterates remain sufficiently close to α . Taylor expansion around α , combined with Equation (7), gives the classical local error relation

$$x_{n+1} - \alpha = [f''(\alpha) / (2f'(\alpha))](x_n - \alpha)^2 + O((x_n - \alpha)^3). \quad (8)$$

Accordingly, Newton–Raphson is locally quadratic at a simple root, provided the usual regularity assumptions hold. Once an iterate enters the appropriate neighborhood, the number of correct digits can approximately double each step. This result is local rather than global: an initial estimate far from the target may generate oscillation, divergence, an undefined step, or convergence to an unintended root.

For a root α of multiplicity $m > 1$, standard Newton iteration often converges only linearly with asymptotic factor $(m - 1)/m$. If the multiplicity m is known, the modified iteration

$$x_{n+1} = x_n - m f(x_n) / f'(x_n) \quad (9)$$

may recover quadratic local convergence under suitable smoothness assumptions. This modification illustrates that convergence speed depends on the structure of the root, not solely on the algorithm name.

5. Research Methodology

A deterministic comparative experiment was designed using three nonlinear functions with different analytic forms. The test equations and initial data are reported in Table 1. All calculations use standard double-precision arithmetic. Bisection terminates only when both $|f(c)| < 10^{-10}$ and the half-width of the retained bracket is below 10^{-10} ; Newton–Raphson terminates when $|f(x_n)| < 10^{-10}$. These criteria are intentionally reported separately because Newton’s residual-only test does not itself certify an interval error bound. A maximum of 100 iterations is allowed for each method; no example reaches that limit.

Table 1. Test functions and initialization.

Test	Equation $f(x) = 0$	Bisection bracket	Newton x_0
A	$x^3 - x - 2 = 0$	[1, 2]	1.5
B	$\cos(x) - x = 0$	[0, 1]	0.5
C	$\exp(x) - 3x = 0$	[0, 1]	0.5

For comparability, the quantity reported as an iteration is one newly computed midpoint or one Newton update. Derivative expressions are calculated analytically: $f'A(x) = 3x^2 - 1$, $f'B(x) = -\sin(x) - 1$, and $f'C(x) = \exp(x) - 3$. Results summarize the final approximation, absolute residual, and number of iterations. Neither elapsed time nor a machine-independent operation count is claimed from these small tests, because implementation and hardware choices can dominate such measurements.

6. Numerical Results and Analysis

6.1 Cubic equation

For Test A, $f(x) = x^3 - x - 2$ changes sign between 1 and 2. Newton iteration from $x_0 = 1.5$ is well defined, since $3(1.5)^2 - 1 = 5.75$. The first three Newton approximations are 1.521739130435, 1.521379805965, and 1.521379706805. The residual falls below 10^{-10} after three updates. Bisection reaches the prescribed residual and interval-width conditions after 33 midpoints. The two procedures agree on the positive root to the displayed digits.

6.2 Transcendental cosine equation

For Test B, the equation $\cos(x) = x$ has a root in [0,1] because $f(0) = 1$ and $f(1) = \cos(1) - 1 < 0$. Newton–Raphson produces 0.755222417106 after its first step and converges to approximately 0.739085133215 in four updates. Bisection requires 34 midpoints. This example demonstrates that Newton’s efficiency extends beyond polynomial equations, although the initial condition remains influential.

6.3 Exponential equation

For Test C, $\exp(x) - 3x$ has a sign-changing bracket [0,1]. With $x_0 = 0.5$, Newton’s first update gives 0.610059654959, and the method meets its stopping criterion in four steps. Bisection needs 33 iterations to satisfy both stipulated tests. The smaller root is approximately 0.619061286736. The same exponential equation also possesses a larger positive root outside the selected interval; this reinforces that bracket selection and starting guesses can determine which solution is obtained.

Table 2. Numerical outcomes under the stated stopping criteria.

Test	Method	Iterations	Final approximation	$ f(x) $
A	Bisection	33	1.521379706799	3.28×10^{-11}
A	Newton–Raphson	3	1.521379706805	4.53×10^{-14}
B	Bisection	34	0.739085133246	5.11×10^{-11}
B	Newton–Raphson	4	0.739085133215	$< 10^{-15}$
C	Bisection	33	0.619061286678	6.67×10^{-11}
C	Newton–Raphson	4	0.619061286736	2.22×10^{-16}

Table 2 indicates a decisive iteration-count advantage for Newton–Raphson on the specified, well-behaved tests. The advantage reflects the quadratic local error reduction proved in Section 4. It should not be interpreted as an experimentally established runtime ratio: a Newton step uses derivative information and may be costlier than a midpoint step. Moreover, the stopping conditions are not equivalent in the strength of their positional-error guarantees.

7. Comparative Discussion

Table 3 summarizes the central mathematical trade-offs. Bisection is globally convergent within a valid sign-changing bracket for a continuous function, in the sense that its nested intervals converge to an enclosed root. Newton–Raphson guarantees rapid convergence only locally under smoothness and simple-root hypotheses. In applications with expensive simulations, the number of function and derivative evaluations may matter more than the nominal number of steps. For models supplied as black boxes, bisection is attractive because derivative information may not be accessible.

Table 3. Theoretical and practical comparison.

Criterion	Bisection	Newton–Raphson
Starting information	Two endpoints with sign change	One initial estimate
Regularity	Continuity on bracket	Differentiability; local smoothness
Typical convergence	Linear; interval halves	Quadratic near a simple root
Derivative required	No	Yes
A priori error bound	Yes, by bracket width	Generally no without extra bounds
Main failure mode	No valid sign-change bracket	Poor guess or near-zero derivative
Per-step work	One new f evaluation	f and f' evaluations

A sensitivity example makes the Newton limitation concrete. For $f(x) = x^3 - 2x + 2$, the starting value $x_0 = 0$ produces $x_1 = 1$, and $x_1 = 1$ returns $x_2 = 0$. Thus the raw method cycles without converging, despite the existence of a real root. In contrast, bisection on $[-2, -1]$ is guaranteed to locate a root because $f(-2) = -2$ and $f(-1) = 3$. Neither example proves universal superiority: a poor bracket can be difficult to identify, while a well-chosen Newton starting point can save many evaluations.

A practical hybrid combines both principles. First identify a sign-changing bracket, then attempt a Newton step from a reasonable point. Accept that step only when it lies inside the current bracket and meets safeguards against tiny derivatives or insufficient progress; otherwise use the midpoint. Update the bracket using the sign of the new residual. Such safeguarding maintains interval-based reliability while allowing rapid local acceleration. Specific implementations require precise acceptance rules, and their convergence characteristics should be established rather than assumed.

7.1 Accuracy, conditioning, and stopping tests

A common source of confusion in root-finding comparisons is the distinction between an approximate zero of the function and an accurate estimate of its root. For a well-conditioned simple root, Taylor expansion gives $f(x) \approx f'(\alpha)(x - \alpha)$, and the residual is approximately proportional to the error. Yet for a nearly flat graph, a small residual can coexist with a substantial displacement from α . The local absolute root condition number is therefore related to $1/|f'(\alpha)|$. Reporting only the number of digits in $f(x)$ would obscure this important issue. A bracket method can additionally report a valid interval containing a root, provided the continuity and sign conditions are maintained.

Stopping rules should reflect intended use. In scientific computation, it is common to combine a residual requirement, a step-size criterion, and a maximum iteration safeguard; bracketed algorithms can replace step size with an explicit interval-width certificate. The present tests deliberately use a more stringent bisection rule because bisection naturally supplies such a certificate. Consequently, the iteration figures establish performance for the stated protocol, rather than a perfectly uniform positional accuracy comparison. If the desired quantity has

physical units, the tolerance should likewise be scaled to the magnitude and interpretation of the unknown.

7.2 Computational cost and implementation considerations

Function-evaluation cost varies between applications. Computing a polynomial and its derivative can be inexpensive; evaluating a nonlinear finite-element simulation and its sensitivity can be much more demanding. A fair timing experiment would record the cost of f , the cost of f' , the setup time, and any failures requiring a restarted initial guess. It should also use the same precision, comparable programming environments, and repeated runs if wall-clock time is reported. Derivatives obtained by finite differences introduce truncation and round-off errors, whereas automatic differentiation can deliver accurate derivatives with additional computational and memory demands. Newton–Raphson also requires protection against division by an extremely small $f'(x_n)$.

In a robust software implementation, invalid inputs must be detected explicitly. Bisection should check that endpoint evaluations are finite and either satisfy the requested tolerance or have opposite signs. Newton iteration should check that both $f(x_n)$ and $f'(x_n)$ are finite before constructing a step. A near-zero derivative, an iterate outside the permitted physical domain, or an increase in the residual can trigger damping, backtracking, or a bracket fallback. These implementation details affect practical reliability even though they are not visible in the compact textbook iteration formulas.

8. Limitations and Implications

The present comparison is deliberately controlled rather than comprehensive. Three scalar examples cannot characterize all nonlinear equations, particularly highly oscillatory functions, nearly multiple roots, discontinuities, noisy evaluations, or ill-conditioned models. The use of analytical derivatives favors Newton–Raphson compared with settings where automatic differentiation, finite differences, or expensive adjoint computations are necessary. Similarly, the initial guesses were chosen in favorable neighborhoods, not sampled at random. Future investigations could vary starting points systematically, examine derivative perturbations, measure function-equivalent computational cost, and evaluate safeguarded algorithms over broader benchmark collections.

For educational and engineering practice, both methods remain valuable. Bisection supplies a transparent baseline, an independent numerical check, and a rigorous stopping certificate. Newton–Raphson offers substantial acceleration when the derivative is available and a reliable starting guess can be obtained. Using them together is often more defensible than choosing either method solely because of its asymptotic convergence order.

9. Conclusion

This study compared bisection and Newton–Raphson through derivations, convergence analysis, and three reproducible numerical experiments. Bisection provided predictable linear convergence and certified shrinking brackets, whereas Newton–Raphson exhibited much faster local convergence for the selected simple roots. Under a residual threshold of 10^{-10} and an additional bracket-width requirement for bisection, the experiments required 33–34 bisection iterations versus 3–4 Newton updates. These results confirm an iteration advantage for Newton–Raphson under favorable initial conditions but do not establish a universal computational advantage. Derivative availability, root multiplicity, numerical conditioning, initial guesses, and the required guarantee should guide solver selection. For reliability-sensitive applications, safeguarded Newton iteration inside an enclosing bracket presents a sound general strategy. The main conclusion is therefore conditional: Newton–Raphson is normally preferable for rapid local refinement, while bisection remains preferable when dependable global bracketing and certified error control dominate.

References

1. Burden, R. L., Faires, J. D., & Burden, A. M. (2016). Numerical Analysis (10th ed.). Cengage Learning.

2. Chapra, S. C., & Canale, R. P. (2015). Numerical Methods for Engineers (7th ed.). McGraw-Hill Education.
3. Kelley, C. T. (2003). Solving Nonlinear Equations with Newton's Method. Society for Industrial and Applied Mathematics.
4. Ortega, J. M., & Rheinboldt, W. C. (1970). Iterative Solution of Nonlinear Equations in Several Variables. Academic Press.
5. Stoer, J., & Bulirsch, R. (2002). Introduction to Numerical Analysis (3rd ed.). Springer.
6. Süli, E., & Mayers, D. F. (2003). An Introduction to Numerical Analysis. Cambridge University Press.