Indian Institute of Technology Roorkee
MAB-103: Numerical Methods

Unit-III Roots of non-linear equations Session-2025-26

In many scientific and engineering problems, we encounter equations of the form

$$f(x) = 0,$$

where $f(x)$ is a nonlinear function. Unlike linear equations, which can be solved exactly using algebraic methods, nonlinear equations often do not have closed-form solutions. For instance, equations like $x^3 - x - 2 = 0$ or transcendental equations such as $\sin(x) - x/2 = 0$ cannot be solved analytically in most cases.

Despite the lack of exact solutions, finding the roots of nonlinear equations is crucial because these roots represent important physical quantities, such as equilibrium points in mechanical systems, concentrations in chemical reactions, and eigenvalues in structural analysis.

Why Numerical Methods?

Given the complexity of nonlinear equations, we rely on numerical methods to approximate their roots. Numerical methods provide systematic procedures for finding approximate solutions to equations that cannot be solved exactly. The importance of these methods lies in their ability to:

- Handle a wide range of nonlinear equations that arise in real-world applications.
- Provide approximate solutions with controllable accuracy.
- Offer practical tools that can be implemented on computers, making them essential for solving large-scale problems in science and engineering.

Examples of Applications

- **Physics:** Determining the energy levels of quantum systems involves solving nonlinear eigenvalue problems.
- **Engineering:** In structural engineering, the buckling load of a column is found by solving nonlinear equilibrium equations.
- **Economics:** Market equilibrium models often require solving nonlinear equations to find prices or interest rates.

In conclusion, numerical methods for finding roots of nonlinear equations are indispensable tools in both theoretical studies and practical applications. They enable us to solve problems that are otherwise intractable, providing insights and solutions that drive advancements in various fields of science and engineering.

0.1 Fixed-Point Iteration

0.1.1 Introduction

- The fixed-point method is an iterative technique for finding approximate solutions to nonlinear equations of the form $f(x) = 0$. The method is based on rewriting the equation in the form $x = g(x)$, where $g(x)$ is a function derived from $f(x)$. The solution is then found by iteratively applying $g(x)$ starting from an initial guess.

- Given a nonlinear equation $f(x) = 0$, we rewrite it as:

  $$x = g(x),$$

  where $g(x)$ is chosen such that the fixed point of $g(x)$, denoted by x^*, satisfies $x^* = g(x^*)$. This x^* is also a solution to $f(x) = 0$.

- The fixed-point iteration is defined by:

  $$x_{n+1} = g(x_n),$$

  where x_n is the n-th approximation to the root. The process is repeated until the difference between successive approximations is smaller than a predetermined tolerance, i.e., $|x_{n+1} - x_n| < \epsilon$, for some small $\epsilon > 0$.

0.1.2 Convergence Criteria

For the fixed-point iteration to converge to the true root x^*, the function $g(x)$ must satisfy the following conditions:

- **Existence of a fixed point:** There exists a point x^* such that $x^* = g(x^*)$.

- **Contraction Mapping:** There exists a constant $0 < \alpha < 1$ such that for all x and y in the interval $[a, b]$,

  $$|g(x) - g(y)| \leq \alpha |x - y|.$$

  This ensures that the sequence $\{x_n\}$ will converge to x^*.

To ensure the convergence creteria, we make three assumptions of $g(x)$:

- $a \leq g(x) \leq b$ for all $a \leq x \leq b$.

- The function $g(x)$ is continuous.

- The iteration function $g(x)$ is differentiable on $I = [a, b]$. Further, there exists a constant $0 < K < 1$ such that

  $$|g'(x)| \leq K, \quad \forall x \in I$$

0.1.3 Example: Solving a Nonlinear Equation

Consider the nonlinear equation:

$$x^2 - 2 = 0.$$

We can rewrite this equation in the form $x = g(x)$. One possible choice for $g(x)$ is:

$$g(x) = \sqrt{2}.$$

Let's perform the fixed-point iteration starting from an initial guess $x_0 = 1.5$.

Iteration 1:

$$x_1 = g(x_0) = \sqrt{2} = 1.4142.$$

Iteration 2:

$$x_2 = g(x_1) = \sqrt{2} = 1.4142.$$

The process continues until the values of x_n stabilize.

0.1.4 Example: Convergence Analysis

Consider $g(x) = \sin(x) + x^2 - 1 = 0$ on $I = [0, 1]$ as an example. There are three possible choices for the iteration function, namely,

- $g_1(x) = \sin^{-1}(1 - x^2)$,
- $g_2(x) = -\sqrt{1 - \sin(x)}$,
- $g_3(x) = \sqrt{1 - \sin(x)}$.

Here $|g_1'(x)| = |\frac{-2}{\sqrt{1-x^2}}| > 1$ for $x \in I$. If we take $g_2(x)$, clearly the assumption 1 is violated and therefore is not suitable for the iteration process. It is evident that $|g_3'(x)| < 1$.

0.1.5 Stopping Criteria

The iteration process is typically stopped when the following criterion is met:

$$|x_{n+1} - x_n| < \epsilon,$$

where ϵ is a small positive number representing the desired accuracy.

0.1.6 Advantages and Limitations

Advantages

- Simple to implement.
- Useful when the function $g(x)$ is easily derived from $f(x)$.

Limitations

- Convergence is not guaranteed for all choices of $g(x)$.
- The method can be slow to converge, especially if $|g'(x)|$ is close to 1.

0.1.7 Conclusion

The fixed-point method is a fundamental iterative technique for solving nonlinear equations. While it is straightforward and useful in many scenarios, careful consideration must be given to the choice of $g(x)$ and the convergence criteria to ensure successful application.

0.2 The Bisection Method

0.2.1 Introduction

- The Bisection Method is a simple and robust numerical technique for finding roots of continuous nonlinear equations of the form $f(x) = 0$. It is based on the Intermediate Value Theorem and works by repeatedly narrowing down an interval that contains the root.

- The Bisection Method requires a continuous function $f(x)$ and an interval $[a, b]$ where the function changes sign, i.e., $f(a) \cdot f(b) < 0$. The method works by iteratively reducing the interval until the root is approximated to within a desired tolerance.

0.2.2 Algorithm

1. Initial Interval: Choose an interval $[a_0, b_0]$ such that $f(a_0) \cdot f(b_0) < 0$.
2. Midpoint Calculation: Compute the midpoint of the interval:

$$c_n = \frac{a_n + b_n}{2}.$$

3. Evaluate the Function : Calculate $f(c_n)$.

- If $f(c_n) = 0$, then c_n is the root.

- If $f(a_n) \cdot f(c_n) < 0$, set $b_{n+1} = c_n$ and $a_{n+1} = a_n$.

- If $f(b_n) \cdot f(c_n) < 0$, set $a_{n+1} = c_n$ and $b_{n+1} = b_n$.

4. Stopping Criterion: Repeat steps 2 and 3 until the width of the interval $[a_n, b_n]$ is smaller than a given tolerance ϵ, i.e., $|b_n - a_n| < \epsilon$.

0.2.3 Pseudocode

```
Given a function f(x), and interval [a, b] where f(a) * f(b) < 0:
  while (b - a) / 2 > tolerance:
    c = (a + b) / 2
    if f(c) == 0:
      return c
    elif f(a) * f(c) < 0:
      b = c
    else:
      a = c
  return (a + b) / 2
```

0.2.4 Convergence Analysis

The Bisection Method converges linearly to the root. The number of iterations required to achieve a given tolerance ϵ can be estimated by:

$$n \geq \frac{\log\left(\frac{b_0 - a_0}{\epsilon}\right)}{\log(2)}.$$

This means that with each iteration, the interval size is halved, ensuring that the error decreases exponentially.

0.2.5 Example

Let's apply the Bisection Method to find the root of the equation:

$$f(x) = x^3 - 4x + 1 = 0$$

in the interval $[1, 2]$.

Step 1: Initial Interval
Check the signs:

$$f(1) = -2, \quad f(2) = 5.$$

Since $f(1) \cdot f(2) < 0$, there is a root in $[1, 2]$.

Step 2: First Iteration
Compute the midpoint:

$$c_1 = \frac{1 + 2}{2} = 1.5.$$

Evaluate the function at the midpoint:

$$f(1.5) = 1.5^3 - 4 \cdot 1.5 + 1 = -1.375.$$

Since $f(1) \cdot f(1.5) < 0$, update the interval to $[1, 1.5]$.

Step 3: Second Iteration
Compute the new midpoint:

$$c_2 = \frac{1 + 1.5}{2} = 1.25.$$

Evaluate the function:

$$f(1.25) = 1.25^3 - 4 \cdot 1.25 + 1 = -0.796875.$$

Update the interval to $[1.25, 1.5]$.

Continue Iterating until the interval width is less than a chosen tolerance, say $\epsilon = 0.001$.

0.2.6 Advantages and Limitations

Advantages

- Guaranteed convergence if $f(x)$ is continuous on $[a, b]$ and $f(a) \cdot f(b) < 0$.
- Simple and easy to implement.

Limitations

- Slow convergence rate (linear).
- Only applicable to continuous functions where a sign change occurs.
- Requires a good initial interval where the function changes sign.

0.2.7 Conclusion

The Bisection Method is a reliable and straightforward technique for finding roots of nonlinear equations. While it has some limitations, its guaranteed convergence makes it a valuable tool in numerical analysis, particularly when other methods may fail to provide a root.

0.3 Secant Method

0.3.1 Introduction

The Secant Method is used for finding roots of a function $f(x) = 0$. It is an improvement over the Bisection method in terms of speed. Unlike Newton's method, it does not require the evaluation of the derivative of the function.

0.3.2 Methodology

Start with two initial guesses x_0 and x_1. The formula for the iterative step is:

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

Continue iterating until $|x_{n+1} - x_n|$ is less than the desired tolerance.

Example

We want to find a root of the function $f(x) = x^2 - 2$ using the secant method.

- Initial Guesses Let $x_0 = 1$ and $x_1 = 2$.
- First Iteration

  $$x_2 = x_1 - \frac{f(x_1) \cdot (x_1 - x_0)}{f(x_1) - f(x_0)}$$

  Substituting the values:

  $$f(x_0) = 1^2 - 2 = -1, \quad f(x_1) = 2^2 - 2 = 2$$

  $$x_2 = 2 - \frac{2 \cdot (2 - 1)}{2 - (-1)} = 2 - \frac{2}{3} = \frac{4}{3} \approx 1.3333$$

- Second Iteration

  Now, using $x_1 = 2$ and $x_2 = \frac{4}{3}$:

  $$f(x_2) = \left(\frac{4}{3}\right)^2 - 2 = \frac{16}{9} - 2 = \frac{16}{9} - \frac{18}{9} = -\frac{2}{9}$$

  $$x_3 = x_2 - \frac{f(x_2) \cdot (x_2 - x_1)}{f(x_2) - f(x_1)}$$

Substituting the values:

$$x_3 = \frac{4}{3} - \frac{-\frac{2}{9} \cdot \left(\frac{4}{3} - 2\right)}{-\frac{2}{9} - 2} \approx 1.4$$

- Third Iteration

  Continuing the iterations:

$$x_4 = x_3 - \frac{f(x_3) \cdot (x_3 - x_2)}{f(x_3) - f(x_2)} \approx 1.4142$$

  After three iterations, we find that $x_4 \approx 1.4142$, which is a good approximation of $\sqrt{2}$.

0.3.3 Algorithm

1. Choose initial guesses x_0 and x_1.
2. Compute x_{n+1} using the formula above.
3. Check for convergence.
4. If the convergence criterion is not met, set $x_{n-1} = x_n$ and $x_n = x_{n+1}$, then repeat the process.

0.3.4 Advantages

- Faster than the Bisection method.
- Does not require the calculation of derivatives.

0.3.5 Disadvantages

- May not converge for poor initial guesses.
- Slower than Newton's method if derivatives are easy to calculate.

0.4 Regula Falsi Method

0.4.1 Introduction

The Regula Falsi Method, also known as the False Position Method, is a bracketing method for finding the roots of a function. The method combines features of the Bisection method and the Secant method.

0.5 Methodology

Start with two initial points x_0 and x_1 such that $f(x_0) \cdot f(x_1) < 0$. Compute the point where the secant line crosses the x-axis:

$$x_2 = x_1 - \frac{f(x_1) \cdot (x_1 - x_0)}{f(x_1) - f(x_0)}$$

Replace the point in the interval that has the same sign as $f(x_2)$, ensuring that the root remains bracketed.

0.5.1 Algorithm

1. Choose initial guesses x_0 and x_1.
2. Compute x_2 using the formula above.
3. Check the sign of $f(x_2)$.
4. Update the interval by replacing x_0 or x_1 with x_2.
5. Repeat until the desired tolerance is achieved.

Example

We want to find a root of the function $f(x) = x^3 - 4x + 1$ using the regular false method.

- Initial Guesses

  Let $a_0 = 0$ and $b_0 = 2$.

- First Iteration

  Calculate the function values at the endpoints:

$$f(a_0) = f(0) = 0^3 - 4 \cdot 0 + 1 = 1$$
$$f(b_0) = f(2) = 2^3 - 4 \cdot 2 + 1 = 8 - 8 + 1 = 1$$

  The next approximation c_1 is given by:

$$c_1 = b_0 - \frac{f(b_0) \cdot (b_0 - a_0)}{f(b_0) - f(a_0)}$$

  Substituting the values:

$$c_1 = 2 - \frac{1 \cdot (2 - 0)}{1 - 1} = 2$$

  Since $f(a_0) \cdot f(c_1) < 0$, update the interval:

$$a_1 = 0, \quad b_1 = c_1 = 2$$

- Second Iteration

  Calculate the new c_2:

$$f(a_1) = f(0) = 1, \quad f(b_1) = f(2) = 1$$

$$c_2 = b_1 - \frac{f(b_1) \cdot (b_1 - a_1)}{f(b_1) - f(a_1)}$$

  Substituting the values:

$$c_2 = 2 - \frac{1 \cdot (2-0)}{1-1} = 2$$

Update the interval:

$$a_2 = 0, \quad b_2 = c_2 = 2$$

- Subsequent Iterations

  Continue in a similar manner until the desired accuracy is achieved.

0.5.2 Advantages

- Guarantees convergence if the initial interval is chosen correctly.
- Does not require the calculation of derivatives.

0.5.3 Disadvantages

- Can be slower than other methods like Newton's method.
- The convergence may be slow if the function is nearly linear over the interval.

0.6 Newton-Raphson Method

0.7 Introduction

The Newton-Raphson Method is one of the most efficient methods for finding roots of a nonlinear equation $f(x) = 0$. It requires the function $f(x)$ to be differentiable and uses the tangent line at an initial guess to approximate the root.

0.7.1 Methodology

Given an initial guess x_0, the next approximation x_{n+1} is found using the formula:

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

This process is repeated until the difference between successive approximations is smaller than a predefined tolerance.

0.7.2 Algorithm

1. Choose an initial guess x_0.
2. Compute x_{n+1} using the Newton-Raphson formula:

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

3. Check for convergence, i.e., if $|x_{n+1} - x_n| <$ tolerance.
4. If convergence is not achieved, set $x_n = x_{n+1}$ and repeat the process.

Numerical Example: Newton-Raphson Method

We want to find the root of the equation $f(x) = x^3 - 2x - 5 = 0$ using the Newton-Raphson method.

Step 1: Define the Function and Its Derivative

The function is:

$$f(x) = x^3 - 2x - 5$$

The derivative of the function is:

$$f'(x) = 3x^2 - 2$$

Step 2: Initial Guess

Let's choose an initial guess $x_0 = 2$.

Step 3: First Iteration

Compute $f(x_0)$ and $f'(x_0)$:

$$f(2) = 2^3 - 2(2) - 5 = -1$$
$$f'(2) = 3(2)^2 - 2 = 10$$

Apply the Newton-Raphson formula:

$$x_1 = 2 - \frac{-1}{10} = 2.1$$

Step 4: Second Iteration

Compute $f(x_1)$ and $f'(x_1)$:

$$f(2.1) = (2.1)^3 - 2(2.1) - 5 = 0.061$$

$$f'(2.1) = 3(2.1)^2 - 2 = 11.23$$

Apply the Newton-Raphson formula:

$$x_2 = 2.1 - \frac{0.061}{11.23} = 2.0946$$

Step 5: Third Iteration

Compute $f(x_2)$ and $f'(x_2)$:

$$f(2.0946) = (2.0946)^3 - 2(2.0946) - 5 \approx 0.0013$$

$$f'(2.0946) = 11.1592$$

Apply the Newton-Raphson formula:

$$x_3 = 2.0946 - \frac{0.0013}{11.1592} = 2.09448$$

Result

The root of the equation $f(x) = x^3 - 2x - 5 = 0$ is approximately $x \approx 2.09448$.

0.7.3 Advantages

- **Quadratic Convergence**: The method converges very quickly when the initial guess is close to the actual root.
- **Simplicity**: The formula is easy to implement and understand.

0.7.4 Disadvantages

- **Requires Derivatives**: The method requires the calculation of the derivative $f'(x)$, which may not always be easy.
- **Sensitivity to Initial Guess**: The method may fail to converge if the initial guess is not close to the root or if the derivative is zero or close to zero at any step.
- **Possible Divergence**: If the function is not well-behaved, the method may diverge or converge to the wrong root.

0.7.5 Geometric Interpretation

The Newton-Raphson Method can be understood geometrically as using the tangent line at a point x_n to approximate the root. The next approximation x_{n+1} is the point where the tangent line crosses the x-axis.

Definition: Order of Convergence for Iterative Root-Finding Methods

The **order of convergence** of an iterative method quantifies *how fast* the sequence of approximations $x_0, x_1, x_2, \ldots$ converges to the exact root α.

It is defined by the behavior of the errors $e_n = |x_n - \alpha|$ as $n \to \infty$.

Mathematical Definition

An iterative method is said to converge with **order** $p \geq 1$ if there exists a positive constant C (called the **asymptotic error constant**) such that:

$$\lim_{n \to \infty} \frac{|e_{n+1}|}{|e_n|^p} = C$$

In simpler terms, for sufficiently large n, the error at the next iteration is *approximately proportional* to the p-th power of the current error:

$$|e_{n+1}| \approx C \cdot |e_n|^p$$

The value of p tells you the **rate of convergence**:

Order (p)	Common Name	Interpretation
$p = 1$	**Linear**	The number of correct decimal places increases by a constant amount each step. Error reduces as $e_{n+1} \approx C \cdot e_n$.
$p = 2$	**Quadratic**	The number of correct decimal places roughly **doubles** each step. Error reduces as $e_{n+1} \approx C \cdot e_n^2$.
$p = 3$	**Cubic**	The number of correct decimal places roughly **triples** each step. Error reduces as $e_{n+1} \approx C \cdot e_n^3$.
$1 < p < 2$	**Superlinear**	Faster than linear but not yet quadratic. Error reduces as $e_{n+1} \approx C \cdot e_n^p$. The Secant Method is a key example ($p \approx 1.618$).

Important Note on $p = 1$ (Linear): For linear convergence, the constant C must be strictly less than 1 ($0 < C < 1$). If $C \geq 1$, the method may not converge or will converge too slowly to be useful.

Why is Order Important?

The order p is the most important measure of an iterative method's efficiency because:

- **It predicts speed.** A higher-order method will eventually converge much faster than a lower-order one, requiring far fewer iterations to achieve the same precision.
- **It helps choose the right method.** For simple roots, Newton's method (order 2) is preferred over the Fixed-Point method (order 1). For multiple roots, the Modified Newton's method (order 2) is preferred over the standard Newton's method (which drops to order 1).

Examples of Methods and Their Orders

Method	**Iteration Formula**	**Typical Order (p)**	**Condition**
Bisection	N/A (bracketing)	**1** (Linear)	$C = \frac{1}{2}$
Fixed-Point Iteration	$x_{n+1} = g(x_n)$	**1** (Linear)	If $g'(\alpha) \neq 0$
Newton-Raphson	$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$	**2** (Quadratic)	Simple root ($f'(\alpha) \neq 0$), good initial guess.
			For a multiple root of order m, it drops to $p = 1$.
Modified Newton's	$x_{n+1} = x_n - m \cdot \frac{f(x_n)}{f'(x_n)}$	**2** (Quadratic)	For a root of known multiplicity m.
Regula Falsi	$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$	1 (Linear)	Less expensive than Newton's.

Convergence of the Fixed-Point Iteration

Consider the nonlinear equation

$$f(x) = 0,$$

which we rewrite in fixed-point form

$$x = g(x).$$

The fixed-point iteration is then

$$x_{k+1} = g(x_k), \quad k = 0, 1, 2, \ldots$$

Theorem 1 (Convergence of Fixed-Point Iteration). *Suppose $g : [a, b] \to [a, b]$ is continuous and satisfies a Lipschitz condition*

$$|g(x) - g(y)| \leq L|x - y|, \quad \forall x, y \in [a, b],$$

with a constant $0 \leq L < 1$. Then:

1. *There exists a unique fixed point $x^* \in [a, b]$ such that $g(x^*) = x^*$.*
2. *For any starting value $x_0 \in [a, b]$, the sequence $\{x_k\}$ defined by $x_{k+1} = g(x_k)$ converges to x^*.*
3. *The convergence is at least linear:*

$$|x_{k+1} - x^*| \leq L|x_k - x^*|, \quad k = 0, 1, 2, \ldots$$

Moreover,

$$|x_k - x^*| \leq \frac{L^k}{1 - L} |x_1 - x_0|.$$

Proof. Since g maps $[a, b]$ into itself and is a contraction with constant $L < 1$.

To show convergence, note that

$$|x_{k+1} - x^*| = |g(x_k) - g(x^*)| \leq L|x_k - x^*|.$$

By induction,

$$|x_k - x^*| \leq L^k|x_0 - x^*|, \quad k = 0, 1, 2, \ldots$$

which tends to zero as $k \to \infty$ since $0 \leq L < 1$. Thus $x_k \to x^*$ with at least linear rate of convergence. □

Remark 1. *If $g'(x^*) = 0$ and g is continuously differentiable, the convergence is superlinear (quadratic if g'' exists).*

Convergence of Newton's Method

Newton's method is an iterative scheme for solving the nonlinear equation

$$f(x) = 0,$$

given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots$$

Theorem 2 (Quadratic Convergence of Newton's Method). *Let $f \in C^2([a, b])$, and suppose $x^* \in (a, b)$ satisfies $f(x^*) = 0$ with $f'(x^*) \neq 0$. If the initial guess x_0 is sufficiently close to x^*, then the sequence $\{x_k\}$ defined by Newton's method converges to x^*. Moreover, the convergence is quadratic:*

$$|x_{k+1} - x^*| \leq C|x_k - x^*|^2,$$

for some constant $C > 0$ when k is large enough.

Proof. By Taylor's theorem, expand $f(x^*)$ about x_k:

$$f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + \tfrac{1}{2}f''(\xi_k)(x^* - x_k)^2,$$

for some ξ_k between x_k and x^*. Since $f(x^*) = 0$ and deviding by $f'(x_k)$, this reduces to

$$0 = \frac{f(x_k)}{f'(x_k)} + (x^* - x_k) + \frac{f''(\xi_k)}{2f'(x_k)}(x_k - x^*)^2.$$

Using Newton Rapshon formula gives

$$x^* - x_{k+1} = -\frac{f''(\xi_k)}{2f'(x_k)}(x_k - x^*)^2.$$

Since $f'(x^*) \neq 0$ and f' is continuous, $f'(x_k)$ remains bounded away from 0 near x^*. Also, f'' is continuous and bounded on $[a, b]$. Hence there exists $C > 0$ such that

$$|x_{k+1} - x^*| \leq C|x_k - x^*|^2,$$

which shows quadratic convergence. □

Convergence of the Regula Falsi Method

Theorem 3. *Let $f : [a, b] \to \mathbb{R}$ be continuous with $f(a)f(b) < 0$. Then the sequence $\{c_k\}$ generated by the Regula Falsi method converges to a root $x^* \in [a, b]$. The convergence is guaranteed and at least linear.*

Proof. At each step, define

$$c_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}.$$

By construction, $f(a_k)f(b_k) < 0$ for all k, so the root is always bracketed in $[a_k, b_k]$.

The update rule ensures:

$$a_{k+1} = \begin{cases} a_k, & f(a_k)f(c_k) < 0, \\ c_k, & f(c_k)f(b_k) < 0, \end{cases} \qquad b_{k+1} = \begin{cases} c_k, & f(a_k)f(c_k) < 0, \\ b_k, & f(c_k)f(b_k) < 0. \end{cases}$$

Hence $\{a_k\}$ is monotone nondecreasing and bounded above, while $\{b_k\}$ is monotone nonincreasing and bounded below. Thus both converge: $a_k \to \alpha$, $b_k \to \beta$ with $\alpha \leq \beta$.

By continuity of f, we must have $f(\alpha)f(\beta) \leq 0$. Since the interval shrinks around a single point, $\alpha = \beta = x^*$ is a root.

Finally, because the method relies on linear interpolation, the improvement in the approximation satisfies

$$|c_{k+1} - x^*| \leq q\,|c_k - x^*|, \quad 0 < q < 1,$$

which shows at least linear convergence. □

Remark 2. *In contrast, the secant method achieves superlinear convergence, and Newton's method achieves quadratic convergence under suitable smoothness assumptions. The Regula Falsi method is slower but always safe because it maintains bracketing.*

The Problem with the Standard Newton-Raphson Method

The standard Newton-Raphson method uses the iteration:

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

It converges **quadratically** (very fast) to a root α if $f'(\alpha) \neq 0$ (a **simple root**).

However, if α is a **multiple root** of multiplicity $m > 1$ (i.e., $f(\alpha) = f'(\alpha) = \cdots = f^{(m-1)}(\alpha) = 0$, but $f^{(m)}(\alpha) \neq 0$), the standard method struggles. Its convergence becomes only **linear**, and the error constant is $(1 - \frac{1}{m})$, which gets worse as m increases.

Why? Because both $f(x_n)$ and $f'(x_n)$ approach zero as x_n approaches the root α. Their ratio $f(x_n)/f'(x_n)$ does not go to zero as quickly, slowing down the convergence.

The Solution: Modified Newton-Raphson Method

The modified method restores quadratic convergence by explicitly accounting for the root's multiplicity m.

The Modified Iteration Formula:

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

How does it work? The factor m compensates for the shallow slope of $f'(x)$ near the multiple root. It effectively "jumps" the correct distance towards the root, just as the standard method does for simple roots.

When to use it?

- When you know the multiplicity m of the root in advance.
- If you suspect a multiple root and can estimate m.

Detailed Explanation: Why the Modification Works

Let α be a root of multiplicity m. Near α, the function can be approximated by:

$$f(x) \approx (x - \alpha)^m \cdot g(x)$$

where $g(\alpha) \neq 0$.

Now, let's compute the derivative (using the product rule):

$$f'(x) \approx m(x - \alpha)^{m-1} \cdot g(x) + (x - \alpha)^m \cdot g'(x)$$

Let's see what the standard Newton step calculates:

$$\frac{f(x)}{f'(x)} \approx \frac{(x - \alpha)^m \cdot g(x)}{m(x - \alpha)^{m-1} \cdot g(x) + (x - \alpha)^m \cdot g'(x)} = \frac{(x - \alpha) \cdot g(x)}{m \cdot g(x) + (x - \alpha) \cdot g'(x)}$$

As $x \to \alpha$, $(x - \alpha) \to 0$ and $g(x) \to g(\alpha)$, so:

$$\frac{f(x)}{f'(x)} \approx \frac{(x - \alpha) \cdot g(\alpha)}{m \cdot g(\alpha)} = \frac{(x - \alpha)}{m}$$

Therefore, the standard Newton step is:

$$x_{n+1} \approx x_n - \frac{(x_n - \alpha)}{m}$$

The error is reduced by a factor of $(1 - \frac{1}{m})$ each time (linear convergence).

Now, let's apply the *modified* step:

$$x_{n+1} = x_n - m \cdot \frac{f(x_n)}{f'(x_n)} \approx x_n - m \cdot \frac{(x_n - \alpha)}{m} = x_n - (x_n - \alpha) = \alpha$$

The modified step cancels the error perfectly in one step (in the approximation), leading to quadratic convergence.

A Complete Example

Let's solve $f(x) = x^4 - 6.75x^2 + 6.25x - 1.5 = 0$. We know (from factoring) that $x = 0.5$ is a **double root** (multiplicity $m = 2$).

Let's use both the standard and modified methods starting from $x_0 = 0.6$.

Define the function and its derivative:

$$f(x) = x^4 - 6.75x^2 + 6.25x - 1.5$$

$$f'(x) = 4x^3 - 13.5x + 6.25$$

Standard Newton-Raphson Method ($m = 1$)

Iteration (n)	x_n	$f(x_n)$	$f'(x_n)$	$x_{n+1} = x_n - f/f'$
0	0.6	-0.0504	-0.986	0.5489
1	0.5489	-0.00667	-0.586	0.5375
2	0.5375	-0.00167	-0.437	0.5337
3	0.5337	-0.000416	-0.362	0.5325
4	0.5325	-0.000104	-0.322	**0.5322**

The method is converging linearly to $\alpha = 0.5$ (the true root). The error is reducing slowly. After 4 iterations, it's still at 0.5322.

Modified Newton-Raphson Method ($m = 2$)

Iteration (n)	x_n	$f(x_n)$	$f'(x_n)$	$x_{n+1} = x_n - 2 \cdot f/f'$
0	0.6	-0.0504	-0.986	0.6 - 2*(0.0511) = **0.4978**
1	0.4978	∼0.0000	∼0.0000	∼0.5

Explanation of Iteration 0:

1. $f(0.6) = (0.6)^4 - 6.75(0.6)^2 + 6.25(0.6) - 1.5 = 0.1296 - 2.43 + 3.75 - 1.5 = -0.0504$
2. $f'(0.6) = 4(0.6)^3 - 13.5(0.6) + 6.25 = 4(0.216) - 8.1 + 6.25 = 0.864 - 8.1 + 6.25 = -0.986$

3. $\frac{f}{f'} = \frac{-0.0504}{-0.986} \approx 0.0511$

4. $x_1 = 0.6 - 2 \cdot (0.0511) = 0.6 - 0.1022 = 0.4978$

The modified method jumps to **0.4978** in a single step, landing almost directly on the true root $\alpha = 0.5$. A second iteration would refine this to an even more accurate value. This demonstrates the restored **quadratic convergence**.

What if the Multiplicity m is Unknown?

You can still use a modified method! One common approach is to use the following iteration, which doesn't require knowing m:

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - f(x_n)f''(x_n)},$$

where $g(x) = f(x)/f'(x)$. This formula also achieves quadratic convergence for multiple roots but requires calculating the second derivative $f''(x)$.

Summary

Feature	**Standard Newton-Raphson**	**Modified Newton-Raphson**
Root Type	Simple Roots ($m = 1$)	Multiple Roots ($m > 1$)
Convergence	Quadratic	Quadratic
Iteration Formula	$x_n - \dfrac{f(x_n)}{f'(x_n)}$	$x_n - m \cdot \dfrac{f(x_n)}{f'(x_n)}$
Requirement	$f'(x) \neq 0$ at root	Know multiplicity m

The Modified Newton-Raphson method is a crucial tool for efficiently solving equations where roots are not simple, ensuring fast convergence where the standard method would be unacceptably slow.