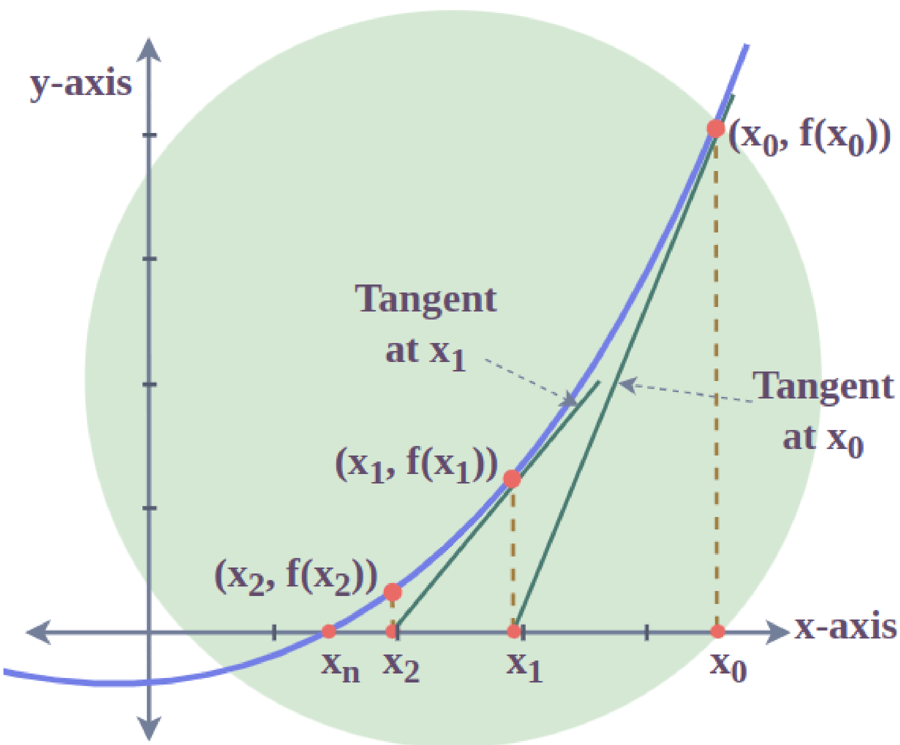
Newton-Raphson Method

The **Newton-Raphson Method** is a different method to find approximate roots. The method requires you to **differentiate** the equation you're trying to find a root of, so before revising this topic you may want to look back at **differentiation** to refresh your mind.

- It is a numerical technique for approximating the roots of real-valued functions.
- It starts with initial guess of root and iteratively refines the result using a formula that involves derivative of the function.
- Compared to other root-finding methods like bisection and secant methods, the Newton-Raphson method stands out due to its significantly faster convergence rate (quadratic while other have linear).
- It requires computation of derivative and preferred over other methods when this computation easier and we can find good estimate of root.

Newton Raphson Method Calculation
Assume the equation or functions whose roots are to be calculated as f(x) = 0.
In order to prove the validity of Newton Raphson method following steps are followed:
Step 1: Draw a graph of f(x) for different values of x as shown below:



Step 2: A tangent is drawn to f(x) at x0. This is the initial value.

Step 3: This tangent will intersect the X- axis at some fixed point (x1, 0) if the first derivative of f(x) is not zero i.e. **f'(x0) ≠ 0.**

Step 4: As this method assumes iteration of roots, this x1 is considered to be the next approximation of the root.

Step 5: Now steps 2 to 4 are repeated until we reach the actual root x.

Now we know that the slope-intercept equation of any line is represented as
y = mx + c,
Where **m** is the slope of the line and **c** is the x-intercept of the line.
Using the same formula we, get
y = f(x0) + f'(x0) (x - x0)
Here f (x0) represents the c and f' (x0) represents the slope of the tangent m. As this equation holds true for every value of x, it must hold true for x1. Thus, substituting x with x1, and equating the equation to zero as we need to calculate the roots, we get:
0 = f(x0) + f'(x0) (x1 - x0)

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

Which is the Newton Raphson method formula.

Thus, Newton Raphson's method was mathematically proved and accepted to be valid.

Newton's Method Algorithm

1. We choose an initial guess value for the new root, say x0.

2. Calculate $f(x_0)$.

3. Calculate $f'(x_0)$.

4. We calculate the new approximation value of the root from the equation:

$$x_1 = x_0 - f(x_0)/ f'(x_0)$$

5. Print the value of x1. 6. If $|x_1 - x_0| < \varepsilon$, then $x1$ is the new root, go to the last step.

7. $x_0 = x_1$, go to step 2.

8. Stop.

Example:

It costs a firm $C(q)$ dollars to produce q grams per day of a certain chemical, where

$$C(q) = 1000 + 2q + 3q^{2/3}$$

The firm can sell any amount of the chemical at \$4 a gram. Find the break-even point of the firm, that is, how much it should produce per day in order to have neither a profit nor a loss. Use the Newton Method and give the answer to the nearest gram.

Find a root of the equation $x^2 - 8x + 11 = 0$ to 5 decimal places using $x_0 = 6$

First we need to **differentiate** $f(x) = x^2 - 8x + 11$:

$$f'(x) = 2x - 8$$

Substituting this into the **Newton-Raphson formula**:

$$x_{n+1} = x_n - \frac{x^2 - 8x + 11}{2x - 8}$$

Starting with $x_0 = 6$:

$$x_1 = 6 - \frac{6^2 - 8(6) + 11}{2(6) - 8} = 6.25$$

Using the formula again to find the following iterations:

$$x_2 = 6.25 - \frac{6.25^2 - 8(6.25) + 11}{2(6.25) - 8} = 6.236111111$$

$$x_2 = 6.25 - \frac{6.25^2 - 8(6.25) + 11}{2(6.25) - 8} = 6.236111111$$

$$x_3 = 6.236111111 - \frac{6.236111111^2 - 8(6.236111111) + 11}{2(6.236111111) - 8} = 6.236067978$$

$$x_4 = 6.236067978 - \frac{6.236067978^2 - 8(6.236067978) + 11}{2(6.236067978) - 8} = 6.236067977$$

Thus a root of $x^2 - 8x + 11 = 0$ is 6.23607 to 5 decimal places.

Example 2.16 Find a root of the equation $x \sin x + \cos x = 0$.

We have

$$f(x) = x \sin x + \cos x \quad \text{and} \quad f'(x) = x \cos x.$$

The iteration formula is, therefore,

$$x_{n+1} = x_n - \frac{x_n \sin x_n + \cos x_n}{x_n \cos x_n}.$$

With $x_0 = \pi$, the successive iterates are given below

n	x_n	$f(x_n)$	x_{n+1}
0	3.1416	–1.0	2.8233
1	2.8233	–0.0662	2.7986
2	2.7986	–0.0006	2.7984
3	2.7984	0.0	2.7984

Problem: Find the root of $f(x) = x^4 - 8x^2 + 16$ using the Newton-Raphson method starting with $x_0 = 2.5$.

Problem: Find the root of $f(x) = x\sin(x) - 1$ using the Newton-Raphson method starting with $x_0 = 1$.

Problem 4: Find the root of $f(x) = e^x - 3x$ using the Newton-Raphson method starting with $x_0 = 1$.

Convergence of Newton-Raphson Method

Let x_n be an estimate of a root of the function $f(x)$. If x_n and x_{n+1} are close to each other, then, using Taylor's series expansion, we can state

$$f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{f''(R)}{2}(x_{n+1} - x_n)^2 \tag{6.22}$$

where R lies somewhere in the interval x_n to x_{n+1} and third and higher order have been dropped.

Let us assume that the exact root of $f(x)$ is x_r. Then $x_{n+1} = x_r$. Therefore $f(x_{n+1}) = 0$ and substituting these values in equation (6.22), we get

$$0 = f(x_n) + f'(x_n)(x_r - x_n) + \frac{f''(R)}{2}(x_r - x_n)^2 \tag{6.23}$$

We know that the Newton's iterative formula is given by

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

Rearranging the terms, we get

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

Substituting this for $f(x_n)$ in Eq. (6.23) yields

$$0 = f'(x_n)(x_r - x_{n+1}) + \frac{f''(R)}{2}(x_r - x_n)^2 \tag{6.24}$$

We know that the error in the estimate x_{n+1} is given by

$$e_{n+1} = x_r - x_{n+1}$$

Similarly,

$$e_n = x_r - x_n$$

Now, equation (6.24) can be expressed in terms of these errors as

$$0 = f'(x_n)\, e_{n+1} + \frac{f''(R)}{2} e_n^2$$

Rearranging the terms we get,

$$\boxed{e_{n+1} = -\frac{f''(R)}{2f'(x_n)} e_n^2} \tag{6.25}$$

Equation (6.25) shows that the error is roughly proportional to the square of the error in the previous iteration. Therefore, the Newton-Raphson method is said to have *quadratic convergence*.

Comparison of Root-Finding Methods

Feature / Criterion	**Bisection Method**	**False Position (Regula Falsi)**	**Secant Method**	**Newton–Raphson Method**
Type of Method	Bracketing	Bracketing	Open	Open
Requires Derivative?	No	No	No	Yes ($f'(x)$)
Initial Requirements	Two points (a, b) such that $f(a) \times f(b) < 0$	Two points (a, b) such that $f(a) \times f(b) < 0$	Two initial guesses x_0, x_1	One initial guess x_0
Convergence Speed	Slow (linear)	Faster than bisection, but still linear	Faster than false position (super-linear)	Very fast (quadratic)
Guaranteed Root?	Yes (if f(a) and f(b) have opposite signs)	Yes	Not guaranteed	Not guaranteed
Dependence on Initial Guess	Very low	Low	High	Very high
Stability	Very stable	Stable	Less stable	May diverge easily
Computational Cost per Iteration	Low	Low	Low	Higher (requires evaluating derivative)
When Interval Changes?	Always halves the interval	Changes only one side → sometimes slow	Not bracketing → no guaranteed interval	No interval concept
Best Use Case	When guaranteed convergence is needed	When bracketing is available but want faster convergence	When derivative is unavailable but good initial estimates exist	When derivative exists and fast convergence is needed
Worst Case Issue	Very slow	Can become extremely slow for certain shapes of f(x)	Can diverge if guesses are poor	Fails if $f'(x)=0$ or near zero, or if initial guess is bad

Short Summary

Bisection	False Position	Secant	Newton–Raphson
• Slowest method • Safest and always converges • Uses interval halving	• A bit faster than bisection • Still guaranteed • One endpoint often stays fixed → can slow down	• Faster than false position • Does not need derivative • Can fail if initial guesses are poor	• Fastest (quadratic convergence) • Requires derivative • Can diverge if initial guess is bad

Rank by Speed

Newton–Raphson > Secant > False Position > Bisection

Rank by Reliability

Bisection > False Position > Secant > Newton–Raphson