Newton-Raphson Method

NC 10 • Module 2: Solution of nonlinear equations

Central question	How can tangent lines be used to approximate a root of f(x)=0?
Lecture focus	Derivation, graphical interpretation, iteration tables and convergence.
Prerequisite	Root isolation, derivatives and fixed-point iteration (NC 06–09).

Learning outcomes

By the end of this lecture, students should be able to...

Derive	Obtain the Newton formula from the tangent line or Taylor expansion.
Compute	Solve nonlinear equations using numerical iteration tables.
Evaluate	Recognize quadratic convergence, failure modes and stopping criteria.

Basic idea

Let f be differentiable near a root r, where f(r)=0.

Initial estimate	Choose x_0 near a root, for example by graphical root isolation.
Approximate	Replace the curve locally by its tangent at the current estimate.
Update	Use the tangent’s x-intercept as the next approximation.

Geometric interpretation

A tangent line predicts the next root estimate.

Current point

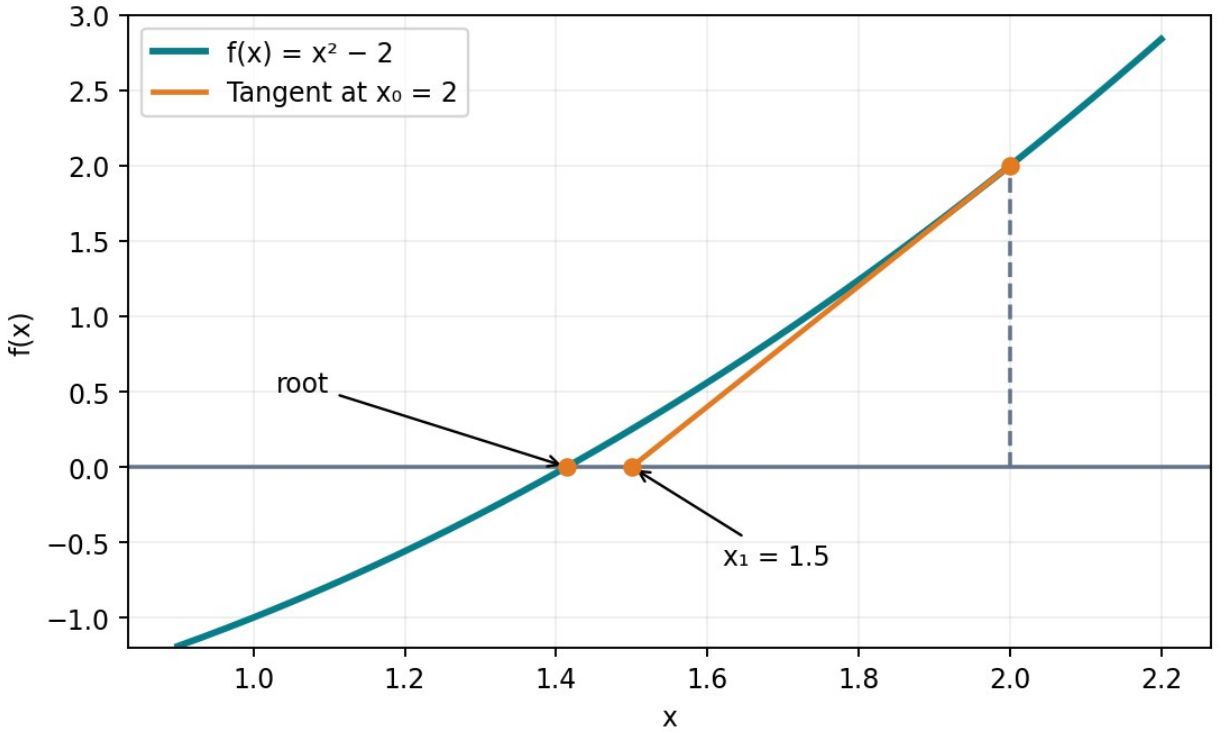
Locate $(x_n, f(x_n))$ on the curve.

Tangent

Draw a line with slope $f'(x_n)$ through that point.

Next estimate

Its intersection with the x-axis is x_{n+1}. Repeat.



Derivation from tangent line

The tangent through $(x_n, f(x_n))$ has equation:

Line equation

$$y - f(x_n) = f'(x_n)(x - x_n).$$

At the intercept

Set $y=0$ and $x=x_{n+1}$, then rearrange.

Newton formula

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

Taylor series interpretation

For f twice continuously differentiable near x_n:

Expansion

$$f(r) = f(x_n) + f'(x_n)(r - x_n) + O((r - x_n)^2).$$

Linearization

Since $f(r)=0$, neglect the second-order remainder.

Approximation

$r \approx x_n - f(x_n)/f'(x_n)$. Use this as the next iterate.

Algorithm

Inputs: f, f′, initial guess, tolerances and maximum iterations.

Step 1	Evaluate f(x); return x if it meets the residual criterion.
Step 2	Check the derivative; compute $y=x-f(x)/f'(x)$.
Step 3	Check finite values and convergence; otherwise set x=y and repeat up to the limit.

Example 1: square root of 2

Solve $x^2-2=0$ for its positive root.

Function	$f(x)=x^2-2$; $f'(x)=2x$; choose $x_0=1$.
Iteration	$x_{n+1} = (x_n + 2/x_n)/2$.
First steps	$x_1=1-(-1)/2=1.5$; $x_2=1.5-0.25/3\approx1.41666667$.

Example 1: iteration table

Columns in each row: n, $x_n \to x_{n+1}$, absolute step. Rounded for display.

Steps 0–1	0: 1.00000000 → 1.50000000; $\lvert\Delta x\rvert$=0.5 1: 1.50000000 → 1.41666667; $\lvert\Delta x\rvert$=0.0833
Steps 2–3	2: 1.41666667 → 1.41421569; $\lvert\Delta x\rvert$=0.00245 3: 1.41421569 → 1.41421356; $\lvert\Delta x\rvert$=2.12e-06
Steps 4–5	4: 1.41421356 → 1.41421356; $\lvert\Delta x\rvert$=1.59e-12 5: 1.41421356 → 1.41421356; $\lvert\Delta x\rvert$=2.22e-16

Example 2: transcendental equation

Solve $\cos(x)-x=0$; angles are in radians.

Function	$f(x)=\cos(x)-x$; $f'(x)=-\sin(x)-1$.
Initial guess	$x_0=0.5$; $f(0.5)\approx 0.37758256$ and $f'(0.5)\approx -1.47942554$.
First update	$x_1\approx 0.5-0.37758256/(-1.47942554)\approx 0.75522242$.

Example 2: iteration table

Newton estimates approach r≈0.7390851332.

Steps 0–1	0: 0.50000000 → 0.75522242; |Δx|=0.255 1: 0.75522242 → 0.73914167; |Δx|=0.0161
Steps 2–3	2: 0.73914167 → 0.73908513; |Δx|=5.65e-05 3: 0.73908513 → 0.73908513; |Δx|=7.06e-10
Verification	Substituting r into cos(r)−r gives a residual near zero.

Quadratic convergence

Suppose r is a simple root, $f'(r) \neq 0$, and f is C^2 near r.

Local error	Define $e_n = x_n - r$. For sufficiently close initial values, Newton converges locally.
Asymptotic formula	$e_{n+1} = [f''(r)/(2f'(r))]\, e_n^2 + o(e_n^2)$.
Meaning	If $f''(r) \neq 0$, convergence is quadratic: the error is approximately squared each step.

Multiple roots

Ordinary Newton iteration may lose quadratic convergence.

Example	$f(x)=(x-2)^2$; $f'(x)=2(x-2)$.
Update	$x_{n+1}=(x_n+2)/2$, so $e_{n+1}=e_n/2$: linear convergence.
Known multiplicity m	Modified Newton: $x_{n+1}=x_n-m\,f(x_n)/f'(x_n)$; under smoothness assumptions it restores at least quadratic convergence.

Failure modes and safeguards

Newton is an open method: it does not maintain a root bracket.

Zero derivative	$f(x)=x^3-1$, $x_0=0$: derivative vanishes while $f(x_0)\neq 0$.
Cycle	$f(x)=x^3-2x+2$; starting at 0 gives $0 \to 1 \to 0$.
Safeguard	Maintain a sign-changing bracket and use bisection when a Newton step is unsafe.

Stopping criteria and comparison

Use both numerical checks and a maximum iteration count.

Step and residual	Test $\lvert y-x\rvert \le \text{atol}+\text{rtol}\lvert y\rvert$ and $\lvert f(y)\rvert \le \text{ftol}$. Neither alone certifies root accuracy.
Error certificate	If $\lvert f'\rvert \ge m>0$ between y and a root r, then $\lvert y-r\rvert \le \lvert f(y)\rvert/m$.
Comparison	Bisection has a guaranteed bracket bound; Newton is often faster locally but requires derivatives.

Python implementation

Example: newton(lambda x:x*x-2, lambda x:2*x, 1.0).

Setup

```
import math
def newton(f, df, x, tol=1e-8, limit=100):
    for _ in range(limit):
```

Update

```
        if f(x) == 0: return x
        d = df(x)
        if abs(d) < 1e-14: raise ValueError("small slope")
        y = x - f(x)/d
```

Stop / repeat

```
        if not math.isfinite(y): raise ValueError("nonfinite")
        if abs(y-x)<=tol and abs(f(y))<=tol: return y
        x = y
    raise RuntimeError("iteration limit")
```

Practice questions

Attempt the questions before viewing the solutions.

Q1 Apply two Newton steps to $x^2-5=0$ from $x_0=2$.

Q2 Derive Newton's iteration for $f(x)=x^3-a$, $x\neq 0$.

Q3 Explain why $f(x)=x^3-1$ with $x_0=0$ cannot take a Newton step.

Solutions and recap

Next: NC 11 — Secant Method.

A1	$x_1=2.25$; $x_2=(2.25+5/2.25)/2\approx 2.23611111$.
A2	$x_{n+1}=(2x_n+a/x_n^2)/3$.
A3 / Takeaway	$f'(0)=0$ and $f(0)=-1$. Newton needs a usable derivative and an appropriate starting point.