MAT 103: Numerical Analysis I

Topic 3: Solutions of Scalar Nonlinear Equations

Dr. Anna Fome

2026-05-20

Table of contents

"The purpose of computing is insight, not numbers." — Richard Hamming, Mathematician

---

1 Introduction

In Topics 1 and 2 we established *why* numerical methods are needed and *how* errors arise. In this topic we put those ideas to work on one of the most common problems in mathematics and engineering: **finding the roots of an equation**.

Consider these situations:

- A chemical engineer needs to find the temperature T at which the reaction rate $k(T) = e^{-T/10} - 0.3T$ equals zero — that is, when $e^{-T/10} = 0.3T$.
- A structural engineer needs the buckling load P of a column, satisfying $\tan(\sqrt{P}) = \sqrt{P}$.

- A financial analyst needs the interest rate r satisfying $(1+r)^{12} - 1.1 = 0$.

In every case, the problem reduces to: **find** x such that $f(x) = 0$.

For simple functions (linear, quadratic), we have exact formulas. For the more complex, transcendental, or higher-degree cases that arise in practice, we need **numerical root-finding methods**.

By the end of this topic, you should be able to:

- Reformulate a root-finding problem as $f(x) = 0$ and locate roots graphically.
- Apply fixed-point iteration and analyse its convergence.
- Apply the bisection, secant, Regula Falsi, and Newton–Raphson methods.
- Apply Newton's method to polynomials using Horner's scheme.
- Understand the principle behind Bairstow's method for polynomial roots.
- Use MATLAB or MAPLE to find roots computationally.

---

2 Finding Roots of $f(x) = 0$

2.1 Definition

> **Definition**
>
> A **root** (or **zero**) of a function $f(x)$ is a value $x = x^*$ such that:
>
> $$f(x^*) = 0$$

Reformulating any equation as $f(x) = 0$:

Any equation can be written in the form $f(x) = 0$ by moving everything to the left side.

Original Equation	Reformulation	$f(x)$
$e^x = 3x + 1$	$e^x - 3x - 1 = 0$	$f(x) = e^x - 3x - 1$
$x^3 = 2x + 5$	$x^3 - 2x - 5 = 0$	$f(x) = x^3 - 2x - 5$
$\sin(x) = x/2$	$\sin(x) - x/2 = 0$	$f(x) = \sin(x) - x/2$
$(1+r)^{12} = 1.1$	$(1+r)^{12} - 1.1 = 0$	$f(r) = (1+r)^{12} - 1.1$

2.2 Locating Roots Graphically

Before applying any numerical method, it is good practice to **sketch** $f(x)$ or plot $y = f(x)$ and look for where the curve crosses the x-axis. This gives us:

1. An estimate of how many roots exist.

2. Approximate locations of the roots.
3. A good initial guess for iterative methods.

The **Intermediate Value Theorem (IVT)** gives us a rigorous way to confirm a root exists:

> **Intermediate Value Theorem (IVT)**
>
> If f is **continuous** on $[a, b]$ and $f(a)$ and $f(b)$ have **opposite signs** (i.e., $f(a) \cdot f(b) < 0$), then there exists at least one root $x^* \in (a, b)$ such that $f(x^*) = 0$.

Solved Example 3.1 — Locating a root

Show that $f(x) = x^3 - 4x + 1$ has a root in each of the intervals $[-3, -2]$, $[0, 1]$, and $[1, 2]$.

Solution:

Evaluate f at the endpoints of each interval:

Interval	f(left end)	f(right end)	Sign change?	Conclusion
$[-3, -2]$	$f(-3) = -27+12+1 = -14 < 0$	$f(-2) = -8 + 8 + 1 = 1 > 0$	Yes	Root in $(-3, -2)$
$[0, 1]$	$f(0) = 1 > 0$	$f(1) = 1-4+1 = -2 < 0$	Yes	Root in $(0, 1)$
$[1, 2]$	$f(1) = -2 < 0$	$f(2) = 8 - 8 + 1 = 1 > 0$	Yes	Root in $(1, 2)$

Since f is a polynomial (continuous everywhere), the IVT guarantees a root in each interval. This function has exactly three real roots.

---

3 General Iteration Methods (Fixed-Point Iteration)

3.1 The Idea

The **fixed-point iteration** method (also called *successive substitution*) converts the equation $f(x) = 0$ into the equivalent form:

$$x = g(x)$$

We call x^* a **fixed point** of g if $g(x^*) = x^*$. Clearly, if x^* is a fixed point of g and g was obtained from f, then $f(x^*) = 0$.

Starting from an initial guess x_0, we generate the sequence:

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

If this sequence converges, it converges to a fixed point x^* which is a root of $f(x) = 0$.

3.2 How to Rearrange $f(x) = 0$ into $x = g(x)$

There are usually **many ways** to rearrange $f(x) = 0$ as $x = g(x)$. Some rearrangements lead to convergence; others lead to divergence.

Example: $f(x) = x^2 - 3x + 2 = 0$ (roots: $x = 1$ and $x = 2$)

Rearrangement	$g(x)$	Converges to?
$x = \frac{x^2+2}{3}$	$g_1(x) = \frac{x^2+2}{3}$	$x = 1$ (from x_0 near 1)
$x = 3 - \frac{2}{x}$	$g_2(x) = 3 - \frac{2}{x}$	$x = 2$ (from x_0 near 2)
$x = x^2 - 2x + 2$	$g_3(x) = x^2 - 2x + 2$	$x = 1$ (from x_0 near 1)

The choice of g is critical. The convergence condition tells us which choices work.

3.3 Convergence Condition for Fixed-Point Iteration

> **Convergence Theorem**
>
> Let g be continuous and differentiable on an interval $[a, b]$ containing the fixed point x^*. If:
>
> $$|g'(x)| \leq K < 1 \quad \text{for all } x \in [a, b]$$
>
> for some constant $K < 1$, then the iteration $x_{n+1} = g(x_n)$ **converges** to x^* for any starting value $x_0 \in [a, b]$.

Interpretation: $|g'(x^*)|$ is called the **asymptotic error constant**.

- If $|g'(x^*)| < 1$: iteration **converges** — the smaller $|g'|$, the faster.
- If $|g'(x^*)| > 1$: iteration **diverges**.
- If $|g'(x^*)| = 1$: the test is inconclusive.

Solved Example 3.2 — Checking convergence

For $f(x) = x^2 - 3x + 2 = 0$, check whether $g_1(x) = \frac{x^2+2}{3}$ converges near $x^* = 1$.

Solution:

$$g_1'(x) = \frac{2x}{3}$$

At $x^* = 1$: $|g_1'(1)| = \frac{2}{3} < 1$

The condition is satisfied, so the iteration **converges** near $x^* = 1$.

Now check $g_3(x) = x^2 - 2x + 2$ near $x^* = 1$:

$$g_3'(x) = 2x - 2$$

At $x^* = 1$: $|g_3'(1)| = |2 - 2| = 0 < 1$

Try $x_0 = 1.5$:

$$x_1 = (1.5)^2 - 2(1.5) + 2 = 2.25 - 3 + 2 = 1.25$$
$$x_2 = (1.25)^2 - 2(1.25) + 2 = 1.5625 - 2.5 + 2 = 1.0625$$
$$x_3 = (1.0625)^2 - 2(1.0625) + 2 = 1.1289 - 2.125 + 2 = 1.0039$$

It does converge here! The convergence theorem's condition is checked at the root — and at $x^* = 1$ the derivative is exactly 0, giving very fast convergence.

3.4 Algorithm: Fixed-Point Iteration

Given x_0 (initial guess), tolerance $\varepsilon > 0$, and maximum iterations N:

1. For $n = 0, 1, 2, \ldots, N$:
   a. Compute $x_{n+1} = g(x_n)$
   b. If $|x_{n+1} - x_n| < \varepsilon$: **stop** — x_{n+1} is the root.
2. If the loop completes without stopping: the method has not converged in N steps.

3.5 Solved Example 3.3 — Full fixed-point iteration

Solve $f(x) = e^{-x} - x = 0$ using fixed-point iteration.

Rearrange: $x = e^{-x}$, so $g(x) = e^{-x}$.

Check convergence: $g'(x) = -e^{-x}$. At the root $x^* \approx 0.567$: $|g'(0.567)| = e^{-0.567} \approx 0.567 < 1$

Start with $x_0 = 0$, tolerance $\varepsilon = 0.0001$:

n	x_n	$x_{n+1} = e^{-x_n}$	$\|x_{n+1} - x_n\|$
0	0.000000	1.000000	1.000000
1	1.000000	0.367879	0.632121
2	0.367879	0.692201	0.324322
3	0.692201	0.500474	0.191727
4	0.500474	0.606531	0.106057
5	0.606531	0.545239	0.061292
6	0.545239	0.579703	0.034464
7	0.579703	0.560115	0.019588
8	0.560115	0.571143	0.011028
9	0.571143	0.564879	0.006264
10	0.564879	0.568429	0.003550
15	0.567560	0.566872	0.000688
20	0.567176	0.567370	0.000194
24	0.567296	0.567189	0.000107
25	0.567189	0.567250	**0.000061** $< \varepsilon$

Root: $x^* \approx 0.5672$ (the true value is 0.56714...)

The method converges but slowly (linear convergence), taking about 25 iterations. This motivates the faster methods we study next.

---

4 Convergence Criterion and Order of Convergence

4.1 Stopping Criteria

We stop an iterative method when one of the following conditions is met:

Criterion 1 — Successive approximations are close:

$$|x_{n+1} - x_n| < \varepsilon_1$$

Criterion 2 — The function value is close to zero:

$$|f(x_{n+1})| < \varepsilon_2$$

Criterion 3 — Relative change is small:

$$\frac{|x_{n+1} - x_n|}{|x_{n+1}|} < \varepsilon_3, \quad x_{n+1} \neq 0$$

In practice, Criterion 3 is preferred because it accounts for the scale of the answer. For example, an absolute change of 0.001 is small when $x \approx 1000$, but large when $x \approx 0.001$.

4.2 Order of Convergence

The **order of convergence** p of an iterative method describes how quickly the error decreases from one iteration to the next.

Let $e_n = x_n - x^*$ be the error at step n. The method has **order** p if:

$$\lim_{n \to \infty} \frac{|e_{n+1}|}{|e_n|^p} = C, \quad C > 0$$

where C is called the **asymptotic error constant**.

For large n, the definition means:

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

The order p determines *how fast* convergence happens while the constant C affects speed.

- If $p = 1$: error shrinks by a constant factor (slow)
- If $p > 1$: faster than linear convergence (higher p usually means dramatically faster convergence.)

4.2.1 How to Estimate p and C in Practice

In real problems, we do not know the exact solution x^*, but we can estimate:

1. Compute approximate errors:

$$e_n \approx |x_n - x_{n-1}|$$

2. Estimate the order:

$$p \approx \frac{\ln\left(\frac{e_{n+1}}{e_n}\right)}{\ln\left(\frac{e_n}{e_{n-1}}\right)}$$

3. Estimate the constant:

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

Example (Estimation)

Suppose:

$$e_1 = 0.1, \quad e_2 = 0.01, \quad e_3 = 0.0001, \quad \cdots$$

Then:

$$e_2 = (e_1)^2, \quad e_3 = (e_2)^2, \cdots$$

- So:

$$p = 2, \quad C = 1$$

This is **quadratic convergence**

---

5 The Bisection Method

5.1 Motivation and Principle

The bisection method is the simplest and most robust root-finding method. It is based directly on the **Intermediate Value Theorem**: if $f(a) \cdot f(b) < 0$, a root exists in (a, b). The method repeatedly **halves** the interval, discarding the half that does not contain the root. This is why the method is also called a **binary search method**.

5.2 Algorithm

Given: f, interval $[a, b]$ with $f(a) \cdot f(b) < 0$, tolerance ε

1. Compute midpoint: $c = \dfrac{a+b}{2}$
2. If $|b - a| < \varepsilon$ or $|f(c)| < \varepsilon$: **stop**, c is the root.
3. If $f(a) \cdot f(c) < 0$: set $b = c$ (root is in left half $[a, c]$). Else: set $a = c$ (root is in right half $[c, b]$).
4. Go to step 1.

5.3 Error Bound

After n bisection steps, the error satisfies:

$$|x_n - x^*| \leq \frac{b-a}{2^n}$$

- Accuracy improves **slowly but steadily** i.e the error decreases by a factor of **1/2 each step**.

- Convergence is **guaranteed** i.e Method is reliable but not efficient for high precision

Number of iterations needed for tolerance ε:

$$n \geq \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2}$$

This tells us exactly how many iterations we need for a given accuracy.

5.4 Order of Convergence

The bisection method has **linear convergence**.

From:

$$|e_{n+1}| \approx \frac{1}{2}|e_n|$$

we identify:

- Order: $p = 1$ (linear)
- Constant: $C = \frac{1}{2}$

5.5 Solved Example

Example 3.4

Find the root of $f(x) = x^3 - x - 2 = 0$ in $[1, 2]$ correct to 2 decimal places , $\varepsilon = 0.005$.

Step 0: Verify bracket. $f(1) = 1-1-2 = -2 < 0$ and $f(2) = 8-2-2 = 4 > 0$, therefore, $f(1) \cdot f(2) < 0$

The interval width is $b - a = 1$. Number of steps we need:

$$n \geq \frac{\ln(1/0.005)}{\ln 2} = \frac{\ln 200}{\ln 2} \approx \frac{5.298}{0.693} \approx 7.6 \implies n = 8 \text{ iterations}$$

Step	a	b	$c = \frac{a+b}{2}$	$f(c)$	New interval
1	1.0000	2.0000	1.5000	$-0.125 < 0$	$[1.5, 2.0]$
2	1.5000	2.0000	1.7500	$1.609 > 0$	$[1.5, 1.75]$
3	1.5000	1.7500	1.6250	$0.666 > 0$	$[1.5, 1.625]$
4	1.5000	1.6250	1.5625	$0.252 > 0$	$[1.5, 1.5625]$
5	1.5000	1.5625	1.5313	$0.059 > 0$	$[1.5, 1.5313]$
6	1.5000	1.5313	1.5156	$-0.034 < 0$	$[1.5156, 1.5313]$
7	1.5156	1.5313	1.5234	$0.013 > 0$	$[1.5156, 1.5234]$
8	1.5156	1.5234	1.5195	$-0.011 < 0$	$[1.5195, 1.5234]$

After 8 steps: root ≈ 1.52, interval width $= 0.0039 < 0.005$

True root: $x^* = 1.5214$ (to 4 d.p.)

Verification: $f(1.52) = (1.52)^3 - 1.52 - 2 = 3.512 - 3.52 = -0.008 \approx 0$

Example 3.5

How many bisection steps are needed to find a root of $f(x) = \cos(x) - x$ in $[0, 1]$ accurate to 5 decimal places (i.e., $\varepsilon = 5 \times 10^{-6}$)?

$$n \geq \frac{\ln\left(\frac{1-0}{5\times10^{-6}}\right)}{\ln 2} = \frac{\ln(200000)}{\ln 2} = \frac{12.206}{0.693} \approx 17.6$$

Answer: 18 iterations.

---

6 Newton–Raphson Method

6.1 Motivation and Derivation

The Newton–Raphson method is the most widely used root-finding method in practice. It converges **quadratically** — the number of correct digits roughly doubles with each iteration. We derive it from the Taylor series.

Derivation via Taylor series:

Suppose x_n is our current approximation to the root x^*. Write $x^* = x_n + h$ where h is a small unknown correction.

Expand $f(x^*)$ in a Taylor series around x_n:

$$f(x^*) = f(x_n) + hf'(x_n) + \frac{h^2}{2}f''(x_n) + \cdots$$

Since $f(x^*) = 0$ and h is small, drop the h^2 and higher terms:

$$0 \approx f(x_n) + hf'(x_n)$$

Solve for h:

$$h \approx -\frac{f(x_n)}{f'(x_n)}$$

So the improved approximation is:

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

> **i** Newton–Raphson Formula
>
> $$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}}$$
>
> Geometrically: x_{n+1} is the x-intercept of the **tangent line** to $y = f(x)$

at the point $(x_n, f(x_n))$.

6.2 Geometric Interpretation

The tangent line to f at $(x_n, f(x_n))$ has equation:

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

Setting $y = 0$: $x = x_n - \frac{f(x_n)}{f'(x_n)} = x_{n+1}$

Starting from x_0, we draw the tangent at x_0, find where it hits the x-axis (x_1), draw the tangent at x_1, find x_2, and so on.

6.3 Algorithm

Given: f, f', initial guess x_0, tolerance ε, max iterations N

1. For $n = 0, 1, 2, \ldots, N$:
   a. If $|f'(x_n)| \approx 0$: **stop** — method fails (zero derivative).
   b. Compute $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
   c. If $|x_{n+1} - x_n| < \varepsilon$: **stop** — x_{n+1} is the root.

6.4 Solved Example 3.8 — Newton–Raphson, step by step

Find the root of $f(x) = x^3 - x - 2 = 0$ near $x_0 = 2$.

$f(x) = x^3 - x - 2$ and $f'(x) = 3x^2 - 1$.

Iteration 1: $n = 0$, $x_0 = 2$

$$f(2) = 8 - 2 - 2 = 4, \quad f'(2) = 3(4) - 1 = 11$$

$$x_1 = 2 - \frac{4}{11} = 2 - 0.36364 = 1.63636$$

Iteration 2: $n = 1$, $x_1 = 1.63636$

$$f(1.63636) = (1.63636)^3 - 1.63636 - 2 = 4.37828 - 1.63636 - 2 = 0.74192$$

$$f'(1.63636) = 3(1.63636)^2 - 1 = 3(2.67767) - 1 = 7.03301$$

$$x_2 = 1.63636 - \frac{0.74192}{7.03301} = 1.63636 - 0.10549 = 1.53087$$

Iteration 3: $n = 2$, $x_2 = 1.53087$

$$f(1.53087) = (1.53087)^3 - 1.53087 - 2 = 3.58827 - 1.53087 - 2 = 0.05740$$

$$f'(1.53087) = 3(1.53087)^2 - 1 = 3(2.34356) - 1 = 6.03068$$

$$x_3 = 1.53087 - \frac{0.05740}{6.03068} = 1.53087 - 0.00952 = 1.52135$$

Iteration 4: $n = 3$, $x_3 = 1.52135$

$$f(1.52135) = (1.52135)^3 - 1.52135 - 2 = 3.52377 - 1.52135 - 2 = 0.00242$$

$$f'(1.52135) = 3(1.52135)^2 - 1 = 3(2.31451) - 1 = 5.94352$$

$$x_4 = 1.52135 - \frac{0.00242}{5.94352} = 1.52135 - 0.000407 = 1.52094 \approx 1.52138$$

$|x_4 - x_3| = 0.000407 \approx 0$

Summary table:

n	x_n	$f(x_n)$	$f'(x_n)$	x_{n+1}	Error $\|x_n - x^*\|$
0	2.00000	4.00000	11.00000	1.63636	0.47862
1	1.63636	0.74192	7.03301	1.53087	0.10949
2	1.53087	0.05740	6.03068	1.52135	0.00949
3	1.52135	0.00242	5.94352	1.52095	0.00003
4	1.52095	≈ 0	—	—	≈ 0

Root: $x^* \approx 1.52138$

Observe the **quadratic convergence**: errors are $0.479, 0.109, 0.0095, 0.00003, ...$ Each error is approximately the **square** of the previous one (times a constant).

6.5 Solved Example 3.9 — Newton–Raphson for a transcendental equation

Find the root of $f(x) = e^x - 3x = 0$ near $x_0 = 0$.

$f'(x) = e^x - 3$.

Check: $f(0) = 1 - 0 = 1 > 0$, $f(1) = e - 3 = -0.282 < 0$. Root in $(0, 1)$.

n	x_n	$f(x_n) = e^{x_n} - 3x_n$	$f'(x_n) = e^{x_n} - 3$	x_{n+1}
0	0.0000	1.0000	−2.0000	0.5000
1	0.5000	0.1487	−1.3513	0.6101
2	0.6101	0.0048	−1.1591	0.6142
3	0.6142	0.000004	−1.1521	0.6142

Root: $x^* \approx 0.6142$ in just 3 iterations!

6.6 Quadratic Convergence — Why Does It Happen?

For Newton–Raphson, the error at step $n + 1$ satisfies:

$$e_{n+1} \approx \frac{f''(x^*)}{2f'(x^*)} \cdot e_n^2$$

This is the **quadratic convergence formula**. The error is proportional to e_n^2, so if $e_n = 0.01$, then $e_{n+1} \approx C \times 0.0001$ — two more decimal places in one step.

6.7 When Newton–Raphson Fails

Newton–Raphson can fail or behave poorly in these situations:

Case 1: $f'(x_n) = 0$ or $f'(x_n) \approx 0$

Division by zero (or near-zero) causes the method to break down.

Example: $f(x) = x^3 - 3x + 2$, $x_0 = 1$. Then $f'(1) = 3(1)^2 - 3 = 0$ — fails immediately.

Case 2: Poor initial guess

If x_0 is far from the root or in a region where f is flat or oscillatory, the tangent line can overshoot wildly.

Example: $f(x) = x^3 - 2x + 2$. Starting from $x_0 = 0$:

$$x_1 = 0 - \frac{2}{-2} = 1, \quad x_2 = 1 - \frac{1}{1} = 0, \quad x_3 = 1, \ldots$$

The method **cycles** between 0 and 1 indefinitely — it never converges!

Case 3: Multiple roots

If $f(x^*) = 0$ and $f'(x^*) = 0$ (a double root), Newton–Raphson still converges but only **linearly** (not quadratically).

> Practical advice
>
> Always plot $f(x)$ before applying Newton–Raphson. Choose x_0 close to the root and check that $f'(x_0) \neq 0$. If convergence seems slow or erratic, try a different starting value or switch to bisection first to get a good initial guess.

6.8 Properties of Newton–Raphson

Property	Details
Order of convergence	Quadratic ($p = 2$) near a simple root
Requires derivative?	Yes — $f'(x)$ must be computable
Requires bracket?	No
Speed	Very fast — typically 4–6 iterations
Risk	Can diverge for poor x_0; fails if $f'(x_n) = 0$

---

7 The Secant Method

7.1 Motivation

The Newton–Raphson method (Section 8) requires $f'(x)$ at each step — but sometimes, $f'(x)$ is difficult or expensive to compute.

- The **secant method** approximates $f'(x_n)$ using the **finite difference** of the two most recent iterates, avoiding the need for an analytical derivative.
- The secant line of a function $f(x)$ on interval $[a, b]$ is the straight line that passes through the two points on the graph: $(a, f(a))$ and $(b, f(b))$.
- Its slope is the **average rate of change** of the function over the interval:

$$m_{sec} = \frac{f(b) - f(a)}{b - a}$$

7.2 Derivation

In Newton–Raphson we use:

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

Replace $f'(x_n)$ with the secant slope through the last two points:

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

Substituting:

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

This is the **secant formula**. It requires two starting values x_0 and x_1, and at each step uses the line (secant) through $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$ to approximate the root.

7.3 Algorithm

Given: f, two initial guesses x_0 and x_1 (need not bracket the root), tolerance ε

1. For $n = 1, 2, 3, ...$:
   a. Compute $x_{n+1} = x_n - f(x_n) \cdot \dfrac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$
   b. If $|x_{n+1} - x_n| < \varepsilon$: **stop**.

Note: If $f(x_n) = f(x_{n-1})$, the denominator is zero — the method fails. In practice, this is rare but must be guarded against.

7.4 Solved Example 3.6 — Secant Method

Find the root of $f(x) = x^3 - x - 2 = 0$ using the secant method.

Start with $x_0 = 1$, $x_1 = 2$. Use tolerance $\varepsilon = 0.0001$.

Iteration 1: $n = 1$

$f(x_0) = f(1) = 1 - 1 - 2 = -2$

$f(x_1) = f(2) = 8 - 2 - 2 = 4$

$$x_2 = 2 - 4 \cdot \frac{2 - 1}{4 - (-2)} = 2 - \frac{4}{6} = 2 - 0.6667 = 1.3333$$

$f(x_2) = f(1.3333) = (1.3333)^3 - 1.3333 - 2 = 2.3704 - 1.3333 - 2 = -0.9629$

Iteration 2: $n = 2$

$$x_3 = 1.3333 - (-0.9629) \cdot \frac{1.3333 - 2}{-0.9629 - 4} = 1.3333 - (-0.9629) \cdot \frac{-0.6667}{-4.9629}$$

$$x_3 = 1.3333 - (-0.9629) \cdot (-0.13432) = 1.3333 - 0.12934 = 1.5262$$

Hmm — let me recompute carefully:

$$\frac{x_2 - x_1}{f(x_2) - f(x_1)} = \frac{1.3333 - 2}{-0.9629 - 4} = \frac{-0.6667}{-4.9629} = 0.13432$$

$$x_3 = 1.3333 - (-0.9629)(0.13432) = 1.3333 + 0.12934 = 1.4626$$

$f(1.4626) = (1.4626)^3 - 1.4626 - 2 = 3.1267 - 1.4626 - 2 = -0.3359$

Iteration 3: $n = 3$

$$x_4 = 1.4626 - (-0.3359) \cdot \frac{1.4626 - 1.3333}{-0.3359 - (-0.9629)}$$

$$= 1.4626 - (-0.3359) \cdot \frac{0.1293}{0.6270} = 1.4626 + 0.3359 \times 0.2062 = 1.4626 + 0.0693 = 1.5319$$

$f(1.5319) = (1.5319)^3 - 1.5319 - 2 = 3.5964 - 1.5319 - 2 = 0.0645$

Iteration 4: $n = 4$

$$x_5 = 1.5319 - 0.0645 \cdot \frac{1.5319 - 1.4626}{0.0645 - (-0.3359)}$$

$$= 1.5319 - 0.0645 \cdot \frac{0.0693}{0.4004} = 1.5319 - 0.0645 \times 0.1730 = 1.5319 - 0.01116 = 1.5207$$

$f(1.5207) = (1.5207)^3 - 1.5207 - 2 = 3.5196 - 1.5207 - 2 = -0.0011$

Iteration 5: $n = 5$

$$x_6 = 1.5207 - (-0.0011) \cdot \frac{1.5207 - 1.5319}{-0.0011 - 0.0645}$$

$$= 1.5207 - (-0.0011) \cdot \frac{-0.0112}{-0.0656} = 1.5207 - (-0.0011)(-0.1707) = 1.5207 - 0.000188 = 1.5207$$

$|x_6 - x_5| \approx 0.0000 < \varepsilon$

Summary table:

n	x_n	$f(x_n)$	$\|x_{n+1} - x_n\|$
0	1.0000	−2.0000	—
1	2.0000	4.0000	—
2	1.3333	−0.9629	0.6667
3	1.4626	−0.3359	0.1293
4	1.5319	0.0645	0.0693
5	1.5207	−0.0011	0.0112
6	1.5214	≈ 0	0.0007

Root: $x^* \approx 1.5214$ (true value: 1.52138...) — found in just 6 steps!

7.5 Properties of the Secant Method

Property	Details
Order of convergence	Superlinear, $p \approx 1.618$ (golden ratio)
Requires derivative?	No — uses two function evaluations per step
Requires bracket?	No — may converge from outside the root's vicinity
Risk	Can fail if $f(x_n) \approx f(x_{n-1})$ (near-zero denominator)
Speed vs bisection	Significantly faster in practice

---

8 The Regula Falsi Method (False Position)

8.1 Motivation

The bisection method is **guaranteed to converge** but is slow. The secant method is **fast** but is not guaranteed to converge (no bracketing).

The **Regula Falsi** (Latin for "false position") method combines the best of both:

- Like **bisection**, it always maintains a bracket $[a, b]$ with $f(a) \cdot f(b) < 0$.
- Like the **secant method**, it uses a straight line (secant) to estimate the root, rather than simply taking the midpoint.

8.2 Derivation

Instead of using the midpoint $c = (a + b)/2$ (as in bisection), we use the x-intercept of the straight line joining $(a, f(a))$ and $(b, f(b))$:

The equation of this line is:

$$\frac{y - f(a)}{x - a} = \frac{f(b) - f(a)}{b - a}$$

Setting $y = 0$ and solving for x:

$$c = a - f(a) \cdot \frac{b - a}{f(b) - f(a)} = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

8.3 Algorithm

Given: f, interval $[a, b]$ with $f(a) \cdot f(b) < 0$, tolerance ε

1. Compute: $c = a - f(a) \cdot \dfrac{b - a}{f(b) - f(a)}$
2. If $|f(c)| < \varepsilon$ or $|b - a| < \varepsilon$: **stop**, c is the root.
3. If $f(a) \cdot f(c) < 0$: set $b = c$. Else: set $a = c$.
4. Go to step 1.

8.4 Solved Example 3.7 — Regula Falsi

Find the root of $f(x) = x^3 - x - 2 = 0$ in $[1, 2]$ using Regula Falsi.

$f(1) = -2$, $f(2) = 4$.

Iteration 1:

$$c_1 = 1 - (-2) \cdot \frac{2 - 1}{4 - (-2)} = 1 + \frac{2}{6} = 1 + 0.3333 = 1.3333$$

$f(1.3333) = -0.9629 < 0$

$f(a) \cdot f(c) = f(1) \cdot f(1.3333) = (-2)(-0.9629) > 0 \Rightarrow$ set $a = 1.3333$

Iteration 2: $a = 1.3333,\ b = 2$

$$c_2 = 1.3333 - (-0.9629) \cdot \frac{2 - 1.3333}{4 - (-0.9629)} = 1.3333 + 0.9629 \cdot \frac{0.6667}{4.9629}$$

$$c_2 = 1.3333 + 0.9629 \times 0.1343 = 1.3333 + 0.1293 = 1.4626$$

$f(1.4626) = -0.3359 < 0 \Rightarrow$ set $a = 1.4626$

Iteration 3: $a = 1.4626,\ b = 2$

$$c_3 = 1.4626 + 0.3359 \cdot \frac{2 - 1.4626}{4 - (-0.3359)} = 1.4626 + 0.3359 \cdot \frac{0.5374}{4.3359}$$

$$c_3 = 1.4626 + 0.3359 \times 0.1239 = 1.4626 + 0.0416 = 1.5042$$

$f(1.5042) = -0.1308 < 0 \Rightarrow$ set $a = 1.5042$

Iteration 4: $a = 1.5042,\ b = 2$

$$c_4 = 1.5042 + 0.1308 \cdot \frac{0.4958}{4.1308} = 1.5042 + 0.01570 = 1.5199$$

$f(1.5199) = -0.0046 < 0 \Rightarrow$ set $a = 1.5199$

Iteration 5: $c_5 \approx 1.5212,\ f(c_5) \approx -0.0001 \approx 0$

Summary table:

Iter	a	b	c	$f(c)$	New a or b
1	1.0000	2.0000	1.3333	-0.9629	$a = 1.3333$
2	1.3333	2.0000	1.4626	-0.3359	$a = 1.4626$
3	1.4626	2.0000	1.5042	-0.1308	$a = 1.5042$
4	1.5042	2.0000	1.5199	-0.0046	$a = 1.5199$
5	1.5199	2.0000	1.5212	-0.0001	Converged

Root: $x^* \approx 1.5214$

⚠ One-sided convergence in Regula Falsi

Notice that in this example, $b = 2$ never changed — only a was updated. This is a known issue with Regula Falsi: one endpoint can get "stuck," leading to **slower convergence** than the secant method. This is improved by the **Modified Regula Falsi** (Illinois method), which halves f at the stuck endpoint.

8.5 Comparison: Bisection vs Regula Falsi vs Secant

Feature	Bisection	Regula Falsi	Secant
Guaranteed convergence	Yes	Yes	No
Requires bracket	Yes	Yes	No
Requires derivative	No	No	No
Order of convergence	1 (linear)	≈ 1 (can be slow)	≈ 1.618
Speed	Slow	Moderate	Fast

---

9 Tutorial Questions

9.1 Section A: Concepts and Definitions

Question 1

(a) Define the term *root* of a function $f(x)$. What does it mean geometrically?

(b) Explain why equations like $e^x = 5x$ cannot be solved analytically, and must be handled numerically.

(c) Write each of the following in the form $f(x) = 0$, then determine whether a root exists in the given interval by evaluating f at the endpoints.

(i) $e^x = x + 5$ on $[1, 2]$

(ii) $\ln(x) = 2 - x$ on $[1, 2]$

(iii) $x^3 = 5 - x$ on $[1, 2]$

Question 2

Explain the difference between the following pairs of terms:

(a) *Order of convergence* and *rate of convergence*

(b) *Convergence criterion* and *convergence condition*

(c) *Bisection method* and *Regula Falsi method*

Question 3

State the three stopping criteria commonly used in iterative root-finding methods. Explain when each one is appropriate and give an example scenario for each.

9.2 Section B: Fixed-Point Iteration

Question 4

The equation $x^3 + x - 1 = 0$ can be rearranged in different ways:

(i) $x = 1 - x^3$

(ii) $x = (1 - x)^{1/3}$

(iii) $x = \dfrac{1}{1 + x^2}$

For each rearrangement $x = g(x)$:

(a) Compute $|g'(x)|$ at the approximate root $x^* \approx 0.682$.

(b) State whether the fixed-point iteration converges or diverges near x^*.

(c) For the convergent rearrangement(s), perform **5 iterations** starting from $x_0 = 0.5$. Record results in a table: n, x_n, x_{n+1}, $|x_{n+1} - x_n|$.

Question 5

The equation $x = 2\sin(x)$ has a root at $x = 0$ (trivial) and non-trivial roots.

(a) Rewrite as $f(x) = 0$ and locate a non-trivial root in a suitable interval by sign-change test.

(b) Use fixed-point iteration with $x_{n+1} = 2\sin(x_n)$ from $x_0 = 1.5$. Perform 8 iterations. Does this converge?

(c) Check the convergence condition. Explain your observation.

9.3 Section C: Bisection Method

Question 6

Use the bisection method to find a root of each function in the given interval. Carry out the iterations until the interval width is less than 0.01. Record each step in a table.

(a) $f(x) = x^3 - 2x - 5 = 0$ on $[2, 3]$

(b) $f(x) = x - e^{-x} = 0$ on $[0, 1]$

(c) $f(x) = \cos(x) - xe^x = 0$ on $[0, 1]$

Question 7

(a) How many bisection iterations are needed to find a root of $f(x) = x^2 - 7$ on $[2, 3]$ to an accuracy of 10^{-5}?

(b) Carry out the required number of iterations and confirm your answer.

(c) Compare the result with $\sqrt{7} = 2.6457513\ldots$

Question 8

The bisection method is applied to $f(x) = x^5 - 3x - 1$ on $[1, 2]$.

(a) Verify the sign change at the endpoints.

(b) After how many iterations is the root guaranteed accurate to 6 decimal places?

(c) Carry out **6 iterations** manually and find the approximate root.

9.4 Section D: Secant and Regula Falsi Methods

Question 9

Apply the **secant method** to find the root of $f(x) = x^2 - 5 = 0$, starting from $x_0 = 2$ and $x_1 = 3$.

(a) Perform 5 iterations. Record n, x_n, $f(x_n)$, x_{n+1}, $|x_{n+1} - x_n|$.

(b) Compare your result with the true value $\sqrt{5} = 2.2360679\ldots$

(c) Compute the ratio $\frac{|e_{n+1}|}{|e_n|^{1.618}}$ for iterations 2 to 4. What does this suggest about the order of convergence?

Question 10

Apply the **Regula Falsi method** to $f(x) = e^x - 3x = 0$ on $[1, 2]$.

(a) Perform 6 iterations. Record a, b, c, $f(c)$, and the updated bracket at each step.

(b) Which endpoint stays fixed? What does this tell you about Regula Falsi?

(c) Compare the convergence speed with bisection applied to the same problem (compare how quickly $|f(c)|$ decreases).

9.5 Section E: Newton–Raphson Method

Question 11

Apply Newton–Raphson to find the root of each function, starting from the given x_0. Iterate until $|x_{n+1} - x_n| < 10^{-6}$.

(a) $f(x) = x^2 - 7$, $x_0 = 3$

(b) $f(x) = \cos(x) - x$, $x_0 = 0.5$

(c) $f(x) = x^3 - 3x + 1$, $x_0 = 2$

(d) $f(x) = e^{-x} - x$, $x_0 = 0.5$

For each: (i) Write down $f'(x)$. (ii) Show all iterations in a table. (iii) State how many iterations were needed.

Question 12

The equation $x - \tan(x) = 0$ has roots at $x = 0$ and near $x = 4.493$.

(a) Apply Newton–Raphson from $x_0 = 4.5$ to find the non-trivial root.

(b) What happens if you start from $x_0 = \pi/2$? Explain.

(c) What property of Newton–Raphson does this illustrate?

Question 13

Consider $f(x) = (x - 1)^2 = 0$, which has a **double root** at $x^* = 1$.

(a) Apply Newton–Raphson from $x_0 = 2$. Perform 8 iterations.

(b) Compute the errors $e_n = |x_n - 1|$ and the ratio e_{n+1}/e_n.

(c) What order of convergence do you observe? Is this consistent with quadratic convergence?

(d) Explain theoretically why Newton–Raphson converges only **linearly** at a double root. *Hint:* Consider $f'(x^*)$ for a double root.

9.6 Section F: Newton's Method for Polynomials and Horner's Scheme

Question 14

Use Horner's scheme to evaluate $P(x) = 3x^4 - 2x^3 + x^2 - 5x + 4$ and $P'(x)$ at:

(a) $x = 2$

(b) $x = -1$

(c) $x = 0.5$

For each, show the full Horner table (both passes) and verify by direct calculation.

Question 15

Use Newton's method (with Horner's scheme) to find a root of $P(x) = x^4 - 3x^3 + x^2 + 5x - 2$ near $x_0 = 3$.

Perform **4 iterations** and record x_n, $P(x_n)$, $P'(x_n)$, x_{n+1} at each step.

9.7 Section G: Comparison and Reflection

Question 16

The equation $f(x) = x^3 - 7x + 2 = 0$ has three real roots.

(a) Show by sign changes that roots exist in $[-3, -2]$, $[0, 1]$, and $[2, 3]$.

(b) Apply bisection (5 iterations each) to find each root to 2 decimal places.

(c) Apply Newton–Raphson to find each root, choosing appropriate starting values. Compare the number of iterations needed with bisection.

(d) For the root in $[0, 1]$, compare the convergence of all four methods (bisection, Regula Falsi, secant, Newton–Raphson) by applying each and tabulating the errors at each iteration.

Question 17

Apply Bairstow's method to find all roots of $P(x) = x^3 - 6x^2 + 11x - 6$.

(a) Start with $r = 1$, $s = -2$. Perform two full Bairstow iterations (computing b_k, c_k, Δr, Δs at each step).

(b) Once a quadratic factor $x^2 - rx - s$ is found, extract the two roots from it.

(c) Divide $P(x)$ by the quadratic factor to find the remaining linear factor and its root.

(d) Verify all three roots by substitution into $P(x)$.

(Note: The roots of this polynomial are $x = 1, 2, 3$ — use this to check your work.)

End of Topic 3 Tutorial Questions
