Regula Falsi (False Position) Method

NC 08 • Module 2: Solution of nonlinear equations

Central question	Can we use function values to choose a better point than the midpoint?
Lecture focus	Secant intercept, bracket updates, convergence, stagnation and implementation.
Prerequisite	Root isolation (NC 06) and bisection (NC 07).

Learning outcomes

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

Explain	Derive the false-position formula from a straight line joining two points.
Compute	Apply the method and construct iteration tables for nonlinear equations.
Evaluate	Compare false position with bisection and identify endpoint stagnation.

Principle of false position

Assume f is continuous on [a,b] and f(a) and f(b) have opposite signs.

Bracket	The intermediate value theorem guarantees at least one root inside the interval.
Approximation	Replace the curve locally by the chord joining (a,f(a)) and (b,f(b)).
Intercept	Use the x-intercept of the chord as the next estimate c; preserve the sign-changing bracket.

Geometric interpretation

The chord intercept is generally different from the midpoint.

Endpoints

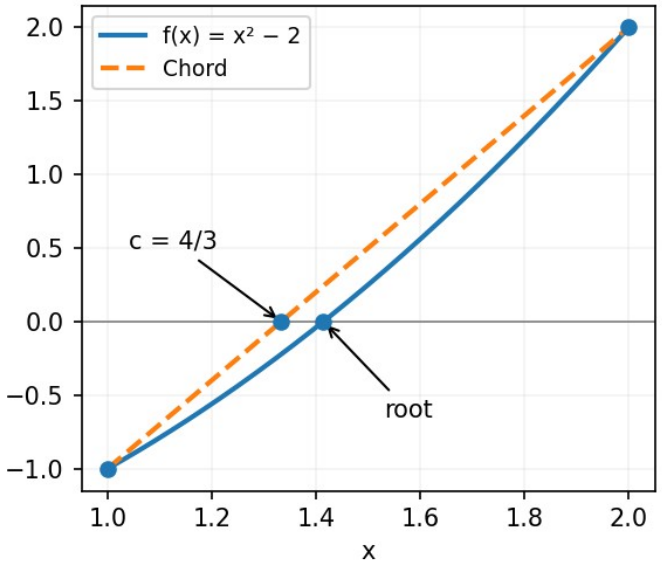
For $f(x)=x^2-2$ on $[1,2]$, $f(1)=-1$ and $f(2)=2$.

Chord

The straight line through $(1,-1)$ and $(2,2)$ crosses the x-axis at $4/3$.

Update

$f(4/3)=-2/9$; retain $[4/3,2]$. The actual root is about 1.414214.



Derivation of the formula

Use the two-point equation of the chord.

Slope	$m = (f(b)-f(a))/(b-a)$.
Set y = 0	$0 = f(a) + m(c-a)$. Solve for c.
Formula	$c = a - f(a)(b-a)/(f(b)-f(a)) = (a\,f(b)-b\,f(a))/(f(b)-f(a))$.

Bracket update and algorithm

Input: continuous f, a < b, opposite endpoint signs, tolerance and iteration limit.

Initialize	Check exact endpoint roots first; otherwise reject a non-sign-changing bracket.
Compute	Evaluate c using the false-position formula, then evaluate f(c).
Update	If f(c)=0, return c. If f(a) and f(c) differ in sign, set b=c; otherwise set a=c. Repeat.

Example 1: square root of 2

Solve $f(x)=x^2-2$ on $[1,2]$.

Initial values	$f(1)=-1$, $f(2)=2$; a valid bracket.
Iteration 1	$c_1 = 1 - (-1)(2-1)/(2-(-1)) = 1.333333$; $f(c_1)=-0.222222$.
Iteration 2	$a=4/3$, $b=2$; $c_2=1.4$, $f(c_2)=-0.04$. New bracket $[1.4,2]$.

Example 1: iteration table

Each row gives the estimate c and residual f(c); b remains 2.

Iterations 1–2	1: c=1.333333333; f(c)=−0.222222222 2: c=1.400000000; f(c)=−0.040000000
Iterations 3–4	3: c=1.411764706; f(c)=−0.006920415 4: c=1.413793103; f(c)=−0.001189061
Iterations 5–6	5: c=1.414141414; f(c)≈−0.000204061 6: c=1.414201183; f(c)≈−0.000035013

Example 1: stopping decision

Use a residual tolerance of 10^{-4} for this example.

First success	At iteration 6, $\lvert f(c)\rvert \approx 0.000035013 < 0.0001$.
Approximate root	$c \approx 1.414201183$. The actual positive root is $\sqrt{2} \approx 1.414213562$.
Certified error	Since $f'(x)=2x \geq 2$ on $[1,2]$, $\lvert c-r\rvert \leq \lvert f(c)\rvert/2 \approx 0.00001751$.

Example 2: cubic equation

Solve $f(x)=x^3-x-2$ on $[1,2]$.

Initial values	$f(1)=-2$; $f(2)=4$. A root exists by continuity.
Iteration 1	$c_1=1.333333$; $f(c_1)\approx-0.962963$. Set $a=c_1$.
Iteration 2	$c_2\approx1.462687$; $f(c_2)\approx-0.333339$. Again set $a=c_2$.

Example 2: convergence

The positive root is approximately 1.521379707.

Iterations 3–4	3: $c \approx 1.504019$; $f(c) \approx -0.101818$ 4: $c \approx 1.516331$; $f(c) \approx -0.029895$
Iterations 5–6	5: $c \approx 1.519919$; $f(c) \approx -0.008675$ 6: $c \approx 1.520957$; $f(c) \approx -0.002509$
Observation	The estimates improve, but the right endpoint remains at 2.

Stopping criteria

A root estimate is not the same as a certified error bound.

Residual	Stop if $\lvert f(c)\rvert \le \varepsilon_f$, reporting residual convergence rather than a certified x-error.
Bracket width	If $b-a \le \varepsilon_x$, any point in the bracket is within ε_x of a root.
Safeguards	Limit iterations; detect $c=a$ or $c=b$ due to floating-point rounding. Small successive changes alone are not proof of accuracy.

Convergence and stagnation

Unlike bisection, false position does not necessarily halve the bracket.

Typical behavior	For a smooth convex increasing function, one endpoint may remain fixed while the other approaches the root.
Rate	Convergence is often linear near a simple root; no universal halving bound applies.
Remedy	Use bisection safeguards or modified false-position methods such as Illinois to reduce endpoint locking.

Bisection versus false position

Both are derivative-free bracketing methods.

Bisection	Uses midpoint; halves width each update; predictable error bound.
False position	Uses chord intercept; adapts to function values; may have a fixed endpoint.
Choice	Bisection is preferred for guaranteed interval shrinkage; false position can be faster on suitable functions.

Python implementation: setup

The following implementation assumes continuity and finite function values.

Function header	def false_position(f, a, b, tol=1e-6, limit=1000):
Validation	if not (a < b and tol > 0): raise ValueError("interval/tolerance") fa, fb = f(a), f(b)
Check bracket	if fa == 0: return a if fb == 0: return b if (fa < 0) == (fb < 0): raise ValueError("no bracket")

Python implementation: loop

Continue inside the function. Here tol is a residual tolerance.

Compute

```
for _ in range(limit):
    c = a - fa*(b-a)/(fb-fa); fc = f(c)
    if abs(fc) <= tol: return c
```

Update

```
    if not a < c < b: raise ArithmeticError("stagnation")
    if (fa < 0) != (fc < 0): b, fb = c, fc
    else: a, fa = c, fc
```

Finish and test

```
raise RuntimeError("iteration limit")
# Test: false_position(lambda x: x*x-2, 1, 2)
```

Practice questions

Attempt the following before reviewing the solutions.

Q1 Find the first two false-position estimates for $f(x)=x^2-5$ on $[2,3]$.

Q2 Explain why a small residual need not guarantee a small root error.

Q3 For $f(x)=x-3$ on $[0,5]$, compare the first false-position and bisection estimates.

Solutions and recap

Next: NC 09 — Fixed-Point Iteration Method.

A1	$c_1=2.2$, $f(c_1)=-0.16$; $c_2=2.230769$, $f(c_2)\approx-0.023669$.
A2	A nearly flat function may have small residual far from a root; a derivative lower bound can help certify the error.
A3 / Takeaway	False position gives 3 exactly; bisection gives 2.5. Use chord intercepts while preserving a valid bracket.