Numerical Computing

Lecture 3: Root-Finding Methods

Francisco Richter Mendoza

Università della Svizzera Italiana
Faculty of Informatics
Lugano, Switzerland

Today's Topics

- **Root-finding problem**
- **Bracketing methods**
- **Fixed-point iteration**
- **Newton's method**
- **Secant method**
- **Convergence analysis**

Root-Finding Problem

Definition

Given continuous function $f : [a, b] \to \mathbb{R}$, find x^* such that:

$$f(x^*) = 0$$

Intermediate Value Theorem

If f continuous on $[a, b]$ and $f(a) \cdot f(b) < 0$, then $\exists c \in (a, b)$ such that $f(c) = 0$.

Types of roots:

- **Simple root:** $f'(x^*) \neq 0$
- **Multiple root:** $f'(x^*) = 0$

Convergence Analysis

Order of Convergence

Sequence $\{x_k\}$ converges to x^* with order $p \geq 1$ if:

$$\lim_{k \to \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = C, \quad 0 < C < \infty$$

- $p = 1$: **Linear convergence**
- $1 < p < 2$: **Superlinear convergence**
- $p = 2$: **Quadratic convergence**

Practical implications:

- Linear: $O(\log(1/\epsilon))$ iterations
- Quadratic: $O(\log\log(1/\epsilon))$ iterations

Root-Finding Methods Comparison

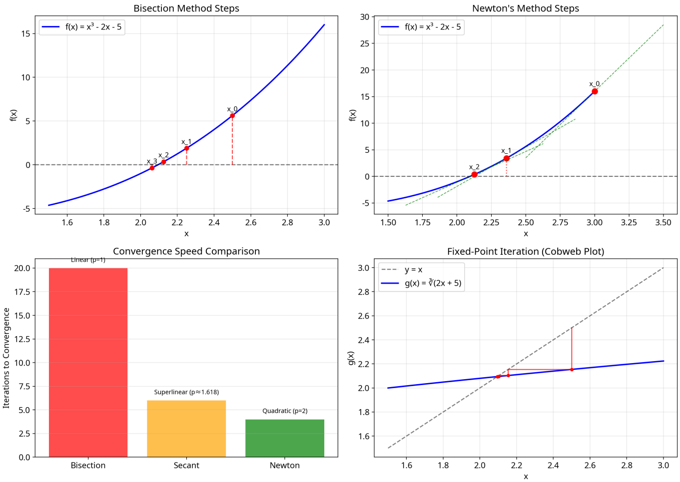

Figure: Comprehensive comparison of root-finding approaches: bisection's systematic interval reduction, Newton's tangent-line strategy, convergence rate analysis, and fixed-point iteration visualization.

Bisection Method

Algorithm

Given f continuous on $[a, b]$ with $f(a) \cdot f(b) < 0$:

1. Set $c = (a + b)/2$
2. If $f(a) \cdot f(c) < 0$, set $b = c$; otherwise $a = c$
3. Repeat until $|b - a| <$ tolerance

Convergence

$$|x_n - x^*| \leq \frac{b_0 - a_0}{2^n}$$

Iterations needed: $n = \lceil \log_2((b_0 - a_0)/\epsilon) \rceil$

Properties: Guaranteed convergence, linear rate, robust but slow

Fixed-Point Iteration

Fixed-Point Problem

Find x^* such that $g(x^*) = x^*$

Root-finding reformulation: $g(x) = x - \alpha f(x)$

Fixed-Point Theorem

If g continuously differentiable on $[a, b]$ with:

- $g([a, b]) \subseteq [a, b]$
- $|g'(x)| \leq L < 1$ for all $x \in [a, b]$

Then $x_{k+1} = g(x_k)$ converges linearly to unique fixed point.

Convergence order:

- If $g'(x^*) \neq 0$: Linear
- If $g'(x^*) = 0$: Quadratic

Newton's Method

Algorithm

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Equivalent to fixed-point with $g(x) = x - f(x)/f'(x)$

Convergence Theorem

For simple root x^* and x_0 sufficiently close:

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = \frac{|f''(x^*)|}{2|f'(x^*)|}$$

Quadratic convergence when $f'(x^*) \neq 0$

Multiple roots: Linear convergence with rate $(m-1)/m$

Modified Newton: $x_{k+1} = x_k - m\frac{f(x_k)}{f'(x_k)}$ restores quadratic convergence

Convergence Analysis

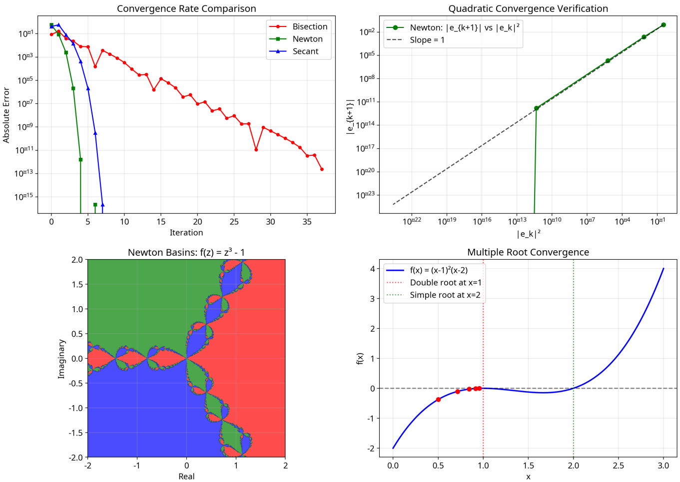

Figure: Detailed convergence behavior: error reduction rates, quadratic convergence verification, complex basin analysis, and multiple root challenges.

Secant Method

Motivation

Approximate derivative using finite differences:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

Algorithm

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

Requires two initial guesses x_0 and x_1

Convergence

Superlinear convergence with order $p = \frac{1+\sqrt{5}}{2} \approx 1.618$

Advantages: No derivative needed, superlinear convergence

Practical Implementation Issues

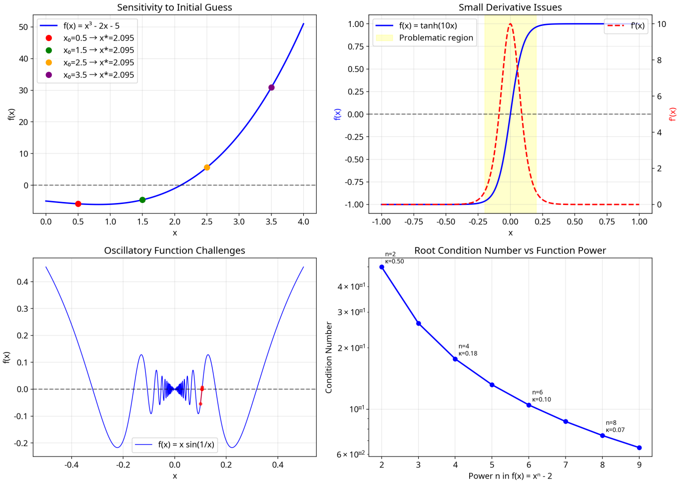

Figure: Real-world challenges: sensitivity to initial conditions, small derivative problems, oscillatory function difficulties, and condition number effects.

Method Selection Guidelines

Bisection Method

Use when: Robustness critical, derivative unavailable

Pros: Always converges, simple **Cons:** Slow, requires bracketing

Newton's Method

Use when: Fast convergence needed, derivative available

Pros: Quadratic convergence **Cons:** Requires good initial guess, may diverge

Secant Method

Use when: Derivative expensive, speed important

Pros: No derivative, superlinear convergence **Cons:** Two initial guesses, may fail near extrema

Advanced Techniques

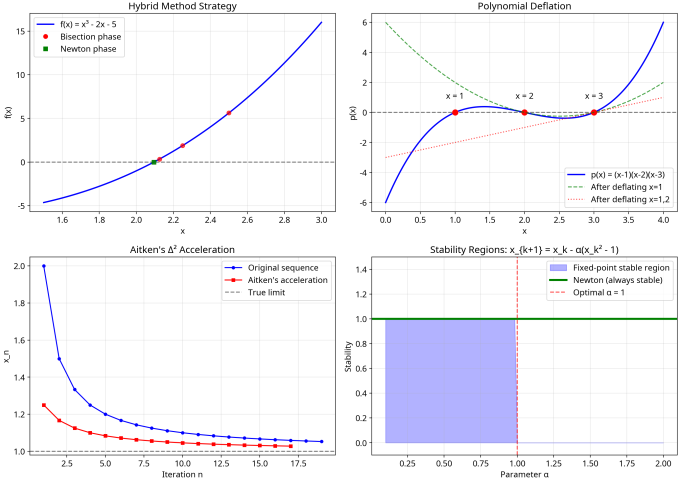

Figure: Sophisticated approaches: hybrid methods, polynomial deflation, Aitken acceleration, and stability analysis.

Error Analysis & Stopping Criteria

Multiple Stopping Conditions

- **Absolute:** $|x_{k+1} - x_k| < \epsilon_{\text{abs}}$
- **Relative:** $\frac{|x_{k+1} - x_k|}{|x_k|} < \epsilon_{\text{rel}}$
- **Function:** $|f(x_k)| < \epsilon_f$

Combined Criterion

$$|x_{k+1} - x_k| < \epsilon_{\text{abs}} + \epsilon_{\text{rel}}|x_k| \quad \text{AND} \quad |f(x_k)| < \epsilon_f$$

Condition Number

$$\kappa = \frac{|x^*|}{|f'(x^*)|}$$

Large κ indicates sensitivity to perturbations

Computational Complexity

Cost per Iteration

- **Bisection:** 1 function evaluation
- **Newton:** 1 function + 1 derivative evaluation
- **Secant:** 1 function evaluation (after initialization)

Total Cost for Tolerance ϵ

- **Bisection:** $O(\log(1/\epsilon))$ function evaluations
- **Newton:** $O(\log\log(1/\epsilon))$ function + derivative evaluations
- **Secant:** $O(\log(1/\epsilon)^{0.618})$ function evaluations

Trade-off: Newton fastest when derivatives cheap, secant when derivatives expensive

Key Takeaways

- **Method selection:** Balance robustness vs speed
- **Convergence theory:** Higher order $\Rightarrow$ faster convergence
- **Initial conditions:** Critical for open methods
- **Hybrid approaches:** Combine methods for optimal performance
- **Error control:** Multiple stopping criteria ensure reliability
- **Condition numbers:** Measure problem sensitivity

Next Lecture: Linear Algebra Foundations