Lec 9: Nonlinear Systems

15-369/669/769: Numerical Computing

Instructor: Minchen Li

Table of Content

- Root-Finding in a Single Variable
- Multivariable Problems

Table of Content

- Root-Finding in a Single Variable
- Multivariable Problems

Root-Finding in a Single Variable

Characterizing Problems

- Goal: Find $x^* \in \mathbb{R}$ such that $f(x^*) = 0$.
- Linear case: $ax - b = 0 \implies x^* = b/a$ (trivial).
- Nonlinear equations (e.g. $y^2 + e^{\cos y} - 3 = 0$) are harder.
- Without assumptions, root-finding may fail:

$$f(x) = \begin{cases} -1, & x \leq 0 \\ 1, & x > 0 \end{cases}, \quad f(x) = \begin{cases} -1, & x \in \mathbb{Q} \\ 1, & \text{otherwise} \end{cases}$$

- Need structural assumptions to make the problem well-posed.

Root-Finding in a Single Variable

Regularity Assumptions on f

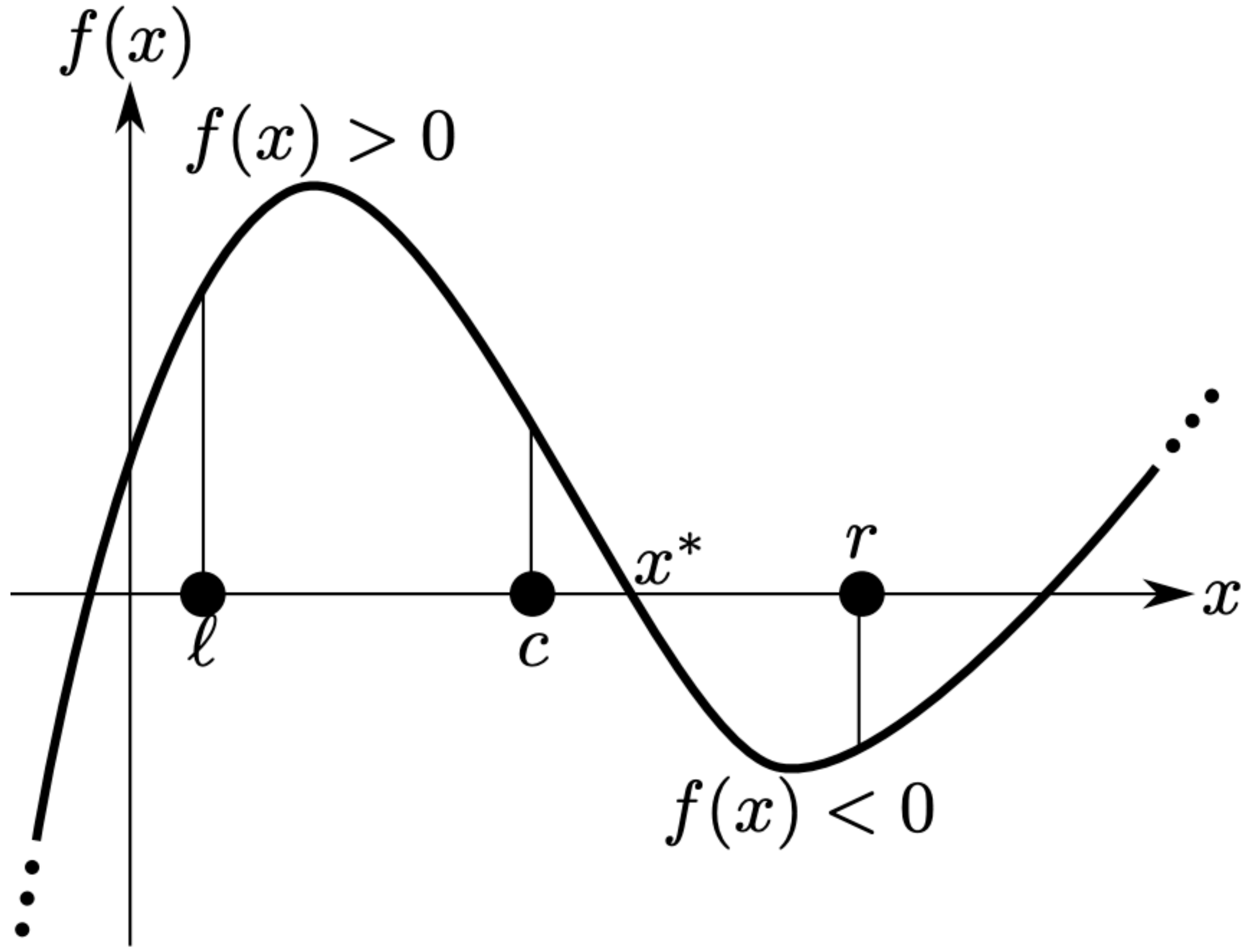
- **Continuity:** $f(x) \to f(y)$ as $x \to y$.
- **Lipschitz continuity:** $|f(x) - f(y)| \leq c|x - y|$. Controls rate of change.
- **Differentiability:** derivative $f'(x)$ exists for all x.
- $\mathbf{C^k}$: f has k continuous derivatives. C^∞: all derivatives exist and are continuous.

- **Example 8.1 (Classifying functions):**
  - $f(x) = \cos x$: C^∞, Lipschitz.
  - $g(x) = x^2$: C^∞, but not Lipschitz on $\mathbb{R}$. Locally Lipschitz on bounded intervals ($[0, 1]$).
  - $h(x) = |x|$: continuous (C^0), Lipschitz, not differentiable at 0.

Root-Finding in a Single Variable

Continuity and Bisection

- **Intermediate Value Theorem (IVT):** If $f : [a, b] \to \mathbb{R}$ is continuous and $f(a) < u < f(b)$ (or reversed), then $\exists z \in (a, b)$ with $f(z) = u$.
- Root existence: if $f(\ell)f(r) < 0$, then a root lies in (ℓ, r).



- **Bisection algorithm:**
  1. Start with $[\ell_0, r_0]$ such that $f(\ell_0)f(r_0) < 0$.
  2. At each step: set midpoint $c_k = (\ell_k + r_k)/2$.
  3. Choose subinterval where sign change occurs.

- Always maintains a valid bracket; requires only continuity.
- **Error:** after k steps, interval length $L_k = 2^{-k} L_0, \quad |c_k - x^*| \leq \frac{1}{2} L_k$.
- Converges unconditionally, but only *linearly*.

Root-Finding in a Single Variable

Bisection: Convergence and Role

- Define error $E_k = |c_k - x^*|$.

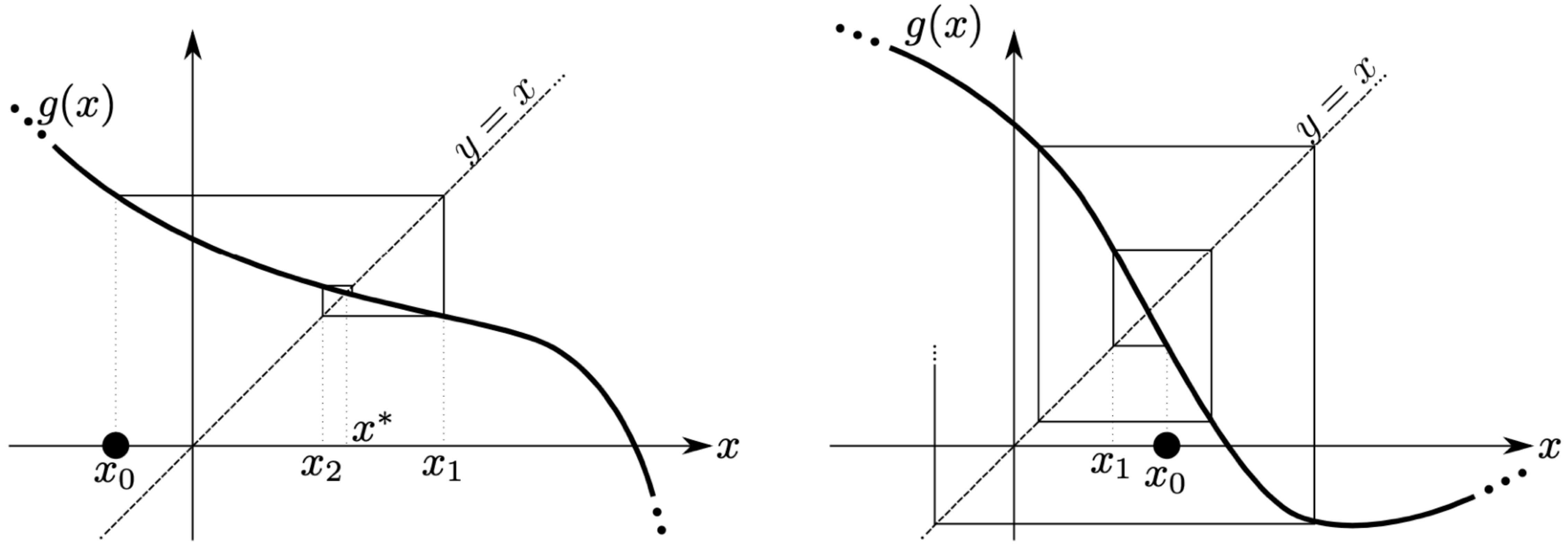
$$E_{k+1} \leq \tfrac{1}{2}E_k \quad \Rightarrow \quad \text{linear convergence.}$$

- Pros: guaranteed convergence, robust, simple.
- Cons: requires bracketing interval; slow (linear rate).
- Often used to *refine* a root estimate when bracket is known, rather than as a standalone global method.

Root-Finding in a Single Variable

Fixed Point Iteration

- Rewrite $f(x) = 0$ as $g(x) = x$.
- Algorithm: choose x_0, iterate $x_k = g(x_{k-1})$.
- If convergent, x^* is a **fixed point** of g.



(a) Convergence (b) Divergence

- **Contraction mapping theorem:**
  - If g is Lipschitz with constant $c < 1$ on a neighborhood of x^*, then

$$E_k = |x_k - x^*| \leq cE_{k-1} \implies E_k \leq c^k E_0.$$

  - Guarantees linear convergence when $|g'(x^*)| < 1$.

Root-Finding in a Single Variable

Fixed Point Iteration: Convergence Rates

- **Linear convergence:** if $|g'(x^*)| < 1$. Local contraction, rate $\rho = |g'(x^*)|$.
- **Quadratic convergence:** if $g'(x^*) = 0$ and g is C^2.

$$g(x_{k-1}) = g(x^*) + \tfrac{1}{2}g''(x^*)(x_{k-1} - x^*)^2 + O(E_{k-1}^3).$$

$$E_k \le \tfrac{1}{2}(|g''(x^*)| + \varepsilon)E_{k-1}^2.$$

- Much faster v.s. bisection: fewer iterations to reach accuracy once close to x^*.

Root-Finding in a Single Variable

Examples of Fixed Point Iteration

- $x = \cos x$: iteration $x_{k+1} = \cos x_k$, linear convergence to $x^* \approx 0.739$.

k	0	1	2	3	4	5	6	7	8
x_k	1.000	0.540	0.858	0.654	0.793	0.701	0.764	0.722	0.750

- $x = \sin(x^2)$: iteration $x_{k+1} = \sin(x_k^2)$, quadratic convergence near $x^* = 0$.

k	0	1	2	3	4	5	6	7	8
x_k	1.000	0.841	0.650	0.410	0.168	0.028	0.001	0.000	0.000

- $g(x) = e^x + e^{-x} - 5$. Derivative: $g'(x) = e^x - e^{-x}$, large in magnitude.
- Not a contraction; iteration diverges.

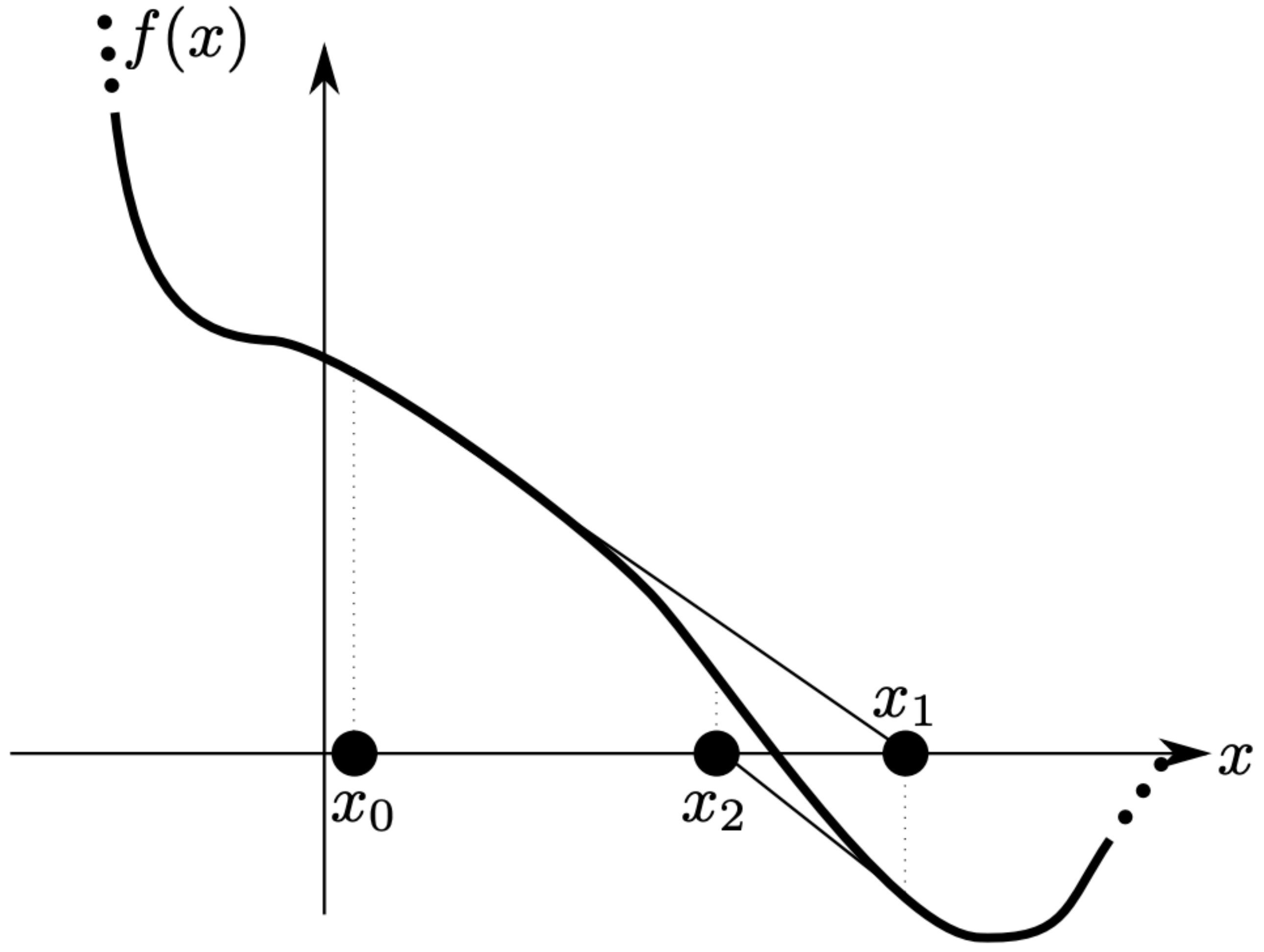
k	0	1	2	3	4	5	6
x_k	1.000	-1.914	1.927	0.212	2.609	8.660	5760.375

Root-Finding in a Single Variable

Newton's Method: Derivation

- Tighten assumptions: now require $f \in C^1$ (continuously differentiable).
- Approximate $f(x)$ near current iterate x_k by tangent line:

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k).$$



- Set this linear model to zero and solve:

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

- Interpretation: replace nonlinear root-finding with root of local linearization.

Root-Finding in a Single Variable

Newton's Method: Fixed-Point Form and Convergence

- Define $g(x) = x - \frac{f(x)}{f'(x)}$. Newton's method is fixed point iteration on g.
- Differentiate:

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}.$$

- At a simple root x^*: $f(x^*) = 0,\ f'(x^*) \neq 0 \implies g'(x^*) = 0$.
- By fixed point theory: Newton's method converges **quadratically** near simple roots if x_0 is close enough.
- If x^* is not simple ($f'(x^*) = 0$), convergence may degrade to linear or worse.

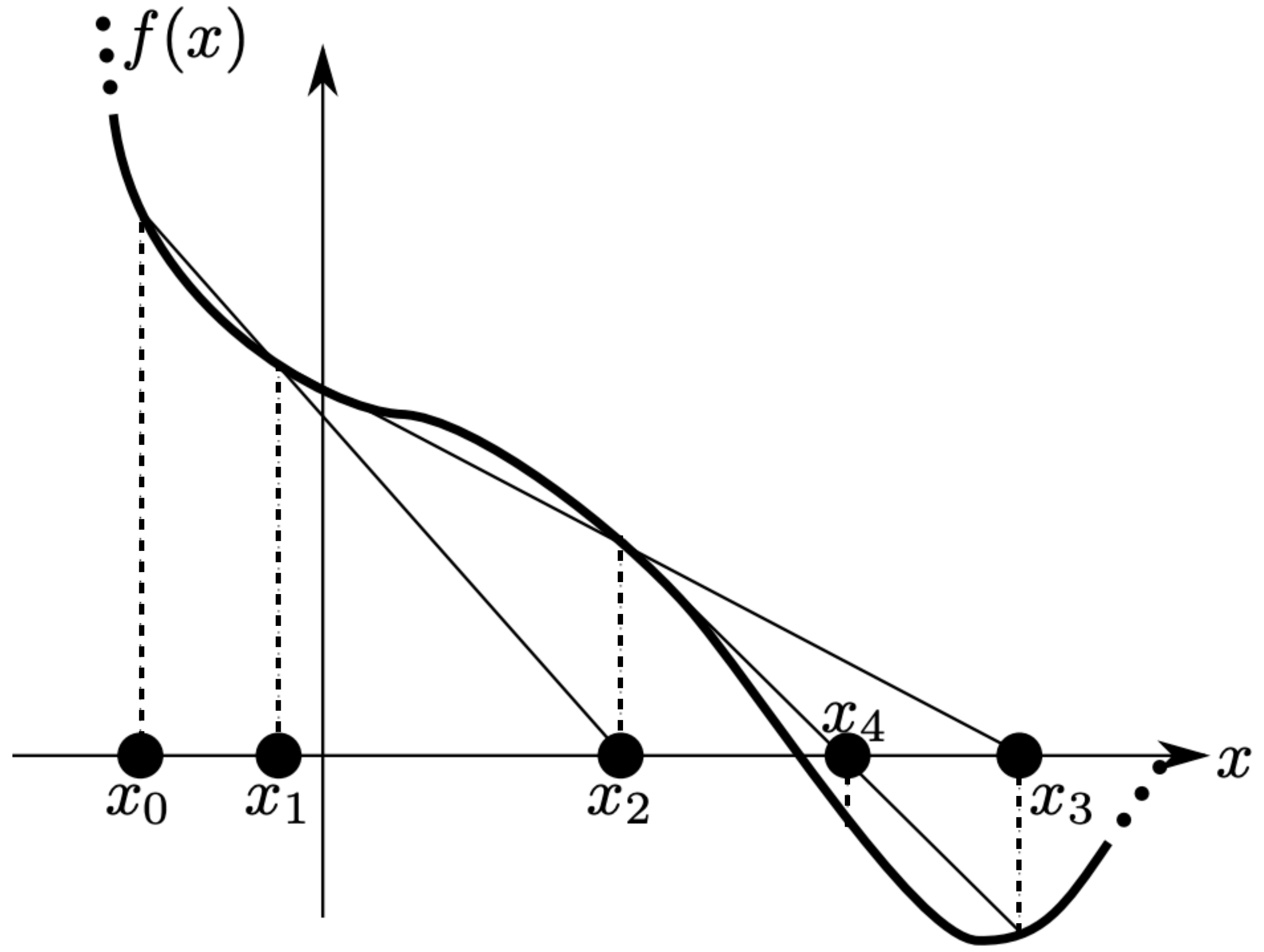
Root-Finding in a Single Variable

Newton's Method: Extensions and Remarks

- Derivation used a first-order Taylor approximation. Higher-order versions exist:
    - **Halley's method:** includes second derivative f''.
    - More generally: **Householder methods** use higher derivatives.
- Trade-off: faster convergence but more expensive derivative evaluations.
- Alternative approaches: replace Taylor expansion with rational or fractional interpolation (e.g. "linear fractional interpolation").

Root-Finding in a Single Variable

Secant Method: Motivation



- Newton's method requires evaluating both f and f'.
- If f is expensive (e.g. simulating rocket dynamics), derivative may be too costly.
- Idea: approximate derivative with finite difference using previous iterates:

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

- Plug into Newton's update: $x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$.
- Requires two starting points x_0, x_1. Reuses past evaluations — less cost, no explicit f'.

Root-Finding in a Single Variable

Secant Method: Convergence Properties

- Error analysis: secant method has order $\frac{1+\sqrt{5}}{2} \approx 1.618$ (“Golden Ratio”).
- Convergence is *superlinear*, between linear and quadratic.
- Practical advantage: often close to Newton’s method speed but avoids derivative evaluations.
- Drawback: not guaranteed to converge; poor initial guesses may fail.

Root-Finding in a Single Variable

Example: Newton vs. Secant

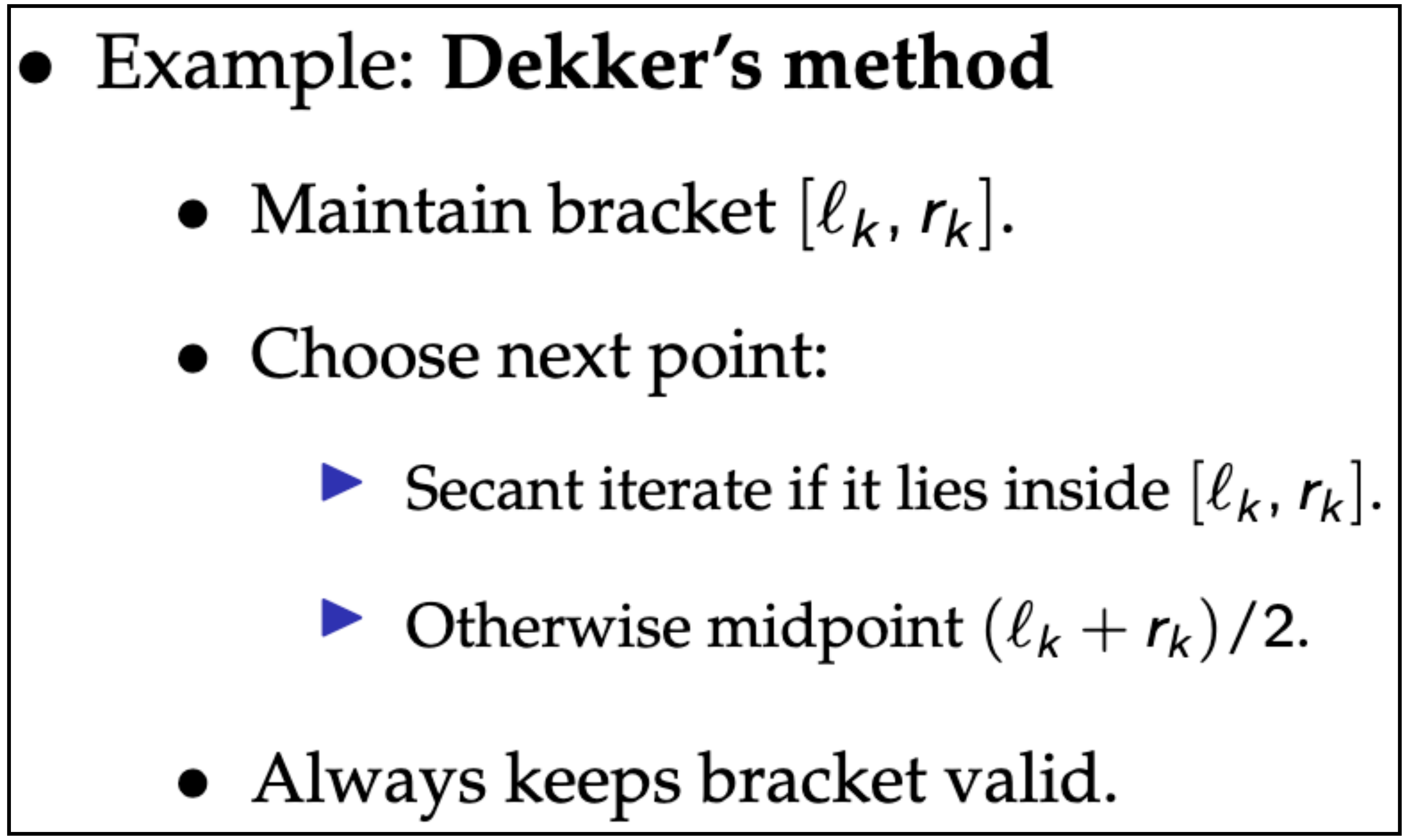
- Function: $f(x) = x^4 - 2x^2 - 4$.
- Newton iteration: $x_{k+1} = x_k - \frac{x_k^4 - 2x_k^2 - 4}{4x_k^3 - 4x_k}$.
- Secant iteration: $x_{k+1} = x_k - \frac{(x_k^4 - 2x_k^2 - 4)(x_k - x_{k-1})}{(x_k^4 - 2x_k^2 - 4) - (x_{k-1}^4 - 2x_{k-1}^2 - 4)}$.
- Numerical comparison (starting $x_0 = 3, x_{-1} = 2$):

k	0	1	2	3	4	5	6
x_k Newton	3.000	2.385	2.006	1.835	1.800	1.799	1.799
x_k Secant	3.000	1.927	1.882	1.809	1.800	1.799	1.799

Root-Finding in a Single Variable

Hybrid Techniques: Motivation

- **Bisection:** always converges, but linear rate.
- **Secant:** faster (superlinear) but may fail if guesses are poor.
- Idea: combine robustness of bisection with speed of secant.



- Example: **Dekker's method**
  - Maintain bracket $[\ell_k, r_k]$.
  - Choose next point:
    - Secant iterate if it lies inside $[\ell_k, r_k]$.
    - Otherwise midpoint $(\ell_k + r_k)/2$.
  - Always keeps bracket valid.

- Other advanced hybrids: **Brent's method** (guaranteed convergence, often faster).

Root-Finding in a Single Variable

Summary

- Many iterative schemes exist, trading off robustness vs. speed.

- Faster convergence means fewer iterations, but often more work per step.

Convergence rates:

1. Linear: $E_{k+1} \leq CE_k$.
2. Superlinear: $E_{k+1} \leq CE_k^r$, $r > 1$.
3. Quadratic: $E_{k+1} \leq CE_k^2$.
4. Cubic (or higher): $E_{k+1} \leq CE_k^3$.

Choice depends on problem:

- Bisection: guaranteed, linear.
- Newton: quadratic, needs derivative, may fail.
- Secant: superlinear, no derivative, may fail.
- Hybrids: combine advantages.

Table of Content

Multivariable Problems

Background

- Problem: solve $f(\mathbf{x}) = 0$ with $f : \mathbb{R}^n \to \mathbb{R}^m$.
- Linear case: $f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ (solving $A\mathbf{x} = \mathbf{b}$).
- General nonlinear case: much harder.
- Challenges: must solve for m equations simultaneously.
- Strategies like bisection do not extend naturally.

Multivariable Problems

Newton's Method in Multiple Dimensions

- Define Jacobian of f: $(Df)_{ij} = \frac{\partial f_i}{\partial x_j}$.
- Linearize near $\mathbf{x}_k$: $f(\mathbf{x}) \approx f(\mathbf{x}_k) + Df(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)$.
- Enforce $f(\mathbf{x}) = 0$: $Df(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) = -f(\mathbf{x}_k)$.
- If $n = m$ and $Df(\mathbf{x}_k)$ invertible:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [Df(\mathbf{x}_k)]^{-1} f(\mathbf{x}_k).$$

- In practice: do not form inverse; solve linear system each iteration.
- If $m \neq n$: use pseudoinverse or least-squares formulations.

Multivariable Problems

Newton’s Method in Multiple Dimensions: Convergence

- Same analysis as single-variable case:
  - If $Df(\mathbf{x}^*)$ nonsingular and $f \in C^1$, quadratic convergence near $\mathbf{x}^*$.
  - Fixed-point interpretation: iterate $\mathbf{x}_{k+1} = g(\mathbf{x}_k)$. Convergence requires largest eigenvalue of Dg to satisfy $|\lambda| < 1$.
- Summary: Newton’s method extends naturally, but requires solving an $n \times n$ linear system each step.

Multivariable Problems

Quasi-Newton and Broyden Methods

- Newton requires new Jacobian $Df(\mathbf{x}_k)$ each iteration (expensive).
- **Quasi-Newton methods:** approximate the Jacobian and update iteratively.
- Secant method in 1D is a quasi-Newton method. Goal: avoid explicit Jacobian computation.
- Idea: construct approximate Jacobian J_k satisfying secant condition:

$$J_k(\mathbf{x}_k - \mathbf{x}_{k-1}) = f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}).$$

- Directional derivative: $D_{\mathbf{v}} f = Df \cdot \mathbf{v}$. Approximation must agree with this along $\mathbf{x}_k - \mathbf{x}_{k-1}$.

Multivariable Problems

Broyden's Method: Derivation

- Maintain Jacobian approximation J_k.
- Update problem: find J_k close to J_{k-1} (in Frobenius norm) while enforcing secant condition:

$$\min_{J_k} \|J_k - J_{k-1}\|_F^2 \quad \text{s.t.} \quad J_k(\mathbf{x}_k - \mathbf{x}_{k-1}) = f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}).$$

- Introduce notations: $\Delta J = J_k - J_{k-1}$, $\Delta\mathbf{x} = \mathbf{x}_k - \mathbf{x}_{k-1}$, $\mathbf{d} = f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) - J_{k-1}\Delta\mathbf{x}$.
- Equivalent update:

$$J_k = J_{k-1} + \frac{\mathbf{d}(\Delta\mathbf{x})^\top}{\|\Delta\mathbf{x}\|_2^2}.$$

- Then take Newton-like step: $\mathbf{x}_{k+1} = \mathbf{x}_k - J_k^{-1} f(\mathbf{x}_k)$.

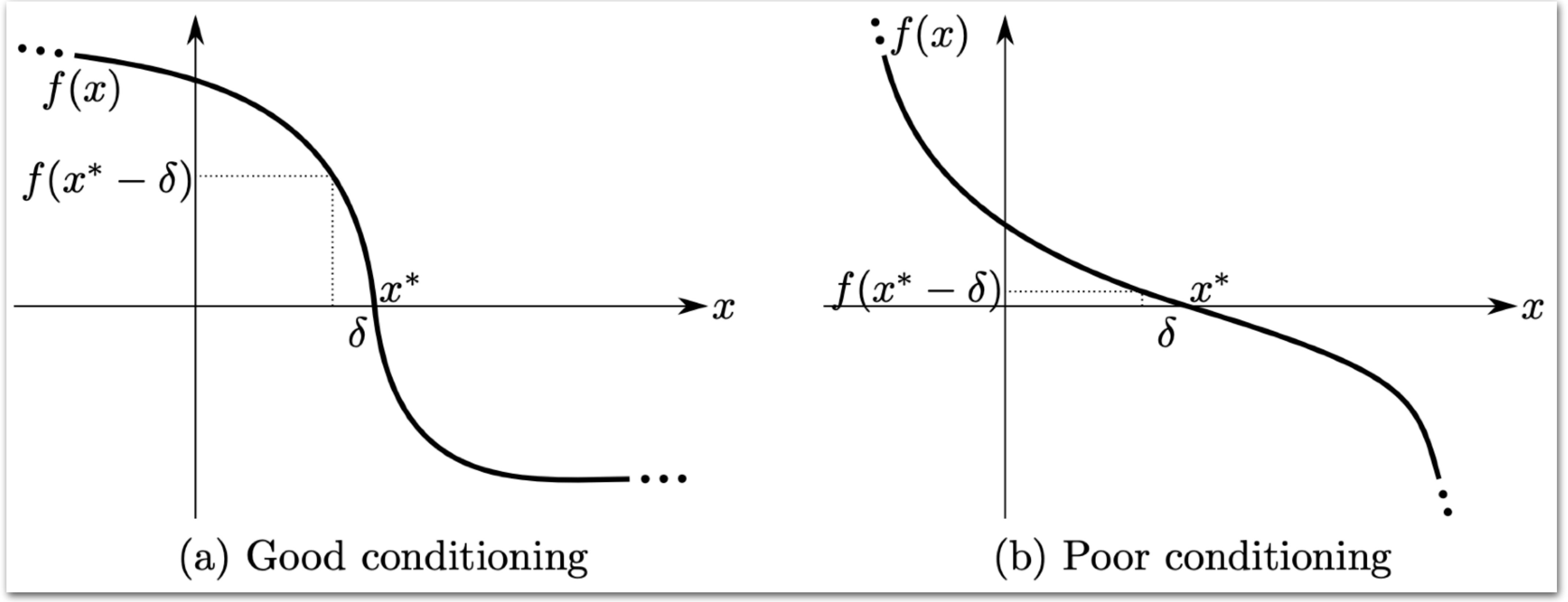
Multivariable Problems

Broyden's Method: Remarks

- Efficient: avoids computing new Jacobian at each step.
- Only rank-one update needed.
- Alternative: update J_k^{-1} directly via Sherman–Morrison formula.
- Widely used in large-scale nonlinear problems (e.g. optimization solvers).

Multivariable Problems

Conditioning of Root-Finding Problems



(a) Good conditioning (b) Poor conditioning

- Single variable: condition number at x^*:

$$\mathrm{cond}_{x^*} f = \frac{1}{|f'(x^*)|}.$$

- Interpretation: if $|f'(x^*)|$ is large, f changes rapidly near x^*; root is well-conditioned.
- If $|f'(x^*)|$ is small, small perturbations can move root far away (ill-conditioned).
- Multivariable case:

$$\mathrm{cond}_{x^*} f = \|Df(x^*)^{-1}\|.$$

- If $Df(x^*)$ is not invertible: condition number is ∞. ⇒ root-finding becomes degenerate and highly unstable.

Table of Content