CS 450: Numerical Analysis – Lecture 17

Nonlinear Equations: Bisection, Newton, and Secant

Edgar Solomonik

October 27, 2026

Today

Motivate and introduce basics of nonlinear systems of equations:

- Nonlinear systems and applications
- Properties of problem and solution
- Basic algorithms

Nonlinear systems of equations

Given $f : \mathbb{R} \to \mathbb{R}$, solve

$$f(x) = 0 \qquad \text{(1D)};$$

given $F : \mathbb{R}^n \to \mathbb{R}^n$, solve

$$F(x) = 0 \qquad (n\text{-D, a system of } n \text{ equations}).$$

A linear system is the special case $f(x) = Ax - b$: solving $Ax = b$ is solving $Ax - b = 0$.

Applications of nonlinear equations.

- A core building block for:
  1. *optimization*: $\min_x g(x) \implies \nabla g(x) = 0$;
  2. *differential equations*: e.g., discretizing $f''(x) - 2f(x)^2 + 3x^2 = 0$ by finite differences,

$$\frac{f(x-h) - 2f(x) + f(x+h)}{h^2} - 2f(x)^2 + 3x^2 = 0,$$

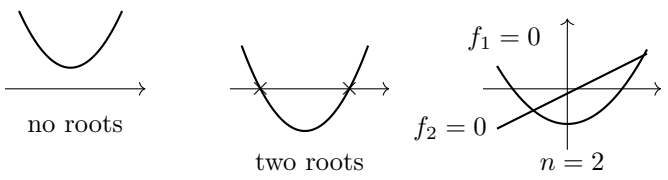
     gives a nonlinear system of equations in the samples of f.
- Nonlinear functions arise in modeling physical laws, and in using nonlinear models for data—e.g., neural networks.

Properties of 1D root finding

$f : \mathbb{R} \to \mathbb{R}$: find $x^* \in \mathbb{R}$ such that $f(x^*) = 0$. Solution existence or uniqueness is hard to ascertain. Examples:

$e^x + 1 = 0$	no solution
$e^{-x} - x = 0$	one solution
$x^2 - 4\sin(x) = 0$	two solutions
$x^3 - 6x^2 + 11x - 6 = 0$	three solutions
$\sin(x) = 0$	infinite number of solutions

Graphical examples.



For $n = 2$, the solution sets of $f_1(x) = 0$ and $f_2(x) = 0$ are curves; solutions of the system are their intersections.

Sensitivity and conditioning

Crudely speaking, we want to evaluate $f^{-1}(0)$—assuming a unique solution and local invertibility—so its local absolute sensitivity is the *reciprocal* of that of function evaluation. Assume f is sufficiently smooth and $f'(x^*) \neq 0$. Given a small constant perturbation $f \to f + \delta f$, the root moves, $x^* \to x^* + \delta x$, and may be approximated as

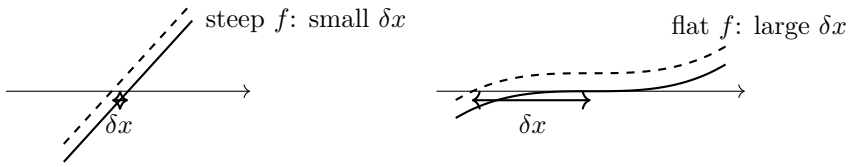
$$|\delta x| \approx \frac{1}{|f'(x^*)|} \, |\delta f| \, .$$

To see this, let $g = f + \delta f$ with root $\hat{x} = x^* + h$:

$$0 = g(\hat{x}) = (\underbrace{f(x^*)}_{0} + hf'(x^*) + O(h^2)) + \delta f = \delta f + hf'(x^*) + O(h^2),$$

so

$$h = \hat{x} - x^* = -\frac{\delta f}{f'(x^*)} + O(h^2).$$



Algorithms for 1D nonlinear equations

Bisection method. Find $x^* \in [a, b]$ such that $f(x^*) = 0$. Return an endpoint if it is already a root. Otherwise, assume:

1. f is continuous on $[a, b]$;
2. $f(a)$ and $f(b)$ have opposite signs, so that f has at least one root in $[a, b]$ by the intermediate value theorem.

Algorithm: evaluate $f(m)$ at $m = (a + b)/2$. If $f(m) = 0$, return m; otherwise retain the half-interval whose endpoints have opposite signs. Uniqueness is not required.

Stopping criterion. For a prescribed absolute root tolerance $\tau > 0$, stop when

$$\frac{b - a}{2} \leq \tau, \qquad \text{and return } m = \frac{a + b}{2},$$

since $|m - x^*| \leq (b - a)/2$ for a root in the retained interval. A test $|f(m)| \leq \tau_f$ controls the residual instead and need not give a small root error when f is flat near the root. For example, $f(x) = 10^{-12}(x - 1)$ has $|f(0)| = 10^{-12}$, but the root error at 0 is 1.

Convergence and cost: the interval width and its midpoint error bound halve each step (linear convergence with $C = \frac{1}{2}$). After k halvings of the initial interval $[a_0, b_0]$,

$$|m_k - x^*| \leq \frac{b_0 - a_0}{2^{k+1}}.$$

Reusing endpoint values, each halving needs one new function evaluation.

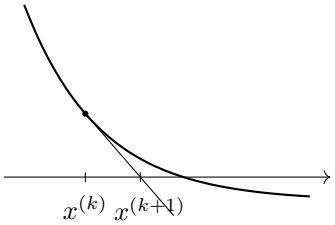
Newton's method. Use calculus: linearize f at the iterate (guess) $x^{(k)}$,

$$f(x) \approx \underbrace{f(x^{(k)}) + (x - x^{(k)})f'(x^{(k)})}_{\hat{f}(x)},$$

and solve $\hat{f}(x) = 0$ for x—the next iterate is the root of the tangent line of f at $x^{(k)}$:

$$f(x^{(k)}) = (x^{(k)} - x)f'(x^{(k)}) \quad \Longrightarrow \quad x = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})},$$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}. \tag{17.1}$$



Convergence rate of Newton's method. It is convenient to think of (17.1) as a *fixed-point iteration*: with

$$g(x) = x - \frac{f(x)}{f'(x)},$$

we want x^* such that $g(x^*) = x^*$ (so $x^{(k+1)} = x^{(k)}$, or up to tolerance), and we study the convergence of $x^{(k+1)} = g(x^{(k)})$.

Theorem. Assume g is sufficiently smooth near x^*. Fixed-point iteration converges at least linearly, locally (when close enough to the fixed point x^*, $g(x^*) = x^*$), so long as $|g'(x^*)| < 1$; if $g'(x^*) = 0$, convergence is at least quadratic.

Why: Taylor expansion of g at x^* with $h = x^{(k)} - x^*$:

$$g(x^* + h) = g(x^*) + hg'(x^*) + \tfrac{h^2}{2}g''(x^*) + O(h^3) = x^* + hg'(x^*) + \tfrac{h^2}{2}g''(x^*) + O(h^3).$$

If $|g'(x^*)| < 1$:

$$|g(x^* + h) - x^*| = |hg'(x^*)| + O(h^2) < |h| \quad \text{for sufficiently small nonzero } h,$$

since $|g'(x^*)| < 1$. If $g'(x^*) = 0$:

$$|g(x^* + h) - x^*| = O(h^2),$$

the error is $O(h^2)$—at least quadratic convergence.

Newton's method analysis: for $g(x) = x - f(x)/f'(x)$,

$$g'(x) = \underbrace{1 - \frac{f'(x)}{f'(x)}}_{0} + \frac{f(x)}{f'(x)^2} \cdot f''(x); \tag{17.2}$$

For a simple root, $f(x^*) = 0$ and $f'(x^*) \neq 0$, so $g'(x^*) = 0$: convergence is locally at least quadratic for sufficiently smooth f.

Newton at a multiple root. Suppose $f(x) = (x - x^*)^m u(x)$ near x^*, where $m \geq 2$ is an integer, u is continuously differentiable, and $u(x^*) \neq 0$. Such a root has *multiplicity* m. For $e_k = x^{(k)} - x^* \neq 0$, Newton's update gives

$$\frac{e_{k+1}}{e_k} = 1 - \frac{u(x^{(k)})}{mu(x^{(k)}) + e_k u'(x^{(k)})} \longrightarrow 1 - \frac{1}{m} \qquad \text{as } e_k \to 0.$$

In a sufficiently small neighborhood the denominator is nonzero and the absolute error ratio is bounded by a constant below 1. Thus nearby starting values converge linearly with factor $1 - 1/m$; a double root has factor $1/2$. This needs finite multiplicity, not merely $f'(x^*) = 0$. Stop if an exact root is reached.

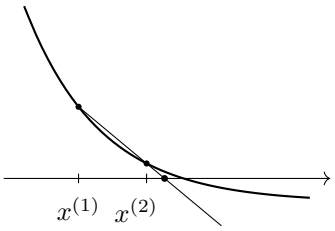
Secant method. What if $f'(x)$ is unavailable? Use

$$f'(x^{(k)}) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} \tag{17.3}$$

in (17.1). Start from two distinct guesses $x^{(0)}$ and $x^{(1)}$, and for $k \geq 1$ compute

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}. \tag{17.4}$$

This is the root of the line through the two most recent points $(x^{(k-1)}, f(x^{(k-1)}))$ and $(x^{(k)}, f(x^{(k)}))$:



For sufficiently smooth f and two starting values close enough to a *simple root* ($f(x^*) = 0$, $f'(x^*) \neq 0$), convergence is superlinear, with the usual order

$$r = (1 + \sqrt{5})/2 \approx 1.618 \quad \text{(the golden ratio)}.$$

Reuse the two stored function values: after the initial evaluations, each new iterate needs only one new evaluation of f, and no derivative evaluation. Newton's method evaluates both f and f' at each step, so the higher convergence order need not mean less total work.

If $f(x^{(k)}) = 0$, stop. If $f(x^{(k)}) = f(x^{(k-1)}) \neq 0$, the secant update is undefined. Unlike bisection, ordinary secant iteration does not preserve a sign-changing bracket and is not guaranteed to converge from arbitrary starting values.

Higher-degree and inverse interpolation. Secant fits a line through two points. Fitting a higher-degree polynomial through more points can give faster local convergence, but computing and selecting its roots can be difficult; real data need not give real roots. Quadratic interpolation of f gives *Muller's method*.

Inverse interpolation instead fits x as a function of $y = f(x)$. For *inverse quadratic interpolation*, use three recent iterates with distinct values $y_j = f(x^{(j)})$, and fit a polynomial p of degree at most two with

$$p(y_j) = x^{(j)}, \quad j = k-2, k-1, k, \qquad x^{(k+1)} = p(0).$$

We evaluate $p(0)$ rather than solve a polynomial equation. The usual local order near a sufficiently smooth simple root is approximately 1.839, with one new function evaluation per step (reusing two old values). Equal function values make the interpolant undefined; a real candidate need not stay inside a bracket or converge. See Heath, Section 5.5.5.

CS 450: Numerical Analysis – Lecture 18

Nonlinear Systems and Introduction to Optimization

Edgar Solomonik

October 29, 2026

Today

Goal: solve nonlinear systems and introduce unconstrained optimization.

- Jacobian, linearization, and sensitivity of roots
- Newton's method, local convergence, and Broyden updates
- Critical points and Hessian tests for local minima
- Golden section search: interval reduction, reuse, and convergence

The nonlinear-system material closes Chapter 5; the optimization material continues in Chapter 6.

n-D nonlinear solve

Given $f : \mathbb{R}^n \to \mathbb{R}^n$, find $x \in \mathbb{R}^n$ such that

$$f(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Sensitivity. In the 1-D case, the local absolute condition number of a simple root x^* is $\kappa = 1/\left|f'(x^*)\right|$. The n-D case needs a generalization of $f'(x)$: consider the *Jacobian matrix*,

$$J_f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix}.$$

If $f : \mathbb{R}^n \to \mathbb{R}^n$ is twice continuously differentiable near y, its first-order Taylor expansion is

$$f(x) = \underbrace{f(y) + J_f(y)\,(x - y)}_{\substack{\text{linearization of } f \text{ at } y \text{ (the affine} \\ \text{first-order Taylor approximation)}}} + O(\|x - y\|_2^2).$$

The local absolute condition number for function *evaluation* at x is $\|J_f(x)\|_2$. For a root x^* with nonsingular $J_f(x^*)$, a small constant perturbation $\widetilde{f}(x) = f(x) + b$ changes the nearby root by approximately

$$\delta x \approx -J_f(x^*)^{-1} b, \qquad \kappa_{\text{abs}} = \left\|J_f(x^*)^{-1}\right\|_2.$$

The inverse-Jacobian formula requires nonsingularity. If $J_f(x^*)v = 0$ for a nonzero v, the function has no first-order change in that direction; higher-order terms may still isolate the root.

Algorithms: Newton's method

Recall the 1-D fixed-point formulation

$$g(x) = x - \frac{f(x)}{f'(x)},$$

which finds the zero of the tangent line $\hat{f}(x) = f(x_k) + f'(x_k)(x - x_k)$.

In n-D, find the zero of the linearization

$$\hat{f}(x) = f(x_k) + J_f(x_k)\,(x - x_k) :$$

Setting $s_k = x_{k+1} - x_k$, compute the zero of this model by a linear solve, rather than forming an inverse:

$$J_f(x_k) s_k = -f(x_k), \qquad x_{k+1} = x_k + s_k. \tag{18.1}$$

Newton n-D fixed-point function.

$$g(x) = x - J_f^{-1}(x)\, f(x) \qquad (\text{if } J_f^{-1}(x) \text{ exists}).$$

If f is twice continuously differentiable near a root x^* and $J_f(x^*)$ is nonsingular, then $g(x^*) = x^*$ and $J_g(x^*) = 0$. Newton's method converges locally at least quadratically: the starting point must be sufficiently close to x^*. If the Jacobian is singular at the root, this guarantee does not apply: convergence may be slower, or the iteration may fail. Unlike in the scalar finite-multiplicity case, analyticity alone does not guarantee linear convergence for a singular system.

Secant updating methods in n-D

The analogue of the 1-D secant equation (cf. the secant figure of Lecture 17) is an approximation to the Jacobian: $J_f(x_k) \approx B_k$ such that

$$B_k\,(x_k - x_{k-1}) = f(x_k) - f(x_{k-1}) \tag{18.2}$$

—n equations for n^2 unknowns. Assume

$$B_k = B_{k-1} + \delta B_k, \qquad \operatorname{rank}(\delta B_k) \le 1, \quad \delta B_k = uv^T.$$

Sherman–Morrison (Lecture 8) allows the efficient inverse update $B_{k-1}^{-1} \to B_k^{-1}$ when its denominator is nonzero; nonsingularity is not automatically preserved. For numerical stability, updating a factorization of B_k (such as QR) is preferred to updating its inverse. A rank-one factorization update also costs $O(n^2)$ for a dense $n \times n$ matrix.

Broyden's method. With $\delta f = f(x_k) - f(x_{k-1})$ and $\delta x = x_k - x_{k-1} \neq 0$, choose

$$\delta B_k = \frac{(\delta f - B_{k-1}\,\delta x)}{\|\delta x\|_2^2}\,\delta x^T \tag{18.3}$$

(so that $B_k\,\delta x = B_{k-1}\,\delta x + \delta f - B_{k-1}\,\delta x = \delta f$, satisfying (18.2)). The secant equation alone does not uniquely determine B_k. This update also leaves its action on directions orthogonal to δx unchanged:

$$\delta x^T z = 0 \quad \Longrightarrow \quad B_k z = B_{k-1} z.$$

Given x_k and a nonsingular approximation B_k, compute the next iterate by

$$B_k s_k = -f(x_k), \qquad x_{k+1} = x_k + s_k.$$

The new function value supplies the difference for the next update. An initial B_0 can be the Jacobian at x_0 or an approximation to it. Update an inverse or factorization instead of refactorizing B_k: for dense matrices this reduces the linear-algebra work from $O(n^3)$ to $O(n^2)$ per iteration, after initialization and when the updates are well-defined. Function-evaluation costs are additional; the full Jacobian need not be reevaluated at every step.

Practical considerations. Newton or Broyden can take large, unreliable steps. A damped step $x_{k+1} = x_k + \alpha_k s_k$ can seek sufficient decrease in the residual measure $\frac{1}{2}\,\|f(x)\|_2^2$. A trust region restricts the step to where the local model is expected to be reliable and can change its direction as well as its length. These safeguards do not guarantee convergence to a root from every starting point. A stopping test such as $\|f(x_k)\|_2 \le \tau$ checks the residual; a small change in the iterates alone may indicate stagnation. For an ill-conditioned problem, the error in the approximate location of the root may be much larger in magnitude than the small residual.

AI-use disclosure. *These notes were prepared with the assistance of Claude and ChatGPT, including transcription from handwritten lecture notes and drafting or editing of prose, examples, and derivations. The author reviewed the final document and assumes responsibility for its content.*