# CS 450: Numerical Anlaysis[1]

## Nonlinear Equations

University of Illinois at Urbana-Champaign

[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

# Solving Nonlinear Equations

- Solving (systems of) nonlinear equations corresponds to root finding:
  - $f(x^*) = 0$
  - $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$
  - Algorithms for root-finding make it possible to solve systems of nonlinear equations and employ a similar methodology to finding minima in optimization.
- Main algorithmic approach: find successive roots of local linear approximations of $\boldsymbol{f}$:

# Nonexistence and Nonuniqueness of Solutions

- Solutions do not generally exist and are not generally unique, even in the univariate case:

- Solutions in the multivariate case correspond to intersections of hypersurfaces:

# Conditions for Existence of Solution

- *Intermediate value theorem* for univariate problems:

- A function has a unique *fixed point* $\boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{x}^*$ in a given closed domain if it is *contractive* and contained in that domain,

$$||\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{z})|| \leq \gamma ||\boldsymbol{x} - \boldsymbol{z}||$$

# Conditioning of Nonlinear Equations

- Generally, we take interest in the absolute rather than relative conditioning of solving $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$:

- The *absolute condition number* of finding a root $x^*$ of $f$ is $1/|f'(x^*)|$ and for a root $\boldsymbol{x}^*$ of $\boldsymbol{f}$ it is $||\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}^*)||$:

# Multiple Roots and Degeneracy

- If $x^*$ is a root of $f$ with *multiplicity* $m$, its $m-1$ derivatives are also zero at $x^*$,

$$f(x^*) = f'(x^*) = f''(x^*) = \cdots = f^{(m-1)}(x^*) = 0.$$

- Increased multiplicity affects conditioning and convergence:

# Bisection Algorithm

- Assume we know the desired root exists in a bracket $[a, b]$ and $sign(f(a)) \neq sign(f(b))$:

- Bisection subdivides the interval by a factor of two at each step by considering $f(c_k)$ at $c_k = (a_k + b_k)/2$:

# Rates of Convergence

- Let $\boldsymbol{x}_k$ be the $k$th iterate and $\boldsymbol{e}_k = \boldsymbol{x}_k = \boldsymbol{x}^*$ be the error, bisection obtains *linear convergence*, $\lim_{k\to\infty} ||\boldsymbol{e}_k||/||\boldsymbol{e}_{k-1}|| \leq C$ where $C < 1$:

- $r$th order convergence implies that $||\boldsymbol{e}_k||/||\boldsymbol{e}_{k-1}||^r \leq C$

# Convergence of Fixed Point Iteration

- Fixed point iteration: $x_{k+1} = g(x_k)$ is locally linearly convergent if for $x^* = g(x^*)$, we have $|g'(x^*)| < 1$:

- It is quadratically convergent if $g'(x^*) = 0$:

# Newton's Method

***Demo:*** *Newton's Method*
***Demo:*** *Convergence of Newton's Method*

- Newton's method is derived from a *Taylor series* expansion of $f$ at $x_k$:

- Newton's method is *quadratically convergent* if started sufficiently close to $x^*$ so long as $f'(x^*) \neq 0$:

# Secant Method

**Demo:** *Secant Method*
**Demo:** *Convergence of the Secant Method*

- The *Secant method* approximates $f'(x_k) \approx \frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$:

- The convergence of the Secant method is *superlinear* but not quadratic:

# Nonlinear Tangential Interpolants

- Secant method uses a linear interpolant based on points $f(x_k)$, $f(x_{k-1})$, could use more points and higher-order interpolant:

- Quadratic interpolation (Muller's method) achieves convergence rate $r \approx 1.84$:

- Inverse quadratic interpolation resolves the problem of nonexistence/nonuniqueness of roots of polynomial interpolants:

# Achieving Global Convergence

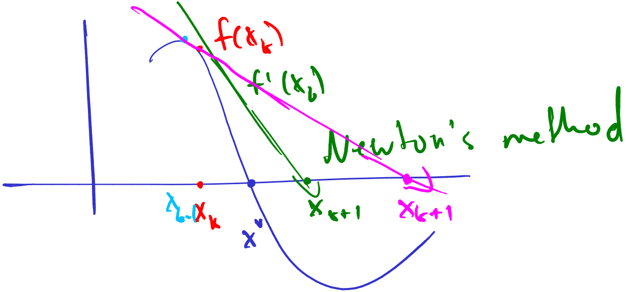
- Hybrid bisection/Newton methods:

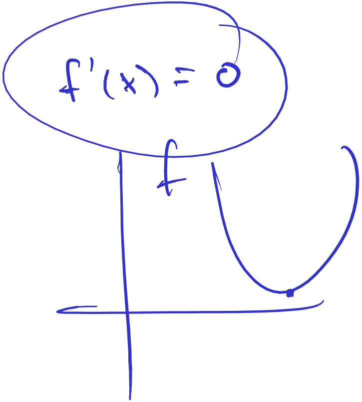
- Bounded (damped) step-size:

# Nonlinear Solve

$f(x^*) = 0$

$x_{k+1} = g(x_k)$



# Systems of Nonlinear Equations

$f(x) \in \mathbb{R}^n \to \mathbb{R}$, $\nabla f(x) \in \mathbb{R}^n \to \mathbb{R}^n$

- Given $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) & \cdots & f_m(\boldsymbol{x}) \end{bmatrix}^T$ for $\boldsymbol{x} \in \mathbb{R}^n$, seek $\boldsymbol{x}^*$ so that $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$

$m = n$

- At a particular point $\boldsymbol{x}$, the *Jacobian* of $\boldsymbol{f}$, describes how $\boldsymbol{f}$ changes in a given direction of change in $\boldsymbol{x}$,

$$\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \begin{bmatrix} \frac{df_1}{dx_1}(\boldsymbol{x}) & \cdots & \frac{df_1}{dx_n}(\boldsymbol{x}) \\ \vdots & & \vdots \\ \frac{df_m}{dx_1}(\boldsymbol{x}) & \cdots & \frac{df_m}{dx_n}(\boldsymbol{x}) \end{bmatrix}$$

$x_{k+1} = x_k + s_k$

$J_f(x_k) s_k = -f(x_k)$

center

$f(x+s) = f(x) + J_f(x) s + \cdots$

$\Rightarrow 0 = f(x) + J_f(x) s$

# Multivariate Newton Iteration

***Demo:** Newton's method in n dimensions*

- Fixed-point iteration $\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$ achieves local convergence so long as $|\lambda_{\max}(\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}^*))| < 1$ and quadratic convergence if $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}^*) = \boldsymbol{O}$:

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

$$e_k = x^* - x_k$$
$$= g(x^*) - g(x_{k-1})$$
$$= J_g(x^*)(x^* - x_{k-1}) + \dots$$

h.o. diff of g $\|x^* - x_{k-1}\|^2$

$$\|e_k\| \le \|J_g(x^*)\|_2 \|e_{k-1}\| + \dots$$

# Multidimensional Newton's Method

- Newton's method corresponds to the fixed-point iteration

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$$

$$J_g(x^*) = I - J_f^{-1}(x^*) J_f(x^*) - \sum_{i=1}^{n} H_{f_i}(x^*) f_i(x^*)$$

- Quadratic convergence is achieved when the Jacobian of a fixed-point iteration is zero at the solution, which is true for Newton's method:

# Estimating the Jacobian using Finite Differences

- To obtain $\boldsymbol{J_f}(\boldsymbol{x}_k)$ at iteration $k$, can use finite differences:

$$f : R \to R^n$$

$$J_f(x) \approx \frac{f(x+h) - f(x)}{h}$$

$$f(x+he_i)$$

$$J_f(x) = \begin{bmatrix} f_1'(x) \\ \vdots \\ f_n'(x) \end{bmatrix}$$

$$f_i'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- $n+1$ function evaluations are needed: $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x} + h\boldsymbol{e}_i), \forall i \in \{1, \ldots, n\}$, which correspond to $m(n+1)$ scalar function evaluations if $\boldsymbol{J_f}(\boldsymbol{x}_k) \in \mathbb{R}^{m \times n}$.

$f: \mathbb{R}^n \to \mathbb{R}^n$

let $j_i$ be the $i$th column of $J_f(x)$

$$j_i \approx \frac{f(x + e_i h) - f(x)}{h}$$

# Cost of Multivariate Newton Iteration

- What is the cost of solving $J_{\boldsymbol{f}}(\boldsymbol{x}_k)\boldsymbol{s}_k = \boldsymbol{f}(\boldsymbol{x}_k)$?

PLU

$O(n^3)$ at each iteration

$J_f(x_k) \neq J_f(x_{k+1})$

- What is the cost of Newton's iteration overall?

k steps

$\gamma$ cost of evaluation of $f_i$

$O(kn^3 + kn^2\gamma)$

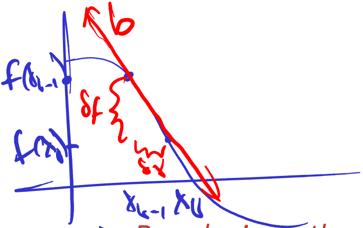
# Quasi-Newton Methods

In solving a nonlinear equation, seek approximate Jacobian $J_f(x_k)$ for each $x_k$

- Find $B_{k+1} = B_k + \delta B_k \approx J_f(x_{k+1})$, so as to approximate *secant equation*

$b\,\delta x = \delta f$

$$B_{k+1}(\underbrace{x_{k+1} - x_k}_{\delta x}) = \underbrace{f(x_{k+1}) - f(x_k)}_{\delta f}$$




- *Broyden's method* solves the secant equation and minimizes $||\delta B_k||_F$:

$$\delta B_k = \frac{\delta f - B_k \delta x}{||\delta x||^2}\delta x^T$$

$B_{k+1}\delta x = B_k \delta x + \delta B_k \delta x = B_k \delta x + \delta f - B_k \delta x$

$O(n^2)$ cost / iteration excl. func. eval.

# Safeguarding Methods

- Can dampen step-size to improve reliability of Newton or Broyden iteration:

converges quadratically ↳ local

$$J_f(x_k)\, s_k = f(x_k)$$

$$x_{k+1} = x_k + \alpha_k s_k$$, e.g., choose $\alpha_k$ so that $\|f(x_{k+1})\| < \|f(x_k)\|$

- *Trust region methods* provide general step-size control:

↓ describes region in which our derivative approximation is reliable