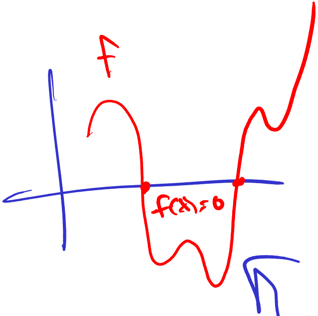
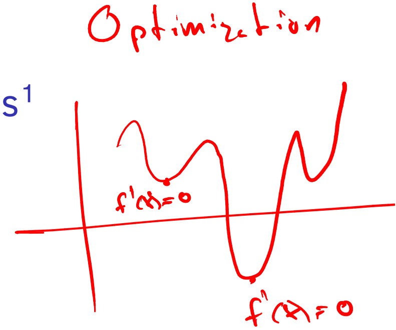
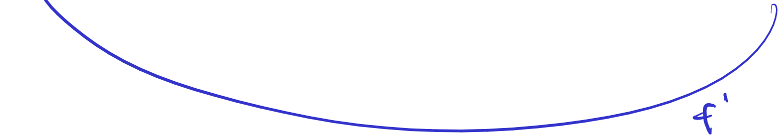
CS 450: Numerical Anlaysis[1]

Nonlinear Equations

University of Illinois at Urbana-Champaign



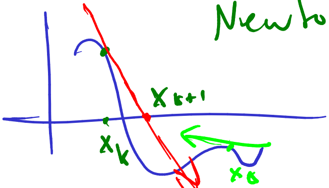




[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

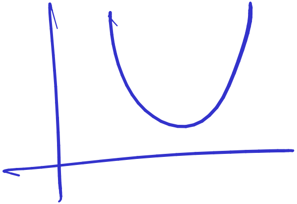
Solving Nonlinear Equations

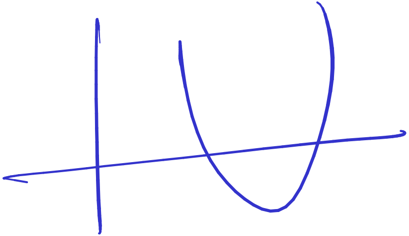
- Solving (systems of) nonlinear equations corresponds to root finding:
  - $f(x^*) = 0$
  - $f(\boldsymbol{x}^*) = 0$
  - $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$
  - Algorithms for root-finding make it possible to solve systems of nonlinear equations and employ a similar methodology to finding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



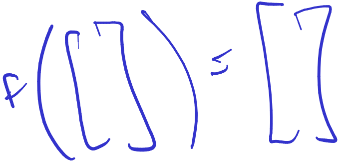
Nonexistence and Nonuniqueness of Solutions

- 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:



$$f_1(x) = 0 \qquad 2x_1 = 0$$

$$\vdots \qquad \vdots$$

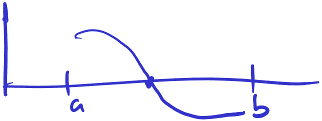
$$f_n(x) = 0 \qquad 3x_2 + x_3 = 0$$

Conditions for Existence of Solution

- *Intermediate value theorem* for univariate problems:

bracket $[a,b]$, $\mathrm{sign}(f(a)) \neq \mathrm{sign}(f(b))$

f is continuous



$\Rightarrow \exists\, x^* \in [a,b],\ f(x^*) = 0$

- 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, $\exists\, \gamma$

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

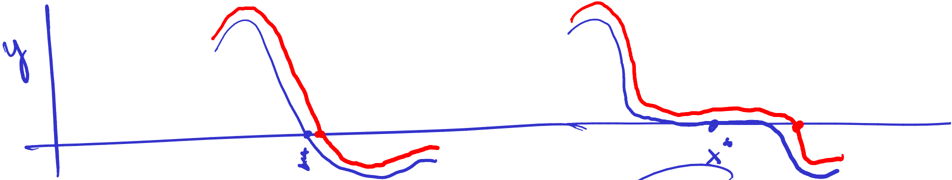
Lipschitz continuous

$g(x^*) = x^* \Rightarrow f(x^*) = 0$

$g(x) = f(x) + x$

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^*,

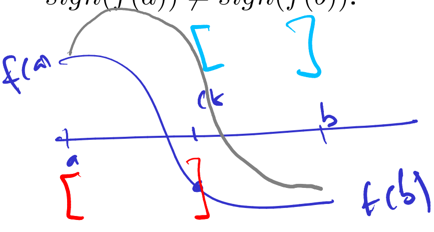
$$f(x^*) = f'(x^*) = f''(x^*) = \cdots = f^{(m-1)}(x^*) = 0.$$

- Increased multiplicity affects conditioning and convergence:

$\kappa = 1/f'(x^*) = \infty$

Bisection Algorithm

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



$$e_k = c_k - x^*$$

$$|e_k| \leq \frac{1}{2^k}(b_0 - a_0)$$

- 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 x_k be the kth iterate and $e_k = x_k = x^*$ be the error, bisection obtains *linear convergence*, $\lim_{k\to\infty} ||e_k||/||e_{k-1}|| \leq C$ where $C < 1$:

for relative accuracy ϵ

$\sim \log_C(\frac{1}{\epsilon})$ steps

$\epsilon = 10^{-10}$ $\log_C(\frac{1}{\epsilon})$ ~ 10

- rth order convergence implies that $||e_k||/||e_{k-1}||^r \leq C$

quadratic $r = 2$

superlinear $1 < r < 2$

cubic $r = 3$

$O(\log_r(\log(\frac{1}{\epsilon})))$

$\log(||e_k||) \approx r \log(||e_{k-1}||)$

increase # digits of accuracy by a factor of r at each step

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

$$e_{k+1} = x_{k+1} - x^* = g(x_k) - g(x^*)$$

$$g(x^* + \delta x) \approx g(x^*) + g'(x^*)\delta x + O(|\delta x|^2)$$

$$e_{k+1} = g'(x^*)(x_k - x^*) + O(|x_k - x^*|^2)$$

$$= g'(x^*) e_k + O(|x_k - x^*|^2)$$

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

$$e_{k+1} = g''(x^*) e_k^2 / 2 + O(|x^* - x_k|^3)$$

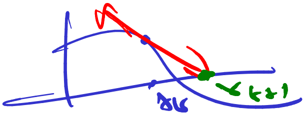
Newton's Method

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:

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + (1/2) f''(x_k)(x - x_k)^2 + \dots$$

secant line approximation



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

$$g(x) = x - f(x)/f'(x)$$

quad. conv. if $g'(x^*) = 0$

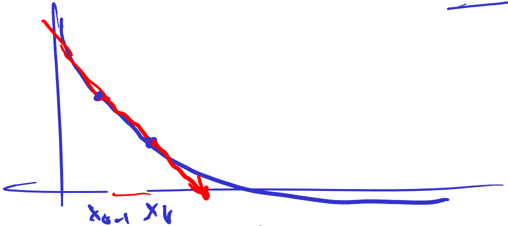
$$g'(x^*) = 1 - f'(x^*)/f'(x^*) + \frac{f(x^*) f''(x^*)}{f'(x^*)^2} = 0$$

Secant Method

Demo: *Secant Method*

Demo: *Convergence of the Secant Method*

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

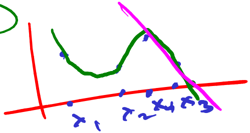


- The convergence of the Secant method is *superlinear* but not quadratic:

Nonlinear Tangential Interpolants

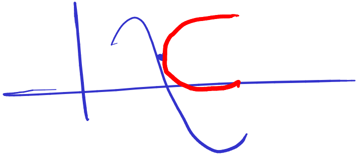
- Secant method uses a linear interpolant based on points $f(x_k)$, $f(x_{k-1})$, could use more points and higher-order interpolant:

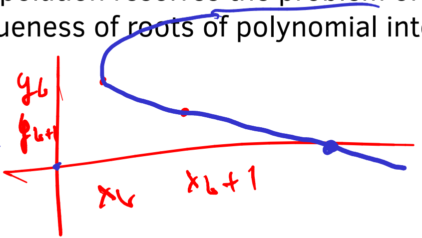
$(x_0, f(x_0)) \ldots (x_k, f(x_k))$



- 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:





(y_k, x_k)

(y_{k+1}, x_{k+1})

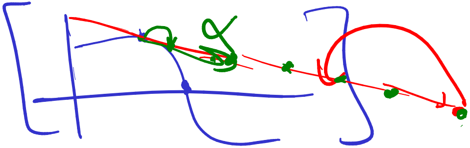
$\tilde{f}(y) = x$

$x^* \approx \tilde{f}(0)$

Achieving Global Convergence

- Hybrid bisection/Newton methods:

Newton with safeguarding using a bracket



- Bounded (damped) step-size:

Systems of Nonlinear Equations

- 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 $\boldsymbol{f}(\boldsymbol{x}^*) = \mathbf{0}$

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

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

Demo: Newton's method in n dimensions

Multivariate Newton Iteration

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

Multidimensional Newton's Method

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

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

- 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:

- $n+1$ 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 $\boldsymbol{J_f}(\boldsymbol{x}_k) \in \mathbb{R}^{m \times n}$.

Cost of Multivariate Newton Iteration

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

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

Quasi-Newton Methods

In solving a nonlinear equation, seek approximate Jacobian $\boldsymbol{J_f}(\boldsymbol{x}_k)$ for each $\boldsymbol{x}_k$

- Find $\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \boldsymbol{\delta B}_k \approx \boldsymbol{J_f}(\boldsymbol{x}_{k+1})$, so as to approximate *secant equation*

$$\boldsymbol{B}_{k+1}\underbrace{(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)}_{\boldsymbol{\delta x}} = \underbrace{\boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}(\boldsymbol{x}_k)}_{\boldsymbol{\delta f}}$$

- *Broyden's method* solves the secant equation and minimizes $||\boldsymbol{\delta B}_k||_F$:

$$\boldsymbol{\delta B}_k = \frac{\boldsymbol{\delta f} - \boldsymbol{B}_k\boldsymbol{\delta x}}{||\boldsymbol{\delta x}||^2}\boldsymbol{\delta x}^T$$

Safeguarding Methods

- Can dampen step-size to improve reliability of Newton or Broyden iteration:
- *Trust region methods* provide general step-size control: