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

Demo: *Three quadratic functions*

- 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 $c_k = (a_k + b_k)/2$:

***Demo:** Rates of Convergence*

Rates of Convergence

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

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

Review

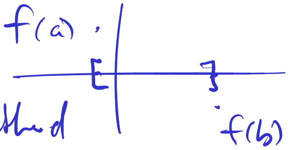
$f(x) = 0$

vector?

vector?
vector-valued

Bisection (1D)

- starts with a bracket
- subdivides it in half



Newton's method

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

↑ fixed-point function

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

$|g'(x^*)| < 1 \Rightarrow$ convergence

$= 0 \Rightarrow$ quadratic conv.

Convergence rates

linear: $O(1)$ digits/step

quadratic: double digits/step

superlinear (rate r): r times $\frac{\#\text{ digits}}{\text{step}}$

$r = 2$

Convergence of Fixed Point Iteration

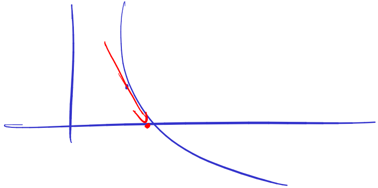
- Fixed point iteration: $x_{k+1} = g(x_k)$ is locally linearly convergent if for $x^* = g(x^*)$, we have $|g'(x^*)| < 1$:

- 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:



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

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

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

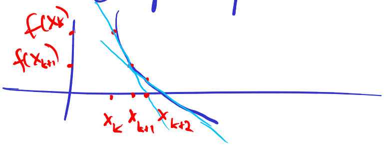
$$= 0 \implies \text{quadratic convergence}$$

Secant Method

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

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

cheapest possible



can instead consider x_k and $x_k + h$

and use

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

should not pick h too small ($> \epsilon_{mach}$)

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

roughly $e_k = x_k - x^*$

$\Rightarrow$ $e_k = e_{k-1} e_{k-2}$

$\log(e_k) = \log(e_{k-1}) + \log(e_{k-2})$ $\Rightarrow$ Fibonacci sequence

grows at a rate of $(1+\sqrt{5})/2 \approx 1.6$

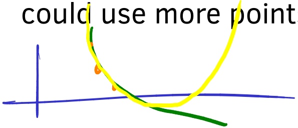
Golden ratio

$-\log(e) \Rightarrow$ # digits of accuracy

$O(\log_r(\log(1/\epsilon)))$

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:

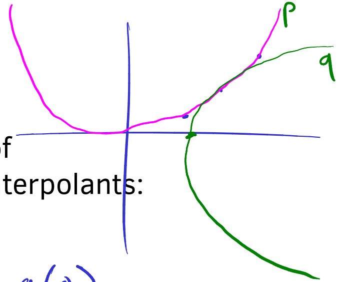


may not a have a root

have multiple roots

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

better than secant



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

$p \rightarrow$

x_k	$f(x_k)$
x_{k-1}	$f(x_{k-1})$
x_{k-2}	$f(x_{k-2})$

$q \leftarrow$

x_k	y_k
x_{k-1}	y_{k-1}
x_{k-2}	y_{k-2}

$q(y_k) = x_k$

$q(y_{k-1}) = x_{k-1}$

$q(y_{k-2}) = x_{k-2}$

so $x^* \approx q(0)$

~~Achieving Global Convergence~~ Safeguarding

- Hybrid bisection/Newton methods:

given a bracket, bisection is always reliable
but Newton is not

however, Newton converges faster
so run Newton steps only if they stay
within bracket, otherwise run bisection

- Bounded (damped) step-size:

use Newton as directional pointer

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

$$\alpha < 1$$

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

  if # equations = # unknowns, $m = n$

  $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

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

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

$J_{ij}(x) = \frac{df_i}{dx_j}(x)$ $\quad f(x+\delta x) \approx f(x_k) + J_f(x_k)\,\delta x$

Multivariate Newton Iteration

$g(x^*) = x^*$

- 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 $\boxed{\boldsymbol{J}_{\boldsymbol{g}}(x^*) = \boldsymbol{O}}$:

$e_{k+1} = x_{k+1} - x^* = g(x_k) - \cancel{g(x^*)}$, then $g(x_k) =$ Taylor at x^*

$= \cancel{g(x^*)} + \dots$

$= \underbrace{J_g(x^*)}_{O}(x_k - x^*) + \frac{1}{2}\left[H_g^{(1)}(x^*)(x_k - x^*) \dots H_g^{(n)}(x^*)(x_k - x^*)\right](x_k - x^*)$

$\|e_{k+1}\|_2 = O\left(\max_i \|H_g^{(i)}(x)\|_2 \cdot \|\underbrace{x_k - x^*}_{e_k}\|_2^2\right)$

$= O(\|e_k\|_2^2)$

Multidimensional Newton's Method

$f(x) = f(x_k) + J_f(x_k)(x - x_k)$

$0 = \dots \quad -f(x_k) = -J_f(x_k)(x - x_k)$

$x = x_k - J_f^{-1}(x_k) f(x_k)$

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

$$g(x) = x - J_f^{-1}(x) f(x)$$

$g'(x) = I - \underbrace{J_f^{-1}(x) J_f(x)}_{I} - \sum_i \underbrace{f_i(x)}_{0} H_f^{(i)}(x)$

$g'(x^*) = I - I - 0$

if $J_f(x^*)$ is singular (quadratic) convergence is not guaranteed

- 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:

$$J_f(x_k) = \begin{bmatrix} j_1 & \cdots & j_n \end{bmatrix}$$

$$j_i = \frac{f(x_k + he_i) - f(x_k)}{h}$$

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

Cost of Multivariate Newton Iteration

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

$O(n^3)$

$x_{k+1} - x_k$

for k-steps

$O(n^3 k)$

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

$n^2 \cdot k$ function eval's to compute $J_f(x_k)$

$n \cdot k$ function eval's of f_i's

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*

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

B_{k+1} is linear interpolant

x_k	$f(x_k)$
x_{k+1}	$f(x_{k+1})$

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

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

least change B_k

$\delta B_k =$ δx^T

$\delta f - B_k \delta x$

Safeguarding Methods

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

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

f

$\alpha < 1$

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

current guess $x_k = (a, b)$