CS 450: Numerical Analysis¹
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 (systems of) nonlinear equations corresponds to root finding:

- \( f(x^*) = 0 \)

- \( f(x^*) = 0 \)

- \( f(x^*) = 0 \)

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 \( f \):
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:

Demo: Three quadratic functions
Conditions for Existence of Solution

- **Intermediate value theorem** for univariate problems:
  
  A function has a unique *fixed point* $g(x^*) = x^*$ in a given closed domain if it is *contractive* and contained in that domain, 
  
  $$||g(x) - g(z)|| \leq \gamma ||x - z||$$
Conditioning of Nonlinear Equations

Generally, we take interest in the absolute rather than relative conditioning of solving $f(x) = 0$:

- The absolute condition number of finding a root $x^*$ of $f$ is $1/|f'(x^*)|$ and for a root $x^*$ of $f$ it is $||J_f^{-1}(x^*)||$:
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 
\[\text{sign}(f(a)) \neq \text{sign}(f(b)):\]

- 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 $x_k$ be the $k$th iterate and $e_k = x_k - x^*$ be the error, bisection obtains linear convergence, $\lim_{k \to \infty} \frac{||e_k||}{||e_{k-1}||} \leq C$.

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

Review

\[ f(x) = 0 \]

vector-valued

\[ \text{vector?} \]

vector?

Convergence rates

linear: \( O(1) \) digits/step

quadratic: double digits/step

superlinear (rate \( r \)): \( r \) times \( \log \) step

\[ r = 2 \]

Bisection (1D)

- starts with a bracket
- subdivides it in half

\[ f(a), f(b) \]

Newton's method

fixed-point function

\[ g(x^*) = x^* \Rightarrow f(x^*) = 0 \]

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

\( = 0 \Rightarrow \) quadratic conv.
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

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

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

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

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

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

- **The Secant method** approximates \( f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \): 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 (> \( \varepsilon_{	ext{num}} \)).

- The convergence of the Secant method is **superlinear** but not quadratic:
  - Roughly \( e_k = x_k - x^* \)
  - \( e_k = e_{k-1} e_{k-2} \)
  - \( \log(e_k) = \log(e_{k-1}) + \log(e_{k-2}) \) \( \Rightarrow \) grows at a rate of \( (1 + 55)^{1/2} \approx 1.6 \) **Golden ratio**
  - \( -\log(e) \Rightarrow \# \text{digits of accuracy} \)
    \( \mathcal{O}(\log(\log(1/e))) \)
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:
  - 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 \( f(x) = [f_1(x) \cdots f_m(x)]^T \) for \( x \in \mathbb{R}^n \), seek \( x^* \) so that \( f(x^*) = 0 \)

\[
\text{if } \# \text{ equations } = \# \text{ unknowns, } m = n
\]

- At a particular point \( x \), the **Jacobian** of \( f \), describes how \( f \) changes in a given direction of change in \( x \),

\[
J_f(x) = \begin{bmatrix}
\frac{df_1}{dx_1}(x) & \cdots & \frac{df_1}{dx_n}(x) \\
\vdots & \ddots & \vdots \\
\frac{df_m}{dx_1}(x) & \cdots & \frac{df_m}{dx_n}(x)
\end{bmatrix}
\]

\[
\frac{d}{dx}f(x) = \frac{df}{dx}(x) \quad f(x + \delta x) \approx f(x_k) + \int_{x_k}^x f'(x) \delta x
\]
Multivariate Newton Iteration

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

$$
\begin{align*}
e_{k+1} &= x_{k+1} - x^* = g(x_k) - g(x^*) \\
&= J_g(x^*)(x_k - x^*) + \frac{1}{2} \left[ H^{(2)}_g(x^*) (x_k - x^*) + \cdots \right] (x_k - x^*)^T \\
&= O \left( \max_i \|H^{(i)}_g(x^*)\|_2 \cdot \|x_k - x^*\|_2^2 \right) e_k \\
&= O(\|e_k\|_2^2)
\end{align*}
$$

Demo: Newton's method in n dimensions
Multidimensional Newton’s Method

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

\[ g(x) = x - \left( J_f^{-1}(x) f(x) \right) \]

\[ g'(x^*) = I - J_f^{-1}(x^*) J_f(x^*) - \sum_i f_i(x^*) H_f^{(i)}(x^*) \]

- 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 $J_f(x_k)$ 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: $f(x)$ and $f(x + he_i)$, $\forall i \in \{1, \ldots, n\}$, which correspond to $m(n + 1)$ scalar function evaluations if $J_f(x_k) \in \mathbb{R}^{m \times n}$. 
Cost of Multivariate Newton Iteration

- What is the cost of solving $J_f(x_k)s_k = f(x_k)$?
  $$O(n^3)$$
  for $k = 1, \ldots, s$
  $$O(n^3k)$$

- What is the cost of Newton’s iteration overall?
  $n^2.6$ function eval’s to compute $J_f(x_k)$
  $n.65$ function eval’s of $f_i$'s
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

\[
B_{k+1}(x_{k+1} - x_k) = f(x_{k+1}) - f(x_k)
\]

\( \delta B_k \) is linear interpolant

\[
\begin{array}{c|c}
  x_k & f(x_k) \\
  x_{k+1} & f(x_{k+1})
\end{array}
\]

- Broyden’s method solves the secant equation and minimizes \( \| \delta B_k \|_F \):

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

Activity: Broyden’s Method
Safeguarding Methods

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

\[ g(x) = x - \alpha \nabla f(x) \]

\[ \alpha < 1 \]

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

\[ x_{k+1} = (a, b) \]

\[ (\text{current guess, } x_k = (a, b)) \]