CS 357: Numerical Methods

Nonlinear Equations

Eric Shaffer

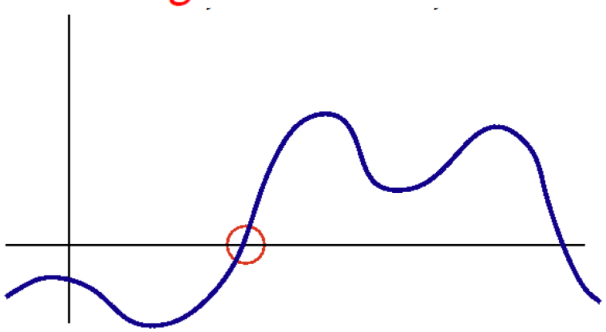
Some slides adapted from:
Scientific Computing: An Introductory Survey, 2nd ed., McGraw-Hill, 2002. By Michael T. Heath

Nonlinear Equations

- Given function f, we seek value x for which

$$f(x) = 0$$

- Solution x is *root* of equation, or *zero* of function f
- So problem is known as *root finding* or *zero finding*



What you should learn….

Goals:

- Find roots to equations
- Compare usability of different methods
- Compare convergence properties of different methods

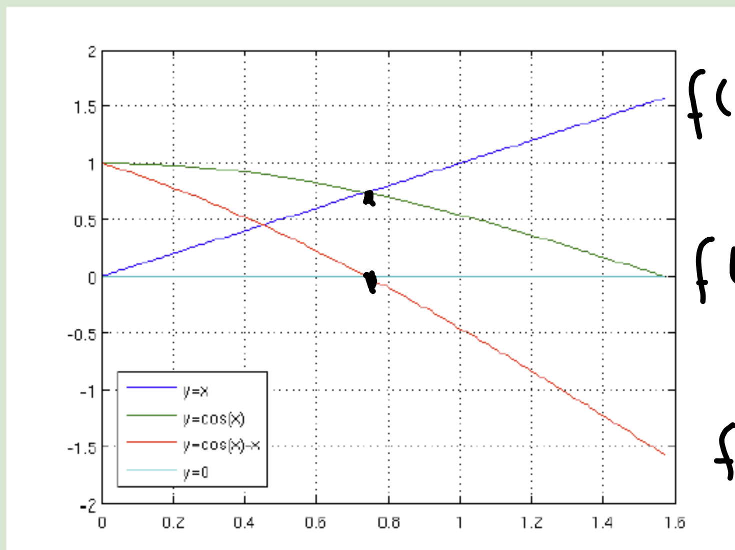
1. bracketing methods
2. Bisection Method
3. Newton's Method
4. Secant Method

Solving Equations as Root-finding

- Any single valued function can be written as $f(x) = 0$

Example

- Find x so that $\cos(x) = x$
- That is, find where $f(x) = \cos(x) - x = 0$



$f(x) = x$

$f(x) = \cos(x)$

$f(x) = \cos(x) - x$

Non-linear Equations

Two important cases

- Single nonlinear equation in one unknown, where

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

  Solution is scalar x for which $f(x) = 0$

- System of n *coupled* nonlinear equations in n unknowns, where

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

  Solution is vector $\boldsymbol{x}$ for which all components of $\boldsymbol{f}$ are zero *simultaneously*, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$

Examples: One Dimension

Nonlinear equations can have any number of solutions

- $\exp(x) + 1 = 0$ has no solution
- $\exp(-x) - x = 0$ has one solution
- $x^2 - 4\sin(x) = 0$ has two solutions
- $x^3 + 6x^2 + 11x - 6 = 0$ has three solutions
- $\sin(x) = 0$ has infinitely many solutions

Examples: Non-linear Equations

- Example of nonlinear equation in one dimension

$$x^2 - 4\sin(x) = 0$$

for which $x = 1.9$ is one approximate solution

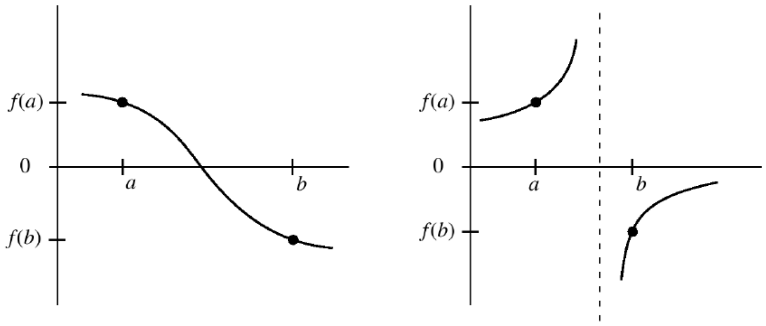
- Example of system of nonlinear equations in two dimensions

$$\begin{aligned} x_1^2 - x_2 + 0.25 &= 0 \\ -x_1 + x_2^2 + 0.25 &= 0 \end{aligned}$$

for which $\boldsymbol{x} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^T$ is solution vector

Existence and Uniqueness

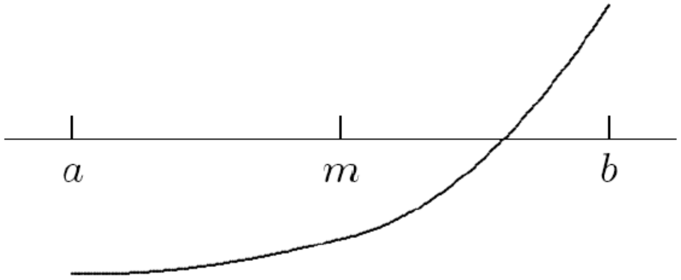
- Existence and uniqueness of solutions are more complicated for nonlinear equations than for linear equations
- For function $f: \mathbb{R} \to \mathbb{R}$, *bracket* is interval $[a, b]$ for which sign of f differs at endpoints
- If f is continuous and $\text{sign}(f(a)) \neq \text{sign}(f(b))$, then Intermediate Value Theorem implies there is $x^* \in [a, b]$ such that $f(x^*) = 0$
- There is no simple analog for n dimensions



Bisection Method

Bisection method begins with initial bracket and repeatedly halves its length until solution has been isolated as accurately as desired

while $((b-a) > tol)$ **do**
    $m = a + (b-a)/2$
    if $\mathrm{sign}(f(a)) = \mathrm{sign}(f(m))$ **then**
        $a = m$
    else
        $b = m$
    end
end



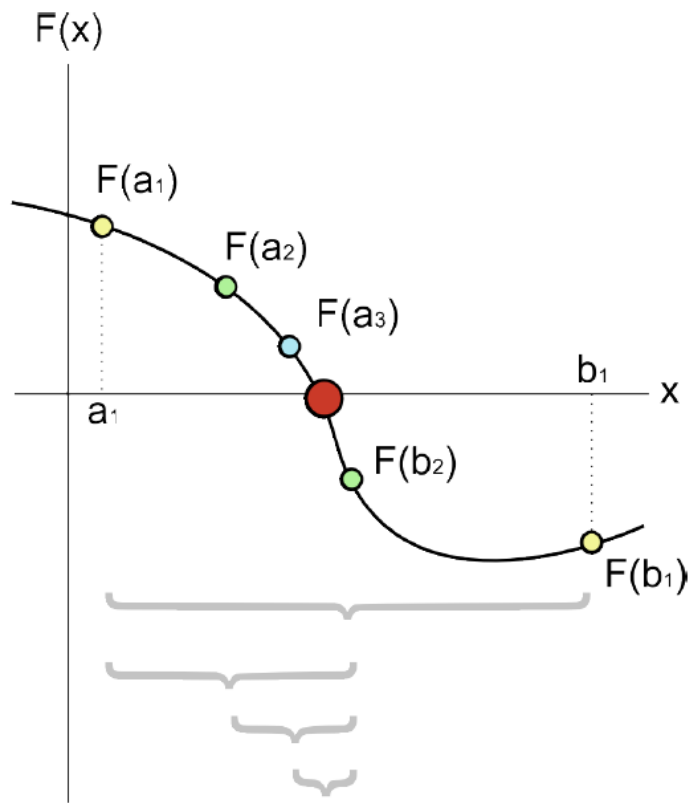
Bisection Method

For the bracket interval $[a, b]$ the midpoint is

$$x_m = \frac{1}{2}(a + b)$$

idea:

1. split bracket in half
2. select the bracket that has the root
3. goto step 1



Convergence Rate

- For general iterative methods, define error at iteration k by

$$\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}^*$$

  where $\boldsymbol{x}_k$ is approximate solution and $\boldsymbol{x}^*$ is true solution

- For methods that maintain interval known to contain solution, rather than specific approximate value for solution, take error to be length of interval containing solution

- Sequence converges with rate r if

$$\lim_{k\to\infty} \frac{\|\boldsymbol{e}_{k+1}\|}{\|\boldsymbol{e}_k\|^r} = C$$

  for some finite nonzero constant C

Convergence Rate

Some particular cases of interest

- $r = 1$: *linear* $\quad (C < 1)$
- $r > 1$: *superlinear*
- $r = 2$: *quadratic*

Convergence rate	Digits gained per iteration
linear	constant
superlinear	increasing
quadratic	double

Convergence Rate

Convergence Rate

1. $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}...$ linear $r=1$ $c=10^{-1}$
2. $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}...$ " " $c=10^{-2}$
3. $10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}...$ superlinear $r=1.X?$
4. $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16}...$ quadratic $r=2$
5. $10^{-2}, 10^{-6}, 10^{-18}, ...$ cubic $r=3$

Convergence Rate For Bisection

$$\lim_{k \to \infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = 1/2$$

Linear

$r = 1 \quad c = 1/2$

Bisection Method

- Bisection method makes no use of magnitudes of function values, only their signs
- Bisection is certain to converge, but does so slowly
- At each iteration, length of interval containing solution reduced by half, convergence rate is *linear*, with $r = 1$ and $C = 0.5$
- One bit of accuracy is gained in approximate solution for each iteration of bisection
- Given starting interval $[a, b]$, length of interval after k iterations is $(b - a)/2^k$, so achieving error tolerance of tol requires

$$\left\lceil \log_2 \left(\frac{b - a}{tol} \right) \right\rceil$$

iterations, regardless of function f involved

Newton's Method

- Truncated Taylor series

$$f(x+h) \approx f(x) + f'(x)h$$

  is linear function of h approximating f near x

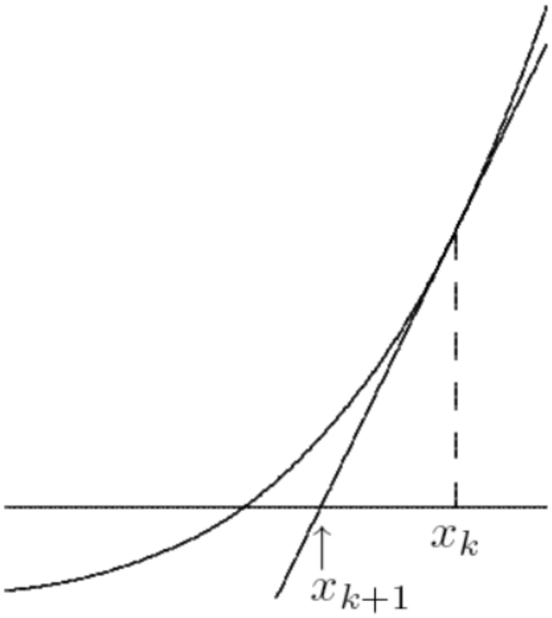
- Replace nonlinear function f by this linear function, whose zero is $h = -f(x)/f'(x)$

- Zeros of original function and linear approximation are not identical, so repeat process, giving *Newton's method*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton's Method

Newton's method approximates nonlinear function f near x_k by *tangent line* at $f(x_k)$



Newton's Method: Example

- Use Newton's method to find root of

$$f(x) = x^2 - 4\sin(x) = 0$$

- Derivative is

$$f'(x) = 2x - 4\cos(x)$$

so iteration scheme is

$$x_{k+1} = x_k - \frac{x_k^2 - 4\sin(x_k)}{2x_k - 4\cos(x_k)}$$

- Taking $x_0 = 3$ as starting value, we obtain

x	$f(x)$	$f'(x)$	h
3.000000	8.435520	9.959970	−0.846942
2.153058	1.294772	6.505771	−0.199019
1.954039	0.108438	5.403795	−0.020067
1.933972	0.001152	5.288919	−0.000218
1.933754	0.000000	5.287670	0.000000

Convergence of Newton's Method

- Newton's method transforms nonlinear equation $f(x) = 0$ into fixed-point problem $x = g(x)$, where

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

  and hence

$$g'(x) = f(x)f''(x)/(f'(x))^2$$

- If x^* is simple root (i.e., $f(x^*) = 0$ and $f'(x^*) \neq 0$), then $g'(x^*) = 0$
- Convergence rate of Newton's method for simple root is therefore *quadratic* $(r = 2)$
- But iterations must start close enough to root to converge

Secant Method

- For each iteration, Newton's method requires evaluation of both function and its derivative, which may be inconvenient or expensive

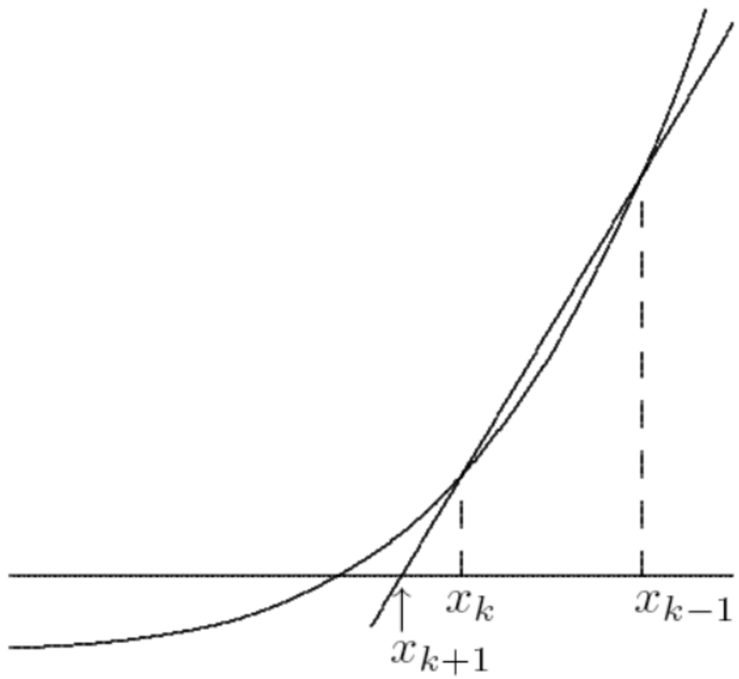
- In *secant method*, derivative is approximated by finite difference using two successive iterates, so iteration becomes

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

- Convergence rate of secant method is normally *superlinear*, with $r \approx 1.618$

Secant Method

Secant method approximates nonlinear function f by secant line through previous two iterates



Safeguarded Methods

- Rapidly convergent methods for solving nonlinear equations may not converge unless started close to solution, but safe methods are slow
- Hybrid methods combine features of both types of methods to achieve both speed and reliability
- Use rapidly convergent method, but maintain bracket around solution
- If next approximate solution given by fast method falls outside bracketing interval, perform one iteration of safe method, such as bisection

Systems of Non-linear Equations

Solving systems of nonlinear equations is much more difficult than scalar case because

- Wider variety of behavior is possible, so determining existence and number of solutions or good starting guess is much more complex
- There is no simple way, in general, to guarantee convergence to desired solution or to bracket solution to produce absolutely safe method
- Computational overhead increases rapidly with dimension of problem

Newton's Method

- In n dimensions, *Newton's method* has form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{J}(\boldsymbol{x}_k)^{-1}\boldsymbol{f}(\boldsymbol{x}_k)$$

  where $\boldsymbol{J}(\boldsymbol{x})$ is Jacobian matrix of $\boldsymbol{f}$,

$$\{\boldsymbol{J}(\boldsymbol{x})\}_{ij} = \frac{\partial f_i(\boldsymbol{x})}{\partial x_j}$$

- In practice, we do not explicitly invert $\boldsymbol{J}(\boldsymbol{x}_k)$, but instead solve linear system

$$\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{s}_k = -\boldsymbol{f}(\boldsymbol{x}_k)$$

  for *Newton step* $\boldsymbol{s}_k$, then take as next iterate

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$$

Newton's Method: Example

- Use Newton's method to solve nonlinear system

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_1 + 2x_2 - 2 \\ x_1^2 + 4x_2^2 - 4 \end{bmatrix} = \mathbf{0}$$

- Jacobian matrix is $\boldsymbol{J}_f(\boldsymbol{x}) = \begin{bmatrix} 1 & 2 \\ 2x_1 & 8x_2 \end{bmatrix}$
- If we take $\boldsymbol{x}_0 = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$, then

$$\boldsymbol{f}(\boldsymbol{x}_0) = \begin{bmatrix} 3 \\ 13 \end{bmatrix}, \quad \boldsymbol{J}_f(\boldsymbol{x}_0) = \begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix}$$

- Solving system $\begin{bmatrix} 1 & 2 \\ 2 & 16 \end{bmatrix} \boldsymbol{s}_0 = \begin{bmatrix} -3 \\ -13 \end{bmatrix}$ gives $\boldsymbol{s}_0 = \begin{bmatrix} -1.83 \\ -0.58 \end{bmatrix}$,

  so $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{s}_0 = \begin{bmatrix} -0.83 & 1.42 \end{bmatrix}^T$

Newton's Method: Example

- Evaluating at new point,

$$\boldsymbol{f}(\boldsymbol{x}_1) = \begin{bmatrix} 0 \\ 4.72 \end{bmatrix}, \quad \boldsymbol{J}_f(\boldsymbol{x}_1) = \begin{bmatrix} 1 & 2 \\ -1.67 & 11.3 \end{bmatrix}$$

- Solving system $\begin{bmatrix} 1 & 2 \\ -1.67 & 11.3 \end{bmatrix} \boldsymbol{s}_1 = \begin{bmatrix} 0 \\ -4.72 \end{bmatrix}$ gives

$\boldsymbol{s}_1 = \begin{bmatrix} 0.64 & -0.32 \end{bmatrix}^T$, so $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{s}_1 = \begin{bmatrix} -0.19 & 1.10 \end{bmatrix}^T$

- Evaluating at new point,

$$\boldsymbol{f}(\boldsymbol{x}_2) = \begin{bmatrix} 0 \\ 0.83 \end{bmatrix}, \quad \boldsymbol{J}_f(\boldsymbol{x}_2) = \begin{bmatrix} 1 & 2 \\ -0.38 & 8.76 \end{bmatrix}$$

- Iterations eventually convergence to solution $x^* = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$

Newton's Method

- Convergence is quadratic
  - Assuming method starts close to the solution
- Computational cost
  - Computing Jacobian matrix costs $O(n^2)$ function evaluations
  - Solving the linear system is $O(n^3)$ operations

Secant Updating Methods

- *Secant updating* methods reduce cost by
  - Using function values at successive iterates to build approximate Jacobian and avoiding explicit evaluation of derivatives
  - Updating factorization of approximate Jacobian rather than refactoring it each iteration
- Most secant updating methods have superlinear but not quadratic convergence rate
- Secant updating methods often cost less overall than Newton's method because of lower cost per iteration

Robust Newton-Like Methods

- Newton's method and its variants may fail to converge when started far from solution

- Safeguards can enlarge region of convergence of Newton-like methods

- Simplest precaution is *damped Newton method*, in which new iterate is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k$$

  where $\boldsymbol{s}_k$ is Newton (or Newton-like) step and α_k is scalar parameter chosen to ensure progress toward solution

- Parameter α_k reduces Newton step when it is too large, but $\alpha_k = 1$ suffices near solution and still yields fast asymptotic convergence rate

Trust-Region Methods

- Another approach is to maintain estimate of *trust region* where Taylor series approximation, upon which Newton's method is based, is sufficiently accurate for resulting computed step to be reliable

- Adjusting size of trust region to constrain step size when necessary usually enables progress toward solution even starting far away, yet still permits rapid converge once near solution

- Unlike damped Newton method, trust region method may modify direction as well as length of Newton step