Module 02:
Finding roots

Starting Monday, January 13

Given a function *f(x)*, computationally find *x** such that *f(x*) = 0*.

- Possible computational problems:
  - *fl(x*)* may not be exact
  - *fl(f(x*))* may not be exactly 0
- Additionally:
  - *x** may be complex

- Realistically, find $\hat{x}$ such that fl(f($\hat{x}$)) ≤ε, for some tolerance ε.
- Assume *f* is continuous.

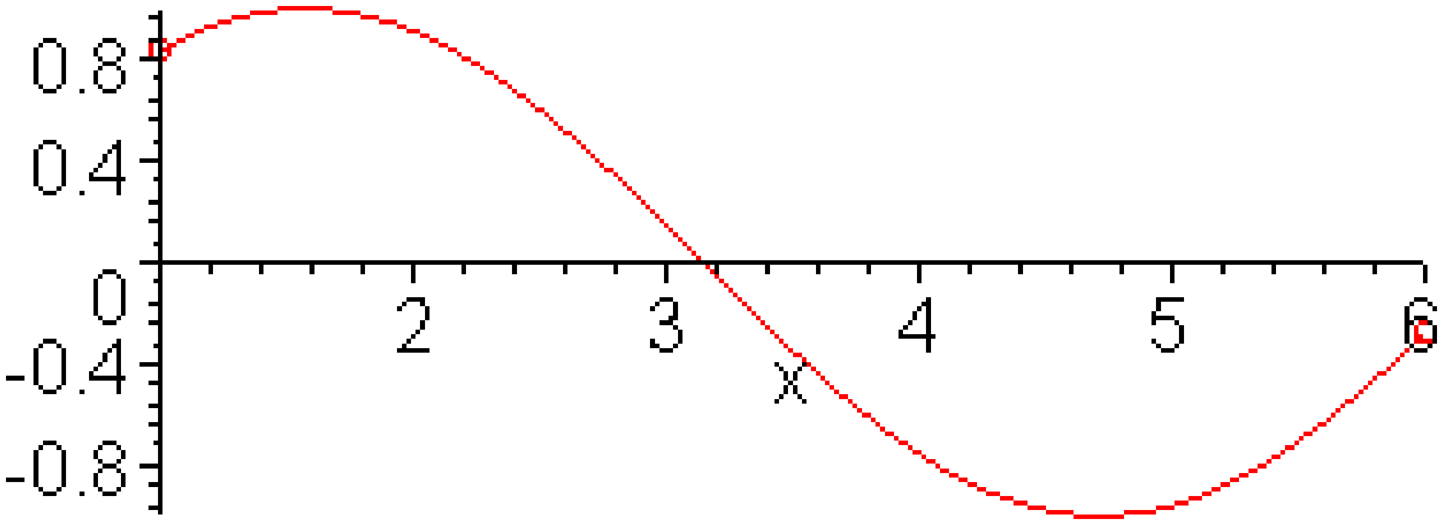
We will examine …

- Computational techniques for finding roots
- Termination conditions (how do we know when to stop?)
- Convergence properties

Technique #1: Bisection

- Background: Assume $a < b$, $f(a) \cdot f(b) < 0$
- By Intermediate Value Theorem: $\exists x^* \in [a, b]$, $s.t.\, f(x^*) = 0$
- Goal: Find smaller and smaller interval containing root:
  - Start with a_0, b_0 where $f(a_0) \cdot f(b_0) < 0$
  - Repeat
    - Determine $c_k = (a_k + b_k)/2$
    - If $f(c_k) = 0$, done
    - If $f(a_k)f(c_k) < 0$: $a_{k+1} = a_k$, $b_{k+1} = c_k$
    - Else: $a_{k+1} = c_k$, $b_{k+1} = c_k$

Graphically: *f(x) = sin(x)* over [1,6]



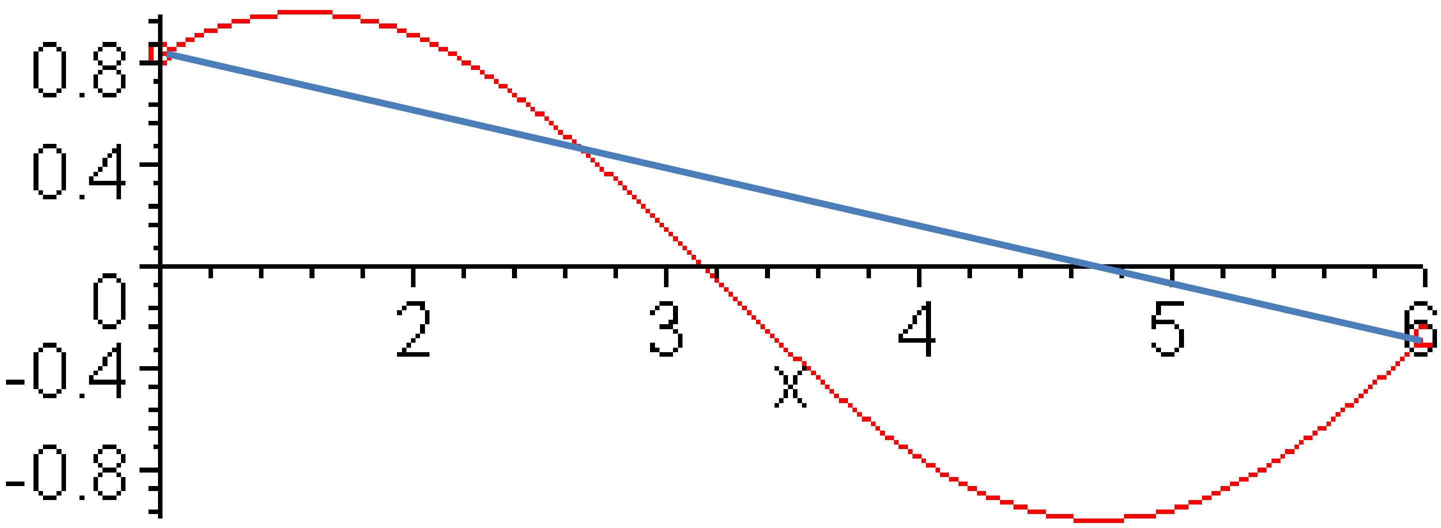
- How to find a and b?
- Suppose $|a_0 - b_0| = 5$. How many steps will it take until $|a_k - b_k| \leq 10^{-5}$?

- In our example, c_0 is pretty close to the actual root π. Can we get reach the actual root faster?

Technique #2: Regula Falsi

- A different approach to bracketing the root.
- Again, start with a_0,b_0 where $f(a_0)\cdot f(b_0) < 0$
- Repeat
  - Determine the line $y = m_k x + d_k$, connecting $(a_k, f(a_k))$ to $(b_k, f(b_k))$.
  - Find c_k satisfying $m_k c_k + d_k = 0$.
  - If $f(c_k) = 0$, done
  - If $f(a_k)f(c_k) < 0$: $a_{k+1} = a_k$, $b_{k+1} = c_k$
  - Else: $a_{k+1} =c_i$, $b_{k+1} =c_k$

Graphically: *f(x) = sin(x)* over [1,6]

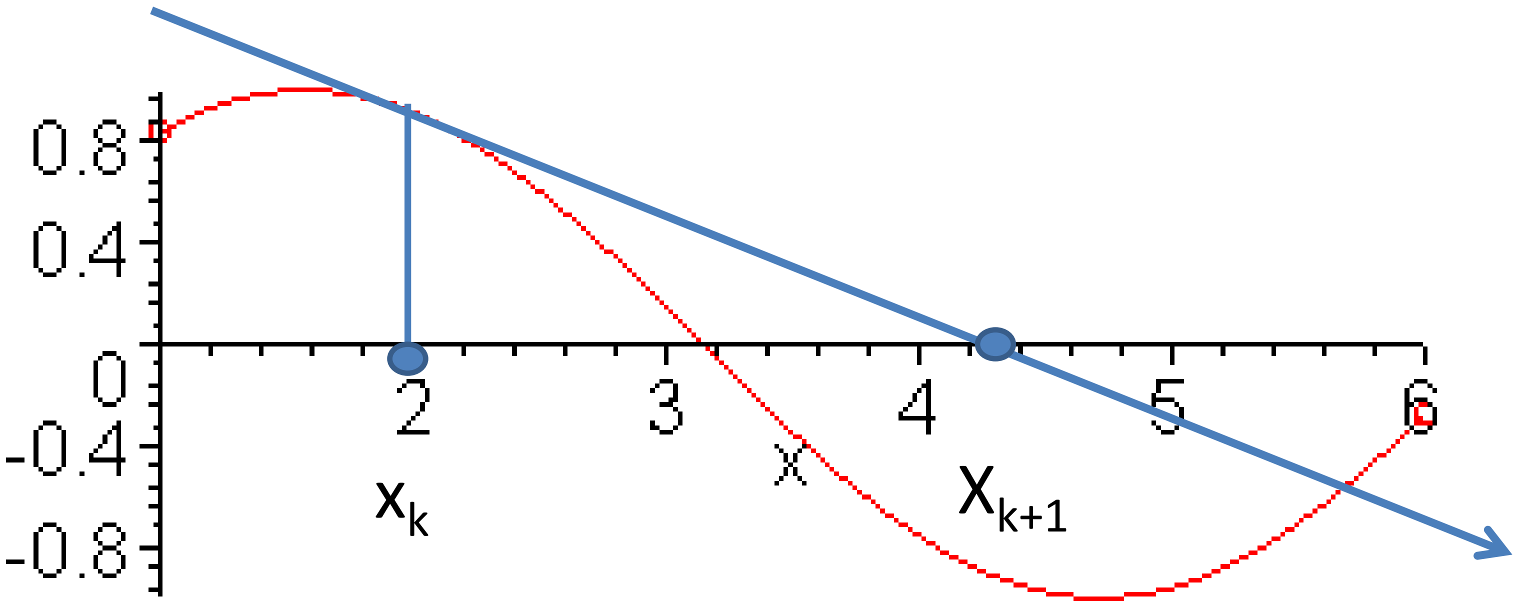


Technique #3: Newton's Method (or Newton-Raphson Method)

- Start with an initial guess x_0
- Repeat
  - Consider the tangent line at x_k: $y = m_k x + b_k$
  - Choose x_{k+1} as the x-intercept of the tangent line

Equivalent derivation: Approximate *f(x)* using the first terms of the Taylor expansion.

Graphically: *f(x) = sin(x)* over [1,6]



About Newton's method

- If "close" to the root, converges pretty quickly *(more later)*
- If not "close" to the root, may not converge
- No *apriori* way to determine if "close"
- Computational issues when $f'(x)$ is close to 0
- Also, requires calculation of derivative (which can be more expensive than function calculations)

Technique #4: Secant Method

- Like Newton, but avoid direct calculation of f'
- Approximate $f'(x_k)$ using x_k and x_{k-1}:

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

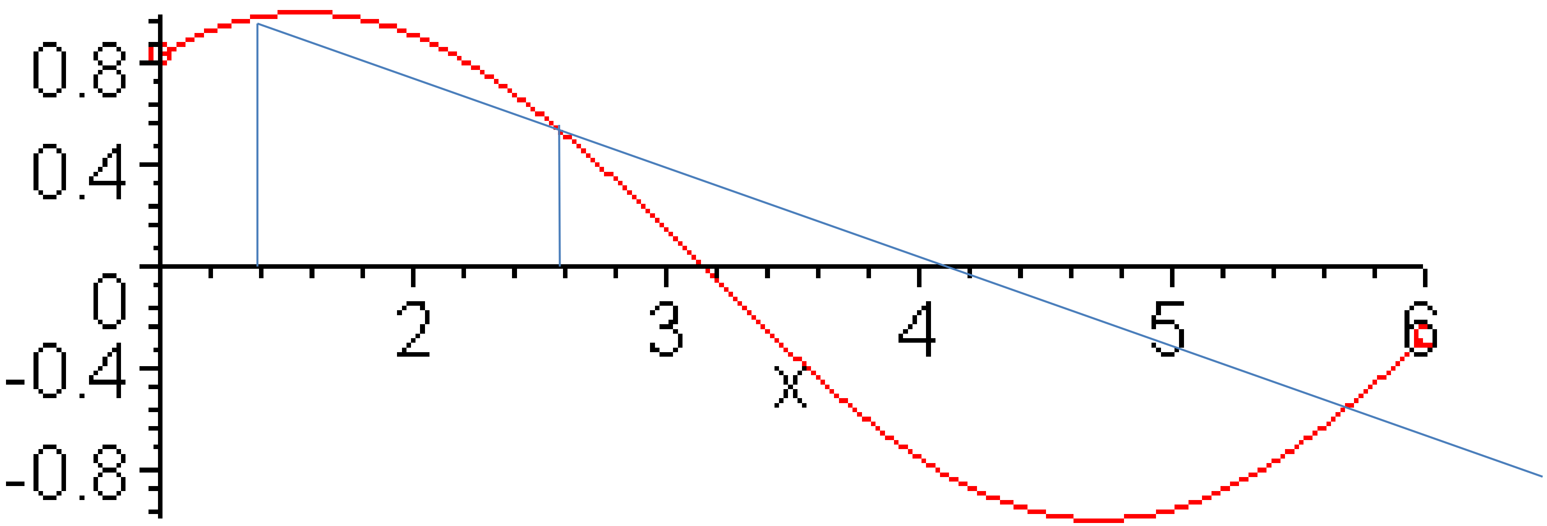
- Now, x_{k+1} will depend on previous two values:

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

- Corresponds to finding root of the secant line connecting x_k and x_{k-1}

(Note: This looks very much like Regula Falsi, but isn't used for bracketing the root.)

Graphically: *f(x) = sin(x)* over [1,6]



About secant method

- Like Newton, concerns about "closeness" of initial guesses
- Computational concerns when two iterates or their function values are very close to each other (cancellation errors)
- When initial guesses are "close", generally converges more quickly than bisection, but not quite as fast as Newton.

Technique #5: Fixed Point Method

- Consider the function differently.
- Define $g(x) = x - f(x)$
- If x^* is a root of f, then it is a fixed point of g, i.e. $x^* = g(x^*)$.
- Choose x_0
- Repeat
  - $x_{k+1} = g(x_k)$

Find root of $f(x) = \sin(x)$

- Choose $g(x) = x - \sin(x)$
- Start with $x_0 = 2$
- $x_1 = x_0 - \sin(x_0) = 1.0907$
- …
- How about $h(x) = x + \sin(x)$
- Start with $x_0 = 2$
- $x_1 = x_0 + \sin(x_0) = 2.9092$
- …
- What is the difference?

About Fixed Point Method

- Any function g for which $x^*=g(x^*)$ if and only if $f(x^*) = 0$ can be used in this approach.
- g must be continuous
- If $|g'(x)| < 1$, and x0 is "close enough", this approach will converge to the fixed point

When to stop?

- Stop when:
  - $| f(x_{k+1})| \leq tol$, for some pre-determined $tol > 0$.
    - Small $f(x_{k+1})$ does not always imply x_{k+1} is close to a root
  - $|x_{k+1}-x_k| \leq tol$, for some pre-determined $tol > 0$
    - Small difference does not mean x_{k+1} is close to root
  - k >= some pre-determined maximum number of step
    - Steps may not have converged to a root
- Neither is ideal. Often a combination is used.

Convergence Properties

- More to come …