CS368 MATLAB Programming

Lecture 13

Young Wu

Based on lecture slides by Michael O'Neill and Beck Hasti

April 28, 2022

Lecture 14

Admin

- Next Wednesday is the last lecture.
- Short lecture: more debug examples, other features of MATLAB.
- Message me on Piazza if you would like me to cover anything specific.

Grades

Admin

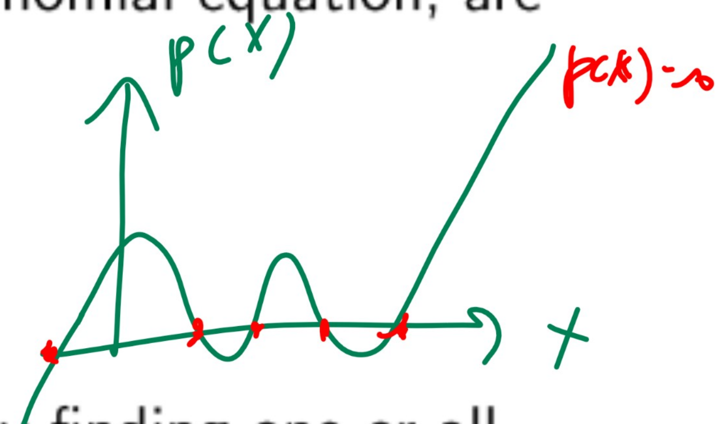
- $P7$ due next week, solution already posted.
- $P1$ to $P7$ and $Q1$ to $Q15$ grades will be updated tonight.
- Late code submissions for $P1$ to $P6$ will be accepted.
- Grades need to be submitted to department by May 16 so May 15 is the absolute latest day to make any submission and message me about your grade.

Algebraic Equations

Math

- An algebraic equation, also called a polynomial equation, are ones in the form,

$$\sum_{i=0}^{n > 4} a_i x^i = 0.$$



- Root finding is the process of numerically finding one or all x's that satisfy the above equation.
- There are in general n solutions or roots (possibly ~~complex~~ or repeated) to the above equation.

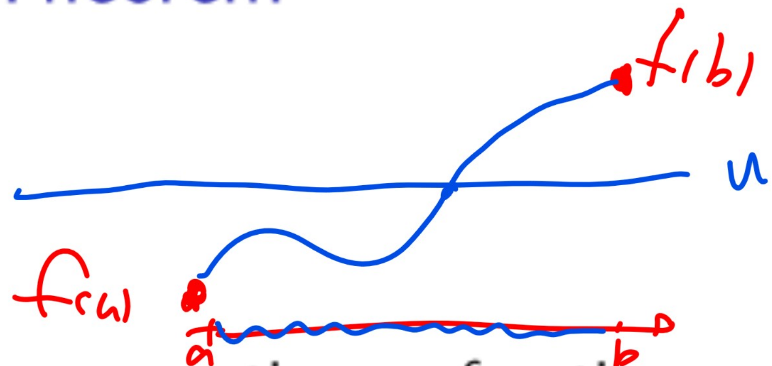
Non-linear Equations

Math

- In general, non-linear equations in the form $f(x) = 0$ are solved using iterative methods.
- Start with a random guess x_0, and compute a sequence $x_1, x_2, \ldots$ with the property that $x^\star = \lim_{n\to\infty} x_i$ satisfies $f(x^\star) = 0$.

Intermediate Value Theorem

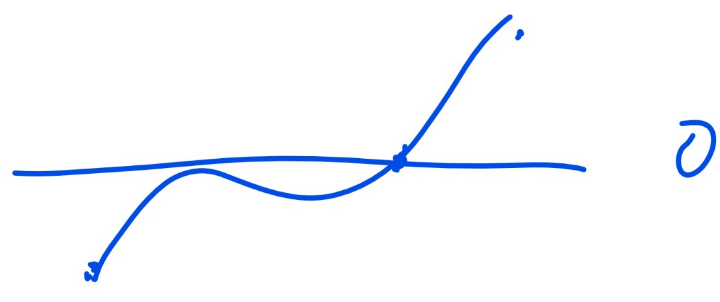
Math



- Intermediate Value Theorem says given a continuous function f, for any u between $f(a)$ and $f(b)$, there exists an $x \in [a, b]$ such that $f(x) = u$.
- IVT implies that if $f(a) \geqslant 0$ and $f(b) \leqslant 0$, then there exists an $x \in [a, b]$ such that $f(x) = 0$.

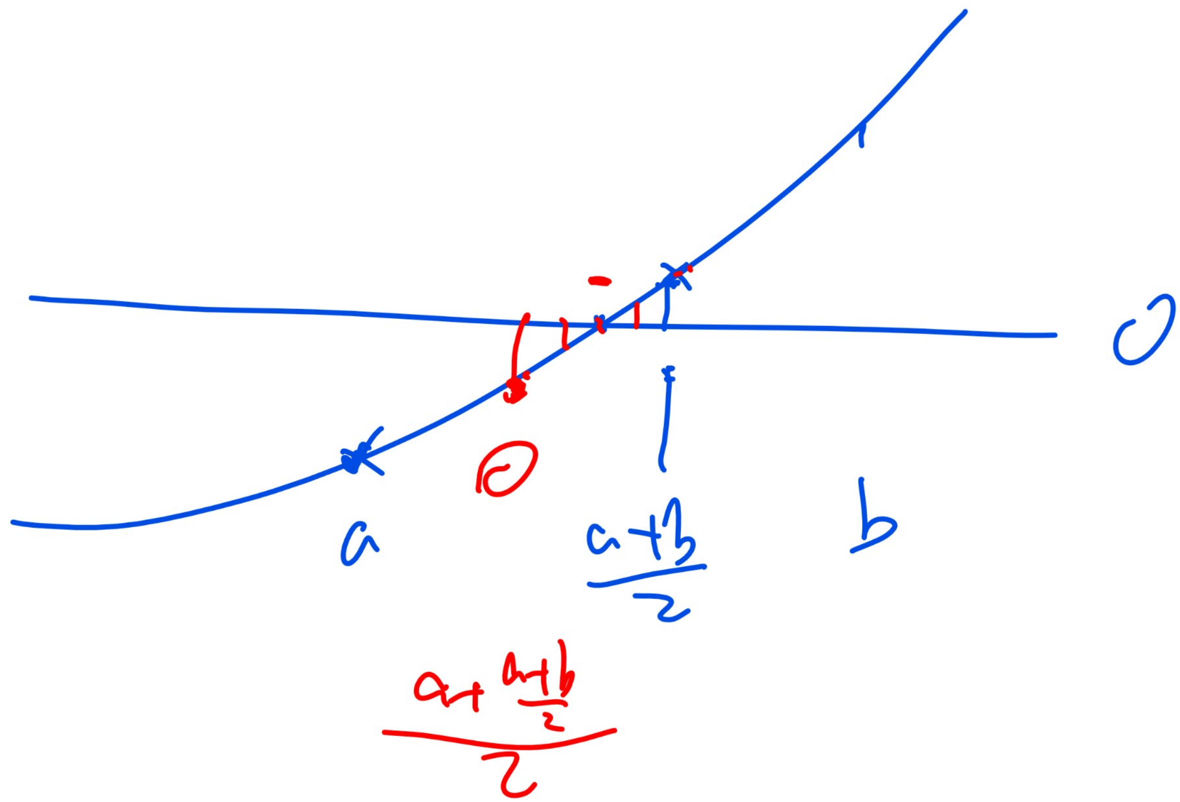
$u = 0$

- Bisection method uses this observation to iteratively reduce the interval [a, b] that contains the root by a half until a and b are close enough.



Intermediate Value Theorem Diagram

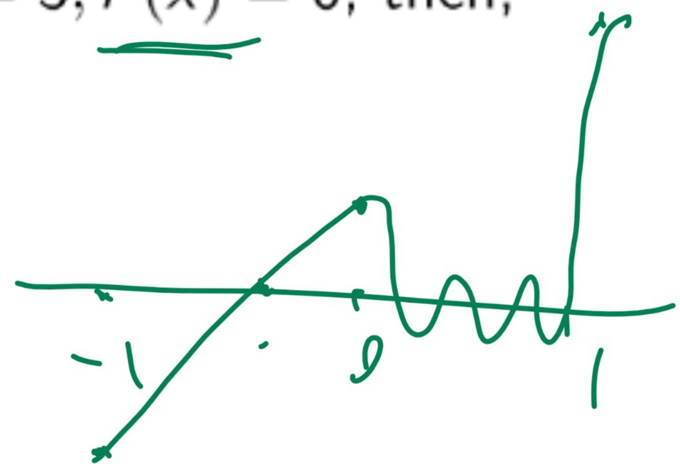
Math



Bisection Method Step 1

Quiz

- $f(-1) = -1, f(0) = 1, f(1) = 3, f(x) = 0$, then,
- A : x ~~must~~ be in [-1, 0]
- B : x ~~must~~ be in [0, 1]
- C : x could be in [-1, 0]
- D : x could be in [0, 1]



Bisection Method Step 2

Quiz

- $f(-1) = -1, f(-0.5) = -0.5, f(0) = 1, f(1) = 3$, there is a unique x such that $f(x) = 0$, then,
- A : x must be in [-1, -0.5]
- B : x must be in [-0.5, 0]
- C : x must be in $[0, 1]$

Bisection Method Step 2

Quiz

- $f(-1) = -1, f(-0.5) = -0.5, f(-0.25) = 0.25, f(0) = 1, f(1) = 3$, unique x such that $f(x) = 0$, then,
- A : x must be in [-1, -0.5]
- B : x must be in [-0.5, -0.25]
- C : x must be in [-0.25, 0]
- D : x must be in [0, 1]

Search

Math

- Bisection method can be used to find a root of $f(x) = 0$ in an interval $x \in [x_0, x_1]$.

1. Start with $[x_0, x_1]$ and $x = \frac{1}{2}(x_0 + x_1)$.
2. If $f(x)$ and $f(x_0)$ has different signs, the solution is between x_0 and x, use bisection method on $[x_0, x]$.
3. If $f(x)$ and $f(x_1)$ has different signs, the solution is between x and x_1, use bisection method on $[x, x_1]$.
4. Stop when $f(x) = 0$ or x_0 and x_1 are close enough.

Bisection Diagram

Math

see demo on W12 page

Search

Code

- Code for bisection search.

recursive

```
function x = bisection(f, x0, x1)
  x = 0.5 * (x0 + x1); % Find midpoint.
  if x1 - x0 < 0.0001 % Solution is close to x.
    return
  elseif f(x0) * f(x) <= 0 % Solution is in [x0, x].
    x = bisection(f, x0, x);
  else % Solution is in [x, x1].
    x = bisection(f, x, x1);
  end
end
```

P7 solution

while

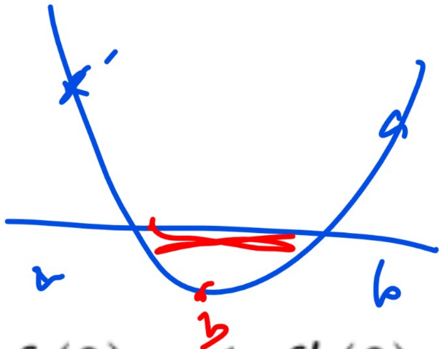
abs

x0

x1

Newton's Method Step

Quiz



- $f(0) = 1$, $f'(0) = -1$, there is a unique x such that $f(x) = 0$, then

local

- ~~A : x must be less than 0~~
- ~~B : x must be more than 0~~
- C : x could be less than 0
- D : x could be more than 0

more likely

Newton's Method

Math

- Newton's method can be used to find a root of $f(x) = 0$, given $f'(x)$, starting from initial guess x_0, preferably close to the solution.

1. Start with the initial guess $x = x_0$.
2. Repeat using Newton's formula $x = \boxed{x - \frac{f(x)}{f'(x)}}$.
3. Stop when $f(x)$ is close enough to 0 (or the number of iterations is too large).

intersection tangent and $y=0$

slope of line

if $f'(x) = 0$

$x = x$

Newton's Method Diagram

Math

Newton's Method

Code

- Code for Newton's Method

```
function x = newton(f, fp, x0)
    if abs(f(x0)) < 0.0001 % Solution is close to x0
        x = x0;
    else % Newton's update
        x = newton(f, fp, x0 - f(x0) / fp(x0));
    end
end
```

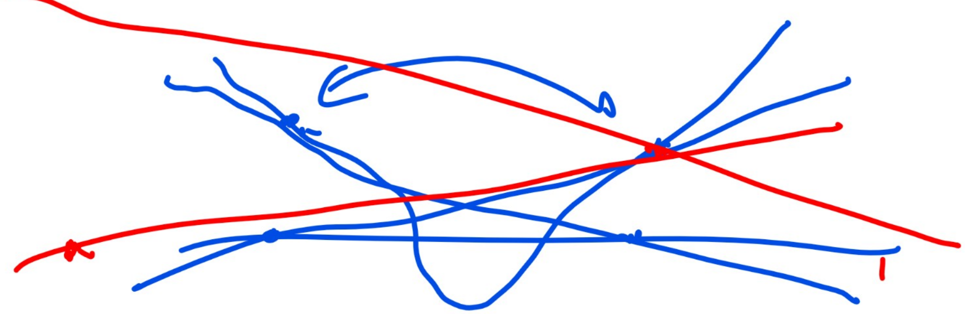
recursive

PT solution

write

Non-Convergence

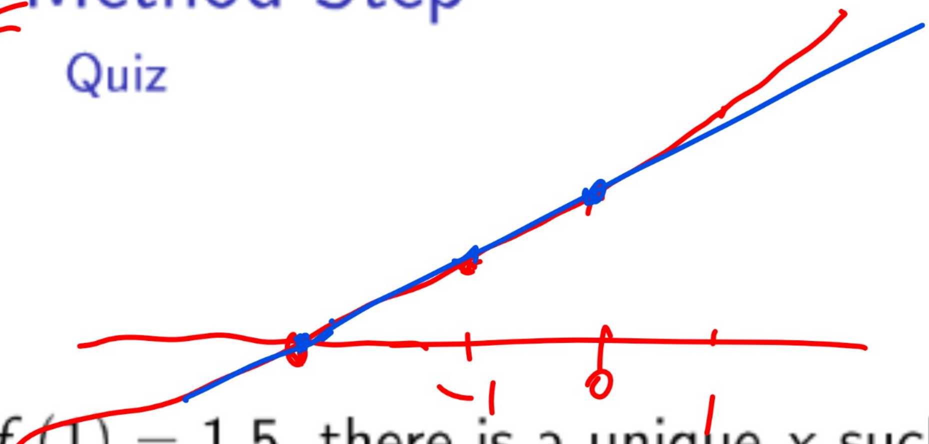
Math



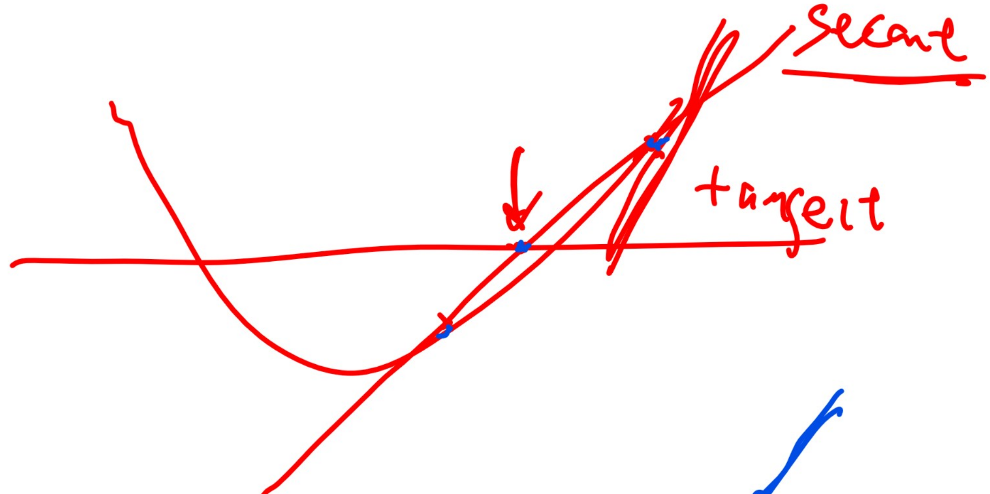
- Newton's method could get stuck when $f'(x) = 0$.
- In that case, start with a different random initial guess.
- Newton's method could also diverge around an unstable root.
- In that case, a variation of Newton's method need to be used.

Secant Method Step

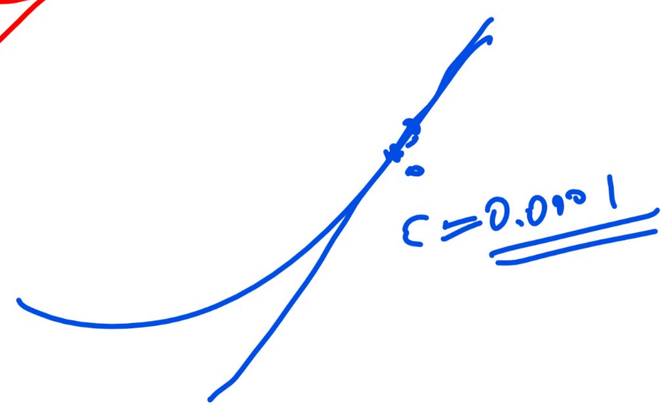
Quiz



- $f(0) = 1, f(-1) = 0.5, f(1) = 1.5$, there is a unique x such that $f(x) = 0$, then
- A : x is likely less than -1
- B : x is likely in $[-1, 0]$
- C : x is likely in $[0, 1]$
- D : x is likely more than 1



$$f' \approx \frac{f(x+\epsilon) - f(x-\epsilon)}{2\epsilon}$$



Secant Method

Math

- Secant method is used instead of Newton's method when the derivative function is unknown or costly to compete.
- Two initial guesses are required, x_0 and x_1, and the Newton's update is replaced by

$$x = \boxed{x - \frac{f(x)}{\boxed{\frac{f(x) - f(x')}{x - x'}}}} = \frac{x' f(x) - x f(x')}{f(x) - f(x')},$$ where x' is the x in the previous iteration.

$\approx f'(x)$

$x' \approx x$

Secant Method Diagram

Math

Secant Method

Math

- Code for Newton's Method

```
function x = secant(f, x1, x0)
    if abs(f(x1)) < 0.0001 % Solution is close to x1
        x = x1;
    else % Secant update
        x2 = (x0 * f(x1) - x1 * f(x0)) / (f(x1) - f(x0))
        x = secant(f, x2, x1);
    end
end
```

fx1   fx0   PT solution

fx0 = f(x1)

only f(x0)

Newton f(x0) f'(x0)

Comparison with Newton's Method

Math

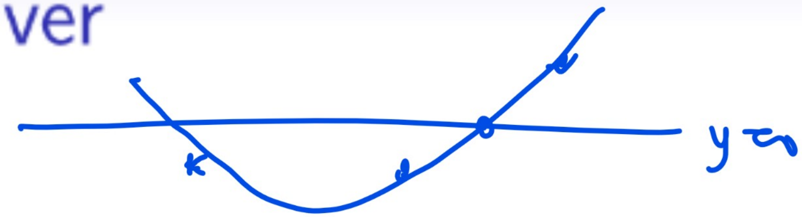
- Secant method is not the same as Netwon's method with the numerical derivative computed using finite differences, but when x and x' are close, a step using Secant method does approximate a step using Newton's method.
- In general, Newton's method usually takes fewer iterations.

f'

- If it is costly to evaluate $f'(x)$, the secant method could be faster than Newton's method.

MATLAB Solver

Code



f $\quad$ @(x)(x^2+1) $\quad$ bisection

- *fzero(f, [x0, x1])* searches for the solution of $f(x) = 0$ between x_0 and x_1, assuming $f(x_0)\, f(x_1) \leqslant 0$.
- *fzero(f, x0)* starts at x_0 and search for the solution of $f(x) = 0$ using a variation of the secant method.

secant

Extension to System of Equations

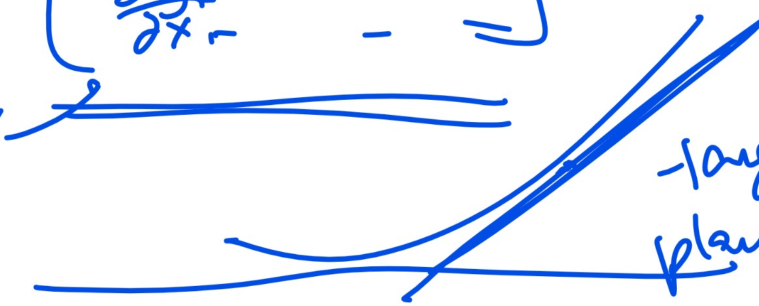
$$F\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Both Newton's method and Secant method can be extended to solving a system of non-linear equations $F(x) = 0$. The Jacobian matrix is used in place of the derivative. The updates are given by $x = x - J_F^{-1}(x)\, F(x)$

$$\frac{\partial f_i}{\partial x_j}$$

$$\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots \\ \frac{\partial y_n}{\partial x_n} & & - \end{bmatrix}$$

$$\rightarrow \left[\frac{\Delta y_i}{\Delta x_i} \right]$$



tangent plane

Blank Slide