**Class 7** – **Polynomial Interpolation (Chp. 18)**

ChE310\_Sec1\_F2019 / 9.17.19

<http://www.reuelgroup.org/numerical-methods-che-310.html>

Announcements:

* Next Tuesday is Career Fair (no class)
* PROJECT Phase 1 Memo: Tuesday 10/1 by Midnight

**Warm Up Group Activity:** submit to SLACK by 2:25pm.

Fit a 4th order polynomial through the following points:

X = [2 4 6 8 10]; Y = [1.1 1.3 1.9 3.2 8]

Solve the following:

1. What is the value of Y predicted at X = 7?
2. What value of X corresponds to 2.3?
3. Can you predict value of Y at X = 11?

**Outline for Class 7 Lecture**

1. Linear regression vs. interpolation (remind)
	1. Piecewise interpolation
2. Matlab built in interpolation
	1. yi = **interp1**(x, y, xi, ‘method’)
	2. Methods:
* ‘nearest’
* ‘linear’ 🡨 Default
* ‘spline’
* ‘pchip’ or ‘cubic’ – piecewise cubic Hermite interpolation – less overshoot and oscillation
	+ Example 18.5 in text
	1. **interp2** and **interp3** are used for 2D and 3D data sets respectively
1. Splines
	1. History (pictures!)
	2. Spline Types







* 1. End conditions



* 1. yy = **spline**(x, y, xx)
	2. Default is cubic spline
		1. Smooth
		2. Higher order introduces oscillation
	3. Default is ‘not a knot’ end conditions
	4. Can force clamped conditions by addition two extra terms in input vector (at first and last position) that denote the start and end slope
	5. Can force ‘natural’ end conditions (or other exotic end requirements) with ‘csape’ command
1. Practice with splines



1. More on plotting

*% Example of a contour plot*

F2 = @(x,y) sin(x).\*tan(y)./x.^2;

xvec = linspace(1,2,200);

yvec = linspace(.5,1,100);

[xmat,ymat] = meshgrid(xvec,yvec); **🡨 USEFUL**

Z = F2(xmat,ymat);

contourf(xvec,yvec,Z)

*% Add a label to the plot*

textstring = '\leftarrow this is a an arrow';

text(1.5,.8,textstring);

% *Save plot*

saveas(gcf,'Output.png')

*% Draw line*

line([1.2 1.2],[0.5 1])

1. If time, review **histogram**