**Class 4** – **Last Day of Matlab Bootcamp (yeah!)**

ChE310\_Sec1\_F2019 9.5.2019

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

**Group Activity:** With your group do the following and submit to Adam on Slack for credit by 2:25pm.

Create a script that uses a **while** loop to determine the number of trials it takes before you throw a Yahtzee (5 dice having the same number, using 6 sided dice).

Now use a **for** loop outside the code you wrote above to determine the average number of trials it takes to throw a Yahtzee if you attempted it 10,000 times.

How can you check your answer to last question using what you know about probability? (analytical solution)

**Outline for Class 4 Lecture**

1. More practice with loops
	1. Finding value of pi – Leibniz formula



 How do we handle this infinite series?

* Set number of iterations
* Error tolerance
1. Plotting 2D data – there are MANY kinds of plots, we will only cover a few in class, read more [here](https://www.mathworks.com/help/matlab/creating_plots/types-of-matlab-plots.html).

**figure** to create new figure window

**close all** at start of code to manage open windows

* Weather data
* Population data
	1. **plot**
		1. Line type (see table last notes)
		2. Marker type (see table last notes)
		3. **xlim([xmin xmax])**
		4. **ylim([ymin ymax])**
		5. **title**
		6. **hold on** / **hold off**
	2. **loglog**
	3. **semilogx**
	4. **semilogy**
	5. **scatter** (I just use **plot** and take off line)
	6. **errorbar**
1. Adjusting and Saving plots

GUI – view🡪property editor

**saveas**(gcf,'FileName.png')

1. Discrete Data
* Skill Survey for 310 entrance
	1. **bar**
	2. **barh**
	3. **bar3(Z)** where Z is the discrete data table
		1. set(gca,'XTickLabel',{'A' 'AA' '1334'}) to change the column labels
		2. set(gca,'YTickLabel',{'A' 'AA' '1334'}) to change the row labels
1. Plotting 3D Data
	1. **plot3** (line plot or points)
		1. Example 🡪 [3D tracking paper](https://pubs.acs.org/doi/abs/10.1021/nn301298e)
	2. Matrix data
* Elevation data
	+ 1. **contour**
		2. **countourf**
		3. **surf**
		4. **surfc** (surface plot with contour)
		5. **mesh**
		6. **meshc** (mesh plot with contour)
1. Working with functions
	1. Separate files
	2. Local function ([link](https://www.mathworks.com/help/matlab/matlab_prog/local-functions-in-scripts.html)) (sub or embedded)
	3. Anonymous functions
		1. fhandle = @(arglist) expression
		2. f1 = @(x,y) x^2 + y^2
	4. Function functions
		1. **fplot**
		2. ode solvers (unit 3)
2. Passing parameters
	1. [Out1 Out2 …] = testfunc(X,Y,Z,…)
	2. Why is this needed? Matlab only looks at current function, disregards all variables from previous function unless they are passed.

🡪**Practice** **Problem** – Compare two methods for binary mixtures. Plot viscosity vs. mass fraction for two oils. Use as an example the following:

 - Crude oil = 3.8 centiStokes (@ 60’C)

- Gas oil = 13.9 centiStokes (@70’C)

M1: Gambill method (1959) proposed the following equation for estimating the kinematic viscosity of a two liquid mixture:



M2: Refutas Equation (slightly more accurate)







BONUS: You can animate the plot!

- Simple, use pause command (tempprofile.m)

- Also can use **videowriter** to save (see help)