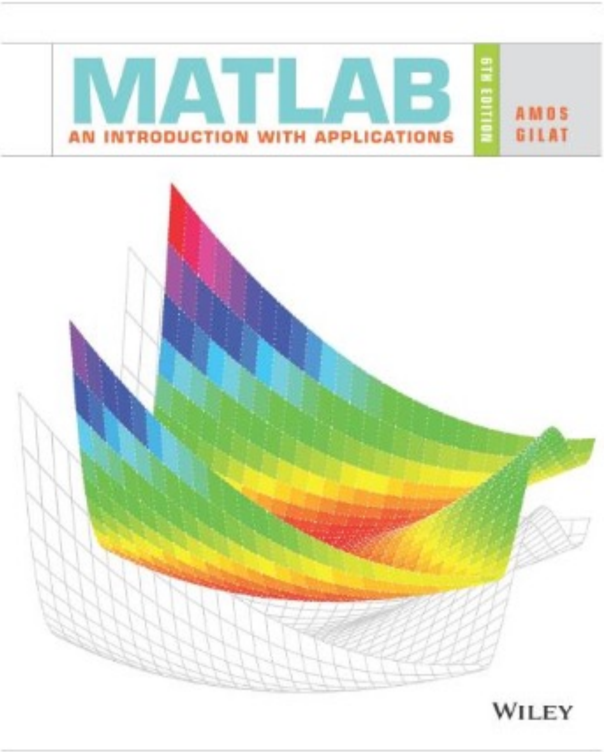
ELE 215
Linear Circuits Laboratory

Recitation 3

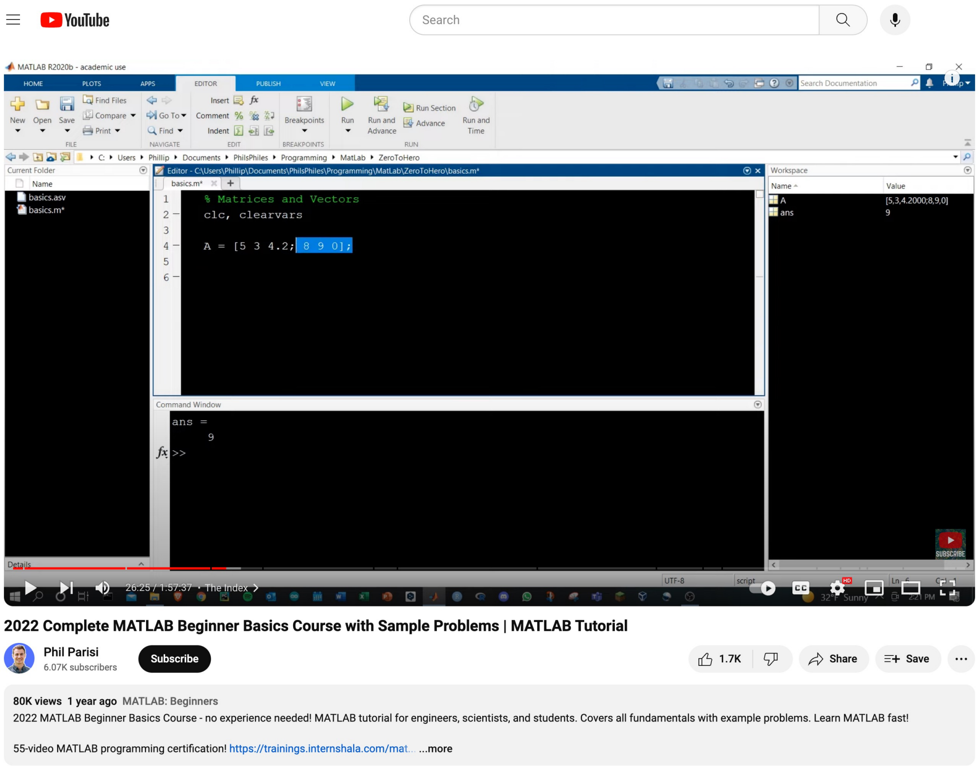
MatLab review from EGR106

Resources

- EGR 106 textbook



- Online; e.g. Phil Parisi's YouTube series



```
% Matrices and Vectors
clc, clearvars

A = [5 3 4.2; 8 9 0];
```

Name	Value
A	[5,3,4.2000;8,9,0]
ans	9

```
ans =
     9
>>
```

2022 Complete MATLAB Beginner Basics Course with Sample Problems | MATLAB Tutorial

Phil Parisi
6.07K subscribers

80K views 1 year ago MATLAB: Beginners

2022 MATLAB Beginner Basics Course - no experience needed! MATLAB tutorial for engineers, scientists, and students. Covers all fundamentals with example problems. Learn MATLAB fast!

55-video MATLAB programming certification! https://trainings.internshala.com/mat... ...more

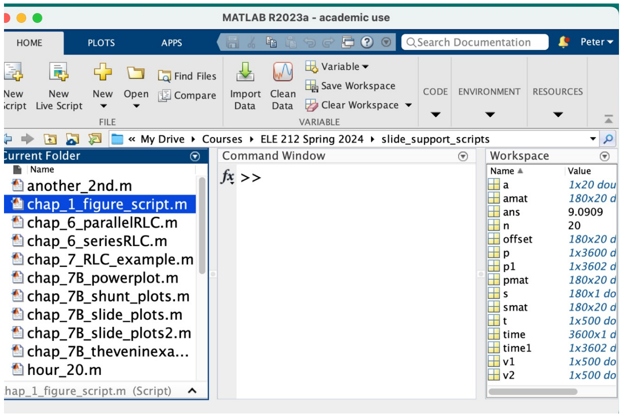
https://www.youtube.com/watch?v=EtUCgn3T9eE&ab_channel=PhilParisi

MATLAB – What is it ?

- Name is from Matrix Laboratory
- Powerful tool for
  - Computation and visualization of engineering and science mathematics
  - Communication of ideas
  - Programming:
    - Built-in editor, debugger, and help
    - Great for quick, one-off problems
    - Many predefined functions (grouped in toolboxes)

The MATLAB Environment

- Multi-paned desktop:
  - Command window
  - Workspace
  - Current folder



Name
another_2nd.m
chap_1_figure_script.m
chap_6_parallelRLC.m
chap_6_seriesRLC.m
chap_7_RLC_example.m
chap_7B_powerplot.m
chap_7B_shunt_plots.m
chap_7B_slide_plots.m
chap_7B_slide_plots2.m
chap_7B_theveninexa...
hour_20.m
hap_1_figure_script.m (Script)

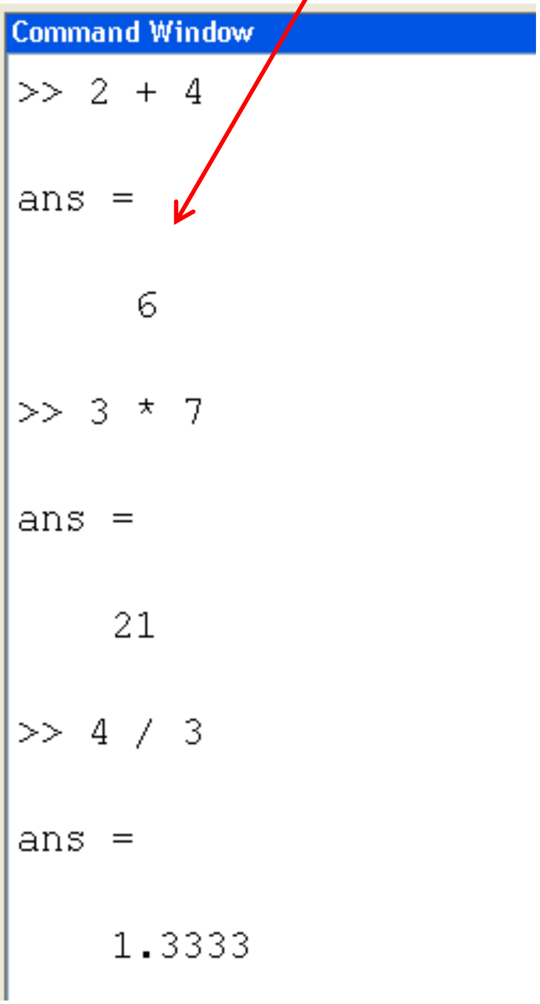
```
>>
```

Name ▲	Value
a	1x20 dou
amat	180x20 d
ans	9.0909
n	20
offset	180x20 d
p	1x3600 d
p1	1x3602 d
pmat	180x20 d
s	180x1 do
smat	180x20 d
t	1x500 do
time	3600x1 d
time1	1x3602 d
v1	1x500 do
v2	1x500 do

  - Other windows: Figure, File Editor, Help, …..

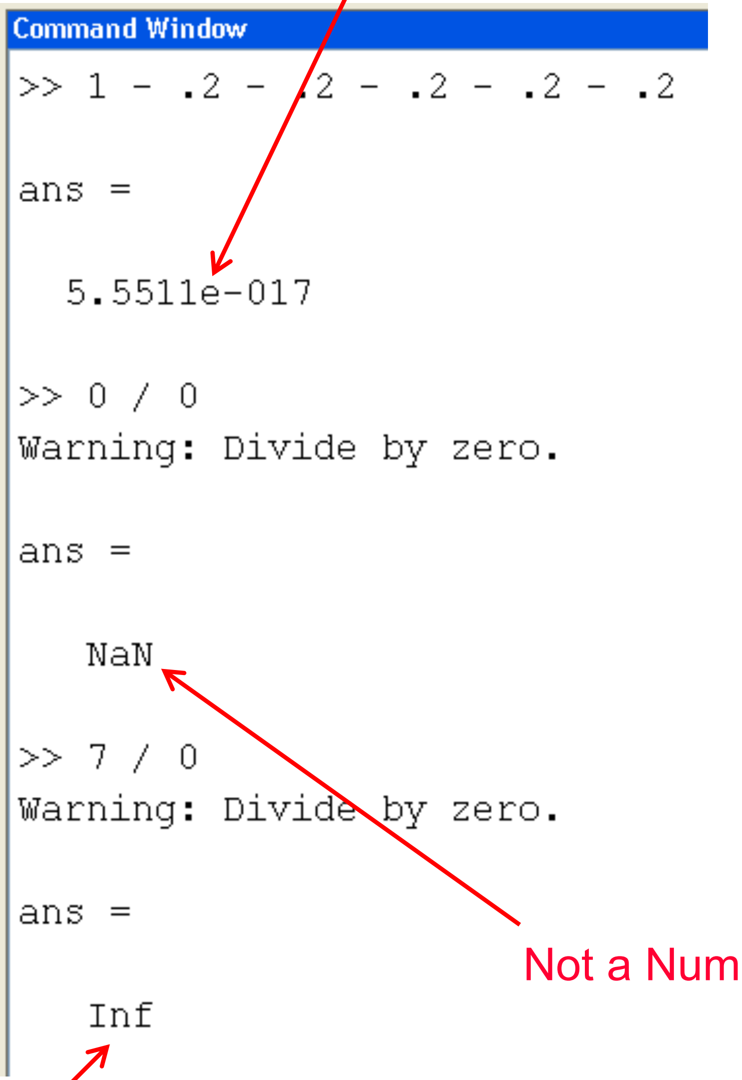
- Command window:
  - Basic math operations are available
  - The “enter” key executes the operation
  - Result is in default variable “ans” with 5 digits of precision shown; perhaps “NaN” or “inf”

Typical result in “ans”



```
>> 2 + 4

ans =

     6

>> 3 * 7

ans =

    21

>> 4 / 3

ans =

    1.3333
```

Not Equal To Zero



```
>> 1 - .2 - .2 - .2 - .2 - .2

ans =

  5.5511e-017

>> 0 / 0
Warning: Divide by zero.

ans =

   NaN

>> 7 / 0
Warning: Divide by zero.

ans =

   Inf
```

Not a Number

Infinity

Features

- Allows for stored variables (using the equal sign as the *assignment* operator)

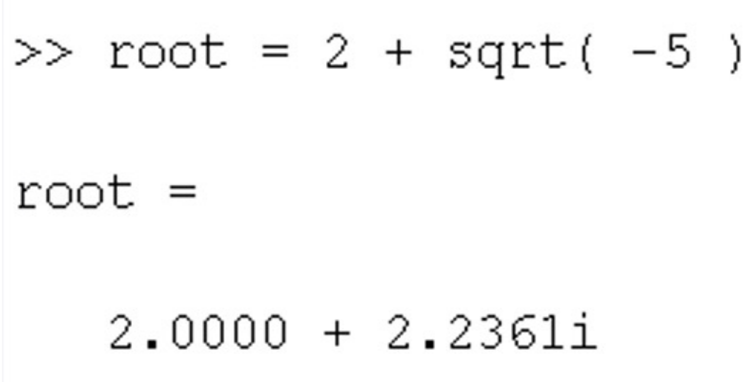
  c = 7.5 bob3 = 3.7789

- Employs PEMDAS operator precedence

  5 – 4 + 3 ^ 4 / (3 – 1) = ? = 41.5

- Complex numbers are native

- Has lots of built in functions



```
>> root = 2 + sqrt( -5 )

root =

   2.0000 + 2.2361i
```

- Data can be “saved” and “loaded”

- Built in help (and lookfor, or just google what you want to do)

- Lots of online help, examples, programs, …

Array Concept

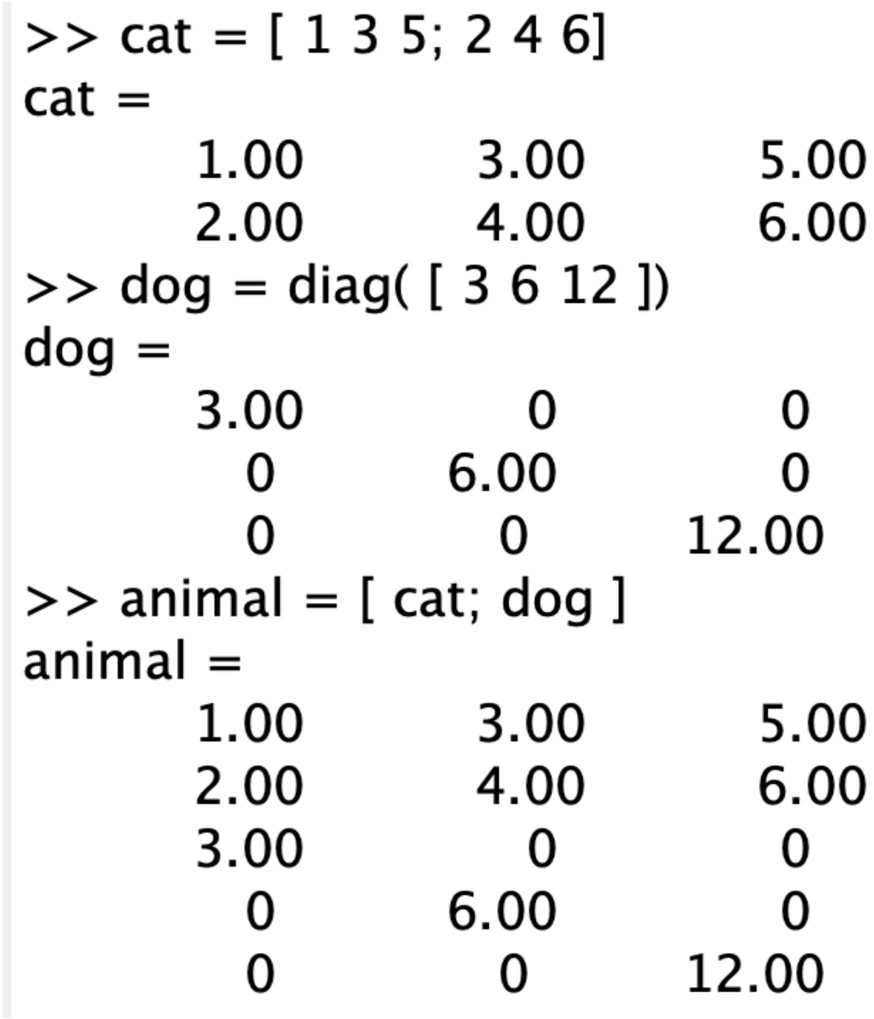
- The fundamental data representation in MatLab

$$\text{yield} = \begin{bmatrix} 4 & 5 & 3 & 9 \\ 10 & 4 & 66 & 20 \\ 18 & -3 & 2 & 0 \end{bmatrix}$$

- Data values organized into rows and columns
  - Size (or dimension) is #rows by #cols, R by C

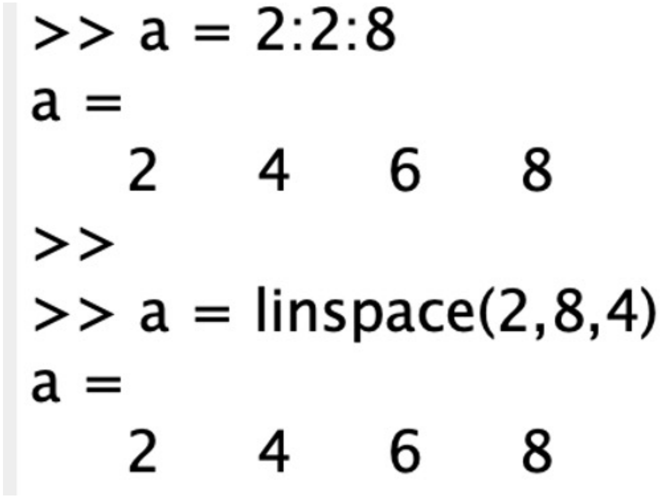
- Terminology
  - Scalar, row vector, column vector, matrix

- Various was to construct (note brackets)
  - Direct enumeration
  - Built-in definition
  - Concatenation



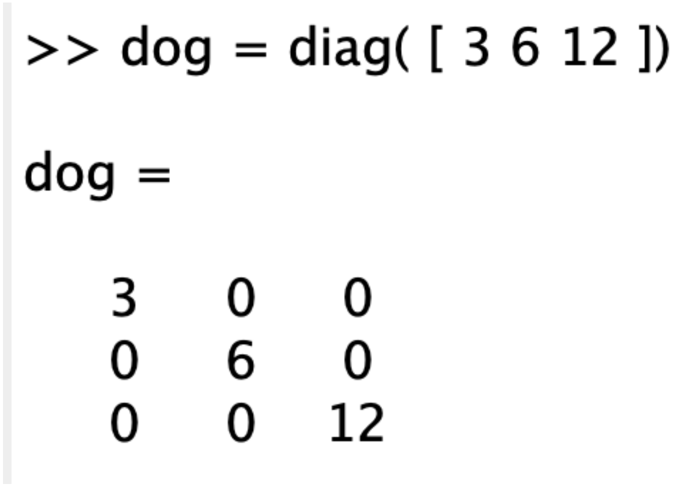
```
>> cat = [ 1 3 5; 2 4 6]
cat =
        1.00          3.00          5.00
        2.00          4.00          6.00
>> dog = diag( [ 3 6 12 ])
dog =
        3.00             0             0
           0          6.00             0
           0             0         12.00
>> animal = [ cat; dog ]
animal =
        1.00          3.00          5.00
        2.00          4.00          6.00
        3.00             0             0
           0          6.00             0
           0             0         12.00
```

- Special constructions for row vectors
  - Colon notation:
  - Linspace command:
  - Logspace command



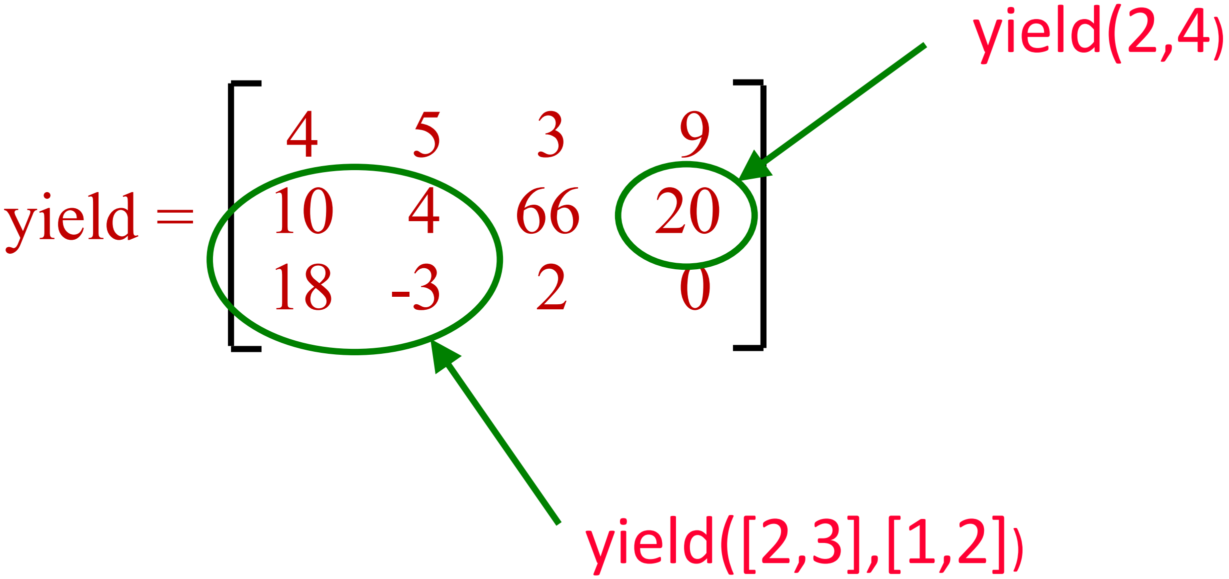
```
>> a = 2:2:8
a =
     2     4     6     8
>>
>> a = linspace(2,8,4)
a =
     2     4     6     8
```

- And special matrices:
  - Zeros, ones, eye, diag



```
>> dog = diag( [ 3 6 12 ])

dog =


     3     0     0
     0     6     0
     0     0    12
```

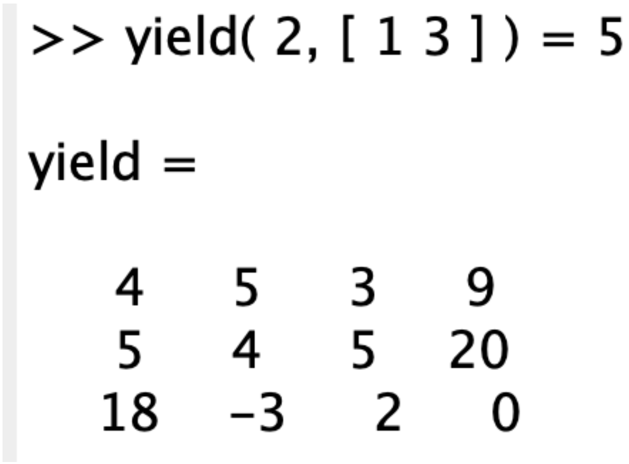
- Addressing, including sub arrays, is accomplished by row/column notation



$$\text{yield} = \begin{bmatrix} 4 & 5 & 3 & 9 \\ 10 & 4 & 66 & 20 \\ 18 & -3 & 2 & 0 \end{bmatrix}$$

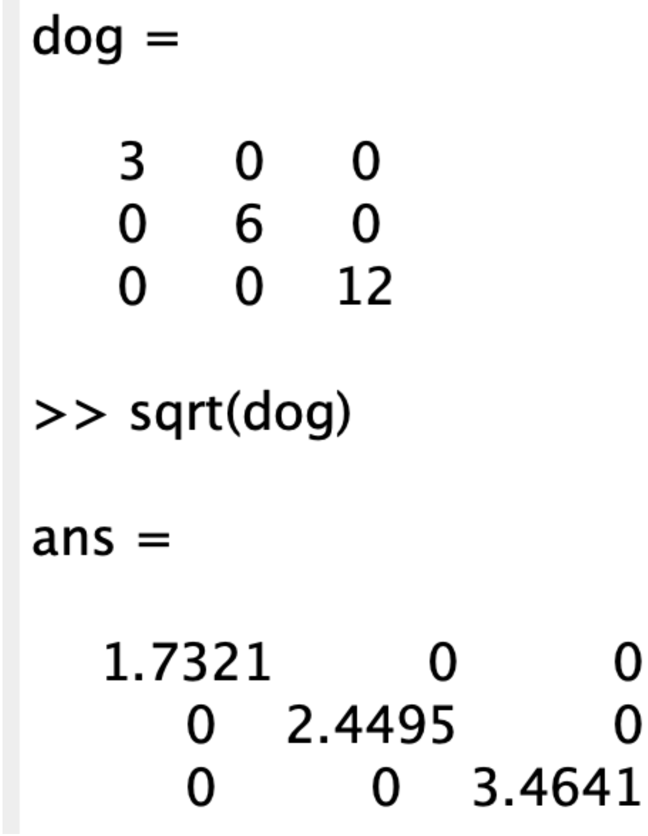
yield(2,4)

yield([2,3],[1,2])



```
>> yield( 2, [ 1 3 ] ) = 5

yield =


     4     5     3     9
     5     4     5    20
    18    -3     2     0
```

- Math works directly on arrays
  - Element-by-element
  - Linear algebra next week



```
dog =

     3     0     0
     0     6     0
     0     0    12

>> sqrt(dog)

ans =

    1.7321         0         0
         0    2.4495         0
         0         0    3.4641
```

- And there are matrix “operators” as well:
  - reshape, size, length, transpose, sort, …
  - determinant, sum, prod, max, min, …

- Many matrix operators are “element-by-element”
  - E.g. square root

- Addition and subtraction of matrices are too
  - Need equal size matrices
  - Scalars “size up”

- How about multiplication? Division? Exponentials? Multiple definitions exist
  - For element-by-element, add a “dot” to the operator

A.*B A./B A.^B

Programs

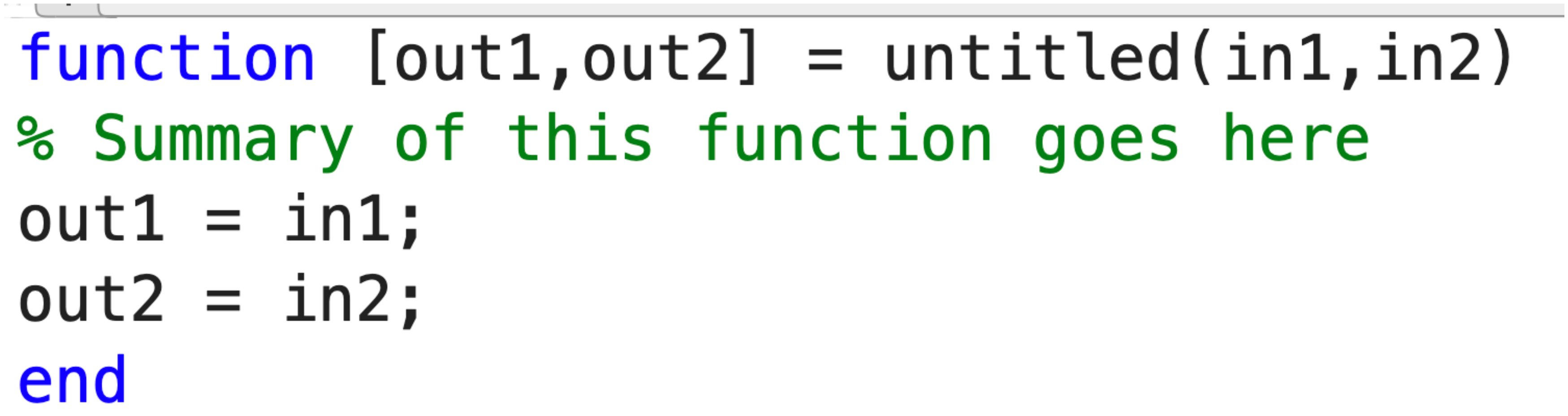
- Script = a file containing MatLab commands
  - Can be re-executed; is easily modified

- Commands are executed one by one, sequentially
  - Run by typing its name (without .m) or by a button in the editor window
  - Results appear in the command window

- Can be created using any text editor
  - .m extension

Programs

- Function = a script file allowed to be run with inputs and outputs
  - Operates in its own workspace
  - Usually does not write any results to screen, just passes output to calling location
  - For example untitled.m

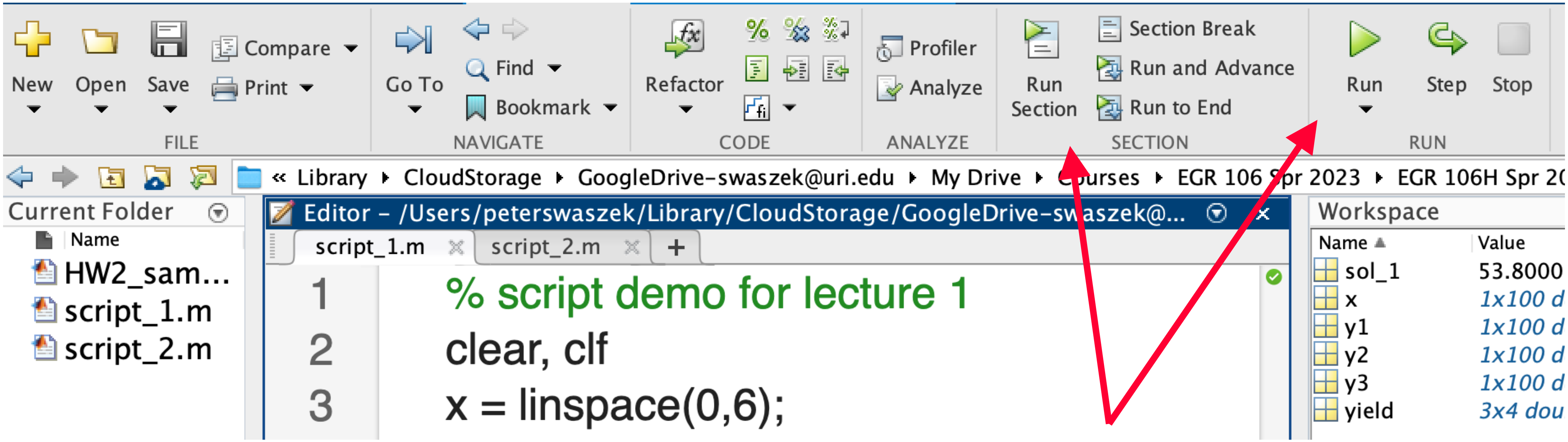


```
function [out1,out2] = untitled(in1,in2)
% Summary of this function goes here
out1 = in1;
out2 = in2;
end
```

MatLab's Editor

- Typical menu choices
- Run and debug capabilities
- Some use of color

- Line numbers
- Auto indent
- Comment lines
- Sections



« Library ▸ CloudStorage ▸ GoogleDrive-swaszek@uri.edu ▸ My Drive ▸ Courses ▸ EGR 106 Spr 2023 ▸ EGR 106H Spr 20

HW2_sam...
script_1.m
script_2.m

Editor – /Users/peterswaszek/Library/CloudStorage/GoogleDrive-swaszek@...

```
% script demo for lecture 1
clear, clf
x = linspace(0,6);
```

Name	Value
sol_1	53.8000
x	1x100 d
y1	1x100 d
y2	1x100 d
y3	1x100 d
yield	3x4 dou

Buttons to "run" it

- Useful script/function Commands:
  - clc – clears the command window
  - clf – clears current figure window
  - clear – clears all variables from memory
  - pause – stops operation and waits for a key press
  - pause(n) – stops operation and waits for n seconds
  - % for comment lines
  - %% for sections
    - Separate button for running just a section

Loops (and Branches)

- MatLab includes relational operators

```
A operator B
```

  - Operator might be greater than, less than, equal to, …
    - More complex operators are included (e.g. `isnan`)
  - Can add logic operators for more complex testing
  - The return is true (1) or false (0)
  - Typically used for branching (if/else/end) and loop control

  - comment - be careful when using = = with real numbers

- Branch instructions allow relational results to decide the following actions
  - If constructions:

```
if x==0
    % what to do
end
```

```
if x==0
    % what to do
else
    % otherwise
end
```

```
if x==0
    % what to do
elseif x > 0
    % then what
else
    % otherwise
end
```

- Other construction types exist:

```
switch x
    case 1
        % if x == 1
    case 2
        % if x == 2
    case 5
        % if x == 5
    case 9
        % if x == 9
    otherwise
        % if none of the above
end
```

```
try
    % some code to run
    % even if it faults
catch
    % what to do on a fault
end
```

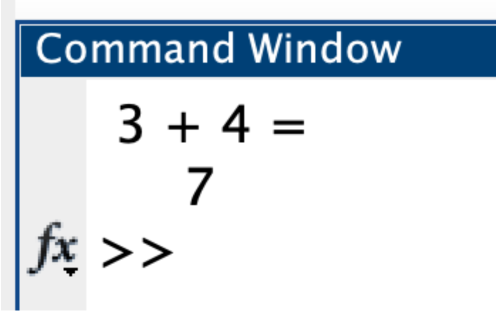
- MatLab also allows repetition of code using loops
  - `for/end` is the most common; fixed number of repetitions
  - `while/end` is sometimes useful when the number of repetitions is unknown
  - Both can be nested
  - Controls:
    - `break` allows for early termination
    - `continue` allows for continuation

- Example.

```
clear
clc
data = rand(5,10) - .5;
% start of loop
for r = 1:size(data,1)
    for c = 1:size(data,2)
        x = data(r,c);
        % iteration until fail or done
        while x < 1
            x = atan2(1,x);
            if isnan(x)
                break
            end
            data(r,c) = x;
        end
    end
end
```

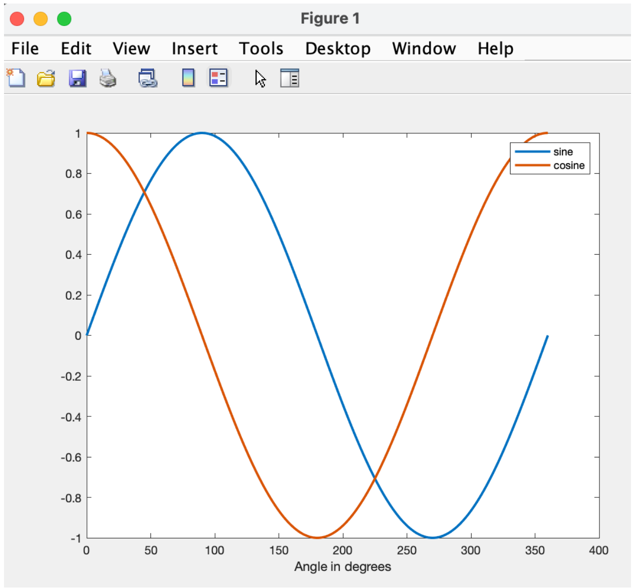
Nice Script Structure

```
%% well documented sample
clf, clear, clc
% section 1 - just show addition
disp(' 3 + 4 = ')
disp(3+4)
```



```
 3 + 4 =
      7
fx >>
```

```
%% use of section division
angle = 0:360; % angle in degrees
s = sind(angle);
c = cosd(angle);
figure(1)
plot(angle,s,angle,c,'linewidth',2)
xlabel('Angle in degrees')
legend('sine','cosine')
```



Specifics for next week

- Lab 2 if you are in the second grouping

- Coming up – Exercise 1 – 75 points (individual)
  - A MatLab programming exercise
  - Instructions posted on the ELE 215 site
  - “shell” of the function is provided
  - Submitted by 9 AM April 6 via Brightspace
    - No late submissions