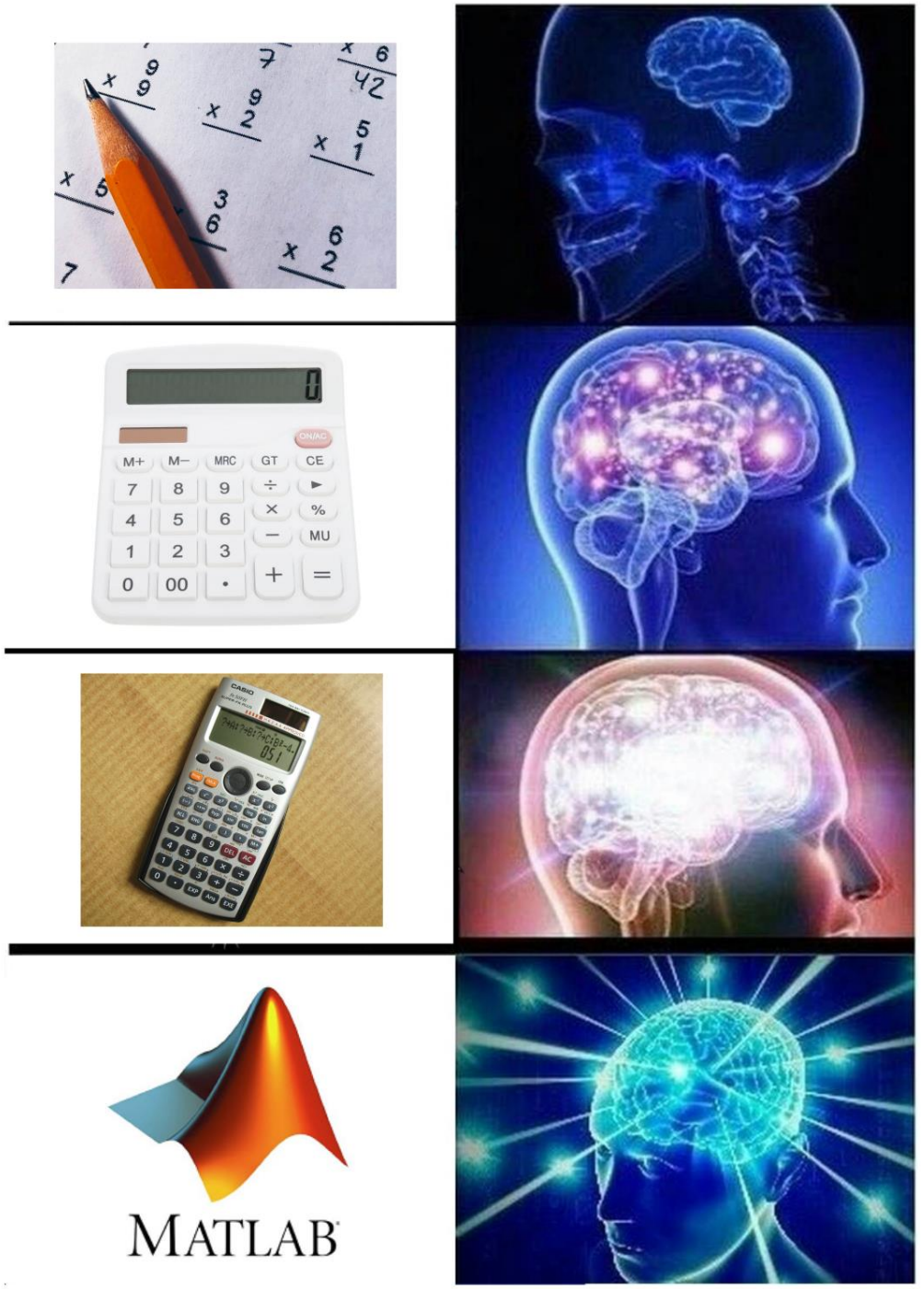
MATH2221
Mathematics Laboratory II

Lecture 2: Basic MATLAB Functions and MATLAB Scripts

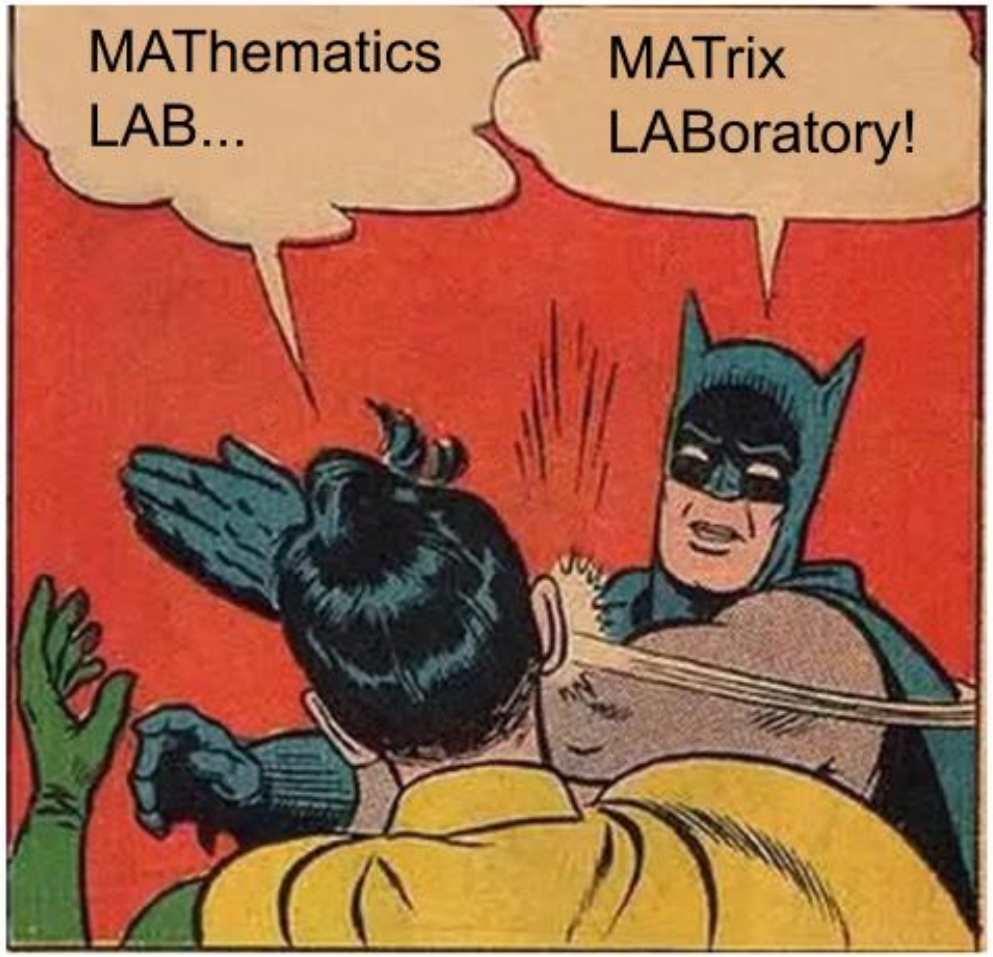
Gary Choi

January 14, 2025



Recall: What is MATLAB?

- **Calculator x Programming language x Interactive visualization and control tool**
  - Facilitate matrix and vector computations
  - Easy-to-use visualization tools
  - Easier to program than C/C++/Python/…



Recall: Basic operations in MATLAB

d = [2 3 4]	Define a row vector (note: need square bracket [ ])
C = [3 1 1; 0 2 4; -1 0 1]	Define a 3-by-3 matrix (note: use semicolon ; to start a new row)
e = d’	e is the transpose of d
3*C	Scalar multiplication
C*e	Matrix-vector product
C*C	Matrix-matrix product
C^2	Square of a matrix

Some basic commands in MATLAB

Description	MATLAB Command	Example
MATLAB documentation	demo	demo
Displays the help text for the functionality	help	help primes
Clear a variable	clear	a = 1; clear a
Clear all variables	clear all	clear all
Clear command window	clc	clc

Some basic functions and special values in MATLAB

Description	MATLAB Command	Example
Square root	sqrt	sqrt(9)
Exponential	exp	exp(2)
Natural logarithm	log	log(24)
Common (base 10) logarithm	log10	log10(100)
Absolute value	abs	abs(-1.5)
π	pi	pi^2
Infinity ∞	Inf	1/Inf

Rounding and number theoretic functions in MATLAB

Description	MATLAB Command	Example
Round	round	round(6.1)
Floor function	floor	floor(4.9)
Ceiling function	ceil	ceil(3.2)
Remainder after division	mod	mod(7,3)
Factorial n!	factorial	factorial(6)
Binomial coefficient	nchoosek	nchoosek(4,2)
Prime factorization	factor	factor(12)

Trigonometric and hyperbolic functions in MATLAB

Description	MATLAB Command	Example
Sine	sin	sin(pi/6)
Cosine	cos	cos(pi/4)
Tangent	tan	tan(pi/3)
Inverse sine (arcsin)	asin	asin(sqrt(3)/2)
Inverse cosine (arccos)	acos	acos(1/2)
Inverse tangent (arctan)	atan	atan(1)
Hyperbolic sine	sinh	sinh(pi/6)
Hyperbolic cosine	cosh	cosh(1)
Hyperbolic tangent	tanh	tanh(pi/4)
Inverse hyperbolic sine	asinh	asinh(0.5)
Inverse hyperbolic cosine	acosh	acosh(0.5)
Inverse hyperbolic tangent	atanh	atanh(0.5)

Functions related to complex numbers in MATLAB

Description	MATLAB Command	Example
$\sqrt{-1}$	i or j or 1i or 1j	2 + 1i
Absolute value (modulus)	abs	abs(2+3*1i)
Phase angle (in radians)	angle	angle(1+1i)
Complex conjugate	conj	conj(4+5*1i)
Real part	real	real(2+3*1i)
Imaginary part	imag	imag(4-5*1i)

Vector commands in MATLAB

Description	MATLAB Command	Example
Create a row vector with entries from m to n with an increment of 1	m:n	1:3 = [1,2,3] (3:6)' = [3;4;5;6]
Create a row vector with entries from m to n with an increment of s	m:s:n	1:2:10 = [1,3,5,7,9] 20:-5:1 = [20, 15, 10, 5]
Obtain the length of the vector	length(v)	length([2,3,5]) = 3 length(0:pi:100) = 32

Vector commands in MATLAB

```
>>  1:5
ans =
    1    2    3    4    5
```

Use colon “m:n” (where n > m) to create a row vector with entries from m to n with an increment of 1

```
>> 3:0.2:4
ans =
   3.0000   3.2000   3.4000   3.6000   3.8000   4.0000
```

Use colon “m:s:n” to create a row vector with entries from m to n with an increment of s

```
>> a = 5:-1:-3
a =
    5    4    3    2    1    0   -1   -2   -3
```

Negative increment can also be used

```
>> 0:pi:10
ans =
         0    3.1416    6.2832    9.4248
```

Values outside the interval will not be included

Matrix commands in MATLAB

Description	MATLAB Command	Example
m-by-m zero matrix	zeros(m)	zeros(2) = 2-by-2 zero matrix
m-by-n zero matrix	zeros(m,n)	zeros(4,3) = 4-by-3 zero matrix
m-by-m matrix of ones	ones(m)	ones(2) = $\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$
m-by-n matrix of ones	ones(m,n)	ones(1,3) = $\begin{bmatrix}1 & 1 & 1\end{bmatrix}$
m-by-m identity matrix	eye(m)	eye(2) = $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
m-by-n matrix with 1's on the diagonal and 0's elsewhere	eye(m,n)	eye(2,3) = $\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\end{bmatrix}$
m-by-m random matrix	rand(m)	rand(2) = 2-by-2 random matrix
m-by-n random matrix	rand(m,n)	rand(4,3) = 4-by-3 random matrix
Obtain the size of the matrix	size(A)	size([1,2; 3 4; 5 6])

Matrix commands in MATLAB

```
>>  M = zeros(2,4)
M =
    0    0    0    0
    0    0    0    0
```

Creating a zero matrix with 2 rows and 4 columns

```
>> A = ones(3,2)
A =
    1    1
    1    1
    1    1
```

Forming a matrix with 3 rows and 2 columns, with all values = 1

```
>> B = ones(2)
B =
    1    1
    1    1
```

Will assume the matrix is a square matrix if only one input is given

Matrix indexing in MATLAB

```
>> A = [1, 2, 3, 4; 5, 6, 7, 8];
```

Define a 2-by-4 matrix A

```
>> A(2,3)
ans =
    7
```

Extract the entry $A_{2,3}$

```
>> A(2,:)
ans =
    5    6    7    8
```

Use colon : to extract the entire row or column

```
>> A(:,3)
ans =
    3
    7
```

Use colon : to extract the entire row or column

```
>> A(1,end)
ans =
    4
```

The end operator is an easy shorthand way to refer to the last element

Matrix indexing in MATLAB

```
>> A = [1, 2, 3, 4; 5, 6, 7, 8];
```

Define a 2-by-4 matrix A

```
>> A(2,2:4)
ans =
    6    7    8
```

Extract row 2, column 2-4

```
>> A([2,1],:)
ans =
    5    6    7    8
    1    2    3    4
```

Swapping row 1 and row 2

```
>>  A([2 1],[1 4 3 2 2])
ans =
    5    8    7    6    6
    1    4    3    2    2
```

Create a matrix with some permutations and repetitions via more complicated indexing

For more examples, see:
https://www.mathworks.com/company/technical-articles/matrix-indexing-in-matlab.html

Entrywise operations for vectors and matrices

- Adding/subtracting a scalar (e.g. [1,2,3] + 1): entrywise
- Scalar multiplication (e.g. 3*[1,2,3]): entrywise
- In other cases, remember to add "." for entrywise operations:
  - Entrywise multiplication A.*B
  - Entrywise division A./B
  - Entrywise exponentiation A.^B
- Applying scalar functions to vectors and matrices: entrywise
  e.g. sin(C), exp(C), sqrt(C), ceil(C)

Entrywise operations for vectors and matrices

```
>> A = [1,2; 3 4];
>> B = [2,0; 2 5];
>> A.*B
ans =
    2    0
    6   20
```

Entrywise multiplication (i.e. having $a_{ij} * b_{ij}$ for all i, j)

```
>> A*B
ans =
    6   10
   14   20
```

Usual matrix multiplication

```
>> A.^2
ans =
    1    4
    9   16
```

Entrywise power (i.e. having a_{ij}^2 for all i, j)

Other common vector and matrix functions

Description	MATLAB Command
Norm of vector v	norm(v)
Dot product	dot(u,v)
Cross product	cross(u,v)
Sum of entries of v	sum(v)
Matrix inverse of A	inv(A)
Determinant of A	det(A)
Eigenvalues of A	eig(A)
Norm of A	norm(A)
Diagonal part of A	diag(A)
Upper triangular part of A	triu(A)
Lower triangular part of A	tril(A)
Sum of entries of A along columns or rows	sum(A,1) (columns) , sum(A,2) (rows)

Other common vector and matrix functions

```
>> u = [4 -1 2];
>> v = [2 -2 -1];

>> dot(u,v)
ans =
    8

>> cross(u,v)
ans =
    5    8   -6

>> norm(u)
ans =
    4.5826
```

Dot product

Cross product

Norm

Other common vector and matrix functions

```
>> A = [1 3 5; 10 -2 3; 2 3  4]
A =
     1     3     5
    10    -2     3
     2     3     4
```

```
>> det(A)
ans =
    51
```

Determinant

```
>> inv(A)
ans =
   -0.3333    0.0588    0.3725
   -0.6667   -0.1176    0.9216
    0.6667    0.0588   -0.6275
```

Matrix inverse

Other common vector and matrix functions

```
>> A = [1 3 5; 10 -2 3; 2 3  4]
A =
     1     3     5
    10    -2     3
     2     3     4
```

```
>> diag(A)
ans =
     1
    -2
     4
```

Diagonal entries

```
>> eig(A)
ans =
    9.4162
   -5.4162
   -1.0000
```

Eigenvalues

Other common vector and matrix functions

```
>> A = [1 3 5; 10 -2 3; 2 3  4]
A =
    1    3    5
   10   -2    3
    2    3    4
```

```
>> triu(A)
ans =
    1    3    5
    0   -2    3
    0    0    4
```

Upper triangular part

```
>> tril(A)
ans =
    1    0    0
   10   -2    0
    2    3    4
```

Lower triangular part

Solving matrix equation: x = A \ b

- In MATLAB, we can solve a matrix equation Ax = b by simply typing x = A \ b
  - Note: (\) is the backslash operator, should not be mixed up with division (/)

Example:

```
>> A = [1,2; 3 4];
>> b = [9; 10];
>> A\b
ans =
   -8.0000
    8.5000
```

Example:

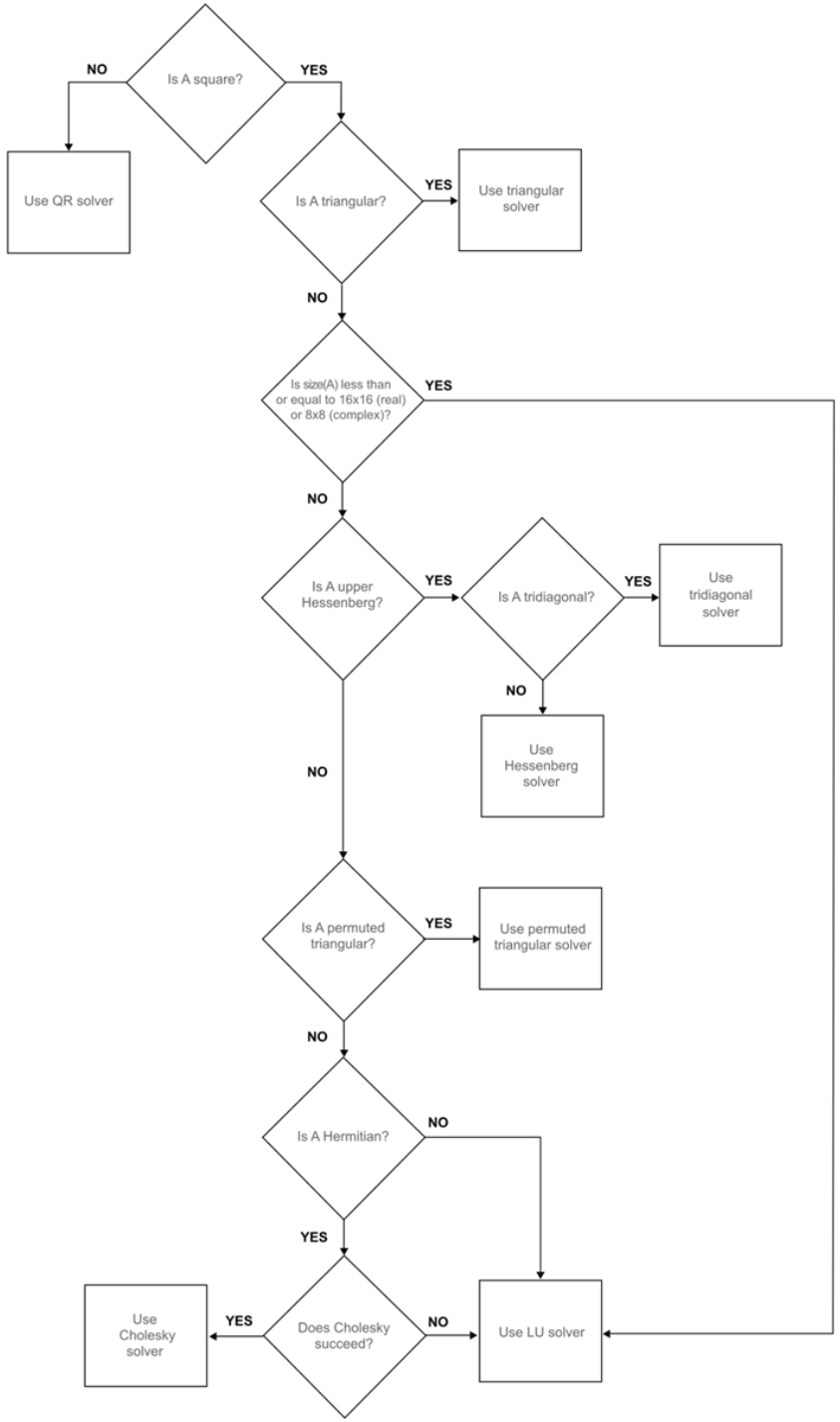
```
>> A = eye(10)+ones(10);
>> b = (1:10)';
>> x = A\b
```

```
x =
   -4.0000
   -3.0000
   -2.0000
   -1.0000
   -0.0000
    1.0000
    2.0000
    3.0000
    4.0000
    5.0000
```

How does x = A \ b work?

- MATLAB automatically checks the properties of the matrix A and finds a suitable method to solve the equation (see the flow chart) https://www.mathworks.com/help/matlab/ref/double.mldivide.html

- Remark:
  - In most cases, x = A\b does not require explicitly finding A^{-1} and hence is much faster than inv(A)*b or A^(-1)*b
  - Avoid the computation of matrix inverse unless it is necessary

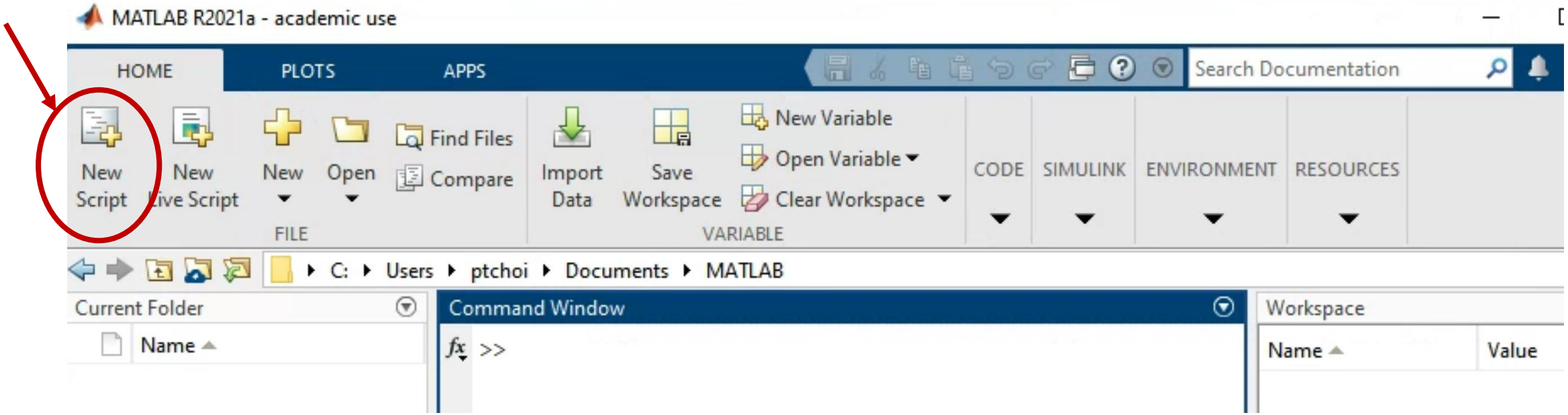


MATLAB scripts

- What is a MATLAB script?
  - A file with a .m extension
  - Can execute a series of MATLAB statements
- What is a MATLAB script good for?
  - Save, edit and debug MATLAB programming code

Creating a MATLAB script

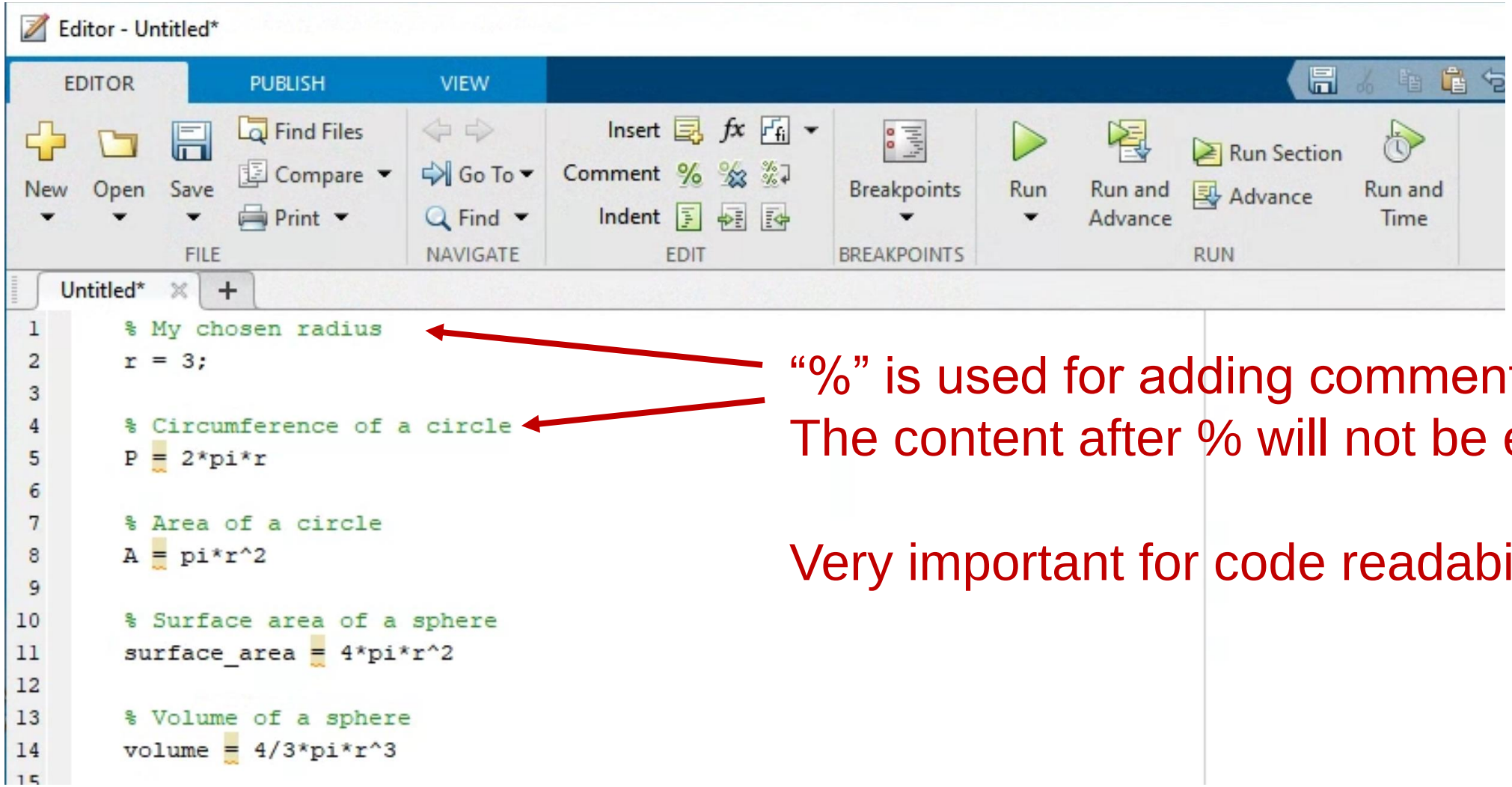
- A new script can be created by clicking the new script button in the Home toolbar.



- It can also be created by pressing ctrl + N

Creating a MATLAB script

- Once a script is created, MATLAB statements can be inserted line-by-line. When the script is run, each line of code will be executed sequentially.

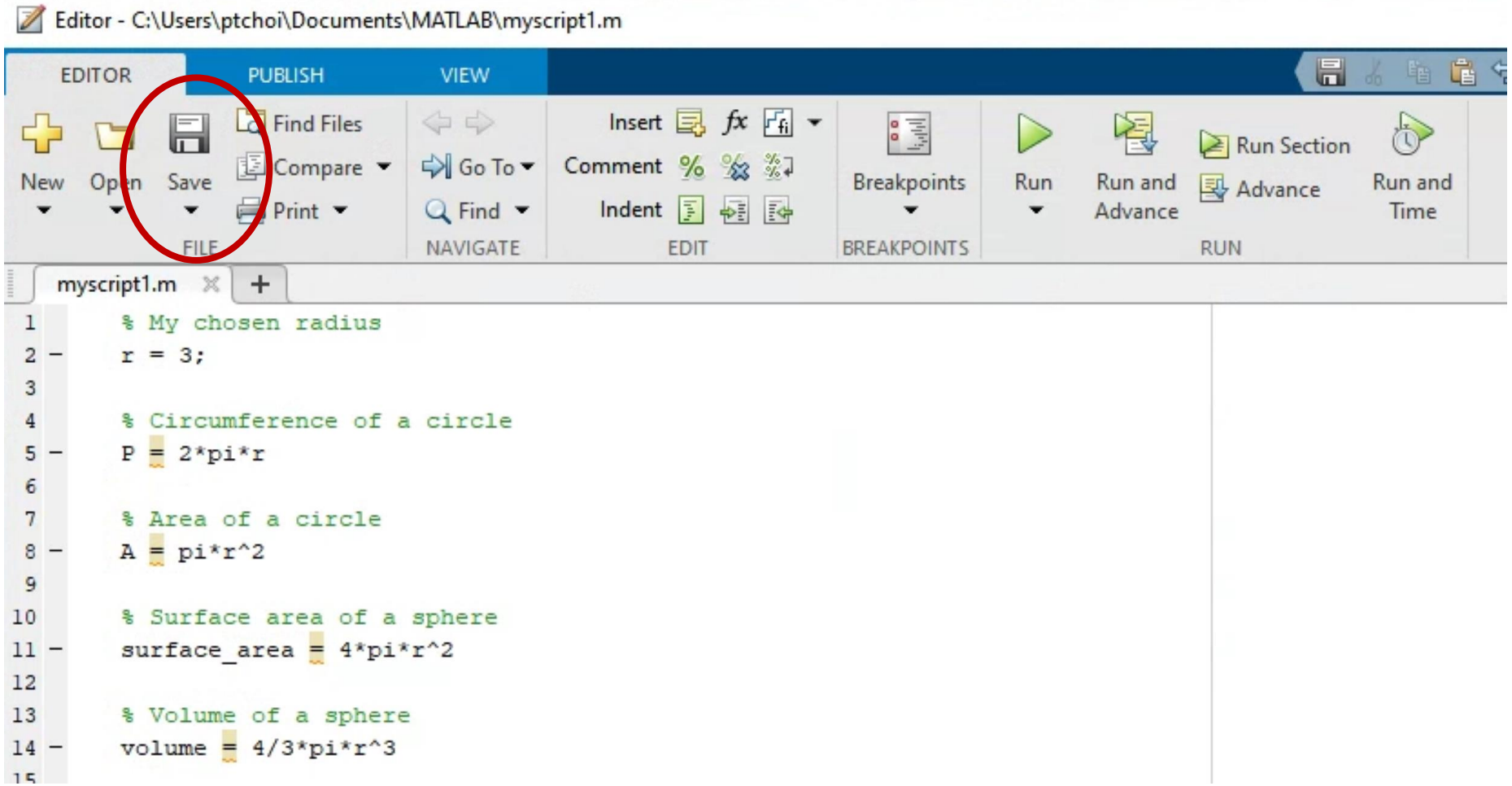


"%" is used for adding comments:
The content after % will not be executed

Very important for code readability!

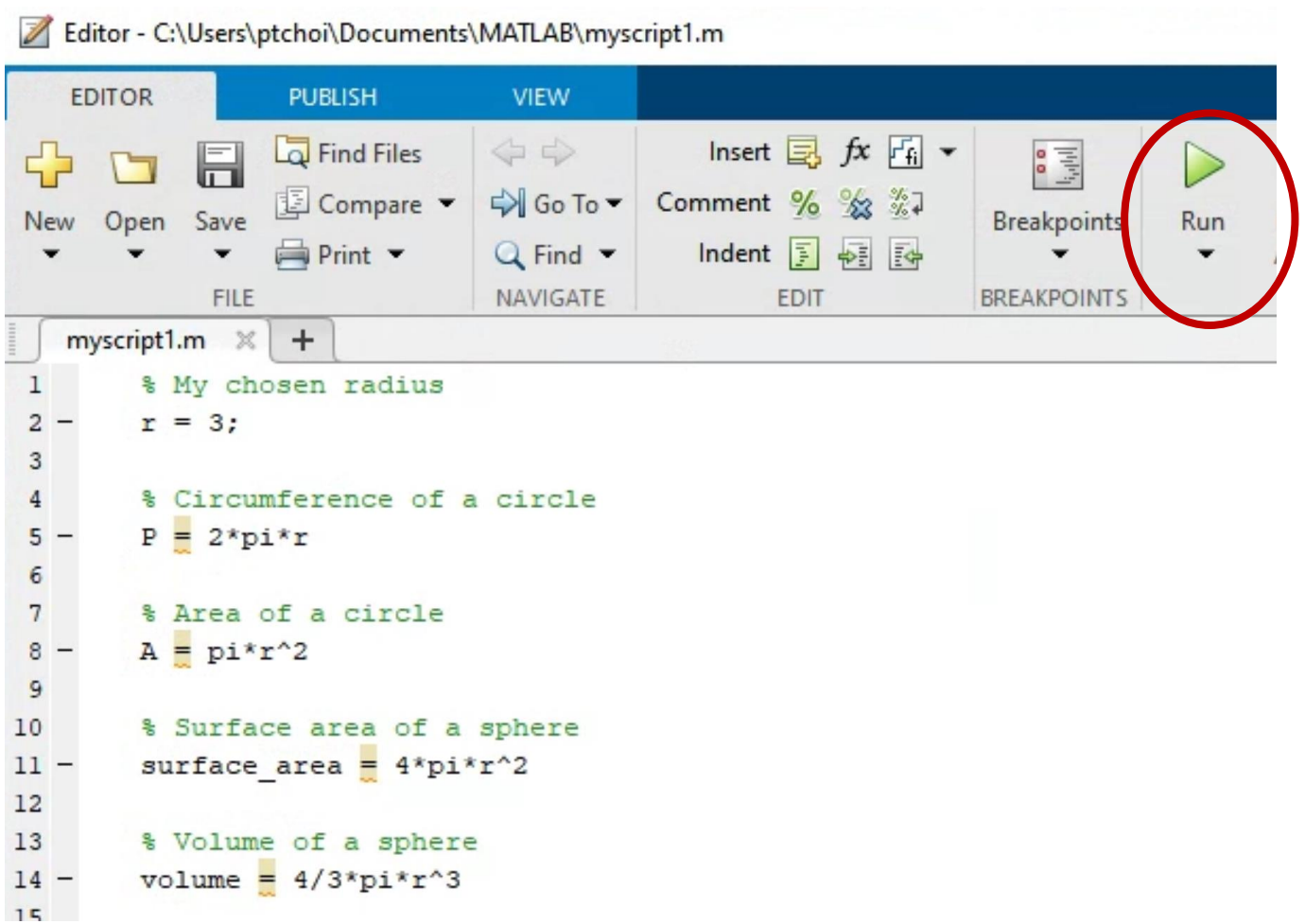
Saving a MATLAB script

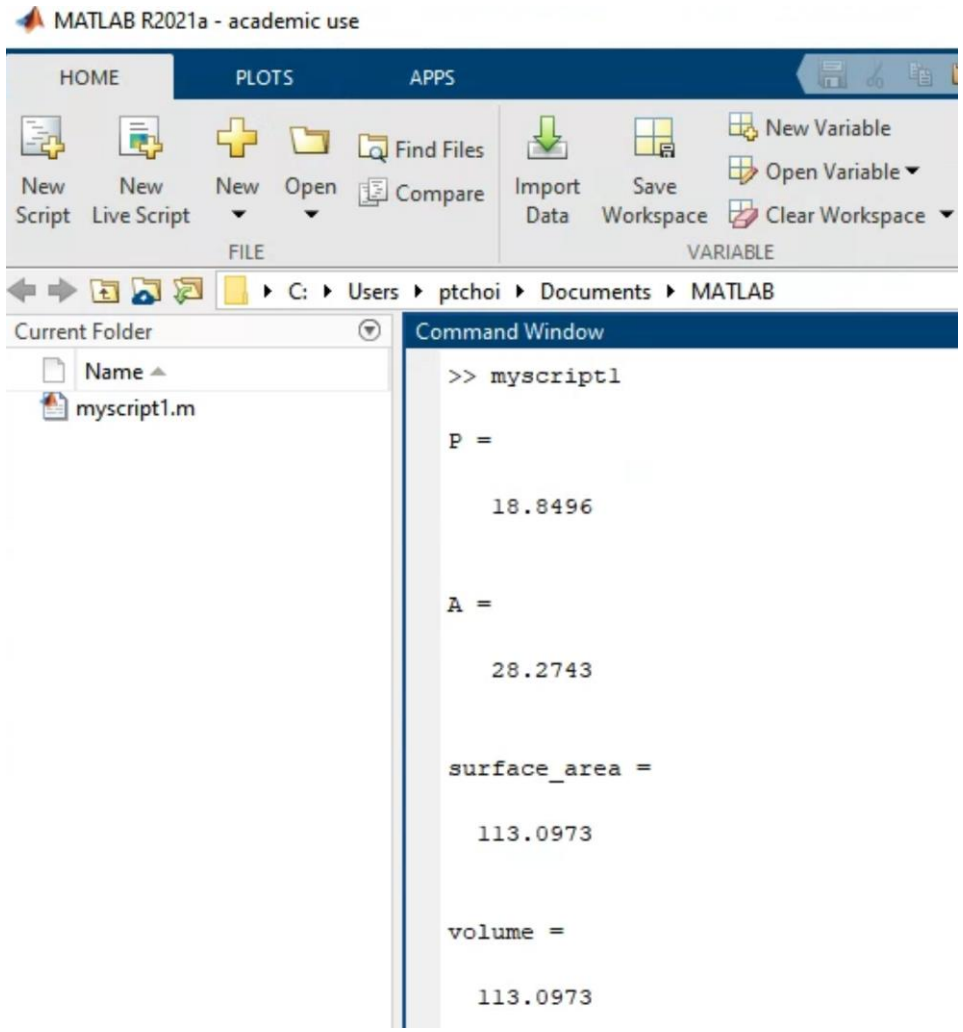
- Once a script is complete, it must be saved before it can be run. If you attempt to run an unsaved script, MATLAB will prompt you to save your script first. By default, MATLAB will save your script in the Current Folder.



Running a MATLAB script

- Once the script is saved, we can run it by clicking the Run button or directly typing the script name and pressing Enter in the Command Window.





Reminder: Lab session starting this week

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	[28]	[29]	[30]	[31]	

February

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						[1]
[2]	[3]	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1

March

Sun	Mon	Tue	Wed	Thu	Fri	Sat
2	[3]	[4]	[5]	[6]	[7]	[8]
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

April

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17		

Lecture 1- Lecture 13

Lab 1 - Lab 10 (40%)

Test 1 (30%)
Test 2 (30%)

Thank you!

Next time:

- Logical flow
- Loops and conditional statements