Lecture 01: A Quick Tutorial on MATLAB

ELEC 372 – Section UUA: Fundamentals of Control Systems

Course instructor: Prof. Walter Lucia

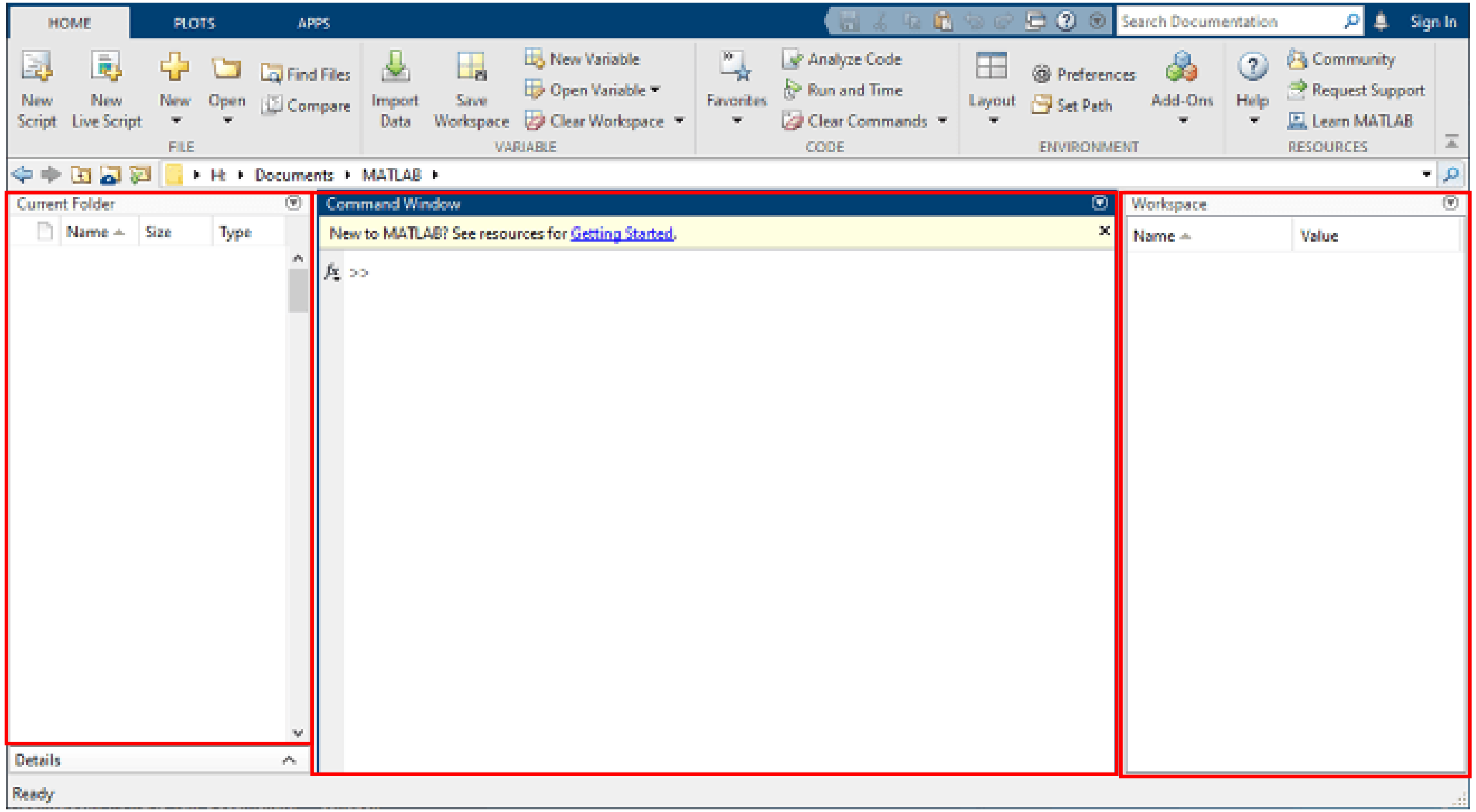
Tutorial TA: Mehran Attar[1]

[1] mehran.attar@mail.concordia.ca

Contents

- Overview of MATLAB Environment
- Matrices in MATLAB
- Plotting in MATLAB
- Flow Control in MATLAB (if, else, for, while)
- Functions in MATLAB (create a function in MATLAB)
- MATLAB Simulink

Overview of MATLAB Environment



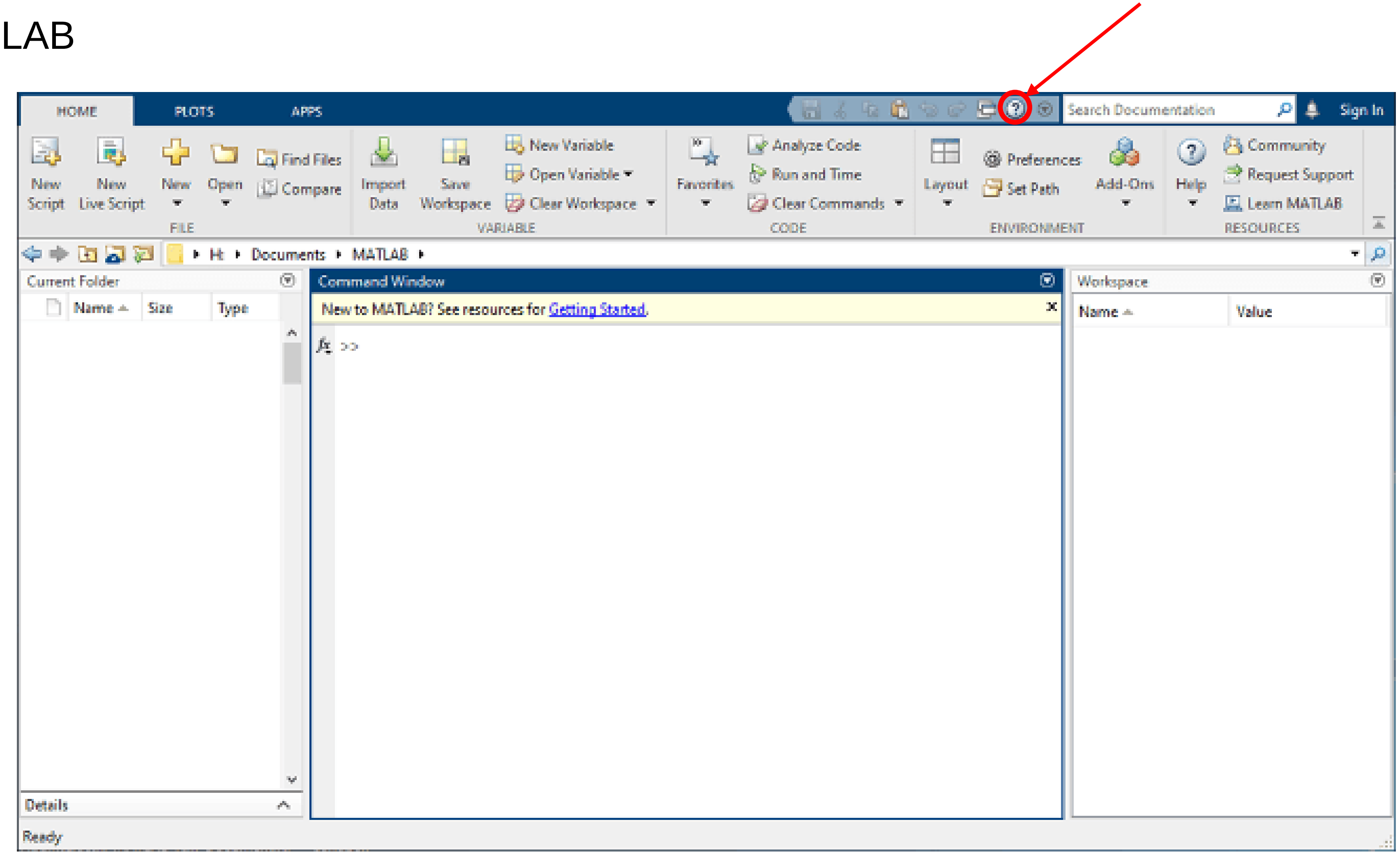
New to MATLAB? See resources for Getting Started.

```
>>
```

Ready

Overview of MATLAB Environment

Help of MATLAB



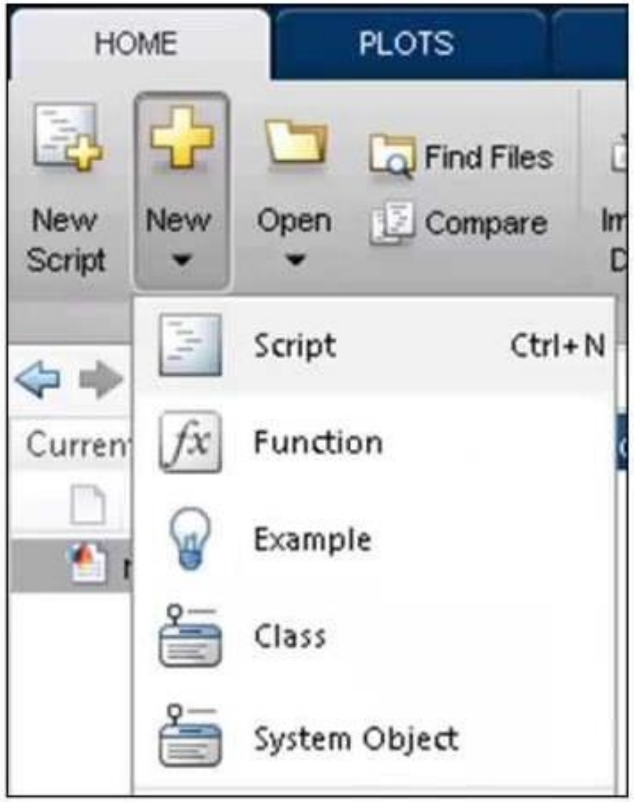
New to MATLAB? See resources for Getting Started.

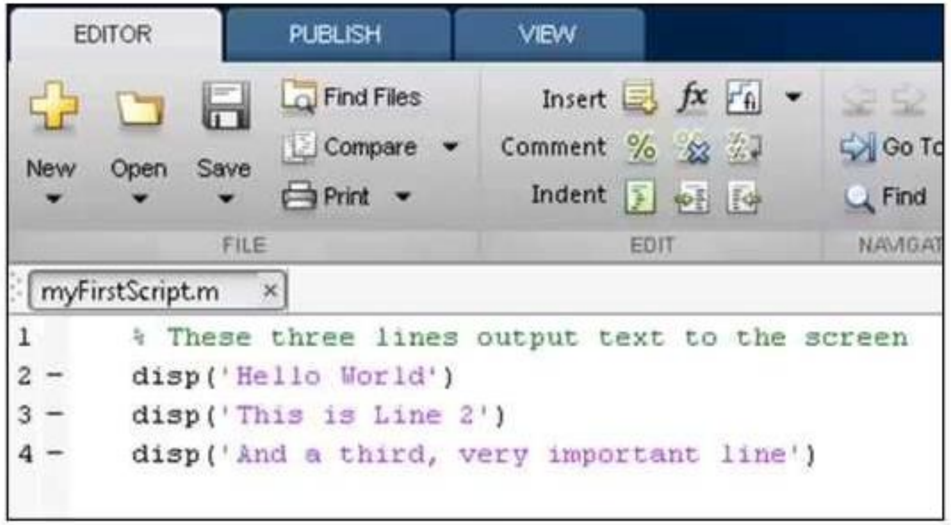
```
>>
```

Ready

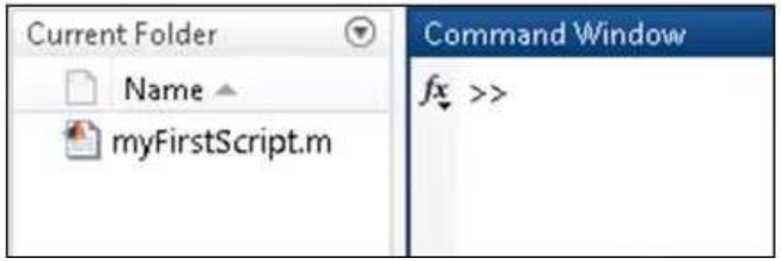
Overview of MATLAB Environment

Creating m-files (scripts) in MATLAB





```
% These three lines output text to the screen
disp('Hello World')
disp('This is Line 2')
disp('And a third, very important line')
```



Overview of MATLAB Environment

Variables in MATLAB

- Variable names are case sensitive.
- Variable names must start with a letter and can be followed by letters, digits and underscores. `(Error: Invalid expression)`
- Variables will be saved in the workspace.

Examples:

```
>> X = 2;
>> y_1_2 = 0.001;
>> New_var = 12;
```

Overview of MATLAB Environment

MATLAB Relational Operators

Operation	Operator
Less than	<
Less than or equal	<=
Greater than	>
Greater than or equal	>=
Equal to	==
Not equal to	~=

MATLAB Logical Operators

Operation	Operator
Not	~
And	&
Or	|

Matrices in MATLAB

- MATLAB treats all variables as matrices.
- Vectors are special forms of matrices and contain only one row OR one column.
- Scalars are matrices with only one row AND one column

Generating matrices

Example 1:

$x = 23$

```
>> x = 23;
```

Example 2: (row vector)

$y = \begin{bmatrix} 1 & 2 & -3 \end{bmatrix}$

```
>> y = [1, 2, -3];
```

Example 3: (column vector)

$$z = \begin{bmatrix} -10 \\ -0.5 \\ 2.5 \end{bmatrix}$$

```
>> x = [-10; -0.5; 2.5];
```

Example 4: (Matrix)

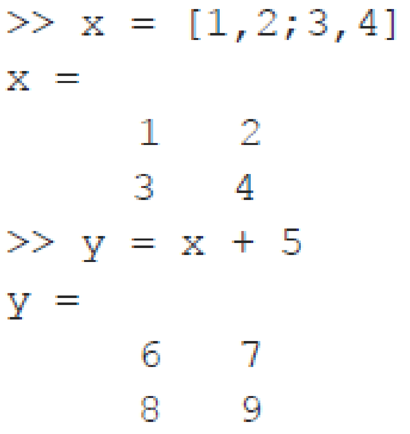
$$x = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
>> x = [1, 2, 3; 4, 5, 6; 7, 8, 9];
```

Matrices in MATLAB

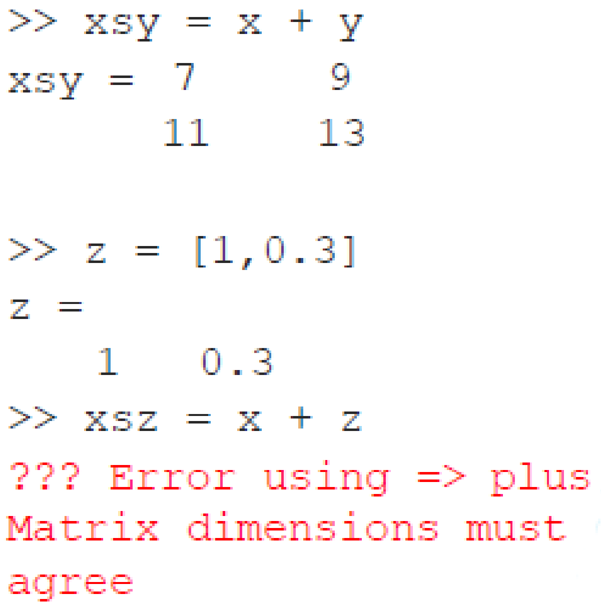
Matrix addition

- Increment all the elements of a matrix by a single value



```
>> x = [1,2;3,4]
x =
      1   2
      3   4
>> y = x + 5
y =
      6   7
      8   9
```

- Adding to matrices

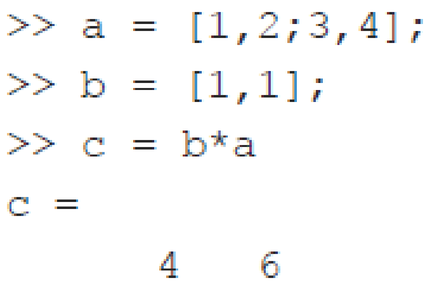


```
>> xsy = x + y
xsy =  7      9
      11     13

>> z = [1,0.3]
z =
   1   0.3
>> xsz = x + z
??? Error using => plus
Matrix dimensions must
agree
```

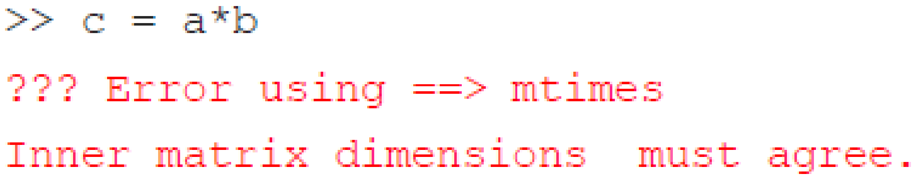
Matrices in MATLAB

Matrix multiplication

- Matrix multiplication

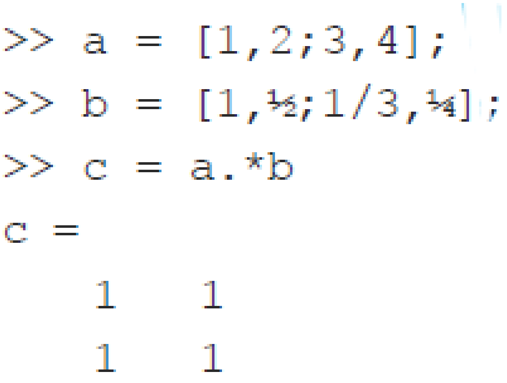


```
>> a = [1,2;3,4];
>> b = [1,1];
>> c = b*a
c =
      4    6
```



```
>> c = a*b
??? Error using ==> mtimes
Inner matrix dimensions  must agree.
```

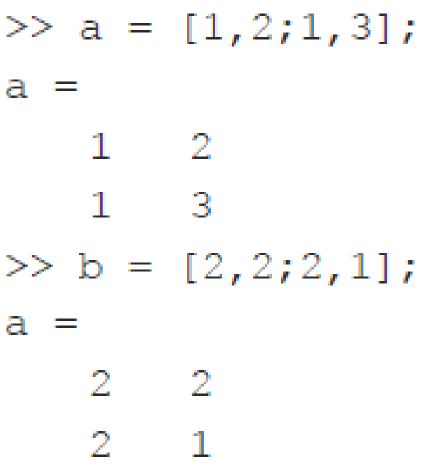
- Element wise multiplication



```
>> a = [1,2;3,4];
>> b = [1,½;1/3,¼];
>> c = a.*b
c =
    1    1
    1    1
```

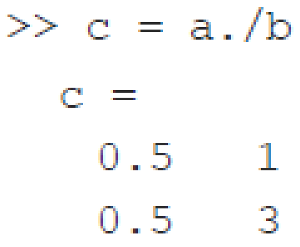
Matrices in MATLAB

Matrix element wise operations



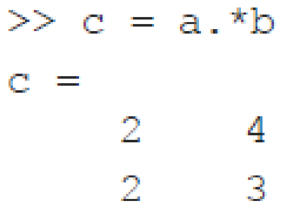
```
>> a = [1,2;1,3];
a =
    1   2
    1   3
>> b = [2,2;2,1];
a =
    2   2
    2   1
```

- Element wise division



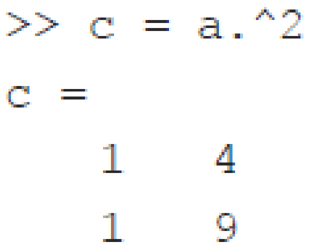
```
>> c = a./b
  c =
    0.5   1
    0.5   3
```

- Element wise multiplication

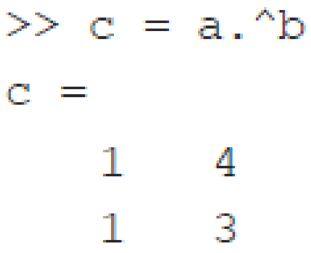


```
>> c = a.*b
c =
       2    4
       2    3
```

- Element wise power operation



```
>> c = a.^2
c =
     1    4
     1    9
```

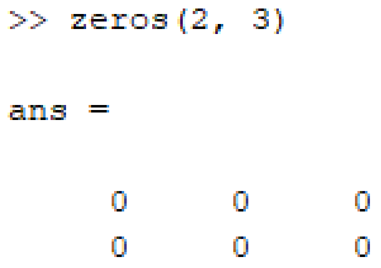


```
>> c = a.^b
c =
     1    4
     1    3
```

Matrices in MATLAB

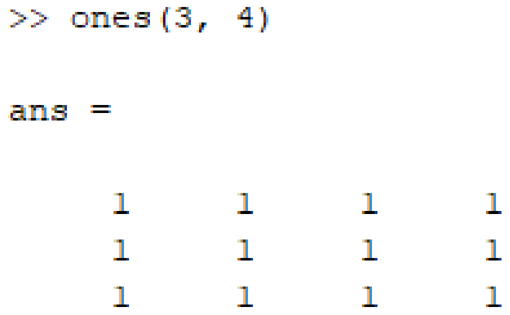
Matrix manipulation functions

- zeros matrix



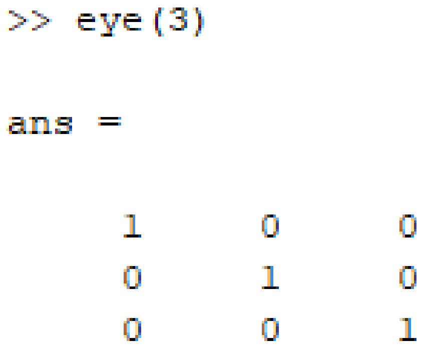
```
>> zeros(2, 3)

ans =

     0     0     0
     0     0     0
```

- ones matrix



```
>> ones(3, 4)

ans =

     1     1     1     1
     1     1     1     1
     1     1     1     1
```

- Identity matrix



```
>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1
```

- Other functions

Function	Description
`det(A)`	Determinant of matrix A
`inv(A)`	Inverse of matrix A
`eig(A)`	evaluates eigenvalues and eigenvectors of matrix A
`rank(A)`	rank of matrix A

Matrices in MATLAB

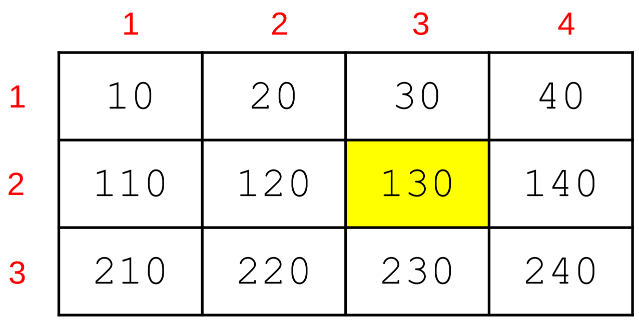
Indexing in matrices

- In MATLAB, indexes start at 1.
- Arrays can be sliced by using another array as an index.
- An array slice makes a copy of the underlying array, which means that the data values are copied to a new location and subsequent updates do not change the original matrix.

$$\texttt{Block\_matrix} = \begin{array}{c|c|c|c|c|} & 1 & 2 & 3 & 4 \\ \hline 1 & 10 & 20 & 30 & 40 \\ \hline 2 & 110 & 120 & 130 & 140 \\ \hline 3 & 210 & 220 & 230 & 240 \\ \hline \end{array}$$

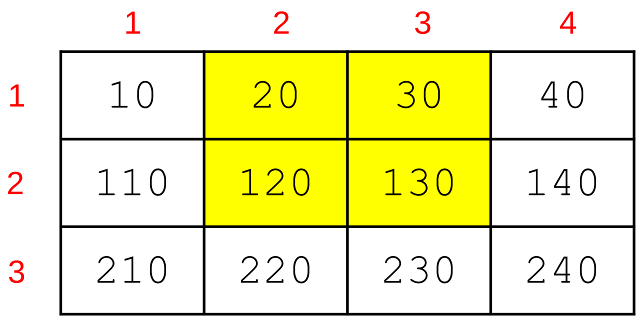
Matrices in MATLAB

Indexing and slicing



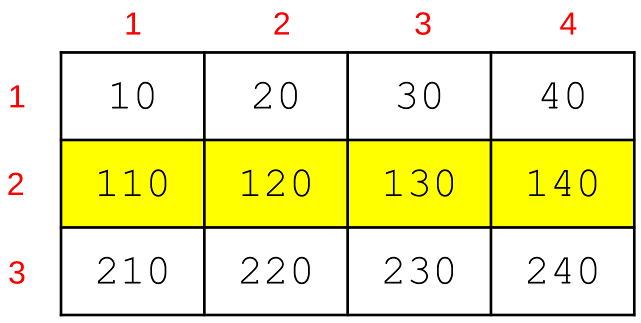
	1	2	3	4
1	10	20	30	40
2	110	120	130	140
3	210	220	230	240

`Block_matrix(2, 3)`



	1	2	3	4
1	10	20	30	40
2	110	120	130	140
3	210	220	230	240

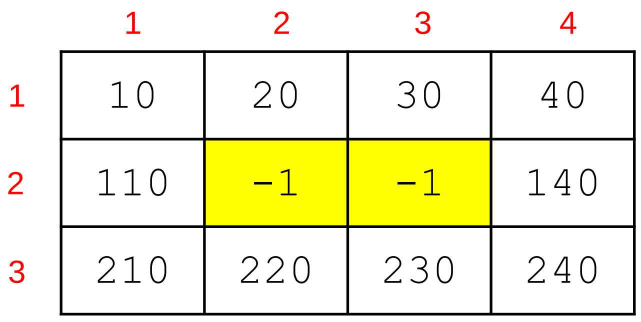
`Block_matrix(1:2, 2:3)`



	1	2	3	4
1	10	20	30	40
2	110	120	130	140
3	210	220	230	240

`Block_matrix(2, :)`

It is also possible to assign to slices



	1	2	3	4
1	10	20	30	40
2	110	-1	-1	140
3	210	220	230	240

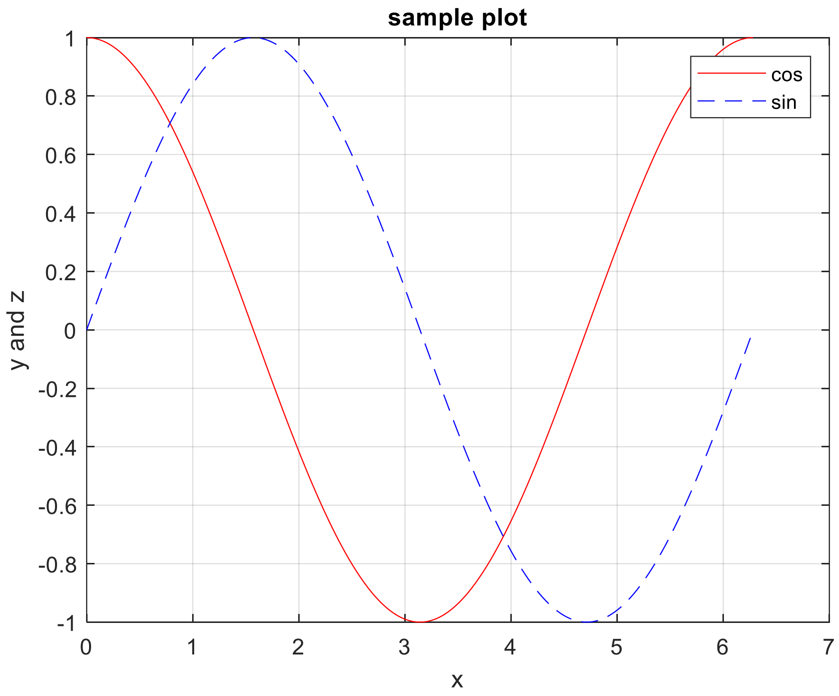
```
Block_matrix(2, 2:3) = -1
```

Plotting in MATLAB

2D plotting

plot $\sin(x)$ and $\cos(x)$ over $[0,\ 2\pi]$, on the same figure with different colors and line styles

```
x = 0:0.01:2*pi;
y = cos(x);
z = sin(x);

figure;
plot(x, y, '-r')
hold on
plot(x, z, '--b')
hold off
grid on

legend('cos', 'sin')
xlabel('x')
ylabel('y and z')
title('sample plot')
```



Vector generation methods

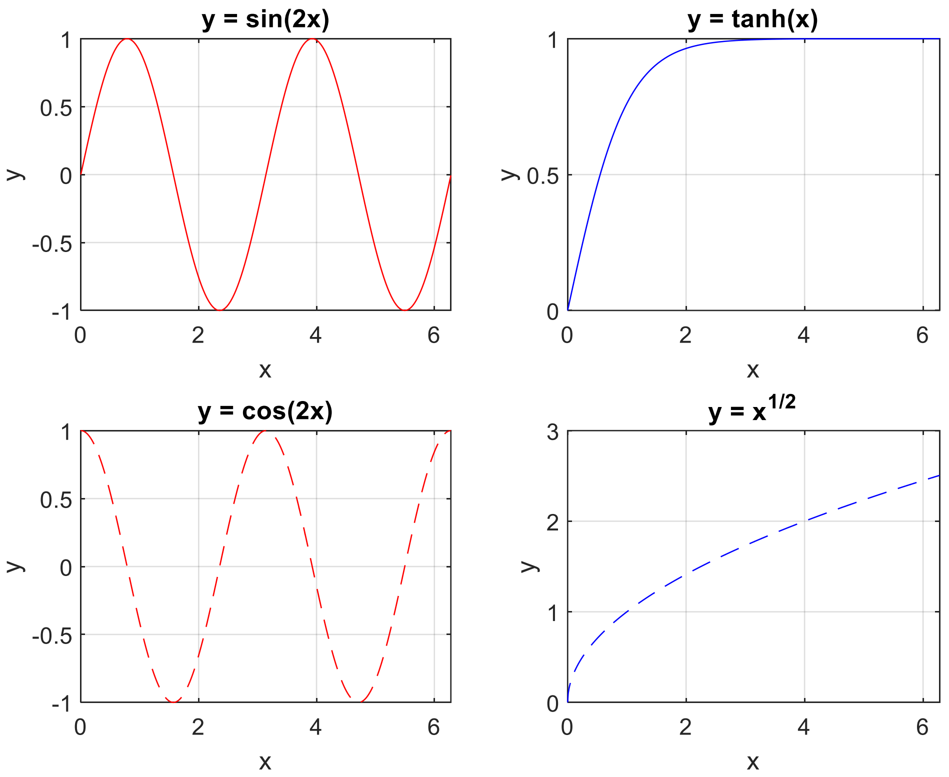
1. Using colon operator
   `X = 0:0.01:2*pi;`

2. Using linspace function
   `X = linspace(0, 2*pi, 1000)`

Plotting in MATLAB

Subplots in MATLAB

```
figure;
subplot(2, 2, 1)
plot(x, sin(2*x), '-r')
xlabel('x')
ylabel('y')
title('y = sin(2x)')
grid on

subplot(2, 2, 2)
plot(x, tanh(x), '-b')
xlabel('x')
ylabel('y')
title('y = tanh(x)')
grid on

subplot(2, 2, 3)
plot(x, cos(2*x), '--r')
xlabel('x')
ylabel('y')
title('y = cos(2x)')
grid on

subplot(2, 2, 4)
plot(x, sqrt(x), '--b')
xlabel('x')
ylabel('y')
title('y = x^{1/2}')
grid on
```



Flow Control in MATLAB

For loops

for loop to repeat specified number of times

Syntax

```
for index = values
   statements;
end
```

Example

```
for i = 1:10
    disp(i + 1)
end
```

Flow Control in MATLAB

If, elseif, else loops

Execute statements if condition is true

Syntax

```
if expression

    statements

elseif expression
    statements

else
    statements

end
```

Example

```
for i = 1:10
    if i >= 5
        disp('True')
    else
        disp('False')
    end
end
```

Example

```
for i = 1:10
    if i <= 2
        disp('i <= 2')
    elseif i>2 & i<=6
        disp('i>2 & i<=6')
    else
        disp('i>6')
    end
end
```

Flow Control in MATLAB

while loops

while loop to repeat when condition is true

Syntax

```
while expression

    statements

end
```

Example

```
n = 10;
while n > 2

    n = n-1;
    disp('n is still greater than 2')
end
disp(n)
```

Functions in MATLAB

How to define a function in MATLAB

Syntax

```
function [y1,...,yN] = myfun(x1,...,xM)

        statements

end
```

declares a function named myfun that accepts inputs x1,...,xM and returns outputs y1,...,yN.

Example

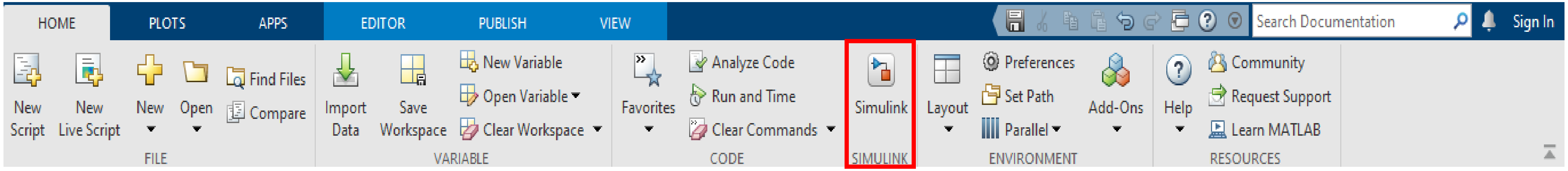
Define a function that accepts an input vector, calculates the average of the values, and returns a single result.

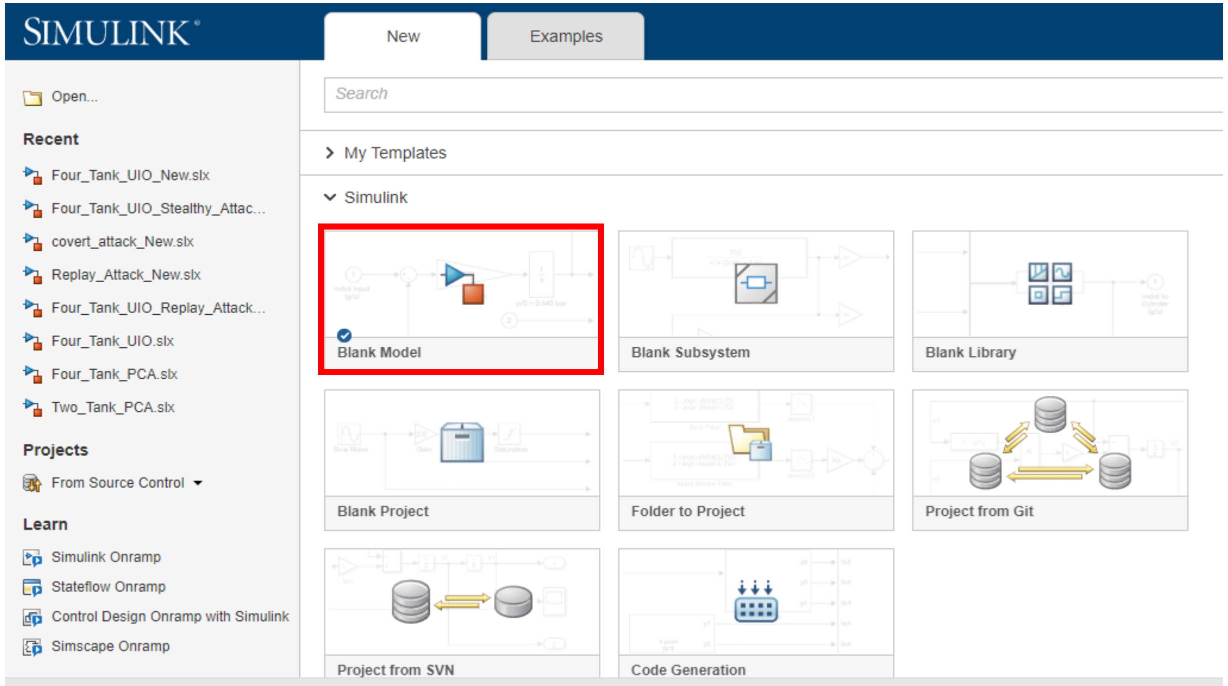
```
function ave = average(x)

    ave = sum(x(:))/size(x,2);

end
```

MATLAB Simulink

Simulink environment (open/create Simulink file)

Simulink® is a block diagram environment for multi domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems.

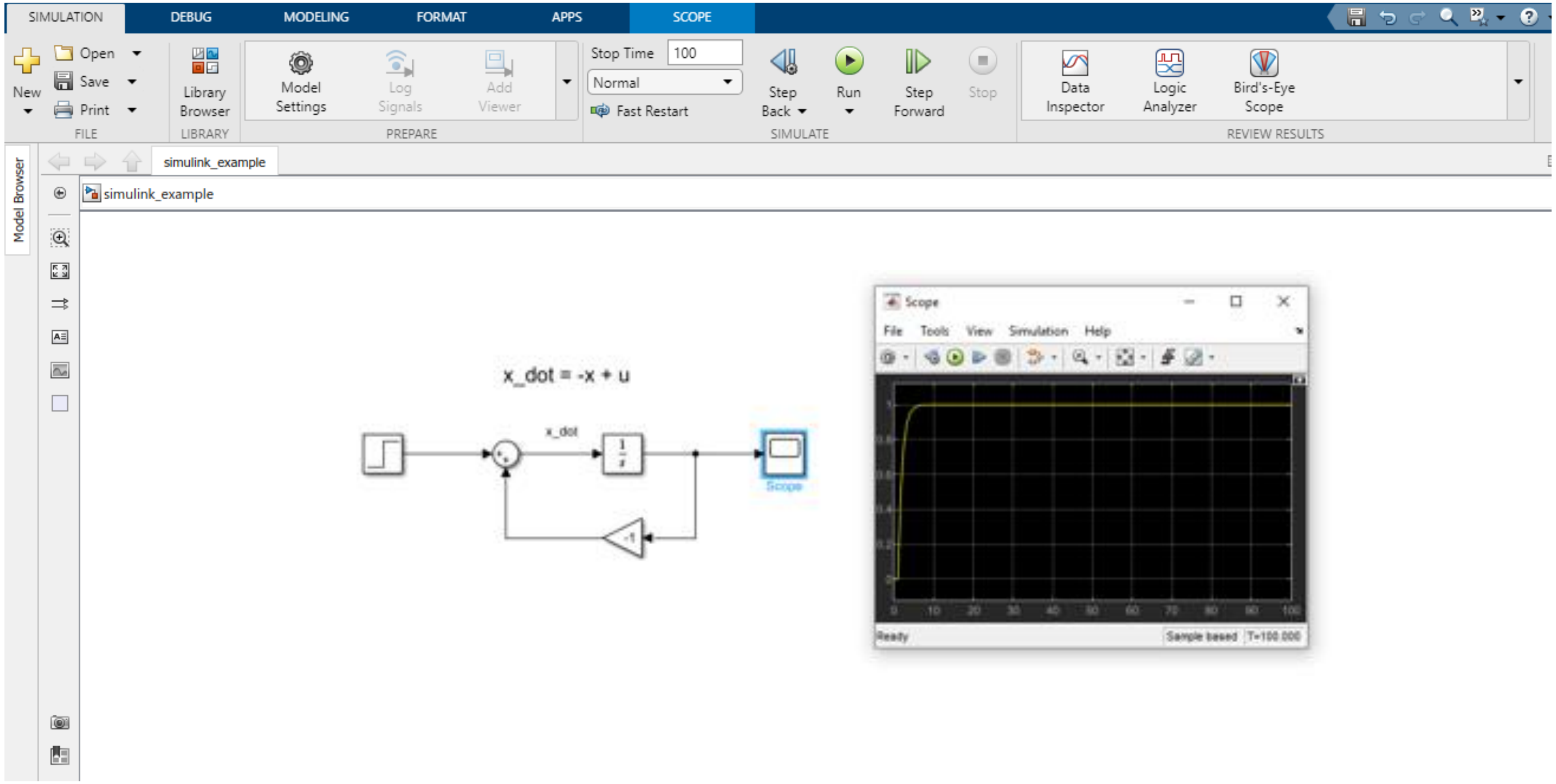




MATLAB Simulink

Simulink environment

implement and simulate the behavior of a differential equation in response to a step function



Stop Time 100

Normal

References

https://www.mathworks.com/help/matlab/getting-started-with-matlab.html

CONCORDIA.CA