

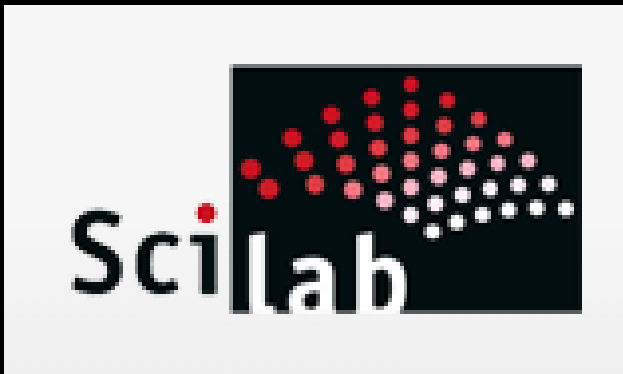
Introduction to MATLAB®

Introduction to Matrices

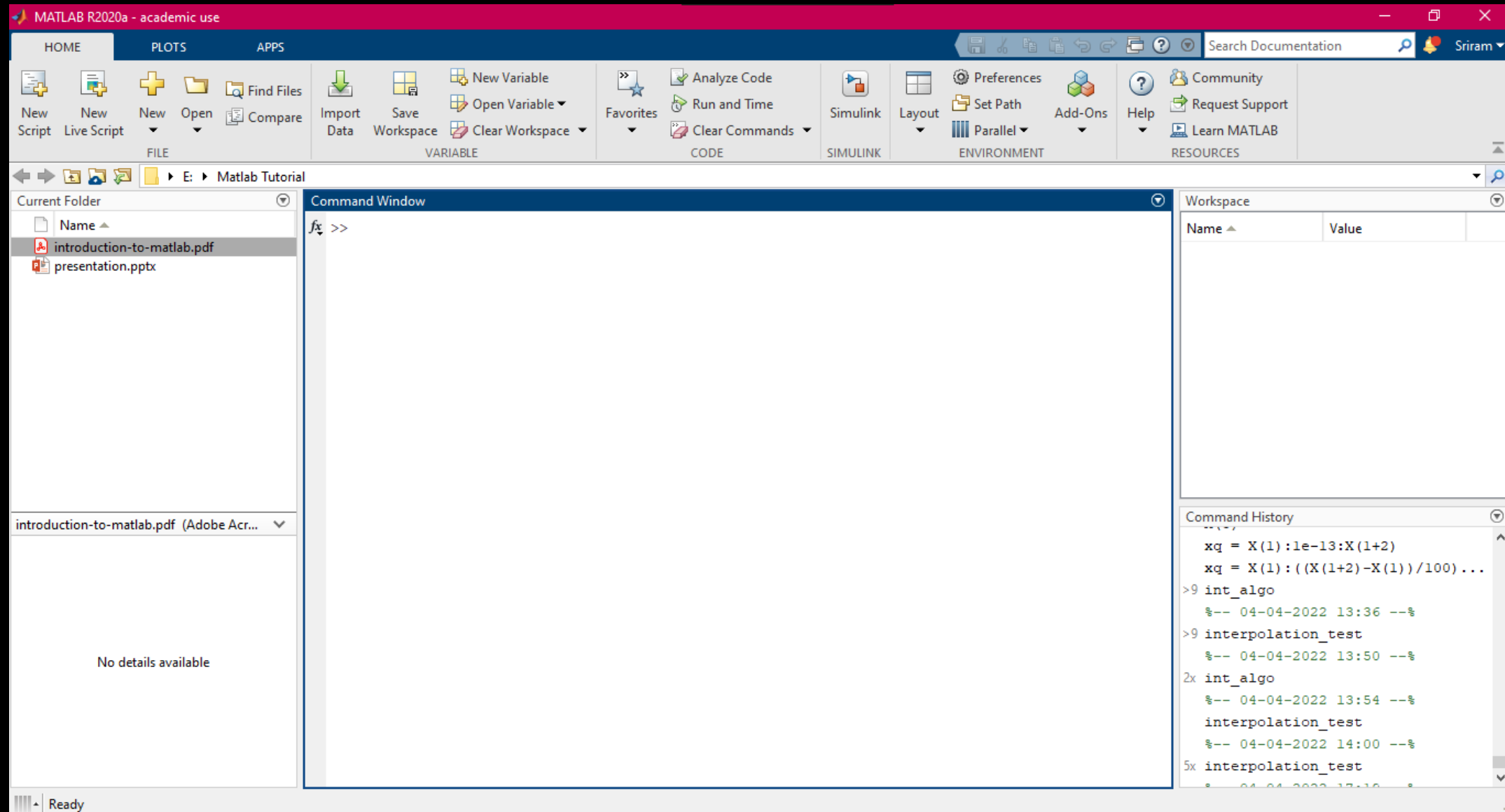
Sriram Krishnamurthy

Basics

- Matlab is a GUI based computing software generally used for Engineering applications.
- Similar to Wolfram Mathematica
- MATLAB stands for MATrix LABoratory
- Open Source alternatives:



The Command Window



The Command Window

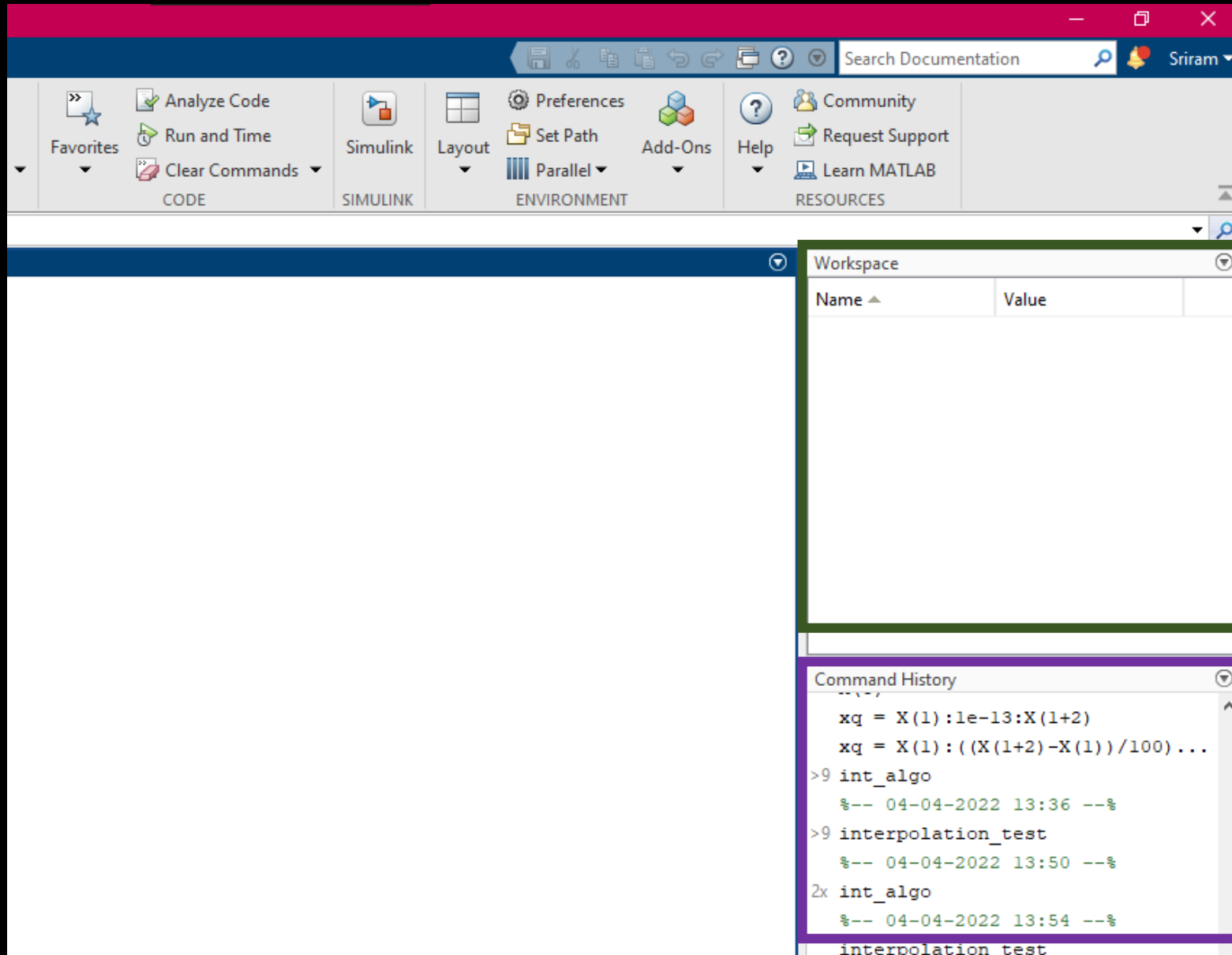
The image shows the MATLAB R2020a interface. The top menu bar includes HOME, PLOTS, and APPS. The HOME tab is active, showing various icons for file operations. The Path bar at the top indicates the current directory is E:\Matlab Tutorial. The Current Folder pane on the left lists files in the current directory: introduction-to-matlab.pdf and presentation.pptx. The Command Window on the right is empty, showing the MATLAB prompt >>. Annotations with green lines point to specific parts of the interface: 'Path' points to the Path bar, 'Contents in the current working folder' points to the Current Folder pane, and 'Details of the Selected file' points to the details pane below the file list.

Path

Contents in the current working folder

Details of the Selected file

The Command Window

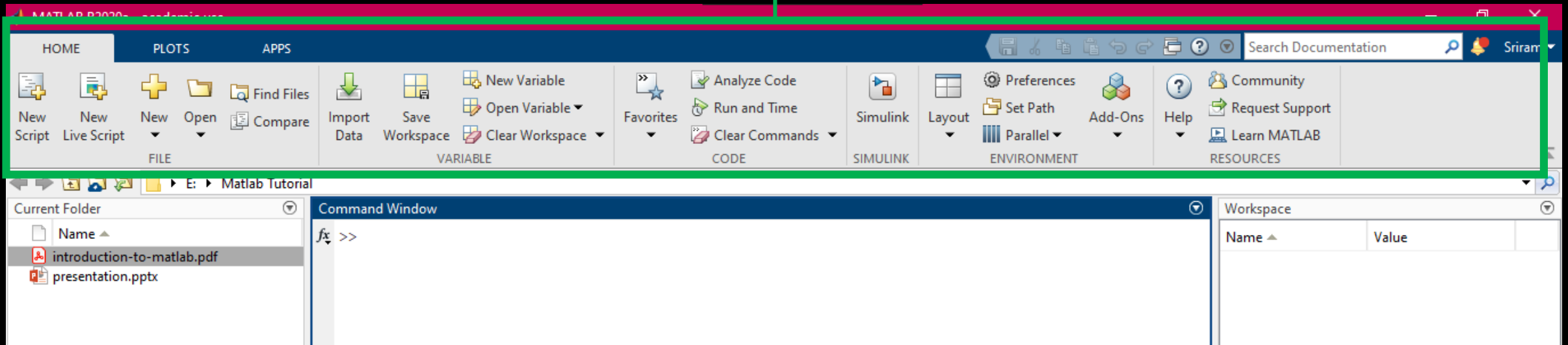


Work Space –
shows all the
variables – very
Important

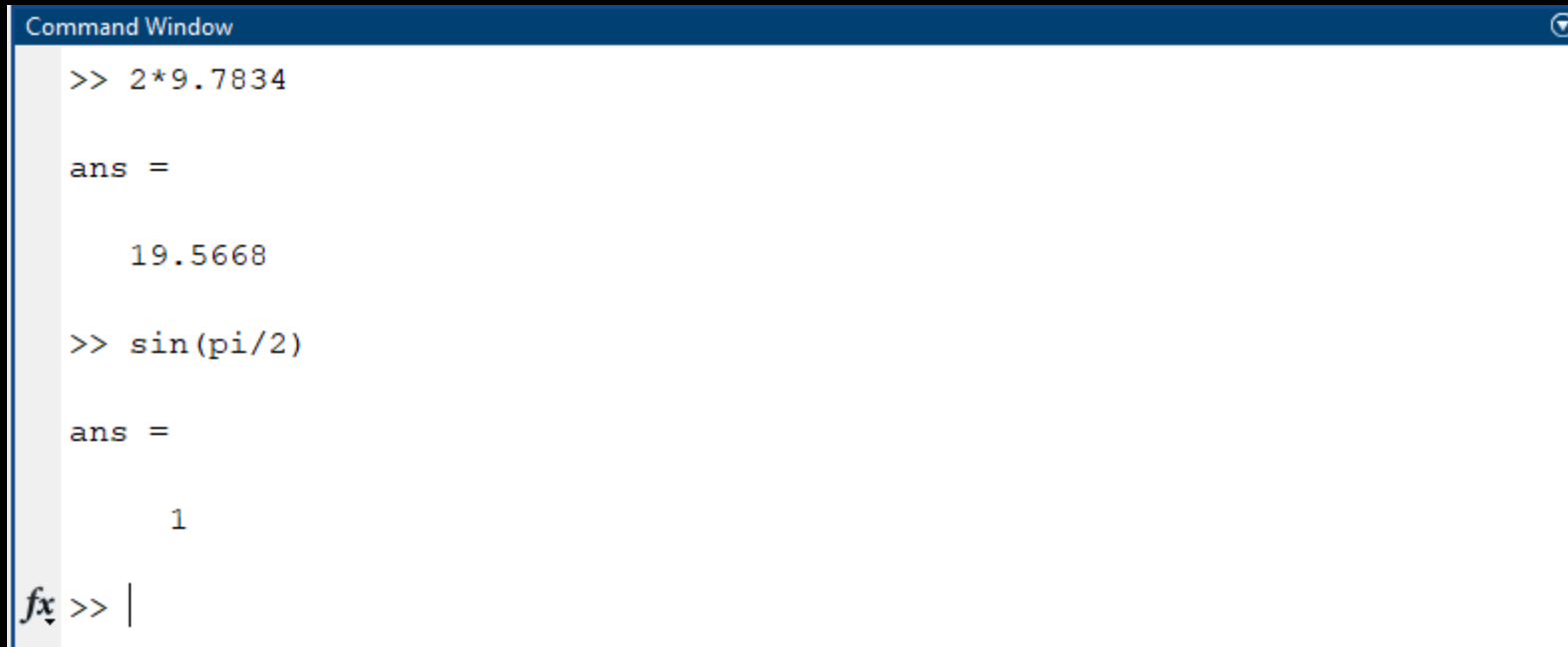
Command
History

The Command Window

Commonly Used tools - ToolBar



MATLAB – A high performance calculator

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and a small icon on the right. The main area is white and contains the following text:

```
>> 2*9.7834  
  
ans =  
  
    19.5668  
  
>> sin(pi/2)  
  
ans =  
  
     1  
  
fx >> |
```

The text is in a monospaced font. The prompt "fx >> |" is at the bottom left, indicating the current command line.

```
Command Window  
  
>> 2*9.7834  
  
ans =  
  
    19.5668  
  
>> sin(pi/2)  
  
ans =  
  
     1  
  
fx >> |
```

Variables

```
Command Window

>> x = 2

x =

    2

>> y = 3

y =

    3

>> x*y

ans =

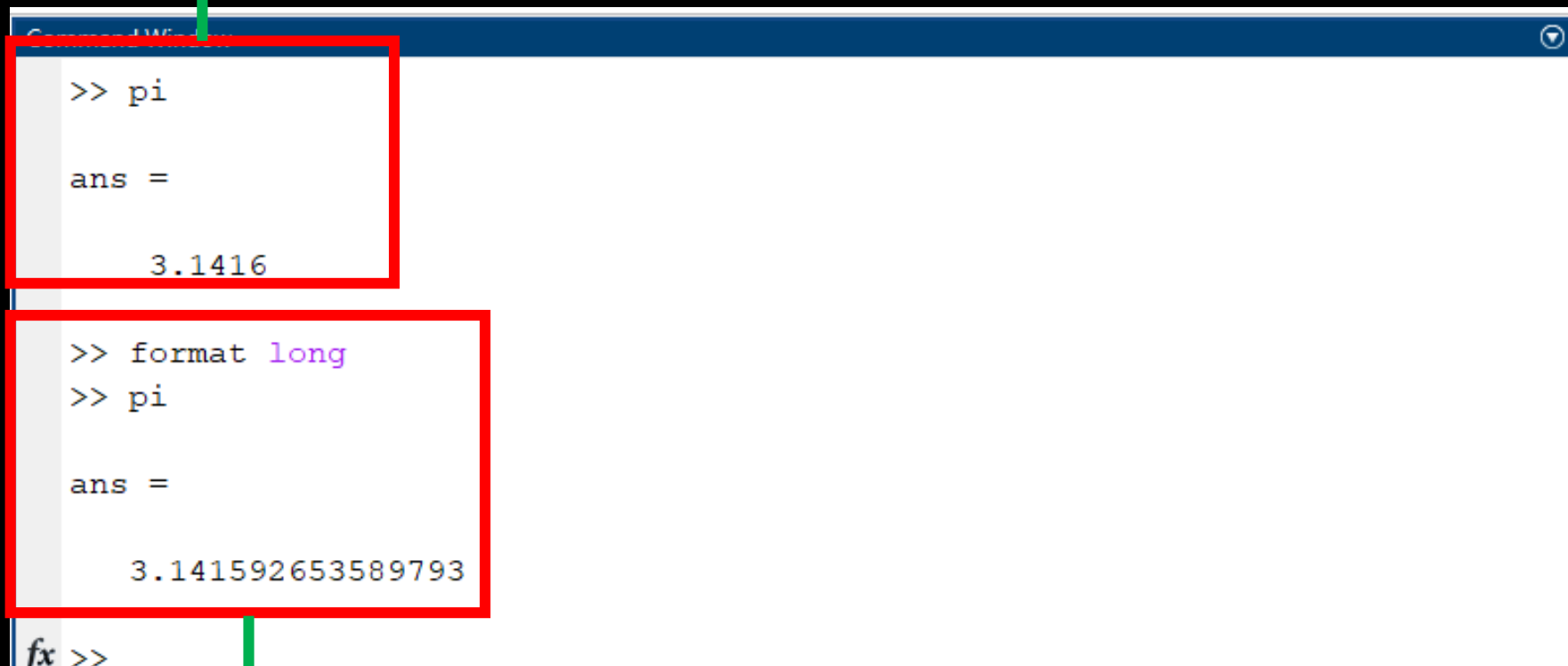
    6

fx >>
```

Variables - x

	x	×
	1x1 double	
	1	2
1	2	
2		
3		
4		
5		

Default format – 4 digits after the decimal
(just on the screen!) [format short]



A screenshot of the MATLAB Command Window. The window has a blue title bar. The command prompt is '>>'. The first command is 'pi', which returns 'ans = 3.1416'. This output is highlighted with a red rectangle. The second command is 'format long', which changes the display format. The third command is 'pi', which returns 'ans = 3.141592653589793'. This output is also highlighted with a red rectangle. A green vertical line is positioned to the left of the first red box, and another green vertical line is positioned to the left of the second red box. The bottom of the window shows the prompt 'fx >>'.

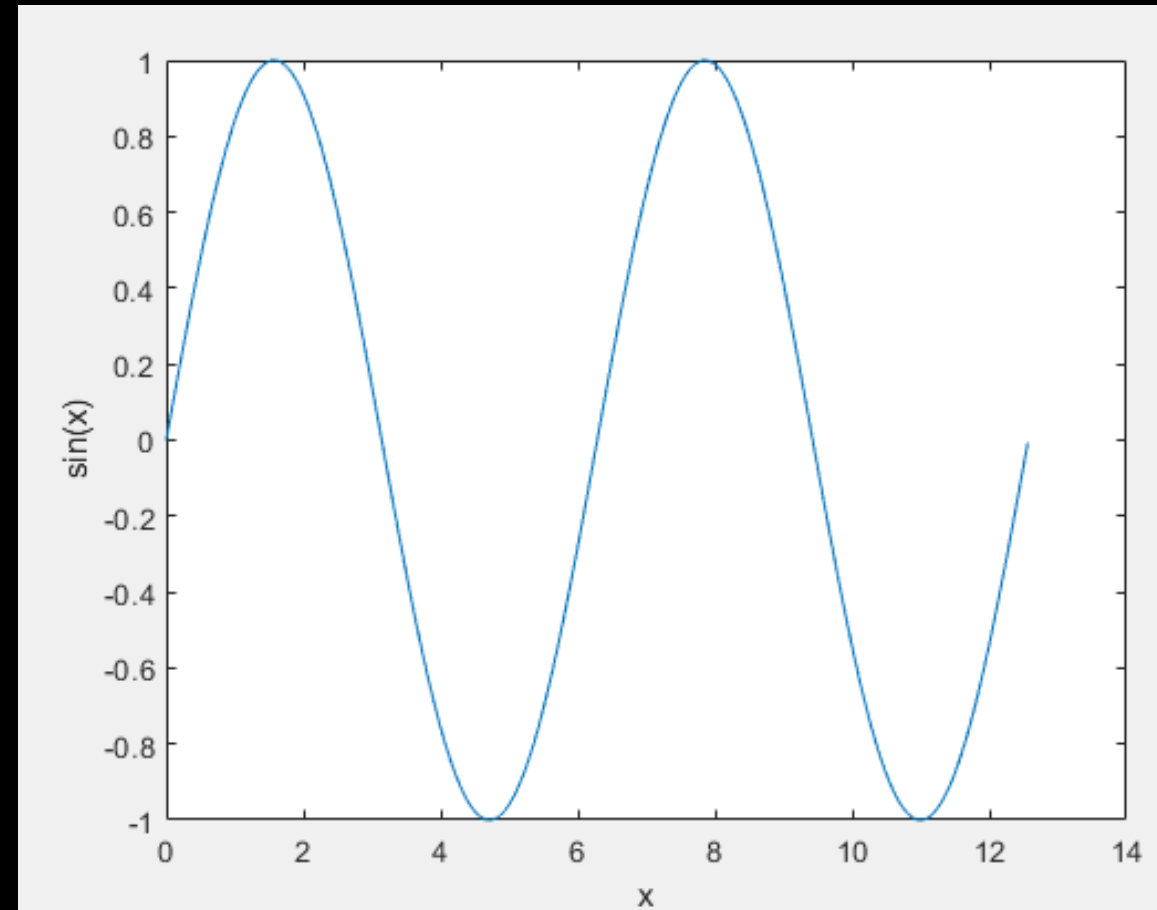
```
>> pi  
  
ans =  
  
3.1416  
  
>> format long  
>> pi  
  
ans =  
  
3.141592653589793  
  
fx >>
```

format long – 15 digits after the decimal
(actual value used for calculation)

Basic Plotting

Command Window

```
>> x = 0:0.01:4*pi;  
>> y = sin(x);  
>> plot(x,y);  
>> xlabel x  
>> ylabel sin(x)  
fx >>
```



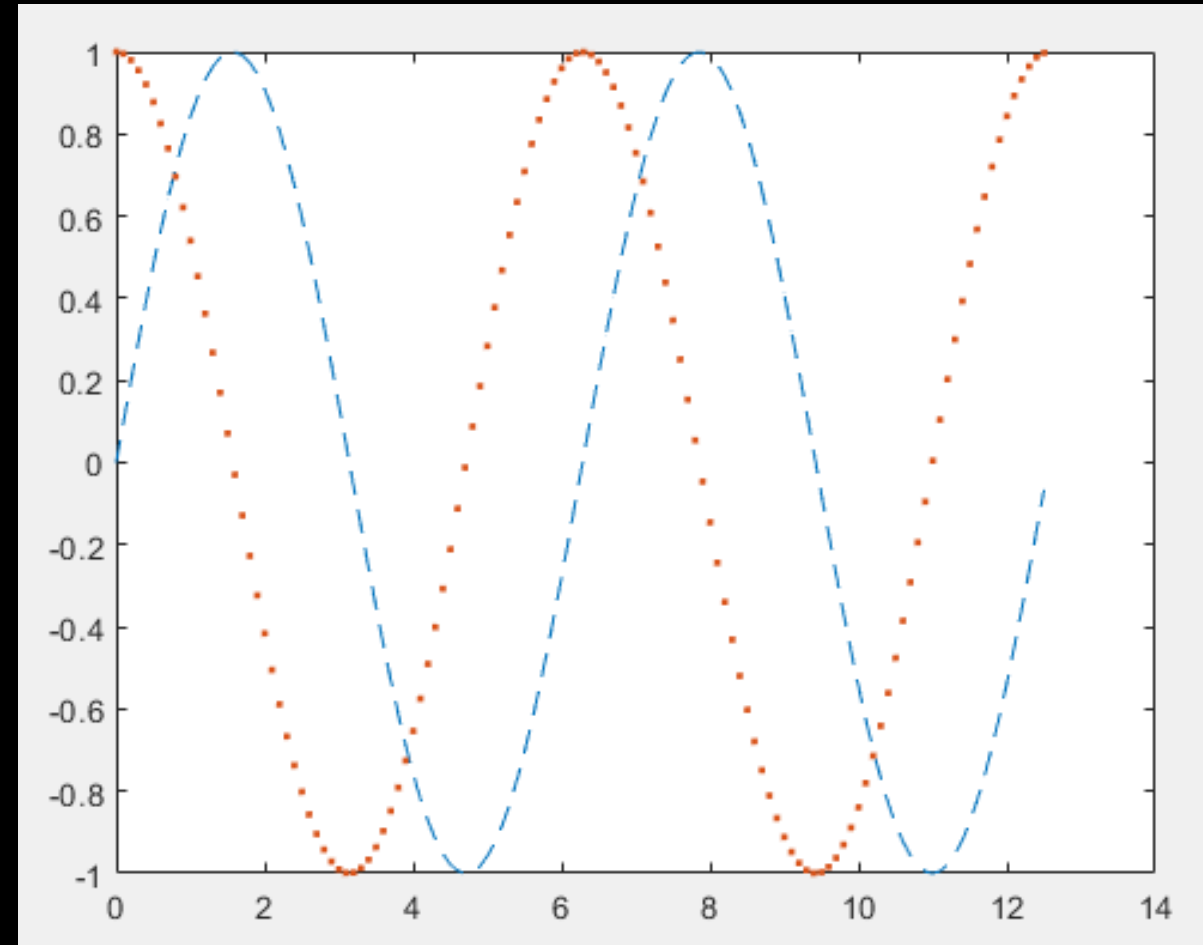
Basic Plotting

Command Window

```
>>  
>> x = 0:0.1:4*pi;  
>> y1 = sin(x);  
>> y2 = cos(x);  
>> plot(x,y1,'--',x,y2,'.')  
fx >>
```

Command Window

```
>> x = 0:0.1:4*pi;  
>> y1 = sin(x);  
>> y2 = cos(x);  
>> plot(x,y1);  
>> hold on;  
>> plot(x,y2,'--')  
fx >> |
```



Matrices

Row Vector

```
Command Window

>> a = [1 2 3 4 5]

a =

     1     2     3     4     5

fx >>
```

Column Vector

```
Command Window

>> b = [1;2;3;4;5]

b =

     1
     2
     3
     4
     5

fx >> |
```

Vector Transpose

Command Window

```
>> a = [1 2 3 4 5]
```

```
a =
```

```
    1    2    3    4    5
```

```
>> b = a'
```

```
b =
```

```
    1  
    2  
    3  
    4  
    5
```

```
>> c = b'
```

```
c =
```

```
    1    2    3    4    5
```

fx >>

Accessing elements

```
Command Window

>> a = [9 8 7 6 5 4 3 2]

a =

     9     8     7     6     5     4     3     2

>> a(3)

ans =

     7

>> a(4:6)

ans =

     6     5     4

fx >>
```

```
Command Window

>> a = [9 8 7 6 5 4 3 2]

a =

     9     8     7     6     5     4     3     2

>> a(6:end)

ans =

     4     3     2

fx >> |
```

- Begin with a square bracket, **[**
- Separate elements in a row with spaces or commas **(,)**
- Use a semicolon **(;)** to separate rows
- End the matrix with another square bracket, **]**.

```
Command Window

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

A =

     1     2     3
     4     5     6
     7     8     9

fx >>
```

```
Command Window

>> size(A)

ans =

     3     3
```

Command Window

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

```
A =
```

1	2	3
4	5	6
7	8	9

```
>> A(2,2)
```

```
ans =
```

5

fx >> |

Command Window

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

```
A =
```

1	2	3
4	5	6
7	8	9

```
>> A(1:2,1:2)
```

```
ans =
```

1	2
4	5

fx >> |


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

A =

     1     2     3
     4     5     6
     7     8     9

>> A(3,3) = 0

A =

     1     2     3
     4     5     6
     7     8     0

fx >>
```

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

A =

     1     2     3
     4     5     6
     7     8     9

>> A(:,2) = []

A =

     1     3
     4     6
     7     9

fx >> |
```

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

A =

     1     2     3
     4     5     6
     7     8     9

>> A([1 3],[3 1])

ans =

     3     1
     9     7

fx >> |
```

Row 1&3, Column 3&1

- $A(:,j)$ is the j th column of A
- $A(i,:)$ is the i th row, and
- $A(\text{end},:)$ picks out the last row of A

```
Command Window
>> A(:,2)

ans =

     2
     5
     8

fx >>
```

```
Command Window
>> A(3,:)

ans =

     7     8     9

fx >>
```

```
Command Window
>> A(end,:)

ans =

     7     8     9

fx >> |
```

Command Window

```
>> B = [4/5 , 7.23*tan(x) , sqrt(6); ...  
        1/x^2 , 0 , 3/(x*log(x)); ...  
        x-7 , sqrt(3) , x*sin(x)]
```

B =

0.8000	3.9498	2.4495
4.0000	0	-8.6562
-6.5000	1.7321	0.2397

fx >> |

```
>> B'
```

ans =

0.8000	4.0000	-6.5000
3.9498	0	1.7321
2.4495	-8.6562	0.2397

fx >>

A =

1	2	3
4	5	6
7	8	9

>> B

B =

0.8000	3.9498	2.4495
4.0000	0	-8.6562
-6.5000	1.7321	0.2397

>> C = [A, B]

C =

1.0000	2.0000	3.0000	0.8000	3.9498	2.4495
4.0000	5.0000	6.0000	4.0000	0	-8.6562
7.0000	8.0000	9.0000	-6.5000	1.7321	0.2397

Command Window

```
>> eye(2,3)
```

```
ans =
```

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

```
fx >> |
```

Command Window

```
>> ones(3)
```

```
ans =
```

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

```
fx >>
```

Command Window

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

```
A =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> diag(A)
```

```
ans =
```

```
    1
    5
    9
```

```
fx >>
```

Command Window

```
>> eye(4)
```

```
ans =
```

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

```
fx >> |
```

Command Window

```
>> rand(3,3)
```

```
ans =
```

```
    0.0975    0.9575    0.9706
    0.2785    0.9649    0.9572
    0.5469    0.1576    0.4854
```

```
fx >> |
```

Arithmetic Operations

Command Window

```
>> A = rand(2,2)
```

```
A =
```

```
    0.8003    0.4218  
    0.1419    0.9157
```

```
>> B = rand(2,3)
```

```
B =
```

```
    0.7922    0.6557    0.8491  
    0.9595    0.0357    0.9340
```

```
>> C = A*B
```

```
C =
```

```
    1.0387    0.5398    1.0735  
    0.9910    0.1257    0.9758
```

```
fx >> |
```

Command Window

```
>> A = [1 5 9]
```

```
A =
```

```
    1    5    9
```

```
>> B = [3;4;2]
```

```
B =
```

```
    3  
    4  
    2
```

```
>> C = A*B
```

```
C =
```

```
   41
```

```
fx >> |
```

Element by element multiplication

```
Command Window

A =

     1     2     3
     4     5     6
     7     8     9

>> B = [10 20 30;40 50 60;70 80 90]

B =

    10    20    30
    40    50    60
    70    80    90

>> C = A.*B

C =

    10    40    90
   160   250   360
   490   640   810

fx >>
```

```
Command Window

>> A = [1 2;3 4;5 6]

A =

     1     2
     3     4
     5     6

>> B = [5 6]

B =

     5     6

>> C = A.*B

C =

     5    12
    15    24
    25    36

fx >>
```


Solving Linear Equations

$$Ax = b$$

$$x + 2y + 3z = 1$$

$$4x + 5y + 6z = 1$$

$$7x + 8y = 1$$

Solution is given by:

$$x = A^{-1}b$$

Method 1

```
A =  
  
     1     2     3  
     4     5     6  
     7     8     0  
  
>> b = [1;1;1]  
  
b =  
  
     1  
     1  
     1  
  
>> x = inv(A)*b  
  
x =  
  
    -1.0000  
     1.0000  
    -0.0000
```

Method 2 Gaussian elimination

```
A =  
  
     1     2     3  
     4     5     6  
     7     8     0  
  
>> b = [1;1;1]  
  
b =  
  
     1  
     1  
     1  
  
>> x = A\b  
  
x =  
  
    -1.0000  
     1.0000  
    -0.0000
```

Command Window

```
>> A = [1 2 3;4 5 6;7 8 0]
```

```
A =
```

```
1    2    3
4    5    6
7    8    0
```

```
>> eig(A)
```

```
ans =
```

```
12.1229
-0.3884
-5.7345
```

Command Window

```
>> A = [1 2 3;4 5 6;7 8 0]
```

```
A =
```

```
1    2    3
4    5    6
7    8    0
```

```
>> [v,d] = eig(A)
```

```
v =
```

```
-0.2998  -0.7471  -0.2763
-0.7075   0.6582  -0.3884
-0.6400  -0.0931   0.8791
```

```
d =
```

```
12.1229    0    0
0  -0.3884    0
0    0  -5.7345
```

Command Window

```
>> A = [1 2 3;4 5 6;7 8 0]
```

```
A =
```

```
1    2    3
4    5    6
7    8    0
```

```
>> rank(A)
```

```
ans =
```

```
3
```

Eigen Vectors

Eigen values