

MAT202E

INTRODUCTION TO MATLAB

Outline

- MATLAB Environment
- Variables
- Manipulating Variables
- Basic Plotting
- Script File
- Polynomial Calculation

MATLAB Environment

The image shows the MATLAB R2012b interface with three main components highlighted by red boxes and red text labels:

- CURRENT PATH:** The left sidebar shows the current folder structure. The path is `C:\Users\Hasan\Desktop`. The files listed are: 8600KMZO, db, output_files, altera_installer.external.exe, blg452_liste.xlsx, deneme.m, deneme.qpf, deneme.qsf, main.vhd, MATLAB R2012b.lnk, matlab_intro.pptx, MIT6_0941AP10_lec01.pdf, and myfunc.m.
- COMMAND WINDOW:** The central area shows the command history and output. The commands entered are:

```
ans =  
2*x + 1  
>> myfunc(1,2)  
ans =  
2  
>> myfunc(1,2)  
ans =  
2  
>> myfunc(1,2)  
ans =  
2  
>> [m n]=myfunc(1,2)  
m =  
2  
n =  
0.5000
```
- WORKSPACE:** The right sidebar shows the workspace variables. The variables are: ans (Value: 2, Min: 2), m (Value: 2, Min: 2), n (Value: 0.5000, Min: 0.5000), x (Value: <1x1 sym>), and y (Value: <1x1 sym>).

The **COMMAND HISTORY** window at the bottom right shows the sequence of commands executed, including `plot(x,y)`, `x=linspace(0, 2*pi, 10);`, `x=linspace(0, 2*pi, 1000);`, `y=sin(x);`, `plot(x,y)`, `syms`, `syms x`, `y=x^2+x`, `derivative(y,x)`, `derivative(y)`, `int(y)`, `int(y,x)`, `log(6)`, `diff(y)`, `myfunc(1,2)`, and `[m n]=myfunc(1,2)`.

Variables

```
>> x=12

x =

    12

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

mx =

     1     2     3
     4     5     6

>> mx2=[5; 6; 7]

mx2 =

     5
     6
     7

>> vec=1:0.5:4

vec =

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000
```

```
>> array1=zeros(2)

array1 =

     0     0
     0     0

>> array2=zeros(3,1)

array2 =

     0
     0
     0
```

- Do not use these names!!!

i and **j** indicate complex numbers.

pi has the value 3.1415...

ans stores last result.

Inf is infinity.

NaN represents "Not a Number".

- Accessing array elements

```
>> arr=[ 9 8 7 6 5 4]

arr =

     9     8     7     6     5     4

>> arr(1)

ans =

     9

>> arr(6)

ans =

     4
```

Manipulating Variables

- Scalar operations

```
>> x=12;
>> y=15;
>> x*y

ans =

    180

>> a=(1+i)/(2-i)

a =

    0.2000 + 0.6000i

>> (-2)^4

ans =

    16

>> exp(4)

ans =

    54.5982
```

- Vector operations

```
>> A=[3 2;1 -1]

A =

     3     2
     1    -1

>> B=[-2; 3]

B =

    -2
     3

>> A*B

ans =

     0
    -5

>> A*A

ans =

    11     4
     2     3
```

- Element-wise operation

```
>> vec=[-2 4 6 9]

vec =

    -2     4     6     9

>> vec2=[-1 -4 12 -3]

vec2 =

    -1    -4    12    -3

>> vec.*vec2

ans =

     2   -16    72   -27

>> vec./vec2

ans =

    2.0000   -1.0000    0.5000   -3.0000

>> vec*vec2'

ans =

    31
```

```
>> A.^2
```

```
ans =

     9     4
     1     1
```

\neq

```
>> A^2
```

```
ans =

    11     4
     2     3
```

```
>> exp(A)
```

```
ans =

    20.0855    7.3891
     2.7183    0.3679
```

```
>> [exp(A(1,1)) exp(A(1,2)); exp(A(2,1)) exp(A(2,2))]
```

```
ans =

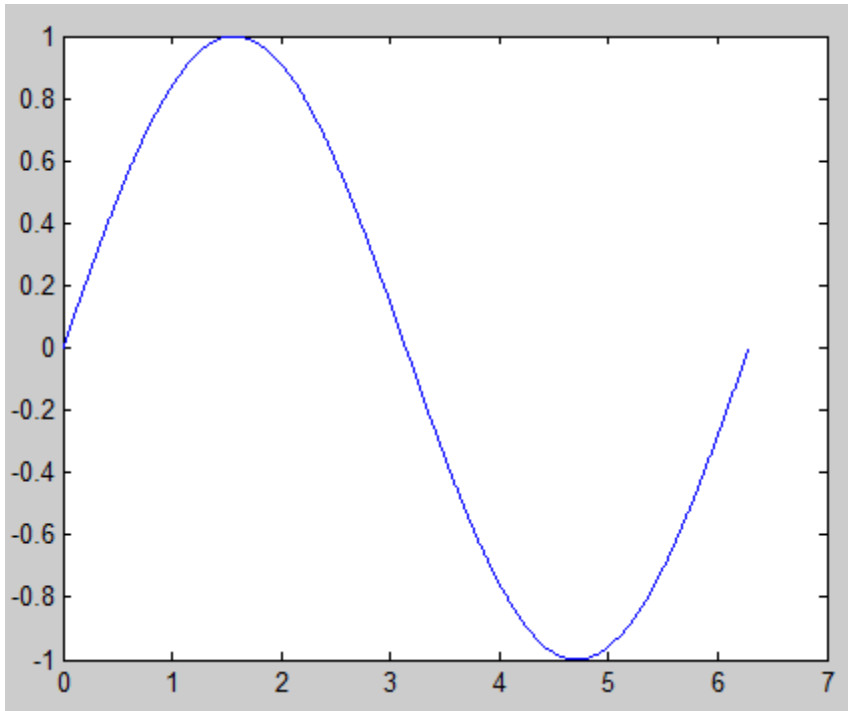
    20.0855    7.3891
     2.7183    0.3679
```

Some Built-in Functions

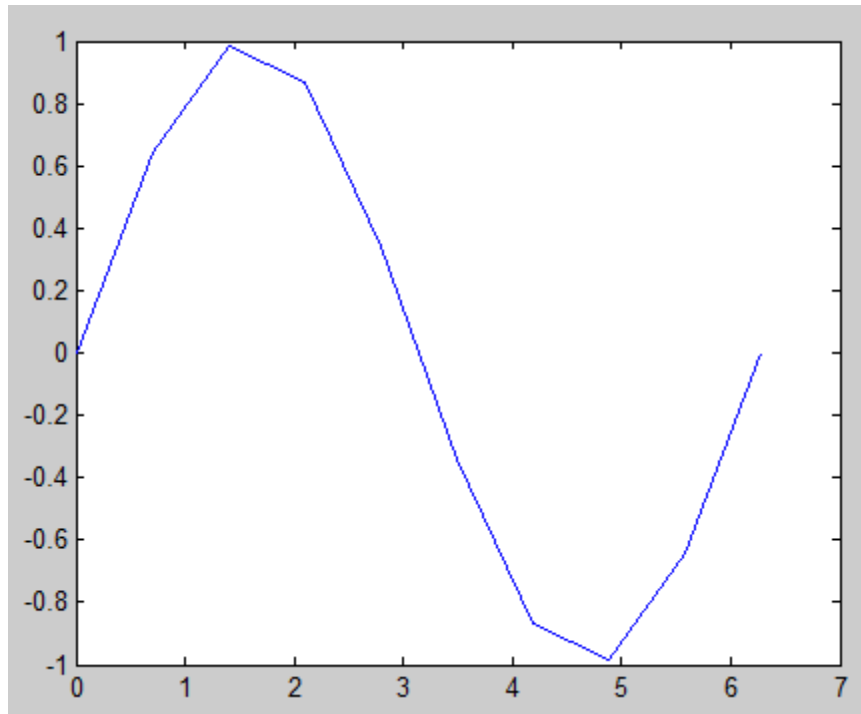
log(4), sqrt(5), log10(12),
floor(2.1), ceil(1.9),
abs(2+i)...

Basic Plotting

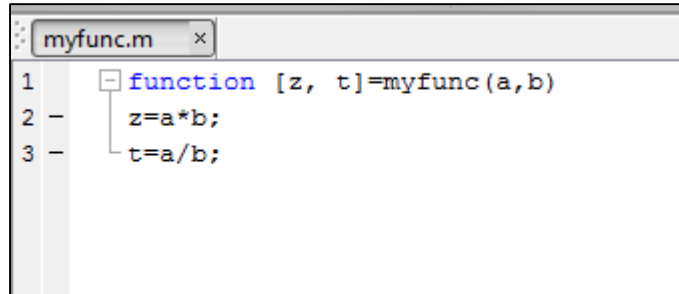
```
>> x=linspace(0, 2*pi, 1000);  
>> y=sin(x);  
>> plot(x,y)
```



```
>> x=linspace(0, 2*pi, 10);  
>> y=sin(x);  
>> plot(x,y)
```



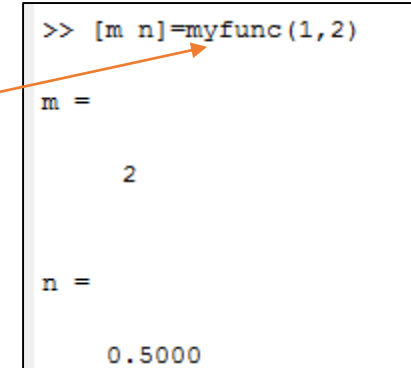
Script File



```
myfunc.m x
1 function [z, t]=myfunc(a,b)
2 -   z=a*b;
3 -   t=a/b;
```

A screenshot of a MATLAB script file editor window titled 'myfunc.m'. The window contains three lines of code: a function definition line 'function [z, t]=myfunc(a,b)', followed by two assignment lines 'z=a*b;' and 't=a/b;'. Line numbers 1, 2, and 3 are visible on the left margin.

z and **t** are return variables.
a and **b** are input parameters.



```
>> [m n]=myfunc(1,2)
m =
    2
n =
    0.5000
```

A screenshot of a MATLAB command window. It shows the command '[m n]=myfunc(1,2)' being executed. The output is displayed in two lines: 'm =' followed by the value '2', and 'n =' followed by the value '0.5000'. An orange arrow points from the function name 'myfunc' in the command window to the function definition in the script file editor on the left.

Calling **myfunc** from command window or .m file.

Some Polynomial Calculations

- Derivative and Integral

```
>> syms x
>> y=x^2+3

y =

x^2 + 3

>> int(y, x)

ans =

(x*(x^2 + 9))/3

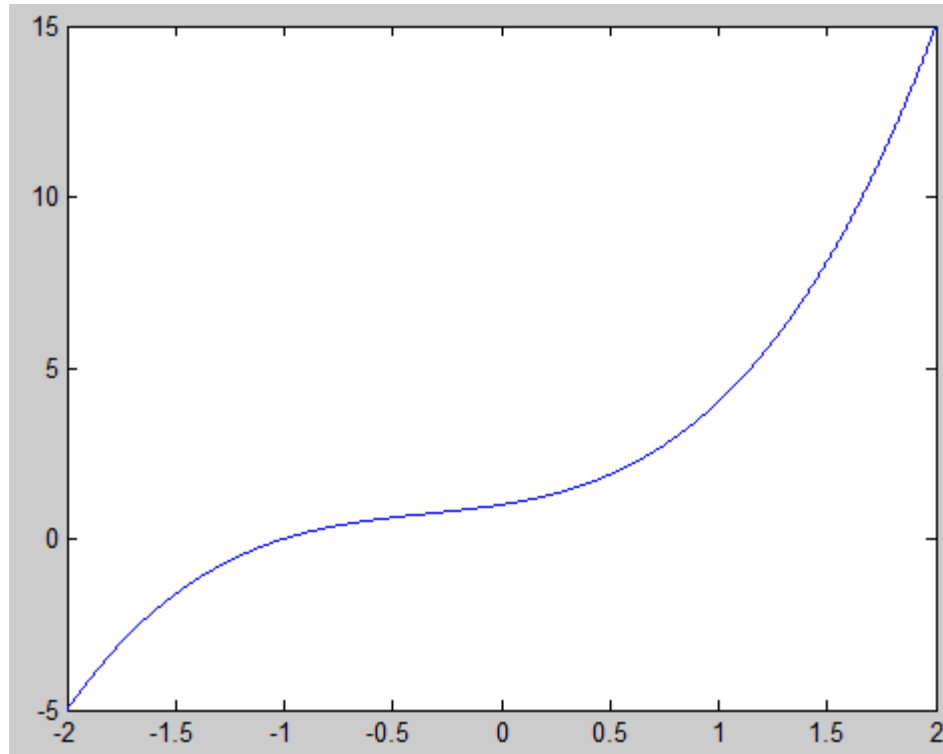
>> diff(y, x)

ans =

2*x
```

- Simple Plot

```
>> x=-2:0.001:2;
>> y= polyval([1 1 1 1], x);
>> plot(x,y)
```



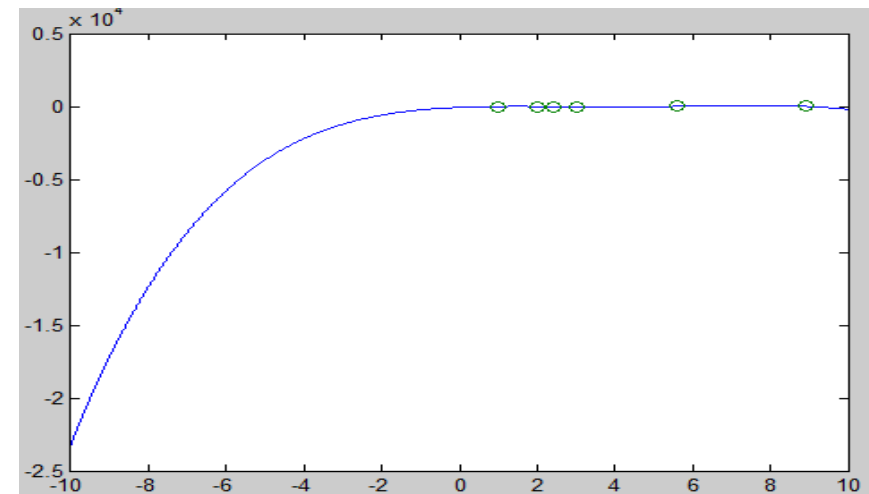
- Poly Curve Fitting

```
>> a=[1 2 3 2.4 5.6 8.9];
>> b=[-4 -8 -22 -1 3 6];
>> p=polyfit(a,b,4)

p =

-0.5956    10.4258   -56.9126   108.6773   -66.1274

>> x=-10:0.001:10;
>> y=polyval(p,x);
>> plot(x,y)
>> hold on
>> scatter(a,b)
```



References

- MIT 6.094 Introduction to Programming in MATLAB (ocw.mit.edu)
- MATLAB 2012b Help Guide