Introduction to Matlab

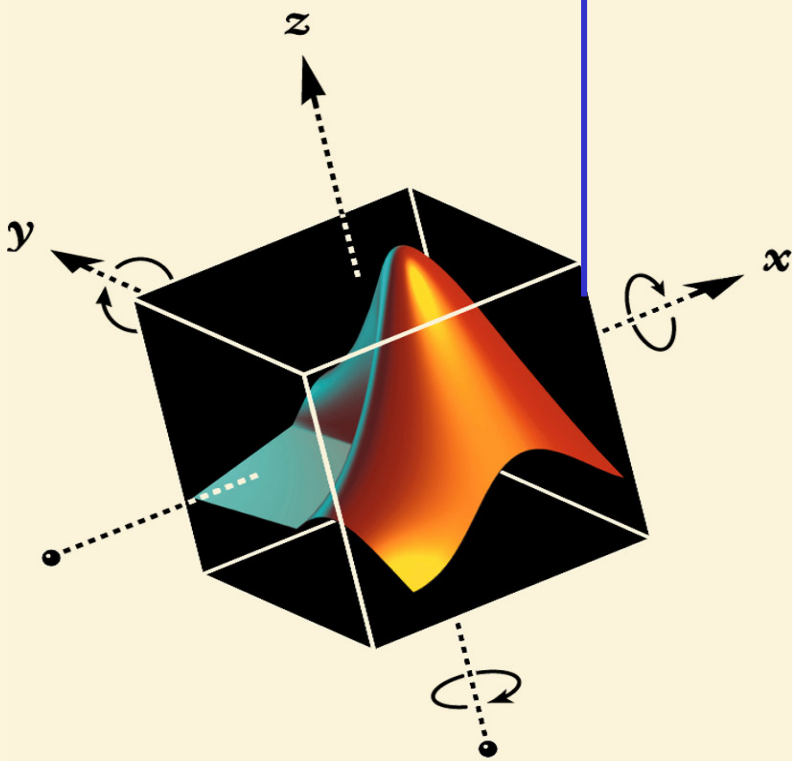
EC 321 Spectral Analysis

EC421 Random Signals and Noise

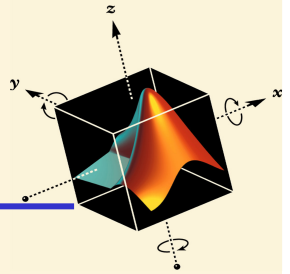
Wesam Gamal Eldin

Maha Abdel-bary

Mohamed Essam Khedr

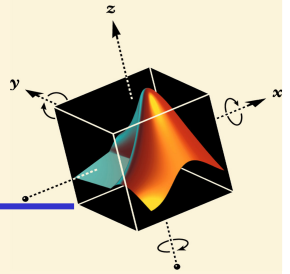


Desktop Tools (Matlab)



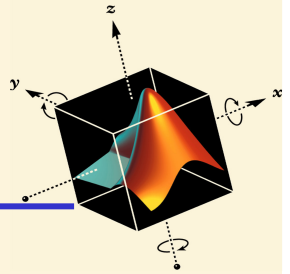
- Command Window
  - type commands
- Workspace
  - view program variables
  - `clear` to clear
  - double click on a variable to see it in the Array Editor
- Command History
  - view past commands
  - save a whole session using `diary`
- Launch Pad
  - access tools, demos and documentation

Matlab Files (.m)



- Use predefined functions or write your own functions
- Reside on the `current directory` or the `search path`
  - add with File/Set Path
- Use the Editor/Debugger to edit, run

Matrices



- a vector `x = [1 2 5 1]`

```
x =
     1    2    5    1
```

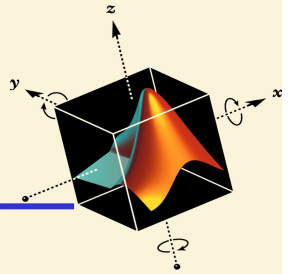
- a matrix `x = [1 2 3; 5 1 4; 3 2 -1]`

```
x =
     1     2     3
     5     1     4
     3     2    -1
```

- transpose `y = x.'`

```
y =
     1
     2
     5
     1
```

Matrices



- x(i,j) subscription

```
y=x(2,3)
y =
     4
```

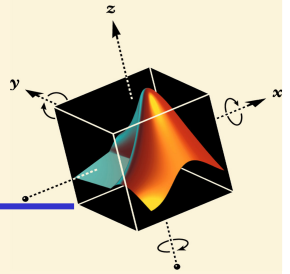
- whole row

```
y=x(3,:)
y =
     3     2    -1
```

- whole column

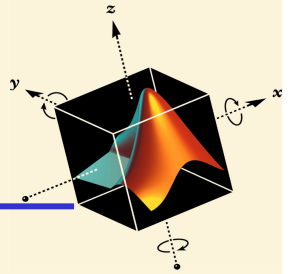
```
y=x(:,2)
y =
     2
     1
     2
```

Operators (arithmetic)



+	addition		
-	subtraction		
*	multiplication	.*	element-by-element mult
/	division	./	element-by-element div
^	power	.^	element-by-element power
‘	complex conjugate transpose	.‘	transpose

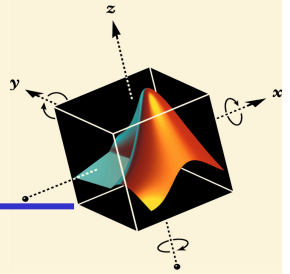
Operators (relational, logical)



==	equal
~=	not equal
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal
&	AND
|	OR
~	NOT

pi	3.14159265…
j	imaginary unit, $\sqrt{-1}$
i	same as j

Generating Vectors from functions



- zeros(M,N) MxN matrix of zeros

```
x = zeros(1,3)
x =
   0     0     0
```

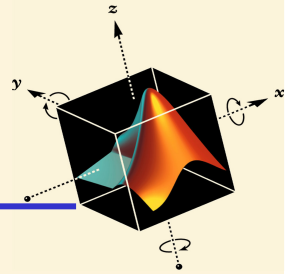
- ones(M,N) MxN matrix of ones

```
x = ones(1,3)
x =
   1     1     1
```

- rand(M,N) MxN matrix of uniformly distributed random numbers on (0,1)

```
x = rand(1,3)
x =
    0.9501    0.2311    0.6068
```

Operators



[ ] concatenation

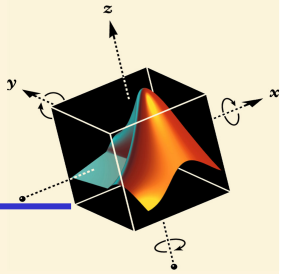
```
x = [ zeros(1,3) ones(1,2) ]
x =
     0   0   0   1   1
```

( ) subscription

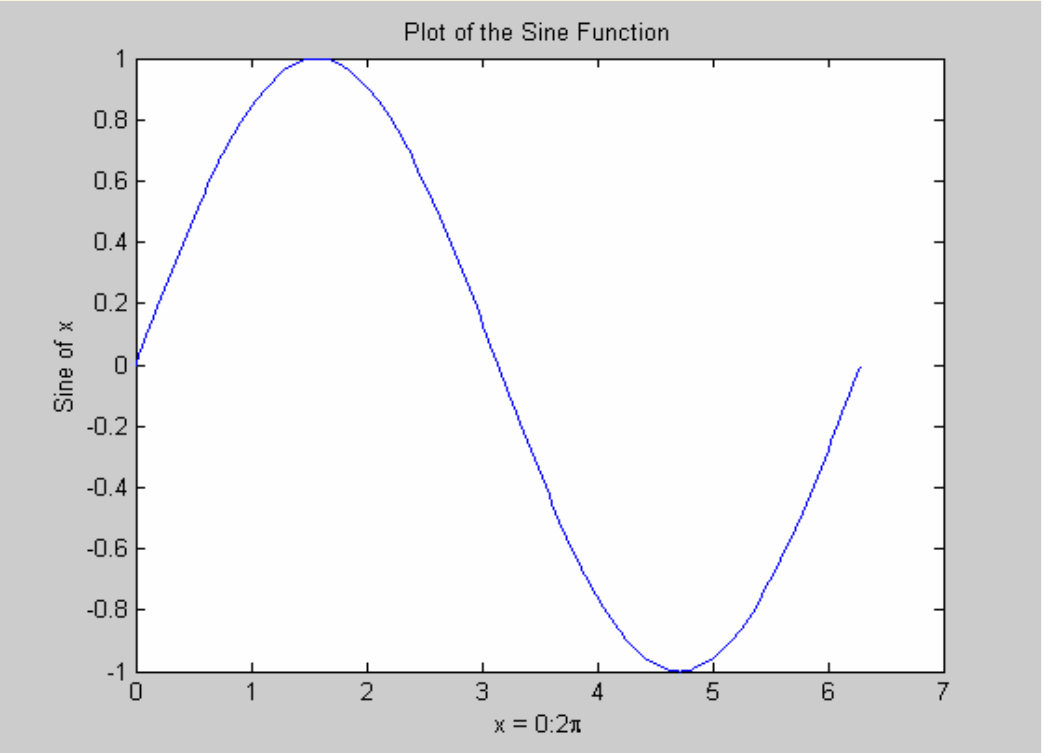
```
x = [ 1 3 5 7 9]
x =
     1   3   5   7   9

y = x(2)
y =
     3
y = x(2:4)
y =
     3   5   7
```

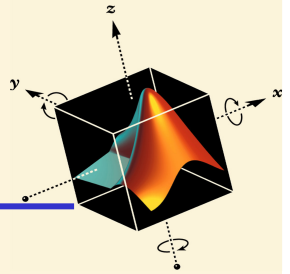
Matlab Graphics



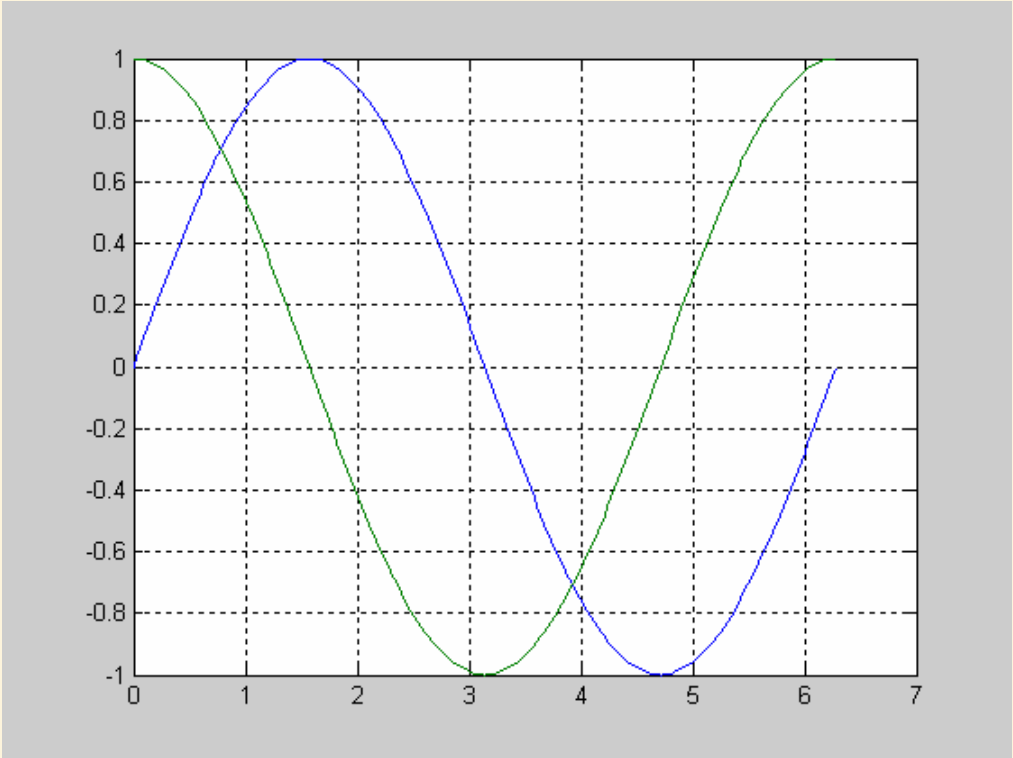

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
xlabel('x = 0:2\pi')
ylabel('Sine of x')
title('Plot of the
   Sine Function')
```



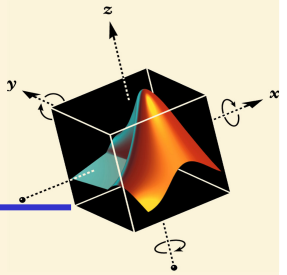
Multiple Graphs



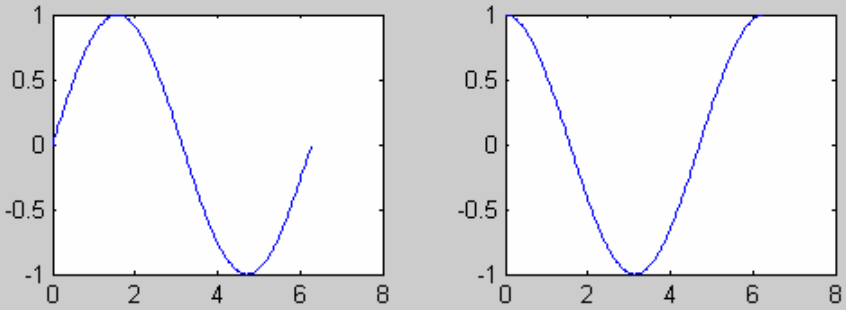

```
t = 0:pi/100:2*pi;
y1=sin(t);
y2=sin(t+pi/2);
plot(t,y1,t,y2)
grid on
```



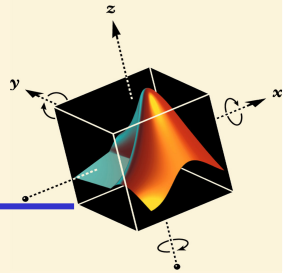
Multiple Plots




```
t = 0:pi/100:2*pi;
y1=sin(t);
y2=sin(t+pi/2);
subplot(2,2,1)
plot(t,y1)
subplot(2,2,2)
plot(t,y2)
```

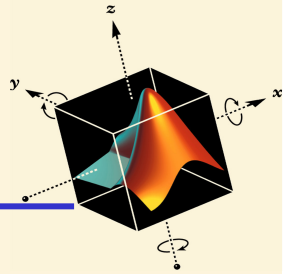


Graph Functions (summary)



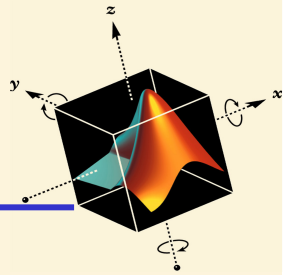
- plot linear plot
- stem discrete plot
- grid add grid lines
- xlabel add X-axis label
- ylabel add Y-axis label
- title add graph title
- subplot divide figure window
- figure create new figure window
- pause wait for user response

Math Functions



- Elementary functions
- (sin,
- cos,
- sqrt,
- abs,
- exp,
- log10,
- round)
  - type `help elfun`

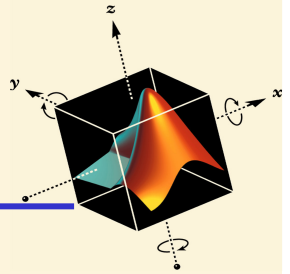
Functions




```
function f=myfunction(x,y)
   f=x+y;
```

- save it in myfunction.m
- call it with `y=myfunction(x,y)`

Flow Control

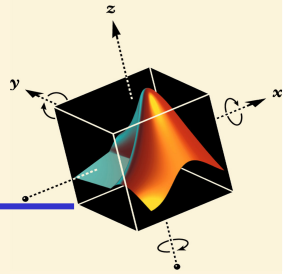


- `if` statement
- `for` loops
- `continue` statement
- `break` statement

```
if A > B
   'greater'
elseif A < B
   'less'
else
   'equal'
end
```

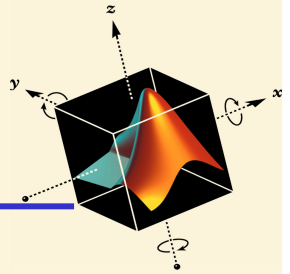
```
for x = 1:10
   r(x) = x;
end
```

Miscellaneous



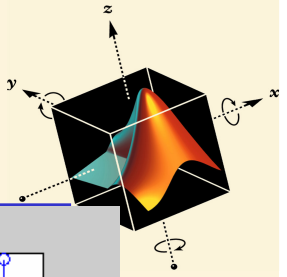
- Suppressing Output
  - `x = [1 2 5 1];`

Getting Help



- Using the Help Browser (.html, .pdf)
  - View getstart.pdf, graphg.pdf, using_ml.pdf
- Type
  - `help`
  - `help` *`function`*, e.g. `help plot`
- Running demos
  - type `demos`
  - type `help demos`

Random Numbers




```
x=rand(100,1);
stem(x);
```