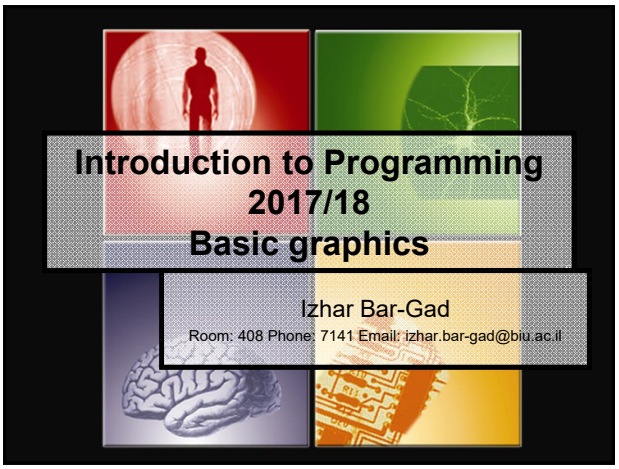
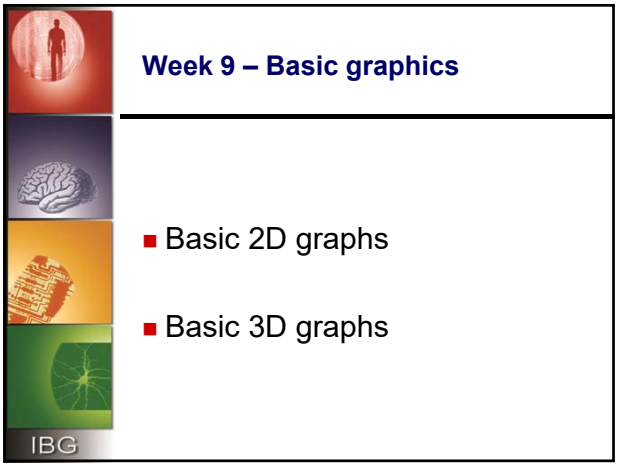
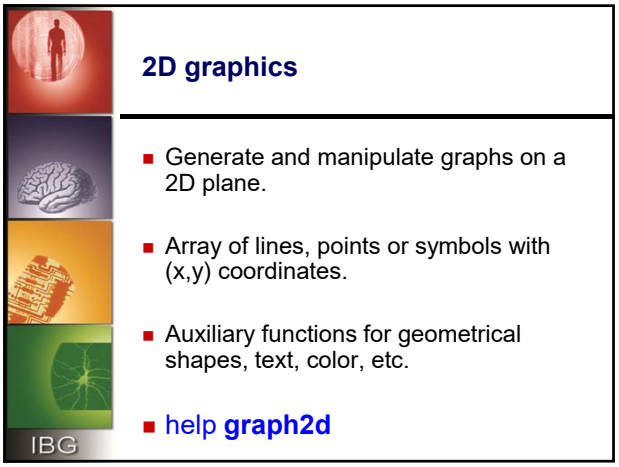

Introduction to Programming
2017/18
Basic graphics
Izhar Bar-Gad
Room: 408 Phone: 7141 Email: izhar.bar-gad@biu.ac.il

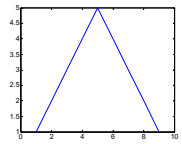

Week 9 – Basic graphics
Basic 2D graphs
Basic 3D graphs
IBG


2D graphics
Generate and manipulate graphs on a 2D plane.
Array of lines, points or symbols with (x,y) coordinates.
Auxiliary functions for geometrical shapes, text, color, etc.
help graph2d
IBG

Simple 2D graphics – Y values

- Plot points or lines within a two dimensional space
- Syntax: plot(*yVec)*
- Example

```
a=[1 2 3 4 5 4 3 2 1];
plot(a);
```



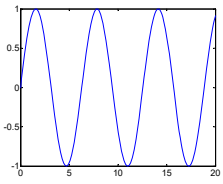
- *xVec* is assumed to be an ascending array of the same length as *yVec.*

IBG

Simple 2D graphics – X values

- Y values versus X values
  - Pairs of points: (X1, Y1) (X2, Y2) ...
  - X and Y are vectors.
- Syntax: *plot(xVec, yVec)*
- Example

```
a=[0:.1:20];
b = sin(a);
plot(a,b)
```

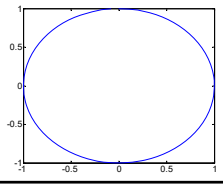


IBG

Simple 2D graphics – X,Y values

- X values might be repetitive. We are **not** plotting mathematical functions.
- Example

```
a = [0:.1:2*pi];
b = sin(a);
c = cos(a);
plot(b, c);
```



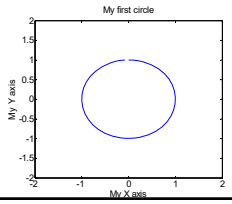
IBG

Nicer figures…

- Changing the axes
  Syntax: *axis([xMin xMax yMin yMax])*
- Adding labels
  Syntax: *xlabel(xString) ylabel(yString)*
- Adding a title
  Syntax: *title(tString)*

- Example

```
axis([-2 2 -2 2]);
xlabel('My X axis');
ylabel('My Y axis');
title('My first circle');
```

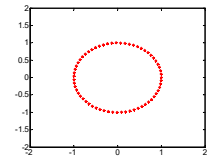


Colors, shapes and patterns

- The color, shape & pattern of the lines may be different from the default blue solid line.
- Set a third parameter:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

- Example:
  plot(cos(x),sin(x),'r*');



Multiple plots - addition

- Plotting multiple times → overriding previous plots or adding the plots.
- *hold on* – next plots will be added and will not override the existing one, adding plots mode.
- *hold off* – next plots will override the existing ones, override plots mode.
- The **default** behavior of MATLAB is override.

Multiple figures

- Opening the first figure is performed automatically for the first graph.
- Opening additional figures is performed by a *figure* command.

Additional 2D graphs

- Discrete data
  - *bar , stairs, …*
- Special graphs
  - *scatter , pie , …*
- help **specgraph**

Multiple sub-figures on one figure

- Using *subplot,* it is possible to plot multiple sub-figures on a single figure.

Syntax: *subplot(m,n,p)*

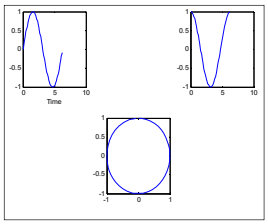
m - number of lines

n – number of columns

p – position → *(line-1)*n+column*

Sub-figures example

```
a=[0:.1:2*pi];
b=sin(a); c=cos(a);
subplot(2,3,1);
plot(a,b);
xlabel('Time');
subplot(2,3,3);
plot(a,c);
subplot(2,3,5);
plot(b,c);
```



IBG

3D graphics

- Display 3 dimensional data.
- Two main subgroups:
  - 2D projection
  - Matrix color encoding
- help **graph3d**

IBG

3D graphics – 2D projection

- Project the 3D data on a 2D plane
- Enables changing the projection.
- Many functions including:
  mesh, surf, fill3, plot3 …
- Two subfamilies:
  - (x,y,z) vectors
  - Matrix projection

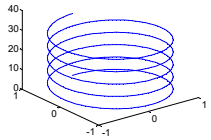
IBG

3D plots – (x,y,z) vectors

- Similar to the 2D plot with an additional *z* parameter – *plot3, fill3*
- Receives three vectors of (x,y,z) coordinates.

- Example:

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t);
```



IBG

An example matrix

```
>> a=[0 1 2 3 4 3 2 1 0]

a =

    0    1    2    3    4    3    2    1    0

>> b=a'*a

b =

    0    0    0    0    0    0    0    0    0
    0    1    2    3    4    3    2    1    0
    0    2    4    6    8    6    4    2    0
    0    3    6    9   12    9    6    3    0
    0    4    8   12   16   12    8    4    0
    0    3    6    9   12    9    6    3    0
    0    2    4    6    8    6    4    2    0
    0    1    2    3    4    3    2    1    0
    0    0    0    0    0    0    0    0    0
```

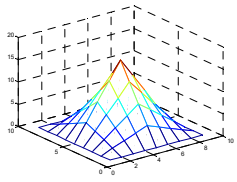
IBG

3D graphics –matrix projection

- Receives two vectors of *(x,y)* coordinates and a z matrix for all *(x,y)* combinations

- Example

```
a=[0 1 2 3 4 3 2 1 0];
b=a'*a;
mesh(b);
```



IBG

3D graphics – color encoding

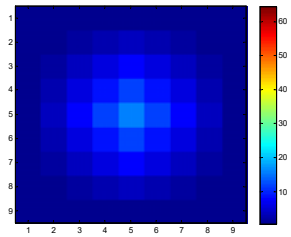
- Use a 2D X,Y plane with the Z axis encoded via a color scale.
- Scaling issues: *image* vs. *imagesc,*
- Scale of Z axis: *colorbar*
- Map of Z axis: *colormap*

IBG

3D graphics – image

- Example:

```
a=[0 1 2 3 4 3 2 1 0];
b=a'*a;
image(b)
colorbar
```

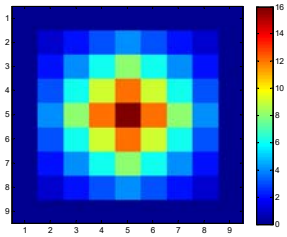


IBG

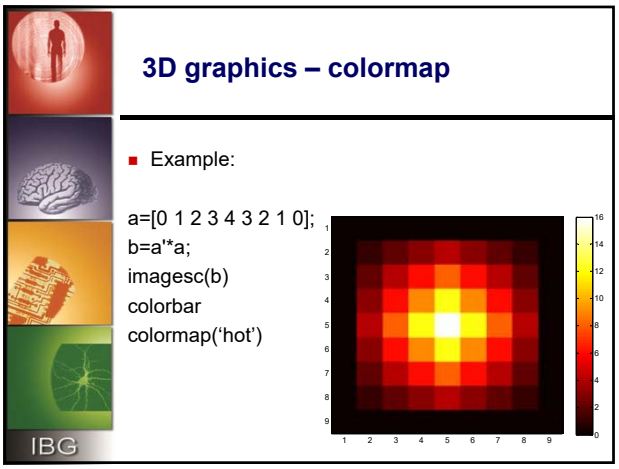
3D graphics – imagesc

- Example:

```
>> a=[0 1 2 3 4 3 2 1 0];
>> b=a'*a;
>> imagesc(b)
>> colorbar
```



IBG


3D graphics – colormap
Example:
a=[0 1 2 3 4 3 2 1 0];
b=a'*a;
imagesc(b)
colorbar
colormap('hot')
IBG