Advanced Plotting

Working with figures

- Each time you call figure(), a new figure is created.
- h = figure('PropertyName', propertyvalue)
- A "handle" is returned, so you can refer to that figure later or bring it back to focus.

```
>> h=figure
>> figure
>> figure(5)
>> figure(h)
>> gcf
```


Good practice

- title(' the figure title ')
- xlabel(' x axis label, with units ')
- ylabel(' y axis label, with units ')

```
>> plot(1990:5:2010, [5.2,5.7,6.1,6.5,7]);
>> title('World population growth from 1990 to 2010');
>> xlabel('population (in billions)');
>> ylabel('year');
```

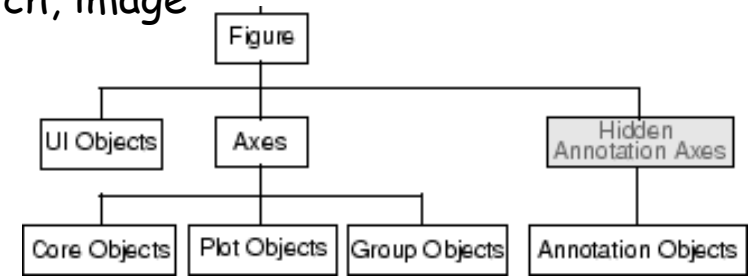

Working with figures

- You can use get/set function to read/change figure properties.

```
>> h=figure
>> get(h)
>> get(h, 'Position')
>> set(h, 'Position',[0 0 500 500])
>> set(h, 'Pointer','hand')
```

Working with figures

- Matlab graphics objects are organized in a hierarchy.
- figure is the main window.
- A figure can contain multiple axes.
- Core objects are: line, text, rectangle, patch, image



ha = subplot(R,C, currentindex)

- subplot creates a matrix of plots in the current figure window.
- The currentindex specifies the "current subplot" where drawing would take place.
- returns a "handle" to the current axes.

Exercise (subplots2x3.m)

- Create a 2x3 subplots, where the top row of subplots show sin(x), cos(x), and tan(x) and the bottom row of subplots show sqrt(x), log(x), and exp(x). Use x=0:0.1:2*pi.

Multiple plots in the same figure

- Each time you plot() data, the old axes is wiped off.
- Use clf( ) and cla( ) to clear the current figure/axes.
- Use: hold('on') and hold('off')
- Use different colors/linetypes for each "series".
- Use legend() to annotate different "series" of data you are plotting.

```
>> plot(x,sin(x),'x');
>> hold on;
>> plot(x,cos(x),'o');
>> legend( {'sine', 'cosine'} );
```

Plot types (plottypes2x2.m)

- Exercise: Create a 2x2 subplots for the following population vs. years data, using bar(), barh(), area(), and stem().

```
>> years=2007:2011;
>> pops = [0.9 1.4 1.7 1.3 1.8];
```


Bar graphs

- Bar graphs (bar(), barh() functions) can work with multiple groups of data given in multiple rows of a matrix

```
>> groupages = [83 88; 90 95];
>> bar(groupages);  %,'stack'
>> ylabel('life expectancy (years)');
>> set(gca,'XTickLabel',{'male','female'});
>> legend( {'smoker','non-smoker'});
```


Histograms

```
>> grades=[10 8 5 10 10 6 9 7 8 10 1 8];
>> hist(grades)
>> xlabel('Grade')
>> ylabel('count')
>> title('Homework Grades')
```


Pie Charts

- pie(vec) function shows a pie chart of the vector vec with percentage of each element compared to the sum of vec.

```
>> pie([11 14 8 3 1])

>> pie([11 14 8 3 1], {'A','B','C','D','E'} )
```

comet()

- comet() function shows an animation of the drawing the plot, moving from point to point.

```
>> x = -2*pi : 1/100 : 2*pi;
>> y = sin(x);
>> comet(x,y)
```


Animation

- Two strategies:
  - Use the pause() function between different plots to pause for a brief time.
  - Use a timer (tic) and only drawnow if time since last tic is greater than a period (e.g., 0.1 sec)
  - Take snapshots using getframe(), and use movie() to show all the frames as a movie.

Exercise: sinanimate()

- Plot sin(x) for x=-2*pi:.01:2*pi animating the plotting point by point for each value of x.
- Solve the problem using pause()
- Solve the problem using tic/toc
- Solve the problem again using getframe() and movie()

3D plots

- Many of the 2D plotting functions have 3D versions, with a "3" at the end of their names.
  - plot3, scatter3, bar3, bar3h, comet3, stem3

```
>> m = rand(10,3);
>> plot3( m(:,1), m(:,2), m(:,3) );

>> t = 0:.001:12*pi;
>> comet3(cos(t), sin(t), t)
```

3D plots

```
>> m = spiral(5)
>> bar3 ( m )
```


mesh and surface plots

```
>> [x,y,z] = sphere( 6 );
>> mesh(x,y,z)
```

```
>> surf(x,y,z)
```


Working with primitive (core) graphics objects

- Core Graphics Objects: line, text, rectangle, patch, image

```
>> x=0:.1:6;
>> h = line (x, sin(x), 'LineWidth', 6 );
>> set( h, 'Color', [1 0 0] )
>> ht = text( 2, 0, 'sin(\pi) \rightarrow');
>> set(ht, 'EdgeColor',[0 0 1 ]);
```