W1005
Intro to CS and Programming in MATLAB

Plotting & Visualization

Fall 2014

Instructor: Ilia Vovsha

http://www.cs.columbia.edu/~vovsha/w1005

Outline

- Plots (2D)
- Plot properties
- Figures
- Plots (3D)

2D Plots

- The general function we will use is plot()
  - The function automatically generates a figure for you
- There are many options available:
  - Can specify the properties of the plot in detail
  - Can have multiple curves on the same plot
  - Can have multiple plots in one figure
  - More interesting plots are possible with bar(), pie()

2D Plots (options)

- Option 1: plot(X, Y)
  - Assuming X & Y are vectors, plots Y vs. X
  - A solid curve connecting the points
  - Y & X must have the same length
- Option 2: plot(Y)
  - Similar to option 1, but plots Y vs. 'index'
- Option 3: plot(X, Y, S)
  - 'S' is an optional parameter
  - A character string that controls plot appearance

2D Plots (options)

- Option 3: plot(X, Y, S)
  - Example: S = 'rs:' → plots a red, dotted, line with a square at each data point on the curve
  - 'help plot' for all the string options
- Option 4: plot(X, Y, S, <param>, <val>)
  - Additional plot properties can be specified with different parameters and values
  - Example: plot(X,Y, S, 'LineWidth', 2) set the line-width to size 2
  - Type 'doc plot' for a list of these properties

2D Plots (more options)

- More options:
  - Can set as many parameters as you wish (just call the function with more parameter-value pairs)
  - Can combine plots: plot(X1, Y1, X2, Y2)
  - Can 'duplicate' a plot: plot(X, Y, S1, X, Y, S2)
  - Plot combination is fine, but not the most elegant approach

Figures (purpose)

- Problem:
  - plot(x,y); plot(x,z); replaces 1st plot with 2nd
- Solution: 'figure' command
  - plot(x,y); figure; plot(x,z);
  - figure(1); figure with handle #1
- Close figures:
  - Specific figure: close 1
  - All figures: close all
- MATLAB stores a handle to each figure

Figures (subplots)

- Multiple plots:
  - plot(x,y); hold on; plot(x,z); hold off;
- Multiple plots, same figure:
  - subplot() command
  - figure(1); subplot(2,2,1);

Plots (appearance)

- Many options, can modify plots using the GUI
- Commands: title(), xlabel(), ylabel(), axis(), legend()
- Example:

```
1. figure(1);
2. title('test');
3. xlabel('quantity'); ylabel('price');
4. axis([1 5 1 10]);          % AXIS([XMIN XMAX YMIN YMAX])
5. grid on;                   % Show grid lines
6. xlim([1 3]);               % Change x-axis limits
```

Figures (control)

- You have created multiple figures. Which one is the 'current figure'?
- Answer: last figure you clicked on
- Better answer: use gcf() to get the handle
- Use figure(#) to make '#' the current handle
- Commands: gcf(), gca(), clf()
- Set 'object' properties using the set() command:
  - set(gca, 'XTick',[1 2 3]) % To set 'ticks' on the x-axis
- Get 'object' properties using the get() command

Figures (saving & loading)

- MATLAB has a special figure format: '.fig'
- Use savefig(h,'name.fig') to save figure 'h' (handle)
- Use openfig('name.fig') to open a saved figure
- Saving figures: use print command
  - **General Form**
  - **print** -d*format* filename
  - **Example**
  - **print –depsc 'figure.eps'**
- 'eps' is a format that stores your image in a vectorized way, which avoids quality loss after rescaling. It's particularly useful when used within LaTeX

Exercise (in class)

- Simple plot
- Write a function that has one input parameter M, a matrix with 3 columns. Columns {1,2} of M are the {x,y} coordinates of points in the plane. Column 3 is the class to which the point belongs. The values of column 3 are one of two unknown integers
- Your function should plot all points, points in one class should be dots in red, points in the 2^{nd} class should be squares in green. Adjust your axes properly so that the furthest points are not on the edge

3D Line Plots

- Very similar to 2D plotting
- Keyword: plot3()
- Example:

```
1. X = 0:0.1:1;  Y = X;  Z = X.^2+Y.^2;
2. plot3(X,Y,Z);
3. xlabel('locX'); ylabel('locY'); zlabel('power');
4. xlim([0 1]);    zlim([0 40]);
```

3D Plots (surface)

- 'plot' functions produce line plots. Suppose we wish to display a 3D surface
- Keywords: meshgrid(), mesh(), surf()
- The command meshgrid(X,Y) creates a grid (domain) with all combinations of {X,Y} elements. mesh(X,Y,Z) / surf() are then used to plot the surface Z = f(X,Y) over the grid

3D Surface (example)

- Example:

```
1. X = 1:4;  Y = 1:3;
2. [Xg, Yg] = meshgrid(X,Y);          % produce a grid
3. Z = Xg.^2 + Yg.^2;                 % Z = f(X,Y)
4. mesh(X,Y,Z);                       % plot
5. surf(X,Y,Z);                       % plot
```

- 'mesh' plots wireframe. 'surf' plots with shading.
- Can pass a 4th parameter specifying color e.g. "surf(X,Y,Z,C)". Otherwise, color is proportional to mesh height.

3D Surface – Example

- surf vs. mesh