TECHNISCHE
UNIVERSITÄT
DRESDEN

Introduction to Matlab

Plots

Dario Cuevas and Vahid Rahmati

DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

Recap: Plot command

- plot(x,y), where x and y are vectors of the same size.
    - x = 1:0.1:10; y = sin(x); plot(x, y, '-.b*') % color + marker
- List of markers and colors: help plot
- You can put more than one function in a plot:
    - plot(x, sin(x), 'g', x, cos(x), 'red')
- Some useful properties:
    - Title, xlabel/ylabel, legend, axis([x1, x2, y1, y2]), LineWidth
    - List of the properties that canchange in plots: get(gca)
    - get(gca) provides a list of all the features you can change in plots
    - plot(x, sin(x), 'g', x, cos(x), 'red', 'LineWidth',2)
- Holding the plots:
    - x1 = -pi:0.1:pi; y1 = cos(x1); x2 = -pi/2:0.05:pi/2; y2=sin(x2);
      plot(x1,y1, '-.ored',x2,y2, '-*green');
      hold on
      plot(x1/3,x1/3,'sblack')
      hold off

Useful commands

- figure create a window for plotting:
  - figure(i) creates the ith figure window, where i=1, 2, 3, …
  - Hint: if the ith figure has been already created, then figure(i), instead of creating a new ith figure, will only select the existed ith figure window as the current one.
  - figure(1); x = 1:0.1:10; y = sin(x); plot(x, y,'b','LineWidth',2)
  - figure(2); x = 1:0.1:10; y = cos(x); plot(x, y, 'ob')
  - figure(1); title(' No new window! ')
- clf clears the current window figure
- close closes the current figure window
  - close(i) closes the ith figure window
  - close all closes all the figure windows

Plot-Hold commands

- To drew multiple plots in one plot:
  1) A combination of plot and hold commands can be used in order to add the new plot to the current one:

```
plot(x1,y1); % first plot
hold on ;  % the first plot will be retained
plot(x2,y2);  % the new plot is added on top of the first plot
plot(x3,y3); % the third plot is added on top of both previous plots
hold off; % the retaining of the previous plots is stopped
plot(x4,y5); % the previous plots were erased, an only this plot is displayed
```

- Hint 1: the vectors x1, x2, x3, and x4 can have different number of elements, and ranges.
- Hint 2: the final range of x and y axes account for their presented smallest and biggest values among the plots.

Plot simoltaneously

- To draw multiple plots in one figure:
  2) The plots can be drawn simultaneously :

```
plot(x1,y1,x2,y2); % draws the plot(x2,y2) on top of the plot(x1,y1)
```

- Hint 3: the hints 1 & 2 are also applied here.
- Example1:

  -
    ```
    x1 = -pi:0.1:pi; y1 = cos(x1); x2 = -pi/2:0.05:pi/2; y2=sin(x2);
    plot(x1,y1,'-.ored',x2,y2, '-*green')
    plot(x2,y2,'-.ored',x1,y1, '-*green')
    plot(x1, cos(x1), ' -.ored',x2,y2, ' -*green')
    ```

  -
    ```
    x1 = -pi:0.1:pi; y1 = cos(x1); x2 = -pi/2:0.05:pi/2; y2=sin(x2);
    clf;
    plot(x1,y1, '-.ored',x2,y2, '-*green');
    hold on
    plot(x1/3,x1/3,'sblack‘)
    ```

Exercises

1. The vectors x=[0:0.2:10] and z=[-5:0.1:5] are given. Plot the following functions on top of each other in one plot: y1 = exp(-x), y2= (z^2)/10, y3=sin(x). The second function should be drawn in red, and third in green. Set the labels for x and y axes as “x label [unit]” and “y label [unit]”, respectively. Activate the gird lines. Set the title as “my first plot”.
2. Using for loop, create the vectors of random numbers:
   x = Mean+ StandardDeviation*randn(1,100)
   for mean={-4,-2, 0, 2, 4} and StandardDeviation = 0.5, and draw plot(1:100, x) for each vector x. The resulted curves should be displayed in one plot. Hint: you can use hold commands. Using the command figure create a new figure, and plot there a white noise (zero mean) for samples 1:500 (instead of 1:100) and StandardDeviation of 1. Increase the line width to 2.

Subplot command

- To draw multiple plots in one figure
  that is by using subplot(m, n, plot number) command, a figure can be divided into m rows and n column, thus m*n plots will be available.
- Hint 1: the plots are numbered row-wise.
- Example:

  -
    ```
    x1 = -pi:0.1:pi; y1 = cos(x1);
    subplot(3,2,1);
    plot(x1,y1,'-.ored‘);
    title('cos function');
    x2 = -pi/2:0.05:pi/2; y2=sin(x2);
    subplot(3,2,2);
    plot(x2,y2,'-.ored',x1,y1, '-*green')
    ylabel('something! ')
    subplot(3,2,5);
    imagesc(magic(5));
    title('MAGIC MATRIX');
    ```

Imagesc command

- imagesc(MAT) can be used to plot the matrix MAT:
  - It uses the full range of the figure’s colormap to show each element of the matrix by a specific color.
  - imagesc(magic(5))
  - colorbar shows the range of matrix values, and the colors by which they were coded.
  - You can change the colormap to, for example, summer, winter, gray, … :
  - colormap gray % set the colormap to gray
  - colormap default % set back to the default colormap
  - imagesc(MAT, clim) first normalizes the matrix to the range defined by clim, and then shows the color-coded matrix
  - y = [rand(5,5),zeros(5,5)]; y(5,10)=100;
    imagesc(y); colorbar
    now compare it to:
    figure;
    imagesc(y, [0 1]); colorbar

Bar plot

- bar(Y) draws a bar for each value in the array Y:
  - Y = randn(1,10);
  - bar(Y)
- You can also specify the location name of each bar:
  - Y = 1:5; years=[2011:2015]
  - bar(years, Y)
- If Y is a matrix, then bar(Y) considers each row as a group, and creates an individual bar graph for each of them:
  - Y = [1:5; randn(1,5)]; X=[2011:2015; -4:0];
  - bar(X, Y)
- To draw the horizontal bar graphs, you just need to use barh(Y), instead of bar(Y).
  - Y = [1:5; randn(1,5)]; X=[2011:2015; -4:0];
  - bar(X, Y)

Histogram

- hist(Y) draws the histogram of vector Y: MATLAB by default divides the space between the max and min of Y element values, and then in each bin it stores the total number of Y elements of Y which have values distributed within that bin.
- V = hist(Y) uses vector V to store the total number of the distributed elements whithin the bins:
  - Y = Y = [1 2 0 8 -1 10 3 15 18 20]; hist(Y)
  - V = hist(Y)

INCOMPLETE