EENG 212/ EENG 221
Fall 2014

Tutorial 3
2D and 3D Plots

1-Array Plotting Capabilities

1.1 Basic Plot Command

Plot command can be used to produce an xy plot of vector x against y, as linear plots with the x and y axes divided into equally spaced intervals. Polar plots use a log scale (base 10) to display a wide range of values.

Plot(x,y) generates a linear ploy of a and y (vertical axis)
Semilogx(x,y) generates a plot of the values of x and y using a log scale for x and linear scale for y.
loglog(x,y) generates a plot with log scales for x and y

Line styles of plot can be controlled. This argument is a character string consisting of for example

___ solid line
: dotted line
-. dash dot line
---- dashed line

Some Plotting commands: the most important commands for basic 2-dimensional plotting are illustrated in the following examples:

```
>>x=0:0.1:10;       % x ranges from 0 to 10 in steps of 0.1
>>y=sin(x);         % y contains the sin of each value of x
>>figure            % creates an empty figure window
>>plot(y)           % plots all of the y values
>>plot(x,y)         % plots y against x (replacing the old plot on the same
                    %figure, now showing the units of x on the x-axis)
>> close            % closes the figure window
>> disp             % diplay command
```

A simple example:

```
>>x=linspace(0, 2*pi ,100);
>>plot(x, sin(x))              % or   plot (x, sin(x), '*')
>>xlabel('x'),ylabel('sin(x)', 'fontsize', 12)
>>title('my plot)        % or   legend(('my plot')
```

Zooming: The command "zoom [on/off]" command is useful for zooming of a plot with the mousse. Type "zoom" or "zoom on" , drag cursor of the mousse on a plot and click the left button of the mousse. Likewise, you can return to the original state by double clicking the left or right button of mousse. The off state turns off this feature.

1.2 Multiple curves

Using Subplot to Layout Multiple Graphs:

A few plots can be placed side by side (not overlay) by using the SUBPLOT command. The format is

subplot (m, n, p)

It divides the graphic window into m by n sub-windows and puts the plot generated by the next plotting command into the p_{th} sub-window where the sub-windows are counted row-wise.

Example:

```
>>subplot(2,2,3), plot (x,y)    % divides the graphics window into 4 sub-windows and plot y
>>                              % versus x in the 3rd sub-window.
```

Example:

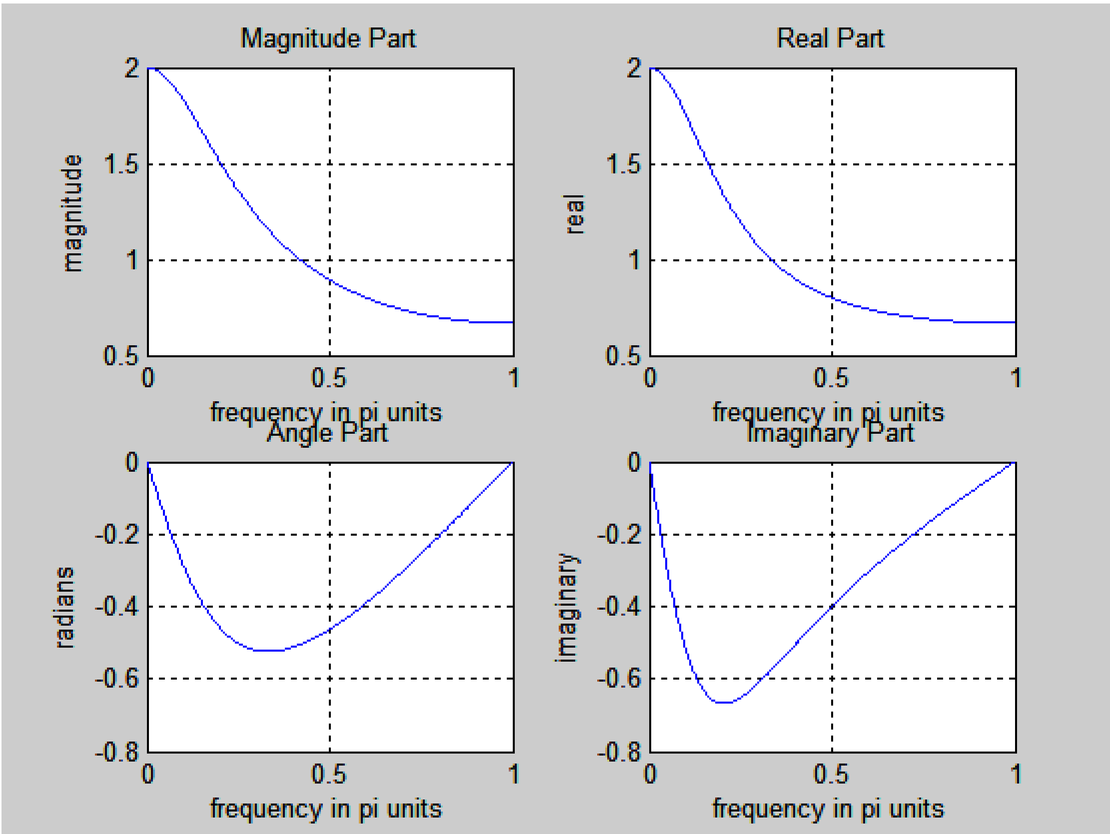
Consider the function $x(e^{jw}) = (e^{jw})/(e^{jw}-0.5)$

Evaluate at 501 equispaced points between [0,pi] and plot Magnitude, Angle, Real and Imaginary Parts on same window.

Solution:

```
>>% Layout of multiple graphs
>>w=[0:1:500]*pi/500;  % [0,pi] axis divided into 501 points
>>x=exp(j*w)./(exp(j*w)-0.5)
>>magX=abs(x); angX=angle(x);
>>realX=real(x); imagX=imag(x);
>>subplot(2,2,1); plot (w/pi,magX);grid   %frequency in pi units
>>xlabel('frequency in pi units'); ylabel('magnitude');
>>title('Magnitude Part')
>>subplot(2,2,3); plot (w/pi,angX);grid   %frequency in pi units
>>xlabel('frequency in pi units'); ylabel('radians');
>>title('Angle Part')
>>subplot(2,2,2); plot (w/pi,realX);grid   %frequency in pi units
>>xlabel('frequency in pi units'); ylabel('real');
>>title('Real Part')
>>subplot(2,2,4); plot (w/pi,imagX);grid   %frequency in pi units
>>xlabel('frequency in pi units'); ylabel('imaginary');
```

>>title('Imaginary Part')
This results in the following graphs.



Practical note: The graphics can be copied and pasted on a MS Word page. To do so, in the graphics window, select EDIT and then "copy figure" and then paste to a word document.

For the command **plot(x,y,w,z)** where x,y,w,z are vectors
y versus x and z versus w are plotted on the same graph
The command **plot(x,F)** where x is a vector and F a matrix , will produce a graph in which each column of F is a curve plotted against x.

To distinguish between plot on the graph, the ***legend*** command is used.
Legend ('string1','string2','string3', ….)
It places a box on the plot, with the curve type for first curve labeled 'string1'.

Pos: a position number can be added to the legend command values of pos.
0: automatic best placement
1: upper right-hand corner
2: upper left hand corner

3: lower left hand corner
4: lower right hand corner
-1: to the right of the plot

Example: consider f1=x^2 -3x +2 and f2= 2x62 + x -3

```
% generates plots of polynomials
x = 0; 0.1 : 5;
f(:,1) = x^1.^2 – 3*x^1 + 2;
f(:,2) = 2*x^1.^2 + x^1 – 3;
plot (x,f), title ('multiple function plot'), xlabel ('x'), grid, legend ('f1','f2')
```

To plot 2 curves with separate y-axis scaling and labeling, the plotyz
For previous example:

```
x= 0: 0.1 : 5;
f1 = x^1.^2 – 3*x^1 + 2;
f2 = 2*x^1.^2 + x^1 – 3;
plotyz (x,f1,x,f2), title ('multiple plot')
xlabel ('x'), grid
```

Note: to create multiple windows to display each plot >> figure (n)
To close current window

```
Close
Close(n)                ;the window only
Close(all)              ;all window close
Clf    will erase the content of the current window
```

Sinefunction.n

```
x= 0 ; .1 ; 20;                  % create vector
y = exp(0.1 * x).* sin(x);               % calculate y
Plot (x,y)
xlabel ('time (t) in seconds')
ylabel ('the response amplitude in mm')
title ('a simple 2-d plot')
```

Now let's try this example
logplot.m

```
%generates a pair of plots with proper scaling
clf                           %clear graph window
v1 = logspace (10,0,20);                 %generate a logarithmic vector
v2 = logspace (2,-2,20);                 %generate a logarithmic vector
v = [v1 v2];                             %concentrate the 2 vectors
subplot (1,2,1), plot (v)                %split screen and plot vector
subplot (1,2,2), plot(log(v))         %plot log of the vector (base e)
```

%here we can also use plot instead
%of subplot

The output window should show vector v plotted as a function of its . Concentration of vectors v1 and v2 means the first 20 elements of v are the elements of v1, the last 20 elements are those of v2. On the right-hand side, the log of v is plotted. The values of v2 are not visible on the left-hand plot while on the right-hand side plot, both values of v1 and v2 are plotted. The command subplot (121) and subplot (122) divide the graphics window into 2 equal halves, side by side. In general, subplot (abc) or plot (abc) where a, b, and c are integers will create on axb array; the argument c indicates the current graph. After you have used the command subplot, you may have to input the command subplot(111) in order for your next plot to use the full screen.

Specialized 2-d plots: some functions

Bar: creates a bar graph
Comet: makes an animated 2-d plot
Contour: makes contour plots
Fill: draws filled polygons of specialized color
Fplot: plots a function of a single variable
Hist: makes histograms
polar: plots curves in polar coordinates
stair : plots a graph
stem: plots a stem graph

2- Formatted output
2.1 using the fprint command

Use of the **fprintf** function gives more control than the **disp** function when we want to display the output. It is the same format as the one used by C language to print text and matrix values. The format is

fprintf(fid, format, matrices)

For format, the specifiers are %e, %f and %g, respectively exponential form, fixed point or decimal form, and either %e or %f (depending on which one is shorter).

The format string usually ends with \n , which means start new line, to show the end of the location of the printed line, the rest is printed on the next line; **fid** is an integer file opener that works with **open** command (can be of value is 1 for the screen or 2 in case of error).

Example:
>>fprint(**‘** The interest is **%f** dollars. **\n’ ,** interest)
>>% if the interest is 30, then the print out will list:
The interest is 30 dollars.

>>fprint(**‘**The price of a shirt in a luxury store **\n** is %f dollars **\n’**, 200)

the print out will reveal:
The price of shirt in a luxury store
is 200 dollars

let’s recapitulate with an example on temperature conversion

```
>>F= -40:5:100;
>>C=(F-32)*(5/9); % convert to temperature Celsius
>>T=[F ; C]'
>>fid=fopen('temperature.table', 'w') %open existing file or open new file temperature.table
>>fprintf(1, ' Temperature    Table\n'); % fid value is 1 for screen.
>>fprintf(1,'------------------------\n');
>>fprintf(1, 'Fahrenheit     Celsius \n');
>>fprintf( 1, '%4i         %8.2f    \n', T)    % i: integer format and  f: decimal format
>>fclose(fid)
```

2.2 Capturing output
The session in the workspace area can be captured and saved to a report (MS Word doc for example). To do so, use the **diary** command.
To start the recording session, assuming the name of your file is **report1,**
diary report1 % begin session
diary off % end session

3 - 3D plots

M provides extensive facilities for visualization of 3D data:
- space-curvings, surfaces, contour tracing, colors and shadings and even displaying images of common functions

plot3 (x,y,z, ‘style option’) plots a curve in 3d space with the specified line style. Can be done with xlabel, ylabel, title, text, grid,…,zlabel
comet3( ) makes animated 3d plots
fill3( ) draws filled 3d polygons
mesh( ) draws 3d mesh surfaces
contour3 ( ) makes 3d contour plots
meshc draw 3d mesh surfaces along with contour
surf creates 3d surface plots
surfc creates 3d surface plots along with contour
cylinder generates a cylinder
sphere generates a sphere
surfl creates a 3d surface plot with specified light source

Next, using the view function.

view (azimuth , elevation) where azimuth in degrees, specifies the horizontal rotation from the y-axis, measured positively 0 CCW, and
elevation means elevation from the xOy plan.

Viewer on the positive x-axis, looking straight on the yz plane. This produces a 2d projection of the object on the yz plane.

View (0,90)	view along the z axis from above
View (0,0)	view along the y axis
View (z)	is same as view (0,90)
View (3)	is same as view (-37.5, 30) 3d default view

EX:
```
t = linspace (0, 6*pi, 100);
z = cos t; y = sin t; z = t;
subplot (2,2,1)% 2x2 and plot in 1st window
plot3 (x,y,z), grid
xlabel( ‘cos(t)’), ylabel (‘sin(t)’), zlabel (‘t’)
title (‘circular helix’)
Subplot (2,2,2)
Plot3 (x,y,z), view (0,90)
xlabel ( ), ylabel ( ), zlabel ( )
Title (‘projection in the xy phase’)
```

Some mesh plots:
x = -3;3;
y = -1;3;
[x-grid, y-grid] = meshgrid (x,y)
%we have just generated the grid metrics

x-grid = -3 –2 –1 0 1 2 3
-3 –2 –1 0 1 2 3
-3 –2 –1 0 1 2 3
-3 –2 –1 0 1 2 3
-3 –2 –1 0 1 2 3

y-grid = -1 –1 –1 –1 –1 –1 -1
0 0 0 0 0 0 0
1 1 1 1 1 1 1
2 2 2 2 2 2 2
3 3 3 3 3 3 3

underlying grid:
x = f(x,y) = 1/x^2+y^2
y = 1./(x-grid^2 + y-grid^2);

Plotting z using the underlying grid use either mesh function or the surf function (both 3d plots)
Former: mesh(x-pts, y-pts, z) generates a mesh plot of the surface defined by the matrix z
Surf(x-pts, y-pts, z) generates a shaded mesh plot of the surface defined by the matrix z
% to above program
subplot (2,1,1) mesh (x-grid, y-grid, z)
title (‘mesh plot’), x label (‘x’), y label (‘y’)

contour plots:
use function contour (x,y,z,) generates a contour plot of the surface defined by matrix z. x and y are vectors

Finally here are some functions and their illustrative applications in 2D and 3 D plots

To do for next week:

Tutorial 3:

Due next week: Produce a Matlab Sript for all questions and include output.

a)
Use MATLAB to produce a single plot displaying the
graphs of the functions sin(kx)
across [0, 2π], for k = 1, ... , 5.
b)
Use MATLAB to print a table of values x, sin(x), and cos(x), for x = 0, π/6,
2π/6, ... , 2π. Label the columns of your table. (use fprintf command)
c)
Plot each of the functions below over the range spe cified. Produce 4 plots on the same page using the subplot command.
a.
$f(x) = |x-1|$ for $-3 \le x \le 3$. (Use abs in MATLAB)
b.
f(x) = sqrt(abs(x)) for $-4 \le x \le 4$.
c.
f(x) = exp((-x)$^{\wedge 2}$) for $-4 \le x \le 4$.
d. $f(x) = 1/(10x^2 + 1)$ for -2 ≤ x 2