Worksheet 5. Matlab Graphics

Two dimesional graphics

Simple plots can be made like this

```
x=[1.5 2.2 3.1 4.6 5.7 6.3 9.4];
y=[2.3 3.9 4.3 7.2 4.5 6.1 1.1];
plot(x,y)
```

plot can take an additional string argument controlling any or all of the color, point marker or line style. E.g.

```
plot(x,y,’r>-.’)
```

Code	Colour
r	red
g	green
b	blue
c	cyan
m	magenta
y	yellow
k	black
w	white

Code	Marker
o	Circle
*	Asterisk
.	Point
+	Plus
x	Cross
s	Square
d	Diamond
ˆ	Upward triangle
v	Downward triangle
>	Right triangle
<	Left triangle
p	Five poined star
h	Six pointed star

Code	Line style
-	Solid line
--	Dashed line
:	Dotted line
-.	Dash-dot line

- Plot 10 points forming an ellipse with the semi-major axis at 45° to the x-axis. The points should be cyan downwards facing triangles and joined by a red line.

Plot will accept additional arguments controlling various aspects of the plot as ‘key’,value pairs. Some examples and their default values are given below.

Key	Default
LineWidth	0.5
MarkerSize	6
MarkerEdgeColor	auto
MarkerFaceColor	none
FontSize	10
FontAngle	normal

After using the plot command we can issue commands to alter the display xlabel, ylabel and title should be self explanatory (but if not use help xlabel). The axes can be controlled by the following commands

Command	Action
axis([xmin xmax ymin ymax])	Set limits for both axes
axis auto	Automaticaly choose axes limits
axis equal	Eqalise units on x and y axes
axis off	No axes
axis square	Axes are square round plot
axis tight	Axes limits same as those of data
xlim([xmin xmax])	Set limist of x axis
ylim([ymin ymax])	Set limits of y-axis

For instance

```
hold off
for i=3:2:200
 plot(fft(eye(i))); axis equal; axis off;
 title(sprintf('n=%d',i))
 pause(1)
end
```

pause(1) waits for one second before displaying the next plot. hold off ensures a new plot is generated each time. (hold on would retain the old plotted data as well as plotting the new data.) sprintf has the same format as fprintf but prints to a string rather than the screen or a file. fft stands for fast fourier transform. Use CTRL-C to cancel the calculation if you get bored before n reaches 200.

- Plot a circle, and adjust the axes to ensure it appears circular on the screen.

Matlab graphics includes a LaTeX interpreter you can use to typeset equations in the axis labels, the title or anywhere in the plot.

```
n=10; m=100;
x=1:n;
y=zeros(m,1);

pvals=linspace(1,10,m);
for i=1:m
  y(i)=norm(x,pvals(i));
end
```

```
plot(pvals,y,’LineWidth’,2)
% Octave does not support latex text objects
if ˜exist(’OCTAVE_VERSION’)
  % Running matlab
  options={’Interpreter’,’latex’,’FontSize’,18};
  ylabel(’$\| x \|_p$’,options{:},’HorizontalAlignment’,’right’)
  xlabel(’$p$’,options{:})
  title([’\slshape Vector $p$-norm, for $x=’ ...
 ’[1 2 \ldots˜’ int2str(n) ’]ˆT$’],options{:})
  s=’$$\|x\|_p=\biggl(\sum_{i=1}ˆn |x_i|ˆp\biggr)ˆ{1/p}$$’;
  text(options{:},’String’,s,’Position’,[3 40])
else
  % Running octave
  % this part is only here so I can run the script
  % at home.
  options={’Interpreter’,’tex’,’FontSize’,18};
  ylabel(’||x||_p’,options{:},’HorizontalAlignment’,’right’);
  xlabel(’p’,options{:});
  title([’Vector p-norm, for x=’ ...
 ’[1 2 .. ’ int2str(n) ’]ˆT’],options{:});
  s=’||x||_p=(\Sigma_{i=1}ˆn |x_i|ˆp)ˆ{1/p}’;
  text(options{:},’String’,s,’Position’,[3 40])
end
```

- Adjust the limits on the previous plot so that the y axis goes down to zero.

- Plot a graph of your favourite function including axes labels and use the title to explain why it is your favourite.

The same figure can include multiple plots using the `subplot` command. `subplot(n,m,p)` splits the plotting window into $n \times m$ array of regions and ensures that the next plot will go into the $p$th region.

```
subplot(2,2,1), fplot(’exp(sqrt(x)*sin(12*x))’,[0,2*pi])
subplot(2,2,2), fplot(’sin(round(x))’,[0,10],’--’)
subplot(2,2,3), fplot(’cos(30*x)/x’,[0.01 1 -15 20], ’-.’)
subplot(2,2,4), fplot(’[sin(x),cos(2*x),1/(1+x)]’,[0,5*pi,-1.5,1.5])
```

`fplot` plots a function contained in a string. An additional arguement `[xmin xmax]` or `[xmin,xmax,ymin,ymax]` controls the axis limits.

More complex geometries can be made up by using different mesh patterns.

```
x=linspace(0,15,100);
subplot(2,2,1), plot(x,sin(x))
subplot(2,2,2), plot(x,round(x))
subplot(2,1,2), plot(x,sin(round(x)))
```

A full list of the matlab 2D plotting functions is given below. Use the `help` command to learn more about them

`plot`	Simple x-y plot
`loglog`	Plot with logarithmic axes
`semilogx`	Plot with logarithmic x-axis
`semilogy`	Plot with logarithmic y-axis
`plotyy`	Plot with y-axes on left and right
`polar`	Plot in polar coordinates
`fplot`	Plot a function
`ezplot`	Easy to use function plotter
`ezpolar`	Easy to use polar plotter
`fill`	Polygon fill
`area`	Filled area graph
`bar`	Bar graph
`barh`	Horizontal bar graph
`hist`	Histogram
`pie`	Pie chart
`comet`	Animated, comet like, x-y plot
`errorbar`	Errorbar plot
`quiver`	Quiver (vector field) plot
`scatter`	Scatter plot
`stairs`	Stairstep plot

- Create four plots in the same window each containing different types of plot.

- Create three side by side plots in the same window which describe functions which give straight lines when plotted on loglog, semilogx and semilongy graphs.

Three dimensional graphics

Lines The `plot3` command plots a line from x, y and z vectors.

```
t=-5:.005:5;
x=(1+t.^2).*sin(20*t);
y=(1+t.^2).*cos(20*t);
```

```
z=t;

plot3(x,y,z)
grid on
FS='FontSize';
xlabel('x(t)',FS,14)
ylabel('y(t)',FS,14)
zlabel('z(t)',FS,14,'Rotation',0)
title('plot3 example',FS,14)
```

It accepts the same line/point colour/style options as `plot`. E.g. `plot3(x,y,z,'r+')`.

- Create a plot of a double helix.

Contours The contour command produces contour plots.

```
x=-2:0.01:2; y=-1:0.01:1;
[X,Y]=meshgrid(x,y);
Z=sin(3*Y-X.^2+1)+cos(2*Y.^2-2*X);
contour(x,y,Z,20) % 20 contour levels
```

Meshes and surfaces Mesh plots

```
x=0:0.1:pi; y=x;
[X,Y]=meshgrid(x,y);
Z=sin(Y.^2+X)-cos(Y-X.^2);

subplot(221)
mesh(Z)

subplot(222)
meshc(Z)

subplot(223)
mesh(x,y,Z)
axis([0 pi 0 pi -5 5])

subplot(224)
mesh(Z)
hidden off
```

Surface plots

```
subplot(221)
surf(Z)

subplot(222)
surfc(Z)

subplot(223)
surf(Z), shading flat

subplot(224)
waterfall(Z)
```

Exporting graphics

To export a figure as an eps file use the print command

```
print -deps temp.eps
```

The figure can then be included in a LaTeX document.

N.B. All the defaults in Matlab are set for screen viewing. To make them suitable for printing you need to make the font and marker sizes bigger and the lines thicker.