fplot and plot functions 1
using fplot 1
using plot 3
plot multiple x,y pairs 4
axis settings, limits, tickmarks 5
semilogy, semilogx, loglog plots 7
scatter plots 9
subplots 12
mesh, surf, surfc, surface plots 14
Wigley Hull example 16
NACA Airfoils 17
plot3 19
surfaces of revolution 20
Toriccelli's trumpet, Gabriel's horn 20
Gabriel's cake 21
movies 22

```
% week04.m - week-4 class examples
% 440:127 - S. J. Orfanidis
%
% examples: fplot and plot functions
%           axis settings, limits, tickmarks
%           semilogy, semilogx, loglog plots
%           scatter plots - Monte Carlo simulations
%           subplots
%           mesh, surf, surfc, surface plots
%           plot3
%           surfaces of revolution
%           movies

clear all

% ------------------------------------------------------------------------
```

fplot and plot functions

---

using fplot

```
f = @(x) exp(-0.5*x).*sin(5*x);

figure; fplot(f, [0,5]);           % quick function plot

figure; fplot('exp(-0.5*x).*sin(5*x)', [0,5], 'b-');
xlabel('x'); ylabel('y'); grid;
title('f(x) = e^{-0.5x} sin(5x)')
```

$f(x) = e^{-0.5x} \sin(5x)$

using plot

```
x = linspace(0,5,101);
y = f(x);

figure; plot(x,y,'b-');                    % blue solid line
xlabel('x'); ylabel('y'); grid;
title('f(x) = e^{-0.5x} sin(5x)');


figure; plot(x,y,'b-', ...
             x,exp(-0.5*x),'r--', ...
             x,-exp(-0.5*x),'r--');        % plot three curves

xlabel('x'); ylabel('y'); grid;
title('f(x) = e^{-0.5x} sin(5x)');
xaxis(0,5,0:5);                       % in sakai's course-functions folder
yaxis(-1,1,-1:0.5:1);
legend(' damped sinusoid', ' envelope', 'location','ne');
```

plot multiple x,y pairs

```
% the plot function can have multiple x,y pairs, not necessarily of
% the same length. For example, one pair can consist of a single point, or
% a couple of points. Here is an example:
%
% the task is to plot the curve y = f(x) = 2*x + sin(pi*x),
% then add the points at the local maximum and minimum,
% then draw a straight line between the beginning and ending points
% of the curve, indicating those points with markers

f = @(x) 2*x + sin(pi*x);

x = 0 : 0.01 : 2;                   % same as x = linspace(0,2,201);

x1 = fminbnd(@(x) -f(x), 0.5,1);    % search for max in [0.5,1]
y1 = f(x1);                         % true x1 = acos(-2/pi)/pi, from calc
x2 = fminbnd(f,1,2);                % search for min in [1,2]
y2 = f(x2);                         % true x2 = 2-acos(-2/pi)/pi

figure; plot(x,f(x),'b-', [0,2],[0,4],'r.--', ...
             x1,y1,'ro',  x2,y2,'go');

legend(' f(x)', ' line', ' max', ' min', ...
       'location', 'se');
```

```
xaxis(0,2, 0:0.5:2);
yaxis(0,4, 0:4); grid on;
xlabel('x');



% ----------------------------------------------------------------------
```

axis settings, limits, tickmarks

```
clear all

f = @(x) exp(-0.5*x).*sin(5*x);

x = linspace(0,5,101);    % define a vector of x-points
y = f(x);                 % vector of function values at x

figure; plot(x,y,'b-');
xlabel('x'); ylabel('y'); grid on;
title('f(x) = e^{-0.5x} sin(5x)');

% axis equal;     % equal x,y units
% axis square;    % square box
```

```
% axis off;       % remove axes
% axis on;        % restore axes
% axis ij;        % matrix mode (i=vert, j=horiz)
% axis xy;        % cartesian mode
% axis normal;    % default axis

%axis([xmin,xmax,ymin,ymax]);              % set axis limits
%axis([xmin,xmax,ymin,ymax,zmin,zmax]);

% xlim([xmin,xmax]);        % set x-axis limits
% ylim([ymin,ymax]);
% zlim([zmin,zmax]);
%
% set(gca, 'xtick', v);     % v = tickmark vector
% set(gca, 'ytick', v);
%
% xaxis(xmin, xmax, v);     % v = tickmark vector

% box on;
% box off;



% ----------------------------------------------------------------------
```

semilogy, semilogx, loglog plots

```
clear all
                                        % Moore's law data
Y = load('transistor_count.dat');       % file on sakai

y = Y(:,1);                             % transistor count
t = Y(:,2);                             % year

% num2str(Y,'%1.0f')                    % look at Y

y1 = y(1);                              % y1 = 2300, starting count
t1 = t(1);                              % t1 = 1971, starting year

ypred = y1 * 2.^((t-t1)/2);             % predicted count

figure;
semilogy(t,y,'b.', t,ypred,'r-');               % semilogy plot of y
                                                % similar to plot of log10(y)
xlabel('year'); ylabel('count');
title('transistor count - Moore''s law');           % note double '' quote
legend(' data', ' predicted', 'location','nw');

% yaxis(10^2, 10^10, 10.^(2:10))                % adjusting semilogy limits

figure;
plot(t,log10(y),'b.', t,log10(ypred),'r-');     % plot of log10(y) vs. t

xlabel('year'); ylabel('log10(count)');
title('transistor count - Moore''s law');
legend(' log10(data)', ' predicted', 'location','nw');

figure;
plot(t,y,'b.', t,ypred,'r-');                   % ordinary plot of y

xlabel('year'); ylabel('count'); title('Moore''s law');
legend(' data', ' predicted', 'location','nw');



% ---------------------------------------------------------------------------
```

transistor count - Moore's law
data
predicted
count
year
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
1970
1980
1990
2000
2010

transistor count - Moore's law
log10(data)
predicted
log10(count)
year
10
8
6
4
2
1970
1980
1990
2000
2010

scatter plots

```
% Example 1  -  Monte Carlo calculation of the area under a sine curve

clear all

N=10000; rng(101);
x = pi * rand(1,N);
y = rand(1,N);

i = find(y < sin(x));
j = find(y > sin(x));

figure;

scatter(x(i),y(i),1,'r');    % plot points under the sine curve
hold on;
scatter(x(j),y(j),1,'b');    % plot points abive sine curve

x = linspace(0,pi,100);
y = sin(x);
plot(x,y,'r-');              % plot sine curve itself

xaxis(0,pi,0);
```

```
text(0.99*pi,-0.04, '\pi');

yaxis(0,1,[0,1]);
xlabel('\itx'); title('sin({\itx})');

A = length(i)/N * pi;

S = {['calculated area = ',num2str(A)]; 'exact area = 2'}
text(0.3, 0.1, S, 'fontsize', 16, 'color','blue');

box on;


% Example 2  -  Monte Carlo calculation of the area of a circle

clear all

rng(100);                       % initialize random number generator;

N=10000;                        % number of points in rectangle
x = 2*rand(1,N) - 1;            % rand(), is random number in [0,1]
y = 2*rand(1,N) - 1;            % x,y are in the range [-1,1]

i = find(x.^2 + y.^2 <= 1);     % find all points inside circle
j = find(x.^2 + y.^2 > 1);      % find all points outside circle

figure;
scatter(x(i),y(i),1,'r');       % plot points inside circle
hold on;                        % hold graph
scatter(x(j),y(j),1,'b');       % plot points outside circle

t = linspace(0,1,361);
x = cos(2*pi*t);
y = sin(2*pi*t);

plot(x,y,'r-');                 % add circle itself

A = length(i)/N * 4             % why the factor 4?

axis equal;
axis square;
box on;

line([0,0], [-1,1], 'linestyle', ':', 'color', 'b');
line([-1,1], [0,0], 'linestyle', ':', 'color', 'b');

hold off



% ------------------------------------------------------------------------
```

```
S =
```

```
    'calculated area = 1.9915'
    'exact area = 2'


A =

    3.1408
```

subplots

```
x = linspace(1,10,200);

y1 = sin(x).^2;
y2 = 1./x;
y3 = exp(-0.3*x).*cos(5*x);
y4 = 1./floor(x);

figure;

subplot(2,2,1);             %   |--------|--------|
plot(x,y1,'b');             %   |   y1   |   y2   |
                            %   |--------|--------|
subplot(2,2,2);             %   |   y3   |   y4   |
plot(x,y2,'r');             %   |--------|--------|

subplot(2,2,3);
plot(x,y3,'m');

subplot(2,2,4);
plot(x,y4,'g');
yaxis(0,1.05, 0:0.5:1);
```

```
% -------------------------

figure;

subplot(2,3,1);              %   |------|------|------|
plot(x,y1,'b');              %   |  y1  |  y2  |  y3  |
                             %   |------|------|------|
subplot(2,3,2);              %   |      |      |      |
plot(x,y2,'r');              %   |------|------|------|

subplot(2,3,3);
plot(x,y3,'m');

subplot(2,1,2);              %   |--------------------|
plot(x,y4,'g');              %   |                    |
yaxis(0,1.05, 0:0.5:1);      %   |--------------------|
                             %   |         y4         |
                             %   |--------------------|



% ------------------------------------------------------------------------
```

mesh, surf, surfc, surface plots

```
clear

x = linspace(-5,5,51);
y = linspace(-5,5,51);

[X,Y] = meshgrid(x,y);

Z = (X.^2 + Y.^2) .* exp(-(X.^2 + Y.^2)/2);

% alternative way of calculating matrices X,Y,Z using for-loops:
% note that x is replicated row-wise, so i is the row index
% and y is replicated vertically, so j is the column index
%
% for j=1:length(x)
%     for i=1:length(y)
%         X(i,j) = x(j);              % fill i-th row with x
%         Y(i,j) = y(i);              % fill j-th column with y
%         Z(i,j) = (x(j)^2 + y(i)^2) * exp(-(x(j)^2 + y(i)^2)/2);
%     end
% end

figure; mesh(X,Y,Z);
```

```
figure; surf(X,Y,Z);

colorbar;

% shading interp;
% shading faceted;
% shading flat;
%
% colormap jet;
% colormap hsv;
% colormap hot;
% colormap cool;
% colormap spring;
% colormap autumn;
% colormap summer;
% colormap winter;
% colormap gray;
% colormap copper;
```

Wigley Hull example

```
% Reference:
% A. Gilat, "MATLAB, An Introduction with Applications", 4/e, Wiley, 2011

B = 0.5; L = 4; D = 0.5;
x = linspace(-L/2, L/2, 21);
z = linspace(-D, 0, 21);
[X,Z] = meshgrid(x,z);

Y = B/2 * (1 - 4*X.^2/L^2) .* (1 - Z.^2/D.^2);

figure;
surfc(X,Y,Z);
hold on;
surfc(X,-Y,Z);
```

NACA Airfoils

```
% NACA = National Advisory Committee for Aeronautics

% Reference:
% A. Gilat, "MATLAB, An Introduction with Applications", 4/e, Wiley, 2011

c = 1; t = 0.2;

f = @(x) t*c/0.2 * (0.2969*sqrt(x/c) ...
         - 0.1260*(x/c) -0.3516*(x/c).^2 ...
         + 0.2843*(x/c).^3 - 0.1015*(x/c).^4);

x = linspace(0,c,101);

y = f(x);

figure;
plot(x,y,'b-', x,-y,'b-');

figure;
area(x,y);            % area plot
hold on;
area(x,-y);
```

```
colormap summer;



% ----------------------------------------------------------------------
```

plot3

```
t = 0:0.01:100;

x = exp(0.03*t).*cos(t);
y = exp(0.03*t).*sin(t);
z = t;

figure;
plot3(x,y,z,'b');
grid on;



% ---------------------------------------------------------------------------
```

surfaces of revolution

rotate a function f(x), a<=x<=b about the x-axis

Toriccelli's trumpet, Gabriel's horn

```
x = linspace(1,15,50);
th = linspace(0,2*pi,31);

[X,Th] = meshgrid(x,th);

F = 1./X;             % function to be rotated about the x-axis

Y = F.*cos(Th);
Z = F.*sin(Th);

figure;

surf(X,Y,Z);

view(-25,15);

colormap hsv;

% rotating a function f(z), a<=z<=b about the z-axis
% simply interchange roles of x,z, but do surf(X,Y,Z)
```

```
% or, use the built-in function cylinder
%
% z = linspace(1,15,50);
% th = linspace(0,2*pi,31);
%
% [Z,Th] = meshgrid(z,th);
%
% F = 1./Z;             % function to be rotated about the x-axis
%
% Y = F.*cos(Th);
% X = F.*sin(Th);
%
% figure;
%
% surf(X,Y,Z);
%
% view(-25,15);
%
% colormap hsv;
```

Gabriel's cake

```
x = linspace(1,10,100);
th = linspace(0,2*pi,31);

[X,Th] = meshgrid(x,th);
```

```
F = 1./floor(X);      % step-wise version of 1./X

Y = F.*cos(Th);
Z = F.*sin(Th);

figure;

surf(X,Y,Z);

view(25,10);

colormap cool;


% ------------------------------------------------------------------------
```

movies

```
% hoops      % throwing the perfect basketball shot

% receiver   % moving wide-receiver catching a ball thrown by the QB

% cycloid    % cycloid curve traced by a point on a rolling wheel
```

```
% dipmovie   % EM wave emitted by an antenna, e.g., a cell phone antenna
```

Published with MATLAB® R2013b