Aaron Klapheck

```
% Final_word (word document form)
clear, clc
Date = date
```

Date =

19-Dec-2007

Problem 1.

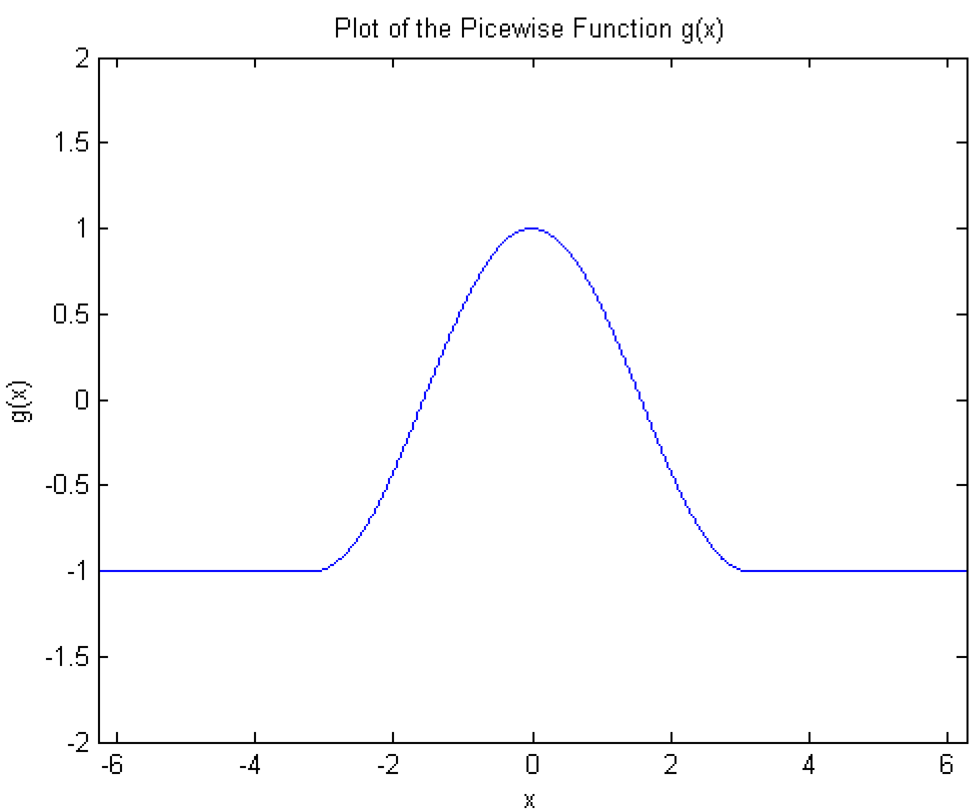
```
% a.
Z = [3, -2, 1, 6; 6, 8, -5, 7; 7, 9, 10, -8; 9, 5, 2, -4]

% b.
Y = Z(2:4, :)

% c.
V = [Z(:, 1), Z(:, 2).*Z(:, 4), Z(:, 3)]

% d.
Z_ascending = sort(Z');
Z_ascending = Z_ascending'
```

```
Z =

     3    -2     1     6
     6     8    -5     7
     7     9    10    -8
     9     5     2    -4
```

```
Y =

     6     8    -5     7
     7     9    10    -8
     9     5     2    -4


V =

     3   -12     1
     6    56    -5
     7   -72    10
     9   -20     2


Z_ascending =

    -2     1     3     6
    -5     6     7     8
    -8     7     9    10
    -4     2     5     9
```

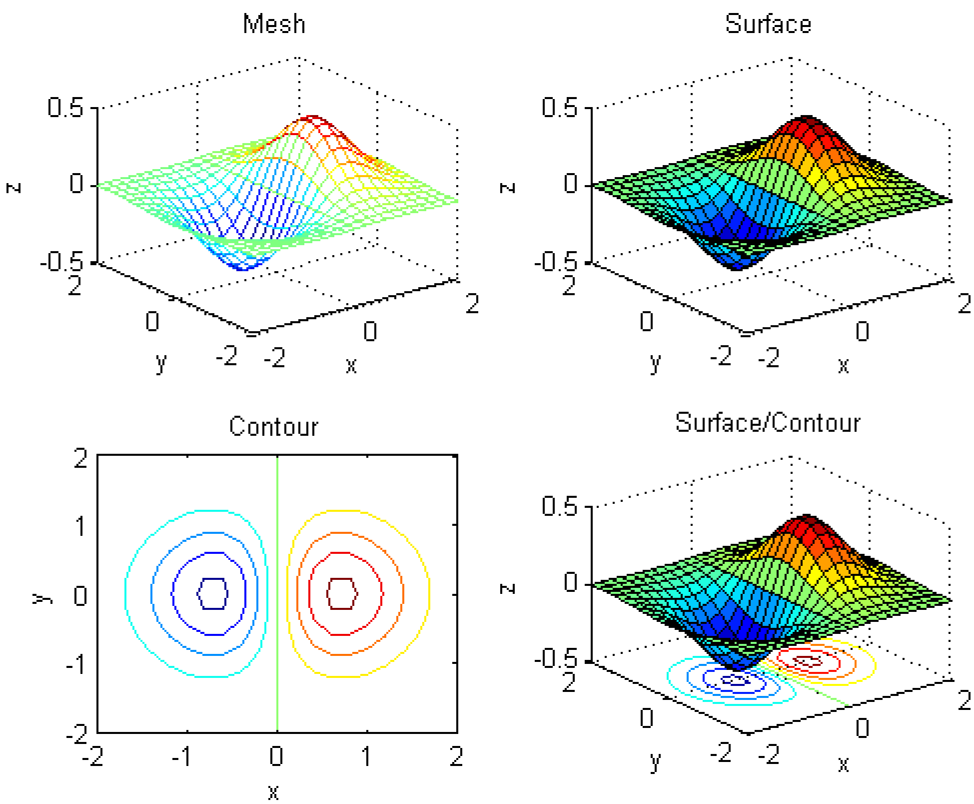
Problem 2.

```
% Create and plot a function.

x = [-2*pi:0.001:2*pi];

% g(x) is created in a seperate script file. See the end of this document to
% view the script file.
plot(x, g(x)), axis([-2*pi 2*pi -2 2]), xlabel('x'), ylabel('g(x)'), ...
    title('Plot of the Picewise Function g(x)')
```



Problem 3.

```
[x, y] = meshgrid(-2:0.2:2);
z = x.*exp(-x.^2 - y.^2);

subplot(2, 2, 1)
mesh(x,y,z), xlabel('x'), ylabel('y'), zlabel('z'), ...
    title('Mesh')

subplot(2, 2, 2)
surf(x,y,z), xlabel('x'), ylabel('y'), zlabel('z'), ...
    title('Surface')

subplot(2, 2, 3)
contour(x,y,z), xlabel('x'), ylabel('y'), ...
    title('Contour')

subplot(2, 2, 4)
surfc(x,y,z), xlabel('x'), ylabel('y'), zlabel('z'), ...
    title('Surface/Contour')
```



Problem 4.

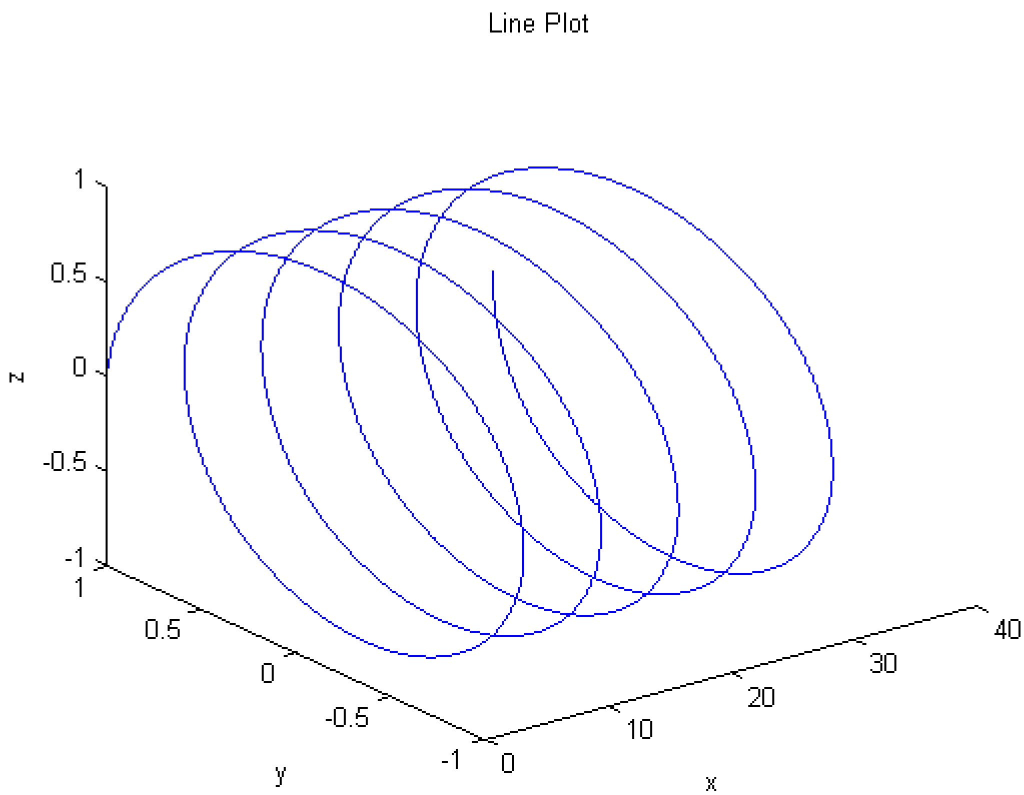
```
% Set-up inputs and outputs of a function y.
x = [0:0.01:1];
y = x.^2 - 3.*x + 2;
% Use the logical array C as a mask.
C = (y < 0.2);
% Looking at the values of C, the range of x values seems to be near
% the end and all clumped together.

% Get the length of matrix x to find the number of x values that corespond
% to y values that are less than 0.2.
length_of_x = length(x);
number_of_x_values = C*ones(length_of_x, 1)
```

```
% The domain (x-values) that corospond to y < 0.2.
domain_for_y_less_than_point_two = x(C)

% The range (y-values) corosponding to these x-values (domain, see above).
range_for_y_less_than_point_two = (x(C)).^2 - 3.*(x(C)) + 2
```

```
number_of_x_values =

    18


domain_for_y_less_than_point_two =

  Columns 1 through 9

    0.8300    0.8400    0.8500    0.8600    0.8700    0.8800    0.8900    0.9000
0.9100

  Columns 10 through 18

    0.9200    0.9300    0.9400    0.9500    0.9600    0.9700    0.9800    0.9900
1.0000


range_for_y_less_than_point_two =

  Columns 1 through 9

    0.1989    0.1856    0.1725    0.1596    0.1469    0.1344    0.1221    0.1100
0.0981

  Columns 10 through 18

    0.0864    0.0749    0.0636    0.0525    0.0416    0.0309    0.0204    0.0101
0
```

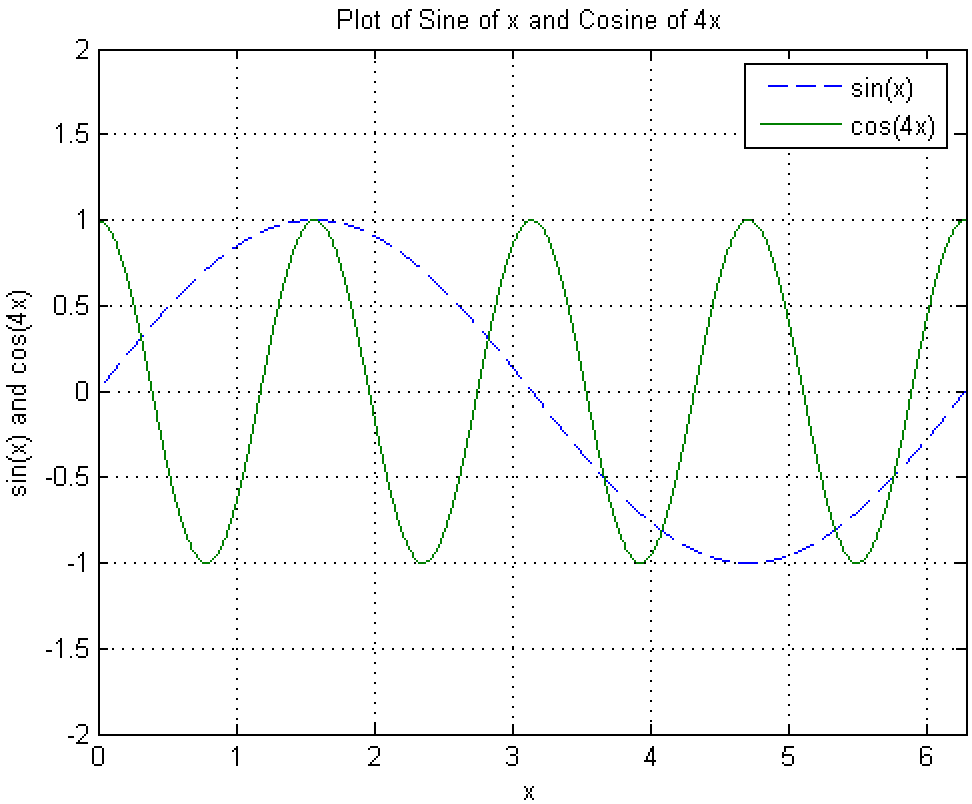
Problem 5.

```
x = [0:0.001:10*pi];
y = cos(x);
z = sin(x);

subplot(1, 1, 1)
plot3(x, y, z), xlabel('x'), ylabel('y'), zlabel('z'), ...
    title('Line Plot')
```



Problem 6 - General plot

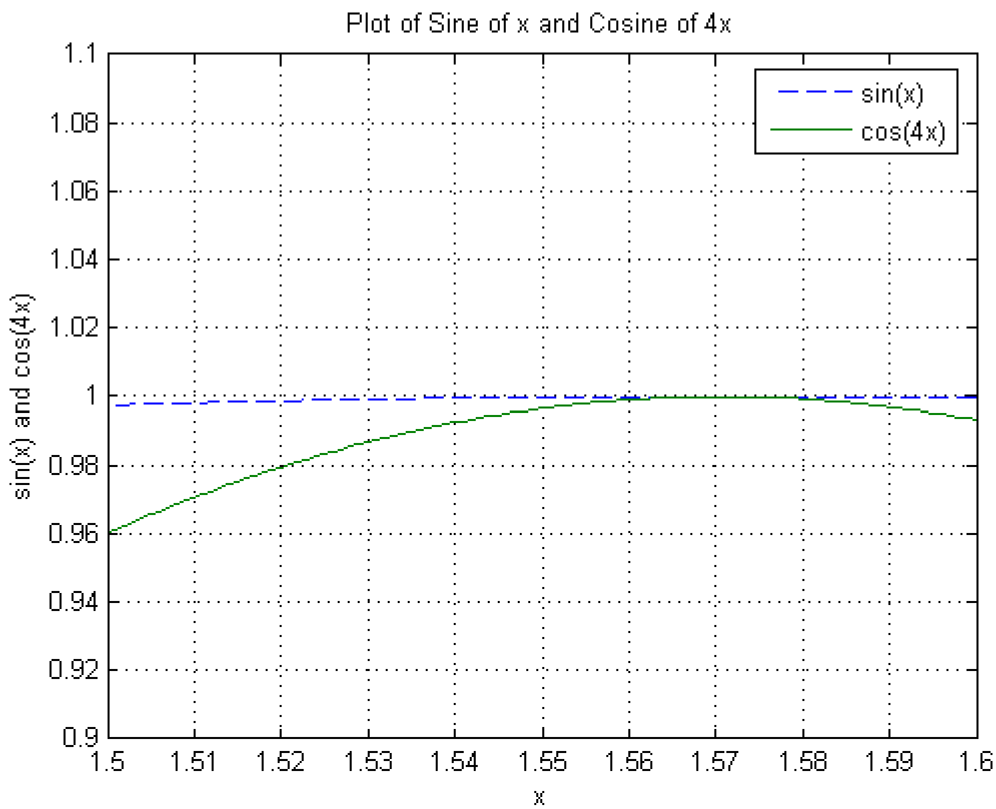
```
%Use this plot to find an approximate range where the two funtions
%intersect at a maximum.

x = [0:0.001:2*pi];
f_x = sin(x);
g_x = cos(4.*x);

plot(x, f_x, '--', x, g_x), xlabel('x'), ylabel('sin(x) and cos(4x)'), ...
    title('Plot of Sine of x and Cosine of 4x'), ...
    legend('sin(x)', 'cos(4x)'), axis([0 2*pi -2 2]), grid

% They intersect at approxamatly x = 1.6 and y = 1.
```



Problem 6 - Finding intersection

```
plot(x, f_x, '--', x, g_x), xlabel('x'), ylabel('sin(x) and cos(4x)'), ...
    title('Plot of Sine of x and Cosine of 4x'), ...
    legend('sin(x)', 'cos(4x)'), axis([1.5 1.6 0.9 1.1]), grid, ...
    [x, y] = ginput(1)

% The x and y coordinates where the two graphs intersect are listed below.
```

```
x =

    1.5725

y =

    1.0003
```



Problem 7.

```
p_1 = [20, -7, 5, 10];
p_2 = [5, 0, 15, -3];

product = conv(p_1, p_2)
```

```
product =

   100    -35    325   -115     96    135    -30
```

Problem 8.

```
% Given
% p = h*d*g, where h is height, d is density, and g is gravitational
% acceleration.
% h = p/(dg)
% p (Pa) range = 10^3.*[0:10:100].
% d_mercury = 13,560 kg/m^3
% d_water = 1000 kg/m^3
% g = 9.81 m/s^2

p = 10^3.*[0:10:100];
d_mercury = 13560;
d_water = 1000;
g = 9.81;

% Find height (h) for both d_mercury and d_water.

h_mercury = p./(d_mercury*g);
h_water = p./(d_water*g);
```

```
% Create matrix A to be use in formating the output in a desired way.
A = [p; h_mercury; h_water];
% Format the data in an easy to read way. The numerical data underneith
% each barometer corosponds to the height the liquid has risen in meters.
fprintf('Pressure (Pa)    Mercury Barometer   Water Barometer \n')
fprintf('%2.1e \t %1.5f \t \t %1.5f \n', A) % very hard to align these exactly.
```

```
Pressure (Pa)    Mercury Barometer   Water Barometer
0.0e+000        0.00000               0.00000
1.0e+004        0.07517               1.01937
2.0e+004        0.15035               2.03874
3.0e+004        0.22552               3.05810
4.0e+004        0.30070               4.07747
5.0e+004        0.37587               5.09684
6.0e+004        0.45105               6.11621
7.0e+004        0.52622               7.13558
8.0e+004        0.60140               8.15494
9.0e+004        0.67657               9.17431
1.0e+005        0.75175               10.19368
```

Problem 9.

```
% Given
% Innitial amount (A) = $10,000
% 7.5% interest is earned.
% At the end of the year $1,000 is given out.


A = 10000;
t = 0;

while A > 0
    A = A + A*0.075 - 1000;
    t = t + 1;
end % A > 0

Time_in_years = t
```

```
Time_in_years =

    20
```

Problem 10.

```
% For A and B find: a. mean, b. median, c. the standard deviation.

A = [60, 68, 95, 88, 74, 65];
B = [71, 77, 78, 72, 79, 73];

% a.
mean_of_A = mean(A)
mean_of_B = mean(B)

% b.
median_of_A = median(A)
median_of_B = median(B)

% C.
standard_deviation_of_A = std(A)
standard_deviation_of_B = std(B)
```

```
mean_of_A =

    75


mean_of_B =
```

75

median_of_A =

71

median_of_B =

75

standard_deviation_of_A =

13.7405

standard_deviation_of_B =

3.4059

Problem 11.

```
month = input('your birth month (a number from 1 to 12): ');

if month < 5 & month > 0
    fprintf('MATLAB is fun! \n \n')
elseif month < 10 & month > 5
    fprintf('I am happy I am a student! \n \n')
elseif month <= 12 & month > 10
    fprintf('I do not want ME75 to end! \n \n’) % can't use the word "don't".
else
    fprintf('please enter a natural number from 1 to 12 \n \n')
end
```

your birth month (a number from 1 to 12): 2
MATLAB is fun!

your birth month (a number from 1 to 12): 7
I am happy I am a student!

your birth month (a number from 1 to 12): 11
I do not want ME75 to end!

your birth month (a number from 1 to 12): -2
please enter a natural number from 1 to 12

your birth month (a number from 1 to 12): 30
please enter a natural number from 1 to 12

Problem 12.

```
% Given
% P = P_0e^(rt). where:
% P = current population,
% P_0 = original population,
% r = continuous growth rate expressed as a fraction, and
% t = time.

% P starts at = 100 flies
% r = 45%/day
```

```
% 15 flies die per day

P = 100;
r = 0.45;

% Find P for every day.

% Need matrix pop to store the population for the given 10 days
pop = [1:10];
for t = 1:10;
    P = P*exp(r*t) - 15;
    pop(t) = P;
end % t = 1:10

days = [1:10];
A = [days; pop];
fprintf('Days     Population \n')
fprintf('%2.0f \t %2.4e \n', A)
```

```
Days     Population
 1      1.4183e+002
 2      3.3385e+002
 3      1.2728e+003
 4      7.6850e+003
 5      7.2898e+004
 6      1.0847e+006
 7      2.5312e+007
 8      9.2639e+008
 9      5.3172e+010
10      4.7864e+012
```

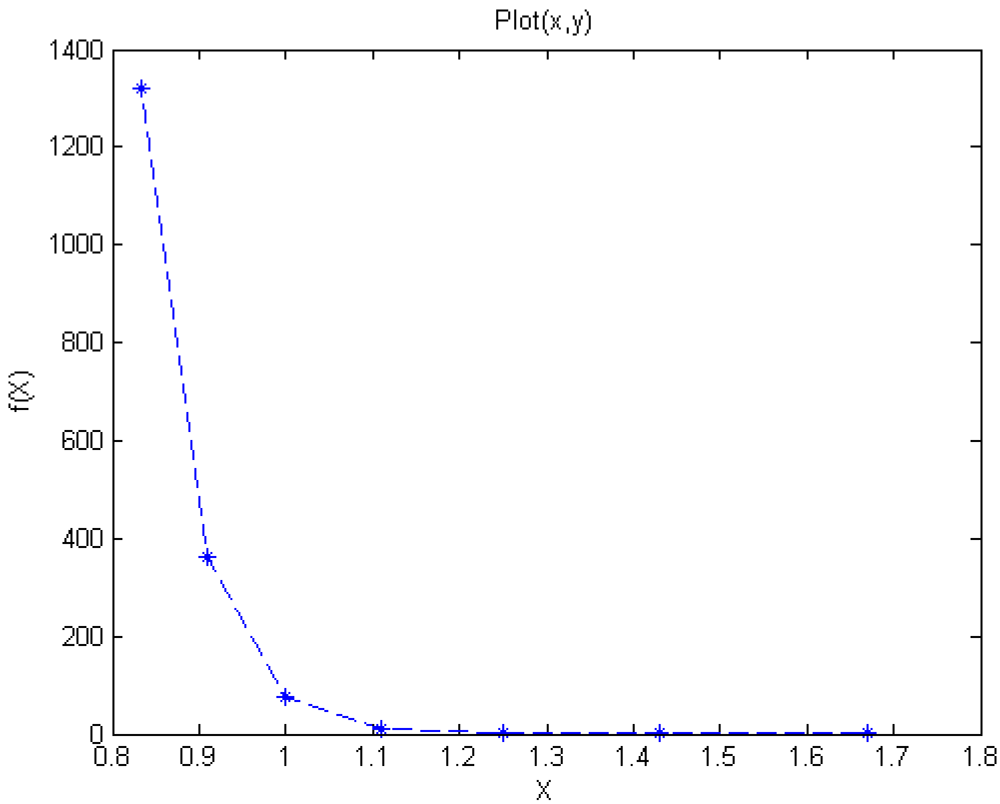
Problem 13.

```
% a.
x = [0:0.001:2];
length_of_x = length(x)

% b.
y = 3.*x.^2 - 4.*x - 2;
[y, x_position] = min(y);
y_value = y
x_value = x(x_position)
```

```
length_of_x =

        2001


y_value =

   -3.3333


x_value =

    0.6670
```

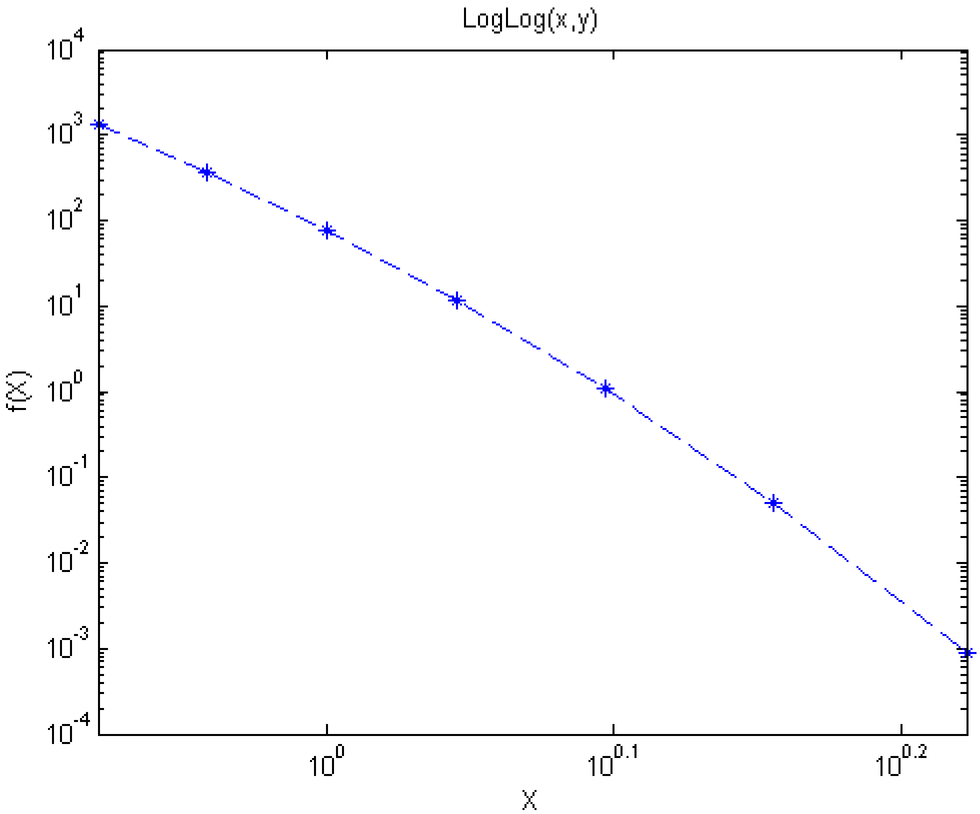
Problem 14 - a.

```
% Linear

X = [1.67, 1.43, 1.25, 1.11, 1.0, 0.909, 0.833];
f_of_X = [0.00088, 0.051, 1.07, 11.5, 76.5, 361, 1320];

plot(X, f_of_X, '*--'), xlabel('X'), ylabel('f(X)'), ...
    title('Plot(x,y)')

% the data is definitly not linear.
```



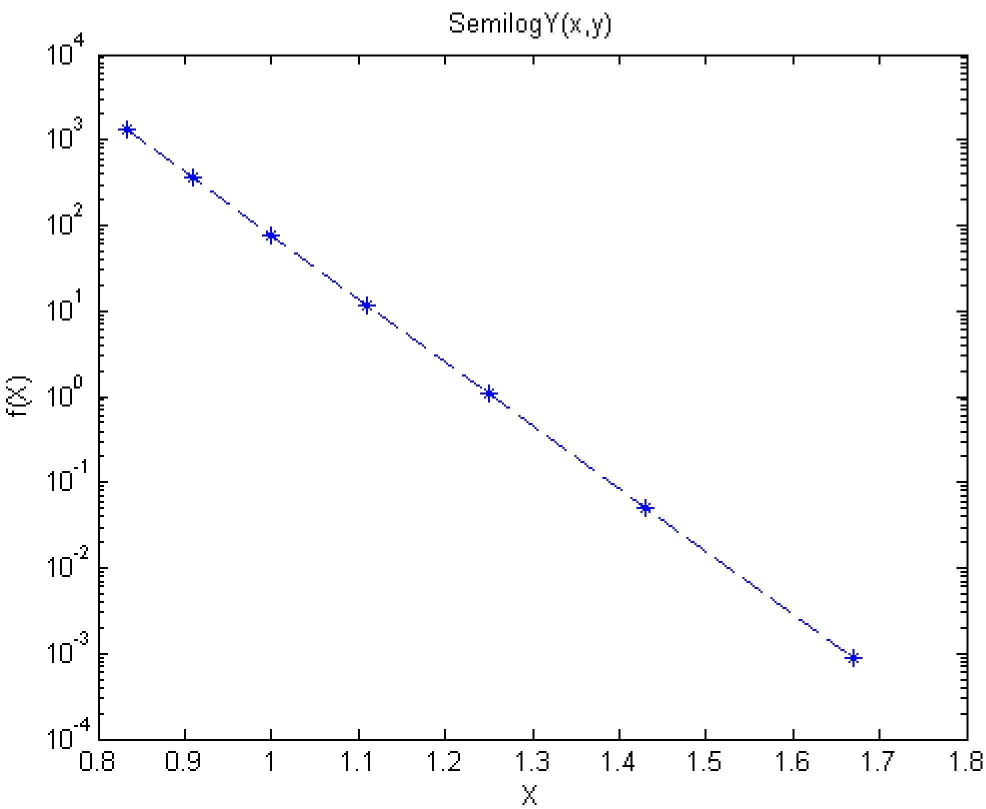
Problem 14 - b.

```
% loglog

loglog(X, f_of_X, '*--'), xlabel('X'), ylabel('f(X)'), ...
    title('LogLog(x,y)')

% Although this plots the data much straighter than in a. it is still
% curved.
```



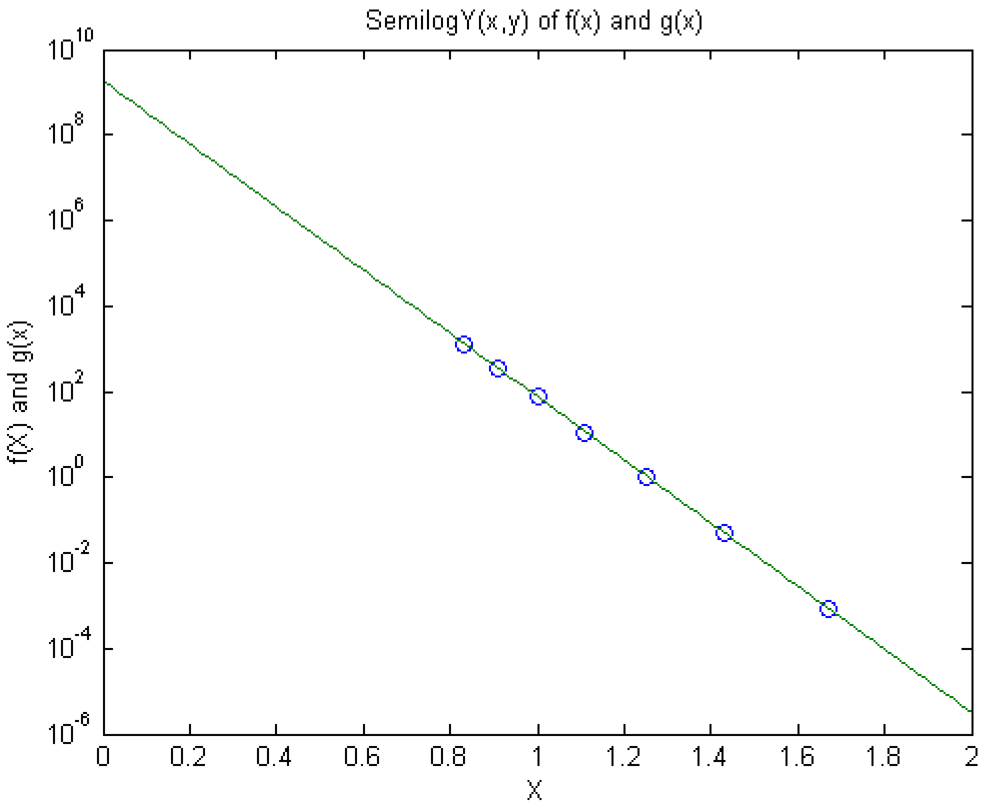
Problem 14 - c.

```
% semilogy

semilogy(X, f_of_X, '*--'), xlabel('X'), ylabel('f(X)'), ...
    title('SemilogY(x,y)')

% Using semilogy the data is plotted along a straight line.
```



Problem 14 - d.

```
% Find a function (g_of_x) that best fits the data.

p = polyfit(X, log10(f_of_X), 1);

x = [2:-0.01:0];

g_of_x = 10^(p(2)).*(10).^(p(1).*x);

semilogy(X, f_of_X, 'o', x, g_of_x), xlabel('X'), ...
    ylabel('f(X) and g(x)'), title('SemilogY(x,y) of f(x) and g(x)')

% Using semilogy the data is plotted along a straight line.
```



Problem 15.

```
% Solve the following set of equations
E = [5, -2, -6; 12, 5, -7; 6, -3, 4], b = [-14; -26; 41],
A_B_C = E\b
% verify
A_B_C = E^(-1)*b
```

```
E =

     5    -2    -6
    12     5    -7
     6    -3     4


b =

   -14
   -26
    41


A_B_C =

     2
    -3
     5


A_B_C =

    2.0000
```

```
   -3.0000
    5.0000
```

Problem 16.

```
% The most important thing I learned in this course was plotting.
% I love this part of the course and got the most out of it.
```

```
%% Function for problem 2

function cosine_of_x_or_negative_1 = g(x);

cosine_of_x_or_negative_1 = -1 + (1 + cos(x)).*heaviside(x + pi) - ...
    (1 + cos(x)).*heaviside(x - pi);
```