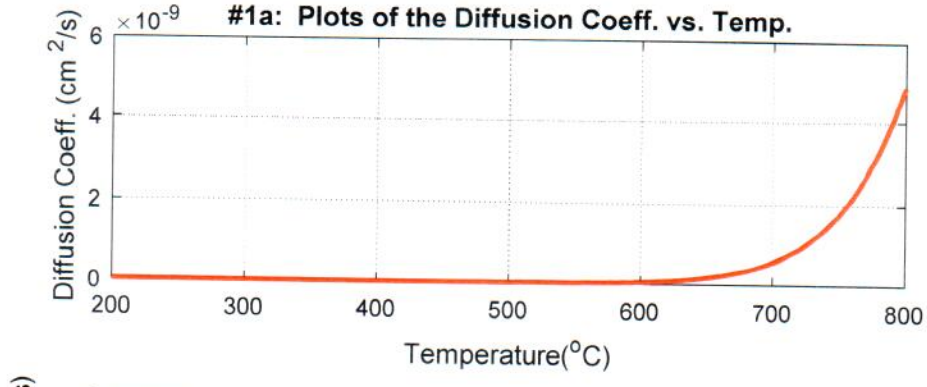
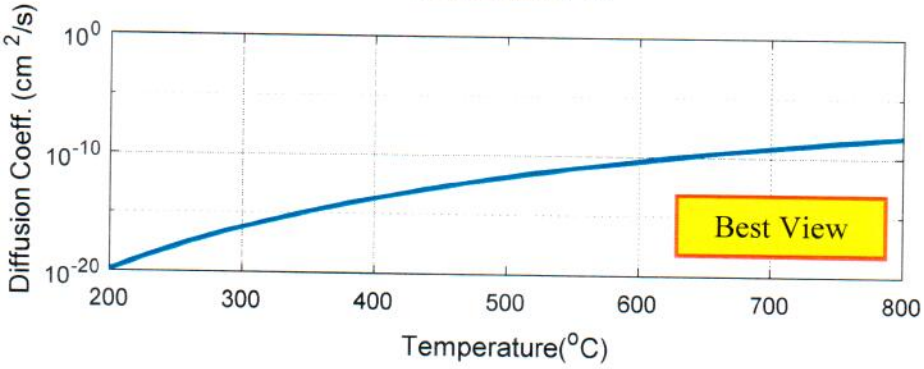
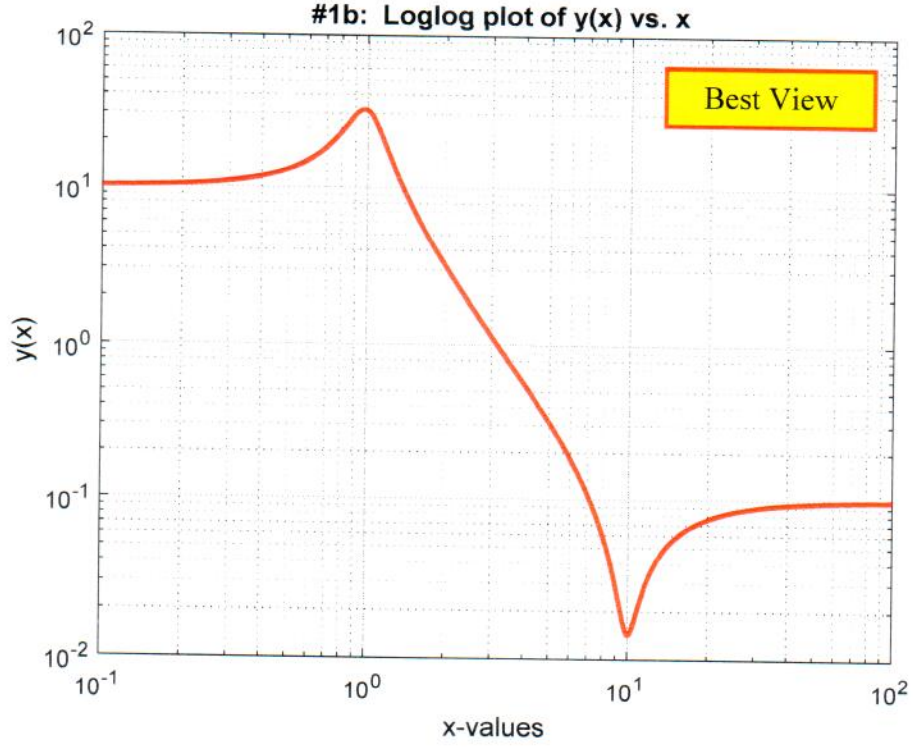
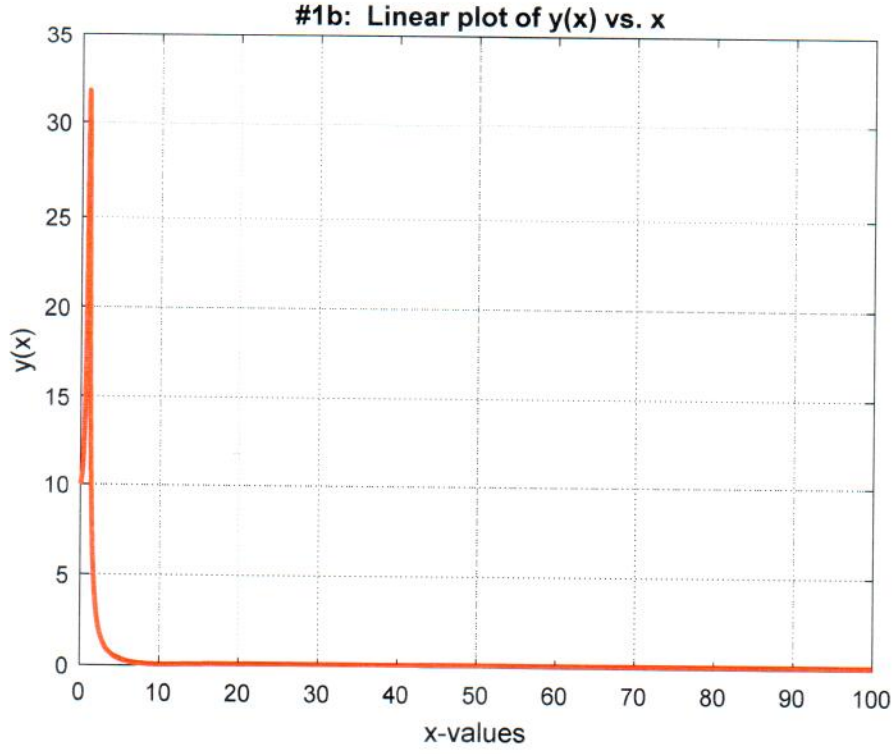
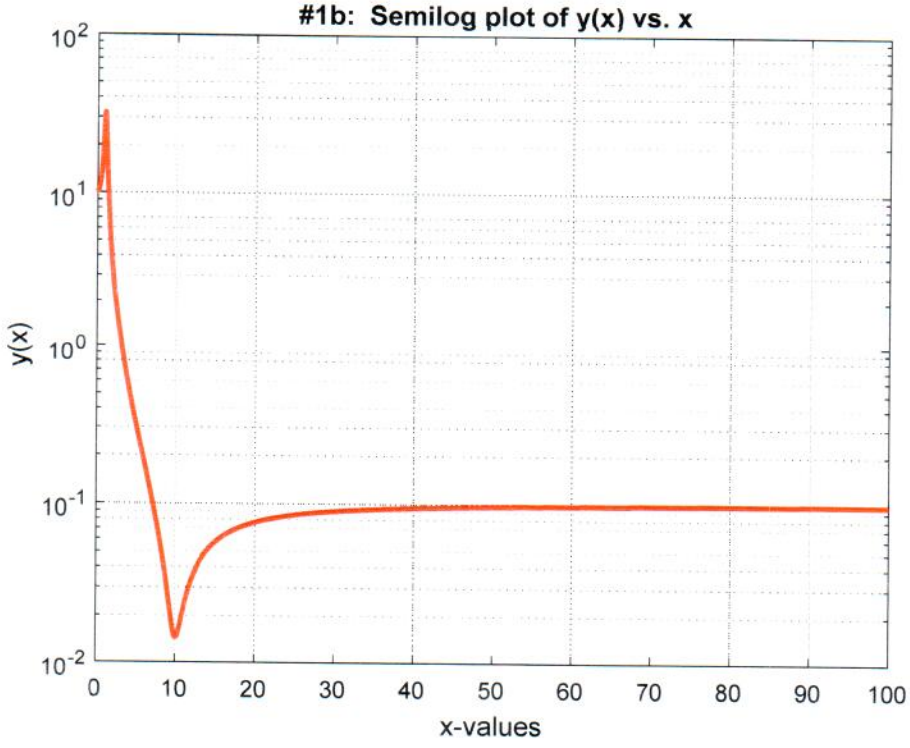
Function #1 Results








#1b: Linear plot of y(x) vs. x
y(x)
x-values
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60
70
80
90
100


#1b: Semilog plot of y(x) vs. x
y(x)
x-values
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
0
10
20
30
40
50
60
70
80
90
100

```
%
%  FUNCTION_1.M   Illustrate some plotting capability in Matlab
%
%  This program simply illustrates how to evaluate and plot some functions in Matlab.
%  The selection of axis scaling (linear vs logarithmic) is emphasized --  always
%  be sure to investigate various options to get the best visualization for the
%  given problem...
%
%  File prepared by J. R. White, UMass-Lowell (last update: Sept. 2017)
%

  clear all,  close all,   nfig = 0;
%
%  Part A
%
%  evaluate D vs. T
  R = 8.314;                % gas constant (J/mol-K)
  Do = 6.18;                % pre-exponential factor (cm^2/s)
  Ea = 187e3;               % activation energy (J/mol)  <-- notice the units
  Tc = 200:10:800;          % temperature (C)
  Tk = Tc+273.15;           % temperature (K)
  D = Do*exp(-Ea./(R*Tk));  % difussion coeff (cm^2/s)
%
%  plot D vs. T
  nfig = nfig+1;   figure(nfig)
  subplot(2,1,1),plot(Tc,D,'r-','LineWidth',2),grid on
  title('#1a:  Plots of the Diffusion Coeff. vs. Temp.')
  xlabel('Temperature(^oC)'),ylabel('Diffusion Coeff. (cm ^2/s)')
  subplot(2,1,2),semilogy(Tc,D,'b-','LineWidth',2),grid on   % best view...
  xlabel('Temperature(^oC)'),ylabel('Diffusion Coeff. (cm ^2/s)')
%
%  Part B
%
%  evaluate function on linear x axis (be careful with the vector arithmetic)
%  NOTE: sometimes it is easier to use intermediate variables for long eqns)
  x1a = linspace(0,10,1000);                          % define 1st range of x values
  x1b = linspace(10,100,1000);                        % define 2nd range of x values
  x1 = [x1a x1b];                                     % put full x vector together
  top1 = 100*(1 - 0.01*x1.^2).^2 + 0.02*x1.^2;        % numerator of function
  bot1 = (1 - x1.^2).^2 + 0.1*x1.^2;                  % denominator of function
  y1 = sqrt(top1./bot1);                              % desired y(x)
%
%  plot function on a LINEAR axis
  nfig = nfig+1;   figure(nfig)
  plot(x1,y1,'r-','LineWidth',2), grid
  title('#1b:  Linear plot of y(x) vs. x');
  xlabel('x-values'),ylabel('y(x)')
%
%  plot function on a SEMILOG axis
  nfig = nfig+1;   figure(nfig)
  semilogy(x1,y1,'r-','LineWidth',2), grid
  title('#1b:  Semilog plot of y(x) vs. x');
  xlabel('x-values'),ylabel('y(x)')
%
%  evaluate and plot function on a LOGLOG axis
%  NOTE:  If you plan to use a logarithmic scale for the x-values, you should
```

```
%   really use "logspace" instead of "linspace" to set up the x-vector.  Thus
%   we re-evaluate all the parameters here (just to demonstrate the use of logspace).
    x2 = logspace(-2,2,2000);                          % define range of x values
    top2 = 100*(1 - 0.01*x2.^2).^2 + 0.02*x2.^2;       % numerator of function
    bot2 = (1 - x2.^2).^2 + 0.1*x2.^2;                 % denominator of function
    y2 = sqrt(top2./bot2);                             % desired y(x)
%
%   plot function on a LOGLOG axis  (Note: this is really the best view...)
    nfig = nfig+1;   figure(nfig)
    loglog(x2,y2,'r-','LineWidth',2), grid
    title('#1b:  Loglog plot of y(x) vs. x');
    xlabel('x-values'),ylabel('y(x)')
    v2 = axis;   v2(1) = 0.1;   axis(v2);              % sets min x for plot
%
%   end of program
```