John Sigmon
Js85773

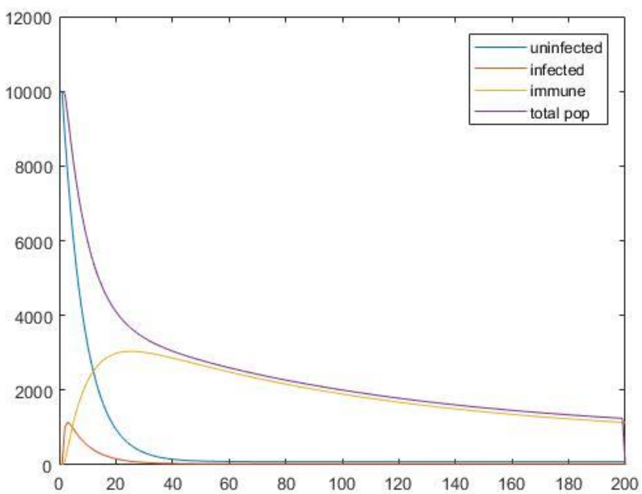
EE 362K

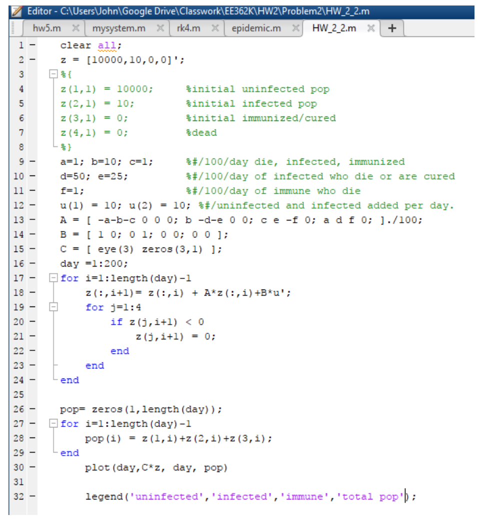
HW 2

1.) I completed the exercises

2.)

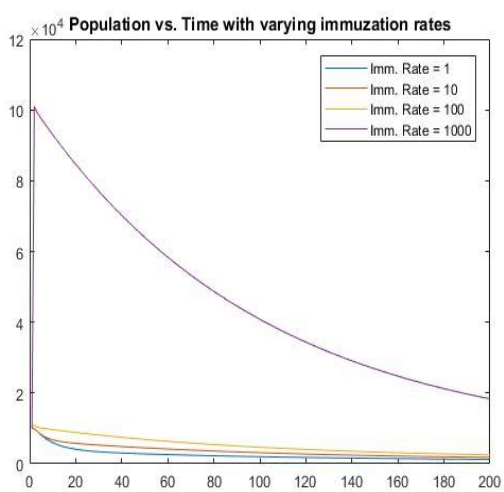
Code and Graph for part 1 (modification of code)

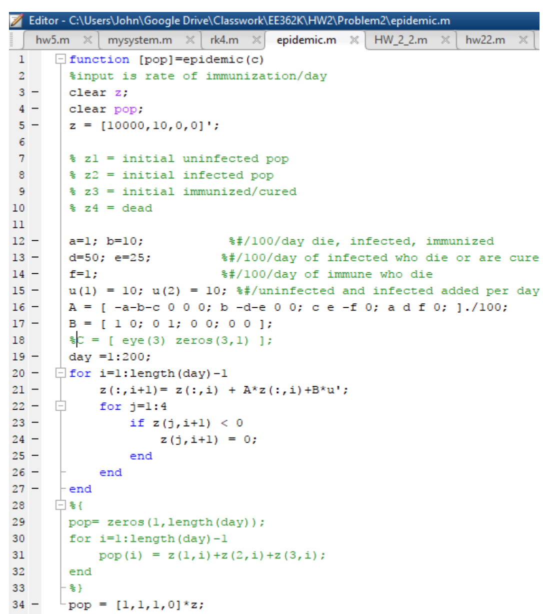





```
clear all;
z = [10000,10,0,0]';
%{
z(1,1) = 10000;     %initial uninfected pop
z(2,1) = 10;        %initial infected pop
z(3,1) = 0;         %initial immunized/cured
z(4,1) = 0;         %dead
%}
a=1; b=10; c=1;     %#/100/day die, infected, immunized
d=50; e=25;         %#/100/day of infected who die or are cured
f=1;                %#/100/day of immune who die
u(1) = 10; u(2) = 10; %#/uninfected and infected added per day.
A = [ -a-b-c 0 0 0; b -d-e 0 0; c e -f 0; a d f 0; ]./100;
B = [ 1 0; 0 1; 0 0; 0 0 ];
C = [ eye(3) zeros(3,1) ];
day =1:200;
for i=1:length(day)-1
    z(:,i+1)= z(:,i) + A*z(:,i)+B*u';
    for j=1:4
        if z(j,i+1) < 0
            z(j,i+1) = 0;
        end
    end
end

pop= zeros(1,length(day));
for i=1:length(day)-1
    pop(i) = z(1,i)+z(2,i)+z(3,i);
end
    plot(day,C*z, day, pop)

    legend('uninfected','infected','immune','total pop');
```

Code and Graph for part 2 (pop vs imm. rates)






```
function [pop]=epidemic(c)
%input is rate of immunization/day
clear z;
clear pop;
z = [10000,10,0,0]';

% z1 = initial uninfected pop
% z2 = initial infected pop
% z3 = initial immunized/cured
% z4 = dead

a=1; b=10;              %#/100/day die, infected, immunized
d=50; e=25;            %#/100/day of infected who die or are cure
f=1;                   %#/100/day of immune who die
u(1) = 10; u(2) = 10; %#/uninfected and infected added per day
A = [ -a-b-c 0 0 0; b -d-e 0 0; c e -f 0; a d f 0; ]./100;
B = [ 1 0; 0 1; 0 0; 0 0 ];
%C = [ eye(3) zeros(3,1) ];
day =1:200;
for i=1:length(day)-1
    z(:,i+1)= z(:,i) + A*z(:,i)+B*u';
    for j=1:4
        if z(j,i+1) < 0
            z(j,i+1) = 0;
        end
    end
end
%{
pop= zeros(1,length(day));
for i=1:length(day)-1
    pop(i) = z(1,i)+z(2,i)+z(3,i);
end
%}
pop = [1,1,1,0]*z;
```

John Sigmon
Js85773

Editor - C:\Users\John\Google Drive\Classwork\EE362K\HW2\Problem2\hw22.m

hw5.m | mysystem.m | rk4.m | epidemic.m | HW_2_2.m | hw22.m

```
%hw 2 problem 2

t = 1:200;
plot(t,epidemic(1),t,epidemic(10),t,epidemic(100),t,epidemic(1000));
title('Population vs. Time with varying immuzation rates')
legend('Imm. Rate = 1', 'Imm. Rate = 10', 'Imm. Rate = 100', 'Imm. Rate = 1000');
```

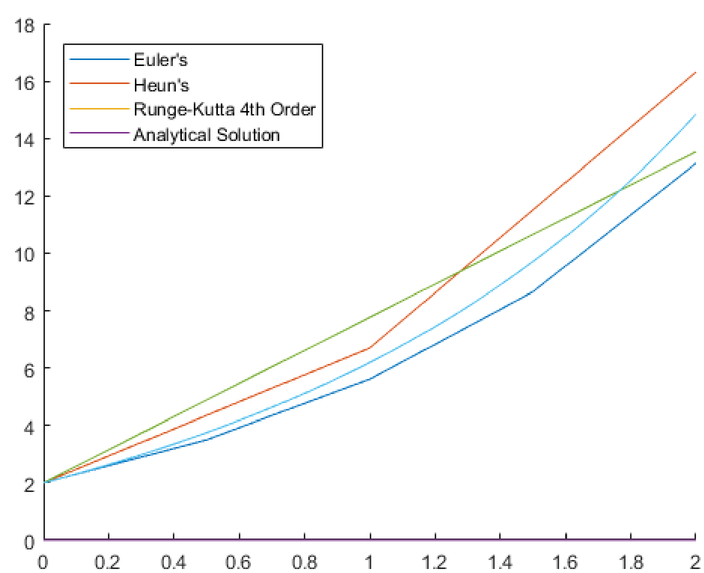
3.)

a.) x(2) = 13.1334

b.) x(2) = 16.3198

c.) x(2) = 13.5388

d.) graph and code



John Sigmon
Js85773

```
function [y] = eulers

% This function solves equations of the type dy/dt = f(y, t)
% Method: y(new) = y(old) + slope * step size


% First define variables
clear t
clear y

h = 0.5;          % Step size
t = 0:h:2;

y = zeros(1, length(t));   % y = a vector
y(1) = 2;        % y(IC) = y0

for i = 1: length(t)-1
    phi = 4*exp(0.8*t(i)) - 0.5 * y(i);
    y(1, i+1) = y(1, i) + phi * h;
end


plot(t,y,'b:'); hold on;
```

```
function [y] = heuns

% This function solves equations of the type dy/dt = f(y, t)
% Method: y(new) = y(old) + slope * step size


% First define variables
clear t; clear y;

h = 1;           % Step size
t = 0:h:2;
y = zeros(1, length(t));   % y = a vector
y(1,1) = 2;        % y(IC) = y0

for i = 1: length(t)-1
    si = 4*exp(0.8*t(i)) - 0.5*y(1,i);        % si is the slope at beg of interval
    y(1, i+1) = y(1,i) + si*h;              % find y(i+1) w/ si
    sf = 4*exp(0.8*(t(i+1))) - 0.5*y(1,i+1);  % estimate final slope
    phi = (si+sf)/2;                        % let slope be avg between i/f
    y(i+1) = y(i) + phi*h;
end

plot(t,y, 'r--'); hold on;
```

```
function [y] = rk4

clear y;
h=2;
t = 0:h:2;
y = zeros(1,length(t));
y(1)=2;              %Initial Condition
F_ty = @(t,y)  (4)*exp(0.8*t) - 0.5*y;

for i=1:length(t)-1
    k1 = F_ty(t(i), y(i));
    k2 = 2 * F_ty(t(i) + 0.5*h, y(i) + 0.5*h*k1);
    k3 = 2 * F_ty(t(i) + 0.5*h, y(i) + 0.5*h*k2);
    k4 = F_ty(t(i) + h, y(i) + h*k3);
    y(i+1) = y(i) + (k1 + k2 + k3 + k4)/6;
end

plot(t,y, 'g-.*'); hold on;
```

John Sigmon
Js85773

```
function [yt]=yt

t = linspace(0,2);
yt = (4/1.3)*(exp(0.8*t)-exp(-0.5*t)) + 2*exp(-0.5*t);

plot(t, yt, 'c');
```

```
function hw3

eulers;
heuns;
rk4;
yt;
legend('show','Location','northwest');
legend('Euler''s','Heun''s','Runge-Kutta 4th Order','Analytical Solution');
hold off;
```

4.)

Something is wrong with these numbers… I went back over code. Ran out of time.

y(2.5) = 18.657 by runge kutta

y(2.5) = 7.9102 by analysis

absolute error = 1.36

```
function [error] = hw4

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% HW question 4 - Error calculation of RK4 analysis of ODE (f_ty) and
% analytical solution (f_t = y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f_ty = @(t,y) (1+t)*sqrt(y);            %%ODE
tf = 2.5;
h = 2.5;
ic = 1;


[t1, y1] = rk4(f_ty,h,tf,ic);              %%RK4 approximation of solution
n = length(t1);
rkans = y1(n);
f_t = @(t) ((t+(t.^2)/2)/2).^2;
ans = f_t(2.5);
error = abs((f_t(2.5) - rkans))/f_t(2.5);
```

John Sigmon
Js85773

```
function [t,y]=rk4(F_ty,h,tf,ic)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Pass in an ODE F_ty and a step size h, this method returns two
% vectors, t and y which you can plot as classical fourth order Runge Kutta
% solutions to your ODE F_ty
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear y;
t = 0:h:tf;                        %Time span of interest
y = zeros(1,length(t));
y(1)=ic;                         %Initial Condition

for i=1:length(t)-1
    k1 = F_ty(t(i), y(i));
    k2 = 2 * F_ty(t(i) + 0.5*h, y(i) + 0.5*h*k1);
    k3 = 2 * F_ty(t(i) + 0.5*h, y(i) + 0.5*h*k2);
    k4 = F_ty(t(i) + h, y(i) + h*k3);
    y(i+1) = y(i) + (k1 + k2 + k3 + k4)/6;
end
```

5.)

HW2 Problem 5

Wednesday, September 20, 2017 6:40 PM

Convert ODE to S.S. Form: $\dddot{x} + 3\ddot{x} + 0.5\dot{x} + 4x = u(t)$

Consider $u(t) = 0$

Then;

$$z_1 = x \qquad \dot{z}_1 = z_2$$
$$z_2 = \dot{x} \longrightarrow \dot{z}_2 = z_3$$
$$z_3 = \ddot{x} \qquad \dot{z}_3 = -4z_1 - 0.5z_2 - 3z_3$$

So;

$$\frac{d\bar{z}}{dt} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -0.5 & -3 \end{bmatrix} \bar{z}$$

Then, we are looking for z_1 ($z_1 = x$) at $t=4$

Running the code below shows $\boxed{z_1 = 0.1373 \times 10^{-3} \text{ for 1 step and, } z_1 = 0.5824 \times 10^{-3} \text{ for 4 steps}}$

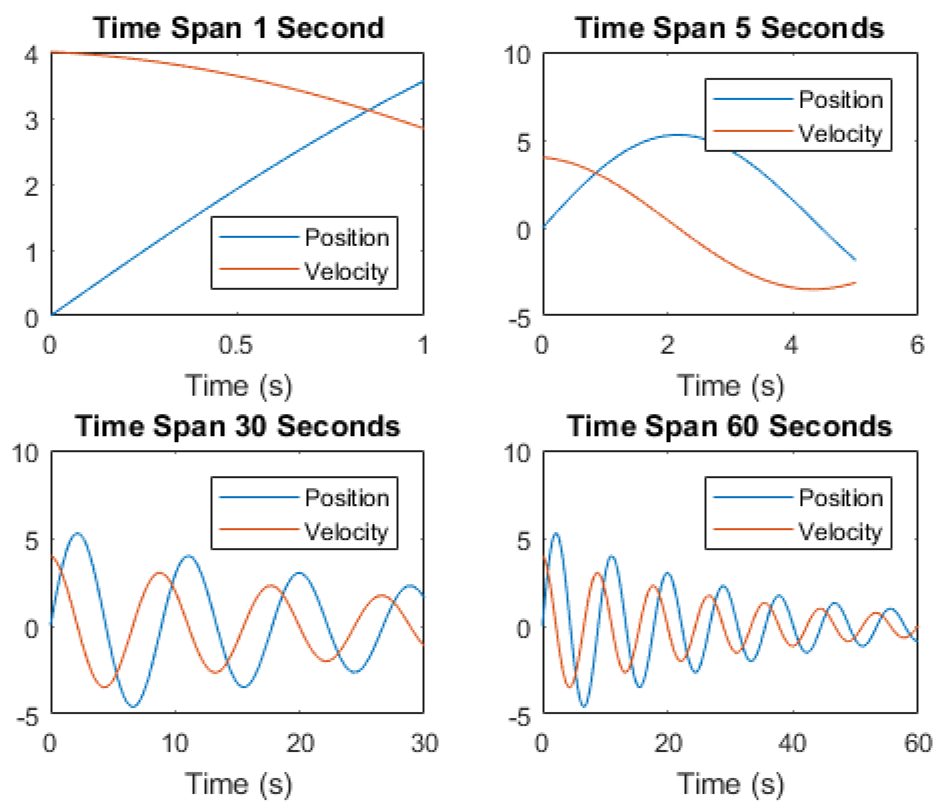
```
clear all

h = 1;
tf = 4;
clear y;
t = 0:h:tf;                    %Time span of interest
y = zeros(3,1,length(t));
y(:,:,1)= [1,1,1]';                      %Initial Condition

dy = @(y) [0,1,0;0,0,1;-4,-0.5,-3]*y(:,:);

%k's become vectors, so

for i=1:length(t)-1
    k1 = dy(y(:,:,i));
    k2 = 2 * dy(y(:,:,i) + 0.5*h*k1);
    k3 = 2 * dy(y(:,:,i) + 0.5*h*k2);
    k4 = dy(y(:,:,i) + h*k3);
    y(:,:,i+1) = y(:,:,i) + (k1 + k2 + k3 + k4)/6;
end
```

John Sigmon
Js85773

```
function zprime=mysystem(t,z)

zprime = [z(2);z(3);-4*z(1) - 0.5*z(2) - 3*z(3)];
%zprime is a column vector
```

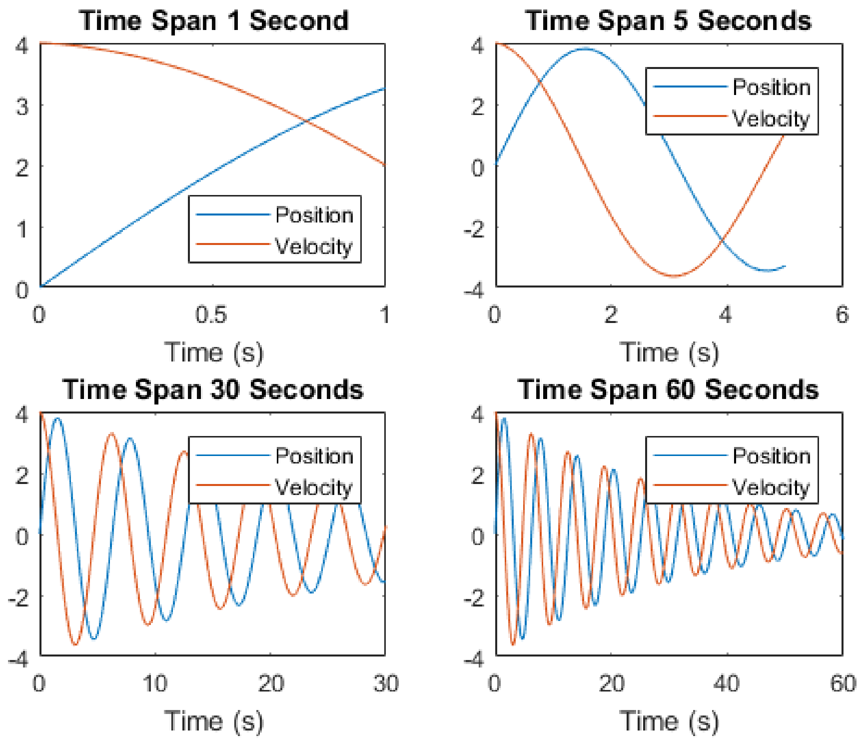
6.)

First graph, k = 2



Second graph, k =4

John Sigmon
Js85773



Code:

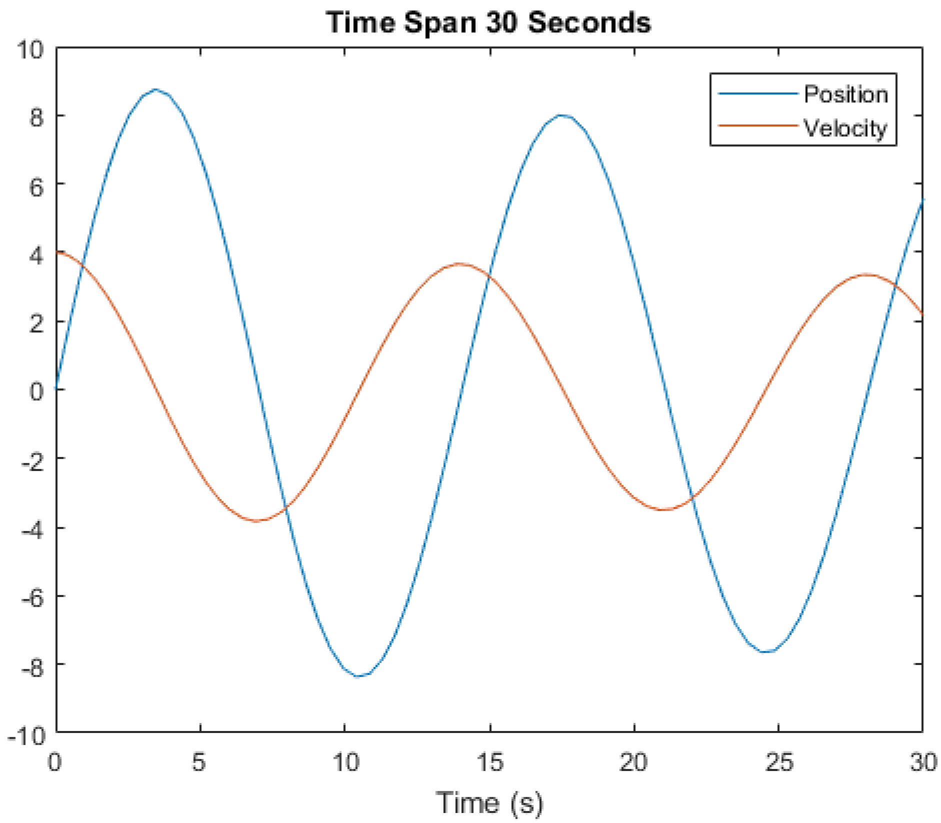
```
function zprime = MSD(t,z)
k = 4;
b = 0.25;
m = 4;
zprime = [z(2); -k/m*z(1)+ -b/m*z(2)];
```

John Sigmon
Js85773

```
function hw6

%first round of subplots for k = 2;
subplot(2,2,1);
[t,y] = ode45(@MSD,[0 1],[0 4]);
plot (t,y(:,1),t,y(:,2))
xlabel('Time (s)')
legend('Position','Velocity', 'Location','southeast')
title('Time Span 1 Second')

clear t; clear y;
subplot(2,2,2);
[t,y] = ode45(@MSD,[0 5],[0 4]);
plot (t,y(:,1),t,y(:,2))
xlabel('Time (s)')
legend('Position','Velocity', 'Location','northeast')
title('Time Span 5 Seconds')

clear t; clear y;
subplot(2,2,3);
[t,y] = ode45(@MSD,[0 30],[0 4]);
plot (t,y(:,1),t,y(:,2))
xlabel('Time (s)')
legend('Position','Velocity', 'Location','northeast')
title('Time Span 30 Seconds')

clear t; clear y;
subplot(2,2,4);
[t,y] = ode45(@MSD,[0 60],[0 4]);
plot (t,y(:,1),t,y(:,2))
xlabel('Time (s)')
legend('Position','Velocity', 'Location','northeast')
title('Time Span 60 Seconds')
```

For the final plot, I changed the mass to 20 kg. As expected it increased the settling time of the system. It also increased the amplitude of the position, and decreased velocity.

John Sigmon
Js85773



7.) Please see attached sheet.

8.)