Numerical Methods – Spring '18

PA #1

Created By: Clarisa Leu-Rodriguez

Dr. Christopher Willett

Programming Assignment #1

Problem 1:

The equation for a damped, non-driven oscillating spring is given by:

$$y = Ae^{-kt}\sin(\omega t + \phi)$$

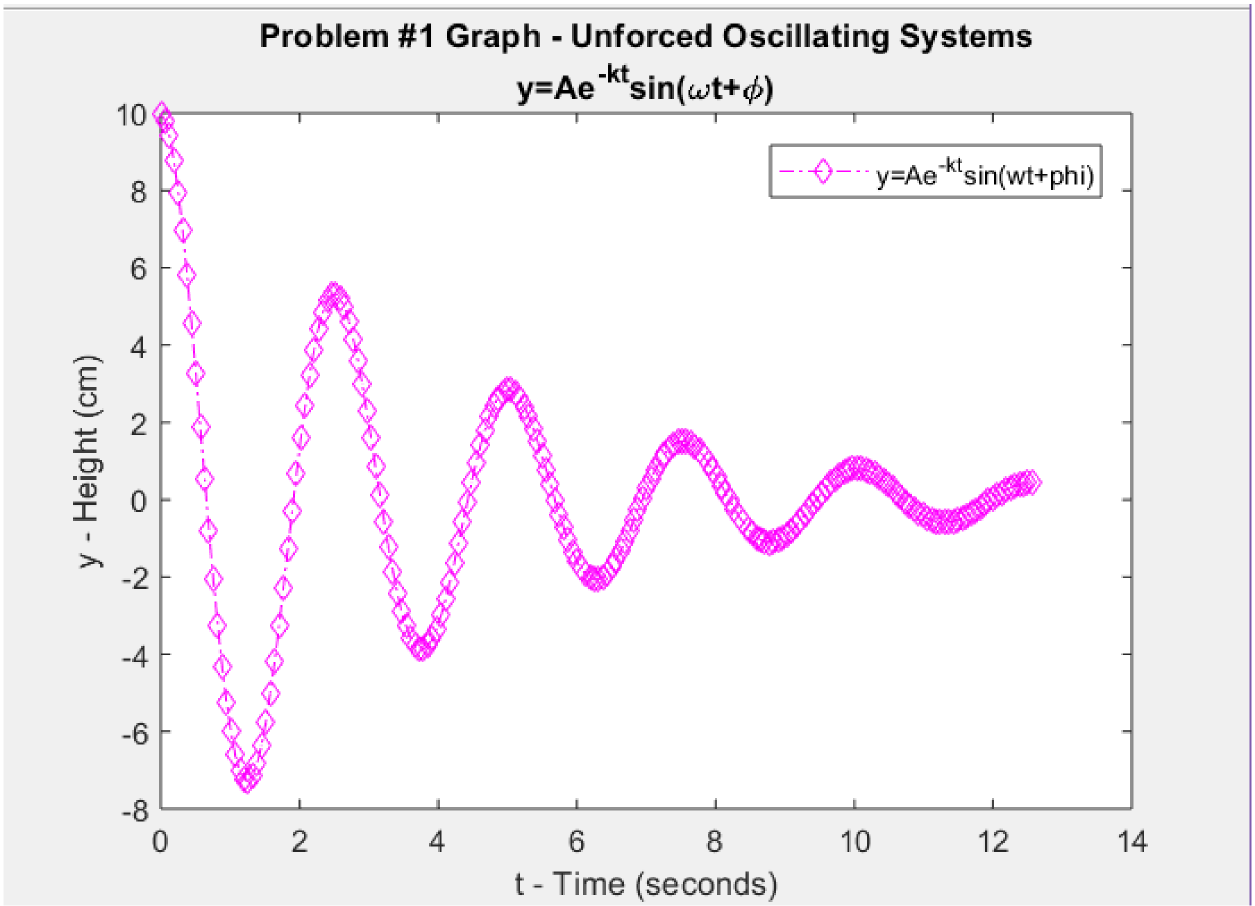
A and k are positive constants. Omega and phi are any constants. Write a function that plots the first five periods of the decay function by letting the user enter values for A, k, ω, and ϕ in order.

My Solution:

leurodriguez1.m

```
%Problem 1
%A function to plot the graph of an unforced oscillating system over time.


function y = leurodriguez1(A,k,w,phi)
%A,k>0.
if A<0 || k<0
    disp('A and k are positive constants. Try again!')
end

if A>0 && k>0

%t=0 to t=10pi/w.
t = linspace(0,10*pi/w,200);
y = A.*exp(-k.*t).*sin(w*t+phi);

%Display
plot(t,y,'-.dm');
title({'Problem #1 Graph - Unforced Oscillating Systems';'y=Ae^{-kt}sin({\omega}t+{\phi})'})
xlabel('t - Time (seconds)')
ylabel('y - Height (cm)')
legend('y=Ae^{-kt}sin(wt+phi)')

end

end
```

leurodriguez1(10,.25,2.5,1.5)



Problem 2:

The basic Newtonian equations of motion for an object launched with an initial velocity of v_0 at an angle of θ radians are given by the parametric equations

$$x = v_0 \cos(\theta)\, t \qquad y = -\frac{1}{2} g t^2 + v_0 \sin(\theta)\, t$$

Write a function that inputs the horizontal distance to a target and the initial velocity of the project and returns the angle (in milliradians) needed to hit the target.

Then, write a script that tests your function by calling it for four separate cases.

My Solution:

Solving the system of equations for θ.

$$\theta = \frac{1}{2} \arcsin(\frac{gx}{v_0^2})$$

leurodriguez1_projectile.m

```
%Problem 2
%A function to find the required angle of projection in milliradians to launch
%a projectile x meters at an initial velocity v0 (meters/second).

function z = leurodriguez1_projectile(x,v0)
%x,v0>0
if v0<0 || x<0
    disp('The initial velocity and distance to travel must be positive. Try again!')
end

if v0>0 && x>0
theta=asin(((x.*9.81)./(v0.^2))).*.5;
z=theta.*10.^(3);
end

end
```

leurodriguez1_projectile_test.m

```
%A script to test problem number 2

disp('Now runninig test for problem 2...')

test1=leurodriguez1_projectile(5,50);
test2=leurodriguez1_projectile(3,20);
test3=leurodriguez1_projectile(300,90);
test4=leurodriguez1_projectile(400,900);

disp('The required angle of trajectory (in mrad) to travel 5m at an initial velocity of 50m/s is:')
disp(test1)
disp('The required angle of trajectory (in mrad) to travel 3m at an initial velocity of 20m/s is:')
disp(test2)
disp('The required angle of trajectory (in mrad) to travel 300m at an initial velocity of 90m/s is:')
disp(test3)
disp('The required angle of trajectory (in mrad) to travel 400m at an initial velocity of 900m/s is:')
disp(test4)
```

output:

```
>> leurodriguez1_projectle_test
Now runninig test for problem 2...
The required angle of trajectory (in mrad) to travel 5m at an initial velocity of 50m/s is:
    9.8106

The required angle of trajectory (in mrad) to travel 3m at an initial velocity of 20m/s is:
   36.8208

The required angle of trajectory (in mrad) to travel 300m at an initial velocity of 90m/s is:
  185.9216

The required angle of trajectory (in mrad) to travel 400m at an initial velocity of 900m/s is:
    2.4222
```

Problem 3:

Write a script that finds the potential energy of five hanging springs with the compiled data:

Force (Newtons)	14	18	8	9	13
x (meters)	0.013	0.020	0.009	0.010	0.012

My Solution:

From Hooke's Law:

- F=kx
- The force of a spring F with spring constant k and expansion or compression length x

The Potential Energy of a Spring:

- U=$\frac{1}{2}kx^2$

Solve k directly & plug into U.

leurodriguez1_potentialenergy.m

```
%A Script to find the potential energy of a spring given the force of the
%spring & the corresponding stretch of the spring


disp('Now running the script for problem 2...')
%Given information for the force of a spring (Newtons)
force=[14;18;8;9;13];
%Given information for corresponding stretch (meters) of spring
xdistance=[0.013;0.020;0.009;0.010;0.012];

%Hooke's Law: F=kx
springConstant=force./xdistance;
%Potential Energy of the corresponding spring (Joules) in table
potentialEnergy=.5.*springConstant.*(xdistance.*xdistance);

%Display
disp('The corresponding potential energies for the spring systems in Joules are:')
disp(potentialEnergy)
```

output:

```
>> leurodriguez1_potentialenergy
Now running the script for problem 2...
The corresponding potential energies for the spring systems in Joules are:
    0.0910
    0.1800
    0.0360
    0.0450
    0.0780
```