PROJECT: SIMPLE PENDULUM

Name: Muhammad Ayyan Tahir

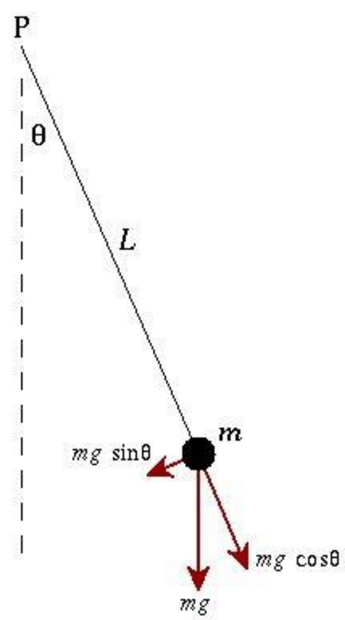
Roll no: 20K-0278 Sec: BCS-1A

Problem statement:

Simulate simple pendulum with small angle and large angles (at least two) approximation for large amplitude) keeping the same lengths. Plot the amplitude (θ) as a function of time on a single graph and explain if any difference occurs.

Solution:

In this simulation a roof is made as the pivot point and a bob is attached at the end of the string making an angle (θ) with the pivot point that is the roof . The bob attached to the string oscillates from left to right in the simulation for 25 seconds with different angles which depend upon the angular velocity. If the velocity increase then the angle from the pivot and the string also increases. Finally a graph of amplitude(θ) vs t is plotted for the movement of the pendulum.



Procedure:

We will use an online platform of GlowScript 3.0 VPython on trinket.io for simulation. Further elements and properties for this simulation are described below.

Animation:

For animation we will use a **while** loop till a certain a value of time at which the pendulum will stop oscillating.

Objects in Simulation:

- A cylinder of suitable radius and height named string attached to the pivot point at an angle.
- A sphere of suitable radius named Bob attached at the end of the string

Variables:

- Bob position
- string position
- pivot,roof
- time (t) initially set at t=0, dt(time interval) set at dt=0.01 ,
- len(length of the pendulum) that is the magnitude of bob.pos-pivot
- cosX(cos(θ))
- theta(angle with vertical direction)
- w (angular velocity) set at w=0.25 ,w=0.5,w=0.75 respectively to increase the angles
- g(acceleration due to gravity set at g=9.8)
- acc(acceleration).

Graphing:

Functions and variables (f1,f2,f3) for the graphs of all three simulations.

Code Block:

The following code was used to carry out the simulation results. Values of w(angular velocity) were changed in order to carry out the simulation for larger angles

```
GlowScript 3.0 VPython

x_ax=curve(pos=[vector(-15,0,0),vector(15,0,0)],color=color.white)
y_ax=curve(pos=[vector(0,-15,0),vector(0,15,0)],color=color.white)

tgraph=graph(title="Amplitude vs time graph",xtitle="Time[seconds]",ytitle="Amplitude[theta]"
f1=gcurve(color=color.blue)
f2=gcurve(color=color.red)
f3=gcurve(color=color.green)

g=9.8  # acceleration due to gravity

bob=sphere(pos=vector(5,2,0),radius=1,color=color.red)
pivot=vector(0,15,0)
roof=box(pos=pivot,size=vector(4,1,1),color=color.white)
string=cylinder(pos=pivot,axis=bob.pos-pivot,radius=0.1,color=color.blue)
t=0 # time
dt=0.01 # time interval
len=mag(bob.pos-pivot) # length of pendulum
cosX=(pivot.y-bob.pos.y)/len # calculation of cos(theta)
theta=acos(cosX) # angle with vertical direction
w=0.25 # angular velocity
while (t<25):
 rate(500) # maximum 500 calculations per second
 acc=-g/len*sin(theta) # updating of angular acceleration
 theta=theta+w*dt # updating of angular position
 w=w+acc*dt # updating of angular velocity
 bob.pos=vector(len*sin(theta),pivot.y-len*cos(theta),0) # calculating position
 string.axis=bob.pos-string.pos # updating other end of rod of pendulum
 t=t+dt # updating time
 f1.plot(t,theta)

#greater angle1
bob=sphere(pos=vector(5,2,0),radius=1,color=color.red)
pivot=vector(0,15,0)
roof=box(pos=pivot,size=vector(4,1,1),color=color.white)
string=cylinder(pos=pivot,axis=bob.pos-pivot,radius=0.1,color=color.blue)
t=0 # time
dt=0.01 # time interval
len=mag(bob.pos-pivot) # length of pendulum
cosX=(pivot.y-bob.pos.y)/len # calculation of cos(theta)
theta=acos(cosX) # angle with vertical direction
w=0.5
while (t<25):
 rate(500) # maximum 500 calculations per second
 acc=-g/len*sin(theta) # updating of angular acceleration
 theta=theta+w*dt # updating of angular position
```

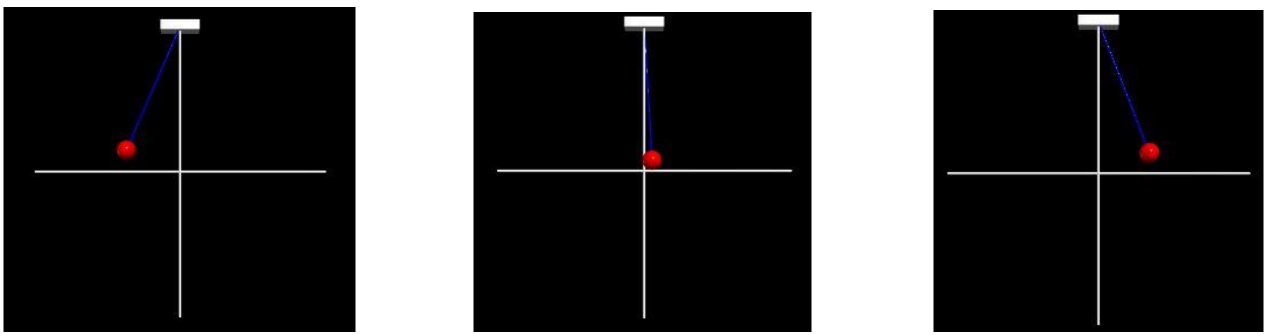
```
 theta=theta+w*dt # updating of angular position
 w=w+acc*dt # updating of angular velocity
 bob.pos=vector(len*sin(theta),pivot.y-len*cos(theta),0) # calculating position
 string.axis=bob.pos-string.pos # updating other end of rod of pendulum
 t=t+dt # updating time
 f2.plot(t,theta)


#greater angle2
bob=sphere(pos=vector(5,2,0),radius=1,color=color.red)
pivot=vector(0,15,0)
roof=box(pos=pivot,size=vector(4,1,1),color=color.white)
string=cylinder(pos=pivot,axis=bob.pos-pivot,radius=0.1,color=color.blue)
t=0 # time
dt=0.01 # time interval
len=mag(bob.pos-pivot) # length of pendulum
cosX=(pivot.y-bob.pos.y)/len # calculation of cos(theta)
theta=acos(cosX) # angle with vertical direction
w=0.75
while (t<25):
 rate(500) # maximum 500 calculations per second
 acc=-g/len*sin(theta) # updating of angular acceleration
 theta=theta+w*dt # updating of angular position
 w=w+acc*dt # updating of angular velocity
 bob.pos=vector(len*sin(theta),pivot.y-len*cos(theta),0) # calculating position
 string.axis=bob.pos-string.pos # updating other end of string of pendulum
 t=t+dt # updating time
 f3.plot(t,theta)
```

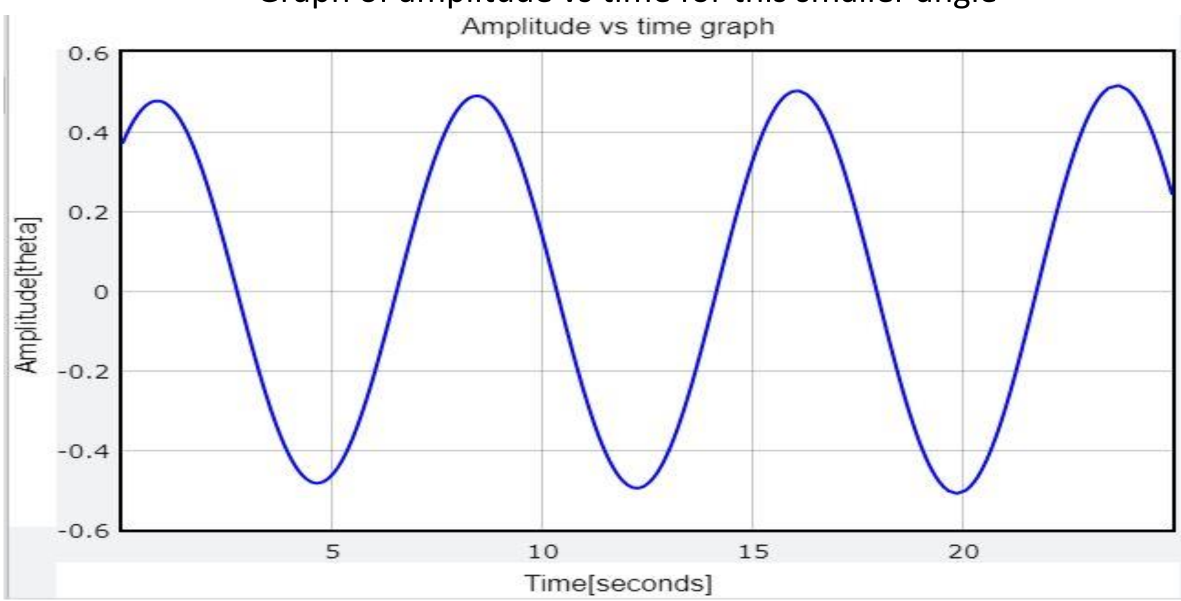
Results:

The figures below show the successful simulation of the said problem. Dt , the time interval was taken as dt =0.01 so that smaller steps were taken by the object and the while loop condition was checked frequently and the loop breaks as soon as the value of t reaches 25.The value of w(angular velocity) was also increased gradually in order to obtain the simulation for larger angles.

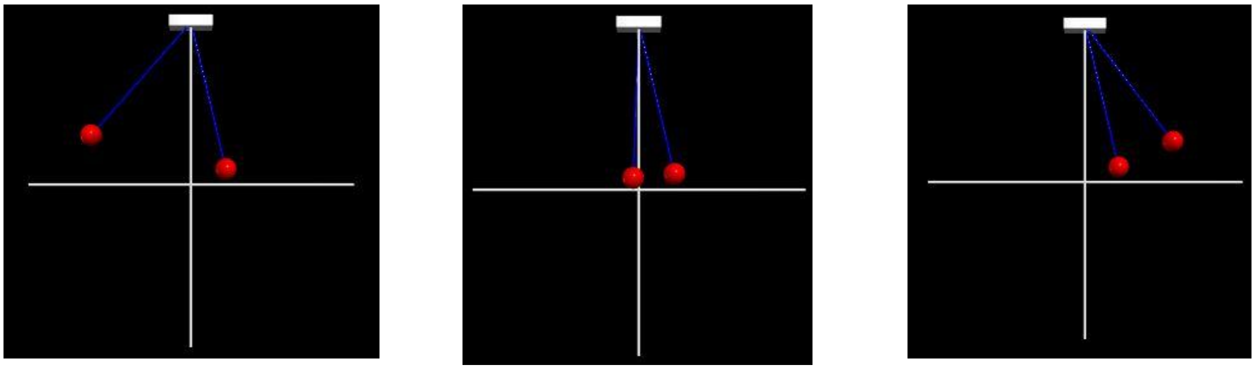
OSCILLATIONS FROM RIGHT TO LEFT AT w=0.25



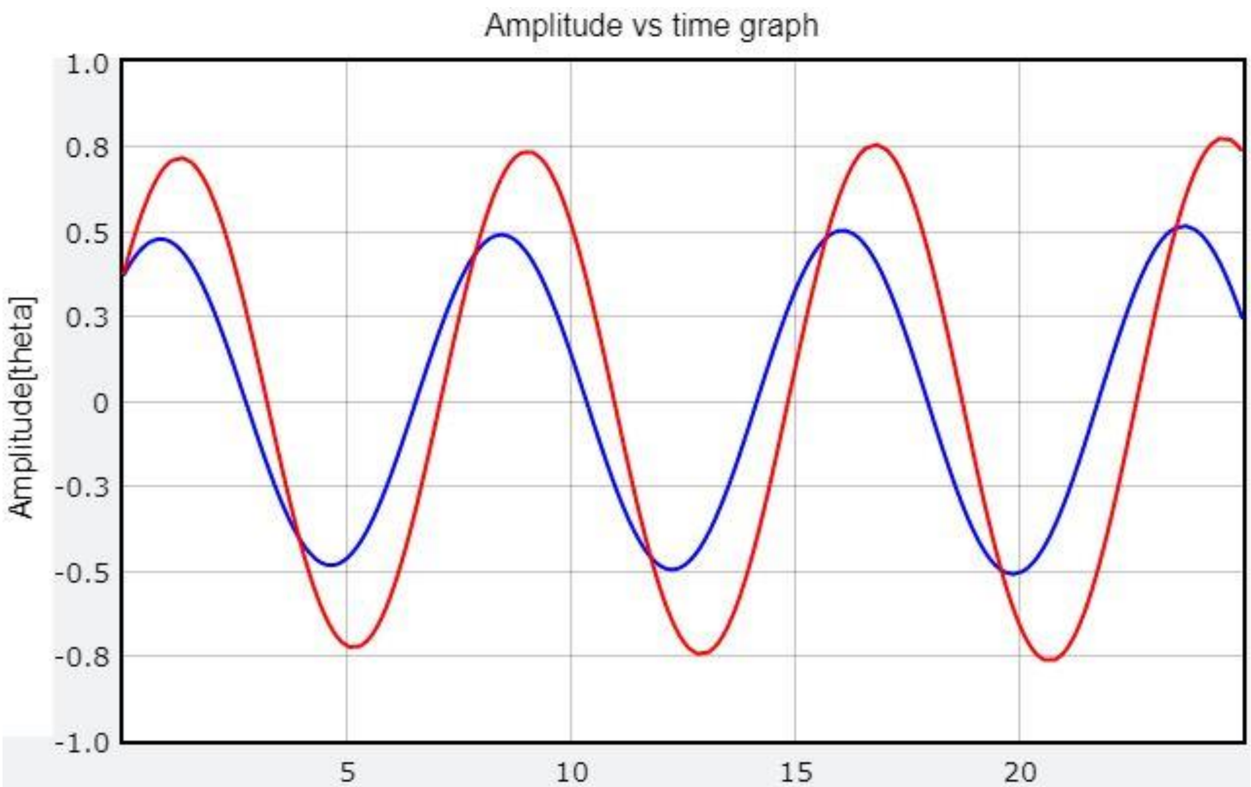
Graph of amplitude vs time for this smaller angle



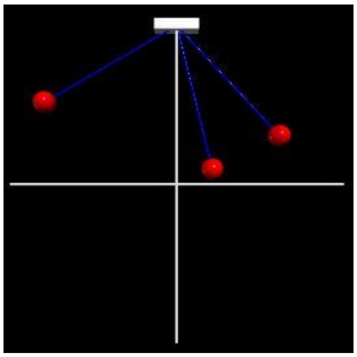
OSCILLATIONS FROM RIGHT TO LEFT AT w=0.5

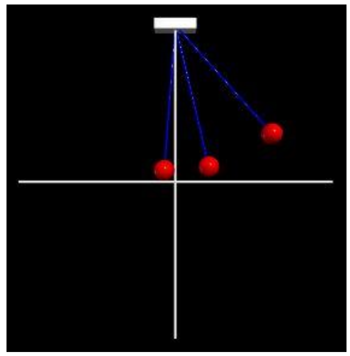


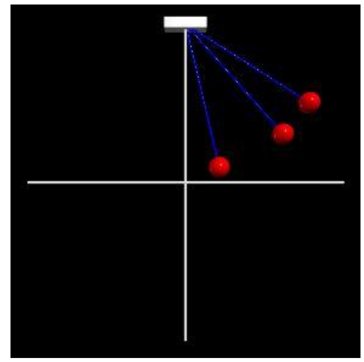
Graph of amplitude vs time for this larger angle (red curve)



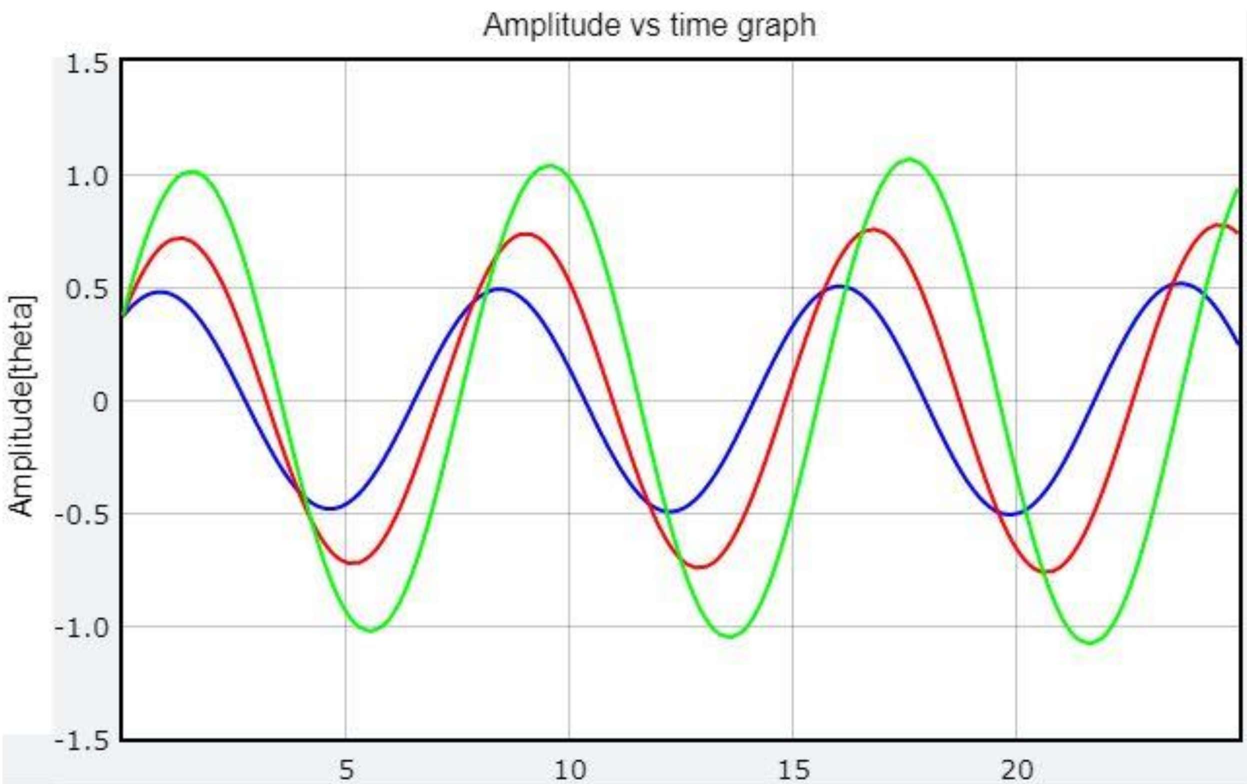
OSCILLATIONS FROM RIGHT TO LEFT AT w=0.75







Graph of amplitude vs time for the largest angle (green curve)



CONCLUSION:

As it can be seen from the above graphs , the amplitude of the third simulation is greater than the the first two and the amplitude for the second simulation is also greater than the first . It can also be seen that there is also a slight increase in amplitude with respect to time in each graph.The frequency also decreases slightly as the angle is increased as it can be seen clearly that the frequency of the first simulation with smaller angle to the pivot (blue curve) is greater than the other two simulations with larger angles.