33-151 Matter & Interactions I Fall 2019

Your Name: ______________________________________ **Section:**______
Instructor Signature: __________________________________
Due Date: Wednesday of Week 7 in lecture.

4.P68. We will consider sending a spacecraft from just above the atmosphere of the Earth to the moon. We will assume that the atmosphere is $50\,km$ high, and the initial speed of the spacecraft is $1.3 \times 10^4\,m/s$ and it has a mass $170\,kg$. (a) Write a program that will let the spacecraft coast to the Moon. Take your initial time step to be one hour, but explore if this makes sense. Also make sure you stop the program before the spacecraft crashes into the surface of the Moon. As part of your program, plot the kinetic energy of the space craft and the potential energy of the spacecraft relative to the Earth and the Moon as a function of distance. (b) Add a calculation of the work done on the spacecraft by the Earth ad the Moon. Add this to your plot from part (a). Explore the effects of your time step on your answers.

Background: Using the VPython shell at the end of this note as the starting point for your program, add the needed code to complete the assignment. The shell assumes that a spacecraft is launched from $50\,km$ above the surface of the Earth with some initial speed. It then coasts to the Moon under the influence of gravity from both the Earth and the Moon. Newton's 2'nd law has already been incorporated to do this this correctly. The program stops when the ship either crashes into the surface of the Moon, or falls back to the surface of the Earth.
Assignment: Do Problem 4.P68 to study energy conservation in sending a ship to the Moon. This will be a powerful check on the accuracy of our numerical integration.

1. Make a graph showing the kinetic energy of the ship as a function of its position. Is the kinetic energy a constant?

2. Show the potential energy on your plot as a function of position.

3. Next add a calculation of the work done on the ship by both the Earth and the Moon.

Recall from lecture that Work is given as the sum of the dot-product of the vector force and the vector displacement. (Note: VPython will do a dot-product of two vectors, $\vec{A}$ and $\vec{B}$ as: `C=dot(A,B)`). There is a common misconception to avoid here. One can calculate the work done on the ship at every step, (the incremental work), or one can sum up all these incremental works and produce the total work done up to the latest step.

4. Add a plot of the both the total work and the incremental work done on the ship as a function of the ship's position. What stays constant?

5. Is there some combination of kinetic energy and the total work that is constant?

6. What happens if you vary the step size in your program?

7. Vary the launch speed and explain the effects on your graphs. You may need to slow your program down for this to work.

8. If the Earth and Moon had exactly the same mass and size, what would you predict the speed of the ship to be when it was $50\,km$ above the surface of the Moon? (Why?)

9. Do you expect that your computer simulation will give this answer? (Explain).

Optional Ideas: What happens if we launch the ship in a direction that is not pointed directly at the Moon? Vary your program to plot the energy as a function of the distance from the Earth in this case.

Shell:

```
from math import *
from vpython import *
#
#  Define needed constants for the program
#
#  Initial speed of the space ship
#
ShipSpeed = 0
ShipMass  = 0
#
G=6.67E-11
# Radius of Earth and Moon
rad_earth = 0
```

```
rad_moon  = 0
#
EarthMass = 0
MoonMass  = 0
#
# Locate the Earth and moon olong the x-axis.
#
earthlocation=vector(0,0,0)
moonlocation=vector(4.0e8,0,0)
#
#
# Set up the displays
#
scene2 = canvas(title='Voyage to the Moon',caption='Animated Display',width=800, height=
     center=0.5*moonlocation, background=color.black)
#
earth = sphere(pos=earthlocation, radius=rad_earth, color=color.blue)
moon  = sphere(pos=moonlocation, radius=rad_moon, color=color.cyan)
#
earth.mass = EarthMass
moon.mass  = MoonMass
#
# Initialize the spaceship
#
shiplocation=vector(earth.radius+50000,0,0)
#
ship = cylinder(pos=shiplocation, axis=vector(5e6,0,0),radius=2e6)
#
ship.mass  = ShipMass
ship.speed = ShipSpeed
#
ship.momentum=vector(ship.mass*ship.speed,0,0)
ship.trail = curve(color=ship.color)
#
# Create graphic for the energy display.
#
energyplot = graph(title='Energy versus Position',xtitle='Ship Position',ytitle='Energy'
                   xmax=moon.pos.x, ymin=-1e10,ymax=1e10)
#
drawKE = gcurve(color=color.cyan,label='Kinetic Energy')
drawPE = gcurve(color=color.blue,label='Potential Energy')
drawTE = gcurve(color=color.magenta,label='Total Energy')
```

```
#
#
t=0
dt= 10
#
#
runit=1
while (runit==1):
#
    rate(100)
#
# Calculate the total force on the ship, and then use this
# in Newton’s 2nd law.
#
    ship.force    = 0
    ship.momentum = 0
    ship.pos      = 0
#
#append a piece to the end of the ship’s trail
#
    ship.trail.append(pos=ship.pos)
#
#  Caluclate and plot the Kinetic and Potential energy
#
    KE =
    PE =
    TE = KE + PE
#
    drawKE.plot(pos=(ship.pos.x,KE))
    drawPE.plot(pos=(ship.pos.x,PE))
    drawTE.plot(pos=(ship.pos.x,TE))
#
    t=t+dt
#
# Check if we fell back to the earth or hit the moon:
#
    if (mag(ship.pos-earth.pos) <= earth.radius ):
        print(’Ship crashed back on the earth at time’,t,’seconds’)
        runit=0
    elif (mag(ship.pos-moon.pos) <= moon.radius/2 ):
        print(’Ship crashed on the moon at time ’,t,’seconds’)
        runit=0
```

```
#
#
print('All done.')
```