Name (print) _____ Section:_____
Instructor Signature _____
Due: Wednesday of week 12 in lecture.

Angular Momentum in the Orbit Program

Problems

Do problem 7.P60 and 7.P61 in your textbook.

7.P60. Earlier in the book, we may have written a program that modeled the Earth orbiting the Sun. We now want to expand on the program to study angular momentum. (a) Start by adding a calculation of the angular momentum, $\vec{L}$ of the Earth about the Sun, and then display this angular momentum vector on your program. In displaying it, place the tail of the vector on the point about which is is calculated. (b) Choose a different point that is in the plane of the orbit, but inside the orbit of the Earth. Compute and plot $\vec{L}$ about this second point as the Earth orbits the Sun. (c) Repeat part (b) for a point that is outside the orbit of the Earth. (d) Modify your program so that the Earth travels in a fairly elliptical orbit about the Sun. Repeat your work in parts (a) through (c) for the elliptical orbit and comment on what you see.

Comment on what you see about the three points for both the circular and elliptical orbits.

7.P61. This problem builds on problem 7.P70, **where the orbit is elliptical**,to look at other quantities that may be conserved in the central force problem. In addition to the angular momentum about the *force center*, there is a second vector which is also conserved in the $1/r^2$ force. This is known as the Runge-Lenz vector. For a central force that can be expressed as

$$\vec{F} = -k\frac{\hat{r}}{r^2},$$

the Runge-Lenz vector, $\vec{A}$, about the force center can be written as:

$$\vec{A} = \vec{p}\times\vec{L} - mk\hat{r}.$$

Given that $k = GmM$, we have for our orbits problem that

$$\vec{A} \;=\; \vec{p} \times \vec{L} - m^2 MG\hat{r}\,.$$

(a) Add a calculation of $\vec{A}$ to your program, and display it (scaled appropriately) in addition to $\vec{L}$. (b) Modify the force law from $1/r^2$ to $1/r^3$ and adjust your program so that it runs. Are the vectors $\vec{L}$ and $\vec{A}$ as computed about the force center still constant?

What do you observe for both $\vec{L}$ and $\vec{A}$ about the force center for the two force laws?

In diplaying the vector $\vec{A}$, display the tail of the vector at the location of the Sun. Note that this vector should be constant. If it is not, try decreasing the size of your time step.

VPython Hints

In order to compute the angular momentum and the Runge-Lenz vector, we will need to compute cross products within VPython. These are accomplished with the operation

```
L = cross(r,p)
```

which sets `L` to be the cross product of `r` and `p`. As a reminder, the other vector operation that we have is the dot product,

```
W = dot(p,r)
```

which sets `W` to be the dot product of `p` and `r`.

Program Shell

```
from math import *
from vpython import *
#
# Distance from Sun to Earth: 149,597,870 km
# Mass of the Sun:     1.989e30 kg
# Radius of the Sun:    695.5e6 m
# Mass of the Earth:   5.972e24 kg
# Radius of the Earth   6.371e6 m
# Distance Sun-Earth    149.578 e9 m
# Length of a year      3.15 e7 seconds
#
```

```
# Set constants
EarthSunDist = 0
SpeedEarth   = 2 * pi * EarthSunDist / 3.15e7
#
# Make the radius of each object large enough to see them
#
earth=sphere(pos=vector(EarthSunDist,0,0),radius=1e7,color=color.green)
sun=sphere(pos=vector(0,0,0),radius=7e9,color=color.yellow)
#
G = 6.67e-11
#
earth.mass = 5.792e24
sun.mass = 1.989e30
#
# We speed this up by 25% to make an eliiptical orbit.
#
earth.speed = 1.25 * SpeedEarth
earth.momentum = earth.mass*vector(0,earth.speed,0)
#
#  We will now define three points about which we will compute the angular momentum.
#  The first is the sun (force center), the second is inside the orbit and the third
#  is outside the orbit.
#
point1 = sun.pos
point2 = 0.5*(earth.pos-sun.pos)
point3 = 2.0*(earth.pos-sun.pos)
#
#  We now create some arrows that will show the force acting on the Earth, the
#  momentum of the Earth, and the angular momentum about each of the three points.
#
forcearrow = arrow(shaftwidth=3e9,color=color.red)
momentumarrow = arrow(pos=earth.pos, axis=earth.momentum, shaftwidth=3e9,color=color.gre
#
angmomarr1 = arrow(pos=point1,shaftwidth=3e9,color=color.cyan)
angmomarr2 = arrow(pos=point2,shaftwidth=3e9,color=color.cyan)
angmomarr3 = arrow(pos=point3,shaftwidth=3e9,color=color.cyan)
#
#  We can also define an arrow for the RungeLenz Vector
#
#rungearr = arrow(pos=point1,shaftwidth=3e9,color=color.magenta)
#
# set the time step to be one day (in seconds)
```

```
#
earth.trail = curve(color=color.cyan) #,retain=250)
#
time = 0
dt = 3600*12
#
# Make a graph of the magnitude of the momentum
# Move the mouse over the graph to explore its interactivity.
# Drag a rectangle in the graph to zoom. Examine the icons at the upper right.
# Click the "Reset axes" icon to restore. Drag along the bottom or left to pan.
#
s='<b>Length of Angular Momentum</b>'
graph(title=s, xtitle='Time', ytitle='Angular Momentum',xmin=0,xmax=6*earth.year,
      ymax=15, ymin=-5)
#
drawit1 = gcurve(color=color.cyan, label='About Sun')
drawit2 = gcurve(color=color.magenta, label='Inside Orbit')
drawit3 = gcurve(color=color.orange, label='Outside Orbit')
#
# We will run until we have returned to the initial position. We will check by
# comparing the polar angle of the objct:
# earth.angle = atan2(earth.pos.y,earth,pos.x)
# earth.oldangle = earth.angle
#
while time < 6*earth.year:
    rate(100)
#
    time = time + dt
#
    earth.force = 0
#
    earth.momentum = 0
    earth.pos      = 0
#
# Compute the angular momentum about each point.
#
    earth.angmom1 = 0
#    earth.angmom2 = 0
#    earth.angmom3 = 0
#
# Compute the Runge Lenz vector.
#
```

```
#    rungelenz = 0
#
    momentumarrow.pos = earth.pos
    momentumarrow.axis = earth.momentum*1e-19
#
# set the arrow direction and length for the three angular momentum vectors.
#
    angmomarr1.axis = earth.angmom1*3e-30
#    angmomarr2.axis = earth.angmom2*3e-30
#    angmomarr3.axis = earth.angmom3*3e-30
#
#    rungearr.axis = rungelenz*3e-59
#
# Update the force vector.
#
    forcearrow.pos  = earth.pos
    forcearrow.axis = earth.force*1e-12
#
# add a point to the trail
#
    earth.trail.append(pos=earth.pos)
#
# Get the length and sign of the three angular momentum vectors for
# graphing.
#
    mom1 = mag(earth.angmom1)*1e-40
    if earth.angmom1.z < 0:
        mom1 = -mom1
#    mom2 = mag(earth.angmom2)*1e-40
#    if earth.angmom2.z < 0:
#        mom2 = -mom2
#    mom3 = mag(earth.angmom3)*1e-40
#    if earth.angmom3.z < 0:
#        mom3 = -mom3
#
# Add the three lengths to the graph.
#
    drawit1.plot(pos=(time,mom1))
#    drawit2.plot(pos=(time,mom2))
#    drawit3.plot(pos=(time,mom3))
```