Rotational Motion, Torque, and Angular Momentum

In the previous chapters you saw how a binary star system or a comet can be simulated in VPython. In this chapter, we will review the motion of a comet, and explore torque and angular momentum. The physics of a binary star system and a Sun-comet system are exactly alike. As a matter of fact, the same physics governs any gravitational interaction (in a classical mechanics sense): a cluster of stars, a moon and its planet, or a cluster of galaxies.

8.1 The Physics

Recall Newton's Law of Universal Gravitation:

$$\vec{F} = -\frac{GMm}{r^2}\hat{r} \tag{8.1}$$

(Note this looks similar to Equation (7.4), the Coulomb's Law force between two charges).

Note that the direction of the position unit vector $\hat{r}$ is always pointing in the exact opposite direction (because of the minus sign) of the force vector.

Angular momentum (symbol $\vec{L}$), is the physical quality that is the rotational equivalent of linear momentum. Remember that linear momentum $\vec{p} = m\vec{v}$. Angular momentum $\vec{L} = I\vec{\omega}$, where I is the *moment of inertia*, and $\vec{\omega}$ we encountered way back in Chapter 3 - it is called the angular velocity. **Torque** (symbol $\vec{\tau}$) can be thought of as the rotational equivalent of force. It is determined from the cross

product $\vec{r}$ and $\vec{F}$

$$\vec{\tau} = \vec{r} \times \vec{F} \tag{8.2}$$

Equation (8.2) shows that the torque is the **cross product** of the position vector $\vec{r}$ and the force applied $\vec{F}$. The cross product is a vector. Remember that the magnitude of the cross product here is

$$|\vec{r}||\vec{F}| \sin\theta,$$

where θ is the angle between $\vec{r}$ and $\vec{F}$. Its direction is perpendicular to both of the two vectors in the cross product. Here, therefore, the torque $\vec{\tau}$ is perpendicular to both $\vec{r}$ and $\vec{F}$.

It is very easy to program the cross product in VPython. To calculate the cross product of $\vec{A}$ and $\vec{B}$, one uses the command

```
cross(A,B)
```

Remember that A and B **must** have been declared to be vectors in VPython. VPython will know automatically that the value assigned to `cross(A,B)` is a vector implicitly.

Just as force is the change of linear momentum with respect to time, see Equation (7.3), torque is the change of angular momentum with respect to time:

$$\vec{\tau} = \frac{d\vec{L}}{dt} \tag{8.3}$$

Furthermore, just as linear momentum is conserved in the absence of an outside force, angular momentum is conserved in the absence of an outside torque. Note that this follows from Equation (8.3).

Look back at Equation (8.1). Recalling that the unit vector $\hat{r}$ is rotated 180° from $\vec{F}$, what is the value of $\vec{\tau}$?

```
GlowScript 2.8 VPython

scene.forward = vector(0,-.3,-1)
scene.height=600
scene.width=1200
scene.range=3e11

G = 6.7e-11 # Newton gravitational constant

sun = sphere(pos=vector(0,0,0), radius=2e10, color=color.yellow,
                make_trail=True, trail_type='points', interval=10)
sun.mass = 2e30

comet = sphere(pos=vector(-1.5e11,0,0), radius=1e10, color=color.white,
                make_trail=True, interval=10)
comet.mass = 1.25e13
comet.p = vector(0,-41.89e3,0)*comet.mass

sun.p = -comet.p
dt = 1e5
t=0
while True:
    rate(2000)
    r = comet.pos - sun.pos
    F = G * sun.mass * comet.mass * r.hat / mag2(r)

    sun.p = sun.p + F*dt
    comet.p = comet.p - F*dt
    sun.pos = sun.pos + (sun.p/sun.mass) * dt
    comet.pos = comet.pos + (comet.p/comet.mass) * dt
    t=t+dt
```

Figure 8.1: Program for Comet Orbiting the Sun

Because $\tau = 0$, then $\vec{L}$ must be conserved.

See the sample program and output in Figure 8.1.

8.2 Exercises

- Add to the code in Figure 8.1 to calculate the torque and plot it (remember it is a vector and you can print the components of a vector!).

Is it pointing in the correct direction?
Show that the program is calculating a torque that is close to the correct answer (it may not, due to limitations in how precisely VPython does calculations).

- Write code to calculate the angular momentum and plot it. Is *it* pointing in the correct direction? Print the angular momentum with every iteration. What do you notice? Does it make sense?

- Why is the Sun not appearing to move?

- Change the parameters of the program to illustrate another scenario (for example, the Earth revolving around the Sun). Discuss what you see.