Experiment 108: The Gravitational Force and Orbits of Comets

Aims

The aim of this experiment is to study some basic celestial mechanics by computing the orbit of a comet about the Sun. Issues to be addressed include the variation of the kinetic and potential energies of the comet with time, the angular momentum of the comet, and the eccentricity and curvature of its orbit.

Simulating the Equations of Motion

We use a Copernican coordinate system and fix the Sun at the origin. We consider only the gravitational force between the comet and the Sun, and neglect all other forces (e.g., forces due to the planets). The force on the comet is

$$\mathbf{F} = -\frac{GMm}{\|\mathbf{r}\|^3}\,\mathbf{r}$$

where $\mathbf{r}$ is the position of the comet, m is its mass, M $(= 1.99 \times 10^{30}$ kg) is the mass of the Sun, and G $(= 6.67 \times 10^{-11}\ \mathrm{m^3/kg\,s^2})$ is the gravitational constant.

The natural units of length and time for this problem are not metres and seconds. Rather, as a unit of distance we will use the astronomical unit (1 AU = 1.496×10^{11} m), which equals the mean Earth-Sun distance. The unit of time will be the AU year (the period of a circular orbit of radius 1 AU). In these units, the product $GM = 4\pi^2\ \mathrm{AU^3/yr^2}$. We take the mass of the comet, m, to be unity; in MKS units the typical mass of a comet is $10^{15\pm3}$ kg.

We assume that the trajectory of the comet lies in the xy-plane, so that we may write the position and velocity vectors as

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ 0 \end{pmatrix}.$$

The equations of motion of these two vectors are

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} \quad \text{and} \quad \frac{d\mathbf{v}}{dt} = \mathbf{a} = \frac{\mathbf{F}}{m} = -\frac{GM}{\|\mathbf{r}\|^3}\,\mathbf{r}\,.$$

A script file called `comet.m` has already been written for you to use. Type in the command `type comet.m` to see this file. This program solves the equations of motion using a higher-order (Runge-Kutta) integration method (the linear approximation, or Euler, method is not very suitable for this problem as it requires extremely small timesteps in order to avoid "blowing up" due to numerical instability).

The program stores the position and velocity vectors of the comet in the arrays `rplot` and `vplot`, respectively. In particular, the position vector after `j-1` timesteps is given by `rplot(j,:)`, and the corresponding velocity vector is given by `vplot(j,:)`. The time after `j-1` timesteps is stored in `tplot(j)`.

Try running `comet.m` for a variety of different input parameters. For example, start with the following inputs, which correspond to a comet starting at the position $\mathbf{r} = (1, 0, 0)$ AU with velocity $\mathbf{v} = (0, 4, 0)$ AU/yr:

```
Enter initial radial distance (AU): 1

Enter initial tangential velocity (AU/yr):  4

Enter number of steps:  500

Enter time step (yr):  0.001
```

You will see the comet's trajectory drawn in the xy-plane. Note its elliptical shape, with the Sun at one focus of the ellipse.

(1) Try different values of the initial tangential velocity. Describe what happens to the orbit as this initial velocity is made larger or smaller. Note that for small values of the initial tangential velocity you may need to decrease the size of the timestep, otherwise numerical instability causes the trajectory to deviate from an ellipse. For large values of the initial velocity you may need to use more timesteps to see the full shape of the trajectory.

Kinetic Energy and Potential Energy

The kinetic energy E_K and gravitational potential energy E_P of the comet are given by

$$E_K = \frac{1}{2} m \|\mathbf{v}\|^2 \quad \text{and} \quad E_P = -\frac{GMm}{\|\mathbf{r}\|}.$$

Note that `comet.m` defines the constant `GM = 4*pi^2` and defines `mass = 1` to be the mass of the comet.

(2) Write a script file (call it `kepler.m`) that computes and plots E_K, E_P, and their sum $E_K + E_P$ versus time for a particular trajectory (i.e., for a particular `rplot` and `vplot` generated by `comet.m`). Describe in words what your results illustrate.

Angular Momentum

The angular momentum of the comet about the Sun is

$$\mathbf{L} = m\,(\mathbf{r} \times \mathbf{v}).$$

(3) Using the product rule and the properties of the cross product and the gravitational force, show that the angular momentum of the comet is constant with time (i.e., calculate $d\mathbf{L}/dt$).

(4) Add a line to your script file `kepler.m` which computes the angular momentum vector of the comet at each timestep. Hence, confirm for several different trajectories that $\mathbf{L}$ is a constant, and give its value for each case.

(5) **Optional**: Referring to Figure 1 below, the area swept out by the position vector in the (small) time interval Δt is approximated by $\Delta A = \frac{1}{2} \|\mathbf{r}(t) \times \mathbf{r}(t + \Delta t)\|$. (Why?) Using this, show that the rate of change of the area, $\Delta A/\Delta t$, is a constant. This amounts to Kepler's Second Law of Planetary Motion – the line joining the Sun and a planet (or orbiting comet) sweeps out equal areas in equal times.

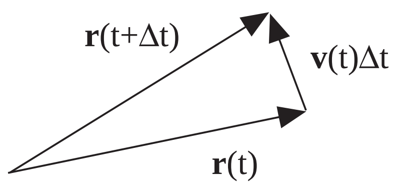

Figure 1: Change in position vector in time interval Δt.

Eccentricity of the Orbit

The eccentricity of an elliptical orbit is given by the formula

$$e = \sqrt{1 - \frac{b^2}{a^2}},$$

where a and b are the semimajor and semiminor axes of the ellipse, respectively (see Figure 2).

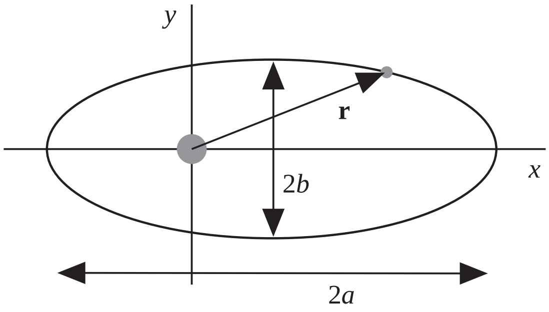

Figure 2: Elliptical orbit about the Sun.

Consider the following combinations of initial position and velocity vectors:

(a) $\mathbf{r}(t=0) = (1,0,0)$ AU, $\mathbf{v}(t=0) = (0,3,0)$ AU/yr

(b) $\mathbf{r}(t=0) = (1,0,0)$ AU, $\mathbf{v}(t=0) = (0,4,0)$ AU/yr

(c) $\mathbf{r}(t=0) = (1,0,0)$ AU, $\mathbf{v}(t=0) = (0,5,0)$ AU/yr

(6) Run `comet.m` with these inputs and, by examining `rplot`, determine a and b for each case. Hence, calculate the eccentricity e for each orbit. Also compute the value of $h = \|\mathbf{r} \times \mathbf{v}\| = \|\mathbf{L}/m\|$ for each case (by running `kepler.m`). You will use e and h below.

Polar Coordinates

Another way of representing the (two-dimensional) orbit is in terms of polar coordinates $r = \|\mathbf{r}\| = \sqrt{x^2+y^2}$ and $\theta = \tan^{-1}(y/x)$.

(7) Add a line to your script file `kepler.m` which computes the angle θ at each point along the trajectory. Use the Matlab function `atan2(y,x)` for the inverse tangent.

You can now use the command `polar(theta,rlength)` to generate a polar plot of a particular orbit. Try this for the three cases specified above.

Using calculus and geometry, one can show theoretically that the orbit of the comet in polar coordinates is in fact given by the expression

$$r = \frac{h^2/GM}{1 - e\cos(\theta)},$$

which describes an ellipse of eccentricity e with one focus at the origin (i.e., at the Sun). This, of course, is Kepler's First Law of Planetary Motion.

(8) For the three sets of initial conditions specified above, use the following commands to superimpose the simulated and theoretical orbits on a polar graph. For the theoretical orbits, use the values of e and h that you calculated in question (6) (you will need to define the variables `e` and `h`).

```
kepler
polar(theta,rlength)
hold on
polar(-pi:pi/20:pi,(h^2/GM)./(1-e*cos(-pi:pi/20:pi)),'*')
hold off
```

Curvature

The curvature of a curve at a given point is a measure of how quickly the curve changes direction at that point. Formally, it is defined as the magnitude of the rate of change of the unit tangent vector to the curve with respect to arc length, i.e.,

$$\kappa = \left\| \frac{d\mathbf{T}}{ds} \right\|, \quad \text{where} \quad \mathbf{T} = \frac{d\mathbf{r}/dt}{\|d\mathbf{r}/dt\|} = \frac{\mathbf{v}}{\|\mathbf{v}\|},$$

and s is the arc length. For a small timestep Δt, an approximation to the curvature is given by

$$\kappa(t) = \frac{\|\mathbf{T}(t+\Delta t) - \mathbf{T}(t)\|}{\|\mathbf{r}(t+\Delta t) - \mathbf{r}(t)\|}.$$

(9) Add a `for` loop to your script file `kepler.m` which calculates an estimate for the curvature κ at each point along the trajectory according to the approximate formula given above. For initial conditions (a) and (b) given earlier, plot the curvature versus time. Explain what you see. (Note: You will need to write `plot(tplot(1:nStep-1),kappa)` to plot the curvature versus time.)

(10) Work out the maximum and minimum values of the curvature, κ_{max} and κ_{min}, for both cases, and identify the points (give x and y coordinates) on the trajectory where they occur. On xy plots of each trajectory, sketch (by hand) circles of radii $1/\kappa_{\text{max}}$ and $1/\kappa_{\text{min}}$ which touch the trajectory at the relevant points