Day 5 Activities: Scientific Programming

Anna L. Rosen

Background

The gravitational force on a satellite with mass m is

$$\vec{F} = \frac{GMm}{r^3}\vec{r} \tag{1}$$

where M is the mass of the sun, G is the Newtonian gravitational constant, and $\vec{r}$ is the satellite position vector pointing from the sun to the satellite. From Newton's second law of the equations of motion the motion of the satellite is given by

$$\begin{aligned} \frac{d^2x}{dt^2} &= \tfrac{GM}{r^3}x \qquad (2) \\ \frac{d^2y}{dt^2} &= \tfrac{GM}{r^3}y \qquad (3) \end{aligned}$$

where $r = \sqrt{x^2 + y^2}$.

The solutions are elliptical orbits with the Earth at one focus of the ellipse. The period of the orbit T is related to the semi-major axis a of the ellipse:

$$T^2 = \frac{4\pi^2}{GM}a^3 \tag{4}$$

If we use a system of units where we measure time in years, distance in AU (1 AU $= 1.50 \times 10^{13}$ cm, and mass in units of $M_\odot = 1.99 \times 10^{33}$ g then

$$G = 4\pi\,\mathrm{AU}^3\,\mathrm{M}_\odot\,\mathrm{yr}^{-2} \tag{5}$$

and for the solar system Kepler's third law becomes $T^2 = a^3$. Use this convenient system of units in all programs for this activity.

Activity 1: Solving ODEs with Euler's Method

Write a program which solves the orbit equation (equation 1) using Euler's method.

1. Assume that the central mass is at the origin.
2. Assume that the initial velocity is perpendicular to the x axis in the positive y direction ($v_{x,0} = 0,\ v_{y,0} = v_0$. Prompt the user to input the initial velocity (units of AU/year), distance of the satellite (units of AU), time step size, and the ending time for the computation.
3. Plot the orbit and the energy per unit mass

$$\frac{E}{m} = \frac{1}{2}(v_{\mathrm{x}}^2 + (v_{\mathrm{y}}^2) - \frac{GM}{x^2 + y^2} \tag{6}$$

   Note that energy should be conserved. Do you see that it isn't in some cases?

4. Try this for: (1) $r = 1.0$ AU, $v = 6.3$ AU/yr and time steps of $\tau = 0.01$ yr; (2) $r = 1.0$ AU, $v = 4.0$ AU/yr and time steps of $\tau = 0.01$ yr; and try other initial conditions.
5. By using Euler's method, do you find that energy is conserved for all initial conditions you tried? What are the limitations of this method?

Activity 2: Solving ODEs with Scipy's ODEINT

Repeat activity 1 but instead of solving the Orbit problem with Euler's method use the scipy.integrate routine odeint. You will have to import this function to your program with the command:

```
from scipy.integrate import odeint
```

Run your program with the same initial conditions outlined in activity 1 and compare your new results with your results from activity 1. Do you see any differences?