V. ORBITAL MECHANICS

Physics 580, SDSU. Due 11:59 pm, Friday, November 25, 2011

In this assignment you will model the orbits of two celestial objects (two planets or a planet and a moon) about a primary (the Sun). To simplify things, we will restrict motion to the xy plane. The equations of motion that arise from Newton's Law of Gravitation can be written as eight coupled first-order ordinary differential equations. You will integrate these equations using the 4th order Runge Kutta method.

A. Basic Project (worth 75%)

For the basic project, consider just one planet in orbit around a primary. We restrict the planet to the xy plane. The independent variable is time t, and the dependent variables are the position (x_1, y_1) and velocity (v_{x1}, v_{y1}). Therefore you will have to write a Runge-Kutta routine for four dependent variables. You can find the equations of motion below.

For simplicity, let Newton's constant, $G = 1$. You will have to supply the mass, M, of the primary and the mass, m_1, of the planet. For this project, I recommend passing these masses to the functions via common block statements ("global variables"), *e.g.*,

```
real M, m1, m2
common/masses/M,m1,m2
```

These lines must appear both in the main program (or wherever you ask for the masses) and in the routines that use them. In Fortran 77 the common block acts like a global variable. (In Fortran 90 you can use a module for global variables, which is much better programming practice.) In C, you declare a global variable by declaring the variable in the preamble.

In addition, it is suggested that you keep track of the energy of the system as it evolves. Although Runge-Kutta is very good, if the time step is too large, the orbits can go haywire. A clear signal is the total energy, which ought to be nearly constant. (Although more clever algorithms will exploit energy conservation, we will not use that here.)

To test your code:

1. Set the primary mass, $M = 0$. In this case there is no gravitational attraction and the orbit should be a straight line.

2. Try the following parameters, $M = m_1 = 1$, with initial conditions $(x_1, y_1) = (1.0, 0.0)$ and $(v_{x1}, v_{y1}) = (0.0, 1.0)$. Try different time steps until you find that the energy is roughly constant in time. The orbits should be circular.

3. As you increase or decrease the initial position, you should find the orbit is elliptical, but maintains a closed orbit. In testing the program, it may be useful to create an input file which can be easily changed with a simple edit and then you can pipe the input file into your program.

Your program should as for the masses of the primary and the planet, the initial position, initial velocity, size of time step, and the number of time steps to take. It should print out to a file with the location of the planet at each step. Also print, in a separate file, the energy of the system as a function of time.

Provide the graphs of two orbits, one circular and one elliptical. On each graph, provide the initial conditions and the size of your time step.

B. Advanced Project (worth 100%)

Generalize your code from the basic project to work with two planets. This will require eight dependent variables.

Suggestions to test your code:

1. Set $M = m_2 = 0$. The solution should be a straight line.

2. Set $M > 0$, but keep $m_2 = 0$. This should replicate the results of the basic project because the second planet has no gravitational effect.

Your program should ask for the three masses, M, m_1, and m_2, the initial positions and velocities, the time step, and the number of time steps to take. Your output files should have the orbits of the two planets and a separate file for the energy as a function of time.

Provide two graphs as your result. Each graph should present the orbit of both planets. One graph has the two planets well separated and orbiting the primary. The second graph should have one planet and the other orbiting the planet as a moon.

C. Equations of motion

Consider the motion of two planets with masses m_1 and m_2 and positions (x_1, y_1) and (x_2, y_2) about a stationary primary of mass M. For simplicity, we will approximate the primary as being stationary without dealing with the reduced mass for the other objects.

Newton's laws of motion are

$$m_1 \frac{d^2}{dt^2}\vec{\mathbf{x}}_1 = -\frac{GMm_1}{r_1^3}\vec{\mathbf{x}}_1 - \frac{Gm_1m_2}{r_{12}^3}(\vec{\mathbf{x}}_1 - \vec{\mathbf{x}}_2) \tag{7}$$

$$m_2 \frac{d^2}{dt^2}\vec{\mathbf{x}}_2 = -\frac{GMm_2}{r_2^3}\vec{\mathbf{x}}_2 - \frac{Gm_1m_2}{r_{12}^3}(\vec{\mathbf{x}}_2 - \vec{\mathbf{x}}_1) \tag{8}$$

with $r_i = \sqrt{x_i^2 + y_i^2}$ and $r_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. For Runge-Kutta integration, we introduce velocities to change things into a first order differential equation, so that the velocities are defined as the derivatives of the positions and

$$\frac{dv_{x1}}{dt} = -\frac{GM}{r_1^3}x_1 - \frac{Gm_2}{r_{12}^3}(x_1 - x_2) \tag{9}$$

$$\frac{dv_{y1}}{dt} = -\frac{GM}{r_1^3}y_1 - \frac{Gm_2}{r_{12}^3}(y_1 - y_2) \tag{10}$$

$$\frac{dv_{x2}}{dt} = -\frac{GM}{r_2^3}x_2 - \frac{Gm_1}{r_{12}^3}(x_2 - x_1) \tag{11}$$

$$\frac{dv_{y2}}{dt} = -\frac{GM}{r_2^3}y_2 - \frac{Gm_1}{r_{12}^3}(y_2 - y_1) \tag{12}$$

In addition, the total energy, which ought to be conserved is

$$E = \frac{m_1}{2}(v_{x1}^2 + v_{y1}^2) + \frac{m_2}{2}(v_{x2}^2 + v_{y2}^2) - \frac{GMm_1}{r_1} - \frac{GMm_2}{r_2} - \frac{Gm_1m_2}{r_{12}} \tag{13}$$