Physics 218 – Spring, '07 - Assignment #3 – "Apollo 13" or "The Moon Shot"

1) For the last assignment involving ordinary differential equations, you will write a program in IDL to simulate the flight of Apollo 13 around the Moon. Recall that Apollo 13 was a near tragedy, as despite the fact that an oxygen tank exploded not too far from the Earth, the laws of physics condemned the three men to endure a flight around the Moon with dwindling supplies of oxygen to breathe and energy to keep warm, before they could return to Earth. Part of your assignment is to determine how long such a journey takes.

Product 3.1: First, write a program in IDL that implements the equations describing the circular restricted three body problem in a co-rotating coordinate system using the LSODE routine. The equations, the origins of which were described in class, are (in non-dimensional units):

$$\frac{dx}{dt} = v_x$$

$$\frac{dv_x}{dt} = \frac{\mu(1-\mu-x)}{((1-\mu-x)^2+y^2)^{3/2}} + \frac{(1-\mu)(-\mu-x)}{((-\mu-x)^2+y^2)^{3/2}} + x + 2v_y$$

$$\frac{dy}{dt} = v_y$$

$$\frac{dv_y}{dt} = \frac{-\mu y}{((1-\mu-x)^2+y^2)^{3/2}} - \frac{(1-\mu)y}{((-\mu-x)^2+y^2)^{3/2}} + y - 2v_x$$

In these equations, all masses are measured in units of $M = M_1 + M_2$, where M_1 *and* M_2 are the masses of the Earth and the Moon respectively (given below). All times are measured in units of (the period of the Moon's orbit/2π). All distances are measured in units of the distance between the centers of the Earth and the Moon (given below). The constant $\mu = M_2/M$.

Note that despite the fact that LSODE appears as an IDL routine, it is in fact a freely available program about which there is much documentation on the Web if you want to learn more about it. However, the IDL documentation should be enough to describe its use.

Test your program using some special cases: First, for the case $\mu = 1$ (all the mass is in the Earth) and with the centrifugal and Coriolus terms commented out to see that you get what you expect for a single gravitating body. You should get circular orbits for initial conditions $x = r, v_x = 0, y = 0, v_y = 1/\sqrt{r}$. You should see elliptical orbits for moderate changes of these initial conditions. Now put the centrifugal and Coriolus terms back in. You should now get circular orbits for initial conditions $x = r,$

$v_x = 0, y = 0, v_y = \left(1/\sqrt{r}\right) - r.$ Change these conditions somewhat to produce elliptical orbits and interpret what you see. As a final test, repeat the above two tests but now for the case $m = 0$ (all the mass is in the Moon). (It will be of enormous help to you if you draw pictures of the positions of the Earth, Moon, and spacecraft in these two limits.) You will need to alter the initial conditions appropriately to get circular orbits with and without the centrifugal and Coriolus terms in the equations.

2) At this point you should have some confidence in your code. Now is the time to simulate the Apollo 13 mission. Here are some numbers:

Mass of Earth: 6.0×10^{24} kg Mass of Moon: 7.4×10^{22} kg
Radius of Earth: 6.36×10^{3} km Radius of Moon: 1.74×10^{3} km
Orbital period of the Moon: 27.3 days
Distance from center of the Earth to center of the Moon is: 3.84×10^{5} km.
Typical *altitude* (height above Earth's surface) of a near Earth orbit: 300 km.

Product 3.2: Use your program from part 1 to attempt to find a figure-eight orbit that starts from near-Earth orbit and loops around the Moon. When you get close (I was never able to do a perfect one!) create a plot of the orbit along with circles (to scale) representing the Earth and the Moon and submit it as part of your report. Determine the period of your figure-eight orbit (in non-dimensional units). Now "unscale" this non-dimensional period to find the orbit's period in days. Compare your result with what you know about the time it took the real Apollo 13 mission to return to Earth.

Project ideas:

1. There are chaotic orbits in the restricted three body problem. Map out some of them. It might help to do some web research to get ideas on where to look.

2. Use the GUI commands of IDL to make an interactive game out of this problem. Sell it for $100M and donate $2M back to Kenyon's Scientific Computing program. *("Extra credit??? We don't need no stinking extra credit!")*