PHYS580 Lab 6, Sep 26, 2019

Assignment:

1. Use the provided starter program *hyperion.m* (or your own equivalent program) to study the motion of Hyperion, one of Saturn's moons, in the dumbbell model discussed in class. First, observe and display the orbital motion and the motion of the dumbbell axis. In particular, try different initial conditions for the Hyperion center of mass location and velocity, and the dumbbell axis orientation and angular velocity. Study both the case of a hypothetical circular orbit as well as a few (2-3) elliptic orbits with different eccentricities. Make sure to include the real orbit of Hyperion with perihelion $a(1\text{-}e) \approx 1.3\times10^6$ km $\equiv 1$ HU ("Hyperion unit") and eccentricity $e = 0.123$. How would you characterize the nature of the spinning motion of Hyperion, based on your simulations? Does the kind of spinning motion depend on the type of orbit, and if yes, how?

   [Note 1] With length and time units chosen such that the perihelion distance is 1 HU and the orbital period for a hypothetical circular orbit is 1 Hyr ("Hyperion year"), the speed along that hypothetical circular orbit is $v = 2\pi$ [HU/Hyr]. For an elliptic orbit instead, the velocity at perihelion is then
   $v_{\max} = 2\pi\sqrt{\dfrac{1+e}{a(1-e)}} = 2\pi\sqrt{1+e}$ [HU/Hyr].

2. Now modify *hyperion.m* (or create your own program) to study the "butterfly effect", i.e., follow the evolution from two slightly different initial conditions for the dumbbell axis. Track both the angular orientation difference and angular velocity difference for the same circular and elliptic orbits as in (1), including the real Hyperion orbit. Use the results to further substantiate the conclusions you reached in (1), and calculate the Lyapunov exponents for each case.

   [Note 2] In order to extract Lyapunov exponents, I suggest that you first pick the local maxima of the time evolution series of the angle and angular velocity differences, and then do a linear, least-squares fit **on a semi-log scale** to (some of) those maxima. You will need to pick the range of data points that gives the most reasonable fits (rather than fit to all the maxima). Thus, it will be best to save those data so that you can repeat the fits for different range choices. You can either write the data to text files, and use separate Matlab scripts (or even different utilities, such as *gnuplot*) to do the fitting. Or, you could apply the fits on Matlab's command line to the arrays of data present at the end of your program execution. For the latter approach, you may find Matlab's *polyfit* function useful.

   [Note 3] For saving data into text files and reading it back, the Matlab functions *dlmwrite* and *dlmread* are probably the easiest solution. Text files are also useful if you prefer another utility (or even a program in another language) to do the fits and extract Lyapunov coefficients..

   [Note 4] Make sure to confirm that the Lyapunov exponents you obtained are reliable and not artifacts of numerical error accumulation. You can get an idea of numerical errors by improving your calculation (either by making your time step smaller or by going to a higher order approximation) and checking how the results change (or not change).