ASTR450 Homework # 8
Due Thursday, April 9

1. Use the "Lagrange Point Explorer" on the class webpage to investigate orbits near the L1, L2, and L3 Lagrange points. Set the velocity to zero and consider small displacements in the x, y, and z directions. Make a table of your findings and work out a rule for what types of orbits you get as a function of displacement for each Lagrange Point. Do you find any stable orbits?
b) Repeat part a) with the displacement equal to zero and considering small velocity increments (V=0.1 km/s).
c) Now consider non-zero displacements in both position and velocity. Can you find any orbits that stay in the vicinity of the starting Lagrange point? What do you conclude about the stability of these three points from your numerical experiments alone? Explore and have fun!

2. Use the "3D Binary Star Integrator" on the class webpage to investigate orbits near the L4 and L5 Lagrange points. Set "Mass of Star 2" to .001, and the "Eccentricity", "Inclination", "Argument of Pericenter" and "Longitude of Ascending Node" to zero.
a) Try different True Anomalies to see what range gives Tadpole orbits around L4, Tadpole orbits around L5, and Horseshoe orbits (ones that surround both L4 and L5). Make a table of your results. How do the Zero Velocity Curves limit the motion?
b) For a Jupiter-mass planet, roughly how many orbital periods does it take to go once around the equilibrium point (This is the Libration Period)? For small Tadpole orbits, try increasing the secondary mass - how does the libration period change, and at what mass ratio is stability lost? Compare this with Danby's value on page 265. Can you spot Danby's error? Make a table of your results and discuss.
c) Try some small eccentricities and inclinations - what happens? Rotating coordinates are most useful, but try some inertial coordinates so that you see what is going on. Write up a page or so discussing your findings and attach some relevant orbits. Explore and have fun!

3. The following program is due in three weeks (with HW #10).
Two-Body Problem. Write a two-part computer program that translates i) from orbital elements $(a, e, i, \Omega, \omega, \nu)$ to positions and velocities (x, y, z, v_x, v_y, v_z) and ii) from positions and velocities back to orbital elements. Devise your own algorithms, or use the ones given in Danby, Section 6.15. For Danby's algorithm, note that Eq. 6.15.4 comes from 6.2.5 and that $\mathbf{P}$ is a vector pointing from the mass-occupied focus to pericenter with magnitude e.
Basic Program Your program should work for 2D ($i = \Omega = 0$) elliptical orbits, but does not have to handle 3D or unbound (parabolic and hyperbolic) orbits.
Extra Credit Your program should work for all 3D bound orbits.
More Extra Credit Your program should handle hyperbolic and parabolic orbits as well as elliptical ones.
Be thorough in testing your program! Test it by translating $(x, y, z, v_x, v_y, v_z) \rightarrow (a, e, i, \Omega, \omega, \nu) \rightarrow (x, y, z, v_x, v_y, v_z)$ and $(a, e, i, \Omega, \omega, \nu) \rightarrow (x, y, z, v_x, v_y, v_z) \rightarrow (a, e, i, \Omega, \omega, \nu)$ for a number of cases. This is why it is best to write both subroutines at the same time! Also test your programs against your intuition for a number of special cases (e.g. circular orbits ought to have $\mathbf{r} \perp \mathbf{v}$). Finally, compare your results with the "Changing the Elements" application in the Working with Orbits section of the Astronomy Workshop. Do this comparison last, as in normal coding situations you usually do not have an already-working example available to you!
Even More Extra Credit Can you find any non-trivial errors in the "Changing the Elements" application? Trivial errors include round-off issues, etc.