

Final Project: A Binary Star Simulation

Can a planet survive in a binary star system?

Background

You may have read science fiction set in a world that has two or more “suns”. Actually, at least half of all stars are members of multiple systems where two or more stars orbit a common center of mass. When there are just two, we call them binary stars. Binary stars are extremely interesting to astronomers because their orbital parameters (semi-major axes, period) can often be measured, allowing a direct determination of the masses of the stars (or, at least, the ratio of the masses). In close binary systems, mass transfer can occur between the two stars, affecting the evolution of both and leading to various kinds of novae. The aim of this project, however, is to find out if science fiction stands up to an investigation of the physics. Can an earth-like (hence presumably life-supporting) planet survive in a stable orbit in a binary system? With major research programs currently searching for “exo-planets” (planets orbiting other stars), this is a particularly timely subject: is it worth including binary stars in such searches?

In order to calculate the orbits, you will need to compute the acceleration of each object due to the gravitational forces exerted on it by the other two. For example, the force exerted on star A by star B is

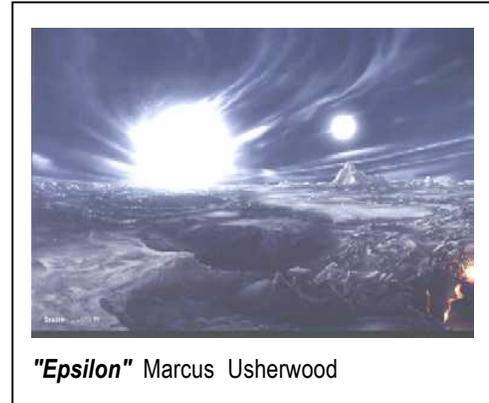
$$\mathbf{F}_{AB} = -\frac{GM_A M_B}{r_{AB}^3} \mathbf{r}_{AB},$$

where $\mathbf{r}_{AB} = \mathbf{r}_A - \mathbf{r}_B$, and $\mathbf{r}_A, \mathbf{r}_B$ are the radius vectors from the origin to the current position of the two stars. (Here, we have written vectors in un-italicized bold script.)

The program

Tackle this project in stages. First, investigate the motion of a planet around a single star (“the sun”). You can use the `orbitx.py` program on the course webpage (where some lines are blank and need to be filled in by you) and, in fact, it is strongly recommended that you work through at least a good part of “Project Briefing 8” on the course webpage. It will also be helpful to look at “Lecture 8”: Simulating Orbital Motion.

Next, modify the `orbitx.py` program to compute and plot the orbital trajectories of both stars in a binary system. For the planet orbiting the sun, we assumed that the sun was stationary (because it is so much more massive than the planet). For two binary stars, we can no longer assume that one has negligible mass, so we will now have two objects exerting gravitational forces on each other and they will end up orbiting about their common center-of-mass. Think about reducing the number of independent parameters. For example, it is



really not both stars' masses that are important but just the ratio of the masses. **Also, work in a reference frame where the center-of-mass of the stars is at rest at the origin.**

Finally, when the binary system is working to your satisfaction, add a third body with an earth-like mass, the planet. To simplify things, you can probably neglect the gravitational forces exerted by the planet on the two stars.

Note: You will need to think very carefully about writing the gravitational forces correctly. For each star, the gravitational force due to the other star will depend on the actual distance between the stars, \mathbf{r}_{AB} , not on the distance of the star from the center-of-mass. Similarly, when you add the planet, the gravitational force on the planet due to each star will depend on the actual distance (and direction) between that star (A) and the planet (P), \mathbf{r}_{AP} , not on the distance and direction of the planet from the center-of-mass of the stars. You are advised to show your code implementing this to your instructor to be sure that you are doing it correctly.

Programming issues: You may need a large number of time steps to accurately follow the orbits.

- If you're bold, consider implementing an adaptive time-step; that is, adjust the timestep depending on position along the orbital trajectory.
- To save memory, you could plot the positions of the stars and planet every n^{th} timestep, where the value of n can be set by the user.

Investigation

First, try exploring binary star orbits without the planet. Start with a fixed mass ratio, say 2:1, and see if you can set up both stars in circular orbits around the center-of-mass.

Try varying (1) the mass ratio; (2) the initial speed of one of the stars.

Now introduce the planet. You could assign it a fixed (small) mass, but as noted above, it is easiest and reasonable to work in the small-mass limit where you assume that you can neglect the gravitational effect of the planet on the stars. It might be a good idea to begin with the stars in circular orbits. Keep the stellar mass ratio fixed and run simulations for different sets of initial values for the planet (i.e. explore different combinations of the initial position and velocity of the planet).

The object is to find a stable orbit for the planet, that is, one in which it doesn't get "kicked out" of the system after a few orbits of the binary star.

What orbital parameters are necessary for stability? Could a planet on such an orbit support life? Could it be detected with current techniques?

There are lots of other things you could try — varying the stellar mass ratio; changing the ellipticity of the star orbits, etc.

Resources

Young, H. D. & Freedman, R. A., *University Physics*, 13th ed., Pearson (2012), Chapter 13.

<http://hyperphysics.phy-astr.gsu.edu/hbase/orbv.html#co> (the basic physics of orbital motion, including binary orbits.)

<http://farside.ph.utexas.edu/teaching/336k/Newton/node50.html> (more detail here.)

<http://burtleburtle.net/bob/physics/index.html> (lot's of ideas, information and simulations here!)

<http://exoplanet.eu/> (The Exoplanet Encyclopedia, including a frequently updated catalog.)

<http://exoplanets.org/> (Exoplanet Data Explorer. Both of these sites are maintained by research teams active in the field.)

<http://astro.unl.edu/naap/> (Various applet based tutorials illustrating concepts in astronomy, including binary stars and exoplanets.)