Day 2 Activities: Modular Programming

Anna L. Rosen

Activity 2: Stellar Flux

Write a program, which incorporates the techniques of modular programming, to calculate

- the flux (power per unit area) received by a planet that is orbiting a star, with luminosity L, at a distance d.
- the amount of power absorbed by a planet which has a radius R, (a) assuming all energy is absorbed by the planet (albedo = 0), and that some radiation is reflected by the planet (0<albedo< 1). Take albedo to be an optional keyword where the default value is 0.
- and total energy absorbed by the planet for a given time.

1. Write a main program that prompts the user to input L in units of solar luminosity ($L_\odot = 3.84 \times 10^{33}$ erg s^{-1}), d in AU units (AU $= 1.5 \times 10^{13}$ cm), R the planet's radius, and a the a planet's albedo (fraction of radiation that is reflected).
2. Generalize this to read in a list of the planets in our solar system (use the sun's luminosity). Data taken from http://hyperphysics.phy-astr.gsu.edu/hbase/solar/soldata2.html. Plot the flux as a function of distance. The file ssplanets.txt contains the planet's names (column 1), distance (column 2), diameter (column 3), albedo (column 4). To read in this file easily use the numpy function genfromtxt (documentation here http://docs.scipy.org/doc/numpy/reference/generated/numpy.genfromtxt.html). This will give you an array of the data in the table. Since the first column are the planet's names you will get junk but you can write your own list or read it in a similar fashion like we did with the previous example using the tab as the final character.
3. Make simple line or scatter plots of the flux received at each planet and the power and energy absorbed by each planet. Use the age of the solar system, 4.6 Gyr, as your time argument needed for your energy function.
4. If there's time, Import the module zams.py which contains functions to compute the zero age main sequence (i.e., birth) luminosity, radius, and surface temperature of a star with a given mass (in units of $M_\odot$). Now use this to get the luminosity of a star

with a mass chosen by the user and make the same plot as before but for different luminosities.

5. If there's time. Use the numpy routines linspace() or logspace() which returns an array of linearly or log spaced values to create a distance array. Documentation here: http://docs.scipy.org/doc/numpy/reference/generated/numpy.linspace.html, http://docs.scipy.org/doc/numpy/reference/generated/numpy.logspace.html. Use this array to calculate the flux received and energy absorbed for a list of planets with different radii and albedos. Plot your results.