ASTR4260: Problem Set #3

Due: Wednesday, 2023 October 4

Submit in a new directory in your main repository.

Problem 1

In this problem, you will read in light curve data from the Kepler satellite and plot it.

(a) Download the light curve data associated with the host star of Kepler 7b, one of the first Kepler exoplanets detected (see the ApJ paper describing it: ApJL, 2010, 713, L140). If you wish, you may use a different detected planet. As usual, this can be done a number of ways.

One nice way, which I recommend for graduates, and those with some operating system experience, is to use the `Lightkurve` python package, which is described at docs.lightkurve.org. If you have a simple environment, with just the anaconda install from Problem Set 0, you can use the package installer `pip` by issuing the command: `python -m pip install lightkurve`. (Using this formulation guarantees that you use the pip associated with your active python installation.)

If you have a more complex environment with many other packages installed (I'm looking at you, graduate students and CS majors), pip can make a mess of things. In that case, or just if you want to get experience with this technique, you can set up a virtual environment that will isolate what pip does to a sandbox:

```
python -m venv <directory name>   #you can use . for your current dir
cd <directory name>
source bin/activate    #assuming your shell is sh, bash, zsh...
#source bin/activate.csh     #if your shell is csh
```

Your prompt will change to show that you are in an altered environment. At this point, you can use pip as above to install `numpy, scipy, jupyter`, and `lightkurve` in your sandbox. Doing it in this order will reduce incompatibilities. `python -m pip freeze` will list what is installed. After installing everything, you may need to issue the command `rehash` to force recognition of the new packages that have been installed. When you are finished with the environment for the moment, you can use `deactivate` to exit the environment. You can always return to it by sourcing the activate script again.

Once you have `lightkurve` installed, you can use it to fetch the data from MAST, the online data repository, using something like the following code snippet:[1]

```
import numpy as np
import lightkurve as lk

lcs = lk.search_lightcurve("Kepler-7b", mission="Kepler", cadence="short").download_all()
time     = lcs[0].time.value                        # get the time of each observation
flux     = np.array(lcs[0].flux.value.data)         # get the flux
flux_err = np.array(lcs[0].flux_err.value.data) # get the error in the flux
```

[1]Note that this requires an active internet connection. See the `Lightkurve` documentation for more information about using the package.

As an alternative, you could visit the MAST archive and download the data directly, at: archive.stsci.edu/kepler. This takes a bit of work as the files are in FITS format (which can be read using the `pyfits` package—more on this in a later problem set).

However, you read it in, you should have three arrays: `time`, `flux` and `flux_err` for one set of observations of a single star observed by Kepler. The time of each observation is Coordinated Universal Time (UTC) - 2454833 (this arbitrary number is subtracted in order to reduce round-off errors and also to make it more convenient to deal with), and the flux is in counts/s, although we won't worry too much about the units of the flux since we're primarily interested in the *ratio* of fluxes (eclipsed to not eclipsed).

(b) Plot the flux vs. time, including error bars using the `matplotlib` package[2] . You may have to clean the data if there are NaN's. For our purposes, they can be filled in with values taken from nearby times. Please label the axes accurately and include the plot with submission. The `errorbar` function from matplotlib will be useful. Note that there are enough points that the error bars will blend together!

Problem 2

In this problem, we want to compare the data to a theoretical light curve. To do this you will need to use machinery from Problem Sets 1 and 2. First, examine the data you have plotted above and find a section that looks like it corresponds to a single eclipse. Then extract a section of the data corresponding to that eclipse (include a little time before and after the eclipse to show the baseline clearly). These are the data that we will try to fit with a theoretical eclipse curve.

(a) Begin by trying to fit the eclipse using the flux ratio code from problem set #1 with uniform stellar intensity. Make an estimate of the required parameters by eye; we'll get into statistical model fitting later in the course.

You will need to convert from time t to z, which you can do with

$$z(t) = (t - t_0)/\tau \tag{1}$$

where τ and t_0 are constants: τ is related to the duration of the eclipse and t_0 is the time of maximal eclipse. Guess values for p, τ and t_0 and calculate $F(p, z(t))$ for the same t_i values as the data for Kepler-7b that you downloaded in problem 1. Experiment to find an eclipse shape that approximately matches the data. Generate a labelled plot and include it with your submission. If you are having trouble getting reasonable parameters, ask me for a hint.

(b) Use the code you wrote for Problem Set 2, but now with the more realistic limb-darkening function

$$I(r) = 1 - (1 - \mu^{3/2}) \tag{2}$$

where $\mu = \cos\theta = (1 - r^2)^{1/2}$.

Problem 3

There are many definitions of the habitable zone around a star, but typically they require liquid water on the surface of planets orbiting within it. A theoretical determination of the habitable zone range is

[2] You may either create an image containing the plot or, better yet, include the image inline in your jupyter notebook—to do so, be sure to include `%matplotlib inline` with the imports in your notebook)

given in Kopparapu et al (2013, ApJ 765, 131). They determine that the inner edge of the habitable zone is given in Astronomical Units (AU):

$$d = \left(\frac{L/L_{\odot}}{S_{\mathrm{eff}}} \right)^{1/2} \mathrm{AU}, \tag{3}$$

where $L/L_{\odot} = (T_{\mathrm{eff}}/5780\ \mathrm{K})^4$ is the luminosity of the star in terms of its effective temperature T_{eff}, S_{eff} is given approximately by

$$S_{\mathrm{eff}} = S_{\mathrm{eff}\odot} + aT_* + bT_*^2 \tag{4}$$

and $T_* = T_{\mathrm{eff}} - 5780$ K. For the inner edge of the habitable zone, $S_{\mathrm{eff}\odot} = 1.014$, $a = 8.177 \times 10^{-5}$ and $b = 1.706 \times 10^{-9}$. One AU is the mean distance from the sun to the earth.

Using one of the root finders described in class, determine the value of T_{eff} for a radius of $d = 0.5$ AU (i.e. find T_{eff} for which $f(T_{\mathrm{eff}}) = d(T_{\mathrm{eff}}) - 0.5\ \mathrm{AU} = 0$). This radius marks the point beyond which it is challenging to find exoplanets through the transit technique and so, if we want to find planets in the habitable zone using this technique, we are forced to focus on cool stars with T_{eff} values smaller than this. (Of course, this yields "habitable" planets so close to their stars that they risk being tidally locked and heavily irradiated by stellar activity, but that's a discussion for another time.)