# Examination of Planetary Systems with Kepler Data

C. E. Teclaw

with Prof. Philip Arras

5/2/2018

University of Virginia
Department of Astronomy

This thesis is submitted in partial completion of the requirements of the BS Astronomy-Physics major 5/2/2018

**Abstract**

In this period of study we attempted to use data from the Kepler archive to deduce information about planetary systems observed by that mission. The ultimate goal was to potentially identify features of the planets in these systems by examining ellipsoidal variations in their light curves. For this purpose we attempted to, separately, apply PyKE utilities and create new code to search the mission's fits files. We discovered potential issues with the PyKE code regarding its compatibility with Kepler data and were unable to produce any meaningful results from new code due to time constraints.

# 1 Introduction

The transit method of exoplanet detection gives a large amount of information about the planetary system, including the ratio of the planet's size to its star's, the semimajor axis in stellar radii and the orbital period. Related flux calculations can even give insights as to the composition of the exoplanet. Transits, the passage of a planet between its host star and the observer, are nearly undetectable as the light blocked by the transiting object is proportional to the square of the cosine of its orbital plane to the plane of observation. owever, once a transit as been found, it is certain that te inclination of the system is very close to 1, so any equations involving that variable are made simpler. The change in fluxat the midpoint of a transit is

$$\frac{\Delta F}{F_*} = \left(\frac{R_p}{R_*}\right)^2 \tag{1}$$

assuming that the star's emission models that of a blackbody, since the flux received from a sphere by a stationary observer is

$$F = \pi R^2 \sigma T^4 \tag{2}$$

by integrating the Stefan-Boltzmann law. Kepler's third law provides the semimajor axis of the planet given the period, which is easy to derive from data. If we approximate the planet's mass as small relative to thaht of its star, this is

$$a = \left(\frac{GM_* T^2}{4\pi}\right)^{1/3} \tag{3}$$

where G is Newton's constant, $M_*$ is the stellar mass, and $T$ is the measured period. While the mass cannot be derived by the transit method, it can be estimated using other methods, for instance using the star's spectrum to calculate its temperature, which is closely related to mass
The transit has three distinct stages. if the first stage,the orbiting body (the planet) begins to pass in front of its host star. This corresponds to a sharp decrease in measured flux. In the second stage, the light blocked by the planet is relatively constant as it crosses the star, leading to a constant flux measurement. While it is not truly constant due to limb darkening increasing the observed brightness from the center of the star relative to the outer region, the effect is small compared to the complete passage of the planet in front of the star. In the third stage, the planet passes away from the star, the light received from which returns initial value. Figure 1 shows several periods of the kplr012557548 system illustrating this description.

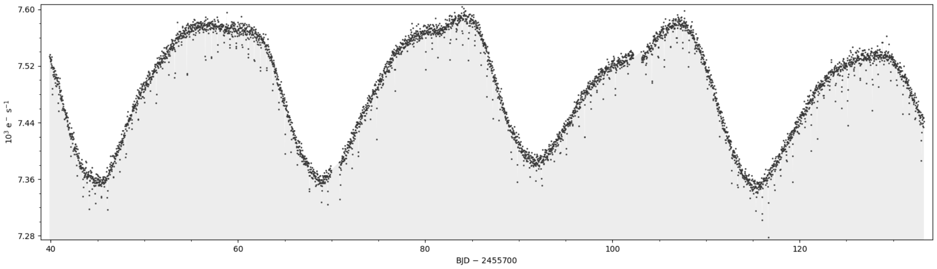

Figure 1: Three periods of kplr012557548's planet,in appropriate units, with a time offset. This graph was produced using data available on the Kepler website using Pyke utilities.The vertical scale is in units related to the instrumentation that correspond to received flux. A more detailed description of this figure is in Sections 2 and 3.

Here the intermediate phase of the transit is short due to the rapidity of the planet's orbit (roughly 20 days). The lack of uniformity in the signal (i.e. the difference in heights of the peaks) is an observational effect due in part to the positioning of the Kepler telescope.
When a planet passes behind its star, light is also blocked. The planet reflects a small amount of light when it is not directly in front of the star. But half

an orbital period after the transit, no light is reflected as the planet passes behind the star (the secondary transit), leading to a dip in flux substantially smaller than that from the primary transit. Secondary transits can be deduced from looking at the peaks of Figure 1. Finding a way to locate and find the parameters of these secondary transits can yield information about the planet.

This was to be done by examining the relative changes in flux through the system's orbital period to evaluate the deviation between the observed light curve and an idealized sinusoid. Based on that the depth of the secondary transit is known, which, when compared to the maximum observed flux, tells how much light is reflected by the planet.

Data from the Kepler mission is particularly ripe for such analysis, as its explicit goal was the detection of exoplanets. Thus all of the systems it targeted were good candidates for being planetary systems. Kepler produces a large number of examples of transits with the same data format, for which utilities ave been made which can easily read the data and output it in a more useful form than that in which it appears, and correct for inconsistencies associated with the system.

Using transits to learn about exoplanetary systems teaches us about the character of the universe at the mid scale. How often and where in a system various kinds of planet appear is valuable in creating more accurate models of the formation of planetary systems, which can then be applied to similar systems such a the formation of galaxies.

The Kepler mission began in 2009, when the Kepler space telescope was launched by NASA to search for terrestrial exoplanets via transit. In order to capture potentially multiple periods, the Kepler telesccope remained stationary for tens of days. Due to the nature of transit, larger planets closer to their stars are easier to detect, resulting in far greater numbers of hot Jupiters, planets matching both criteria, than Earth sized planets being discovered.[7]

## 2 Technique

PyKE is an open source set of python utilities that describes itself as "a software package for the reduction and analysis of Kepler data." [1] It uses photometry to analyze light curves, particularly in correcting systemic features that hinder the acquisition of useful data. The particular variety of

PyKE[5][6] used here was the stand alone version, which bypassed PyRAF which is a graphical interface for PyKE. Instead, the version used ran from the command line.

The web page is old, and has not been updated, however. As an example, the keptransit command, which fits a model of planetary transit to a light curve, is no longer include with the PyKE package though it is listed on the home page. This created many difficulties, as it was impossible to simply use the listed information on usage of the utilities, since they had undergone updates and changes since the last update to the website. It was not unusable, as it was possible to find other documentation that was more recent elsewhere.

The kepclip and kepstitch utilities were still functional. They enabled merging and partitioning light curves across time segments. Though these proved only tangentially useful they functioned precisely as described. Kepdraw simply graphed the light curve without any complications

Kepfold, which was supposed to fold the data series given a period so as to maximize the signal to noise from the data. However, as will be described in more detail in the next section, this in particular was different from its documentation and made using it difficult. Figure 1 was produced using the kepdraw function.

Kepflatten also functioned differently from how its documentation suggested and had little noticeable effect on the data.

The other method used in the course of this study was the independent pasting of the fits files supplied by the Kepler website. This method was difficult in particular due to the opacity of fits file data layout. The organization of Kepler fits files could be found in the Kepler Archive Manual, together with similar information about the data as presented online for download.[2]

This method requires taking in the data from the mission and compensating for systemic features such as consistent trends not part of the star's signal. According the the archive manual referenced above, the fits files provided by Kepler contained PDCSAP flux, which had been corrected for instrumental and other measurement error and detrended (corrected for trends outside of the signal).

One of the advantages of the more home brewed approach is that it allowed one to use the data in more elaborate ways. With all of the data points it becomes possible to take the Fourier transform of the light curve to find the dominant frequency of the signal, which gives the orbital period easily. That can be done using the lombscargle function, which is part of the astropy library. Astropy is part of the Anaconda distribution, which contains a large amount of Python code focused on science and statistics. Lombscargle was a sufficiently effective replacement for the PyKE Fourier transformer that that was discontinued with a recommendation to use the former.

To retrieve the data from a fits file, it must be opened with a special utility (an open source library of which is part of astropy), and its data structure must be navigated manually to pick relevant data. After that, they may simply be put into numpy arrays which can be easily manipulated to produce scientifically useful results.The inclusion of error ranges in the files allows for a more rigorous result.

The Kepler data is located at https://archive.stsci.edu/, where it can be accessed by inputting the name of a star covered by Kepler's field of view and looking through the data series for one covering the desired time range. Some can even be examined in browser, though the graphs shown are uncleaned and do not match those output by kepdraw, which lends credence to the idea that they are differently packaged data.

# 3 Process

The first step was to determine the usability and function of PyKE library. This naturally involved reading the documentation of the use of relevant functions and following the examples to ensure everything functions properly. kepdraw, kepflatten, and kepstitch all worked as expected, but kepfold did not. It produced a result very different from that shown in the example, Figure 4. These plots cover the data referenced in Example 3 of PyKE's photometry tutorial.

Contrasting the output with the advertised output is quite alarming, since it is reasonable to expect something resembling a transit when there is one clearly present, as can be seen in Figure 1.

While the figure from the example is clean and depicts what is clearly a transit, the actual output is entirely dissimilar from that, showing no order. As mentioned, the command for kepfold worked very differently from how it was portrayed in the example. Kepfold rejected some of the arguments used in the example, such as niter. Multiplying and dividing the period by 2, 3, 4, 6, 12, and 24 failed to improve the result. Slight changes also had no effect. To ensure that this was not a unique effect that was limited to the example system, we attempted to repeat the process of the example on a different light curve. That chosen was Kepler 22b, which was discovered by the tran-

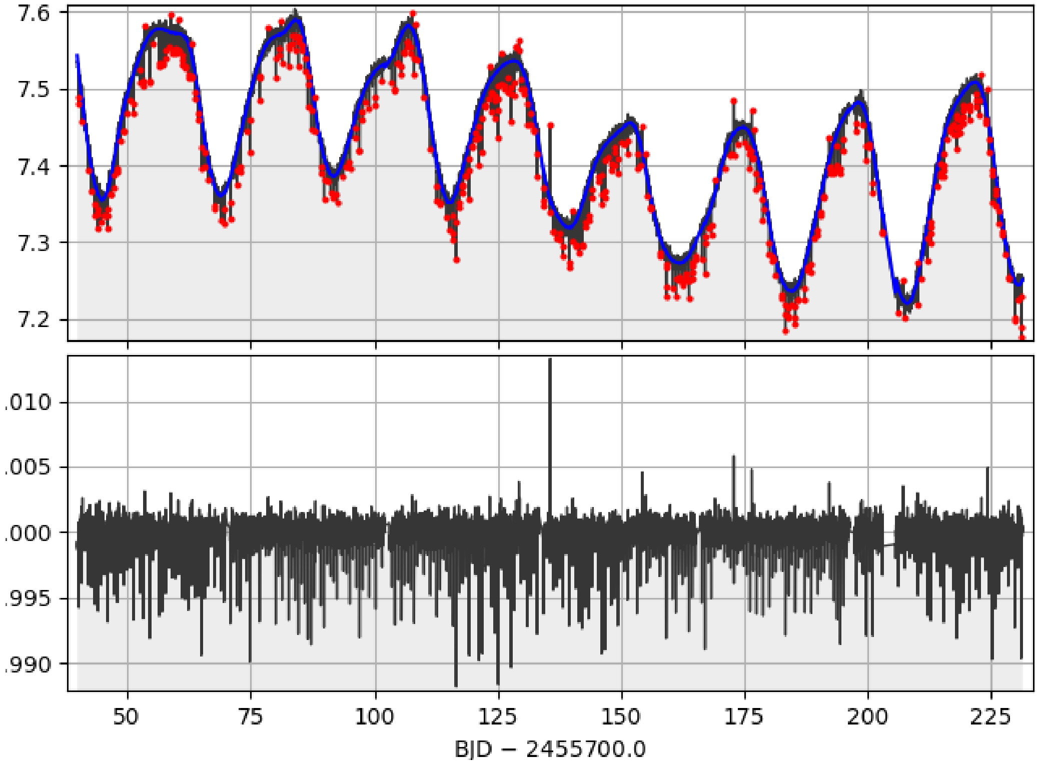

Figure 2: Kepflatten, attained as laid out in the example directions. Identical to the example's graph. The units and normalized flux and thousands of electrons per second.

sit method, and so must have a noticeable signal. However, the results of this test showed the same issues for the example system. Data came from the first data file chronologically from Kepler 22b on the MAST archive for data collected by the Kepler mission.

While the data in Figure 5 is not pristine, it contains a clear sine pattern indicative of some sort of flux oscillation. Given that there is a known transit derived from the same data, it must certainly be this. Note that the deviation measured in the bottom half of the plot is much more sporadic in the 22b system. Further, there is a much lower level of flux received from this source, which could have some effect in increasing noise.

Applying kepflatten to a different time series from Kepler 22b7 provided no novel results, and in effect merely replicated the first. This continuity

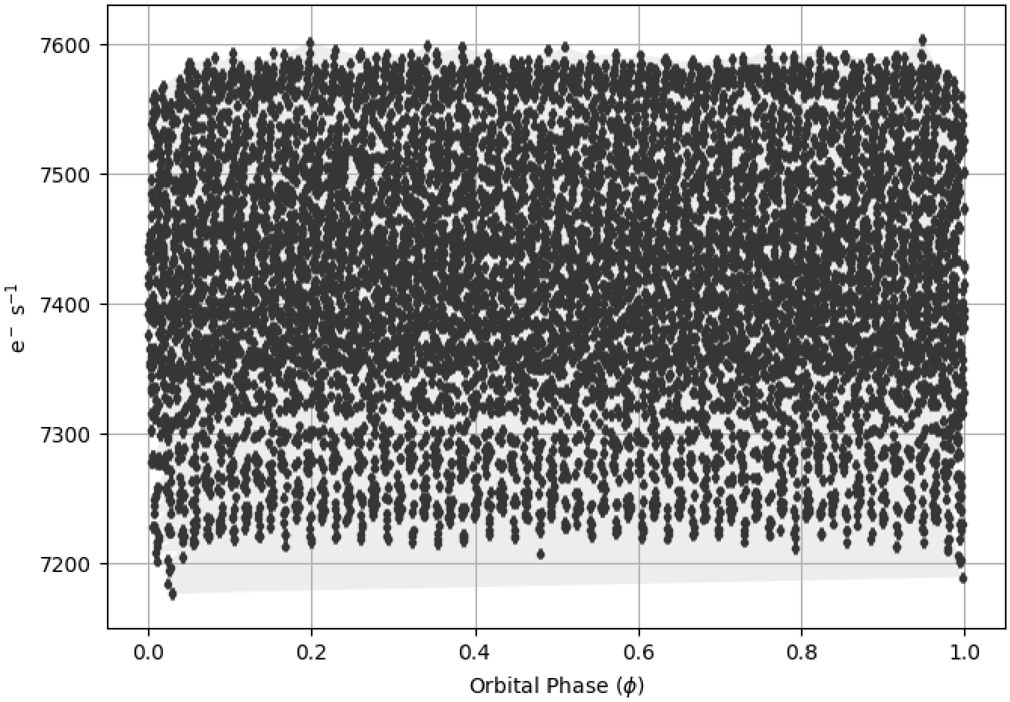

Figure 3: Kepfold, attained as laid out in the example directions. The period used was the same as that used my the example command line entry

reduces the plausibility that the deviation from true results is due to faulty individual data sets. Attempting to apply kepflatten more times in order to further reduce inconsistencies in the data proved fruitless, with results effectively indistinguishable from a single application.I suspect this may be due way kepflatten is coded, though without a lengthy examination of the code it is impossible to be sure.

In order to prove even more fully that there was indeed an error with the PyKE code we tested it on a third system, Hat-p-7, which has a well documented transit. An article in *Science* magazine lists the flux drop as 130 parts per million, though their graph shows a troughof roughly 7000 ppm.[4] For unknown reasons this source produced correct results. The flattened results showed incontrovertible signs of transits.

More noteworthy than the flattened light curve is the folded light curve, which is properly formed, with a clearly defined transit at phase zero relatively constant elsewhere. The period used in deriving this graph (Figure

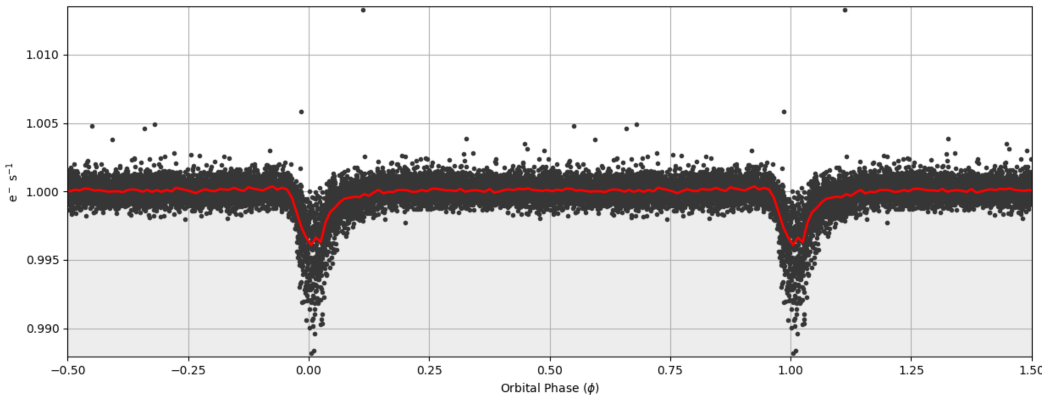

Figure 4: The results of kepfold shown in the example. There is an even distribution of points with obvious dips in flux at the transit points which are located at 0 and 1 orbits, as expected.

9) was that found by the *Science* paper. Additionally, the drop during the transit is on the order of 7000 parts per million, estimated visually, roughly matching the result found by Borucki et al., graphed in their figure.

This result is especially puzzling because kepflatten was ineffective for the other two transiting systems. Though it is clearly different code from that used in the example (the lack of a red line through the center being the most obvious sign of that). As such, it is impossible to conclude anything from the results described above, except that it is very plausible that there is some error in the revised PyKE code, or there has been a change in syntax that has not been updated in the documentation.

Considering that there is little to be learned from continuing to attempt meaningful discovery with tools that we could not reliably utilize, we switched to attempting a manual analysis of the Kepler data. The complexities enumerated in the previous section proved real, blocking rapid progress from this approach.

The utilities for manipulating fits proved only compatible with Python 2. This required a slightly different coding style than had astropy been able to cooperate with Python, though the functional effects were minimal. As by this time there was little of the term remaining, it was not possible to make advances in this line of inquiry beyond graphing an undetrended light curve of Hat-p-7, Figure 10.

An attempt to manually fold the data by creating a while loop to subtract to

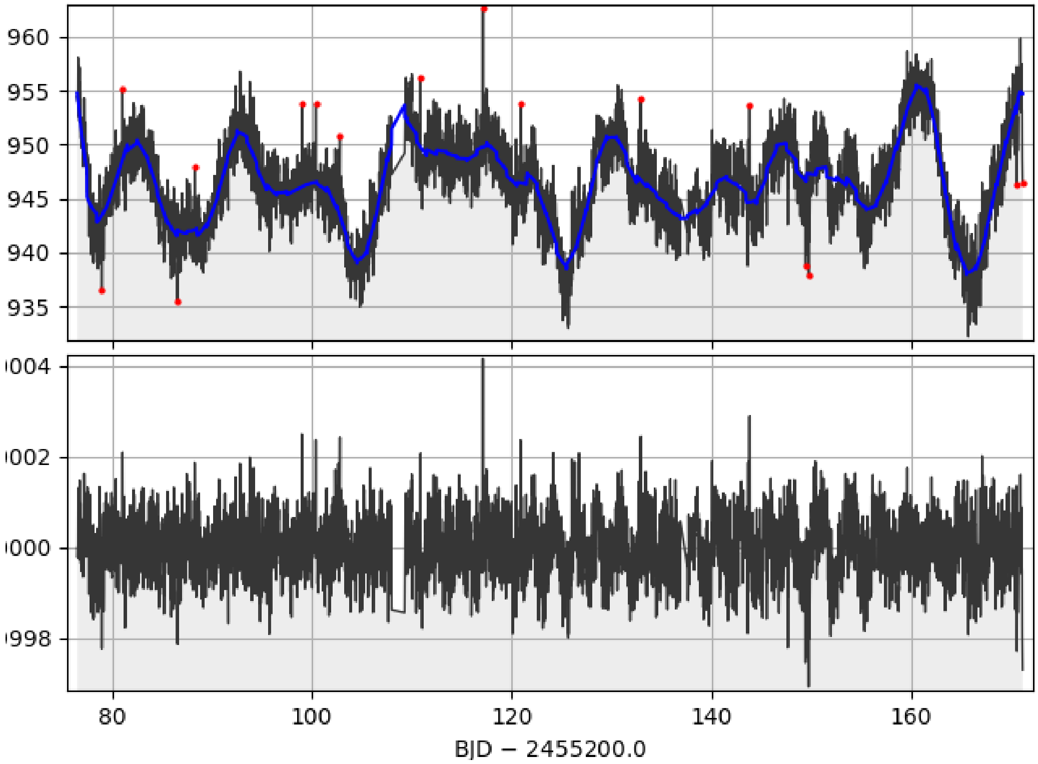

Figure 5: Kepflatten applied to Kepler 22b. Not a smooth pattern as was in the example. The units are identical to those of the example relative flux and kiloelectrons per second).

period from each data point was unsuccessful. An attempt to use the modulo function of Python had also failed. Given the lack of success of these two methods, it is reasonable to assume that making progress in manual folding would require dedication beyond the scope of this project. Additionally, even had it been successful, the computational time consumed by the while loop method is prohibitively high.

I believe that the Python coding method could be successful in providing a replacement to the uncertainty of PyKE's operation. It is also worth investigation whether PyKE is truly compatible with Python 3, or is simply unable to return an error message, and so cannot say when it is unable to act as expected.

Due to the brevity of the spring term, there was not sufficient time to inves-

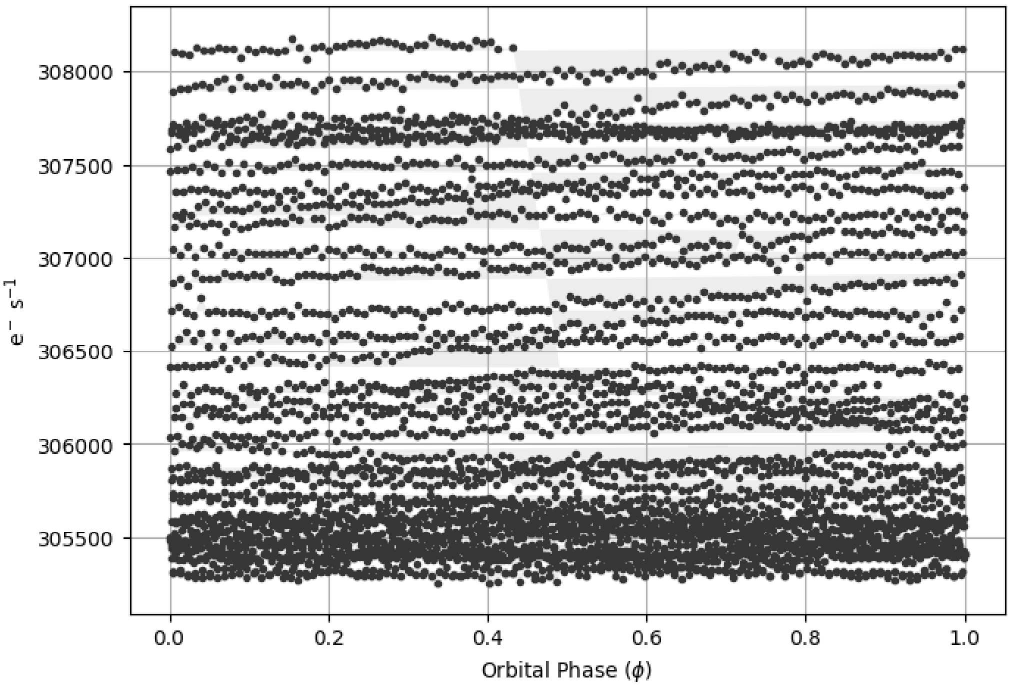

Figure 6: The results of kepfold for Kepler 22b. No qualitative difference from the example, except that here the points appear to be in horizontal rather than vertical lines.

tigate whether lombscargle would be effective in calculating periods for these transiting systems. While at-p-7 showed no need of another calculation,both Kepler 22b and the example source may have been resolved if their true periods were different from those listed, perhaps from a peculiarity of the data. Our work towards independent Python code yielded no results from which meaningful conclusions can be drawn.

# 4 Conclusion

A heavy investigation of the function of the PyKE library shows that its utilities cannot necessarily be relied upon. Outdated documentation describing arguments no longer present and undoubtedly neglecting newly added ar-

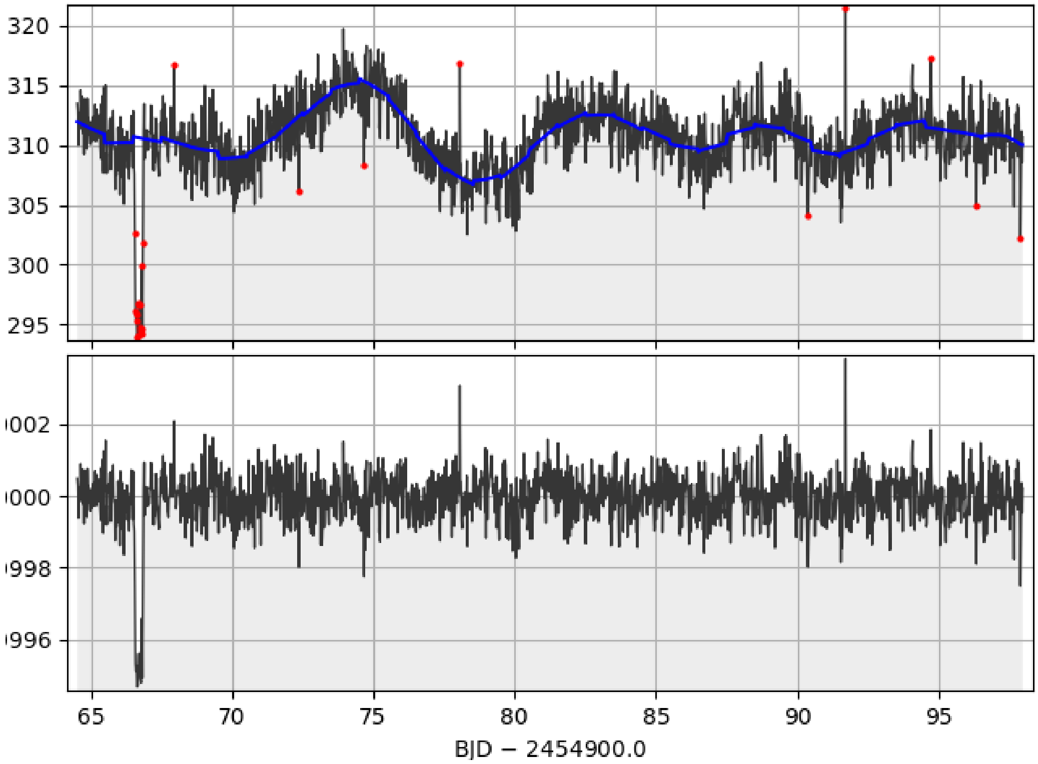

Figure 7: A second flattened light curve from Kepler 22b. These data are from a different observational period from the first, but contain the same units on the plot. While smoother than the first, it is still highly irregular and does not show a clear signal of planetary transit.

guments, possible unmentioned clashes with Python 3, and unpredictable functionality mean that it is prudent to be wary when using PyKE. This conclusion is alarming, considering that PyKE is the standard prepared toolkit for analyzing Kepler data files. We hope that this inconsistency is an isolated incident that does not affect other users of the data.

A sufficient attempt to create independent Python code to bypass PyKE was outside of our scope, but the progress made under given conditions is promising that such code would not be excessively difficult to create in context other than that of an undergraduate thesis. The lack of noteworthy success along these lines should not be taken as indication of its imprudence.Clearly the creators of PyKE were able to create utilities differing from those already

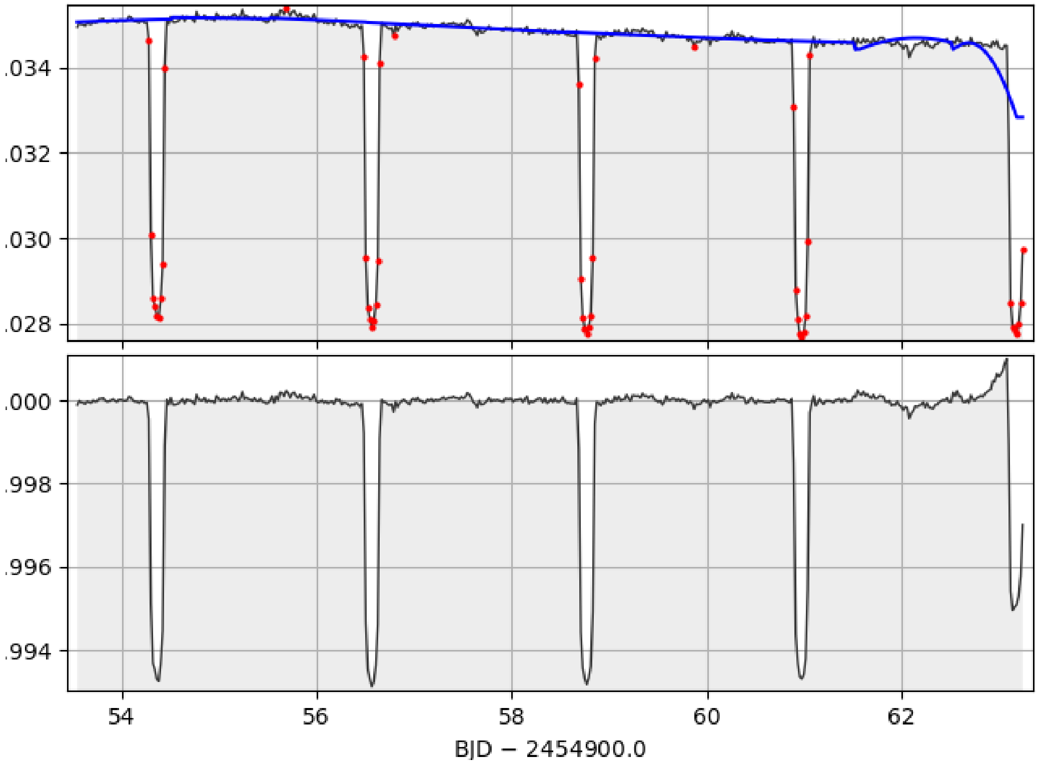

Figure 8: The results of kepflatten for Hat-p-7.It displays precisely the ideal profile of a transit.

existing.
Indeed, these conclusions should not be taken as a wholesale condemnation of PyKE, as it is sufficiently complex that a small rare inconsistency could easily be obscured. Regardless, we have demonstrated a clear possibility of inconsistency between its output and external finding which cannot be explained at present. While nothing pertinent to exoplanetary trans was found, there is sufficient evidence to be cautious of PyKE's output in the future.

# References

[0] ll websites accessed 5/2/2018.

[1] PyKE website, https://keplergo.arc.nasa.gov/PyKE.shtml

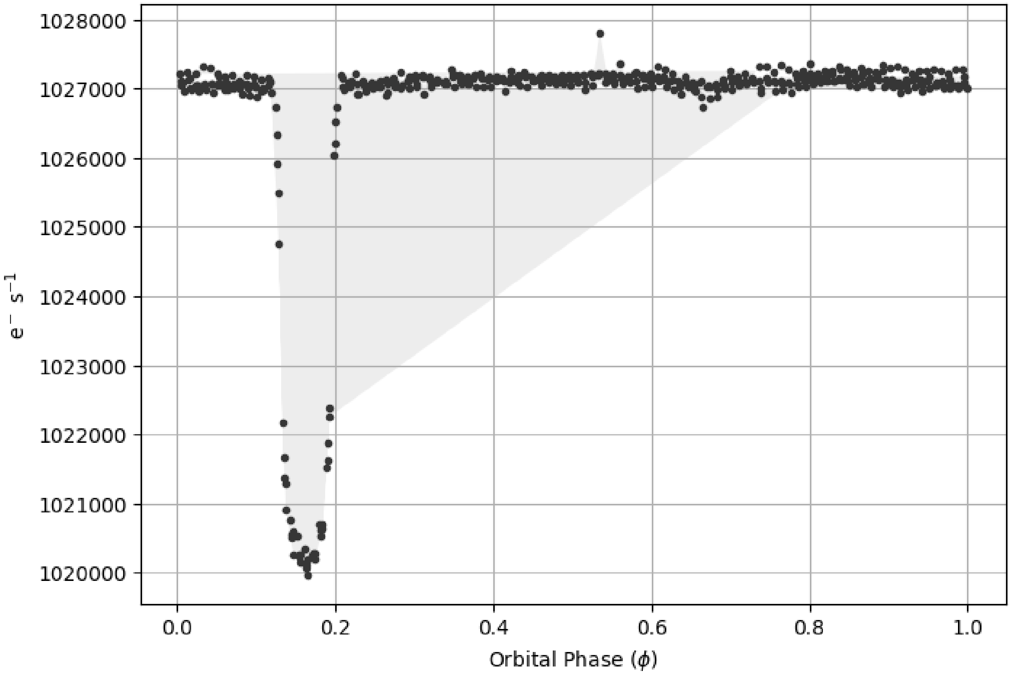

Figure 9: The results of kepfold for Kepler 22b. No qualitative difference from the example, except that here the points appear to be in horizontal rather than vertical lines.

[2] Kepler Archive Manual, http://archive.stsci.edu/kepler/manuals/archive_manual.pdf

[3] Example 3 from PyKE, http://pyke.keplerscience.org/tutorials/removing-stellar-astroph-quarter-stitch.html

[4] Borucki, W. J., et al. 2009, *Science*, DOI: 10.1126/science.1178312

[5] Vinícius, Z., et al. 2017, DOI: 10.5281/zenodo.835583

[6] "PyKE: Reduction and analysis of Kepler Simple Aperture Photometry data", Still, M, and Barclay, T., 2012, http://adsabs.harvard.edu/abs/2012ascl.soft08004S

[7] https://en.wikipedia.org/wiki/Kepler_(spacecraft)(for general purpose information only)

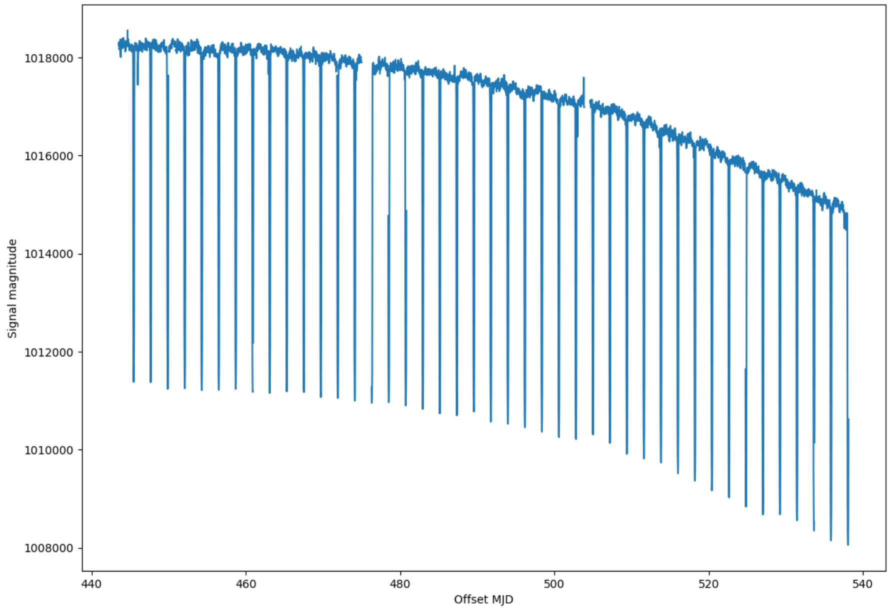

Figure 10: Graph of hat-p-7 created by gathering data directly from the Kepler fits file. The decreasing trend in flux could be eliminated with sufficient time.