Finding Exoplanets with Lightkurve

Alison Thomas - AST390

May 9, 2025

Abstract

1 Introduction

The discovery and study of exoplanets have dramatically expanded our understanding of planetary systems. One of the most successful methods of detecting exoplanets is the transit method, which observes periodic dips in a star's brightness caused by a planet passing in front of it.

I developed a Python-based workflow using the *Lightkurve* [LCH+18] package to analyze data from the Kepler Space Telescope and detect a planet orbiting the star Kepler-8 (KIC 6922244) . The goal was not only to identify transit signals but also to interpret them within a broader astrophysical context, including estimating the host star's physical properties.

Lightkurve is an open-source Python library designed for downloading, analyzing, and visualizing time-series data from space telescopes such as Kepler, K2, and TESS. It simplifies many of the technical aspects of photometric data analysis, such as downloading light curves, folding data over suspected orbital periods, and utilizing periodograms and seismology. These aspects make *Lightkurve* a valuable tool for both students and professional astronomers. In this project, *Lightkurve* allowed me to efficiently obtain and process the light curve for Kepler-8 as well as understand the scientific interpretation of the data.

Kepler-8 is a well-studied system that is already known to host at least one exoplanet, Kepler-8b, a gas giant first confirmed in 2009. Using *Lightkurve*, I was able to reproduce evidence of Kepler-8b's transit which helped me gain insight into what astronomers are able to accomplish with the help of Python.

This report outlines the algorithmic approach I used, the relevance of the results, and the broader implications as well as the limitations of this detection method.

2 Overview of the Algorithm

Firstly, to detect a transiting planet around Kepler-8, I used *Lightkurve*'s 'search_lightcurve' feature [GSB+19] to bring up the data found from the data telescope and plotted what that light curve looked like [OHb]. The light curve can be seen below. The data retrieved from the feature contained the Pre-search Data Conditioning Simple Aperture Photometry (PDCSAP) which removed instrumental noise and systematic trends; it also automatically flattened the data to make it easier to plot and measure.

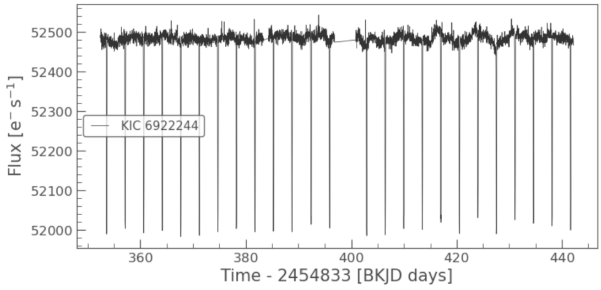

Figure 1: Graph of Kepler-8 Flux vs. Time, revealing a periodic dip in brightness

From the graph, we can see a periodic dip in flux that occurs a little less than every five days. The fact that the dip is so periodic indicates the presence of a planet. To get a better look at the true orbital period of the planet, I implemented the package's Box Least Squares (BLS) method to see that compared to the period. The plot returned a periodogram that showed the likelihood of the BLS fit for each period in an inputted array from one to twenty days [Sau].

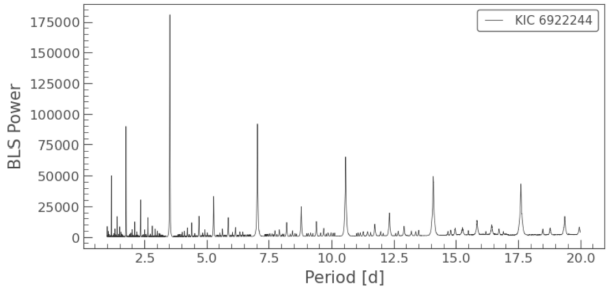

Figure 2: Graph of Box Least Squares Power vs. Period, revealing maximum peak at around 3.5 days

From the figure above we see that the highest peak is at around 3.5 days; the smaller peaks are simply fractional harmonics of the period. To get the actual value of the peak I used the package's 'period_at_max_power' feature [Sau]. This gave an output value of around 3.523 days.

Next, I wanted to see how true that value was, so I normalized the flux and compared it to the star's phase. I wanted to see that the normalized flux stayed at a value of one and that it only dipped where the phase was zero. I was able to do this by folding the light curve 1 and plotting it as seen in

the graph below.

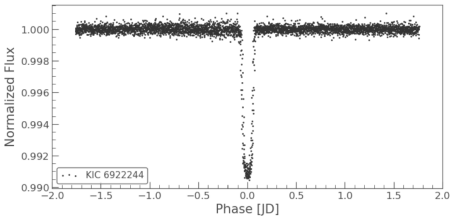

Figure 3: Graph of Normalized Flux vs. Phase

From the plot above, we can see that the period found is pretty accurate. So as we can see, by using *Lightkurve* we can plot a star's light curve data, find a possible orbiting exoplanet by spotting periodic dips in flux, and calculate that planet's period using a periodogram.

Now that the period of the exoplanet was calculated, I wanted to find out more about the star's attributes. The data received from *Lightkurve* had the effective temperature of the star but not other basic attributes such as the mass, radius, and surface gravity. To be able to calculate these values, I first needed to find ν_{max} and $\Delta\nu$. I plotted the star's power versus frequency which would make it easier to find ν_{max} [OHa].

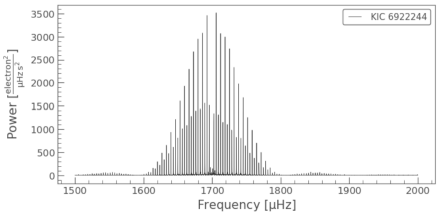

Figure 4: Graph of Power vs. Frequency

From the plot above we can see that the peak was around 1700 μHz. To get a bit of a closer look I plotted the Signal to Noise Ratio (SNR) versus Frequency. The SNR is just a flattened version of the power that we can use with *Lightkurve*'s seismology object and its 'diagnose_' tool.

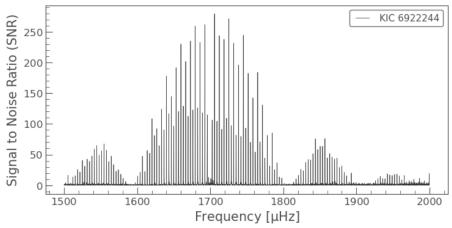

Figure 5: Graph of SNR vs. Frequency

The next step was to use 'diagnose_numax' which showed how the package could calculate the value of ν_{max}.

Starting at the top panel in the figure, the SNR versus Frequency graph is simply zoomed in to

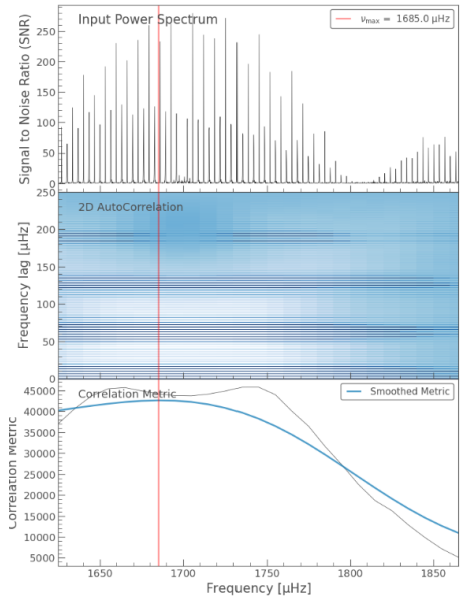

Figure 6: Three graphs showing SNR vs. Frequency, Frequency Lag vs. Frequency, and the Correlation Metric vs. Frequency.

see where the max peak is. The middle panel showed the Autocorrelation Function (ACF) values calculated for different parts of the power spectrum; the darker the color, the stronger the oscillation. The bottom panel showed the same ACF values but with a collapsed y-axis. The blue line was the smoothed metric, the maximum value of that line is ν_{max} which was determined to be 1685.0 μHz [OHa].

Once again, I used the 'diagnose_' feature but for $\Delta\nu$ this time. The value was calculated with the Full Width Half Max (FWHM) on either side of ν_{max} and found the peak that was the closest to the empirical $\Delta\nu$ which was found to be 91.1 μHz.

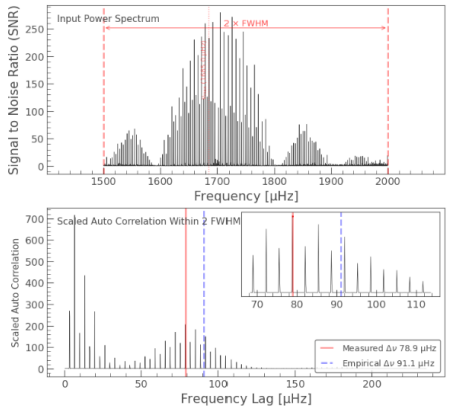

Figure 7: Two graphs showing SNR vs. Frequency and the Scaled Auto Correlation vs. Frequency Lag

The top panel in the figure above was the region where the ACF was evaluated. The bottom panel

was the result of calculating the correlation of the data with itself to find the peak nearest to the empirical value to find the measured value which was found to be 78.9 μHz [OHa].

After finding the values ν_{max} and $\Delta\nu$, the mass, radius, and surface gravity could be determined through the built-in *Lightkurve* scaling relations. The scaling relations they used are shown below.

$$\frac{M}{M_\odot} \simeq \left(\frac{\nu_{\rm max}}{\nu_{\rm max,\odot}}\right)^{3} \left(\frac{\Delta\nu}{\Delta\nu_\odot}\right)^{-4} \left(\frac{T_{\rm eff}}{T_{\rm eff,\odot}}\right)^{3/2},$$

$$\frac{R}{R_\odot} \simeq \left(\frac{\nu_{\rm max}}{\nu_{\rm max,\odot}}\right) \left(\frac{\Delta\nu}{\Delta\nu_\odot}\right)^{-2} \left(\frac{T_{\rm eff}}{T_{\rm eff,\odot}}\right)^{1/2} \text{and}$$

$$\frac{g}{g_\odot} \simeq \left(\frac{\nu_{\rm max}}{\nu_{\rm max,\odot}}\right) \left(\frac{T_{\rm eff}}{T_{\rm eff,\odot}}\right)^{1/2}$$

Figure 8: Scaling relations for Mass, Radius, and Surface Gravity

Using *Lightkurve*'s 'estimate_' feature, I plugged in mass, radius, and surface gravity–expressed in log space so the value was written as log g. All the values that were calculated could then be seen by inputting the object name.

```
Seismology(ID: KIC 6922244) - computed values:
 * numax: 1685.00 uHz (method: ACF2D)
 * deltanu: 78.92 uHz (method: ACF2D)
 * mass: 1.56 solMass (method: Uncorrected Scaling Relations)
 * radius: 1.66 solRad (method: Uncorrected Scaling Relations)
 * logg: 4.19 dex (method: Uncorrected Scaling Relations)
```

Figure 9: Seismology outputs: ν_{max}, $\Delta\nu$, mass, radius, and log g

3 Astrophysical Applications

The algorithm I used doesn't just help find planets, it also helps us learn more about what those planets and their stars are like. After finding the repeating dips in brightness that showed Kepler-8b passing in front of its star, I was able to use that data to find some basic physical properties of both the planet and the star.

For example, I was able to use the light curve of Kepler-8 to determine the orbital period of Kepler-8b, about 3.5 days. I was also able to find some of the star's physical properties by simply using a periodogram of the star's frequency and plugging in some values to the scaling relations 8.

This project has shown me how astronomers can take simple light curve data and use it to not only find stars' properties but also to see if they have orbiting planets. It helped me connect what we see in the data to what is actually happening in space.

4 Limitations and Challenges

While this project was a great way to explore how astronomers detect exoplanets, there were several limitations I ran into during the process. One of the biggest challenges was working with real telescope data, which can be noisy or have gaps due to spacecraft issues or observational limits. Even after using *Lightkurve*'s built-in tools to clean and flatten the light curve, there were still small variations in the data that made it harder to clearly see the transits at first.

Another limitation was that even though Kepler-8b was easy to find due to its larger size, many planets are harder to find to due either its smaller size or if they have irregular orbits. And since I didn't perform any model fitting or use detailed uncertainty analysis, my final results are more approximate rather than precise.

Finally, this method can only work if the planets pass directly in front of their star from our point of view. If a planet's orbit is tilted just a little, we might miss it entirely. This is a basic limitation of the transit method itself, and it shows that even powerful techniques have blind spots.

Despite these challenges, the project still gave me a deeper understanding of how planetary detection works and what makes analyzing space data both powerful and complex.

5 Conclusion

This project gave me hands-on experience with one of the most important techniques in exoplanet research: the transit method. By using the *Lightkurve* Python package to analyze data from the Kepler Space Telescope, I was able to detect the transits of Kepler-8b and explore the physical properties of both the planet and its host star. Through this process, I calculated key quantities like the planet's orbit and the star's mass, radius, and surface gravity, which helped me better understand how scientists turn light curve data into real information about distant worlds.

Along the way, I also encountered several limitations, such as data noise, reliance on visual inspection, and the inherent biases of the transit method. These challenges helped me appreciate the complexity of real astronomy research and why more advanced tools are necessary.

Overall, this project deepened my understanding of how exoplanets are discovered and studied, it also connected what I have learned throughout my college career to real-world applications. The project has also revealed to me just how powerful open-source tools like *Lightkurve* can be for exploring and understanding the universe.

References

[GSB+19] A. Ginsburg, B. M. Sipőcz, C. E. Brasseur, P. S. Cowperthwaite, M. W. Craig, C. Deil, J. Guillochon, G. Guzman, S. Liedtke, P. Lian Lim, K. E. Lockhart, M. Mommert, B. M. Morris, H. Norman, M. Parikh, M. V. Persson, T. P. Robitaille, J.-C. Segovia, L. P. Singer, E. J. Tollerud, M. de Val-Borro, I. Valtchanov, J. Woillez, The Astroquery collaboration, and a subset of the astropy collaboration. astroquery: An Astronomical Web-querying Package in Python. , 157:98, March 2019.

[LCH+18] Lightkurve Collaboration, J. V. d. M. Cardoso, C. Hedges, M. Gully-Santiago, N. Saunders, A. M. Cody, T. Barclay, O. Hall, S. Sagear, E. Turtelboom, J. Zhang, A. Tzanidakis, K. Mighell, J. Coughlin, K. Bell, Z. Berta-Thompson, P. Williams, J. Dotson, and G. Barentsen. Lightkurve: Kepler and TESS time series analysis in Python. Astrophysics Source Code Library, December 2018.

[OHa] Geert Barentsen Oliver Hall. How to estimate a star's mass and radius using asteroseismology.

[OHb] Geert Barentsen Oliver Hall. Using light curve files with lightkurve.

[Sau] Nicholas Saunders. Identifying transiting exoplanet signals in a light curve.