Nel Masamune

MACS 341 Scientific Computing

Dr. Kruse

May 13, 2025

Final Project: Calculating the Period of W UMa Eclipsing Binary Star Systems

Eclipsing binary star systems (EB) are a type of variable star system composed of two stars orbiting around a common center of mass (see figure 1) (Eclipsing Binary Variables).

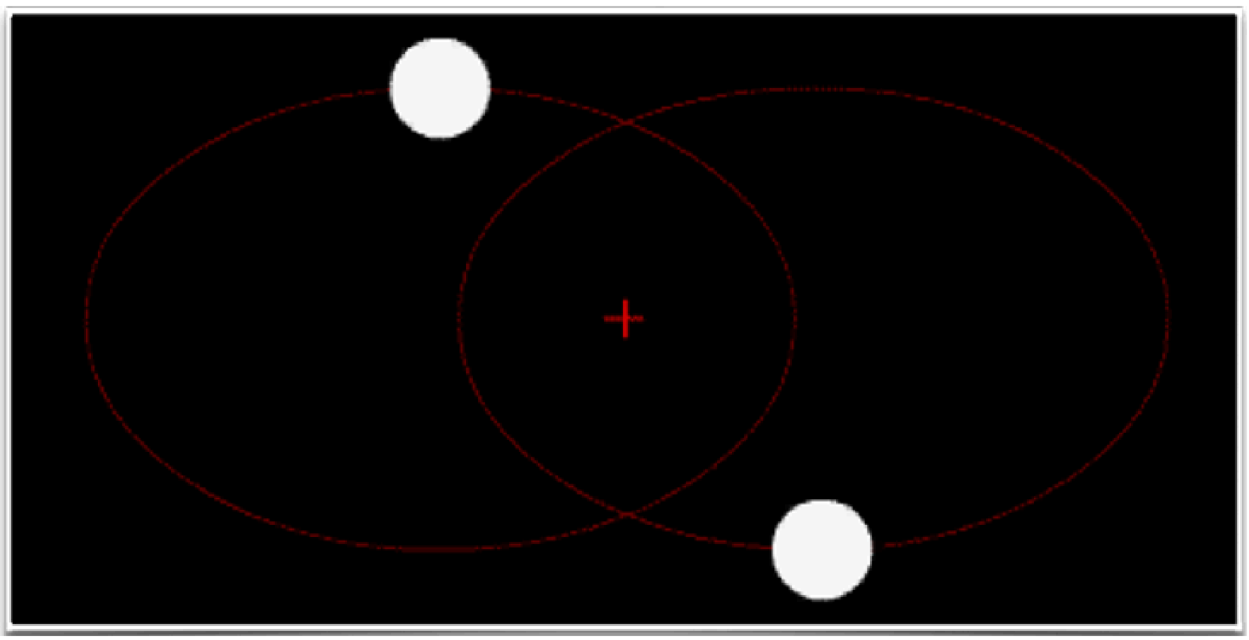

Figure 1. An eclipsing binary star system with orbits and center of mass marked. (Public domain, http://en.wikipedia.org/wiki/Binary_star#mediaviewer/File:Orbit5.gif)

These systems are incredibly common- around 50-70% of all stars are estimated to actually be pairs of stars. If their orbit is aligned with Earth's line of sight, then we can observe their eclipses through changes in the system's brightness, even if the individual stars are not resolvable. A system has maximum brightness when the two stars are un-eclipsed from Earth's point of view, and decreases in brightness when one star covers the other (Beaky). These systems are crucial in understanding stars in general, as they represent a unique opportunity to see stars interacting with each other (on average, stars are about 5 lightyears apart from each other in the Milky Way) (Pester). Observation of their interactions can be used to calculate their individual masses, which essentially allows you to derive almost anything about a star such as lifespan or temperature. By constructing a light curve of these systems, which is a graph of brightness over phase, you can also model these systems to determine star shape, luminosity, temperature, and more. EB's also hold significant historical significance as well: graphing these stars' luminosity compared to their mass allowed astronomers to develop the Hertzsprung-Russell Diagram, which shows a strong correlation between luminosity and mass, and essentially allowed astronomers to be able to predict the masses of single stars given only their luminosity (Beaky).

Despite the significant amount of research done, there are still many things that are unknown about EB's. One of these things is a phenomenon called the 'O'Connell Effect,' first written about in 1951 by Daniel J. K. O'Connell, for whom the phenomenon is named after (O'Connell). This effect refers to the asymmetry of the two maxima of a light curve, which occurs when the two stars are side-by-side (from Earth's perspective). This phenomenon is yet to be explained, though there are many possible theories (Akiba, et al).

Research at Juniata College focuses on a specific subset of short-period EB systems, called W Ursae Majoris (W UMa) types, that exhibit the O'Connell Effect, in an effort to add more data to databases and provide updates to changes in the differences in peak brightness. Quantifying this difference is a complicated task, with several different methods proposed on how to measure it (Wilsey and Beaky). A crucial step in the process is creating an accurate light curve from the raw images acquired overnight. Our analysis method currently takes close to half a day to fully complete, which has created a significant backlog in data. This is primarily due to the fact that we use several different programs, each with their own corresponding file types and systems that are incompatible with each other. With this final project, I worked to write software in Python that can fully replace one of these programs, which will help to streamline the process significantly. The vast majority of astronomical programming is done in Python, so my program can be used as a flexible, easy-to-integrate module in a complete analysis pipeline. In contrast, subscriptions to commercially available analysis software can be costly, and make it difficult for students to conduct research on their own personal machines.

The program I wrote analyzes the orbital periods of these star systems. While these systems have an already published period, W UMa systems have been observed to change periods. The stars that make up these systems are actually so close to each other that they are often in physical contact. This could potentially result in mass transfer, leading to a change in orbital velocity and changing their periods.

Calculating the orbital period of an EB can be difficult. It requires the use of a periodogram, which identifies the frequencies found in a dataset using Fourier analysis. There are further complications: since these periods are often six or more hours long, it is rare that we are able to get the entire light curve in one observation period (a single night). This means that when we graph our raw data, we often see several segments of the complete light curve scattered across the course of multiple nights (see figure 2). To tackle this, I used the Lomb Scargle periodogram, which is made specifically for finding the periods from datasets with uneven sampling (VanderPlas). As many astronomers face the same dilemma, there is a Lomb Scargle periodogram function conveniently built into Astropy, which is a massive python package written for astronomical data analysis (Installing Astropy).

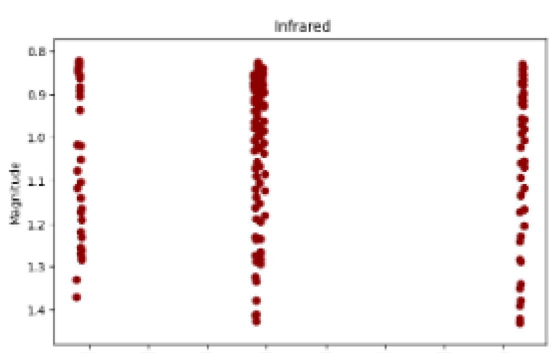

Figure 2. A graph of the raw data of the infrared filter of CW Cas, with magnitude on the y-axis and time on the x-axis.

The code begins by defining the star name and its period as published by the American Association of Variable Star Observers (AAVSO). I have uploaded the data set of the star CW Cas alongside the code as a sample set of data. CW Cas has an accepted period of 0.3188 days (VSX: Details for CW Cas). CW Cas is short for CW Cassiopeiae, meaning that this is a variable star found in the constellation Cassiopeia.

The dataset is a .npy file with several columns of data. This is a file produced by another program I wrote to combine the datasets of multiple nights of observation (which are kept in separate .csv files for each night), and to transfer relevant data. The original datasets include information that is not needed in later analysis such as the brightness of the control stars. The data that is transferred is the signal-to-noise ratio (SNR), the heliocentric Julian Date, the filter color, and the magnitude of our star system.

SNR essentially quantifies the quality of the data. The previous program also cuts data with an SNR less than 90, ensuring that low quality data, which could be caused by something like a cloud going over the star, is not included in our analysis. The heliocentric Julian Date is the time stamp with corrections made to account for the Earth's position relative to the Sun (British Astronomical Association). We also collect data through four different filter colors: blue, visual (green), red, and infrared. Finally, magnitude is the astronomical way to quantify brightness. What is referred to as brightness or magnitude later on is actually the difference in brightness between our system and some non-variable comparison star at the time the data point was taken. This is because the brightness of something depends largely on sky conditions at the time. By taking the differential brightness of our system and a non-variable star, we ensure that sky conditions do not contribute to the changes in brightness we see. It is important to note that the magnitude scale is “upside-down,” meaning that brighter objects have lower magnitudes. All of my plots have their y-axes flipped to reflect this.

A dictionary is used to store magnitude (mag) and time data. I found dictionaries the easiest way to store arrays of different lengths. First, the raw data is graphed, which is a simple scatter plot of brightness over time. This is done through the raw_graph function. Then, a Lomb Scargle analysis is run on the data, which outputs the frequency with the strongest power, which is done by the period_analysis function. This is run for each of the four filters, and an average is taken as I assumed that there the period should have no wavelength dependence. Period is calculated from 1/frequency. This period is then used to phase the data, using the equation:

$$\frac{JDhel - JDhel_{firstobservation}}{period} = phase$$

which is the standard way to calculate the phase (Akiba, et al). A correction was implemented to ensure that phase = 0 corresponds to the global minimum of the light curve, as is convention. This is done by subtracting the phase associated with the minima from all phase data points, and then adding 1 to all negative phase data points. This has the effect of shifting the graph until the minimum is the first data point, and then wrapping all the negative phases onto the end of the graph; all of this is done using the time_calc function. Finally, this phased data is graphed via the phased_graph function. All functions are wrapped into another function called 'graph_ls,' meaning you only have to run one function for all of this to run.

An issue I ran into while implementing this code was that sometimes the Lomb Scargle period would be incorrect, and when I graphed the phased light curve it would not show a complete light curve (figure 3). The CW Cas dataset I provided displays this behavior strongly. I realized I needed to also implement a way to adjust the period based on what visually looked correct, to ensure that the period used in later data analysis was accurate. This was done using tkinter's scale button and by embedding a graph which would continually update as the user moved the slider on the button, and classes so that the code updates the coordinates graphed, and does not output a new graph every time the scale slider is adjusted. This is done only on one filter color, red. The best period is then printed, and must be fed into the program when prompted. This is defined under the period_choosing function. Finally, I defined a function called revisgraph that outputs new phased light curves based on this new period.

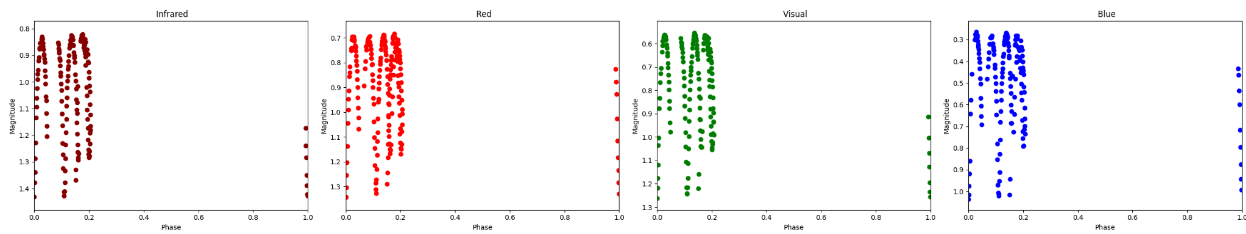

Figure 3. The graphs of magnitude versus phase calculated using the Lomb Scargle period. This is not the graph shape we would expect from a phased EB light curve, indicating that the calculated period is incorrect.

We can check that the code is working as expected, since the best period decided visually is also 0.3188 days, matching the AAVSO period. This also confirms that the Lomb Scargle did not calculate the correct period for CW Cas specifically, despite being implemented correctly. We can also see it in the final graphs of the phased light curve, which are shaped the way we would expect from a W UMa system (see figure 4).

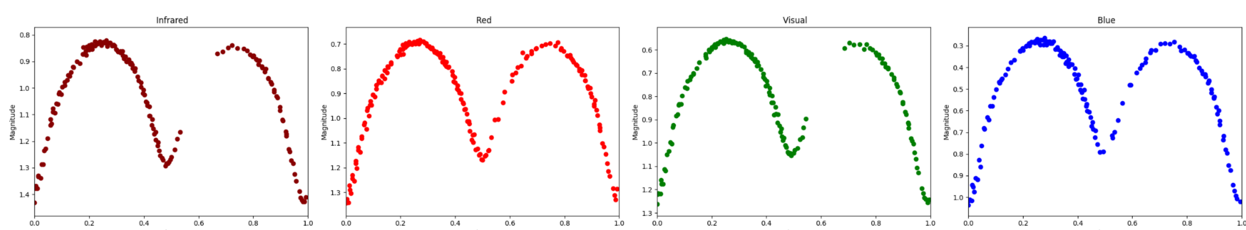

Figure 4. These are the phased graphs of magnitude versus phase calculated using the period obtained from visual inspection. These show the graph shapes we would expect from EB light curves, indicating that we have chosen the correct period. The gaps in the infrared and visual (green) curves indicate periods of missing data in these filters, possibly due to poor weather.

In conclusion, this program offers a flexible way to calculate the periods of star systems, and allows us to compare the current periods of stars compared to historical values, which is a crucial part of the creation of light curves of EB's. This will hopefully prove to be very useful in future analysis, as it will allow me to conduct data analysis without having to use the Astronomy lounge computer. The big advantages of this program are the fact that it allows multiple calculation methods, and that it is designed to function along with other python programs that I have written or plan to write, which should streamline our current data analysis process significantly. This could also be a disadvantage- if your computer is not organized the same way as mine, or you have differently formatted data, the program would require lots of editing to make it work like it should. As long as file creation, naming, and storage is kept standardized at Juniata College, this should not present as a problem for anyone.

Works Cited

Akiba, Tatsuya, et al. “Multi-filter Photometric Analysis of W Ursae Majoris (W UMa) Type Eclipsing Binary Stars KID 11405559 and V342 Boo.” *JAAVSO*, vol. 47, no. 2, Dec. 2019, p. 186.

Beaky, Matthew. “Eclipsing Binary Stars as Astrophysical Laboratories.” Bookend Seminar, Juniata College, 22 Jan. 2014.

“British Astronomical Association - Computing Section - Britastro.Org/Computing.” *BAA Computing Section*, britastro.org/computing/applets_dt.html. Accessed 13 May 2025.

D. J. K. O'Connell, The So-Called Periastron Effect in Eclipsing Binaries , Monthly Notices of the Royal Astronomical Society, Volume 111, Issue 6, December 1951, Page 642, https://doi.org/10.1093/mnras/111.6.642

“Eclipsing Binary Variables.” *Sensing the Dynamic Universe: Eclipsing Binaries*, lweb.cfa.harvard.edu/sdu/eclipsingbinaries.html. Accessed 13 May 2025.

“Install Astropy¶.” *Astropy*, www.astropy.org/. Accessed 13 May 2025.

Pester, Patrick. “How Far Apart Are Stars?” *LiveScience*, Purch, 17 September 2023, www.livescience.com/space/how-far-apart-are-stars.

Pickard, Roger. “Eclipsing Binary Observing Guide.” British Astronomical Association, 2011.

VanderPlas, Jacob T. “Understanding the lomb–scargle periodogram.” *The Astrophysical Journal Supplement Series*, vol. 236, no. 1, 1 May 2018, p. 16, https://doi.org/10.3847/1538-4365/aab766.

VSX : Detail for CW Cas, AAVSO, 2007, www.aavso.org/vsx/index.php?view=detail.top&oid =6454.

“W Ursae Majoris.” *Aavso*, www.aavso.org/vsots_wuma#:~:text=W%20UMa%20stars%20are %20easily,to%20the%20less%20massive%20one. Accessed 13 May 2025.

Wilsey, Nicholas, and Matthew Beaky. “Revisiting the O’Connell Effect in Eclipsing Binary Systems.” *The Society for Astronomical Sciences 28th Annual Symposium on Telescope Science*, May 2009, p. 107.