1 Task 1

The task was to create a folded light curve from the light curve data and determine the optimal parameters for a simple inverted boxcar transit model using the Metropolis-Hastings algorithm. The folded light curve was simple to make given that the period of 3.5485 days was provided (if it were not, a rough FFT or periodogram calculation would have sufficed). The model, $s_i(t_i|\Delta I, t_{min}, t_{max})$, is described by Equation 1 with everything outside the dip set to unity and the dip being represented by the height of the inverted boxcar. Figure 1 shows the comparison of the folded light curve data to the model with an initial guess at the parameters being $\Delta I = 0.007, t_{min} = 2, t_{max} = 2.3$.

$$\begin{cases} 1 & t < t_{min} \\ 1 - \Delta I & t_{min} \leq t \leq t_{max} \\ 1 & t > t_{max} \end{cases} \quad (1)$$
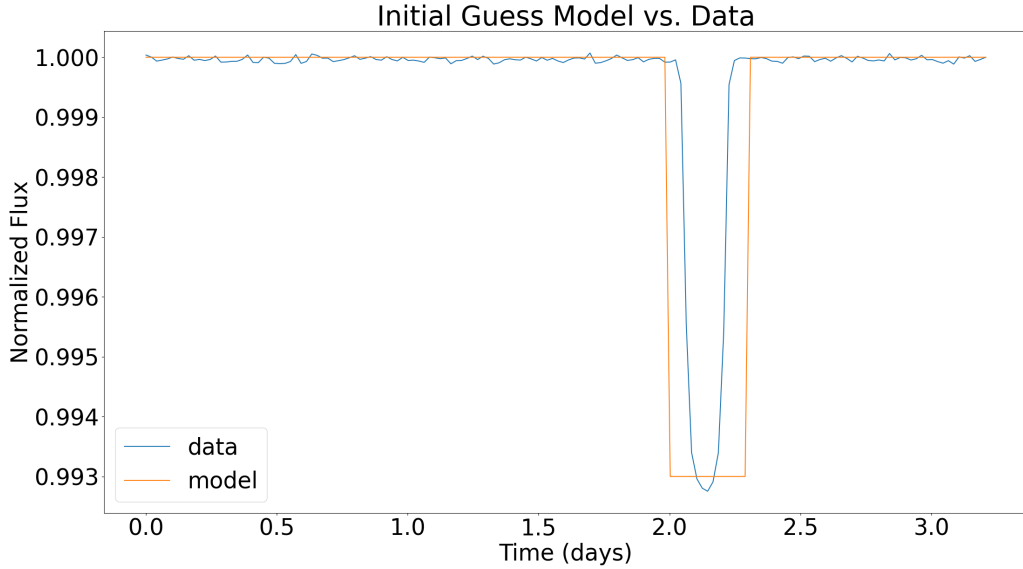

Figure 1: Comparison of model to data using initial guess at the transit depth, transit start, and transit end parameters. The model given the initial guess at the parameters seems to be a decent fit to the data.

In an attempt to improve the initial guess, the Metropolis-Hastings algorithm was employed to try and find a parameter combination which would better fit the model to the data. The first set of parameters chosen were the initial guess parameters, $\{\Delta I, t_{min}, t_{max}\}$. In order to generate new samples, a three-dimensional normal distribution was used as the proposal distribution, with the mean in each dimension centered

at the current value of each of the model parameters. With some checks to ensure that the proposed parameter values remained physical, three proposed values for the model parameters were drawn. Equation 2 was used to evaluate the likelihood of the model given the current and proposed parameters.

$$\mathcal{L}(\Delta I, t_{min}, t_{max}|t_i, I_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{I_i - s(t_i|\Delta I, t_{min}, t_{max})}{\sigma}\right)^2\right] \qquad (2)$$

In order to determine whether the proposed parameters would be accepted as an improvement to the model's representation of the data, the ratio r of the proposed and current parameter likelihoods (more generally, the Metropolis ratio) was compared. If $r > 1$, then the proposed parameters were immediately accepted as this implied that the proposed paremeters were a considerable improvement of the model.

$$r = \frac{\mathcal{L}(\Delta I_{proposed}, t_{min,proposed}, t_{max,proposed}|t_i, I_i)}{\mathcal{L}(\Delta I_{current}, t_{min,current}, t_{max,current}|t_i, I_i)} \qquad (3)$$

If $r < 1$, the proposed parameters still had the possibility of being selected by comparing the value of r to a draw from a uniform distribution ranging from 0 to 1. If the value drawn from the uniform distribution was **less than** the value of r, the proposed parameters were accepted. Otherwise, the current values of the parameters remained unchanged, and a new set of proposed parameters was chosen from the aforementioned normal distribution. It is important to note that if the proposed parameters are chosen, subsequent proposed paramaters are chosen from a three-dimensional normal distribution centered at the values of the current iteration of the parameters, which should (ideally) not be the same as the initial guess values.

After attempting the above method for 1000 iterations, the "best" parameters identified for my model were far worse than the initial guess and created the model shown in Figure 4. I was unable to diagnose where the problem arose, but my guess is that something involving the likelihood was not functioning properly.

In particular, the histogram of the values for the transit depth is shown in Figure 2, with the evolution of the transit depth value shown in Figure 3. Both of these show a large spread away from the initial guess of the transit depth which implies poor representation of the actual improvement to the model given some new proposed parameters.

As such, the M-H algorithm failed to identify new parameters, however if we use Equation 4 with the initial guess for the dip being $\Delta I = 0.007$, a normalized intensity $I = 1$, and the radius of Kepler-5 being $R_{star} = 1.79R_\odot$, we get out that $R_{planet} \approx 0.1498R_\odot$ which is about $1.498R_{jup}$. This is reasonably close to the cited value of $1.431R_{jup}$.

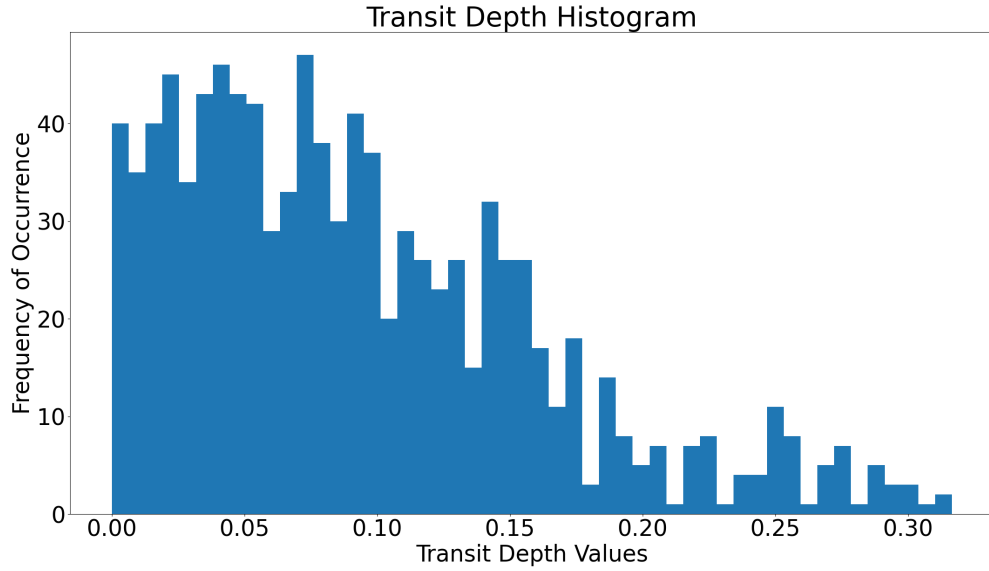$$R_{planet} = \sqrt{\frac{R_{star}^2 \Delta I}{I}} \qquad (4)$$

Figure 2: Histogram of transit depth values explored by the M-H algorithm. These deviate very far from what the transit depth should be...
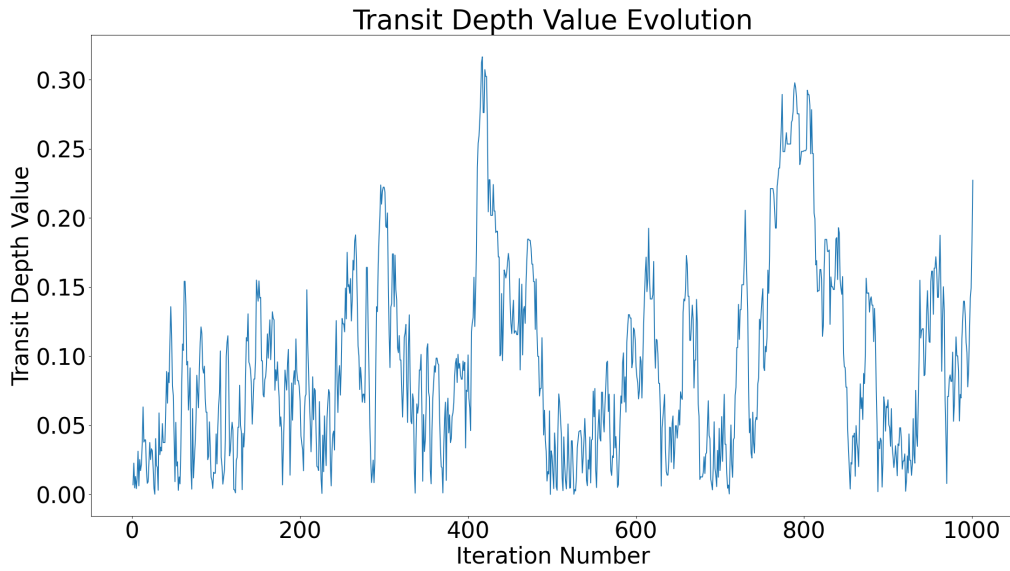

Figure 3: Evolution of the transit depth values explored by the M-H algorithm. The lack of convergence implies that the likelihood is not representing how well proposed parameters are improving the fit of the model to the data.
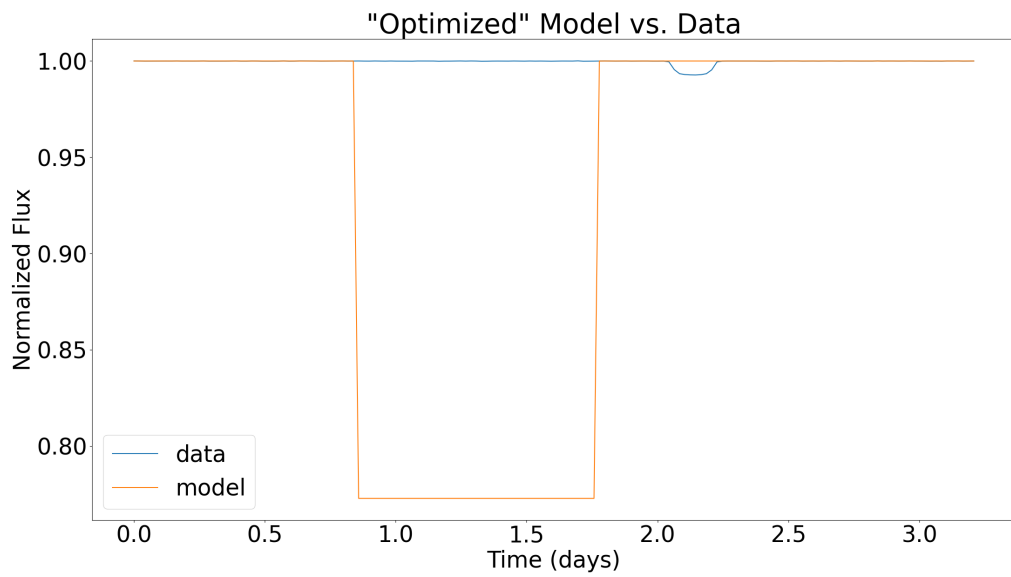
Figure 4: Model with "optimized" parameters at the end of the M-H algorithm. The model is clearly worse than the initial guess.

2 Bonus Task

Even though my MCMC was not working, I wanted to attempt the bonus task. For the calculation in my **tasks.py** bonus task section, I chose parameters by eye which decently fit a sine wave to the folded radial velocity data. Extracting the velocity of the star to be roughly 265 m/s allowed for a linear momentum calculation for the star. The planet had to have the same linear momentum, but its mass and velocity were unknown. By using Kepler's Third Law, the orbital radius was found, from which the planet's velocity (assuming a circular orbit) was determined. This left the only unsolved variable as the planet's mass which, when combined with the planet's radius determined in Task 1, produced a density of 1.073 g/cm^3. This density puts the planet in the gas giant regime with the highest semblance to Jupiter and Uranus. Considering the radius of the star is roughly 1.5 times that of Jupiter and the orbital period is only roughly 3.5 days, this seems like a strong candidate for a hot Jupiter-like planet.
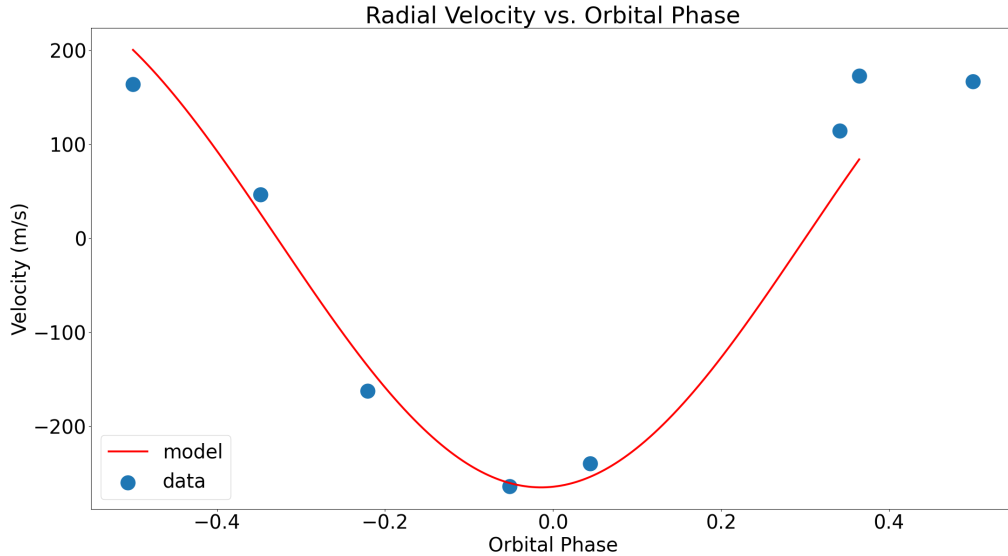

Figure 5: Folded radial velocity curve showing the star's velocity toward (negative) and away from (positive) the observer as a function of orbital phase.