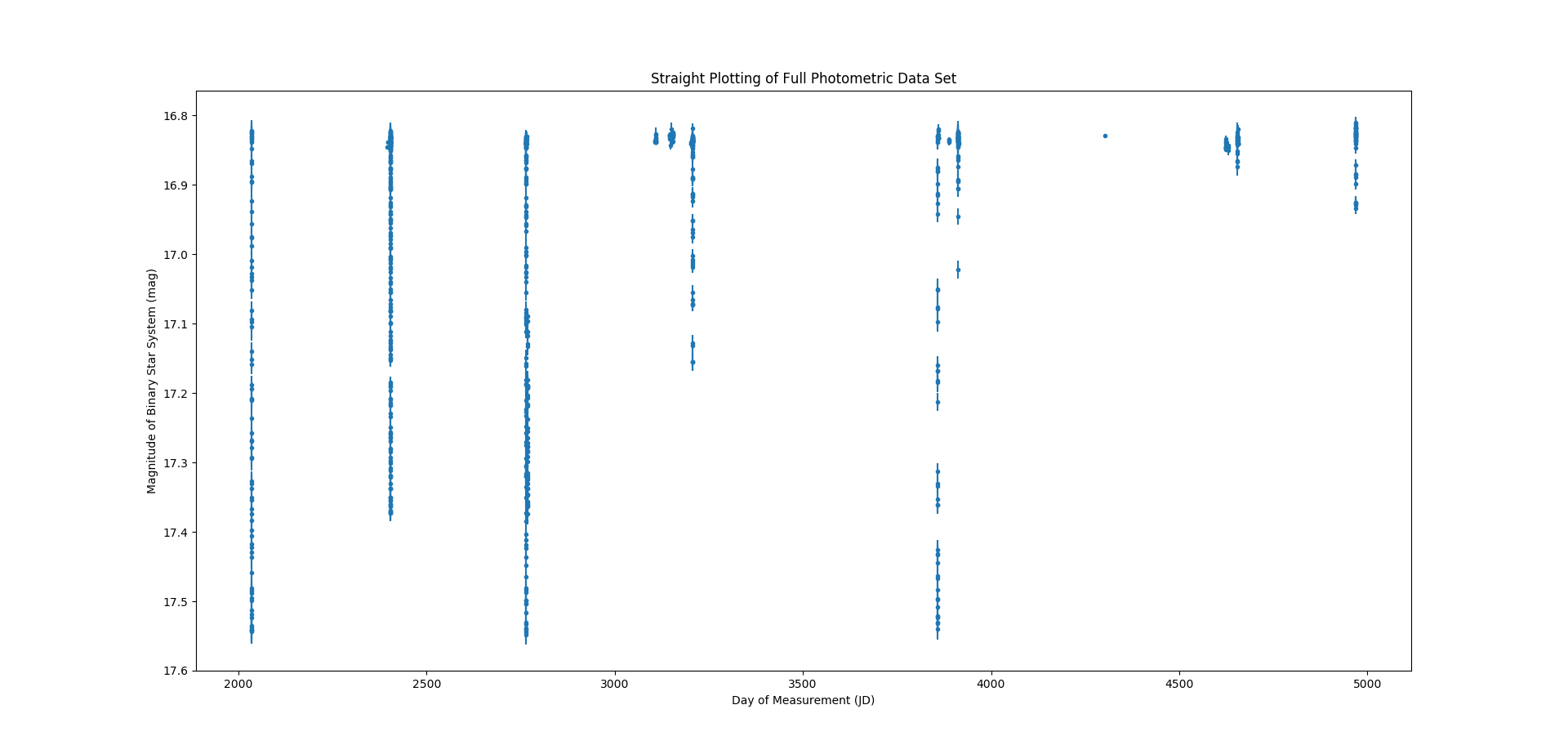
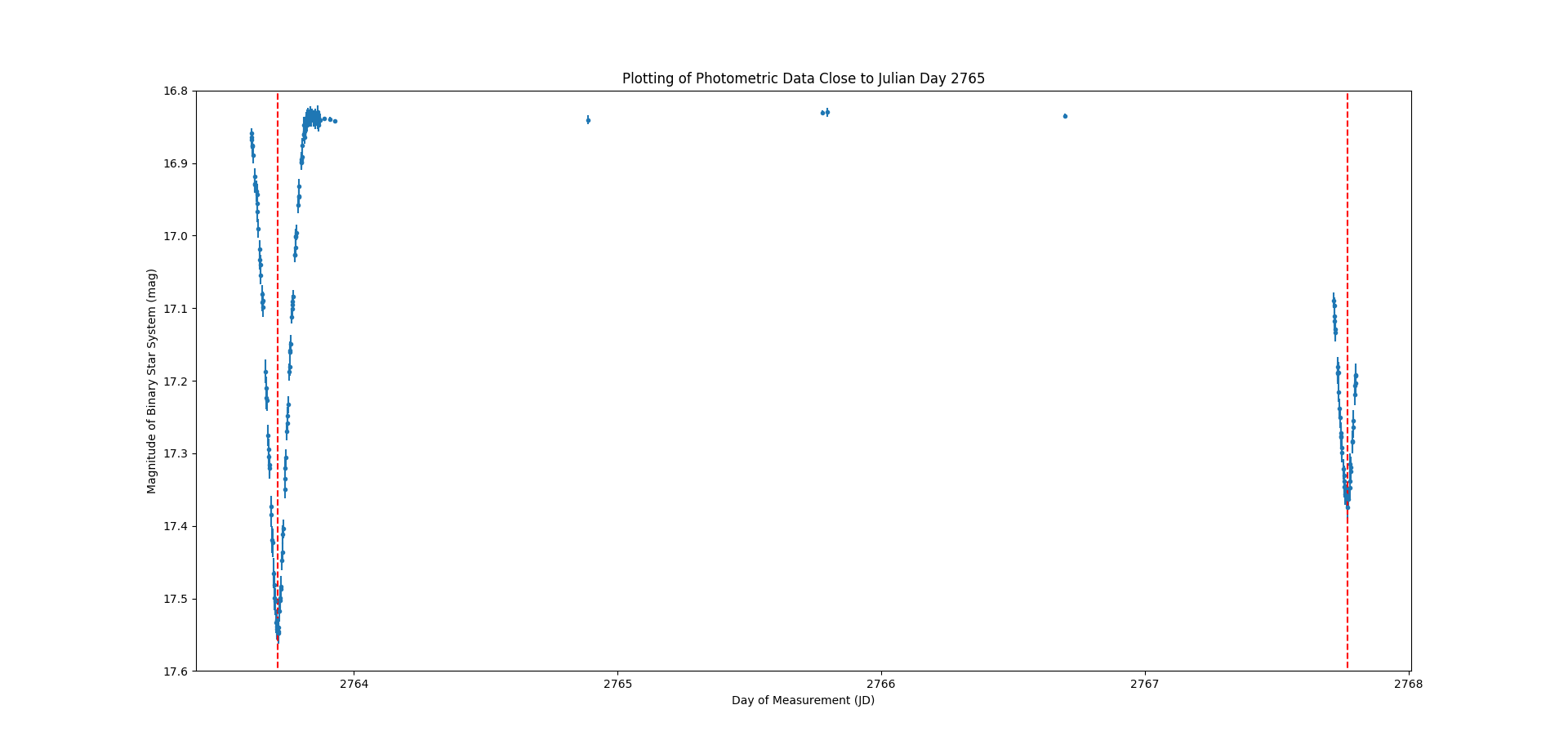
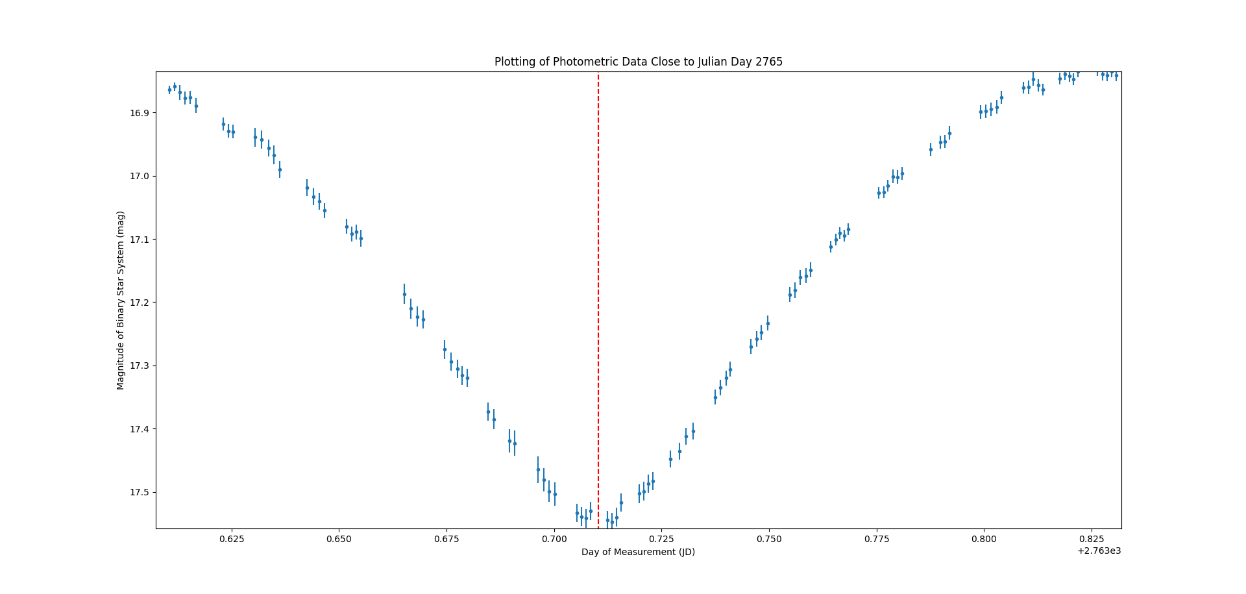
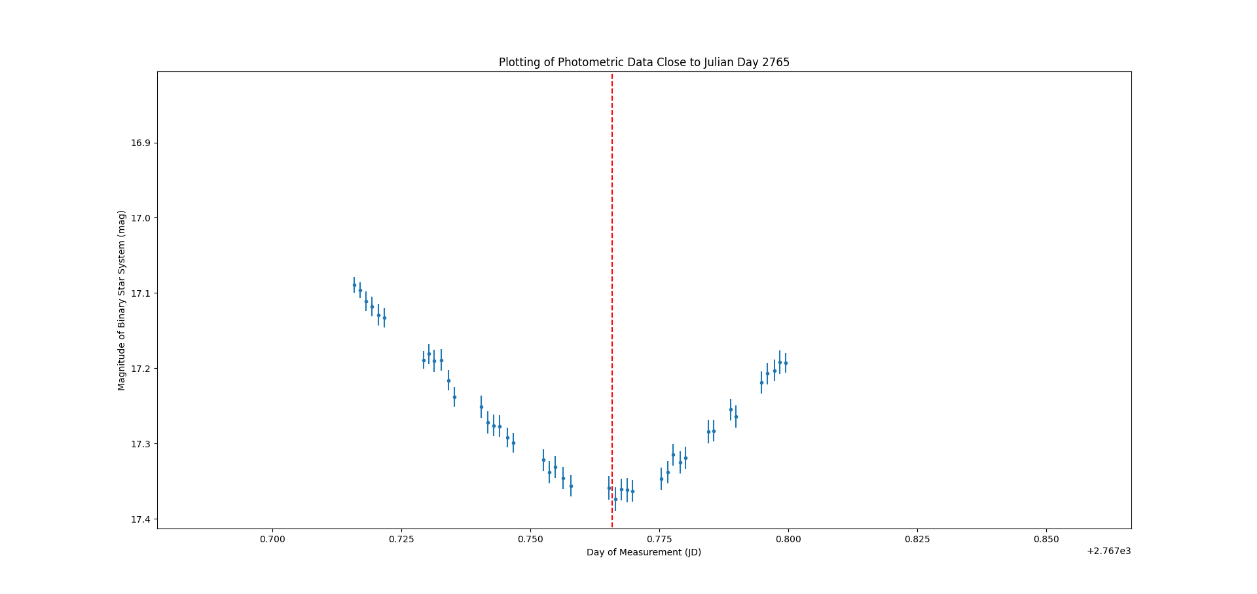
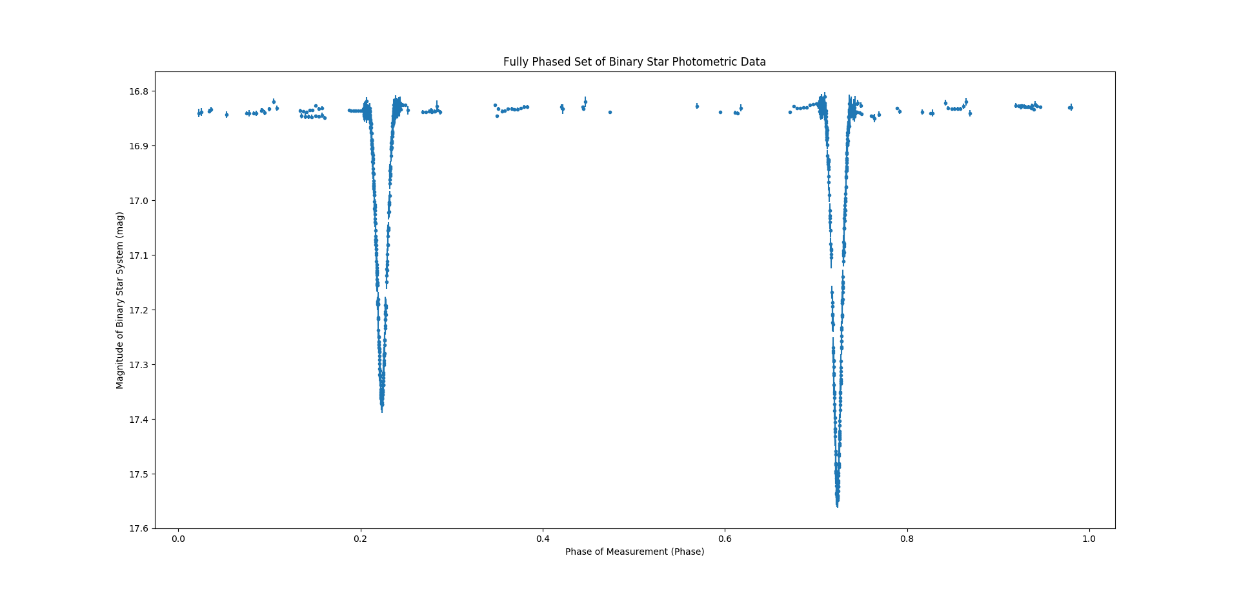
Astronomy 205: Homework 3

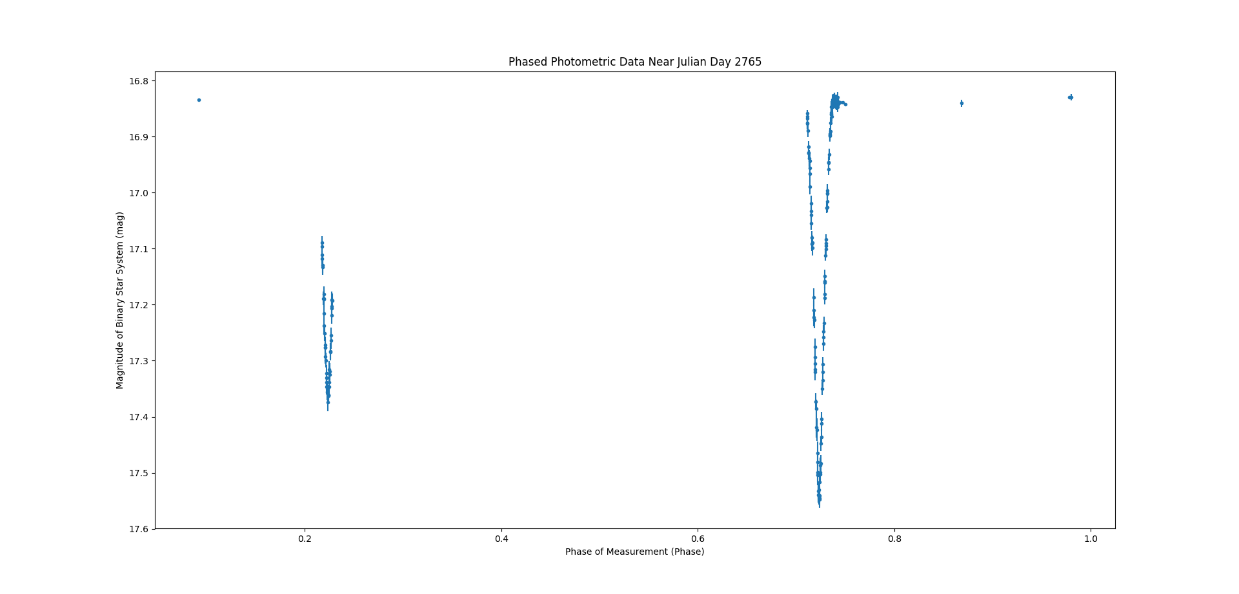
1. We begin this assignment by opening our photometric data for the binary star system. We can plot the data as it is given to us to see what our data set it.

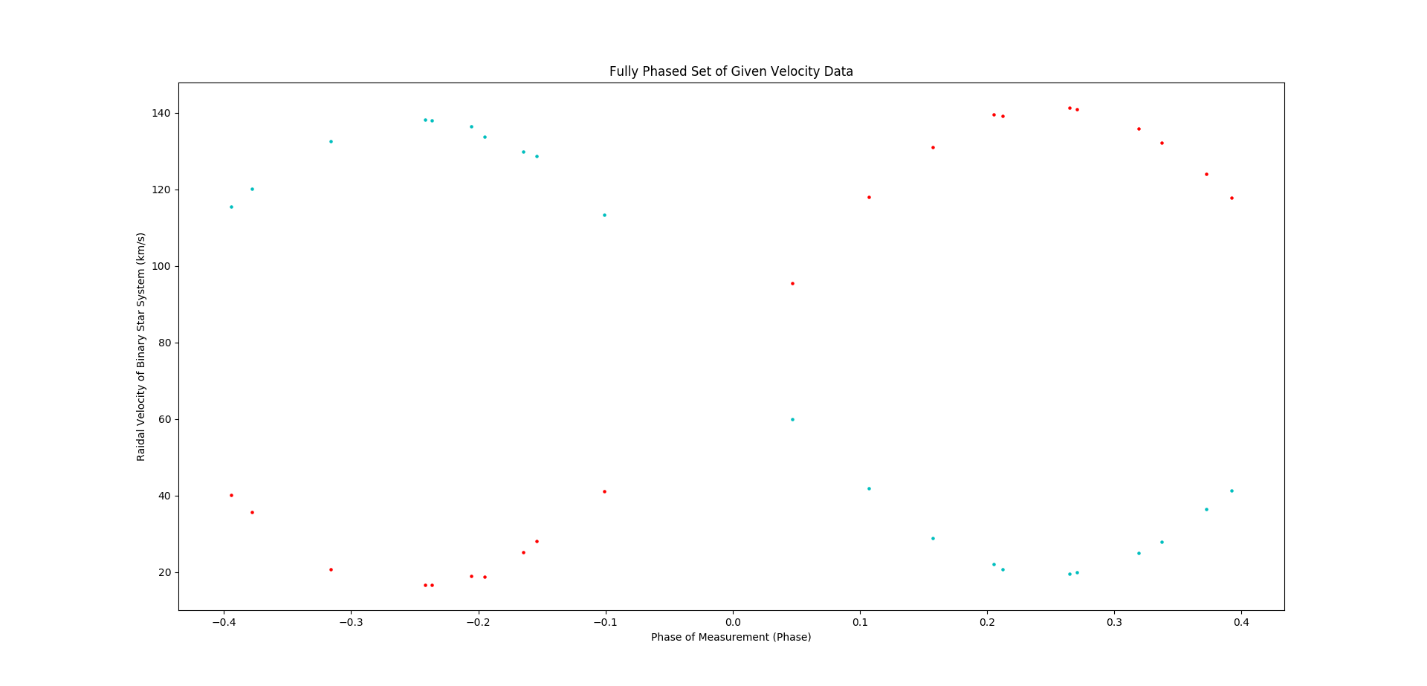
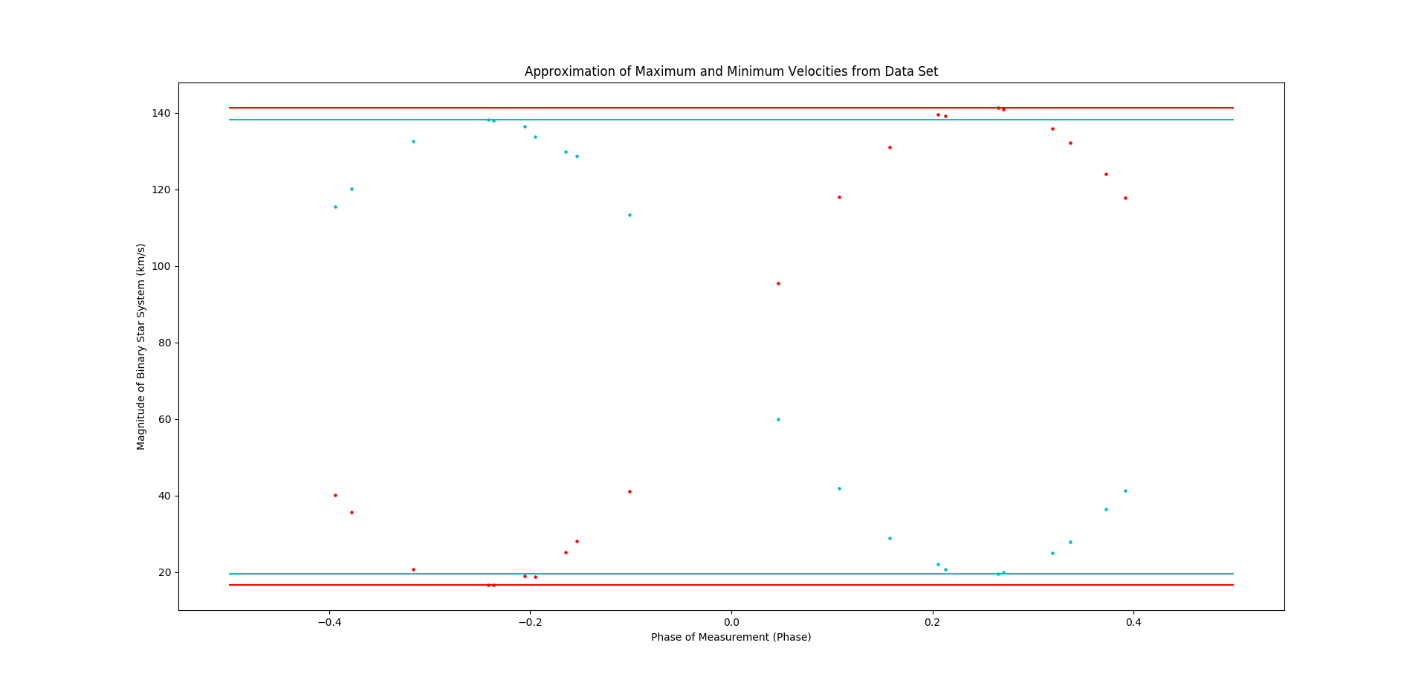
To phase this data, we need an approximation of the period. The period can be found as the time between two eclipses. Thus, we need a period that shows both eclipses after semi-continuous observation. The data near Julian Day 2765 is one such period. We can graph just the part around this time to see its period:

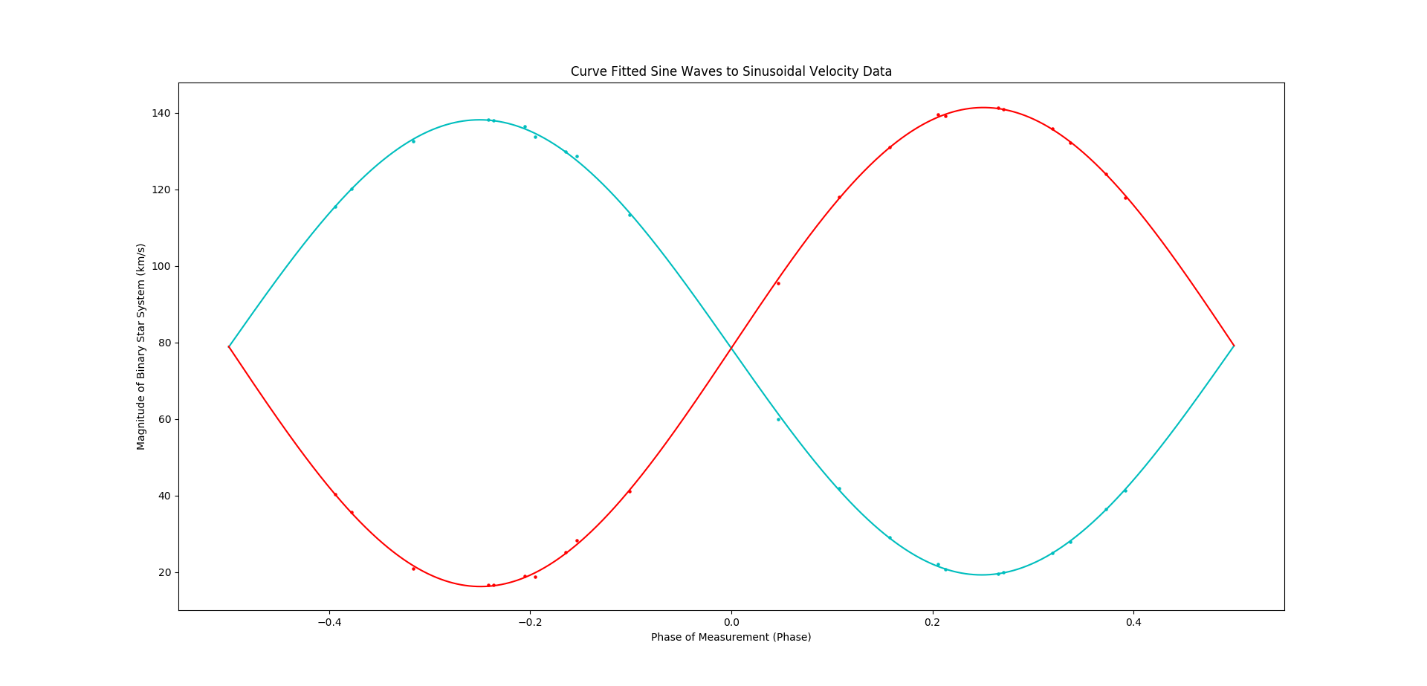
Then, by approximating the midpoints of the eclipse we can find the half-period of the system. From the above, we approximated the first eclipse to reach full coverage around JD2763.71025 and the second eclipse to reach full coverage around JD2767.7659:



Thus, the full period is 8.1113 days. With this period, we can phase the data as what fraction of the period is was in when the data was taken:



1. The second portion of the assignment looks at the velocity data for the binary star system that we were given. The data for the velocity was already properly phased. When each of the velocities are plotted with respect to the phase, we get the following:
2. Lastly, we want to derive as many quantities describing the binary star system using our two data sets. We can start with basic calculations, including the maximum velocities and the velocity of the center of mass. To approximate this, we curve fit sine waves to each of the velocity data sets. The maxima and minima associated with the curve can be got from the amplitude of the wave. Furthermore, the center of mass speed is equivalent to the offset parameter of the equation. Then, we approximated reasonable values and confirmed them with the curve-fit:

The code gave us an amplitude of 59.496km/s and 62.593km/s for star 1 and star 2, respectively. The curve fit also gave use an approximation of the center of mass velocity of the binary star system. We averaged the values to get a center of mass velocity of 78.766km/s.

Here we must then start to make assumptions. Firstly, the maximum radial velocities are close enough for us to simply consider them in a circular orbit. Secondly, we are not sure of the inclination of the binary star system. This thus imposes the term “minimum” on any value that we derive. It can also be stated that the value should be multiplied by 1/cos(i) depending on the inclination (1/cos^3(i)) in the case of mass. Keeping this in mind, we will simply refer to all derived values as being minimums.

Now we can get the semi-major axes of the star system. We can find the semi-major axes as follows from a relationship with velocity:

Giving us the following values for the minimum semi-major axis of star 1 and star 2, respectively, 6633000km and 6982000km.  
 Next, we can look for the masses of the stars. Recall the formulae relating the ratio of masses and the sum of the masses:

Thus:

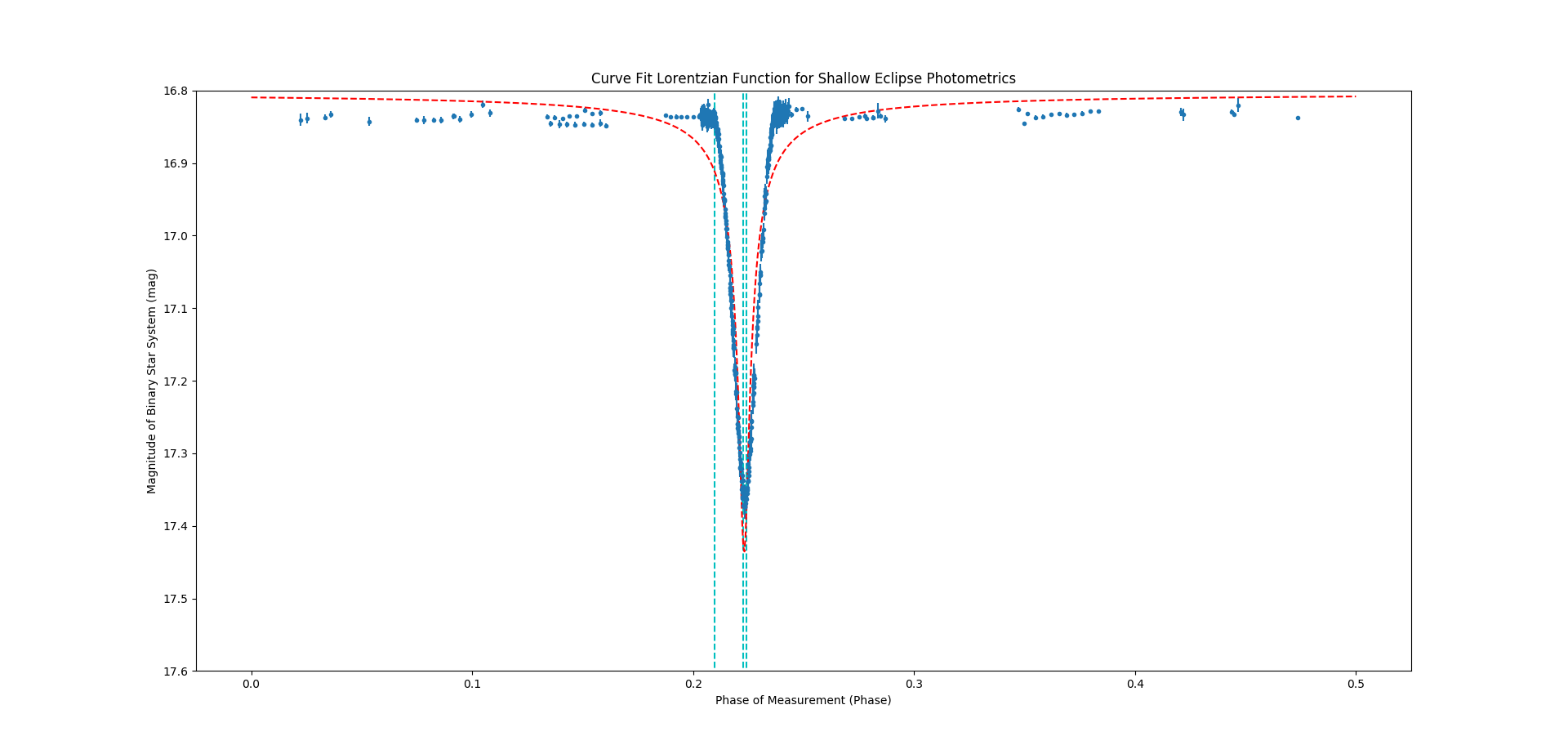
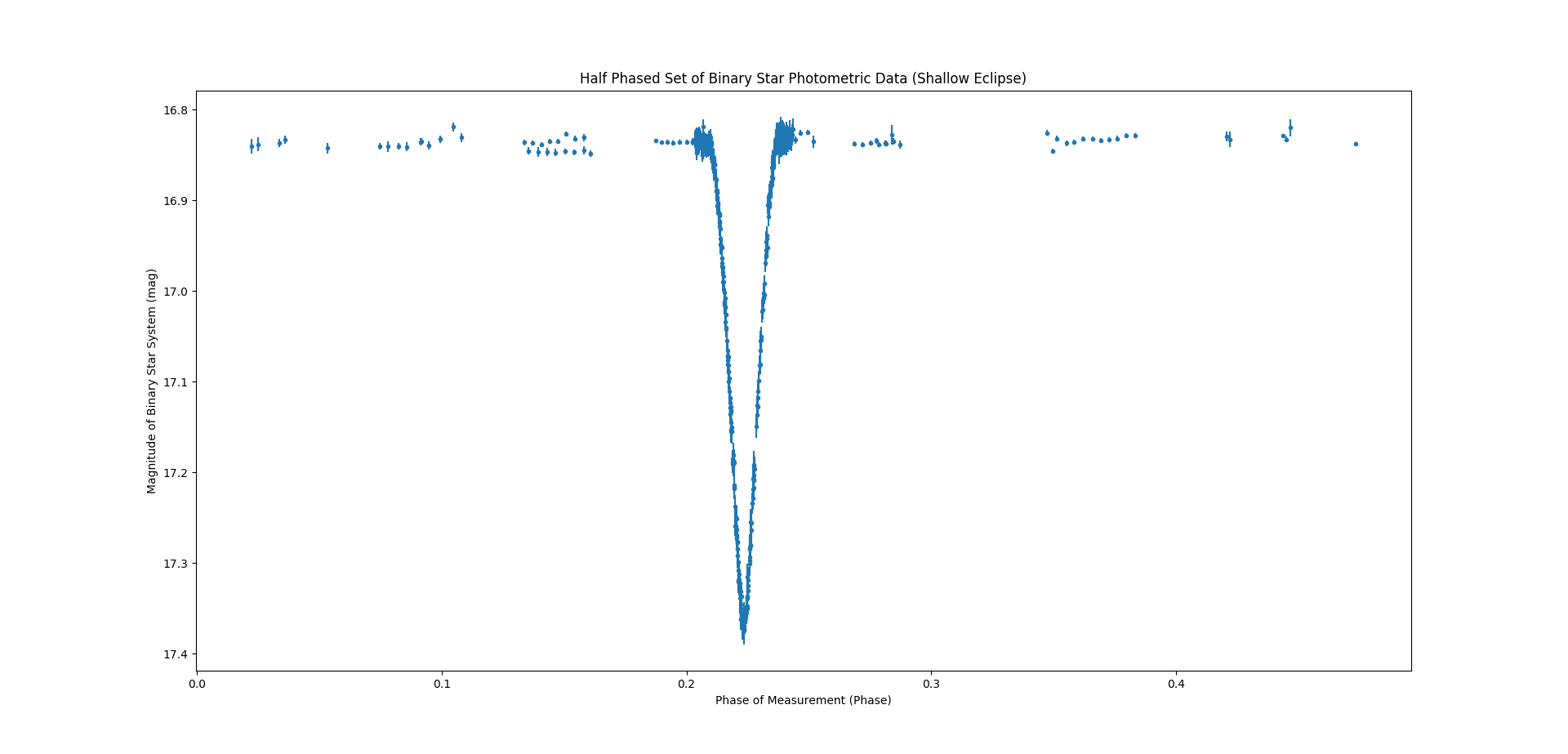
Plugging in our values of velocity gives us the masses of star 1 and star 2, respectively, as 1.559e+30kg and 1.481e+30kg. These values occur with a factor of sine cubed i for their true mass.

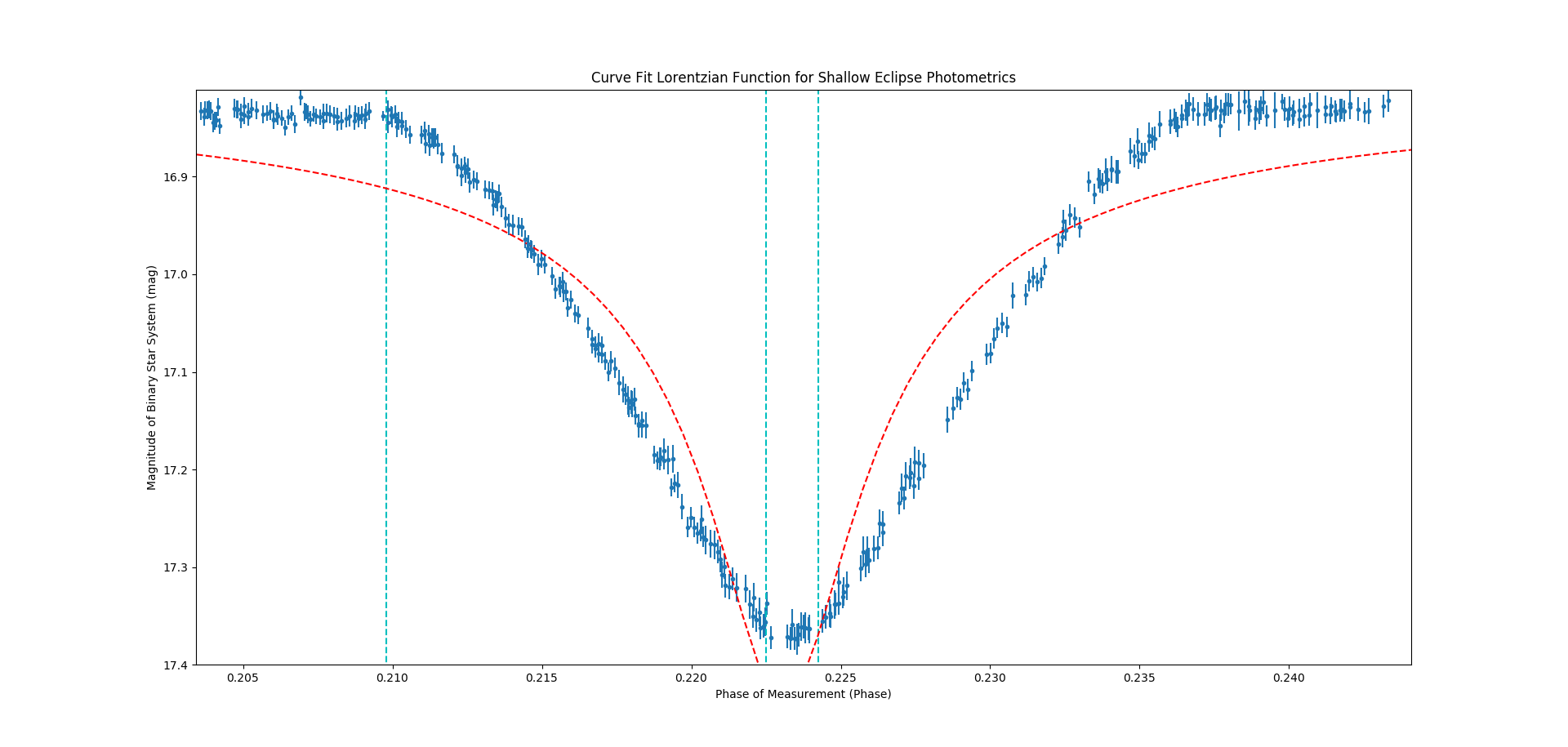
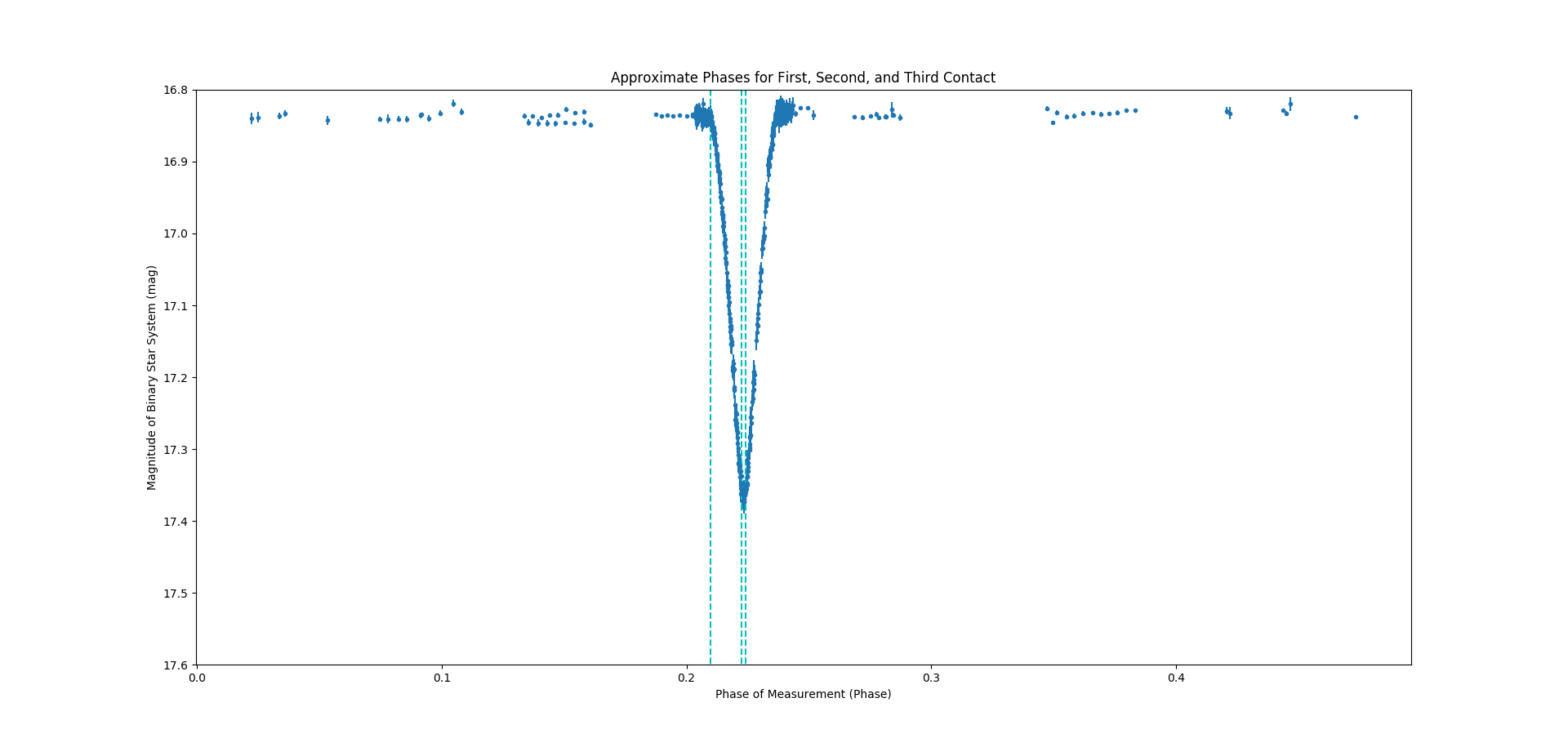
Finally, we need stellar radius. We can get this by observing the photometric data again. Looking at the photometric data, we can discern four major turning points in the curve. There are two at the top where the luminosity begins to dip and there are two at the bottom where the luminosity seems to flatten off. Physically we can describe these points as the four points of contact. The points can be described as such:

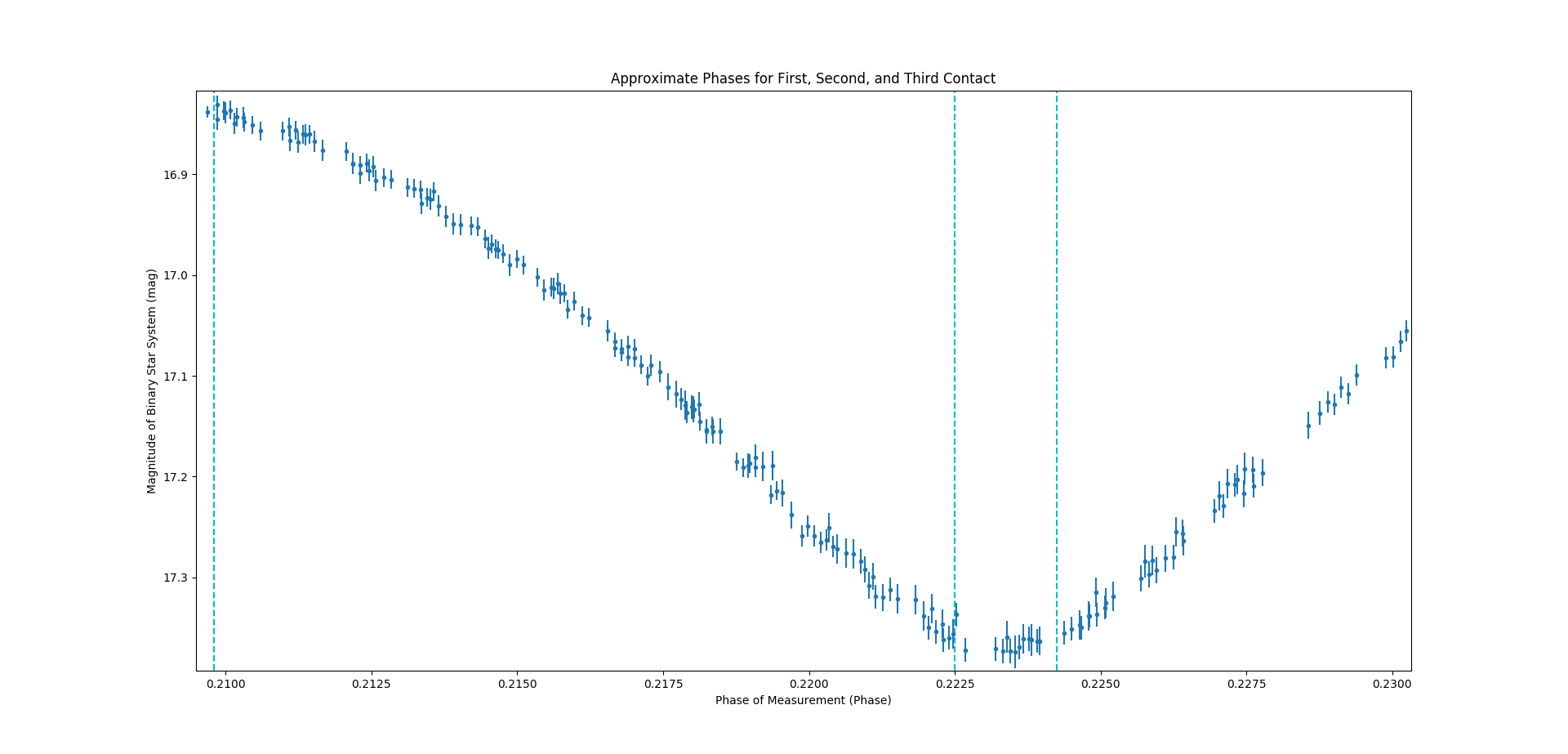
In the photo above, with respect to the orange background star, a foreground star in the grey position correspond to first contact, when the star first begins to pass in front of the background star. A foreground star in the yellow position corresponds to second contact, when the star first begins to leave the edge of the background star. A foreground star in the green position corresponds to third contact, when the star has passed entirely over the background star and reached the other edge, ready to end its transit. Finally, a foreground star in the blue position corresponds to fourth contact, when the star eventually leaves the outer edge of the background star and concludes its transit.

If we focus on a point on the foreground stars right edge, we can tell that the point must travel the diameter of the background star between the first and third contact and must travel the diameter of the foreground star between the first and second contact. Then, we must approximate the time for contacts using the photometric data set.

To do this, we wanted to observe the shape of the photometric data. Due to its curve that rises out of flatness, we decided to curve fit a Lorentzian distribution to the data. We know that the fit is going to be bad, but we only want to observe a precise point where the data no longer follows a curve but flattens off. We used half of the data so the Lorentzian does not get confused with a second dip. We chose the shallower luminosity to fit:

The above graph shows our data, Lorentzian curve fit, and approximations of contact. Notice from this angle, the Lorentzian curve is attempting to fit the graph, but it is failing near the top of the graph and the bottom of the dip. We can zoom into the graph to see a bit better:

Again, in the above graph, the cyan lines are our approximations of points of contact and our red line is our Lorentzian curve fit. Notice that we picked places that showed an obvious difference in the path of the luminosity curve or where the data reached a flatter state where the Lorentzian continued below. We can get a better look at the contacts below:

Find the time between these points and multiply by the relative velocity (the sum of the velocities). The multiply each of our phases by the period in seconds, then by the relative velocity, and then divide by two, so that we get radius back. We get radii of 625500km and 543200km for star 1 and 2, respectively. We found which star has which radii by making assumptions about the star’s classification. From the masses and radii, it seems that these stars are approximately sun-like stars and thus the higher mass star has the higher radius.

The full data for the binary system is as follows:

**import numpy as np**

**import matplotlib.pyplot as plt**

**from scipy.optimize import curve\_fit as cf**

**def velocityWave(phase, amplitude, frequency, shift, offset):**

**return amplitude\*np.sin(frequency\*phase+shift)+offset**

**def solveMasses(period, velocity1, velocity2, semimajor1, semimajor2):**

**GravCon = 6.674\*(10\*\*(-20))**

**VelSum = velocity1+velocity2**

**SmaSum = semimajor1+semimajor2**

**Mass1 = semimajor2\*period\*86400\*(VelSum\*\*3)/(2\*np.pi\*SmaSum\*GravCon)**

**Mass2 = period\*86400\*(VelSum\*\*3)/(2\*np.pi\*GravCon)-Mass1**

**return Mass1, Mass2**

**def Lorentzian(Day, LMax, Delta, FullEclipse, Loff):**

**return LMax/((Delta\*\*2+(Day - FullEclipse)\*\*2)\*\*(1/2))+Loff**

**filename = "C:/Users/ryank/Desktop/Work/Classes/Python/ASTR205/Data/"**

**filename += "VBandData.txt"**

**PhotometricData = np.loadtxt(filename)**

**filename = "C:/Users/ryank/Desktop/Work/Classes/Python/ASTR205/Data/"**

**filename += "VelocityObservations.txt"**

**VelocityData = np.loadtxt(filename)**

**###############################################################################**

**plt.gca().invert\_yaxis()**

**plt.errorbar(PhotometricData[:, 0], PhotometricData[:, 1],**

**yerr=PhotometricData[:, 2], marker='.', linestyle='')**

**plt.title("Straight Plotting of Full Photometric Data Set")**

**plt.xlabel("Day of Measurement (JD)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**Data2770 = PhotometricData[np.where(PhotometricData[:, 0]<=3000), :][0]**

**Data2770 = Data2770[np.where(Data2770[:, 0]>=2500), :][0]**

**plt.gca().invert\_yaxis()**

**plt.errorbar(Data2770[:, 0], Data2770[:, 1], yerr=Data2770[:, 2], marker='.',**

**linestyle='')**

**plt.vlines([2767.7659, 2763.71025], 0, 20, colors='r', linestyles='--')**

**plt.ylim(17.6, 16.8)**

**plt.title("Plotting of Photometric Data Close to Julian Day 2765")**

**plt.xlabel("Day of Measurement (JD)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**period = 2\*(2767.7659-2763.71025)**

**Phase2770 = Data2770[:, 0]/period**

**Phase2770 = Phase2770%1**

**plt.gca().invert\_yaxis()**

**plt.errorbar(Phase2770, Data2770[:, 1], yerr=Data2770[:, 2], marker='.',**

**linestyle='')**

**plt.title("Phased Photometric Data Near Julian Day 2765")**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**FullPhase = PhotometricData[:, 0]/period**

**FullPhase = FullPhase%1**

**plt.gca().invert\_yaxis()**

**plt.errorbar(FullPhase, PhotometricData[:, 1], yerr=PhotometricData[:, 2],**

**marker='.', linestyle='')**

**plt.title("Fully Phased Set of Binary Star Photometric Data")**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**HalfData = PhotometricData[np.where(FullPhase<=0.5), :][0]**

**HalfPhase = FullPhase[np.where(FullPhase<=0.5)]**

**plt.gca().invert\_yaxis()**

**plt.errorbar(HalfPhase, HalfData[:, 1], yerr=HalfData[:, 2], marker='.',**

**linestyle='')**

**plt.title("Half Phased Set of Binary Star Photometric Data (Shallow Eclipse)")**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**initialLorentz = [0.001, 0.002, 0.22, 16.8]**

**testPhase = np.linspace(0, 0.5, 1000)**

**plt.gca().invert\_yaxis()**

**plt.errorbar(HalfPhase, HalfData[:, 1], yerr=HalfData[:, 2],**

**marker='.', linestyle='')**

**plt.plot(testPhase, Lorentzian(testPhase, \*initialLorentz), c='r',**

**linestyle='--')**

**plt.title("Initial Guess of Lorentzian Fit for Shallow Eclipse Photometrics")**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**LorentzParams, LorentzCOV = cf(Lorentzian, HalfPhase, HalfData[:, 1],**

**sigma=HalfData[:, 2], p0=initialLorentz,**

**maxfev=10\*\*5)**

**GuessPhases = [0.2098, 0.2225, 0.22425]**

**Distance1 = -(0.2098-0.224425)\*period\*86400**

**Distance2 = -(0.2098-0.2225)\*period\*86400**

**plt.gca().invert\_yaxis()**

**plt.errorbar(HalfPhase, HalfData[:, 1], yerr=HalfData[:, 2], marker='.',**

**linestyle='')**

**plt.plot(testPhase, Lorentzian(testPhase, \*LorentzParams), c='r',**

**linestyle='--')**

**plt.vlines(GuessPhases, 0, 20, colors='c', linestyle='--')**

**plt.title("Curve Fit Lorentzian Function for Shallow Eclipse Photometrics")**

**plt.ylim(17.6, 16.8)**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**plt.gca().invert\_yaxis()**

**plt.errorbar(HalfPhase, HalfData[:, 1], yerr=HalfData[:, 2], marker='.',**

**linestyle='')**

**plt.vlines(GuessPhases, 0, 20, colors='c', linestyle='--')**

**plt.title("Approximate Phases for First, Second, and Third Contact")**

**plt.ylim(17.6, 16.8)**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (mag)")**

**plt.show()**

**###############################################################################**

**plt.scatter(VelocityData[:, 3], VelocityData[:, 1], s=5, c='c')**

**plt.scatter(VelocityData[:, 3], VelocityData[:, 2], s=5, c='r')**

**plt.title("Fully Phased Set of Given Velocity Data")**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Raidal Velocity of Binary Star System (km/s)")**

**plt.show()**

**FirstMax = max(VelocityData[:, 1])**

**FirstMin = min(VelocityData[:, 1])**

**SecondMax = max(VelocityData[:, 2])**

**SecondMin = min(VelocityData[:, 2])**

**MaxMins = [FirstMax, FirstMin, SecondMax, SecondMin]**

**plt.scatter(VelocityData[:, 3], VelocityData[:, 1], s=5, c='c')**

**plt.scatter(VelocityData[:, 3], VelocityData[:, 2], s=5, c='r')**

**plt.hlines(MaxMins, -0.5, 0.5, colors=['c', 'c', 'r', 'r'])**

**plt.title("Approximation of Maximum and Minimum Velocities from Data Set")**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (km/s)")**

**plt.show()**

**initial = [(FirstMin-FirstMax)/2, 2\*np.pi, 0.5, (FirstMin-FirstMax)/2+FirstMin]**

**FirstParams, FirstCov = cf(velocityWave, VelocityData[:, 3],**

**VelocityData[:, 1], p0=initial, maxfev=10\*\*5)**

**SecondParams, SecondCov = cf(velocityWave, VelocityData[:, 3],**

**VelocityData[:, 2], p0=initial, maxfev=10\*\*5)**

**TestX = np.linspace(-0.5,0.5,1000)**

**FirstTestY = velocityWave(TestX, \*FirstParams)**

**SecondTestY = velocityWave(TestX, \*SecondParams)**

**plt.scatter(VelocityData[:, 3], VelocityData[:, 1], s=5, c='c')**

**plt.scatter(VelocityData[:, 3], VelocityData[:, 2], s=5, c='r')**

**plt.title("Curve Fitted Sine Waves to Sinusoidal Velocity Data")**

**plt.plot(TestX, FirstTestY, c='c')**

**plt.plot(TestX, SecondTestY, c='r')**

**plt.xlabel("Phase of Measurement (Phase)")**

**plt.ylabel("Magnitude of Binary Star System (km/s)")**

**plt.show()**

**###############################################################################**

**PER = period**

**COMVEL = (SecondParams[3]+FirstParams[3])/2**

**VELStar1 = abs(FirstParams[0])**

**VELStar2 = abs(SecondParams[0])**

**RATIO = VELStar1/VELStar2**

**SMAStar1 = VELStar1\*period\*86400/(2\*np.pi)**

**SMAStar2 = VELStar2\*period\*86400/(2\*np.pi)**

**MASStar1, MASStar2 = solveMasses(PER, VELStar1, VELStar2, SMAStar1, SMAStar2)**

**RADStar1 = (Distance1\*(VELStar1+VELStar2))/2**

**RADStar2 = (Distance2\*(VELStar1+VELStar2))/2**

**PERString = "The period of the orbit is {:.4f} days."**

**COMString = "The center of mass velocity is {:.3f} km/s."**

**VELString = "The maximum radial velocity of star {:d} is {:.3f} km/s."**

**RATString = "The ratio of {} is {:.6f}."**

**MASString = "The minimum mass of the star {:d} is {:.3e} kg."**

**RADString = "The minimum radius of the star {:d} is {:.3e} km."**

**SMAString = "The minimum semi-major axis of the star{:d} is {:.3e} km."**

**print(PERString.format(period))**

**print()**

**print(COMString.format(COMVEL))**

**print()**

**print(VELString.format(1, VELStar1))**

**print(VELString.format(2, VELStar2))**

**LargerVelocity = max([VELStar1, VELStar2])**

**if (abs(VELStar1-VELStar2)<0.1\*LargerVelocity):**

**print("Stars are on approximate circular orbit.")**

**print()**

**print(RATString.format("velocities", RATIO))**

**print()**

**print(SMAString.format(1, SMAStar1))**

**print(SMAString.format(2, SMAStar2))**

**print()**

**print(MASString.format(1, MASStar1))**

**print(MASString.format(2, MASStar2))**

**print()**

**print(RADString.format(1, RADStar1))**

**print(RADString.format(2, RADStar2))**