Griffin Lehrer Numerical Exam Final

Question 1 Write Up

This program calculates the drop in brightness we can see from a star when an exoplanet goes in front of it. This is achieved though several steps. The first step for this program is to gather all relevant data needed to find the drop in brightness. This data includes the orbital period, noise level, the radius of the planet, and the radius of the star. The user then picks a planet size to see how much the brightness level will drop depending on the size of the planet. The rotation of the planet is then calculated by finding its location on the unit circle given a theta value. This theta value starts at zero and is incremented by (1/100) \* 2pi. This is because we want to move 1/100 along the unit circle to get 100 points along the orbit.

After the orbit points are calculated the noise level is found for the star. The noise level is calculated by finding a random number between 1 and 10 representing a 1 to 10% change in the brightness. The value is added or subtracted from the star brightness by another random number between 1 and 2. 1 and the value is added and 2 the value is subtracted. After the noise level is found and applied to the total brightness the percentage drop based on the planet is calculated. This is done trough a formula radius of planet ^2 / radius of star ^2. Then if the planet is below the x axis (y values are negative) the brightness percentage of the star is reduced by the calculated value. The x values y values star brightness and I value (which represent time) are then saved into text files for the user. After every iteration, the star brightness is reset to 100% to be found again for the next point.

One computational drawback my program has is the need to recalculate the brightness of the star for every point. It would be more efficient to find the star brightness once and then from that data get the rest of the brightness levels. However, I could not find an accurate way to do this, so I had to recalculate the brightness of the star for every datapoint. Another way my program could be improved is to simulate all three planets at once and save the brightness values into three files rather than have it be user defined. This would be more computationally expensive as three brightness values are being calculated rather than one but the user would only have to run the program once.