Our Hiding Place in Plain Sight: Earth Viewed from Exoplanets Via the Planetary Transit Method

ALLEGHENY COLLEGE

PHYSICS

By: Paul Allen

Advisor: James Lombardi

Second Reader: Daniel Willey

Department of Physics

Allegheny College

Meadville, Pennsylvania

May 2022

Abstract

We explore the theory and effects of solar limb darkening and the planetary transit method for detecting exoplanets, and then apply these principles to begin to be able to catalogue stars in Earth's sky which are in a favorable position to see Earth via the planetary transit method. The end goal is to sort through stellar data catalogues to lay the groundwork in formulating a list of ideally positioned stars that host exoplanets within the habitable zone and that have existed long enough to potentially produce intelligent life. From this, we would have a list of candidates for potentially inhabited worlds for which two-way knowledge of the other would be possible.

Table of Contents

Abstract 1
Table of Contents 2
Table of Figures 3
Introduction 5
Background 7
Exoplanets 7
Limb darkening 8
Earth transit zone 11
Methods 14
Orbital Transit Simulations 14
Celestial Coordinates 20
Results 28
Limb Darkening Transit Simulation 28
Astrometric Data Analysis 30
Further Research 31
Appendices 33
Appendix A- Earth Transit Zone Derivations 33
Appendix B- Limb Darkening Transit Simulation Code 36
Appendix C- Real Data Analysis Code 39
Appendix D- Data Access Code 43
References 46
Acknowledgments 48

Table of Figures

Figure 1: Parallel plane atmosphere at the surface of a star. The axis labeled as z is the radially outward direction, so when an observer is looking at a star and their line of sight is pointing directly at the star's center, the angle Θ will be zero, and when their line of sight is pointing at the star's edge, Θ will be 90˚ or $\pi 2$ radians. 10
Figure 2: Intensity of the light leaving the surface of a Sun-like star versus the angle of exit, using Equation 1. 11
Figure 3: The ETZ, defined as the area of the sky in which an observer would be able to view an annular or partial eclipse of the Sun by the Earth. The angles shown reference the angular height above and below the ecliptic at which the regions end. Note that the Sun, the Earth, the distances shown, and the angles shown are not to scale. 13
Figure 4: The pixel grid showing the Sun. Pixels which are x'ed out are defined to not exist within the Sun, and the location of each pixel is defined by the location of its center. The radiant luminosity of each pixel is determined by its distance to the center of the Sun. 15
Figure 5: The half-semicircle shown is a portion of a cross section of the Sun, viewed along a vector which is perpendicular to the vector between the Sun and an observer (i.e., top-down). Since point A is at the center of the Sun and point C is at its outer edge, the hypotenuse of triangle ABC in this figure is the Sun's radius and side x is the horizontal distance from the center of the Sun to the center of any given pixel in Figure 3 (given by the variable xP in the code). 16
Figure 6: The above is an example portion of the pixel field like that portrayed in Figure 3. Pixel 2 is at the center of the circle defining the disk of the Earth as it passes in front of the Sun, and so it defines the radiant intensity of all light blocked by the Earth. Pixel 1 is farther from the center of the Sun, and so by itself it would emit less light than pixel 2, but when the Earth eclipses it, it is defined to be emitting the same amount of light as pixel 2, and a similar issue exists for pixel 3 in that alone it emits more light than pixel 2, but when eclipsed by the Earth the simulation treats it as emitting the same amount of light as pixel 2. 18
Figure 7: An example of the pixel simulation for a star which lies directly on the ecliptic, and so lines up perfectly with a line connecting the Earth and the center of the Sun. Note that in this diagram, the circle defining the Earth and the circle defining the Sun are not drawn to scale. 19
Figure 8: Orientation of the ecliptic and bounds of the ETZ relative to the celestial equator (not to scale). Triangle ABC shows how at different locations, the vertical distance from the ecliptic to either bound (parallel to the lines of declination) and the perpendicular distance from the ecliptic are the same only at the solstices. Note that this diagram is not drawn to scale; the segments labelled 0.238˚ of arc and 0.26˚ of arc are not actually this small in the diagram. 22
Figure 9: Like Figure 7, this shows the orientation of the ecliptic and bounds of the ETZ relative to the celestial equator (not to scale). Triangle DEF shows the relationship between the vertical distance to the ecliptic and the perpendicular distance to the ecliptic of a star at point D. Determining how to calculate the magnitude of vDE will be necessary for determining the vertical position of the Earth as it passes in front of the Sun for the pixel simulation. 25
Figure 10: An enlarged view of triangle DEF from Figure 8. 25
Figure 11: Percentage of light eclipsed from the Sun by the Earth transiting across its face. The Sun was centered at the origin and the Earth had an initial *y*-position of zero. 28

Figure 12: Percentage of light eclipsed from the Sun by the Earth transiting across its face. The Sun was centered at the origin and the Earth had an initial *y*-position of 0.87. 29
Figure 13: Histogram of the percentage of sunlight that stars see eclipsed by the Earth as it passes in front of the Sun. 430 data points are represented total on the plot. 30

Introduction

One of the most fundamental questions regarding human existence is about whether we are alone in the universe. Do other intelligent species (at least how we define intelligence) coexist with us somewhere in the universe, and if so, where are they? Many people have endeavored to answer this question, though to date no one has been able to give an answer (at least publicly) better than, “I don’t know yet.” If this was to be answered in the affirmative, another question with the potential to be even more fear inducing would follow: “can they see us?” While we have put much thought and effort into searching out any potential cosmic neighbors, it is also interesting to assume that extraterrestrial civilizations are plentiful and then attempt to apply the methods we use to search the skies for aliens to determine from which star systems they would be able to see us.

The inspiration for this paper and experiment came from a similar study published in a paper by Kaltenegger and Faherty in June of 2021 [1] which found that there are approximately 1,402 stars which would be able to see the Earth transit across the Sun’s face and because of the corresponding drop in the apparent brightness of the Sun, determine that the Earth exists. Among these are several well-known and studied stars such as Wolf-359 (hello, Star Trek fans) and the TRAPPIST-1 system. The latter is of particular interest, as not only is it rather close (12 parsecs away), but it also hosts seven known exoplanets, three of which are at a distance where the temperature from their star would theoretically allow liquid water to exist on their surfaces [2].

Although all good science involves some degree of replication of past experiments, the goal here is to not simply redo this work, but to expand on it. Importantly here is the idea of different star systems having different probabilities of detecting the Earth as it transits the Sun. A star which could view the entirety of the Earth crossing the Sun’s face would have a much better

chance of detecting our presence than one that sees a partial eclipse where only one tenth of the Earth blocks out the Sun's light at maximum (though a partial eclipse would occur is far fewer cases). This is the reason for creating computer simulations of planetary transits, rather than simply deriving a formula based in spherical geometry that relates the position of a star in the sky to whether it can see Earth. The overarching goals here are two-fold; to one: create a way to simulate the Earth passing in front of the Sun as seen from a star with a given right ascension and declination (see Appendix B- Limb Darkening Transit Simulation Code) and two: to create a way to sort through astrometric data to determine if a star can see Earth pass in front of the Sun, and if it can, to determine how much light would be blocked by the Earth from that star's perspective.

Confirmation of the existence of extraterrestrial life, along with two-way communication with it, has the potential to alter human life in ways we cannot yet understand, and should it happen, it would fundamentally change how we view the universe, for better or for worse. If human sentience is any indicator of the nature of sentience elsewhere, this is likely to be true for any other civilization with whom we could interact, and as such, the danger for cultural upheaval could threaten both us and them. When searching for life, it may be beneficial to look where such life might also be aware of us, which would do just a little bit more to ensure that not only are we ready to know about them, but also whether they are ready to know about us.

Background

Exoplanets

As would be expected, our solar system is not the only one to host planets, and after the initial discovery of extrasolar planets, or exoplanets, in 1992 [3], study into the nature of these alien worlds exploded. During its four-year mission starting in 2009, the Kepler space telescope catalogued the existence of roughly 3,600 exoplanets, and notably a large number are of a type which does not exist in our solar system. These so-called "mini Neptunes" or "super Earths" typically have between 1 and 3.8 times the radius of the Earth, but also are thought to have mostly solid compositions and are more terrestrial than jovian. [4]

Planets do not typically give off their own light and generally reflect only a small percentage of the light radiated by their parent star or stars, and so are relatively difficult to image directly.[1] [5] As such, there are a couple different methods for indirectly detecting the presence of exoplanets, namely the previously mentioned transit technique, and also the doppler technique, or the measurement of variations of a star's motion as it and its planet or planets all orbit their shared center of mass [6] (this can also be called the gravitational "wobble" method).

Ideally, knowing the planet's mass, orbital period, and therefore semimajor axis[2] from the doppler method and its radius and therefore volume from the transit method can tell us a lot about it, such as its density and therefore likely composition, or based on its distance to the star

[1] Only 104 out of 29,985 exoplanets catalogued in the NASA Exoplanet Archive were discovered by this method. [5]

[2] Knowing the mass of the star and the orbital period of the planet can tell us its orbital semimajor axis, or its average distance from the star, from Kepler's third law: $mp^2=a^2$, where m is the star's mass in solar masses, p is the orbital period in years, and a is the semimajor axis in AU. [12]

whether it could potentially support liquid water on its surface. Additionally, the transit method can tell us about the exoplanet's atmosphere; by looking at the absorption spectrum of the light which reaches us after passing through the planet's atmosphere (if any), it can tell us about the atmosphere's chemical composition. [7]

An exciting new development in the field of exoplanetary research that bears mentioning here is the James Webb Space Telescope, or JWST. This telescope represents the most advanced piece of technology for viewing the universe at the time of its launch, which finally arrived on December 25th of 2021 after seemingly endless delays [8]. Once in space, the telescope had to unfold correctly and align the 18 gold-plated hexagonal segments of its mirror over a period of two weeks, and any one of the steps going wrong would have rendered the JWST inoperable, though the journal *Nature* described the process in a daily briefing as have been "flawless" [9].[3] Under some circumstances, we believe that the JWST will be able to detect methane, carbon dioxide, oxygen, and water with reasonable certainty after observing only a few dozen transits of a planet across a star [10] and has the potential to increase the number of planetary atmospheres we can study via spectroscopy by a factor of ten [11].

Limb darkening

The Sun is a roughly spherical object, and as such its column thickness will change with the angular position of any given portion of it which an observer sees. Additionally, as is shown in Figure 1, the fact that the density of the Sun increases with depth has the effect of an observer being able to see farther into the Sun when the vector of sight enters at a shallower angle to the Sun's surface. As such, the brightness will change as the line of sight of an observer moves

[3] Over the course of the deployment, there were approximately 300 potential "single point failures" which could have been lethal to the telescope [21].

across the Sun's surface; specifically, the brightness will decrease with radial distance from the center of the Sun's disk. This phenomenon is called limb darkening and presents a level of complexity when simulating planetary transits and the effect that a planet will have on the amount of light blocked. A planet eclipsing the center of a star will cause a greater decrease in light seen by an observer than a planet of the same size eclipsing the edge of the star, simply because there is more light to block coming from the star's center.

We can model this effect with the method of a parallel plane atmosphere, shown in Figure 1. The idea is that we zoom in to the surface of the star to such a degree that the spherical surface can be approximated as being flat and we can therefore more easily measure angles relative to the surface of vectors leaving the star. The result is that in Figure 1 for an observer looking directly at a star, Θ is zero for light rays exiting the star at the center of the solar disk and is 90˚ or $\pi/_2$ radians for light rays leaving the star near the edge. [12]

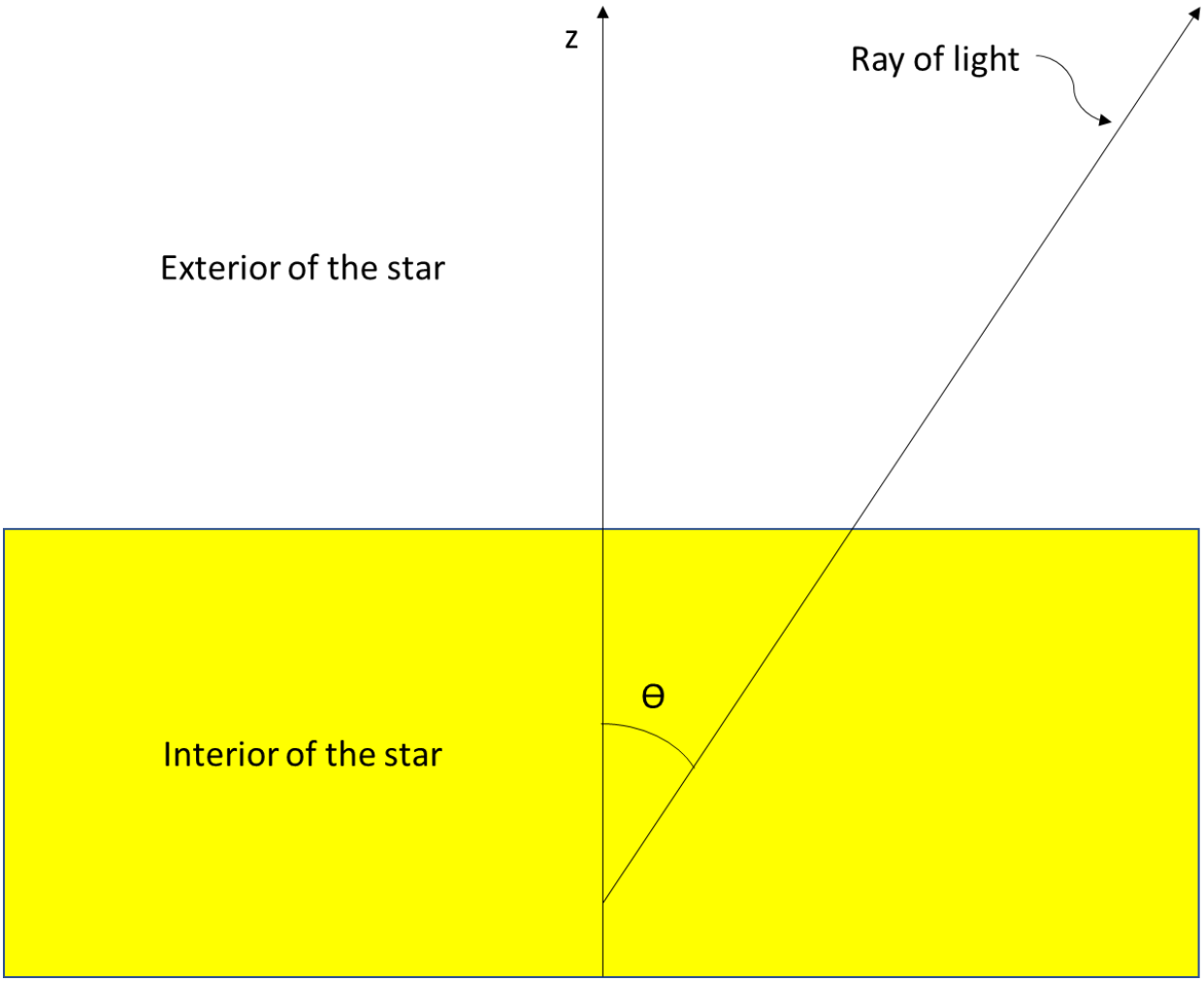


Figure 1: Parallel plane atmosphere at the surface of a star. The axis labeled as z is the radially outward direction, so when an observer is looking at a star and their line of sight is pointing directly at the star's center, the angle Θ will be zero, and when their line of sight is pointing at the star's edge, Θ will be 90˚ or $\pi/2$ radians.

We can then model the varying brightness of the star as a function of position based on the angle of exit for a light ray with:

$$\frac{I(\theta)}{I(0)} = 1 - x\,(1 - \cos\theta) - y\,\cos\theta\log_{10}(\cos\theta)$$

Equation 1

Note that this equation is empirically generated and not derived from first principles. Also, the *x* and *y* coefficients are specific to the type of star in question and have values of *x* = 0.648 and *y* =

0.207 for Sun-like stars. [12] [13] The varying intensity of light versus the angle of exit are graphed in Figure 2.

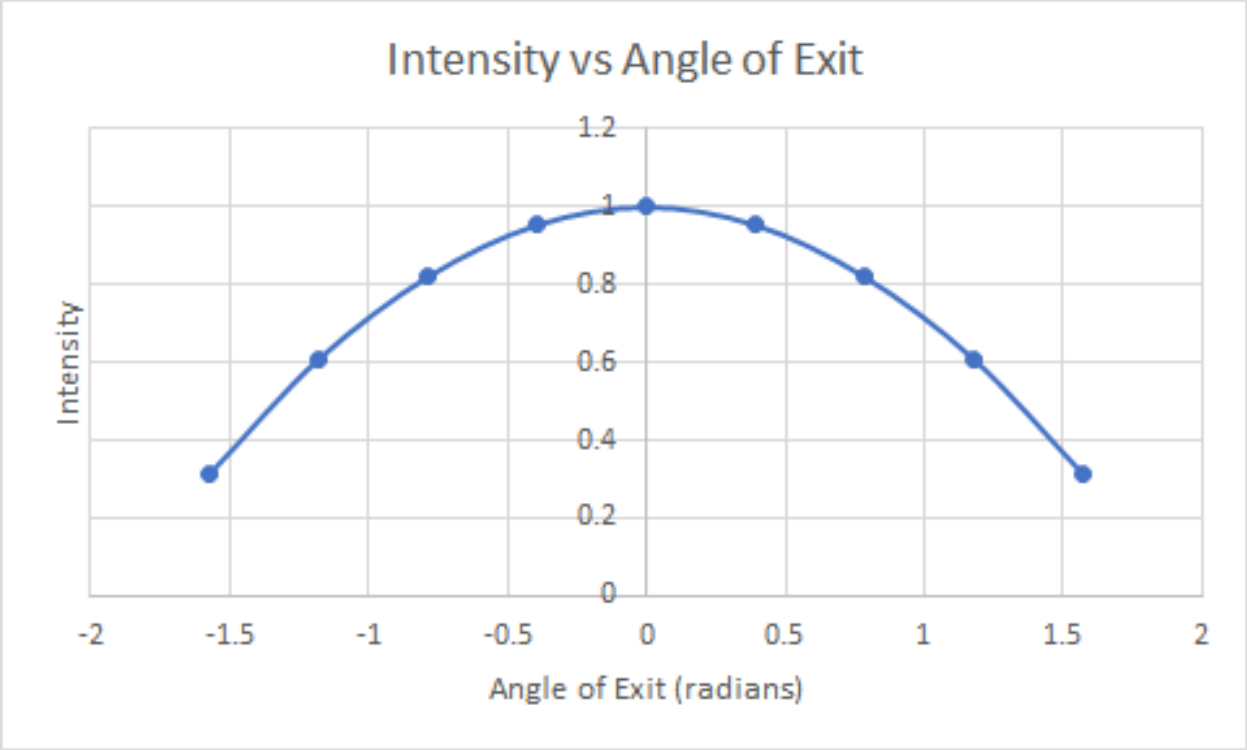


Figure 2: Intensity of the light leaving the surface of a Sun-like star versus the angle of exit, using Equation 1.

Note that in Figure 2, the amount to the total light capable of exiting the Sun never decreases to exactly zero because the Sun has at least some brightness even at the edges of its face due to the brightness of the corona.

Earth transit zone

The method by which astronomers catalogue the apparent locations of celestial objects is similar to the latitude-longitude coordinate system used on Earth's surface; the sky is divided into a grid with the analogue for longitude being an object's right ascension and the analogue for

latitude being declination.[4] Just as longitude lines meet at the north and south geographic poles, the right ascension grid lines run north to south and meet at the north and south celestial poles, measuring angles around the celestial sphere using hours, minutes, and seconds. Right ascension measures 24 hours total around the celestial sphere and increases in value in what would be a counterclockwise direction if the Earth were viewed looking down on the north pole, with the zero-point being the celestial prime meridian. Declination measures angles north to south in degrees, with the celestial equator being its zero-point and the north and south celestial poles having a declination of 90˚ and -90˚ respectively.

The fact that the orbital transit method requires that the observer, the Earth, and the Sun be arranged is a relatively straight line means that stars in only certain areas of the sky would host planets that would theoretically be able to detect the Earth. For instance, the star Polaris has a declination of roughly 90˚ and is nowhere near the ecliptic, and so at no point in its orbit would the Earth pass between the sun and Polaris. The area where and observer would theoretically be able to see the Earth eclipse the Sun is known as the Earth Transit Zone, or ETZ [1], and is illustrated in Figure 3:

[4] These lines exist on the figurative “celestial sphere”, or the total sky surrounding the Earth, and it is important to note that celestial coordinates say nothing of the radial distance of an object to the Earth.

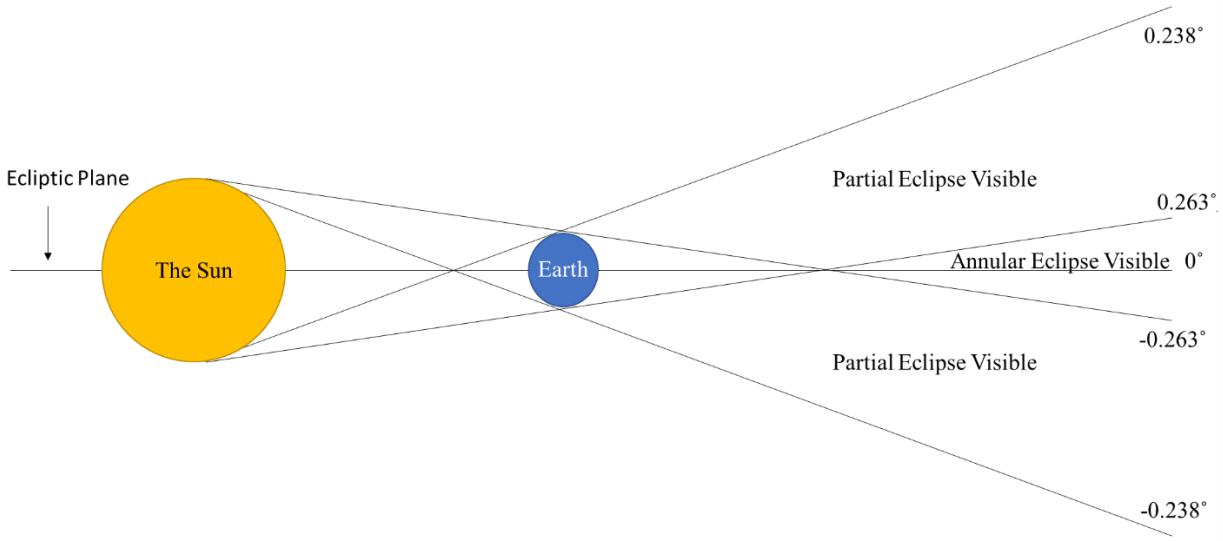


Figure 3: The ETZ, defined as the area of the sky in which an observer would be able to view an annular or partial eclipse of the Sun by the Earth. The angles shown reference the angular height above and below the ecliptic at which the regions end. Note that the Sun, the Earth, the distances shown, and the angles shown are not to scale.

Within the ETZ, there are three distinct areas which are defined by what type of eclipse is would be visible to an observer within them. The two types of eclipses in question are total eclipses, where the Earth would pass completely in front of the Sun, and partial eclipses, where only a fraction of the Earth would pass in front of the Sun.[5] Extending 0.236˚ above and below the ecliptic, an observer would be able to view an annular eclipse of the Sun by the Earth and extending from 0.236˚ to 0.238˚ above and below the ecliptic an observer would be able to see a partial eclipse of the Sun by the Earth. There derivations for these angles are in Appendix A- Earth Transit Zone Derivations.

[5] There is also a region where a total eclipse of the Sun could occur, where no portion of the Sun would be visible around the Earth, but this region ends 0.009AU (1.35×10^6km) away from the Earth, which is far too close for any other star to ever enter.

Methods

Orbital Transit Simulations

Unfortunately, travel to all the nearby star systems to catalogue those within the ETZ is infeasible for an undergraduate student, so data analysis took place using simulations of the Earth transiting across the Sun from the point of view of known star systems. The first few prototype programs to achieve this were based on random number generation to produce coordinates within a specific area defining the sun, and then checking to see if the generated coordinates also fell within a circle defining the Earth. The fraction of coordinates within the Earth over the total coordinates generated would ideally represent the fraction of the Sun eclipsed by the Earth. While this method works, it does take a lot of time for each simulation compared to how much the data converges to expected test values, so later iterations of the code used a pixel-based approaches.

The final versions of the simulations[6] can be divided into two basic steps, the first being the calculation of the Sun's luminosity as a function of its area and the second being the calculation of the percentage of the total light blocked as the Earth transits across the Sun's face. First, for the Sun's luminosity we need a general grid of pixels/cells with each pixel's location being defined by the location of its center. Following this, we assign all pixels within a certain radius from the center of the grid to represent the sun, which essentially appears as a disk on the celestial sphere. The computer then discards the cells not defined to be inside the Sun (those pixels are x'ed out in Figure 4) from the total generated.

[6] See Appendix B- Limb Darkening Transit Simulation to simulate the Earth passing in front of the Sun and Appendix C- Real Data Analysis Code to read in astrometric data to determine which stars in a data set fall within the ETZ, and then determine the maximum amount of light blocked from each's perspective.

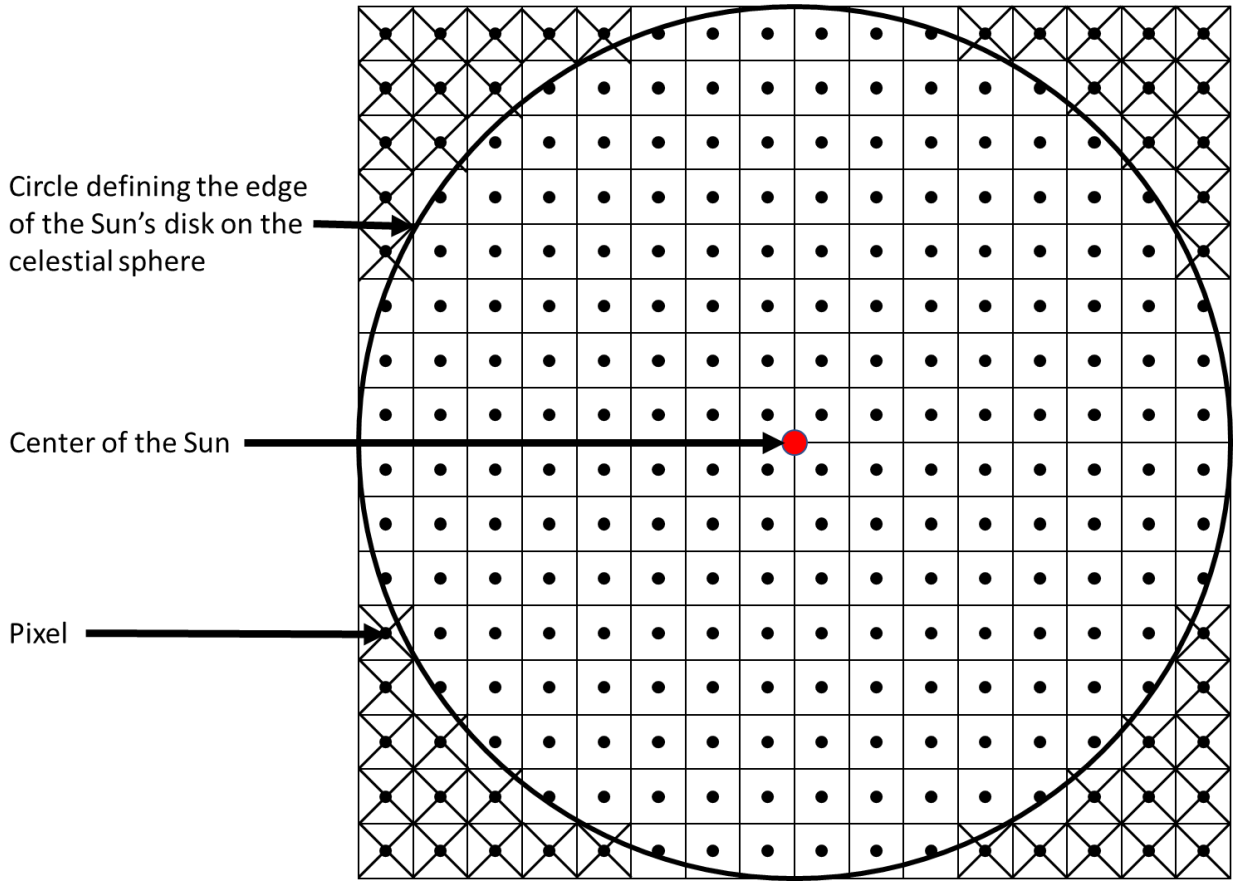


Figure 4: The pixel grid showing the Sun. Pixels which are x'ed out are defined to not exist within the Sun, and the location of each pixel is defined by the location of its center. The radiant luminosity of each pixel is determined by its distance to the center of the Sun.

Next, we need to calculate the luminosity of each pixel and to do this, we need to know the angle of exit for a photon leaving the Sun's photosphere based where it exits the Sun's face relative to the center of the solar disk. This angle is necessary because it is the independent variable in Equation 1 we need to calculate radiant intensity. A simple example of this is that the angle of exit for a photon leaving the Sun very close to the edge is approximately 90˚, and the angle of exit for a photon leaving the Sun at the center of the disk being 0˚. This gets more complicated for locations that are not in the center or at the edge of the solar disk on the pixel field; since the Sun's face is a sphere, it is not a simple linear increase in this angle from the

center as one moves towards the edge. I demonstrate the trigonometry we used to derive and equation for this angle in Figure 5.

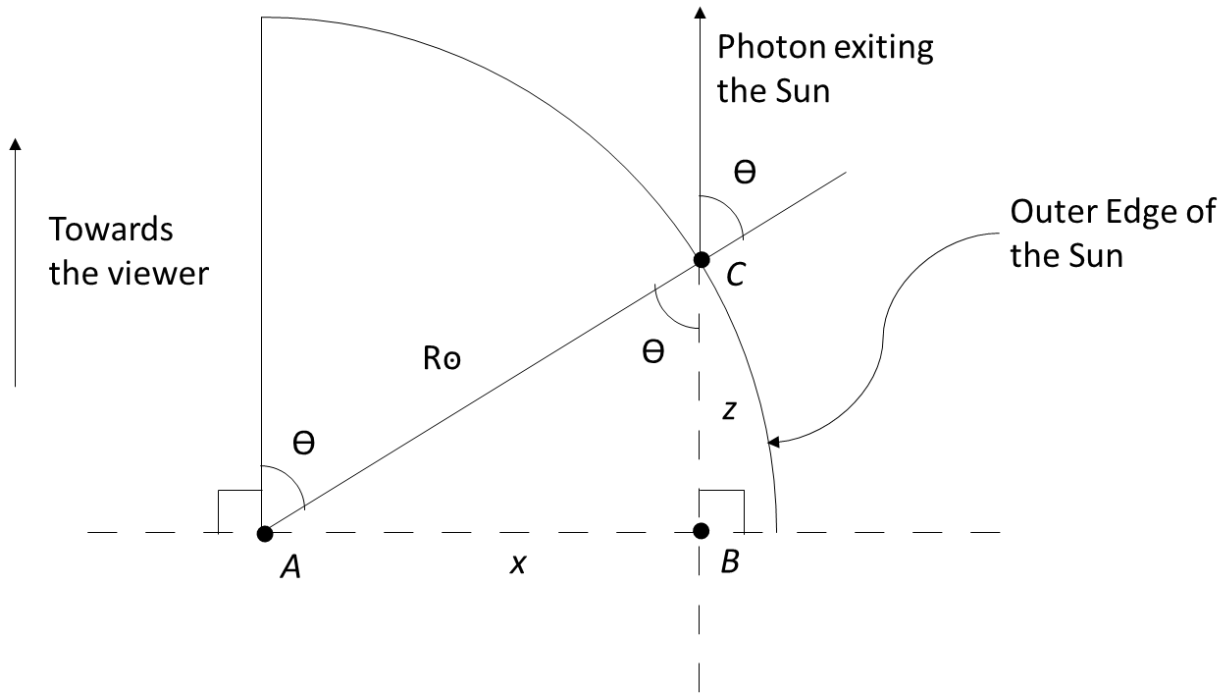


Figure 5: The half-semicircle shown is a portion of a cross section of the Sun, viewed along a vector which is perpendicular to the vector between the Sun and an observer (i.e., top-down). Since point A is at the center of the Sun and point C is at its outer edge, the hypotenuse of triangle ABC in this figure is the Sun's radius and side x *is the horizontal distance from the center of the Sun to the center of any given pixel in Figure 4 (given by the variable xP in the code).*

From Figure 5 and the Pythagorean theorem in three dimensions, we can easily express the Sun's radius in terms of the sides of triangle ABC and the horizontal and vertical positions of any given pixel:

$$R_\odot{}^2 = x^2 + y^2 + z^2 \Rightarrow\ z = \sqrt{R_\odot{}^2 - x^2 - y^2}$$

$$\theta = \tan^{-1}\left(\frac{x}{z}\right) = \tan^{-1}\left(\frac{x}{\sqrt{R_\odot{}^2 - x^2 - y^2}}\right) = \tan^{-1}\left(x\left(R_\odot{}^2 - x^2 - y^2\right)^{-2}\right)$$

So, the angle of exit of a photon relative to a plane tangent to the edge of the Sun, based on its location on the solar disk, is

$$\theta = \tan^{-1}\left(x\left({R_\odot}^2 - x^2 - y^2\right)^{-2}\right)$$

Equation 2

Since we know the *x* and *y* location of the center of each pixel, we can therefore calculate the angle at which a photon would exit the star at that point from Equation 2, and then plugging the value of this angle in for Θ in Equation 1, we can determine the radiant intensity at the center of any given pixel. We then multiply this value by the area covered by pixel and doing this for each pixel will give us the radiant intensity of the light given off by the Sun as a function of location on the solar disk. It is important to note that this creates only an approximation of the Sun, as the radiant intensity would change constantly as a point moves away from the center, not in a step-basis as the pixels would simulate. This imprecision diminishes as the size of each individual pixel diminishes and the total number of pixels increases and running the simulation in this way helps to later calculate the light blocked by the Earth as it passes in front of any one point on the solar disk.

The last portion of the simulation is to calculate how much light the Earth blocks as it passes in front of the Sun. Since the Earth has a cross sectional area of only about 0.0084% of the Sun [12][7] and the width of a pixel in the simulation is 1/10 the radius of the Earth[8], the difference in scale allows us to simulate the Earth as blocking a uniform amount of light across its entire face. It is important to note that in reality this would not happen, as the portions of the Earth's silhouette closer to the center of the solar disk would block more light than the center of the silhouette, and the center would block more light than portions farther from the center of the

[7] $R_\odot = 6.96 \times 10^8$ m, $R_\oplus = 6.38 \times 10^6$ m, and $a_\oplus = 1.5 \times 10^{11}$ m [12]
[8] See Appendix B- Limb Darkening Transit Simulation Code

Sun. Put simply, in the simulation the parts of the Earth that are closer to the solar disk block less light than they would in real life, and the parts that are farther from the center block less. This effect is illustrated in Figure 6:

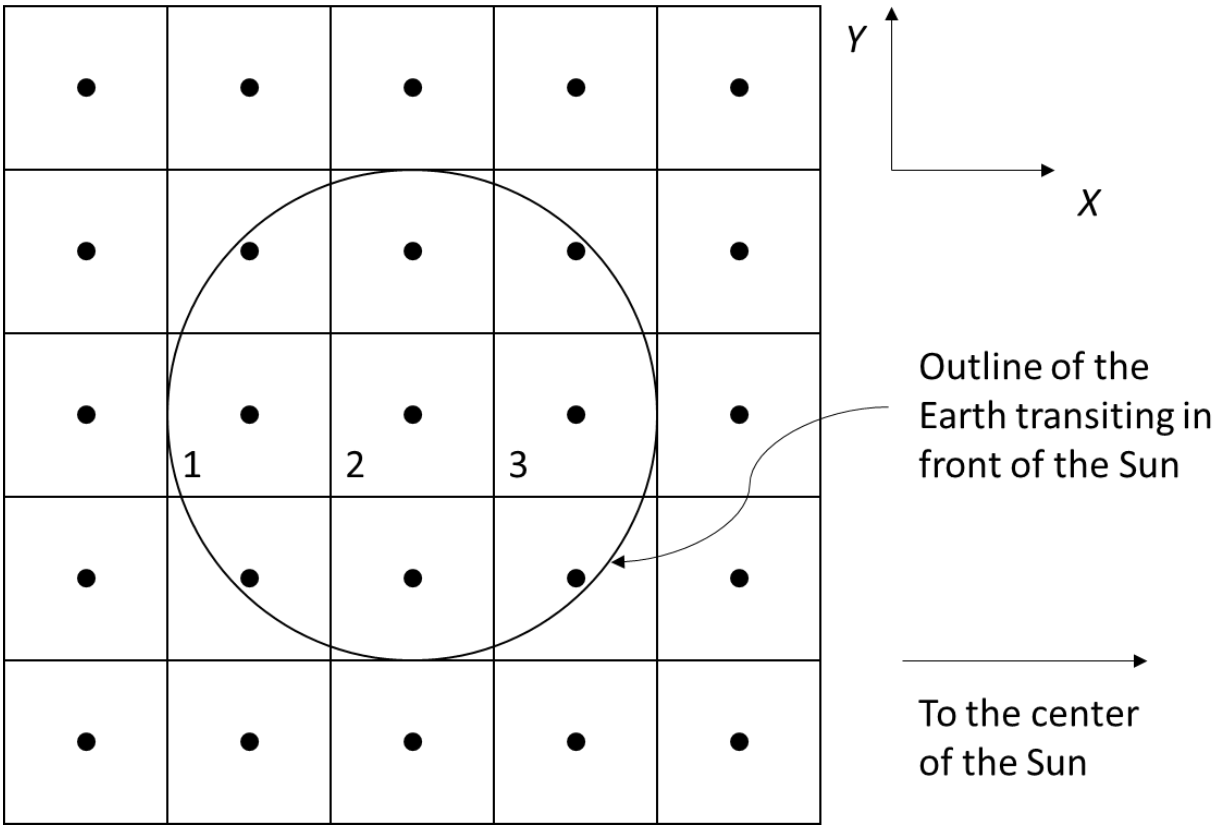


Figure 6: The above is an example portion of the pixel field like that portrayed in Figure 4. Pixel 2 is at the center of the circle defining the disk of the Earth as it passes in front of the Sun, and so it defines the radiant intensity of all light blocked by the Earth. Pixel 1 is farther from the center of the Sun, and so by itself it would emit less light than pixel 2, but when the Earth eclipses it, it is defined to be emitting the same amount of light as pixel 2, and a similar issue exists for pixel 3 in that alone it emits more light than pixel 2, but when eclipsed by the Earth the simulation treats it as emitting the same amount of light as pixel 2.

The effects of this inaccuracy are offset somewhat by the fact that areas of the Earth's disk farther from the Sun's center (simulated to block more light than they would in reality) and areas closer to the Sun's center (simulated to block less light than they would in reality) would at least partially cancel each other out and reduce the net effect of the Earth being simulated to block more or less light than it would in reality.

Data input will be for the first simulation (Appendix B- Limb Darkening Transit Simulation Code) will be typographical from the user; when inputting the data, the program will read in the right ascension and declination of a given object, check to see if it falls within the ETZ, and based on its location relative to the ecliptic, determine at what alignment the star would see the Earth and the Sun (what the Earth's *y*-position on the pixel grid would be, denoted by *yP* in the code). For example, if the star lies directly on the ecliptic, it will see the Earth cross directly in front of the Sun, and so the simulated Earth would have a *y*-position of zero, as in Figure 7:

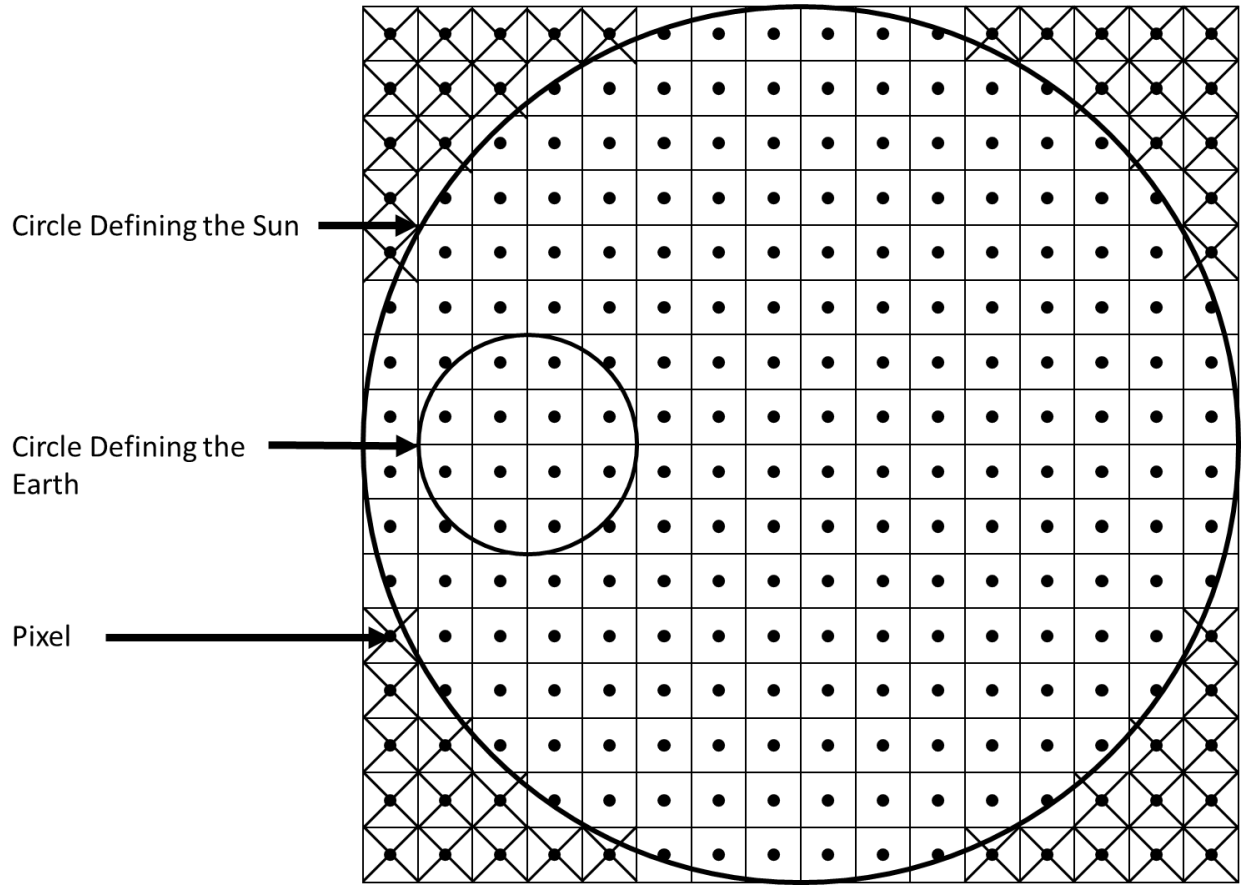


Figure 7: An example of the pixel simulation for a star which lies directly on the ecliptic, and so lines up perfectly with a line connecting the Earth and the center of the Sun. Note that in this diagram, the circle defining the Earth and the circle defining the Sun are not drawn to scale.

Determining the perpendicular distance to the ecliptic and therefore the *y*-position on the pixel grid is not as simple as reading in the declination of the star, and I describe the procedure for

doing this in the next section. In this simulation, the Earth starts out not eclipsing any part of the Sun but is advanced slowly across the Sun's face and the amount of light eclipsed is recorded for each of these iterations.

For the second simulation (Appendix C- Real Data Analysis Code) will be based on the celestial coordinates of nearby stars, which will be read into a list in the program. The code then determines the maximum amount of light that the star would see blocked by the Earth and records this value. This maximum value occurs when the Earth is at or close to an *x*-value of zero[9], and then varies depending on the y-value. Regarding the data download, I found astrometric data through the European Space Agency [14] and used code on the same website to acquire the data[10]. I included a copy of this code in Appendix D- Data Access Code.

Celestial Coordinates

In determining which stars in a celestial catalogue inhabit the ETZ, we must determine a function for the vertical height of the ETZ on the celestial sphere based on the right ascension at any point. That is, we need a function which gives the declinations of the upper and lower boundaries of the ETZ at any given right ascension. This takes on an added degree of complexity since the plane of the ecliptic and the plane of the celestial equator are inclined to each other, not parallel.

Since straight lines on the surface of a sphere become curved lines when translated to a flat/cartesian surface, we can begin to describe the behavior of the lines in question by

[9] I say 'or close to' zero here because of two reasons. For one, entering an *x*-value of exactly zero returns a division by zero error from the program, and two, because the Earth never has an *x*-value of zero at any of its positions in the code in Appendix B- Limb Darkening Transit Simulation Code. The closest it gets is an *x*-value of -0.0022, so this is the *x*-value used to find the maximum amount of light blocked in Appendix C- Real Data Analysis Code.

[10] The code was at https://www.cosmos.esa.int/web/gaia-users/archive/programmatic-access#PythonFlag under the heading, "1.5. PYTHON LOW-LEVEL SCRIPT TO EXECUTE AN ASYNCHRONOUS QUERY AND WAIT UNTIL THE RESULTS ARE AVAILABLE"

designating the ecliptic to be defined by a sine wave. We use a sine equation since the zero-point of right ascension is at the vernal equinox (where, by definition, the ecliptic and the celestial equator cross), and the angular separation of the ecliptic and celestial sphere increase as right ascension increases, reaching the maximum of 23.44˚ (that is, a declination for the ecliptic of 23.44˚) at 6', or 90˚ of right ascension, the location of the sun on the summer solstice. The angular separation, then decreases, reaching zero at the autumnal equinox, and the pattern repeats with the ecliptic in the southern celestial hemisphere. After passing the winter solstice at 18' or 270˚ of right ascension, the ecliptic then crosses the celestial equator again at the vernal equinox, bringing us back to where we started. Knowing now to use the sine function, we then can say that

$$\delta_{ecl} = I \sin \alpha$$

Equation 3

where the declination of the ecliptic is equal to the product of the obliquity of the ecliptic to the celestial equator (23.44˚ [15]) and the sine of the right ascension. *I* here is the obliquity or inclination and is analogous the amplitude of the sine wave that the ecliptic becomes when translated to a flat surface.

The next step is then to derive the equations for the upper and lower bounds of the ETZ based on the location of the ecliptic. If the ecliptic were in the same plane as the celestial equator, this would be as simple as saying that any location with a declination -0.238˚< δ<0.238˚ is within the ETZ, but since the ecliptic is inclined to the celestial equator, the perpendicular distance from the ecliptic and the vertical distance from the ecliptic is seldom the same (it is the same only at the solstices). This is illustrated in Figure 8:

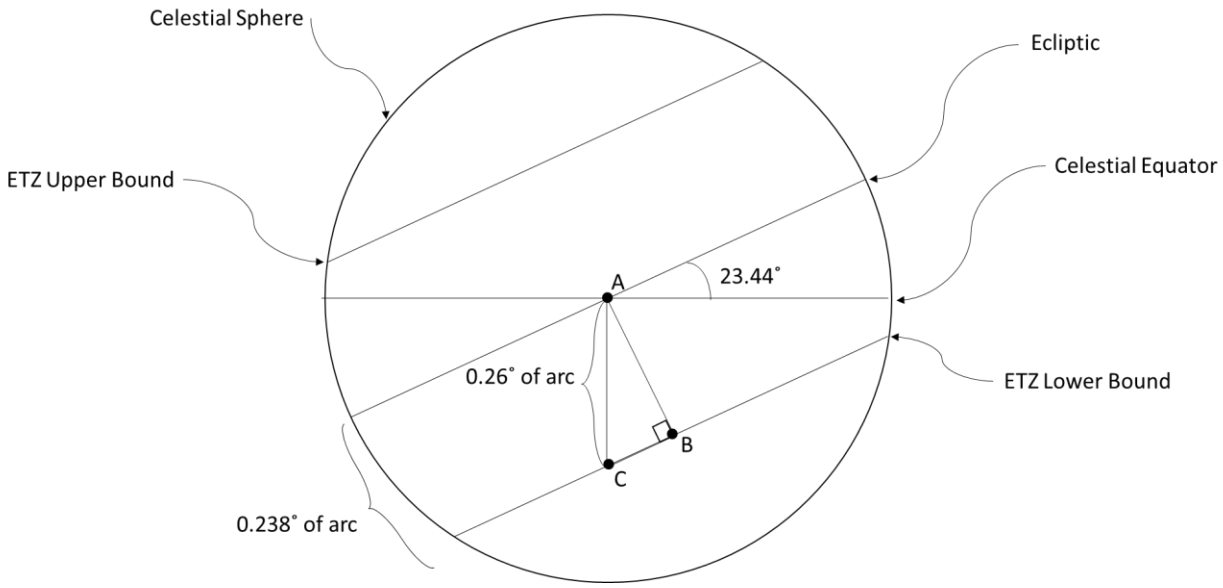


Figure 8: Orientation of the ecliptic and bounds of the ETZ relative to the celestial equator. Triangle ABC shows how at different locations, the vertical distance from the ecliptic to either bound (parallel to the lines of declination) and the perpendicular distance from the ecliptic are the same only at the solstices. Note that this diagram is not drawn to scale; the segments labelled 0.238˚ of arc and 0.26˚ of arc are not actually this small in the diagram.

From triangle ABC in Figure 8, we can determine the maximum value of angle ∠BAC between the vector perpendicular to the ecliptic ($\vec{v}AB$) and vertical separation vector of the ecliptic and the ETZ bounds ($\vec{v}AC$, parallel to the lines of declination), which occurs at the solstices. We know that ∠ABC is a right angle and that at the equinoxes, ∠BAC is at its maximum, and since it is orthogonal to the angle between the ecliptic and the celestial equator, this is the obliquity of the ecliptic, or 23.44˚ [15]. Put simply, ∠BAC = 23.44˚ in Figure 8.

To determine the equation for the ETZ bounds, we need to know the angular value of the vertical separation of the bounds and the ecliptic at the solstices, or ($\vec{v}AC$) in Figure 8. We can determine this with the sine rule for spherical geometry:

$$\frac{\sin(A)}{\sin(a)} = \frac{\sin(B)}{\sin(b)} = \frac{\sin(C)}{\sin(c)}$$

Equation 4

Where the capital letters *A*, *B*, and *C* are the values of each of the angles of the triangle on a sphere's surface, and the lowercase letters *a*, *b*, and *c* are the arc values of the sides opposite each angle of the triangle with the corresponding letter. From Equation 1, we can say:

$$\frac{\sin(90°)}{\sin(\vec{v}AC)} = \frac{\sin(90°-23.44°)}{\sin(0.238°)} \Rightarrow \vec{v}AC = \sin^{-1}\left(\frac{\sin(0.238°)}{\sin(66.56°)}\right) = 0.26°$$

This is already stated in Figure 8, but it helps to go over the derivation.

Now that we have this value, we can begin to build our equations for the upper and lower bounds of the ETZ. We know that they vary in height (vertical distance from the ecliptic) and that they are centered on the ecliptic, so the function we derive for the width will simply add onto the declination of the ecliptic at any point for the upper bound and subtract from the declination of the ecliptic for the lower bound. The width of the ETZ is at its lowest value at the solstices when it is equal to 0.238˚ (refer to Figure 8) and is at its greatest at the equinoxes when it is 0.26˚, and so the function we derive will need to regularly vary between these two values. To do this, we can say

$$Vertical\ ETZ\ Height = W_{ETZ} + |0.022\cos\alpha|$$

Equation 5

where W_{ETZ} is the width of the ETZ (0.238˚) and the 0.022 value is the difference between 0.238˚ and 0.26˚. We are using the difference between these two values here along with the cosine of the right ascension (α) because we need to add the entirety of this difference at the equinoxes where the vertical separation is 0.26˚ (cos(0) = 1), none of this difference at the solstices where the vertical separation is 0.238˚ (cos(90) = cos(270)=0) and differing amounts in between. Additionally, we are using absolute value parentheses here because from $\pi/2$ to $3\pi/2$ $\cos\theta$ is negative, and we want this entire term to always be adding a varying distance to the ETZ's location. This, combined with the fact that the ETZ centers on the ecliptic, gives us

$$\delta_{ETZ,ub} = \delta_{ecl} + (W_{ETZ} + |0.022\cos\alpha|)$$

Equation 6

where $\delta_{ETZ,ub}$ is the declination of the upper bound of the ETZ at any given right ascension α, and

$$\delta_{ETZ,lb} = \delta_{ecl} - (W_{ETZ} + |0.022\cos\alpha|)$$

Equation 7

where $\delta_{ETZ,lb}$ is the declination of the lower bound of the ETZ at any given right ascension α.

Now, for the pixel simulation we need the perpendicular distance between a star and the ecliptic based on its right ascension and declination to generate the vertical height of the Earth as it passes in front of the Sun (the vertical height of the planet on the pixel field). An example of this distance is illustrated by $\vec{v}DE$ in Figure 9:

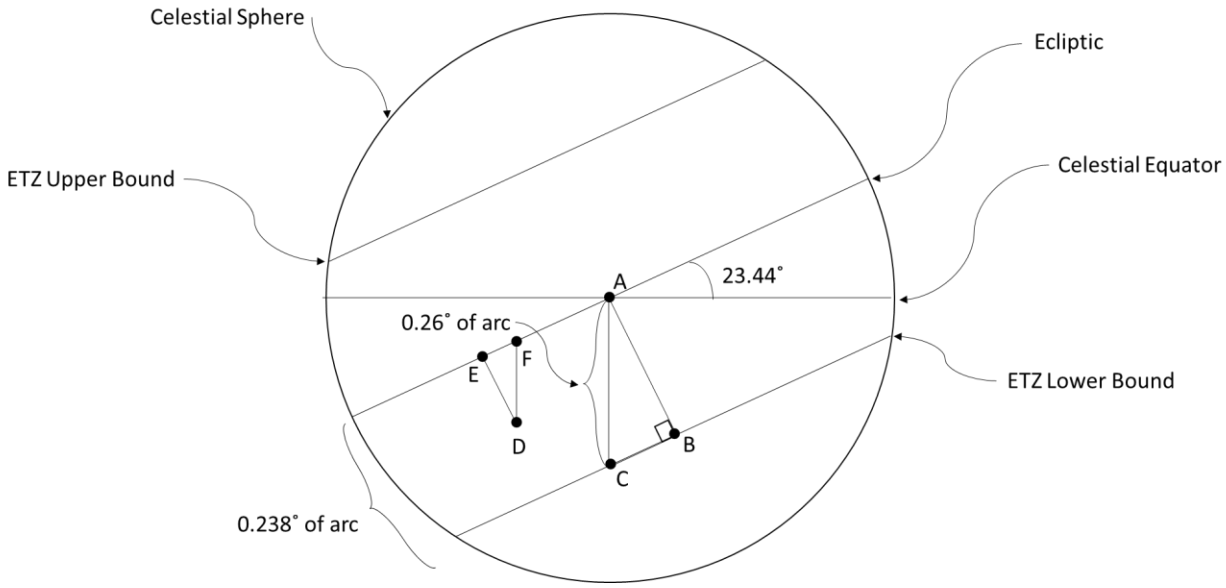


Figure 9: Like Figure 8, this shows the orientation of the ecliptic and bounds of the ETZ relative to the celestial equator (not to scale). Triangle DEF shows the relationship between the vertical distance to the ecliptic and the perpendicular distance to the ecliptic of a star at point D. Determining how to calculate the magnitude of $\vec{v}DE$ will be necessary for determining the vertical position of the Earth as it passes in front of the Sun for the pixel simulation.

To make this easier to visualize, Figure 10 shows and enlarged view of triangle DEF from Figure 9:

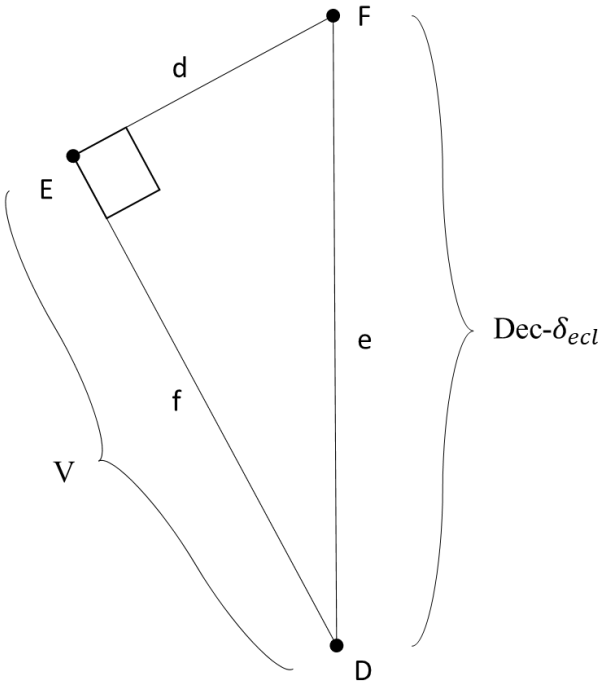


Figure 10: An enlarged view of triangle DEF from Figure 9.

In Figure 10, the star's location is at point *D* and $\vec{v}EF$ is a segment of the ecliptic, meaning that $\vec{v}DF$ is parallel to the lines of declination. This would imply that we can say that the length of $\vec{v}DF$ is equal to the difference between the declination of the star and the declination of the ecliptic at the same right ascension value: $\vec{v}EF = Dec - \delta_{ecl}$.

The segment $\vec{v}DE$ is the perpendicular distance between the star's location on the celestial sphere and the ecliptic, and as such it also represents the vertical height of the Earth in the pixel simulation, so here it is denoted with *V*. Knowing all these, we can say that *yP* is equal to two ratios:

$$yP = \frac{V}{V_{max}} = \frac{Dec - \delta_{ecl}}{\delta_{ETZbound} - \delta_{ecl}}$$

Equation 8

In the second expression of Equation 8, *Vmax* is the maximum value that the perpendicular distance between the star and the ecliptic could be (if it is on the boundary of the ETZ), and in the third, $\delta_{ETZbound} - \delta_{ecl}$ is the maximum value that the vertical distance between the star and the ecliptic could be. We can be more specific by saying

$$yP = \frac{V}{V_{max}} = \frac{Dec - \delta_{ecl}}{\delta_{ETZ,ub} - \delta_{ecl}}$$

Equation 9

if the star is above the ecliptic, and we can say

$$yP = \frac{V}{V_{max}} = \frac{Dec - \delta_{ecl}}{\delta_{ecl} - \delta_{ETZ,lb}}$$

Equation 10

if the star is below the ecliptic. Hence, if the star is above the ecliptic, we can determine *yP* with Equation 11

$$yP = \frac{Dec - \delta_{ecl}}{\delta_{ETZ,ub} - \delta_{ecl}}$$

Equation 11

and if the star is below the ecliptic, we can determine *yP* with Equation 12:

$$yP = \frac{Dec - \delta_{ecl}}{\delta_{ecl} - \delta_{ETZ,lb}}$$

Equation 12

Results

Limb Darkening Transit Simulation

Figure 11 is a sample light curve from a simulated transit of the Earth across the Sun's face using the code in Appendix B- Limb Darkening Transit Simulation Code. The stellar coordinates for the hypothetical star are a right ascension of 0 and a declination of 0, meaning that this hypothetical star is exactly on the vernal equinox and has a *yP* value of 0 in the simulation.

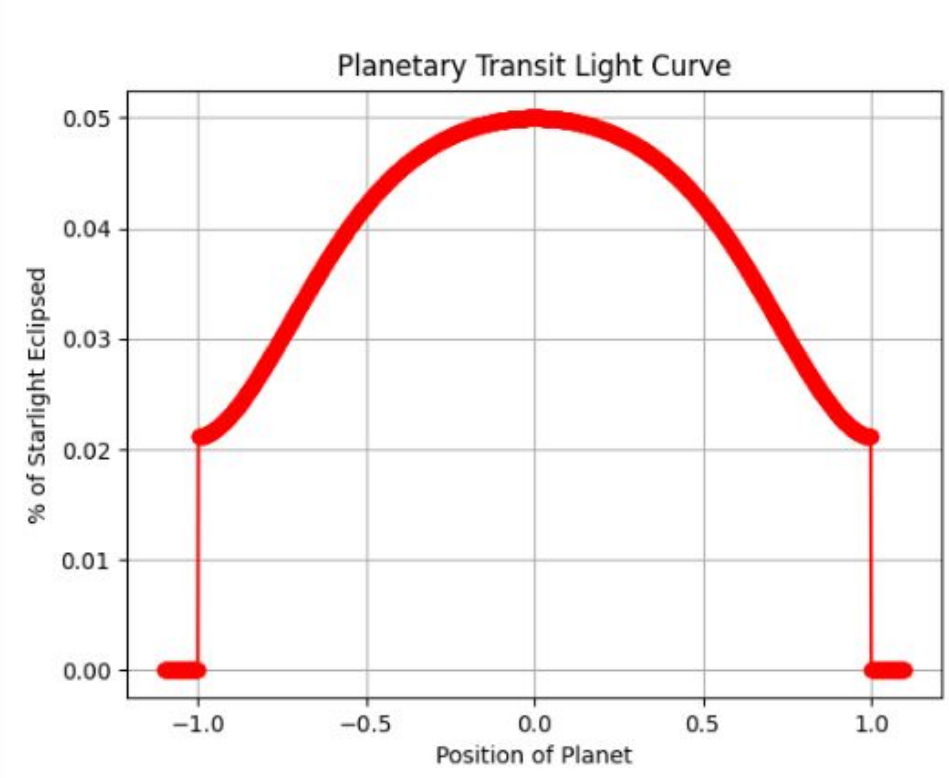


Figure 11: Percentage of light eclipsed from the Sun by the Earth transiting across its face. The Sun was centered at the origin and the Earth had an initial y-position of zero.

This is expected; the Earth starts of eclipsing no light until it starts to cross the Sun's face (beginning at x = -1), increasing the amount of blocked as it passes in front of pixels of increasing radiant intensity, and it reaches a peak at x = 0. The process then runs in reverse; the

amount of light the Earth blocks diminishes, and after the Earth crosses the boundary defining the solar disk at $x = 1$, no light is blocked. We can contrast this light curve with one generated for a star with a right ascension of 0 but a declination 0.004rad, meaning it would have a yP value of 0.87:

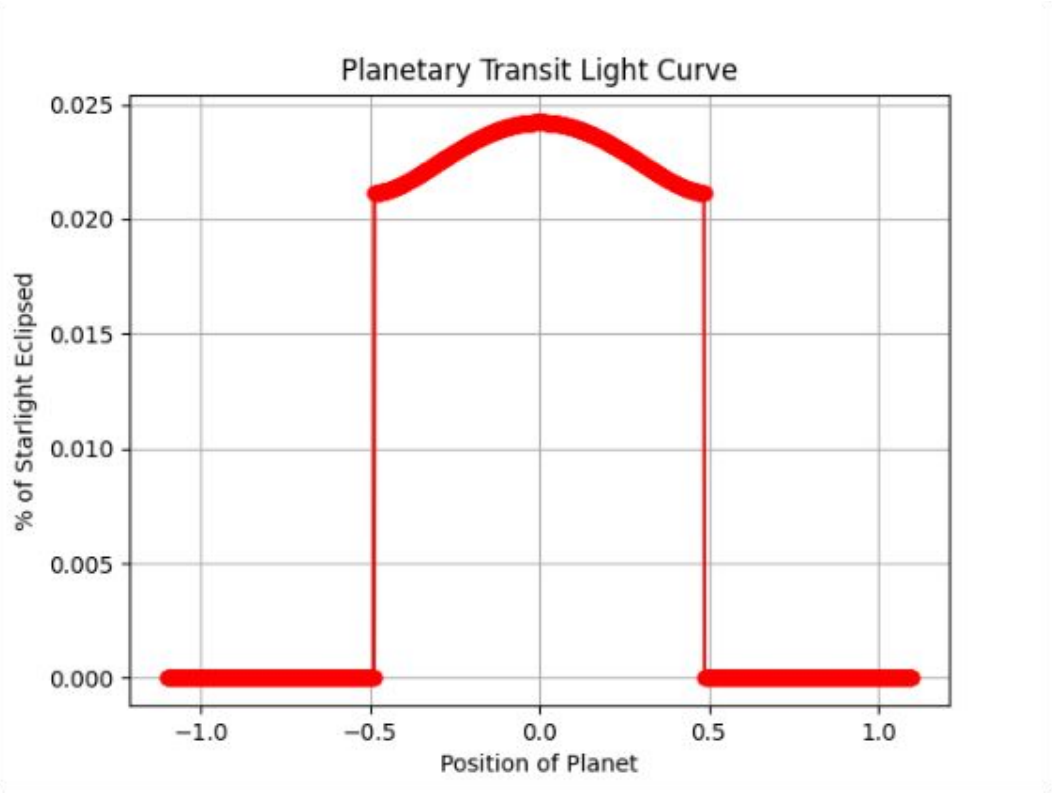


Figure 12: Percentage of light eclipsed from the Sun by the Earth transiting across its face. The Sun was centered at the origin and the Earth had an initial y-position of 0.87.

The two differences here make sense. One, the graph much narrower than Figure 11 since the Earth spends much less time transiting the Sun's face from this star's point of view; it starts with a *y*-position higher on the pixel grid, and so its path across the Sun is not a full diameter. Two, in Figure 11 the Earth crosses areas of the Sun closer to the center of the solar disk and so passes over more luminous parts. This leads to Figure 11 having a higher maximum amount of light eclipsed.

Astrometric Data Analysis

Figure 13 shows the relative frequencies of differing percentages of sunlight blocked by the Earth. The relative lack of a trend if to be expected, as there is no mechanism that would bias these data. After the program sorted through 117,955 data points in the initial downloaded file, it came back with 450 being in the ETZ, or about 0.38%

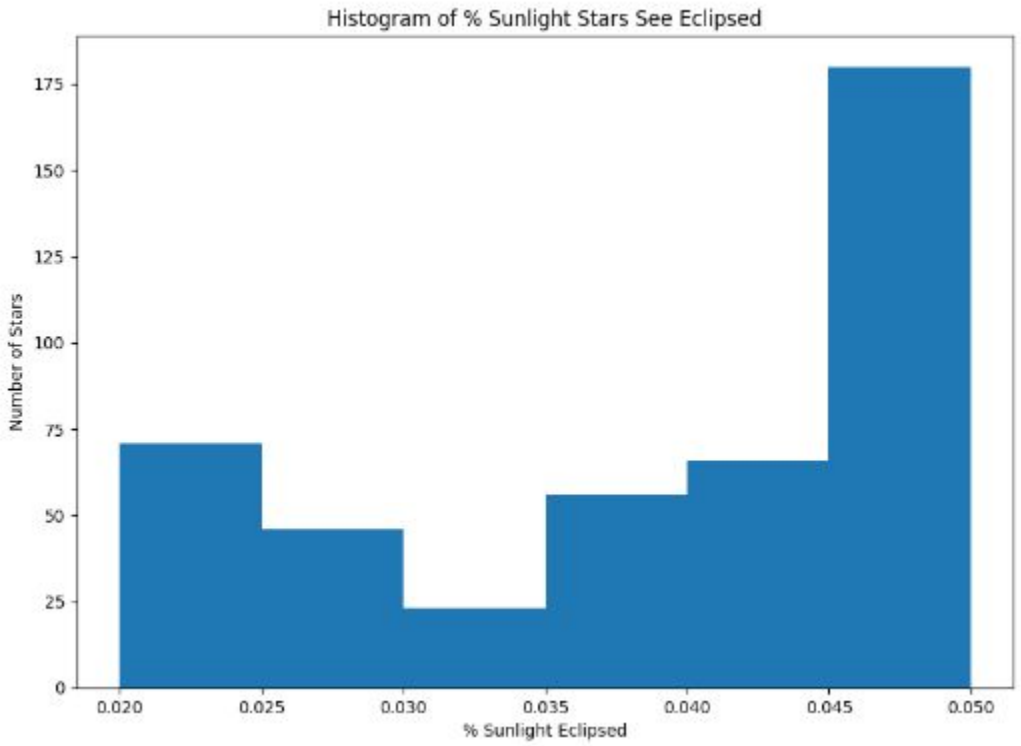


Figure 13: Histogram of the percentage of sunlight that stars see eclipsed by the Earth as it passes in front of the Sun. 430 data points are represented total on the plot.

It is important to note that the data that were fed into the program in Appendix C- Real Data Analysis Code were retrieved from the GAIA Archive of the European Space Agency [16] and are not a subset of the stars surveyed by Kaltenegger and Faherty [1]

Further Research

Given that it is not a special case for the Earth to detectable as it transits in front of the Sun, the likelihood that any theoretical observer would notice the presence of planets in any solar system goes up the more planets there are. As such if an observer were to see Venus transiting but not be in a position to see Earth do so, they could reasonably conclude that there is a high possibility that life exists in our solar system; as Venus does orbit within the habitable zone [17], they would be right, just for the wrong reason.

This increases the number of overlapping transit zones on the celestial sphere by possibly a factor of eight. Though only three out of our solar system's eight planets lie within the habitable zone [18], there is evidence that tidal heating could create environments warm enough to produce life on moons orbiting gas giant planets [19], so any alien astronomer could conclude the possibility of life existing here even if they do so by discovering Jupiter or any of its siblings. Additionally, we also know that water ice exists on Mercury [20], further adding to the list of planets where a search for life may be a long shot but also not quite futile.

The implications of this are interesting. Further programs could consider which areas of the sky constitute transit zones for other planets (the VTZ, or the Venus transit zone for example), and when two or more of these zones overlap, note that an observer within this overlap would have a higher chance of concluding that life exists in our solar system. Additionally, as Kaltenegger et al. researched, the movement of stars across our sky could be considered [1]. Life has existed on Earth for billions of years, so some stars might have previously been able to detect the Earth but have since moved out of the ETZ and are no longer able to. Conversely, we could also consider those stars which are not in the ETZ but whose movement will bring them into it in the near future.

Finally, of all this could be an indexing of nearby stars regarding their likelihood to be able to assume that life exists in our solar system. Such an equation to do this could take the form of:

$$F = N_{lsb} \times T_{il} \times T_{ETZ} \times R^{-2} \times P_{le} \times N_{vsb}$$

Equation 13

Where:

F is the favorability of a star system supporting life that can detect us (via the transit method).

N_{LSB} is the number of potential life- supporting bodies in this hypothetical star system.

T_{il} is the age of the star system, or the time for intelligent life to have evolved there (measured in millions of years).

T_{ETZ} is the amount of time that the star system has been in the ETZ (also measured in millions of years).

R is the average distance of the star system to the Sun during the time that it has been in the ETZ. This variable decreases with distance because the likelihood of a civilization seeing us would presumably decrease with distance, as they would need more powerful instruments. This is raised to the -2 power because this decrease would following the inverse square law of light.

P_{le} is the maximum percentage of light eclipsed that is visible to the hypothetical star. This is the same value as is measured by the code in Appendix C- Real Data Analysis Code and Figure 13.

N_{vsb} is the number of solar system bodies visible to the hypothetical star transiting in front of the Sun, for reasons discussed above.

I would emphasize that Equation 13 accounts for factors of both our solar system and the observing one, since we are considering the potential for two-way knowledge.

Appendices

Appendix A- Earth Transit Zone Derivations

The angles above and below the ecliptic which bound the regions of the sky able to see an annular and partial eclipse are relatively easy to derive with simple geometry and trigonometry. Figure 14 shows a simplified version of Figure 3, with the radii of the Sun and the Earth labelled as $R_\odot$ and $R_\oplus$ respectively.

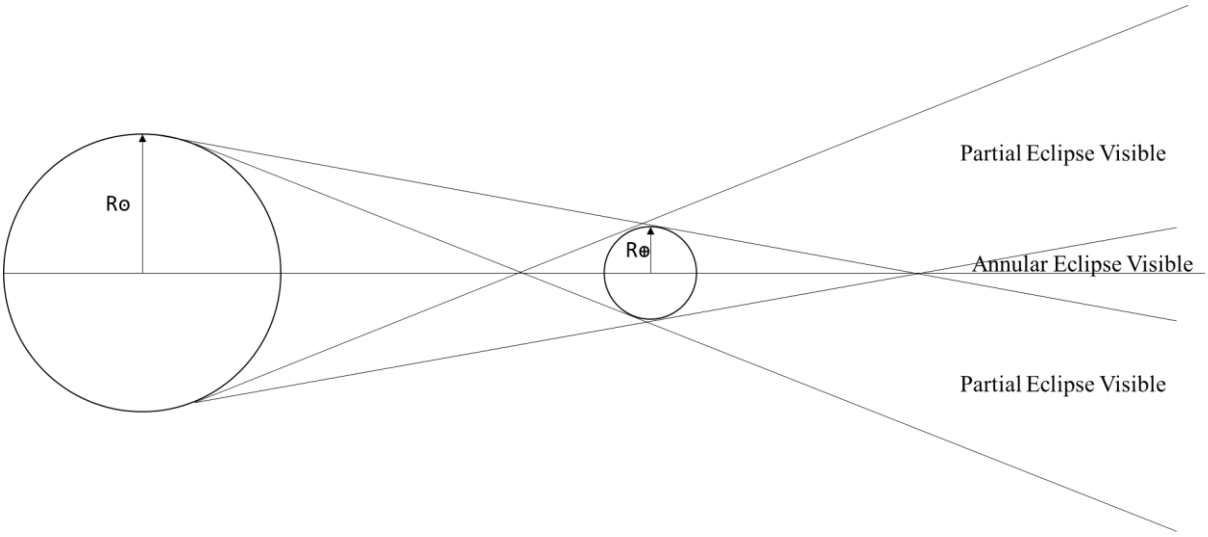


Figure 14: Simplified diagram showing the different zones able to see annular and partial eclipses of the Sun by the Earth with the radii of the Sun and the Earth labelled.

To calculate the angles previously described, it will make it easier to label points, lines, and angles as in Figure 15:

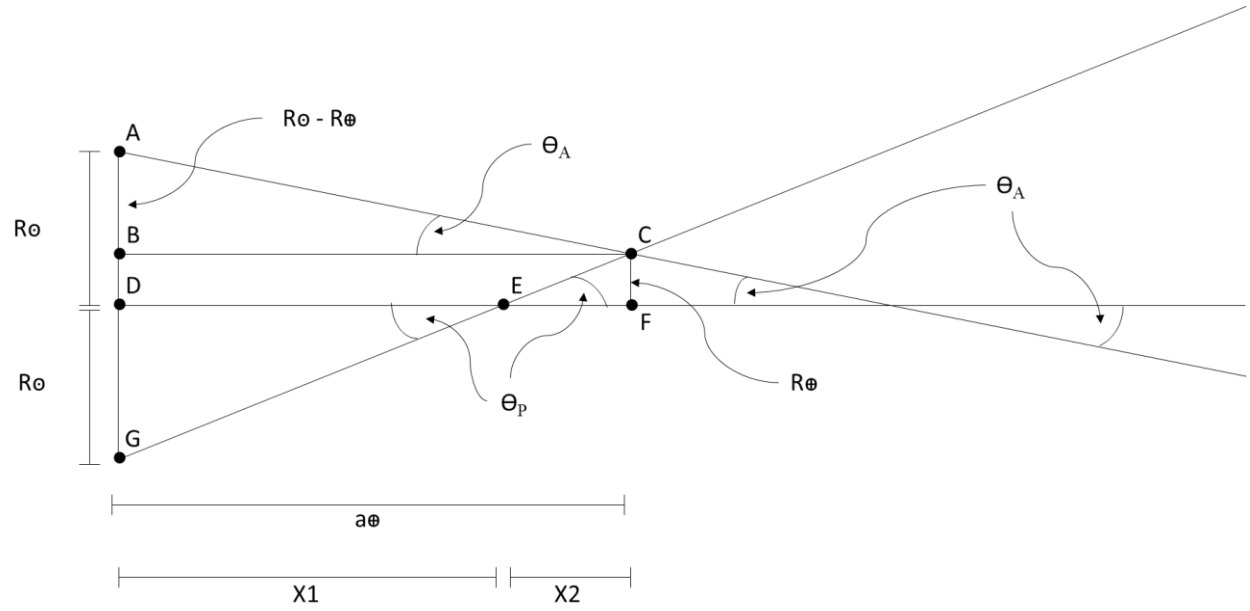


Figure 15: Orientation of the Sun and the Earth necessary for an observer to see partial and annular eclipses with points, lines, and angles labelled.

First calculating Θ_A, we can set up a simple tangent function with this angle and the know side lengths of triangle ABC in Figure 15, and then solve for Θ_A:

$$\tan(\Theta_A) = \frac{R_\odot - R_\oplus}{a_\oplus} \Rightarrow \Theta_A = \tan^{-1}\left(\frac{R_\odot - R_\oplus}{a_\oplus}\right)$$

Plugging in known values $R_\odot = 6.96 \times 10^8$ m, $R_\oplus = 6.38 \times 10^6$ m, and $a_\oplus = 1.5 \times 10^{11}$ m [12], we get:

$$\Theta_A = \tan^{-1}\left(\frac{(6.96 \times 10^8\text{m}) - (6.38 \times 10^6\text{m})}{(1.5 \times 10^{11}\text{m})}\right) = 0.004597\text{rad} = 0.263°$$

Now calculating Θ_P, we have two similar triangles in Figure 15, that is triangles CEF and DEG. The ratios of their sides are proportional as such:

$$\frac{R_\odot}{x_1} = \frac{R_\oplus}{x_2}$$

Knowing that $a_\oplus = x_1 + x_2$, we can simplify this to be:

$$R_\oplus x_1 = R_\odot x_2 = R_\odot(a_\oplus - x_1) = R_\odot a_\oplus - R_\odot x_1$$

$$\Rightarrow R_\odot a_\oplus = R_\oplus x_1 + R_\odot x_1 = x_1(R_\oplus + R_\odot)$$

$$=> x_1 = \frac{R_\odot a_\oplus}{R_\oplus + R_\odot}$$

From triangle DEG, we can say:

$$\theta_P = \tan^{-1}\left(\frac{R_\odot}{x_1}\right) = \tan^{-1}\left(R_\odot\left(\frac{R_\oplus + R_\odot}{R_\odot a_\oplus}\right)\right) = \tan^{-1}\left(\frac{R_\oplus + R_\odot}{a_\oplus}\right)$$

$$=> \theta_P = \tan^{-1}\left(\frac{(6.96 \times 10^8 m) + (6.38 \times 10^6 m)}{(1.5 \times 10^{11} m)}\right) = 0.004682\text{rad} = 0.268°$$

Appendix B- Limb Darkening Transit Simulation Code

```
#---------------------------------------------------------------------
# Author: Paul Allen- allenp@allegheny.edu
#
# Affiliation: Physics Department, Allegheny College
#
# Limb Darkening Prototype Simulator 8
#
# Date Started: 3/7/22
#
# Description: This prototype simulation attempts to modify previous
# simulations by calculating luminosity of the Sun before simulating
# the planetary transit in an attempt to save time. This program
# works on the assumption that there is negligable change in sunlight
# blocked by different parts of the Earth's disk as it passes in front
# of the Sun since the Earth is so much smaller.
#---------------------------------------------------------------------

import math
import numpy
import matplotlib.pyplot as plt

print('Limb Darkening Prototype Simulation 8.')
print('Enter coordinates in terms of radians.')
RAsc = float(input('Right ascension of the observing star: '))
Dec = float(input('Declination of the observing star: '))
run = float #Triggers run if the Earth transit is visible

DecEcl = 0.41 * math.sin(RAsc) #Returns a declination value of the
#ecliptic for a given Right Ascension

UpDecETZ = DecEcl + (0.0042 + 0.00038 * abs(math.cos(RAsc))) #Returns a
#declination value of the upper bound of the ETZ for any given Right
#Ascension

LwDecETZ = DecEcl - (0.0042 + 0.00038 * abs(math.cos(RAsc))) #Returns a
#declination value of the lower bound of the ETZ for any given Right
#Ascension

if Dec < UpDecETZ and Dec > LwDecETZ:
    print('Star lies within the EZT and can view either an annular or \
total eclipse of the sun by the Earth.')
    run = 1
else:
    print('Star does not lie within the EZT and cannot view an eclipse \
of the sun by the Earth.')
    run = 0

RS = 1 #Radius of the Sun (in solar radii)
xS = 0 #Starting x-position of the Sun
RP = 0.0091577 #Radius of the Earth (in solar radii)
xP = -1.1 #Starting x-position of the Earth
if Dec >= DecEcl: #Starting y-position of the planet if it is above the ecliptic
```

```
    yP = (Dec - DecEcl) / (UpDecETZ - DecEcl)
elif Dec < DecEcl: #Starting y-position of the planet if it is below the ecliptic
    yP = (Dec - DecEcl) / (DecEcl - LwDecETZ)
print('yP: ', yP)
nstep = 10 / RP #number of pixels across star
#The positions of the two bodies are defined by the position of
#their centers. The position of the planet will change with time.

xP_list=[] #list containing the x-positions of the planet
percecl_list=[] #list containing the % starlight eclipsed at each x-position
                #of the planet

deltax = 2 * RS / nstep #size of x-steps
lbs = -RS + deltax / 2 #location of the left most pixel
ubs = -lbs #location of the right most pixel
pixsize = RS / nstep #the length of one side of each pixel (1/10 of the Earth's radius)

totradint = 0 #Total radiant intensity of the Sun

if run == 1: # Completes the simulation only if the Earth will pass in front of at least some of the Sun
# The following loop describes the luminosity of the Sun without it being eclipsed by the Earth:
    for spx in numpy.arange(lbs, ubs, deltax): #gives the x-values for each pixel in the Sun
        for spy in numpy.arange(lbs, ubs, deltax): #gives the y-values for each pixel in the Sun
            if spy < math.sqrt(RS ** 2 - spx **2) and spy > -math.sqrt(RS ** 2 - spx **2):
                #eliminates pixels outside the star's circle
                theta = math.atan(spx / spx * (RS ** 2 - spx ** 2 - spy ** 2) ** -2) #angle of exit
                    #out of the star for a ray of light based on location on solar disk
                pixradint = 1 - 0.648 * (1 - math.cos(theta)) - 0.207 * math.cos(theta) * math.log10\
                    (math.cos(theta)) #radiant intensity of the pixel
                totpixradint = pixradint * pixsize ** 2 #This is the total radiant intensity of the pixel;
                #it is the radiant intensity of a light ray leaving the Sun at the pixel, multiplied by the
                #are of the pixel.
                totradint = totradint + totpixradint #Adds the radiant intensity of the most recently
                #calculated pixel to the total radiant intensity of the sun. After all the pixels have been
                #cycled through, this calculates the total radiant intensity of the Sun.
    circradint = 0

print('totradint: ', totradint)

if run == 1: # Completes the simulation only if the Earth will pass in front of at least some of the Sun
#The following loop calculates the radiant intensity of light blocked by the Earth at any given point in
#front of the Sun.
    while xP < 1.1:
        percecl = 0
        xP_list.append(xP)
        if xP > lbs and xP < ubs: #Determines if the planet lies within the Sun's x-boundaries
            if yP < math.sqrt(RS ** 2 - xP **2) and yP > -math.sqrt(RS ** 2 - xP **2): #Determines if the
                #planet lies within the Sun's y-boundaries
                theta = math.atan(xP / xP * (RS ** 2 - xP ** 2 - yP ** 2) ** -2) #angle of exit
                    #out of the star for a ray of light based on location on solar disk
                radint = 1 - 0.648 * (1 - math.cos(theta)) - 0.207 * math.cos(theta) * math.log10\
                    (math.cos(theta)) #radiant intensity of the light leaving the Sun at the center of the
                    #circular area eclipsed by the Earth
                circradint = radint * math.pi * RP ** 2 #Calculates the total light blocked by the circle defining
                #the Earth, assuming that this is a constant value throughout the circle defining the Earth.
                percecl = circradint / totradint * 100 #percentage of the Sunlight eclipsed
```

```
            else:
                percecl = 0
        percecl_list.append(percecl)
        xP = xP + deltax

perceclmax = max(percecl_list)
print('perceclmax: ', perceclmax)

if run == 1:
    plt.plot(xP_list, percecl_list, '-ro')
    plt.title('Planetary Transit Light Curve')
    plt.xlabel('Position of Planet')
    plt.ylabel('% of Starlight Eclipsed')
    plt.grid(True, which='both')
    plt.show()
```

Appendix C- Real Data Analysis Code

```
#---------------------------------------------------------------------
# Author: Paul Allen- allenp@allegheny.edu
#
# Affiliation: Physics Department, Allegheny College
#
# Limb Darkening Prototype Simulator 9
#
# Date Started: 4/20/22
#
# Description: This prototype simulation builds off of ldsp8.py in
# that it attempts to read in information from a text file (the
# default is "GAIAData1.csv") rather than prompting the user to input
# data typographically. Additionally, it records only the maximum
# mount of light that the Earth blocks as seen from a particular star,
# rather than many different values of eclipsed light. The program
# then creates a histogram of these data and saves the GAIA number,
# right ascension, declination, and maximum percent of light blocked
# to an output file (the default is "GAIAData1output.txt"). Both
# default files can be overridden and the user can input from and save
# to other files if desired.
#---------------------------------------------------------------------

import math
import numpy as np
import matplotlib.pyplot as plt

print(' ')
print('Limb Darkening Prototype Simulation 9.')
print(' ')

RS = 1 #Radius of the Sun (in solar radii)
xS = 0 #Starting x-position of the Sun
RP = 0.0091577 #Radius of the Earth (in solar radii)
xP = -1.1 #Starting x-position of the Earth
nstep = 10 / RP #number of pixels across star
#The positions of the two bodies are defined by the position of
#their centers. The position of the planet will change with time.
deltax = 2 * RS / nstep #size of x-steps
lbs = -RS + deltax / 2 #location of the left most pixel
ubs = -lbs #location of the right most pixel
pixsize = RS / nstep #the length of one side of each pixel (1/10 of the Earth's radius)

#choosing the input file:
default1 = None

while default1 not in ("y", "n"):
    default1 = input('Use default file input? (y/n): ')
    if default1 == "y":
        filename1 = "GAIAData1.csv"
    elif default1 == "n":
        filename1 = input('Input File Name: ')
    else:
        print('Please enter "y" or "n": ')
```

```
#choosing the output file:
default2 = None

while default2 not in ("y", "n"):
    default2 = input('Use default file output? (y/n): ')
    if default2 == "y":
        filename2 = "GAIAData1output.txt"
    elif default2 == "n":
        filename2 = input('Output File Name: ')
    else:
        print('Please enter "y" or "n": ')

outfile1 = open(filename2, 'w') # w designates we are opening the file to write into it
outfile1.write('# in File, ') #this numbers the items in the output file
outfile1.write('GAIA #, ') #this records the heading for the first column (the GAIA number) in the output file
outfile1.write('RAsc, ') #this records the heading for the second column (the right asccension) in
    #the output file
outfile1.write('Dec, ') #this records the heading for the third column (the declination) in
    #the output file
outfile1.write('perceclmax') #this records the heading for the fourth column (maximum light eclipsed in the
    #output file
outfile1.write(" \n") #this starts a new line in the output file

linenumber = 1 #this sets the first line in the output file to be labelled "1"

totradint = 0 #Total radiant intensity of the Sun

# The following loop describes the luminosity of the Sun without it being eclipsed by the Earth:
for spx in np.arange(lbs, ubs, deltax): #gives the x-values for each pixel in the Sun
    for spy in np.arange(lbs, ubs, deltax): #gives the y-values for each pixel in the Sun
        if spy < math.sqrt(RS ** 2 - spx **2) and spy > -math.sqrt(RS ** 2 - spx **2):
            #eliminates pixels outside the star's circle
            theta = math.atan(spx / spx * (RS ** 2 - spx ** 2 - spy ** 2) ** -2) #angle of exit
                #out of the star for a ray of light based on location on solar disk
            pixradint = 1 - 0.648 * (1 - math.cos(theta)) - 0.207 * math.cos(theta) * math.log10\
                (math.cos(theta)) #radiant intensity of the pixel
            totpixradint = pixradint * pixsize ** 2 #This is the total radiant intensity of the pixel;
            #it is the radiant intensity of a light ray leaving the Sun at the pixel, multiplied by the
            #are of the pixel.
            totradint = totradint + totpixradint #Adds the radiant intensity of the most recently
            #calculated pixel to the total radiant intensity of the sun. After all the pixels have been
           #cycled through, this calculates the total radiant intensity of the Sun.
circradint = 0

lines1 = [] #Contents of the input file
lines2 = [] #Amended contents of the input file based on location (in ETZ)
xP_list=[] #list containing the x-positions of the planet as it transits across the Sun's face
perceclmax_list=[] #list containing the maximum amount of light blocked
Decmax_list=[] #list containing the declinations of the stars in ETZ

with open(filename1) as f: #opens file with input data
    lines1 = f.readlines() #reads the input file contents into a list

for line in lines1: #goes through  each line in the list individually
    sline = line.split(',') #splits the contents in each row into a new list
```

```
starnumber = sline[0] #this records the numbering on the star assigned by GAIA
RAsc = sline[1]
Dec = sline[2]
RAsc = float(RAsc)
Dec = float(Dec)

#Right ascension and declination are read in in terms of degrees. The following converts these
#values to radians
RAsc = RAsc * math.pi / 180
Dec = Dec * math.pi / 180

DecEcl = 0.41 * math.sin(RAsc) #Returns a declination value of the
#ecliptic for a given Right Ascension

UpDecETZ = DecEcl + (0.0042 + 0.00038 * abs(math.cos(RAsc))) #Returns a
#declination value of the upper bound of the ETZ for any given Right
#Ascension

LwDecETZ = DecEcl - (0.0042 + 0.00038 * abs(math.cos(RAsc))) #Returns a
#declination value of the lower bound of the ETZ for any given Right
#Ascension

if Dec < UpDecETZ and Dec > LwDecETZ: #if the star is in the ETZ
   #percecl_list=[] #list containing the % starlight eclipsed at each x-position of the planet
   if Dec >= DecEcl: #Starting y-position of the planet if it is above the ecliptic
      yP = (Dec - DecEcl) / (UpDecETZ - DecEcl)
   elif Dec < DecEcl: #Starting y-position of the planet if it is below the ecliptic
      yP = (Dec - DecEcl) / (DecEcl - LwDecETZ)
   RAsc = str(RAsc)
   Dec = str(Dec)
   yP = str(yP)
   lines2.append(RAsc) #Adds right ascension for stars within the ETZ to this list
   lines2.append(Dec) #Adds declination for stars within the ETZ to this list
   lines2.append(yP) #Adds starting y-position for stars within the ETZ to this list
   Decmax_list.append(Dec)
   yP = lines2[2] #takes the third element (declination) from the list containing the amended input
   #data file

   #The following calculates the maximum radiant intensity of light blocked by the Earth (the light
   #blocked at xP = 0). It uses the values stored in lines2.
   yP = float(yP)
   xP = -0.0022 #the is the closest x-value the transit simulation in ldsp8.py ever gets to zero. We do
      #not say xP = 0 here because we would get a division by zero error when calculating theta
   theta = math.atan(xP / xP * (RS ** 2 - xP ** 2 - yP ** 2) ** -2) #angle of exit out of the star for
      #a ray of light based on location on solar disk
   radint = 1 - 0.648 * (1 - math.cos(theta)) - 0.207 * math.cos(theta) * math.log10(math.cos(theta))
      #radiant intensity of the light leaving the Sun at the center of the circular area eclipsed by
      #the Earth
   circradint = radint * math.pi * RP ** 2 #Calculates the total light blocked by the circle defining the
      #Earth, assuming that this is a constant value throughout the circle defining the Earth.
   perceclmax = circradint / totradint * 100 #percentage of the Sunlight eclipsed
   perceclmax_list.append(perceclmax)

   outfile1.write(str(linenumber))
   outfile1.write(',') #this separates the entries in the output file
   outfile1.write(starnumber) #this enters the GAIA star number into the output file
```

```
        outfile1.write(',') #this separates the entries in the output file
        outfile1.write(RAsc) #this enters the right ascension of this star into the output file
        outfile1.write(',') #this separates the entries in the output file
        outfile1.write(Dec) #this enters the declinaiton of this star into the output file
        outfile1.write(',') #this separates the entries in the output file
        outfile1.write(str(perceclmax)) #this enters the maximum amount of sunlight blocked for this star into
            #the output file
        outfile1.write(" \n") #this starts a new line in the output file

        linenumber = linenumber + 1
    lines2.clear()

#creating dataset containing the maximum percents
a = np.array(perceclmax_list)

#creating histogram
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a, bins = [0.02, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05])
plt.title('Histogram of % Sunlight Stars See Eclipsed')
plt.xlabel('% Sunlight Eclipsed')
plt.ylabel('Number of Stars')
plt.show()

outfile1.close() #Closes the output file
```

Appendix D- Data Access Code

```
#---------------------------------------------------------------------
# Downloaded by Paul Allen- allenp@allegheny.edu
#
# Affiliation: Physics Department, Allegheny College
#
# GAIA Catalogue Access Test Program 1
#
# Date Started: 3/7/22
#
# Description: This is an attempt to access the Gaia Catalogue using
# code taken from https://www.cosmos.esa.int/web/gaia-users/archive/pr
# ogrammayic-access#PythonFlag
#---------------------------------------------------------------------

#Python 3
import http.client as httplib
import urllib.parse as urllib
import time
from xml.dom.minidom import parseString

host = "gea.esac.esa.int"
port = 443
pathinfo = "/tap-server/tap/async"


#--------------------------------------
#Create job

params = urllib.urlencode({\
        "REQUEST": "doQuery", \
        "LANG":    "ADQL", \
        "FORMAT":  "votable_plain", \
        "PHASE":  "RUN", \
        "JOBNAME":  "Any name (optional)", \
        "JOBDESCRIPTION":  "Any description (optional)", \
        "QUERY":   "SELECT DISTANCE(POINT('ICRS',ra,dec), POINT('ICRS',266.41683,-29.00781)) AS dist,
* FROM gaiadr1.gaia_source  WHERE 1=CONTAINS(POINT('ICRS',ra,dec),CIRCLE('ICRS',266.41683,-
29.00781, 0.08333333)) ORDER BY dist ASC"
        })

headers = {\
        "Content-type": "application/x-www-form-urlencoded", \
        "Accept":       "text/plain" \
        }
```

```
connection = httplib.HTTPSConnection(host, port)
connection.request("POST",pathinfo,params,headers)

#Status
response = connection.getresponse()
print ("Status: " +str(response.status), "Reason: " + str(response.reason))

#Server job location (URL)
location = response.getheader("location")
print ("Location: " + location)

#Jobid
jobid = location[location.rfind('/')+1:]
print ("Job id: " + jobid)

connection.close()

#--------------------------------------
#Check job status, wait until finished

while True:
        connection = httplib.HTTPSConnection(host, port)
        connection.request("GET",pathinfo+"/"+jobid)
        response = connection.getresponse()
        data = response.read()
        #XML response: parse it to obtain the current status
        #(you may use pathinfo/jobid/phase entry point to avoid XML parsing)
        dom = parseString(data)
        phaseElement = dom.getElementsByTagName('uws:phase')[0]
        phaseValueElement = phaseElement.firstChild
        phase = phaseValueElement.toxml()
        print ("Status: " + phase)
        #Check finished
        if phase == 'COMPLETED': break
        #wait and repeat
        time.sleep(0.2)


connection.close()

#---------------------------------------
#Get results
connection = httplib.HTTPSConnection(host, port)
connection.request("GET",pathinfo+"/"+jobid+"/results/result")
response = connection.getresponse()
data = response.read().decode('iso-8859-1')
#print(type(data))
#print(data)
outputFileName = "example_votable_output.vot"
```

```
outputFile = open(outputFileName, "w")
outputFile.write(data)
outputFile.close()
connection.close()
print ("Data saved in: " + outputFileName)    c
```

References

[1] L. Kaltenegger, "Past, present and future stars that can see," *Nature,* vol. 594, pp. 505-507, 2021.

[2] M. Gillon, "Seven temperate terrestrial planets around the nearby ultracool dwarf-star TRAPPIST-1," *Nature,* vol. 542, p. 456–460, 2017.

[3] D. A. F. A. Wolszczan, "A planetary system around the millisecond pulsar PSR1257 + 12," *Nature,* vol. 355, p. 145–147, 1992.

[4] J. J. Lissauer, R. I. Dawson and S. Tremaine, "Advances in exoplanet science from Kepler," *Nature,* vol. 513, no. 7518, pp. 336-344, 2014.

[5] "NASA Exoplanet Archive," NASA Exoplanet Science Institute , [Online]. Available: https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=PS. [Accessed 12 November 2021].

[6] B. R. Mushailov and V. S. Teplitskaya, "On Reliability of Determining the Orbital Parameters of Exoplanets by the Doppler Method," *Cosmic Research ,* vol. 50, p. 421–430, 2012.

[7] A. P. Hatzes, "The role of space telescopes in the characterization of transiting exoplanets," *Nature,* vol. 315, no. 7518, pp. 353-357, 2014.

[8] A. Strickland, "The most powerful telescope ever built is about to change how we see the universe," CNN, 19 November 2021. [Online]. Available: https://www.cnn.com/2021/11/15/world/james-webb-space-telescope-explainer-scn/index.html. [Accessed 6 November 2021].

[9] F. Graham, *Daily briefing: Flawless deployment for James Webb Space Telescope*, Nature, 2022.

[10] M. T. G. e. al, "Characterizing Atmospheres of Transiting Earth-like Exoplanets Orbiting M Dwarfs with James Webb Space Telescope," *Publications of the Astronomical Society of the Pacific,* p. 14, 2021.

[11] C. D. F. a. C. D. Dressing, "A Framework For Optimizing Exoplanet Target Selection For," *Publications of the Astronomical Society of the Pacific,* p. 32, 2020.

[12] B. W. Caroll and D. A. Ostlie, An Introduction to Modern Astrophysics, New York : Cambridge University Press , 2017.

[13] R. E. Wilson and W. V. Hamme, "UNIFICATION OF BINARY STAR EPHEMERIS SOLUTIONS," *The Astrophysical Journal,* vol. 780, no. 151, 2014.

[14] "GAIA Archive," European Space Agency , [Online]. Available: https://gea.esac.esa.int/archive/.

[15] "The Astronomical Almanac," 2020. [Online]. Available: https://asa.hmnao.com/SecM/Glossary.html#_A.

[16] "GAIA Archive," European Space Agency, [Online]. Available: https://gea.esac.esa.int/archive/.

[17] H. G. R. O. A. A.-O. Y. A. T. Kodama, "Inner Edge of Habitable Zones for Earth-Sized Planets," *JGR: Planets,* vol. 124, no. 8, pp. 2306-2324, 2019.

[18] J. F. Kasting and C. E. Harman, "Inner edge of the habitable zone," *Nature,* vol. 504, no. 7479, pp. 221-223, 2013.

[19] "Energy for microbial life on Europa," *NATURE,* vol. 403, 2000.

[20] D. J. L. L. F. T. Vincent R. Ekea, "How thick are Mercury’s polar water ice deposits?," *Icarus*, vol. 248, pp. 407-415, 2017.

[21] B. Resnick, "The largest space telescope in history is about to blow our minds," Vox, 4 1 2022. [Online]. Available: https://www.vox.com/science-and-health/22664709/james-webb-space-telescope-launch-date-december-science-hubble.

Acknowledgments

Firstly of course, thank you Dr. Lombardi for all your patience and guidance with this and all your other classes. I could not have asked for a better learning environment. Dr. Willey, even though your electricity and magnetism course was probably one of the most difficult I've taken, it was also one of the most enjoyable. Especially given that the fall of 2020 was a particularly strange semester to figure out, I really appreciate how much effort you put in. Thank you to the rest of the physics department as well, Dr. Petasis, Dr. Poynor, Dr. Statman, and Dr. Rahman; these four years have been great.

Mom and Dad, you guys are so amazing and supportive. Nothing I can put here can really do justice to how much you guys mean to me. Thank you, and I love you.

Thank you, Grace, for helping a color scientist start to step out of his grey-scale lab.

Everyone else as well: to Joanie, Toby, Max, Tera, Stuart, Susie, Mateusz, Maria, Caramel, Zelda, Andy, Pat, Roan, Teagan, Carrie, Carol, Lee, Zeke, Judy, Katie, Jean, Ellen, Thomas, Callaghan, Kurt, Anna, Dan, Maggie, Jean, Tony, Bill, Beverly; all of you, my family. To Capel, Alex, Connor, Matt, Josh, Isaac, Lili, Max, Kate, Russell, Matt, Carson; amazing friends. To Mr. Kozlowski, for being the absolute best; Carl Sagan reborn.

Most of all, for Claire.