Detecting Transiting Exoplanets in Crowded Fields

Eddie Larsson

Lund Observatory
Lund University

2022-EXA197

Degree project of 15 higher education credits
June 2022

Supervisor: Jens Hoeijmakers

Lund Observatory
Box 43
SE-221 00 Lund
Sweden

Abstract

The first planet orbiting a main-sequence star was discovered in 1995. Since then more than 4000 additional exoplanets have been discovered. This is a rapidly evolving field in astrophysics. One of the most common ways to detect an exoplanet is through transit, as the brightness of the host star decreases. The transit method is suitable for surveying a large number of stars at the same time and most of the exoplanets discovered so far are from this particular method. In this study, it was investigated if it would be feasible to detect transiting exoplanets in globular clusters. So far globular clusters have been surveyed but no exoplanets have been detected in these environments. The possibility of finding exoplanets in globular clusters was investigated by simulating an observation of a globular cluster once every half an hour for the span of five days with a simulated telescope and detector. Light curves were measured from the stars with simulated photon noise and read-out noise. The box least squares method was then used to detect transiting exoplanets from extracted light curves. The efficiency for finding exoplanets was 31% in a cluster with 20 000 stars, 17% in a cluster with 50 000 stars, 1.5 % in a globular cluster with 100 000 stars, and 0% in a globular cluster with 150 000 stars. This leads to the conclusion that surveys for exoplanets are limited by how close the stars are to each other as they can not be resolved and this diminishes the efficiency. However, in clusters with a smaller number of stars exoplanets can be detected as it is less crowded.

Popular science summary

I vårt universum finns det ett nästan ändlöst antal extra-solära planeter, så kallade exoplaneter. Den första planeten utanför vårt solsystem upptäcktes 1995 och antalet har ökat drastiskt sedan dess. Ett av de vanligaste sätten att upptäcka exoplaneter är genom att använda transitmetoden. När en planet är i omloppsbana runt en stjärna så kan det hända att planeten blockerar en del ljus som stjärnan strålar ut från vårat perspektiv vilket vi kan observera. Men det finns miljöer i rymden som har väldigt många stjärnor i ett litet område, och i dessa miljöer är det svårt att upptäcka exoplaneter genom transitmetoden. Ett exempel på en sådan miljö är klotformiga stjärnhopar. En klotformig stjärnhopp är en stor samling stjärnor med mer och mer stjärnor ju närmare centrum man är och det finns hundratals sådana i vår galax.

Det finns ett intresse att hitta exoplaneter i dessa miljöer. Astronomer vill kunna hitta exoplaneter i så många olika miljöer som möjligt för att kunna nå en bättre förståelse till hur planeter skapas och utvecklas. Om planeter kan upptäckas i dessa miljöer skulle det ge nya insikter till processerna bakom hur planeter formas och utvecklas.

Det finns också anledningar till varför just transitmetoden ska användas i detta arbetet. Det finns ett flertal olika metoder för att upptäcka exoplaneter och de har alla olika fördelar och nackdelar. En av fördelarna med transitmetoden är att det är det enda sättet att veta radien av en exoplanet. Transitmetoden kan även användas när ett stort antal stjärnor observeras samtidigt, för många andra metoder kan bara en stjärna i taget observeras. Detta är speciellt användbart när miljöer så som klotformiga stjärnhopar observeras eftersom det är så många stjärnor där. Eftersom alla olika metoder har olika fördelar och nackdelar så är det ideellt att använda ett flertal olika metoder för att undersöka egenskaperna av exoplaneter.

I detta projektet undersöktes det om transitmetoden kunde användas för att upptäcka exoplaneter i miljöer med mycket stjärnor genom en datorsimulering eftersom en upptäckt av en exoplanet i en sådan miljö skulle kunna vidga förståelsen av exoplaneter och hur dem formas

Contents

List of Figures

1 Introduction

The transit method is one of the most important ways of detecting exoplanets. Of extra interest are the Hot Jupiter type planets, planets as large as Jupiter but significantly closer to their host stars. These planets are ideal for using the transit method, as they are larger and block out more light. They also have shorter orbital periods, so it takes a shorter observation time to be able to observe several transits and be more certain that one is in fact observing a transiting exoplanet. Hot Jupiters are a very surprising type of planet. Generally it is thought that giant planets are formed early in the life of a star in the cooler regions farther away from the star where planet-forming materials are in ice-form. Therefore giant planets would be expected to be farther away from the star. This begs the question of how these hot Jupiter planets are formed. To better understand the different processes that influence how these planets are formed one would ideally like to search for them in as many environments as possible, for example searching for exoplanets in crowded environments (Haswell, C.A 2010). It would be of interest to find transiting exoplanets in crowded fields since the environment where an exoplanet is formed has influence over the formation process, for example dynamical processes can influence the formation of planets. However, there is no conclusive evidence for dynamical effects influencing the early stages of planetary formation, though this might just be because of how short-lived those stages are (de Juan Ovelar, M. et al 2018). In the interest of better understanding how these dynamical effects influence planetary formation, one would like to find exoplanets there. However, in crowded environments it can be difficult to find any exoplanets and one needs to observe a large number of stars for enough time to observe several transits, which takes a lot of time and data.

One example of such an environment is globular clusters. Globular clusters are spherical compact gravitationally bound collections of stars with a sharp increase of brightness towards the center. They are thought to be formed alongside galaxies and are as such many billions of years old (P., Sookmee 2020). Globular clusters are often metal-poor environments and it is possible that this lack of metal prohibits the formation of gas giants. (Gratton, R. et al 2019). However, to better understand the processes behind how Hot Jupiters are formed and if the metallicity has an influence on this one would like to search for Hot Jupiters in globular clusters. These environments are particularly challenging when using the transit method. This is caused by the fact that the stars in the center are difficult to resolve and also that the transit method has a high likelihood of finding false positives when detecting exoplanets. These false positives are mostly from fluctuating flux from other nearby stars, for example eclipsing binaries (Santerne, A. 2013). It is then important to develop strategies to mitigate contamination by closely separated stars. Contamination limits the stars that can be surveyed for exoplanets. When there is a limited amount of stars one can observe and the fraction of stars hosting a transiting exoplanet is small it then raises the question if it is at all possible to find exoplanets in crowded environments by using the transit method. To explore this problem in-depth, in this project we will model a globular cluster with a population of transiting hot Jupiters.

2 Theory

2.1 Transiting exoplanets

When an exoplanet is transiting the apparent brightness of the host star decreases. This is illustrated in figure 1. The decrease in brightness is proportional to the area of the planet. This is the only way to measure the size of an exoplanet, which is important for then deriving the density and constraining the composition of a planet. More specifically the formula

$$\frac{\Delta F}{F} = \frac{R_p^2}{R_s^2}, \tag{1}$$

can be used, where ΔF is the change in flux, F is the flux measured from the star, R_p is the radius of the planet, and R_s is the radius of the star.

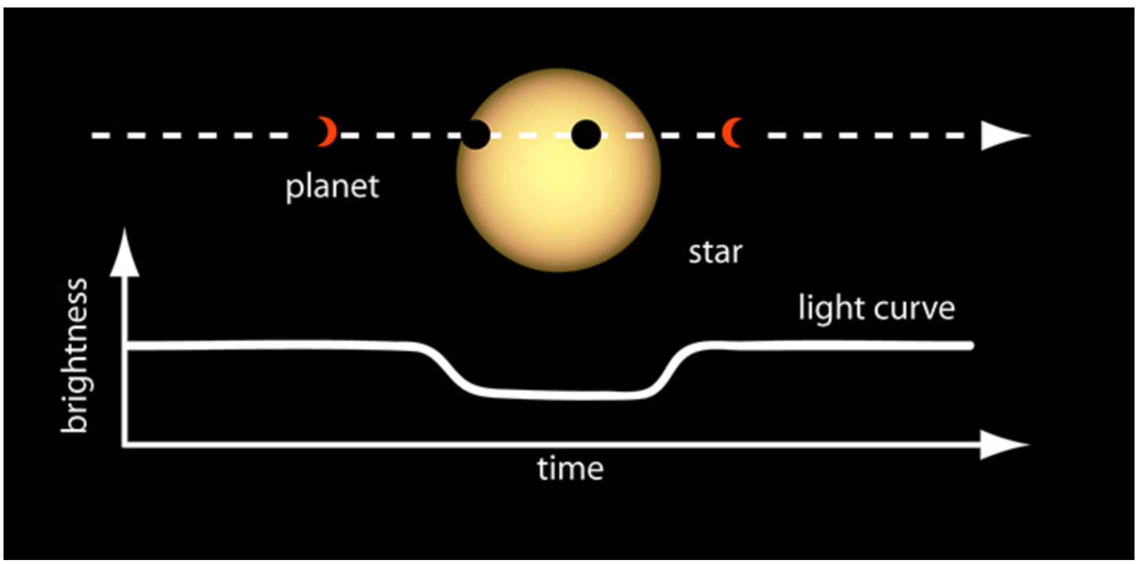

Figure 1: An illustration of a transiting exoplanet, taken from NASA Ames.

There are several parameters that prescribe the orbit of an exoplanet. The semi-major axis can be described using Kepler's third law.

$$a \approx \left(GM_s \left(\frac{P}{2\pi} \right)^2 \right)^{1/3}, \tag{2}$$

where G is the gravitational constant, M_s is the mass of the star, and P is the orbital period. Then the probability that the exoplanet is transiting from the perspective of the observer can be calculated as

$$GTP = \frac{R_s + R_p}{a}. \tag{3}$$

If the planet actually does transit there are several parameters that can be derived from the earlier mentioned parameters. Based on the inclination i of the planet, the impact parameter can be calculated as

$$b = \frac{a \cdot \cos(i)}{R_s}. \tag{4}$$

The orbital geometry of a transiting exoplanet can be seen in figure 2.

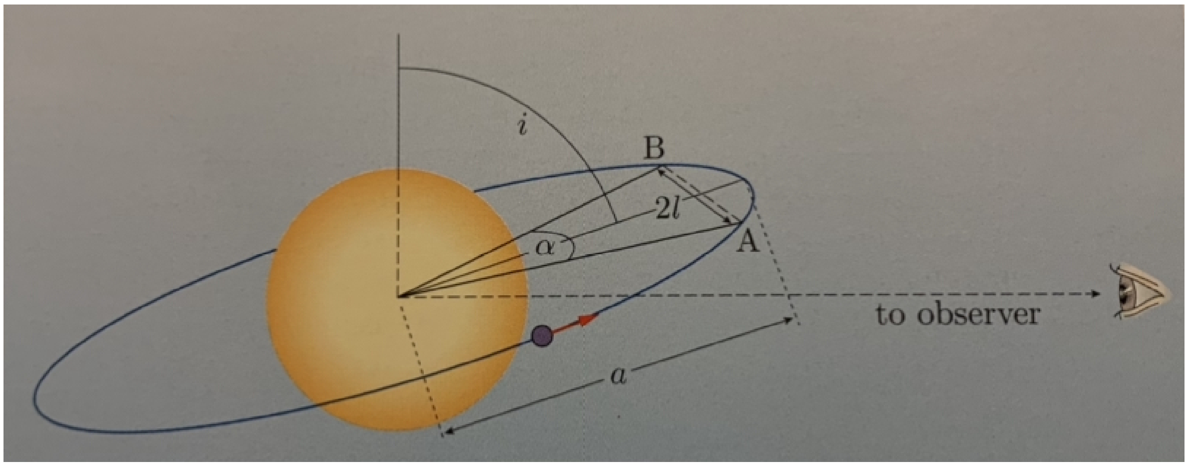

Figure 2: An illustration of the geometry of a transiting exoplanet, were between the points A and B in its orbit it is in transit, taken from Haswell, C.A 2010.

The impact parameter is defined as the distance between the center of the planet and the center of the star at mid-transit and is a unitless number between 1 and 0. This can be seen in figure 3. The distance the planet travels across the star can be calculated using the impact parameter and the formula

$$l = \sqrt{(R_p + R_s)^2 - a^2 \cos(i)^2} \tag{5}$$

The angle α_{max} can be defined as

$$\arcsin(\frac{l}{a}) \tag{6}$$

and if the planet has an angle less than α_{max} or larger than $1 - \alpha_{max}$ in its orbit then it is in transit (Haswell, C.A 2010).

To fully understand the properties of the transiting exoplanets one also needs to understand some of the properties of the stars they orbit, so let us discuss those more in-depth.

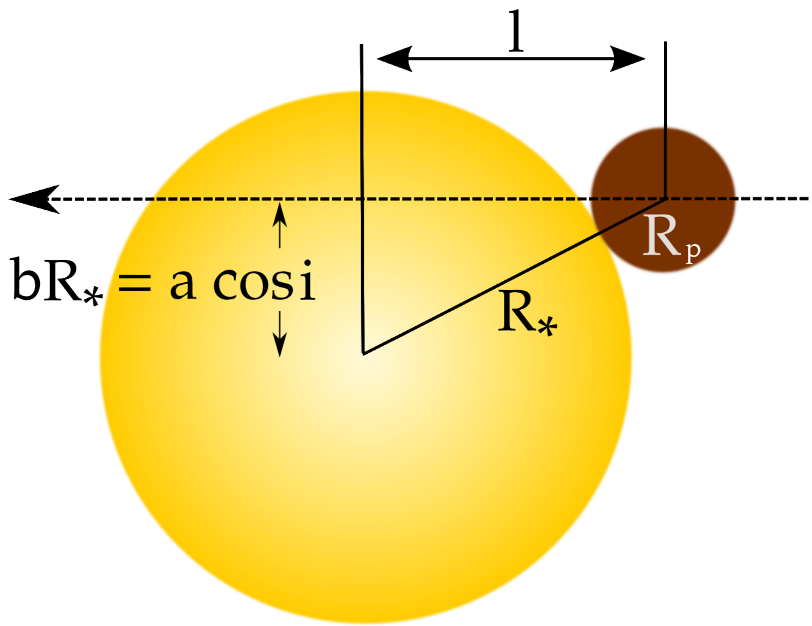

Figure 3: The impact parameter of a transiting exoplanet is how far away it travels from the center of the star during the transit. Taken from Wilson, P.A 2016.

2.2 Stars

In a globular cluster the surface density of stars as a function of the projected circular radius can be described using the formula (King, I.R 1966)

$$f(r) = \frac{f(0)}{1 + (\frac{r}{r_c})^2}. \tag{7}$$

In this formula r_c is the characteristic core radius while f(0) is the brightness in the center of the globular cluster. The number of stars in a Globular cluster can range from 10^5 in smaller clusters to 10^6 in the largest ones (Cintio, P.D 1996).

It is assumed that the target stars are main-sequence stars. For main sequence stars the luminosity can be approximated as (Rolfs, C.E 1988)

$$L \propto M^{3.5}, \tag{8}$$

where L is the luminosity in solar luminosity and M is the mass in solar masses. This relationship is illustrated in figure 4.

These relationships allow us to prescribe the properties of the stars by randomly drawing masses from the following distribution (Chandar, R. et al 2016)

$$\frac{dN}{dM} = M^{-2}, \tag{9}$$

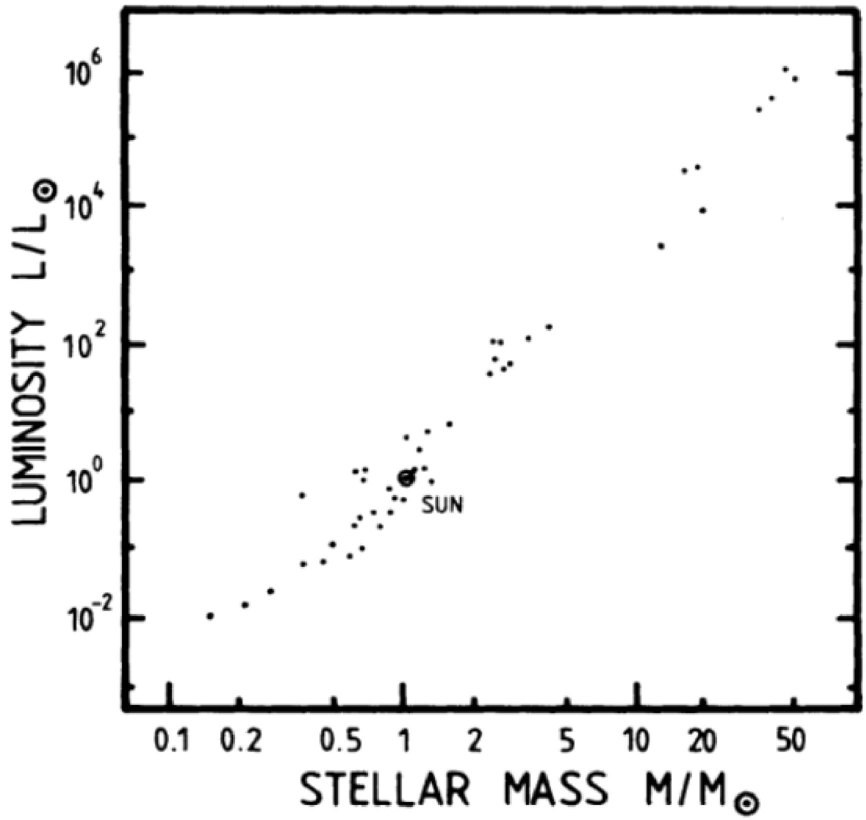

Figure 4: A diagram of the relationship between mass and luminosity for main-sequence stars. Taken from Rolfs, C.E 1988.

where N is the number of stars and M is the mass of the stars. The masses of main-sequence stars were chosen to be limited between 0.3 and 100.
Then the radii of the stars could be approximated using the formula (Kitchin, C.R 1987)

$$R_{star} = R_{sol} \cdot (\frac{M_{star}}{M_{sol}})^{0.9} \tag{10}$$

Here R_{sol} is the radius of the sun, M_{sun} is the mass of the sun, and M_{star} is the mass of the star.

The effective temperature of a star can be described using the equation (Tayler, R.J. 1994)

$$T_{eff} = (\frac{L}{4\pi\sigma R_{star}})^{1/4}, \tag{11}$$

where L is the stellar luminosity, R_{star} is the stellar radius, and σ is the Stefan-Boltzmann constant. The specific intensity that each star then radiates can theoretically be calculated using Planck's law, which can be written as (Rolfs, C.E 1988)

$$B_\nu = \frac{2h\nu^3}{c^2}\frac{1}{exp(h\nu/kT)-1}. \tag{12}$$

In this formula ν is the frequency, c is the velocity of light, T is the temperature, h is Planck's constant, and k is the Boltzmann constant. Choosing one specific wavelength, and therefore frequency, to observe the stars, gives a value for the specific intensity through Planck's law, while the detector itself measures a flux with the same value. The flux that is measured from the star will vary as a function of time and repeated observations with a simulated telescope and detector will yield light curves in which we can search for transit events. To fully understand the limitations of finding transiting exoplanets in crowded environments and how to create synthetic light curves we must look a bit more into how the optical systems work.

2.3 Optical Systems

Diffraction is a phenomenon that occurs when a light beam is partially blocked by an object and then the light is scattered around the object. This creates light and dark bands at the edge of the shadow (Heavens, O.S 1991). In any optical system diffraction will occur which leads to the light being spread out and effectively blurred. If an optical system only has degradation caused by diffraction it is diffraction limited. For a diffraction-limited telescope the first minimum occurs at (Meyers, R.A 2001)

$$\theta \approx 1.22\frac{\lambda}{D}, \tag{13}$$

were λ is the wavelength the telescope is observing and D is the diameter of the telescope and θ is in radians. The spread can be described by using the Point Spread Function (PSF). The PSF for a diffraction-limited telescope is an Airy pattern that can be approximated as a 2D Gaussian (Schowengerdt, R.A. 2007). The 2D Gaussian function that was used was

$$f(x,y) = A \cdot exp(\frac{(x-x0)^2}{2\sigma_x^2} - \frac{(y-y0)^2}{2\sigma_y^2}), \tag{14}$$

were $x0$ and $y0$ were the position of the stars, A was the amplitude of the function chosen such that that the integral of f is normalized, σ sets the angular width of the PSF effectively limiting the resolving power of the instrument. This blurring then needs to be considered to create a realistic simulation of the stars being observed.

For a realistic simulation one also needs to consider noise. Noise can be defined as the purely random fluctuations that depend on time that exists for many different physical variables (Lindsay, B.R 1966). This is a phenomenon that also occurs for photons. Photons are measured as a counting of discrete events, in this case the event is a photon arriving, which means that the statistical behavior of the photon can be determined by Poisson statistics. This means that the statistical fluctuations for photons can be described as $\sqrt{N}$, where N is the number of photons detected (Haswell, C.A 2010). This noise creates more difficulty in finding transiting exoplanets as well as creating potential false positives.

3 Methods

First, a globular cluster was simulated by creating a certain amount of stars and then giving them randomly chosen positions and masses following equations (7) and (9). Then the radii and flux of the stars would be calculated from formulas (10) and (12). Then these stars would be assigned planets, some of which would be transiting from the perspective of the observer. Then a detector was simulated to observe these transits. Apertures were created, which are just the pixels that were chosen to extract the light curve from the stars. There would be certain criteria for apertures to avoid observing contaminated stars. Lastly, a box least squares algorithm was implemented to try and detect any transiting exoplanets that might be orbiting these stars. Now we will look more in-depth at what every step entails.

3.1 Properties of Stars

First, the stars needed to be given positions. To calculate the positions of the stars formula (7) was used. The radius for the globular cluster was chosen to be 49 parsec, since globular clusters can have radii up to a few ten parsecs, it is a large globular cluster (Gratton, R et al 2019). First the positions of the stars were calculated in polar coordinates. In polar coordinates r is the distance from the center and θ is the polar angle, which is the angle around the x-axis and it can vary from 0 to 360 degrees. The globular cluster was split into fifteen different circle segments which are illustrated in figure 5. The number fifteen was chosen as the segments then become quite small and that gives a smooth distribution of the position. A smaller number would not give as good of a distribution of the positions while a larger number would not necessarily give much better results and take a longer time to compute. The normalized integral for formula (7) was used to determine the number of stars in every segment. The upper limit was the radius of the larger circle enclosing the segment while the lower limit was the radius of the smaller circle enclosed by the segment. The lower limit for the first segment is therefore zero. The value of this integral was then multiplied by the total number of stars to calculate the number of stars in that segment. Every star was then given a random r value within the circle segment as well as a random θ value. Then from the r value and θ, the x and y coordinates could be calculated using the formulas (Griffiths, D.J 2012)

$$x = r \cdot \cos(\theta)$$

$$y = r \cdot \sin(\theta).$$

An example of an image with 150 000 stars can be seen in figure 6, where the density of the globular cluster as a function of the radial distance can be seen in figure 7.

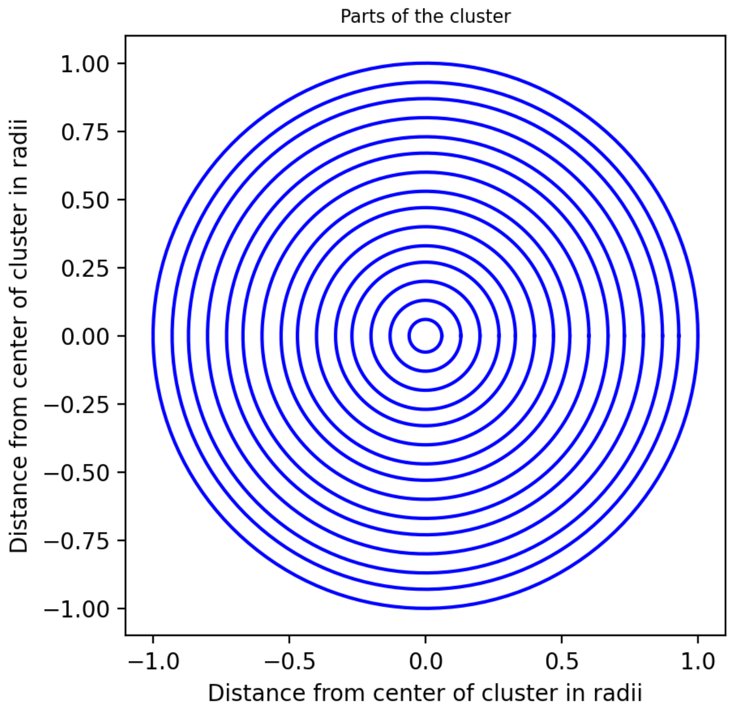

Figure 5: The radial bins the cluster is divided into for the purpose of generating the stellar positions

After this the properties of the stars were calculated. First the masses for the stars were calculated, which was done using formula (9). The range of masses for the stars was chosen to be between 0.3 and 100 solar masses. These were then divided into 2000 sections. The value of the normalized integral for formula (9) was calculated for each of these sections. Every star would be at random put into one of these sections, with the probability of the star being put into any particular section corresponding to the normalized integral. The stars would then be given random masses within the section. An example of the mass distribution can be seen in figure 8.

The radius and luminosity for all the stars were calculated using formulas (10) and (8) respectively. Then the effective temperature could be calculated using formula (11) and the flux was calculated from formula (12) .

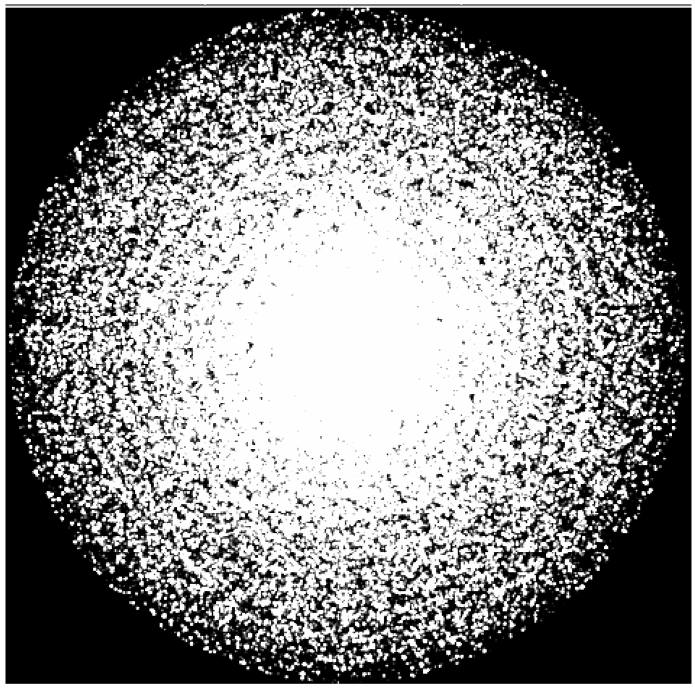

Figure 6: The simulated globular cluster with 150 000 stars.

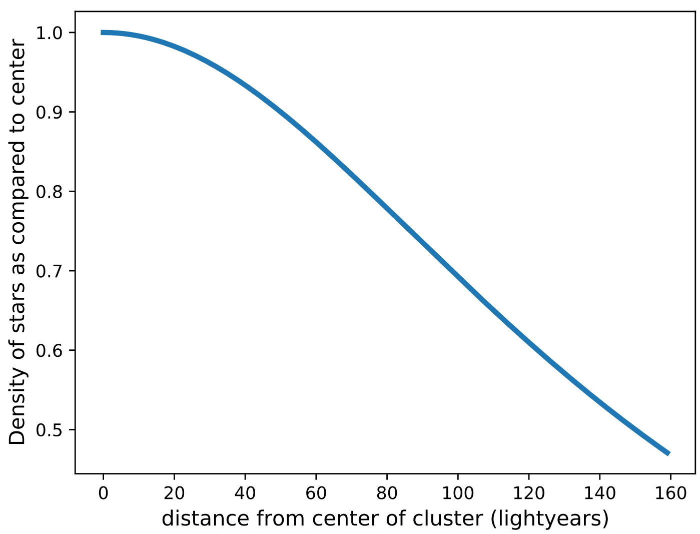

Figure 7: The density of the globular cluster as a function of the radial distance r.

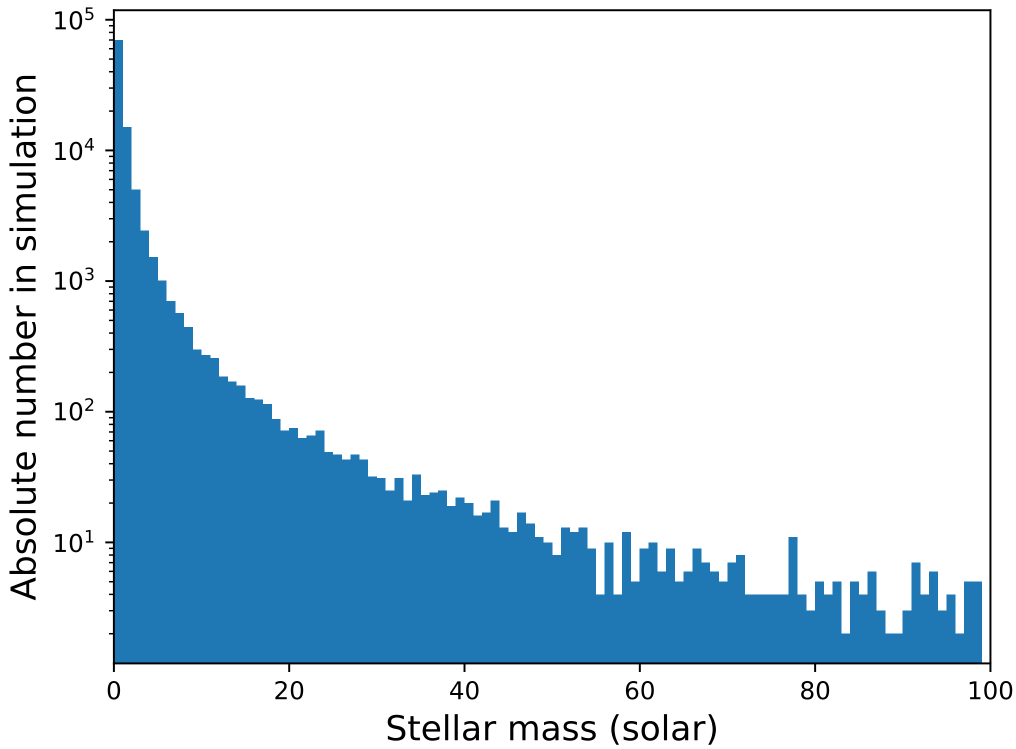

Figure 8: The mass distribution for a globular cluster with 100 000 stars. As can be seen most stars were lower-mass.

3.2 Planet Distribution and Transits

After the stars were defined they were randomly assigned exoplanets. For the planet distribution, the occurrence rate used was taken from Petigura E.A 2018, figure 9. In that study spectroscopy from the California-Kepler Survey, which is a large-scale spectroscopic survey of 1305 Kepler objects of interest, was used to examine the connection between planet occurrence and metallicity. The populations used in this project were the exoplanets with a radius 8 times larger than the earth or more, and an orbital period between 1 and 10 days. As mentioned in the introduction these Hot-Jupiter type planets are ideal when using the transit method with their large radii and short orbital periods, as well as being of scientific interest as their formation and evolution are still shrouded in mystery.

The selected radius-orbit bins used are highlighted in figure 9. For each box, planets would be drawn with random radii and orbital periods within the boxes. The semi-major axis for the planet was calculated using formula (2) and formula (3) would then be used to find the Geometric transit

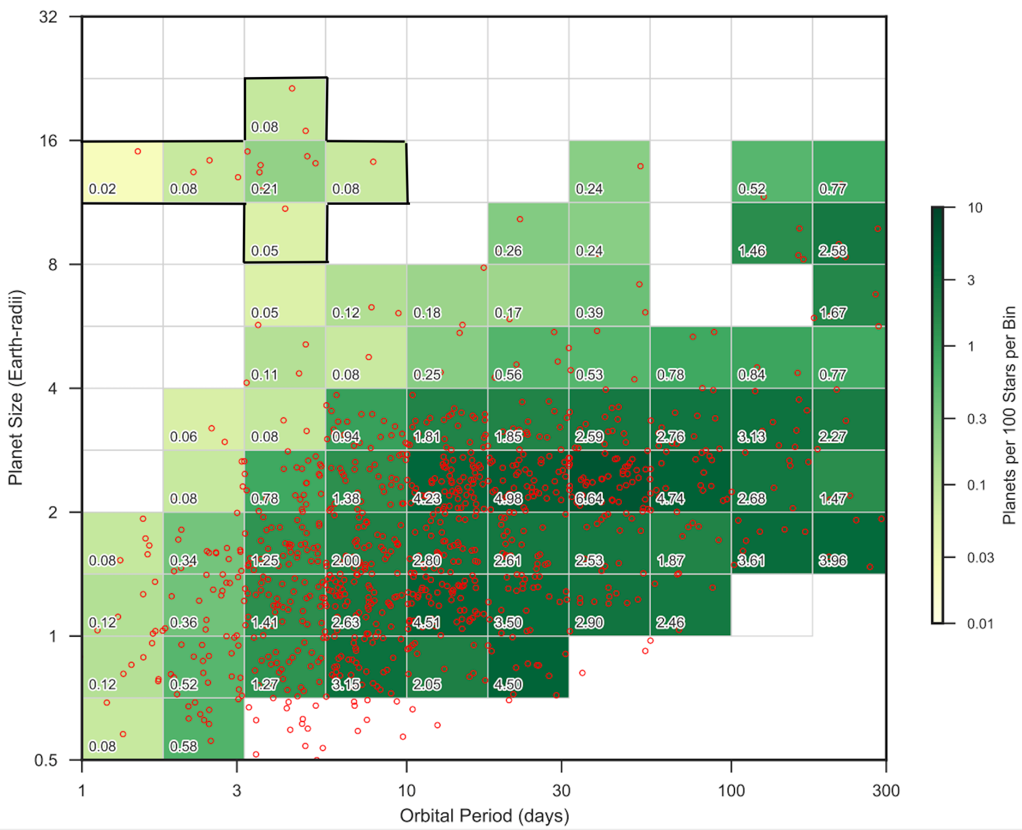

Figure 9: Data for the prevalence of different kinds of exoplanet among stars, were the number on each square signifies the amount of planets among 100 stars, the used boxes are highlighted. Image is taken from Petigura E.A 2018

probability. For each planet that did transit, it was given a random impact parameter b between one and zero. The light blocked by the planets was calculated using formula (1), the inclination of the planets using the formula (4), the distance the planet traveled across the star using formula (5), and the alpha max angle using formula (6). Every planet would be given a random starting value K for its orbital phase between 0 and 1 and its orbital phase could be calculated at any moment using the formula

$$\alpha = \frac{t}{P} - K, \tag{15}$$

where P is the orbital period. For values of $\alpha < \alpha_{max}$ or $\alpha > 1 - \alpha_{max}$, the planet is in transit, making the apparent flux of the star decrease.

3.3 Detector and Noise

To then be able to observe these transits a detector was simulated. A pixel image with 1000x1000 pixels was assumed to be used to observe the stars. The globular cluster was assumed to be 10 000 light years away and have a radius of 160 light years. The Angular size of the cluster can then be calculated as 1.8 degrees. The field of view was chosen to be two degrees, or 7200 arcseconds, so the entire cluster could be observed. This means that every pixel is equivalent to 7.2 arcseconds or 0.002 degrees. The dimensions of the detector are illustrated in figure 10. The simulated telescope observed the cluster at a wavelength of 625 nanometers. The full width half maximum was chosen to be 1.2 pixels which is 8.6 arcseconds. This value was chosen as the pixel scale becomes in between typical values for exoplanet transit survey instruments. Using formula (13) the diameter D for the telescope can then be calculated to be 1.8 centimeters which is quite a small telescope.

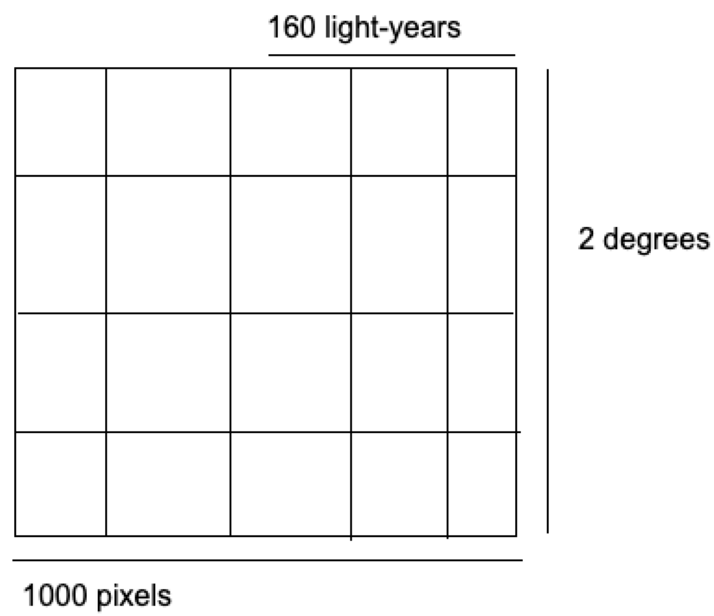

Figure 10: Dimensions of the simulated CCD detector.

The flux of every star at the pixels surrounding them was calculated using formula (14), where A=1. The amplitude A was chosen to be one because all of the fluxes of the stars would be re-normalized later so what was of significance was the difference between fluxes and not the flux itself.

Noise also needed to be added to the detector. The signal-to-noise ratio for photon noise is given by the square root of the number of events. Therefore the flux earlier calculated in erg / (cm2 Hz s sr) had to be converted into photon count. This was done by dividing the flux of all the pixels in the image by the flux of one of the stars in the 90th percentile in erg / (cm2 Hz s sr), and multiplying that by 50 000, which is a typical value of over-saturation for a CCD detector. A star in the 90th percentile was chosen so most of the stars would be well exposed with only the brightest stars being over-saturated. Any pixel that had a photon count value calculated to be over 50 000 would then instead be set to 50 000. this can be seen in figure 11. The photon shot noise for any particular pixel would then be $\sqrt{\frac{flux_{pixel}}{flux_{90}} \cdot 50000}$. It was assumed that the observation of the stars went on until the top 10% of the stars were saturated.

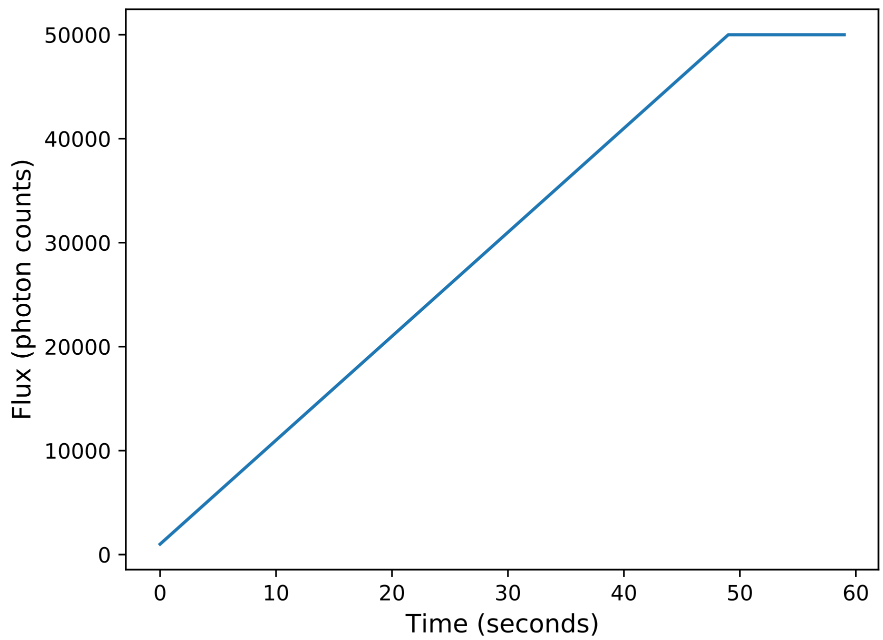

Figure 11: The flux measured increases linearly with time until it reaches 50 000 photon counts, at that point it has become over-saturated. In this illustration the exposure time is assumed to be one minute.

We also assumed that a constant bias level at 100 photon counts with a standard deviation of 2 added to simulate read out noise. Any pixel dimmer than 200 photon counts would not be analyzed as there was too much uncertainty.

3.4 Observation of Stars

When the detector has been simulated the simulated observation of the stars then needs to be established. To create observations of the stars a three-by-three pixels custom aperture was created for every star that fulfilled certain criteria. This was done to counter contamination by nearby stars, which could make the data useless. The closer the nearby stars were the brighter the star had to be to be considered uncontaminated. For every star there was a selection process. First, any star that was used had to have a maximum value under 50 000 counts over the nine pixels to avoid over-saturation. The sum of the flux value over the nine pixels also had to be over 2000 counts since below this the signal to noise ratio would not be high enough. If the star had any other stars less than 2σ pixels away these close-by stars had to be 10 times dimmer than the star for it to be selected, as it would otherwise be considered too contaminated by these other stars. Stars that had other stars somewhere between 3σ and 2σ pixels away would be considered if it was more than 5 times brighter than the stars in this region. If the star had other stars between 5 σ and 3 σ pixels away from it then it could be used if it was twice as bright as any of the stars in that region. If the star had a distance greater than 5 σ pixels away from the closest star it would be used. This is represented in figure 12. An example of an aperture can be seen in figure 13.

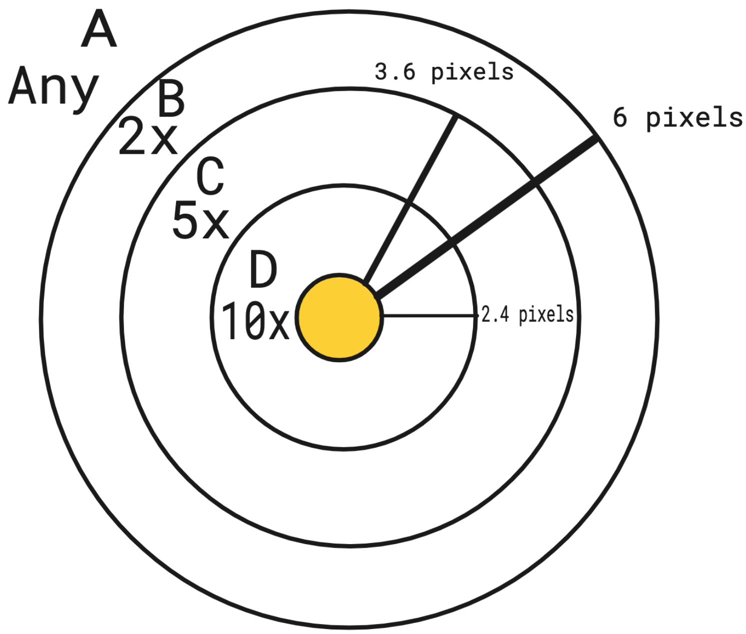

Figure 12: A schematic representation of the selection process. If there were any stars in region D the star would be selected if it was 10 times brighter than those stars. If there were any stars in region C the star would only be selected if it was 5 times brighter then any of those stars.If there were any stars in region D the star would only be selected if it was 2 times brighter then any of those stars. If the closest star was in region A the star in the center would be selected.

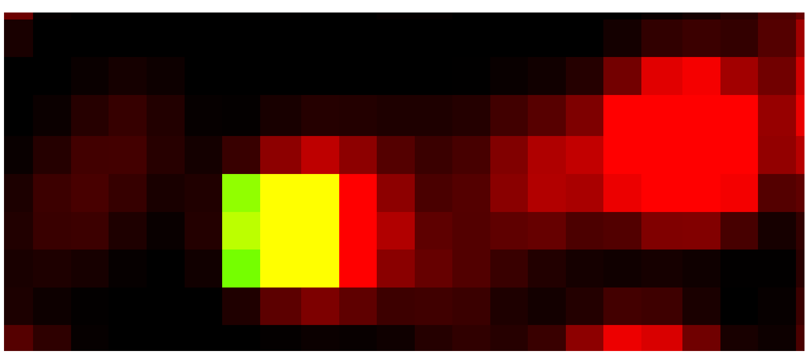

Figure 13: An aperture on a relatively bright star with a few bright stars nearby but none close enough to contaminate the target star. The red to black scale is the number of photon counts while the green pixels is the aperture

A time-series light curve was created from the measured flux of these apertures varying the flux of the stars if a planet was transiting. The uncertainty for every measurement was calculated as $\sqrt{\sum (RD + \sqrt{flux})^2}$, where RD was the standard deviation for the read-out noise, which was set to two. The sum is the total for all the measured pixels in the aperture, as can be seen in figure 15.Two of these light curves, one with noise and one without, can be seen in figures 14 and 15. This planet had a radius twice that of Jupiter, an impact parameter of 0.46, and an orbital period of 1.84 days while the star had a radius of 1.00 solar radii.

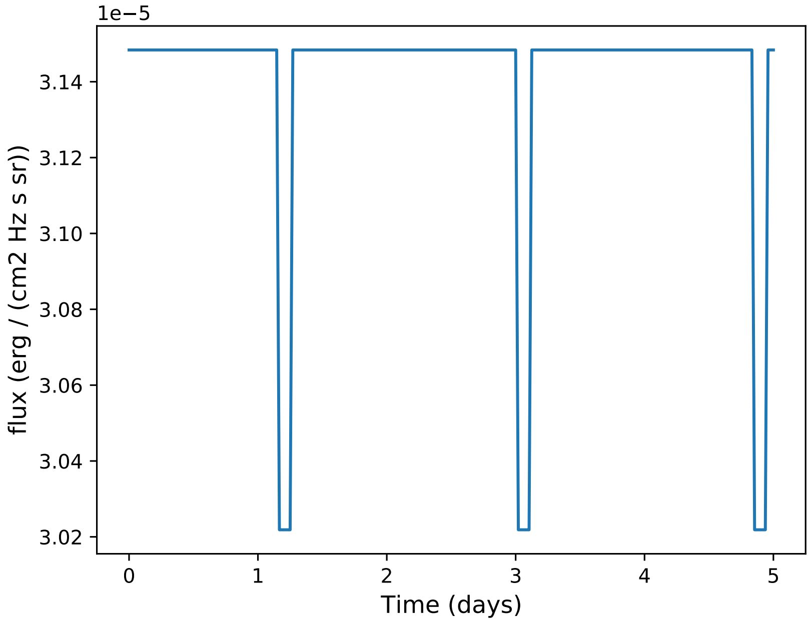

Figure 14: An example of a light curve for a star with a transiting exoplanet.

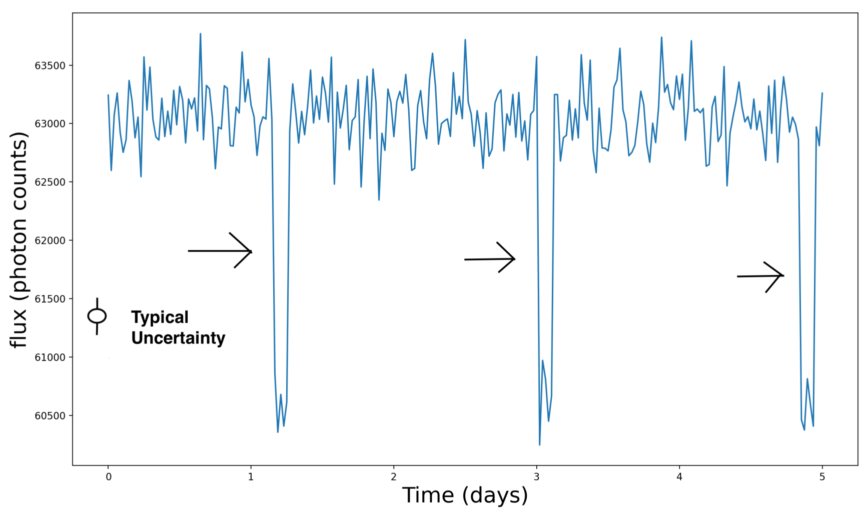

Figure 15: The same star but now with the flux re-normalized to counts, noise added, and the light curve extracted using an aperture. The typical uncertainty for the measurements is also illustrated in the figure.

3.5 Box least Squares

To detect transits in the extracted light curves the Box least squares method was used. The Box least squares method is an algorithm that is used to detect transiting exoplanets in time series photometric data. The BLS method utilizes four different parameters to create a step function, these parameters are duration, depth, period, and reference time. A step function is created with two different flux levels, one for when the planet is transiting and one for when the planet is not transiting. The Box least squares method then tries to find different parameters for the step function that best fits the extracted light curve (Kovacs, G. 2002). The reference time was chosen to be the mid-point of the first assumed observed transit of the data points. The four parameters can be seen in figure 16. For this study, the duration of every transit was assumed to be 0.2 days since that was the mean transit duration for transiting planets in the cluster with 20 000 stars, the transit duration for the planets can be seen in Figure 17.

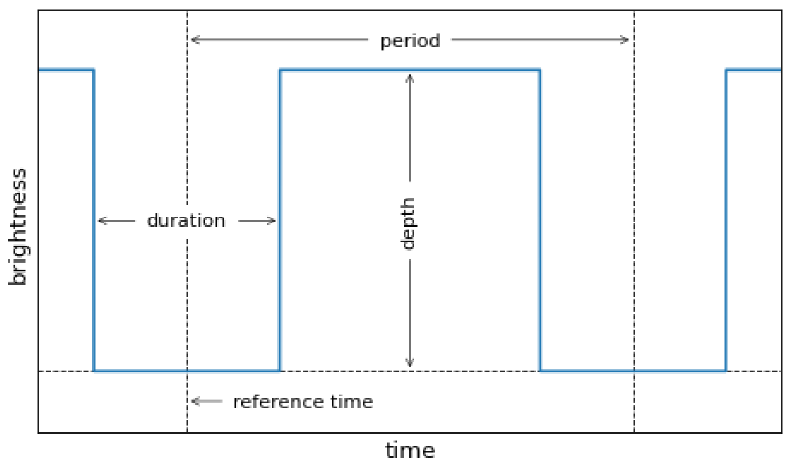

Figure 16: The four parameters used to make a Box least squares model .Image taken from the astropy developers.

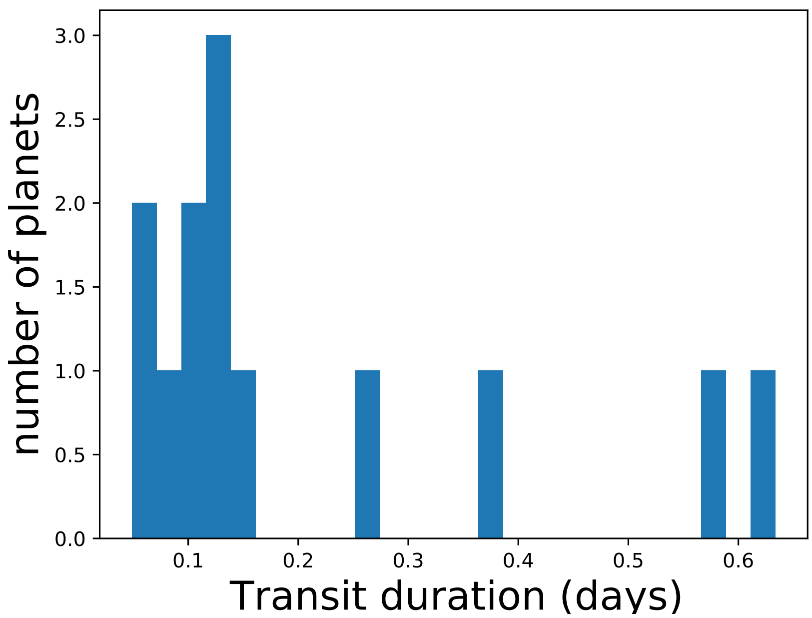

Figure 17: The transit duration for all of the transiting planets in a globular cluster with 20 000 stars.

A periodogram that took a time series observation with brightness measurements y and uncertainties dy at times t as input was used to determine the orbital period. For every possible orbital period put into the program, it returns a value of the log-likelihood of that certain orbital period being the transiting planet's actual orbital period. This is shown in figure 18 and 19. The maximum value would then be the assumed orbital period for the transiting planet.

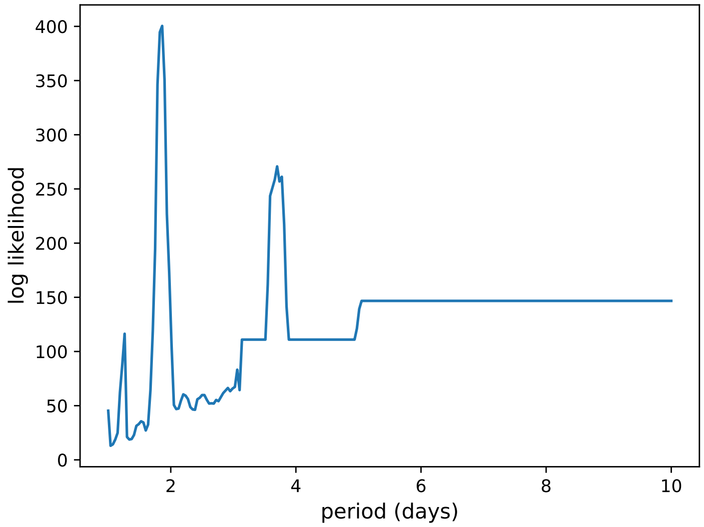

Figure 18: The log likelihood plotted against the potential orbital period from the lightcurve seen in figure 15.

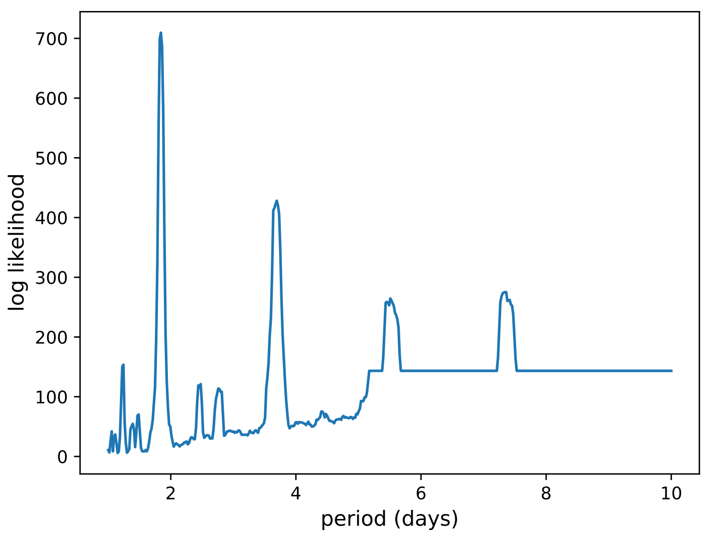

Figure 19: The log likelihood plotted against the potential orbital period from the same star but now for an observation period of 10 days.

Any model with a maximum value for the likeliest orbital period higher than $n_o \cdot 0.05$ was taken as a detected exoplanet, while any between $n_0 \cdot 0.05$ and $n_o \cdot 0.04$ was taken as a potential exoplanet. In these formulas, n_o refers to the number of observations. This was done since the maximum value depends on how long the observation was done, so a higher number of observations give a higher maximum value, as can be seen in figure 19. The number 0.05 was chosen to be able to detect the more clear cases of transits in light curves. While a lower number would have been able to detect more exoplanets it would also have given more false positives. A number much higher would however be unlikely to detect many exoplanets.

Phase folding can then be used to make it clear if there is an orbiting exoplanet or not. When phase folding, one takes the most likely period derived from BLS for the transiting planet and then splits the data into these periods and put them on top of each other, as can be seen in figure 20. The determined orbital period would sometimes be slightly off from the actual period which could sometimes lead to the phase folding being off, for example in figure **??**, there are brighter flux measurements over the transit caused by the fact that the orbital period determined by the box least squares algorithm is not the exact same as the actual orbital period.

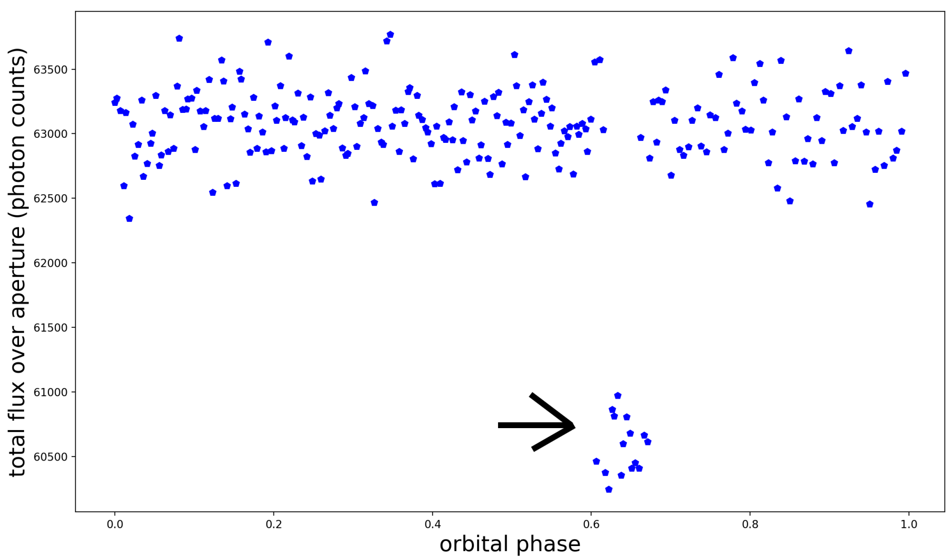

Figure 20: The phase folding of the transiting exoplanet from earlier figures, this was created from the light curve in figure 15.

4 Results and Discussion

First, a globular cluster with 20 000 stars was simulated, with simulated observations every half an hour for 5 days. A total of 10861 apertures could be created that satisfied the selection criteria. The created apertures can be seen in figure 21. In the cluster there was a total of 13 transiting exoplanets, of which six had apertures and four were detected. The parameters for the transiting planets and the corresponding stars can be seen in table 1. The light curves, periodograms, and phase folding for the four detected planets can be seen in figures 22-25. As can be seen in the phase folding of figure 22 the calculated orbital time was not always entirely correct leading to the phase folding being a bit off. This is however not a problem for detecting the planets as the transit is still noticeable.

Radial distance from center (light years)	Radius star (solar radii)	average flux of aperture(photon counts)	Radius planet (Jupiter radii)	Orbital period planet (days)	BLS Period (days)	Detected
127.8	0.50	14199	1.1	1.30	1.2625	Yes
25.2	1.50	235618	1.3	1.68	1.6749	Yes
62.5	0.8	44951	1.3	2.58	2.5	Yes
81.8	1.7	131703	1.4	2.65	2.6875	Yes
34.9	1.4	101724	1.5	5.9	N/A	No
120.5	1.6	164884	1.9	5.7	N/A	No

Table 1: Parameters for the systems with transiting planets for the cluster with 20 000 stars.

More than half of the planets could not be detected as the target stars did not meet the necessary criteria for the aperture. The other two undetected planets had too long orbital periods, planet 5 only transited once during the observations while planet 6 did not transit at all, so neither of them could be detected. The efficiency for finding transiting exoplanets in the cluster was therefore 31%.

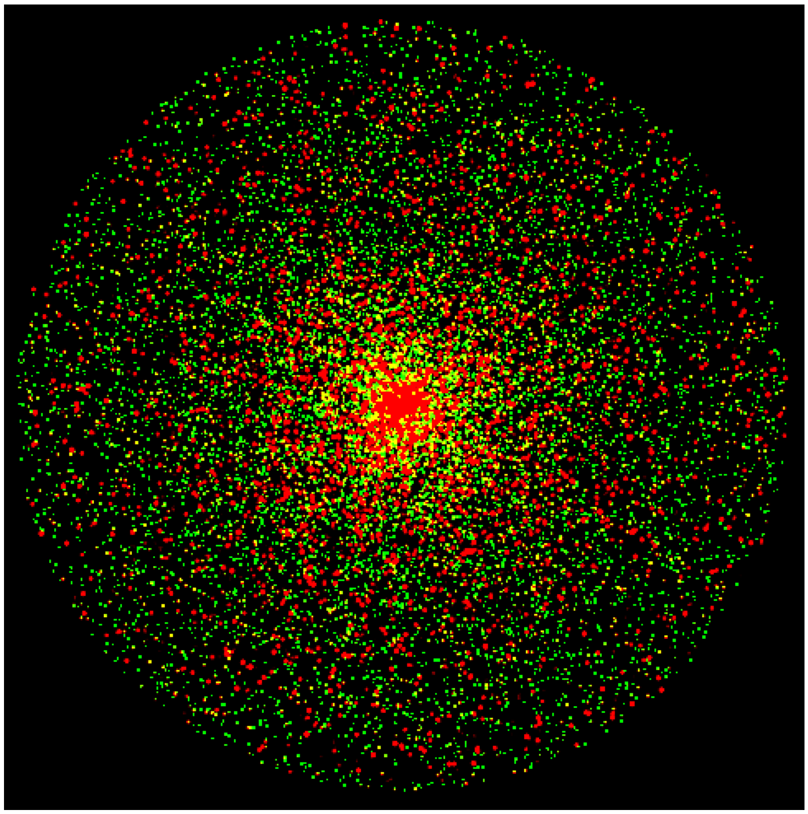

Figure 21: The apertures created for a globular cluster with 20 000 stars. Here the stars are red and the selected apertures are green. Stars without an aperture were disregarded due to our selection criteria. Not all stars selected are visible in the imagine, but if they are the aperture and star overlap to create a yellow colour instead.

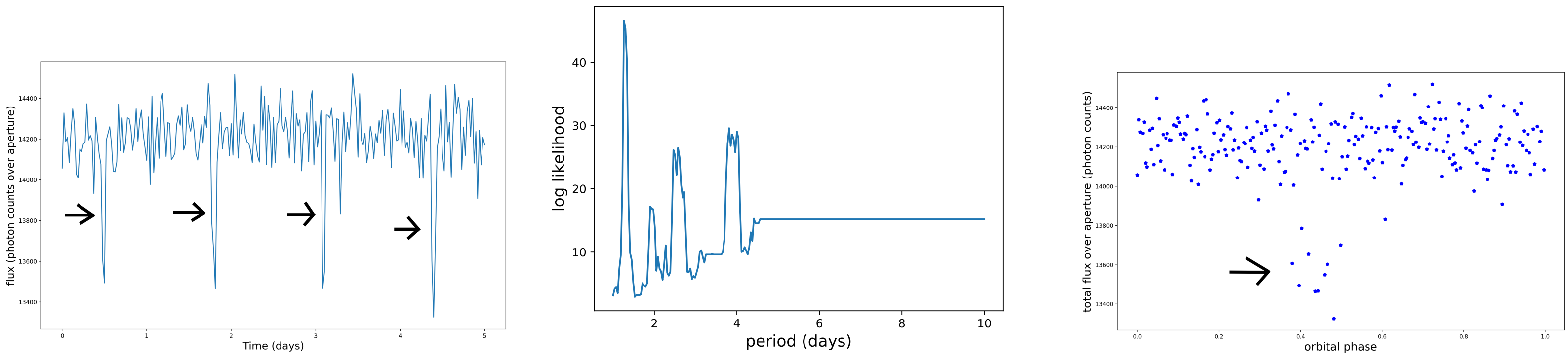

Figure 22: The lightcurve, periodogram, and phasefolding for star and planet 1

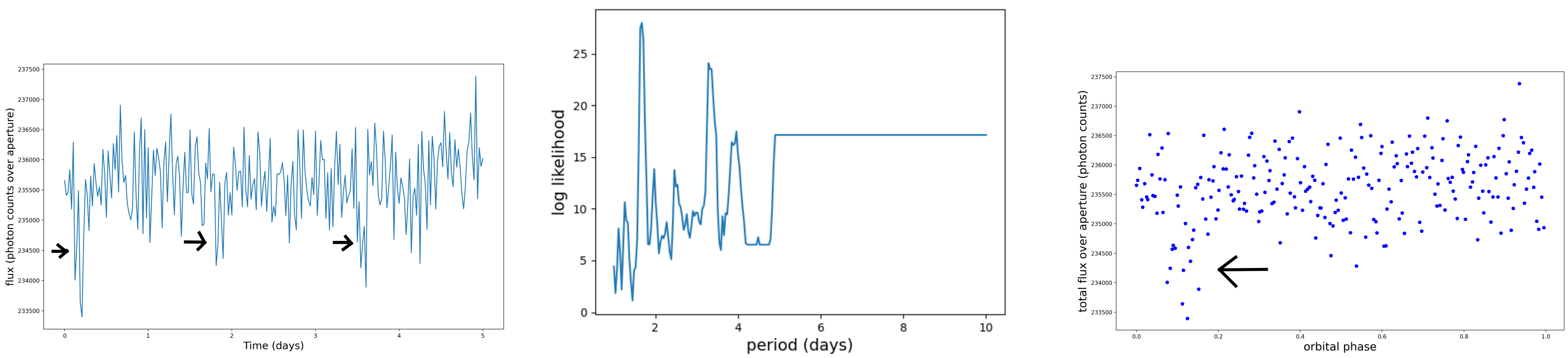

Figure 23: The lightcurve, periodogram, and phasefolding for star and planet 2

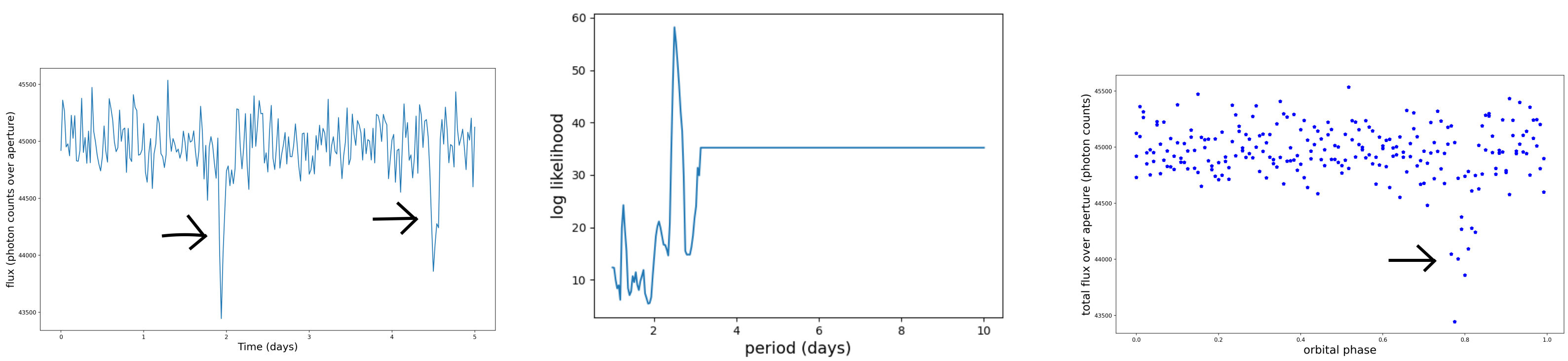

Figure 24: The lightcurve, periodogram, and phasefolding for star and planet 3

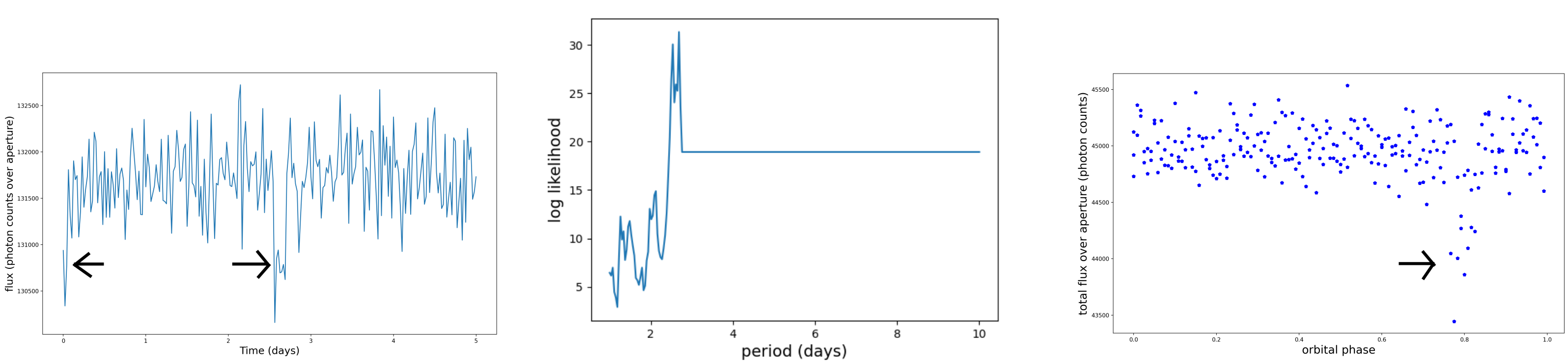

Figure 25: The lightcurve, periodogram, and phasefolding for star and planet 4

For a globular cluster with 50 000 stars, there were 29 transiting exoplanets, a total of 21356 apertures selected, the apertures selected can be seen in figure 27. The parameters for the transiting exoplanets that had stars with selected apertures can be seen in table 2. The fourth, seventh, eighth, ninth, and eleventh planets on table 2 were not detected as they only transited once during the observation. The fifth planet was not detected as the data was unclear. This could be since the planet was small and the star was rather large so it did not block out a significant portion of the star's light. The efficiency for finding transiting exoplanets in this cluster was therefore 17%.

Radial distance from center (light years)	Radius star (solar radii)	average flux of aperture(photon counts)	Radius planet (Jupiter radii)	Orbital period planet (days)	BLS Period (days)	Detected
67.2	0.37	50405	1.4	1.9	1.9	Yes
135	0.65	35427	1.3	3.0	2.9875	Yes
69.8	0.94	174968	1.3	2.6	2.6125	Yes
155.5	0.66	127258	1.1	5.3	N/A	No
68.8	1.7	191943	1.1	3.1	N/A	No
84.1	1.0	850205	1.4	3.1	3.175	Yes
91.2	0.90	54244	1.2	7.2	N/A	No
85.1	1.0	321646	1.1	5.1	N/A	No
61.8	2.4	268555	0.9	3.8	N/A	No
152.5	0.37	14744	1.7	3.3	3.3625	Yes
48.9	1.0	264466	2.0	4.6	N/A	No

Table 2: Parameters for the systems with transiting planets for the cluster with 50 000 stars.

The lightcurve, periodogram and phase folding for the second planet on table 2 can be seen in figures 28-30

Next, a globular cluster with 100 000 stars and 66 transiting planets was observed. This globular cluster was observed for a span of 5 days once every half an hour. The selected apertures can be seen in figure 31 and there was a total of 431 of them. Only one planet was inside the aperture and with an orbital period of 3.7 days, two transits during the observation time, and a radius 15 times larger than that of the earth it was clearly detected. The efficiency of finding exoplanets for these observations was 1.5 %. Considering both the small number of apertures and planets it was

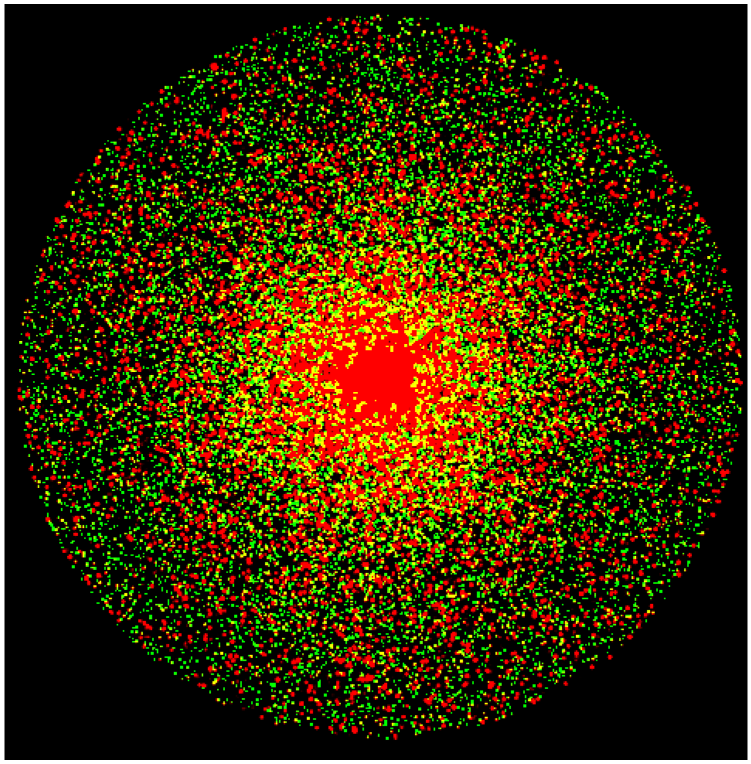

Figure 26: The apertures created for a globular cluster with 50 000 stars. Here the stars are red and the apertures are green.

possibly a lucky coincidence that one of them had a transiting exoplanet. So once again the most limiting factor for the efficiency was the low amount of apertures.

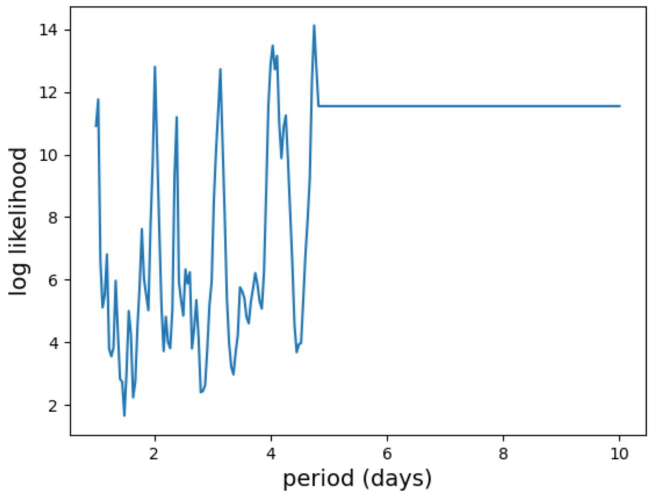

Figure 27: The periodogram for the fifth planet on table 2.

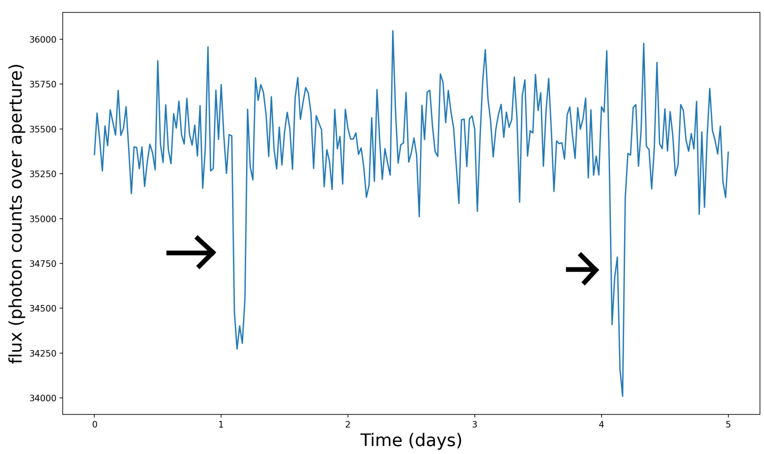

Figure 28: The light curve for the second planet on table 2.

Finally, a globular cluster with 150 000 stars and 90 transiting planets was simulated. 192 apertures were selected, they can be seen in Figure 32.

This globular cluster was observed in two different ways. First, it was observed during a span of 5 days with 30 minutes between every observation. The second time it was observed a total of five times, but after a day of observations, there would be one day without observations before observations would once again continue. In both of these cases, no exoplanets were found since none of the apertures had transiting exoplanets. Things are made difficult by the limited amount of apertures

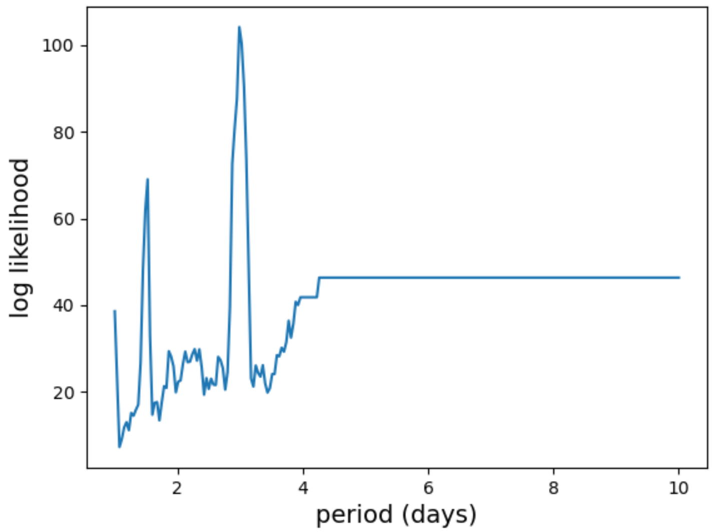

Figure 29: The periodogram for the second planet on table 2.

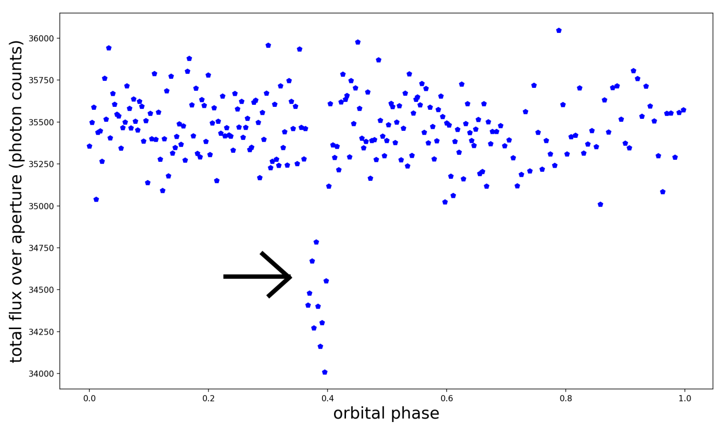

Figure 30: The phase folding for the second planet on table 2.

created which significantly decreases the likelihood of finding any transiting exoplanets. However, having more lenient criteria would give more false detection as stars close by each other would influence each other's flux, which would lead to false positives in the case of eclipsing binaries. This severely limits the possibilities of finding exoplanets in globular clusters. The relationship between the number of stars and apertures can be seen in figure 33

The efficiency could have been improved by putting the selection criteria on every pixel of the

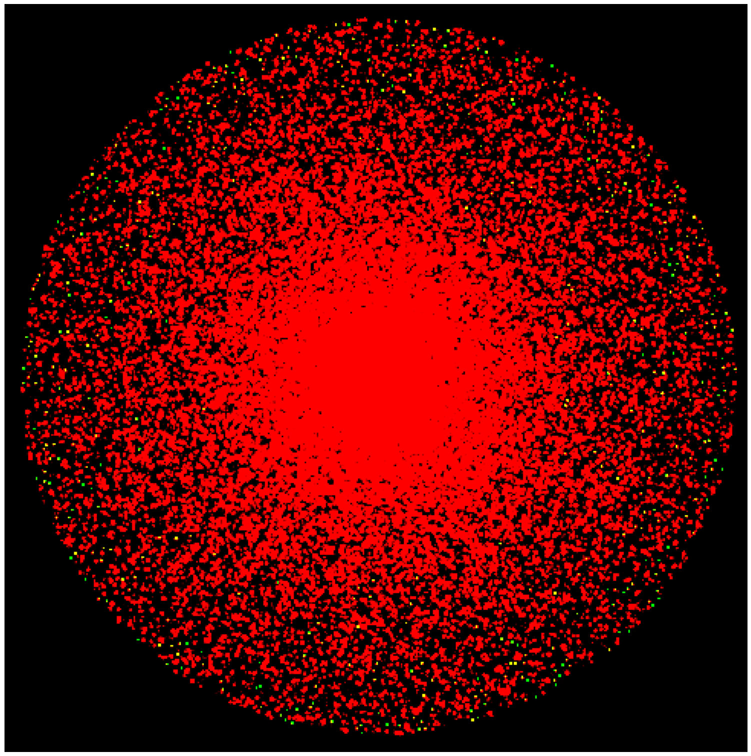

Figure 31: The apertures created for a globular cluster with 100 000 stars and a radius of 160 light years.

aperture instead of the entire aperture. For example, some pixels in the aperture might have been too close to a bright star while others were slightly further away and could have been used. Using a finer selection process would result in more apertures existing and therefore more exoplanets could have been detected. This procedure could then have been used to increase the efficiency. However, even with such a procedure the ability to find exoplanets would likely be limited due to crowding. Most of the stars were located in the center of the cluster where the detector was over-saturated and nothing could be observed. Since the planets were randomly distributed among the stars most of the planets would also have been at the center of the cluster. It is therefore uncertain how much the efficiency would have improved with these more specific criteria, it would most likely help increase the efficiency but only to an extent.

It would also be of interest to further investigate different parameters for the globular clusters to see how the efficiency is affected. Investigating several more globular clusters with different radii and amount of stars for example. This would give greater insight into what exactly the limits are for finding exoplanets in crowded environments. At around 20 000 stars the efficiency was relatively high as many apertures were created.

These results show it to be quite unlikely to find exoplanets in real globular clusters as there is

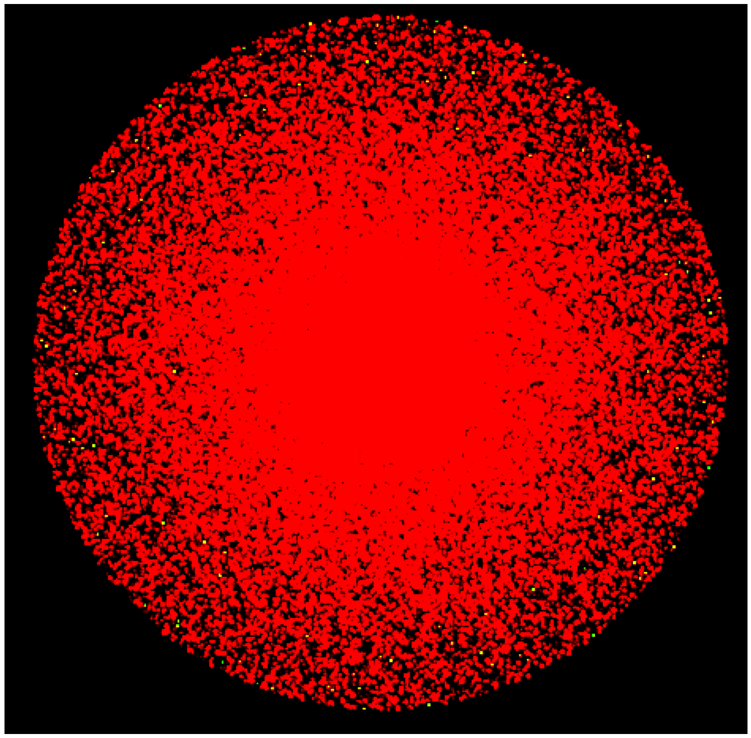

Figure 32: The apertures created for a globular cluster with 150 000 stars. Here the stars are red and the apertures are green.

usually somewhere between 10^5 and 10^6 stars. However, there are some assumptions underlying the simulations, which may influence the results.

The simulation created only dealt with main-sequence stars. This is of course unrealistic as there will also be stars outside of the main sequence in these environments. Stars with masses around 0.3 to 8 solar masses will evolve into red giants later in their life span (Laughlin, G. et al 1997). Some of the stars in this mass range might of instead have instead been red giants had the simulation been more realistic. Stars with masses 8-12 times that of the sun are so called Super-AGB stars (Eldridge, J.J. and Tout, C.A. 2004). Stars with higher masses than this are also expected to become giant stars. Therefore all of the stars in the simulation would be expected to eventually become giants yet none appear in the simulation. Adding giant stars would most likely make it even more difficult to find exoplanets. They have very large radii and any exoplanet transiting such stars would be very difficult to detect. Furthermore, such stars would be bright and contaminate other stars. To truly understand if it is feasible to find exoplanets in crowded environments they need to be simulated accurately, which means that more than main-sequence stars need to be included.

There is also the detector itself to consider. If the detector had been larger, for example 4000x4000 pixels, the stars would have been resolved better. If the stars had not been as smeared out they would have overlapped less and this could greatly help increase the efficiency. It would therefore be

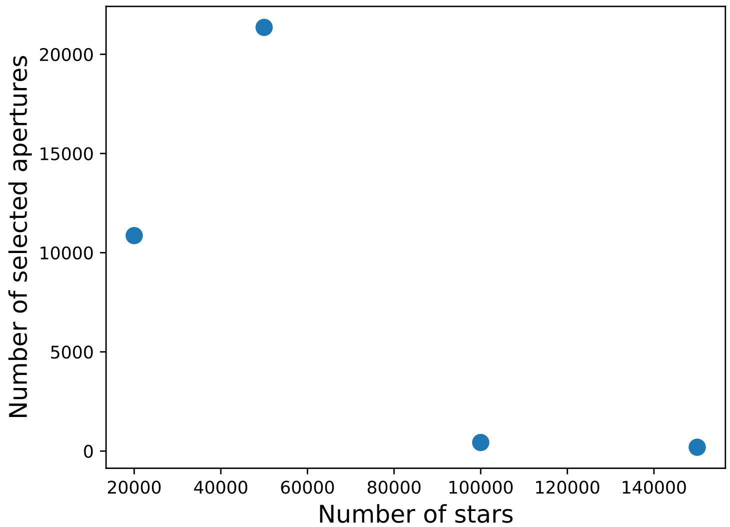

Figure 33: The apertures created compared to the number of stars.

worth investigating with a bigger detector and a smaller pixel scale. If one used a larger telescope it would also give rise to the stars being less smeared out as the Full width half maximum would be smaller and the stars would be easier to resolve.

5 Conclusion

In summary, this project investigated the possibilities of finding transiting exoplanets in crowded environments by simulating a globular cluster and orbiting exoplanets and then simulating observations of the cluster. While in smaller clusters it seems possible to find exoplanets in globular clusters with hundreds of thousands or even millions of stars it very quickly becomes extremely difficult to find any transiting exoplanets due to crowding. This implies that it does not necessarily mean that planets do not exist in these environments if they are not discovered but rather that they are just extremely difficult to detect. There were however some restrictions on the simulation that warrant further investigation, such as the fact that only main-sequence stars were used. The apertures could have also had less stringent criteria, or criteria applied to one pixel at the time instead of all of the pixels of the aperture. This could lead to more apertures existing in the globular clusters with more stars, which could have possibly increased the efficiency significantly as the small number of apertures was the main limiting factor for finding exoplanets in the globular clusters with more stars. There were also only 4 different globular clusters investigated, further investigation for clusters between 50 000 and 100 000 stars would be of interest to see where exactly the efficiency drops off. The results do however seem to indicate that finding transiting exoplanets in globular clusters is challenging but further investigation is warranted.

Bibliography

Chandar, R. et al, (2016) The age, mass and size distributions of star clusters in M51, *The Astrophysical Journal* 824:71 (12pp)

Cintio, P.D, Pasquato, M, Simon-Petit, A, Yoon, S, (1996) introducing a new multi-particle collision method for the evolution of dense stellar systems II:Core collapse, *Astronomy Astrophysics manuscript* no. equipartition June 17, 2021

de Juan Ovelar, M. et al (2018) Can habitable planets form in clustered environments? *Astronomy Astrophysics*, Volume 546

Eldridge, J.J., Tout, C.A (2004) Exploring the divisions and overlap between AGB and super-AGB stars and supernovae *Memorie della Società Astronomica Italiana* v75,694

Gratton, R. et al (2019) What is a Globular Cluster? *The Astronomy and Astrophysics Review* 27:8

Griffiths, D.J. (2012) *Introduction to Electrodynamics.* fourth edition. 1900 E. Lake Ave., Glenview, IL 60025: Pearson Education, Inc.

Haswell, C.A. (2010) *Transiting Exoplanets.* 1st Edition. Cambridge: Cambridge Univerity Press.

Heavens O.S. , and Ditchburn W., (1991) *Insight into Optics* Chichester :Longman and Sons,

King, I.R (1966) The Structure of Star Clusters. IV. Photoelectric Surface Photometry in Nine Globular Clusters *THE ASTRONOMICAL JOURNAL* VOLUME 71, NUMBER 4 pages 276-282

Kitchin, C.R (1987) *Stars, Nebulae and the Interstellar Medium: Observational Physics and Astrophysics.*Bristol: IOP Publishing

Kovacs, G., Zucker, S., Mazeh, T. (2002) A box-fitting algorithm in the search for periodic transits *Astronomy and Astrophysics* 391, 369–377 DOI: 10.1051/0004-6361:20020802

Gregory Laughlin et al (1997) The End of the Main Sequence *The Astrophysical Journal* , 482, 420-432

Lindsay B.R., (1966) Noise and Fluctuations: An Introduction *Physics Today* Oxford University Press Volume 16 Page 74

Meyers, R.A, (2001) *Encyclopedia of physical science and Technology.*Third edition. Elsevier Science Ltd

NASA Ames (2021) Accessed 12th May 2022 $< https://exoplanets.nasa.gov/resources/280/light-curve-of-a-planet-transiting-its-star/ >$

Petigura, E.A (2018) The California-Kepler Survey. IV. Metal-rich Stars Host a Greater Diversity of Planets *The Astronomical Journal* vol. 155, no. 89, pp 1-24.

Santerne, A. et al. (2013) SOPHIE velocimetry of Kepler transit candidates VII. A false-positive rate of 35% for Kepler close-in giant exoplanet candidates *Astronomy Astrophysics* manuscript no. KeplerFPP˙vfinal2

The Astropy Developers (2011) *Box Least Squares (BLS) Periodogram* Accessed 21 st April 2022 < *https://docs.astropy.org/en/stable/timeseries/bls.html#mathematical-background* >

P. Sookmee, et al Globular Cluster Classification in Galaxy M81 Using Machine Learning Techniques 2020 17th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2020, pp. 98-103, doi: 10.1109/JCSSE49651.2020.9268348.

Rolfs, C.E, Rodney, W.S (1988) *Cauldrons of the Cosmos: Nuclear Astrophysics* Chicago: University of Chicago press.

Schowengerdt, R.A. (2007) *Remote Sensing Models and Methods for Image Processing.* 3rd edition. Elsevier Inc.

Tayler, R.J. (1994) *The Stars: their structure and evolution.* 2nd Edition. Cambridge: Cambridge University press.

Wilson, P.A (2016) *The exoplanet transit method* Accessed May 2022

< *https://www.paulanthonywilson.com/exoplanets/exoplanet*

-detection-techniques/the-exoplanet-transit-method/ >

Appendix

Code

```
from PyAstronomy.pyasl import foldAt
from astropy.timeseries import BoxLeastSquares
import lightkurve as lk
import numpy as np
import matplotlib.pyplot as plt
from numpy import random
from matplotlib import cm
from astropy.modeling.models import BlackBody
from astropy import units as u
import astropy.io.fits as fits
import cv2 as cv
import scipy.integrate as integrate
#Amount of stars
np.random.seed(5)

n=80000 #amount of stars

#calculating the positions of the star
```

```

# the density distribution for a globular cluster
def distribution(r):
    rc=160
    f0=1
    return f0/(1+(r/rc)**2)



Rc=150 #radius of the globular cluster
RP=15 #segments

Area=np.zeros((RP))
result=np.zeros((RP,2))
result2=np.zeros((RP))
Positionr=[]
Positionx=[]
Positiony=[]
regions=np.linspace(0,160,15)

#the integral for every segment
for i in range(0,len(regions)-1):
    result[i]=integrate.quad(distribution,regions[i],regions[i+1])
#normalizing the value for every segment
for i in range(0,len(regions)-1):
    result2[i]=result[i][0]*1/np.sum(result)





indices = np.arange(0,len(regions))
chosen_bins = np.random.choice(indices,p=result2,size=n)
# giving stars positions
for i in range(n):
    Positionr.append(np.random.uniform(regions[chosen_bins[i]],regions[chosen_bins[i
    ]+1]))
Positionr2 = np.array(Positionr) #making it into an array
theta=np.random.uniform(0,360,size=n)  #giving the stars a random theta angle
for i in range(0,n):
    Positionx.append(Positionr2[i]*np.cos(theta[i])) #x positions
    Positiony.append(Positionr2[i]*np.sin(theta[i])) #y positions


PositionxFinal=np.array(Positionx) #making it into arrays
PositionyFinal=np.array(Positiony)
del result
del result2
del Positionr
del Positionr2
del theta
del Positionx
del Positiony
del chosen_bins
del indices

#masses for stars

```

```
masses=np.linspace(0.3,100,2000) #the regions for the masses for every star
result=np.zeros((len(masses),2))
result2=np.zeros((len(masses)))
starmasses=[]
m=1.9891*10**30
b2=-2
#dN/dM=M**b
def k(x):
    return x**b2
for i in range(0,len(masses)-1):
   result[i]=integrate.quad(k,masses[i],masses[i+1]) #integrals for every region
for i in range(0,len(masses)-1):
    result2[i]=result[i][0]*(1/np.sum(result))#These are relative probabilities for each of the mass bins.
indices = np.arange(0,len(masses))
chosen_massbins = np.random.choice(indices,p=result2,size=n)
for i in range(n):
    starmasses.append(m*np.random.uniform(masses[chosen_massbins[i]],masses[chosen_massbins[i]+1])) #giving stars masses within the regions
M = np.array(starmasses) #making it into an array
#Luminosity, approximation in main sequence
L=3.916*10**26*(M/m)**3.5

r=696340*10**3   #radius of sun

del result
del result2
del masses
del starmasses

#calculating radius for every star, approximation in main-sequence
R=np.zeros(n)
for i in range(0,n):
        R[i]=r*(M[i]/m)**0.9


#effective temperature

sigma=5.67*10**-8


T_eff=np.sqrt(np.sqrt(L/(4*np.pi*R**2*sigma)))





R2=3.086*10**16*3*10**3 #distance to cluster in meters



unit="erg / (cm2 Hz s sr)"

#Calculating flux for the stars
#flux for red
bb = BlackBody(temperature=T_eff*u.K) #the black body model for every star at its effective temperature
wav = 6250 * u.AA  #the wavelength the instrument observes at in   ngstrm
```

```
flux = (bb(wav)/unit).decompose().value #the flux value for every star

del T_eff
del bb
del L


#Exoplanets

earthradius=6371*10**3 #earth radius in meters

#Planet distributions
#Radius 12-16 earth radius
Planets1number=int((n/100)*0.02)#1-2 days orbital period
Planets2number=int((n/100)*0.08)#2-3 days orbital period
Planets3number=int((n/100)*0.21)#3-6 days orbital period
Planets4number=int((n/100)*0.08) #6-10 day orbital period
#Radius 8-12 earth radius
Planets5number=int((n/100)*0.05) #3-6 days orbital period
#16-24 earth radius
Planets6number=int((n/100)*0.08)#orbitalPeriod 3-6 days

NS=np.sum(Planets1number+Planets2number+Planets3number+Planets4number+Planets5number
    +Planets6number) #number of planets
#this is the number of exoplanets

NP=[i for i in range(0,NS)] #the indexes for stars with exoplanets



#Orbital period for the planets
OrbitalPeriod1=np.random.uniform(1,2,size=(Planets1number))
OrbitalPeriod2=np.random.uniform(2,3,size=(Planets2number))
OrbitalPeriod3=np.random.uniform(3,6,size=(Planets3number))
OrbitalPeriod4=np.random.uniform(6,10,size=(Planets4number))
OrbitalPeriod5=np.random.uniform(3,6,size=(Planets5number))
OrbitalPeriod6=np.random.uniform(3,6,size=(Planets6number))
OrbitalPeriod7=np.random.uniform(-1,0,size=(n-NS))
OrbitalPeriod=np.concatenate((OrbitalPeriod1,OrbitalPeriod2,OrbitalPeriod3,
    OrbitalPeriod4,OrbitalPeriod5,OrbitalPeriod6,OrbitalPeriod7),axis=0)

#The radii for the planets
Radius1=np.random.uniform(12,16,size=Planets1number)*earthradius
Radius2=np.random.uniform(12,16,size=Planets2number)*earthradius
Radius3=np.random.uniform(12,16,size=Planets3number)*earthradius
Radius4=np.random.uniform(12,16,size=Planets4number)*earthradius
Radius5=np.random.uniform(8,12,size=Planets5number)*earthradius
Radius6=np.random.uniform(16,24,size=Planets6number)*earthradius
Radius7=np.random.uniform(-1,0,size=n-NS)*earthradius
RE=np.concatenate((Radius1,Radius2,Radius3,Radius4,Radius5,Radius6,Radius7),axis=0)

b=np.random.random(n) #impact parameter

G=6.67408*10**-11 #gravitational constant

#Calculations for exoplanets
LB=[]
SMA=np.zeros((n))
```

```
Tr=[]
Inclination=[]
L=[]
alpha=[]
alpha3=[]


G=6.67408*10**-11


for i in range(0,n):
 if i in NP:
    LB.append((RE[i]**2)/(R[i]**2)) #light blocked by planet
    SMA[i]=((G*M[i]*((OrbitalPeriod[i]*60*60*24)/(np.pi*2))**2)**(1/3)) #semi major axis in meters
 else:
     LB.append(0)
     SMA[i]=0

del M
#Geometric transit probability

GTP=np.zeros((n))

for i in range(0,n):
    if i in NP:
        GTP[i]=(R[i]+RE[i])/SMA[i]
    else:
        GTP[i]=0


NPP=[] #the indexes for the planets transiting from our perspective

for i in range(0,n):
    if random.random()> (1-GTP[i]):
        NPP.append(i)




for i in range(0,n):
 if i in NPP:
    Inclination.append(np.arccos((R[i]*b[i])/SMA[i])) #inclination for planets in radians,
    L.append((((RE[i]+R[i])**2-((SMA[i]*np.cos(Inclination[i]))**2))**(1/2)))#distance travelled by the planets in meter
    alpha.append((np.arcsin(L[i]/SMA[i]))) #Alpha max angle radians
    alpha3.append(alpha[i]/(2*np.pi))#Alpha max angle orbital phase
 else:
     Inclination.append(0)
     L.append(0)
     alpha.append(0)
     alpha3.append(0)
```

```
T=np.linspace(0,5,241) #Time axis in days
K=np.random.uniform(0,1,size=n) #orbital phase starting value for exoplanets

alpha2=np.zeros((n,len(T)))


for i in range(0,n):
  alpha2[i]=(((T/OrbitalPeriod[i])-K[i])%1) #calculating the orbital phase for the
    planet at every observation


del K
#Calculating were transits occur
lightcurves=np.ones((n,len(T)))
for i in range(n):
    lightcurves[i]=flux[i]
    if i in NPP:
       lightcurves[i][(alpha2[i]<alpha3[i])+(alpha2[i]>1-alpha3[i])]*=(1-LB[i])


del alpha2
del alpha3





#Creating detector


w0=1.2 #The sigma value for the 2D Gaussian


N2=1000#Amount of pixels
AngularSize=np.arctan((300*9.46*10**15)/R2)*206264.806#this is angle in arcsecondss
FOV=2*3600 #Assumed field of view in arcseconds


ArcsecPerPixel=FOV/(N2) #The number of arcseconds per pixel


U4=AngularSize/300 # One Arcsecond is U4 light years


#Making the positions into pixel positions
xm=(PositionxFinal*U4)/ArcsecPerPixel
ym=(PositionyFinal*U4)/ArcsecPerPixel

del PositionxFinal
del PositionyFinal
del LB
del SMA
del Inclination
del Tr
del L

```

```
#Creating image

Xaxis = np.arange((-N2/2),N2/2)
Yaxis = np.arange((-N2/2),N2/2)
X, Y = np.meshgrid(Xaxis, Yaxis)

xi=np.arange(len(Xaxis))
yi=np.arange(len(Yaxis))




#creating data array image series




def f_gaussblur(x,y,x0,y0,z0,sigma):
    #This identifies the four pixels around where the star is located, linearly
    divides the flux
    #over these pixels, adds those to an empty image, and then performs gaussian
    blur to make them
    #into stars.
    x0_g = x0-np.min(x)#These are the x and y positions on the X,Y grid.
    y0_g = y0-np.min(y)
    x_left  = np.floor(x0_g).astype(int)#These are the indices of the pixels
    surrounding the star.
    x_right = np.ceil(x0_g).astype(int)
    y_below = np.floor(y0_g).astype(int)
    y_above = np.ceil(y0_g).astype(int)
    x_right[x_left==x_right]+=1
    y_above[y_above==y_below]+=1
    tx = (x0_g - x_left)/(x_right-x_left)
    ty = (y0_g - y_below)/(y_above-y_below) #This denominator is 1.0 per definition.
    z0_bottom = z0*(1-ty)
    z0_top = z0-z0_bottom
    #Fill in the four pixels surrounding the star:
    z_00 = z0_bottom*(1-tx)
    z_01 = z0_bottom-z_00
    z_10 = z0_top*(1-tx)
    z_11 = z0_top-z_10
    out = np.zeros(np.shape(x))
    for i in range(len(x0)):
        out[y_below[i],x_left[i]] += z_00[i]
        out[y_below[i],x_right[i]] += z_01[i]
        out[y_above[i],x_left[i]] += z_10[i]
        out[y_above[i],x_right[i]] += z_11[i]
    return cv.GaussianBlur(out,(25,25),sigma)



Z007=np.zeros((len(T),N2,N2),dtype=float)
for t in range(len(T)):
    Z007[t] = f_gaussblur(X,Y,xm,ym,lightcurves[:,t],w0)*9 #creating the images for
    every measurment, multiplying it by 9
#as the data is normalized and becomes smaller
```

```


#creating images

SR=2 #the standard deviation for the read out noise


counts=((Z007[:])/np.percentile(flux,90))*50000 #the flux of the stars in the image in photon counts
del Z007
noise=np.random.normal(100,SR,size=(len(T),N2,N2))+np.sqrt(abs(counts[:]))*np.random.normal(0,1,size=(len(T),N2,N2)) #the noise
error=SR+np.sqrt(abs(counts)) #the error of the measurment
data=counts+noise #the data measured
del counts

data[data>50000]=50000


del noise
#custom aperture

Z6=np.zeros((int(n),len(T)))

Z7=np.zeros((int(n),len(T)))
#Z7=[]
dy=1
dx=1

d=np.zeros((int(n),n))







cleancase=np.zeros((n,n))
contaminatedcase=np.zeros((n,n))



#calculating the distance between all the stars

for i in range(0,int(n)):
    d[i]=np.sqrt((xm-xm[i])**2+(ym-ym[i])**2)
    d[i][i]=d[i][i]+35

d_case1 = d<2.4 #all the times cases were there is less than 2.4 pixels between stars
d_case2 = (d>2.4)&(d<3.6) #all the cases stars are between 2.4 and 3.6 pixels between each other
d_case3= (d>3.6)&(d<6)#all the cases stars are between 3.6 and 6 pixels between each other

#calculating the indexes of the stars for all of these cases
index1 = []
index2=[]
```

```
index3=[]

for i in range(n):
    index1.append(np.arange(n)[d_case1[i]])
    index2.append(np.arange(n)[d_case2[i]])
    index3.append(np.arange(n)[d_case3[i]])



#creating the custom apertures


for i in range(0,n):
    print(i)
    yp=yi[Yaxis==int(ym[i])]
    xp=xi[Xaxis==int(xm[i])]
    #any star too bright or too dim will be rejected
    if np.amax(data[0][int(yp-dy):int(yp+dy+1),int(xp-dx):int(xp+dx+1)])<50000 and np.sum(data[0][int(yp-dy):int(yp+dy+1),int(xp-dx):int(xp+dx+1)])>2000:
      if min(d[i])>6: #if there are no other stars nearby it is accepted
         Z6[i]=np.sum(data[:,int(yp-dy):int(yp+dy+1),int(xp-dx):int(xp+dx+1)],axis=(1,2)) #
         Z7[i]=np.sqrt(np.sum(error[:,int(yp-dy):int(yp+dy+1),int(xp-dx):int(xp+dx+1)]**2,axis=(1,2)))

      else:

         if len(index1[i]>0): #the first case
             for k in index1[i]:
                 if flux[i]>10*flux[k]: #the star is accepted if it is significantly brighter than nearby stars
                     cleancase[i][k]=20

         elif len(index2[i])>0: #the second case
                 for k in index2[i]:
                   if flux[i]>5*flux[k]: #the star is accepted if it quite a bit brighter
                     cleancase[i][k]=20

                   else:
                     contaminatedcase[i][k]=20

         elif len(index3[i])>0: #the third case
             for k in index2[i]:
                if flux[i]>flux[k]*2: #the star is accepted if it is somewhat brighter.
                  cleancase[i][k]=20

                else:
                    contaminatedcase[i][k]=20


         if max(contaminatedcase[i])>0:
            Z6[i]=0
            Z7[i]=0

         else:
           Z6[i]=np.sum(data[:,int(yp-dy):int(yp+dy+1),int(xp-dx):int(xp+dx+1)],axis
```

```
    =(1,2)) #
           Z7[i]=np.sqrt(np.sum(error[:,int(yp-dy):int(yp+dy+1),int(xp-dx):int(xp+dx+1)]**2,axis=(1,2)))

    del d
    d=np.zeros((int(n),n))




#Box least squares


things=np.zeros((n))

period=np.zeros((n))
transit_time=np.zeros((n))
duration=np.zeros((n))
#things2=np.zeros((n))
#depth=np.zeros((n))
nottransits=[]
transits=[]
fluxy=np.zeros((n,len(T)))
phases=np.zeros((n,len(T)))

theIndex=[]

theIndexCandidate=[]
periodcandidate=np.zeros((n))
transittimecandidate=np.zeros((n))
durationcandidate=np.zeros((n))
phasescandidate=np.zeros((n,len(T)))
fluxycandidates=np.zeros((n,len(T)))
nottransitscandidate=[]
transitscandidate=[]





# Creating a box least square model for every star with an aperture
# and then saving the data


for i in range(213,214):
  if abs(max(Z6[i]))>0:
     periods = np.linspace(1, 10, len(T)) * u.day
     model=BoxLeastSquares(T*u.day,Z6[i],dy=Z7[i])
     periodogram = model.power(periods,0.2)

     max_power = np.argmax(periodogram.power)

     things[i]=(periodogram.power[max_power])#*1/sum(Z6[i])


     if things[i]>0.05*len(T):
         theIndex.append(i)
         period[i]=(periodogram.period[max_power])/u.day
```

```
        transit_time[i]=(periodogram.transit_time[max_power])/u.day
        duration[i]=(periodogram.duration[max_power])/u.day
        phases[i]=foldAt(T,periodogram.period[max_power],0)
        sortIndi = np.argsort(phases[i])
        phases[i] = (phases[i][sortIndi])#.decompose().value
        fluxy[i]= Z6[i][sortIndi]
        if i not in NPP:
              nottransits.append(i)
        if i in NPP:
               transits.append(i)
    if 0.04*len(T)<things[i]<0.05*len(T):
        theIndexCandidate.append(i)
        periodcandidate[i]=(periodogram.period[max_power])/u.day
        transittimecandidate[i]=(periodogram.transit_time[max_power])/u.day
        durationcandidate[i]=(periodogram.duration[max_power])/u.day
        phasescandidate[i]=foldAt(T,periodogram.period[max_power],0)
        sortIndi = np.argsort(phasescandidate[i])
        phasescandidate[i] = (phasescandidate[i][sortIndi])#.decompose().value
        fluxycandidates[i]= Z6[i][sortIndi]
        if i not in NPP:
              nottransitscandidate.append(i)
        if i in NPP:
               transitscandidate.append(i)
    del model
    del periodogram
    del max_power


efficiency=(len(transits))/len(NPP) #the efficiency of finding transiting exoplanets
```