

UNIVERSITY OF GRONINGEN

BACHELOR THESIS

Simulating the Visibilities of the Kapteyn Interferometer for Short-baseline Solar observations (KISS)

Author:

Jasper STERLINGA (s2889579)

Supervisors:

dr. John P. MCKEAN

Prof. dr. Andrey BARYSHEV

dr. Ronald HESPER

July 8, 2018



rijksuniversiteit
groningen

Abstract

This report describes the simulation of the observations of the KISS, a two-element radio interferometer built in Groningen. The KISS observes the size of the Sun from the interferometric visibility. As the observations depend on all structure pieces of the target source and the Sun's appearance is variable, it is proposed to use a number of models to simulate the possible observations. Using the models, the size of the Sun can be deduced from the observations. It is argued that smaller structures on the Sun's surface require a more advanced observational setup.

Acknowledgements

Firstly, I would like to sincerely thank my supervisor Dr. John McKean for his motivation and patience, for the time he put into the project and for sharing all his knowledge and insights during the project. His guidance gave me a better understanding of the subject interferometry, and also aroused me to think deeper into radio astronomy.

Secondly, I would like to express my appreciation to the other supervisors of the KISS project, Prof. Dr. Andrey Baryshev and Dr. Ronald Hesper for their time and guidance on the projects of my fellow students Casper Farret Jentink and Mathijn Lensen. They were very helpful and thoughtful in the process of constructing the interferometer and the receivers, and on doing observations.

I would also like thank my fellow students Mathijn Lensen and Casper Farret Jentink, who put all their time and effort in constructing the interferometer and performing the observations. We all shared our thoughts on each other's projects where needed/possible, and they were great partners to work with.

Furthermore, I want to thank the Kapteyn Astronomical Institute and SRON for providing a workspace for the construction and computer work of my partners and myself, and room in the storage to keep the components of the interferometer.

To conclude, I would like to thank three of my fellow astronomy students, Roi Kugel, Folkert Nobels and Ewoud Wempe for sharing some of their computational competence on parts of my project, and for keeping me company during the course of the project.

Contents

1	Introduction	3
2	Theory	4
2.1	Interference	4
2.2	Geometry	4
2.3	Correlation	5
2.4	Bandwidth effects	6
2.5	Visibility	6
2.6	Primary beam attenuation	7
2.7	Resolution	7
3	Setup, models and results	8
3.1	Interferometer setup	8
3.2	Coding	8
3.3	Models	8
3.3.1	Square	8
3.3.2	Disk	13
3.3.3	Donut	16
3.3.4	Gaussian function	17
3.3.5	Image with sunspots and solar flares	19
3.3.6	Hippo	21
4	Discussion	22
5	Conclusion and recommendation	25
5.1	Conclusions	25
5.2	Recommendations	26
A	Appendix	28
A.1	Derivations	28
A.1.1	Correlation	28
A.1.2	Bandwidth effect	28
A.1.3	Square Fourier transform	29
A.2	Python Code	29

1 Introduction

The principle of interferometry has been known to exist for several decades. In 1665, priest Francesco Maria Grimaldi performed an experiment where light could come into a dark room through two small holes, which as expected formed bright spots on the wall, but also formed dark regions between the spots. With the physical knowledge in the 17th century, Grimaldi could not explain what was going on. More than a decade later, in 1801, Thomas Young performed the same experiment in a more sophisticated way, creating a similar pattern on a background. This experiment is known today as "Young's interference experiment" [Hecht, Eugene 2014].

Interference has become a very fundamental field of science since Young's experiment, and many experiments have been performed using the principle of interferometry. One of these experiments was the Michelson-Morley experiment. In 1887 Albert A. Michelson and Edward W. Morley tried to show experimentally that the Earth and also light travel in a medium, at that time called an "ether". Their hypothesis was proven to be false by interferometrically measuring that light moves at the same velocity in all directions [Michelson and Morley 1887].

Interferometry has developed a lot since then, but the phenomenon itself still works by a very simple cause: path difference. Even though a wavefront of light emitted by a source has the same phase at the time of emission, it may be that the waves are out of phase at the point of observance. This happens if the waves have traveled a different distance from the source towards observance, and therefore arrive at different times. This so called "delay" is usually caused on purpose, by reflectance or by delayed observance.

Astronomical interferometry is usually done by delayed observance. An example of an astronomical interferometer is the Very Large Array (VLA) in New Mexico, USA. The VLA consists of 27 dishes of 25 meters in diameter, with radio frequency receivers. These dishes are positioned in a Y-shaped pattern and all point towards the same source simultaneously [NRAO 2017]. The difference in path to the different dishes causes the delay in the arrival of the radio waves. When adding the signals received in the different antennae, the resolution of the total array is much better than what it would be for one single dish, making it very useful in astronomy, since very small objects can be observed in detail.

The **Kapteyn Interferometer for Short-baseline Solar observations** (KISS) is a similar interferometer on a much smaller scale. The KISS was assembled in 2018 at the University of Groningen, and consists of two television satellite dishes of 35 cm in diameter each. The dishes operate at a frequency of roughly 11.2 GHz, or a wavelength of 2.7 cm, which is in the radio wave regime. The dishes are mounted on separate tripods, which makes them movable creating the possibility to vary its resolutional capabilities.

The goal of the KISS is to observe the angular size of the Sun by measuring the amplitude of the interferometric fringes as the Sun passes through the field of view. The Sun emits radiation across the complete electromagnetic spectrum, from X-rays to radio waves. Its peak emission is located in the visible part of the spectrum, more than 10 milliard times as much as in the radio regime [Russell, Randy 2007]. Fortunately, the television signal that the dishes and receivers of the KISS are designed for is in the same order of strength, and therefore the Sun will be observable at

radio frequencies.

In this report, the observations of the KISS are simulated. The observations will consist of interferometric fringes, and the amplitude of these fringes will depend on a number of factors, including the observed source, which is the Sun in this case. Various models of objects will be used to see how the amplitude of the fringes (and thus the observations) would be influenced by the structure of the source.

In Section 2 of the report, a theoretical background will be given on the phenomena that influence the observations, and these phenomena will be expressed mathematically. Then, in Section 3 the results of the coding for the various models will be explained, followed by Section 4, which discusses to which extent the simulations are useful and makes a comparison with observations that have been performed by the KISS already. Section 5 contains the conclusions that can be drawn from the simulations, and does a few recommendations on future simulations and observations.

2 Theory

2.1 Interference

The crucial physical principle that makes interferometry possible, is the principle of superposition. It follows from this principle that two waves, both having an amplitude and a period, can be added linearly to construct the combined wave. In popular language this is also known as interference. When two wave peaks meet, the adding of the amplitudes will lead to a peak with an amplitude equal to the sum of the two wave peaks it is composed from. This phenomenon is called constructive interference. The other extreme is when a peak of one wave and a trough of the other wave are added, leading to a smaller amplitude since the amplitudes have opposite signs. This is known as destructive interference.

2.2 Geometry

Measuring and adding electromagnetic radiation in at least two different positions leads to combined waves, as described in the previous paragraph. However, in the observation of the Sun, the amplitude of these combined waves will not be constant, but will change over time because the position of the Sun on the sky changes as time passes. This changes the angle under which the wavefronts arrive at the interferometer, causing a delay in the arrival of the wavefront between the two dishes. The consequence is that constructive interference happens when the vector Sun-interferometer is perpendicular to the baseline, but becomes less constructive as the sun moves away. Fig. 1 shows a sketch of the situation.

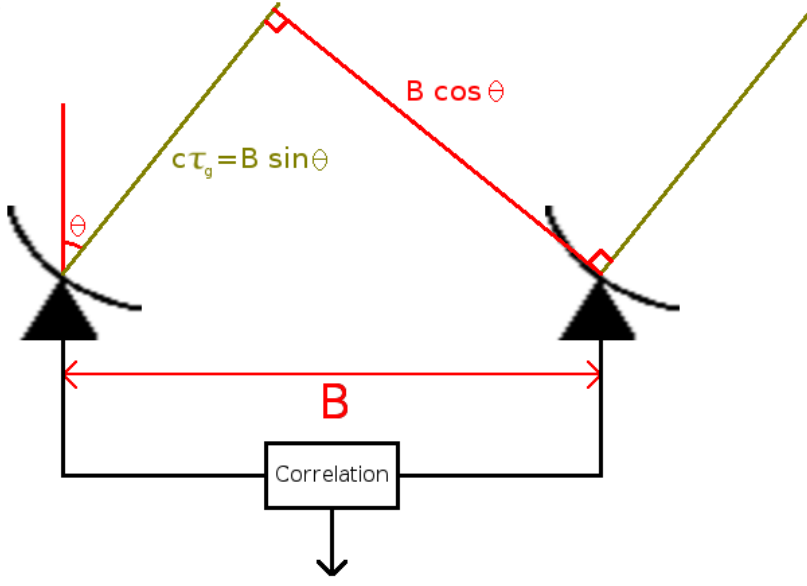


Figure 1: The geometry of an interferometer and the direction of the solar radiation.

From Figure 1 it can be seen that the delay between the two dishes can be expressed as a product of the speed of light and the delay time, and alternatively as a product of baseline length and angle. Rewriting the formula in Figure 1 we obtain

$$\tau_g = \frac{B}{c} \sin \theta, \quad (1)$$

where B is the baseline length between the two dishes in meters, τ_g is the delay time between the arrival of the wavefront for the respective dishes, θ is the angle between the azimuth and Sun's location, and c is the speed of light. This delay τ_g comes in when we look at the voltage of the separate dishes. Both dishes output a voltage that depends on the frequency of the incoming wavefront, and changes as a wave over time because the received radiation is a wave. The voltages output of the dishes can be written respectively as

$$V_1(t) = E \cos(2\pi\nu(t - \tau_g)) \quad \text{and} \quad V_2(t) = E \cos(2\pi\nu t), \quad (2)$$

where V_1 is the delayed voltage coming from the left dish in Figure 1, E is an amplitude factor depending on the strength of the sunlight, ν is the frequency of the light waves in Hz and τ_g is the delay time.

2.3 Correlation

The correlation of the signals is performed electronically in an adding fashion, instead of the more widely used multiplying fashion. The "adder", as can be seen in Fig. 1, squares the sum of the separate voltages. This gives a combined voltage of

$$V_{\text{comb}}(t) = (V_1(t) + V_2(t))^2 = V_1^2(t) + V_2^2(t) + 2V_1(t)V_2(t). \quad (3)$$

The correlated signal is then the average of this combined voltage

$$\langle V_{\text{comb}}(t) \rangle = E^2 [1 + \cos(2\pi\nu\tau_g)], \quad (4)$$

as derived in Appendix A.1. Then, using (1) and rewriting the frequency to cancel out the light speed, we obtain our fringe function

$$F(\theta) = E^2 \left[1 + \cos\left(\frac{2\pi B \sin \theta}{\lambda}\right) \right], \quad (5)$$

where λ is the wavelength of the incoming radiation.

2.4 Bandwidth effects

The fringes of (5) will always be attenuated by two effects. The first effect comes from the fact that the receiver does not receive light from one specific frequency, but from a range of frequencies around the mean frequency. To obtain the correct fringe function (which will change slightly under the influence of the bandwidth) we need to integrate and normalize the fringe function derived in (5) over the frequency range that is received by the telescope:

$$F(\theta) = E^2 \left[1 + \cos\left(\frac{2\pi B \sin \theta}{\lambda_0}\right) \frac{\sin\left(\pi \frac{B \sin \theta}{\lambda_0} \frac{\Delta\nu}{\nu_0}\right)}{\pi \frac{B \sin \theta}{\lambda_0} \frac{\Delta\nu}{\nu_0}} \right], \quad (6)$$

where ν_0 is the mean frequency, λ_0 is the wavelength corresponding to ν_0 and $\Delta\nu$ is the frequency bandwidth. The derivation can be found in Appendix A.1.

2.5 Visibility

The actual goal of the KISS interferometer, i.e. measuring the size of the sun, can not be achieved using the fringe spacings only, since they are only dependent on the geometry and operating frequency of the setup. The most important part of the project is what is called the visibility. The visibility is usually a complex quantity. Its absolute value is meaningful in interferometry, since it influences the amplitude of the fringes. The visibility can be expressed as a function of received power at the peak and the trough of a fringe:

$$|V| = \frac{P_{\text{max}} - P_{\text{min}}}{P_{\text{max}} + P_{\text{min}}}. \quad (7)$$

It can easily be seen that the highest value is reached when the power is zero in a trough, giving a visibility of 1, and the lowest value is reached when the power of a peak and a trough are equal, giving a visibility of 0. In layman's terms, this means that the visibility is 1 when the fringe amplitudes are maximal and 0 when the fringe amplitude is zero.

For small fields of view, the complex visibility of a certain target observing region is the two-dimensional Fourier transform of the sky brightness of the target region. This is stated by the Van Cittert-Zernike theorem, and can also be expressed as follows:

$$V(u, v) = \int T(x, y) e^{2\pi i(ux + vy)} dx dy, \quad (8)$$

where $V(u, v)$ is the complex visibility in Fourier space, and $T(x, y)$ is the sky brightness in real space [Cittert 1934] [Zernike 1938]. In essence, this Fourier transform gives the wavelengths and the amplitudes of the waves that can reconstruct the sky brightness when added together. This Fourier transform will have an amplitude of zero at the wavelengths that fit exactly in the size of the source, since the positive and negative wave will die out, giving no contribution.

This visibility will affect the fringes in the same way as the bandwidth, damping the amplitude of the fringes. The visibility is two-dimensional, but since KISS only has one baseline in more or less the east-west direction, we only need the visibility of all u -values for one specific v . The baseline length u is the projected baseline in wavelengths, therefore the final fringe function is

$$F(\theta) = E^2 \left[1 + \cos\left(\frac{2\pi B \sin \theta}{\lambda_0}\right) \frac{\sin\left(\pi \frac{B \sin \theta}{\lambda_0} \frac{\Delta \nu}{\nu_0}\right)}{\pi \frac{B \sin \theta}{\lambda_0} \frac{\Delta \nu}{\nu_0}} V\left(\frac{B \cos \theta}{\lambda_0}\right) \right]. \quad (9)$$

2.6 Primary beam attenuation

In the case of an observation with fixed dishes, there is an additional effect compared to the situation where the dishes track their source. This effect is the attenuation of the fringe amplitude, caused by the beam pattern of the dish. This beam pattern is the consequence of the fact that the response of the dish is not the same for signals coming from different angles. In reality the beam pattern is a Bessel-like function with one large peak and multiple smaller peaks at the wings. However, these so called "sidelobes" are very small, and therefore the beam pattern can be approximated by a Gaussian function with a full width half maximum (FWHM) of

$$\text{FWHM} \approx 1.22 \frac{\lambda}{D}, \quad (10)$$

where λ is the observed wavelength, D is the diameter of one dish and the FWHM is in radians [Wilson, Rohlfs, and Hüttemeister 2013]. For a Gaussian function, the FWHM and the standard deviation can be related by

$$\sigma = \frac{\text{FWHM}}{2\sqrt{2\ln(2)}} \quad (11)$$

where σ is the standard deviation. This σ can be inserted into the general definition of a Gaussian function to obtain the final beam attenuation

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right) \approx \frac{0.939}{\text{FWHM}} \exp\left(-2.773 \left(\frac{x - \mu}{\text{FWHM}}\right)^2\right), \quad (12)$$

where μ is the mean of the Gaussian, in this case the pointing direction of the dishes in radians, and x is the offset of the Sun's position on the sky w.r.t. the μ .

2.7 Resolution

In radio interferometry, the resolution of two synthesized apertures is a special case of Rayleigh's criterion, and can therefore usually be approximated as

$$\theta_{\text{res}} \approx \frac{\lambda}{B}, \quad (13)$$

where B is the baseline length. This can finally be used to calculate the angular size of the Sun, since the resolving baseline is known from the Fourier transform in (8).

3 Setup, models and results

3.1 Interferometer setup

From the physical dimensions of the interferometer, we can obtain a number of constants which are useful to simulate our fringes. First of all, the frequency that is observed (ν_0) is 11.2 GHz, with a bandwidth ($\Delta\nu$) of 40 MHz. From this it follows that $\lambda_0 = \frac{c}{\nu_0} \approx 2.68$ cm. The diameter of one dish is measured to be 35.2 cm, and it is more or less circular in the pointing direction[Lensen 2018]. Inserting these into (10) gives as a result that the field of view of the dish would have a theoretical FWHM of 0.093 rad or 5.31 degrees.

The baseline is restricted from below by the fact that the dishes can not overlap, and therefore can not be closer to each other than one dish diameter of 35.2 cm. The longest baseline is restricted by the length of the cables, giving a maximal baseline just above 20 m. [Farret Jentink 2018]

3.2 Coding

In order to simulate how the phenomena explained in Section 2 would influence the observations, a code in Python 3.5.2 was used to calculate the magnitude of all these phenomena. This code uses a model to calculate the visibility from a Fourier transform, fits a smooth function to a slice of that Fourier transform, and calculates the fringe pattern where all phenomena are taken into account.

3.3 Models

To simulate the observations of the interferometer we need to know what we will observe. However, the Sun's appearance changes all the time under the influence of various phenomena, including its varying distance, sunspots, solar flares and the transits of other astronomical bodies. A set of different model images was created, to see how the observations would be affected by these phenomena, and if they are relevant enough to take into account when drawing conclusions from the observational data.

All models underwent a Fourier transformation to generate the visibility of the object. This Fourier transform is not a smooth function, but consists of samples since the model images have a finite amount of pixels. However, it is more useful to have a smooth visibility function because it is easier to implement a smooth function in the final simulations of the fringes and the fitting of the observed data. This fit is done by `kmpfit`, a submodule of the module `Kapteyn`.

3.3.1 Square

The most simple representation of the Sun is a uniformly illuminated square with sizes equal to the diameter of the Sun. Actually this is a two-dimensional version of a so called "top-hat" function, a function often used in explaining the fundamentals of interferometry. Its corresponding visibility is easy to predict, since the function is mathematically well defined, and therefore the Fourier transform can be done analytically. This analytic solution can potentially also be used for fitting

the observational data. The model image and part of its Fourier transform can be seen in Figure 2.

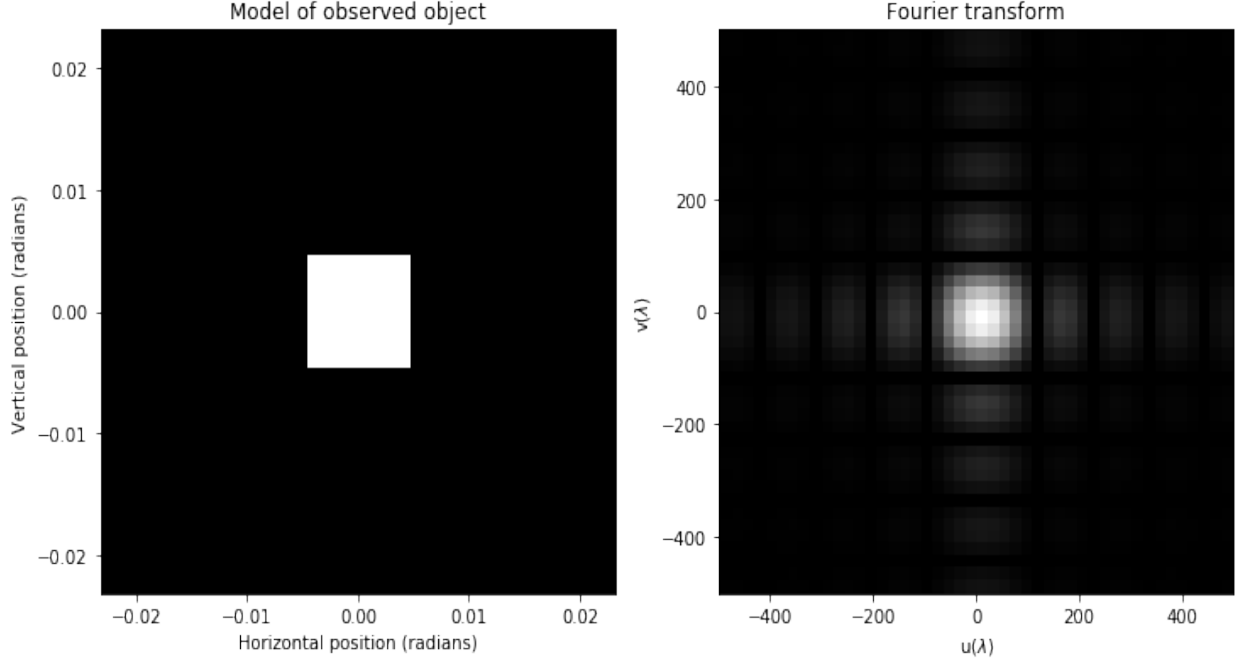


Figure 2: Left: An image of a square, uniformly illuminated Sun. Right: The Fourier transform of the image.

The square function with side a can be expressed analytically as follows:

$$T(x, y) = \begin{cases} 1, & \text{if } |x| \leq a/2 \text{ and } |y| \leq a/2 \\ 0, & \text{else} \end{cases} . \quad (14)$$

This function has a well defined Fourier transform, being

$$V(u, v) = a^2 \text{sinc}(\pi u a) \text{sinc}(\pi v a) \quad \text{where} \quad \text{sinc}(x) = \frac{\sin(x)}{x}, \quad (15)$$

which is derived in Appendix A.1. This Fourier transform is in uv -coordinates, where u is the east-west coordinate and v is north-south, and the distance from the center is the baseline length. Therefore, a horizontal slice through the middle of the Fourier transform will show the behavior of the visibility for different baseline lengths oriented in the east-west direction. A plot of this slice, and a fit of the analytic function from (15) can be seen in Figure 3.

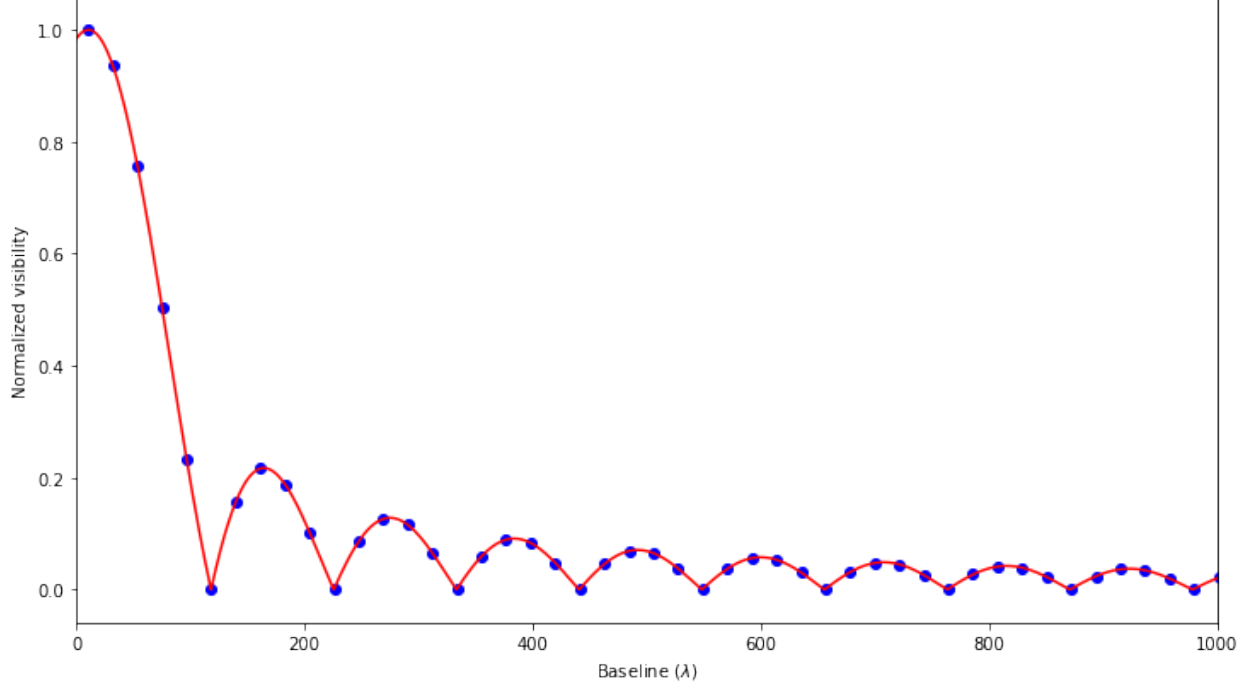


Figure 3: Slice of the Fourier transform of the square Sun, for $v = 0$ and positive u .

From Figure 3 a few characteristics can be seen. Firstly, since the image is real, the Fourier transform should be symmetric in the center. It can be seen that the function has a slight offset from the center, which is probably due to the numerical resolution of the Fourier transform. Secondly, the visibility is maximal at short baselines since the Sun is unresolved and therefore a point source, but then it drops steeply to zero at a baseline of 107.5λ , which is the shortest baseline where a square with the angular size of the Sun would be resolved. Thirdly, the visibility drops to zero periodically, which is explainable since at the zeros an integer number of pixels fits exactly into our created Sun, and this is also what you expect when looking at the behavior of a sinc-function. And lastly, the higher order peaks are smaller than the lower order peaks, which makes sense because at the higher order peaks larger parts are resolved out.

Now that we have a smooth visibility function, we can simulate what the observations will actually look like for both tracking dishes and fixed dishes. For a tracking observation, the fringes will be dampened purely by the bandwidth and by the visibility, as represented in (9). A plot of these fringes for a baseline B of 6 m can be seen in Figure 4.

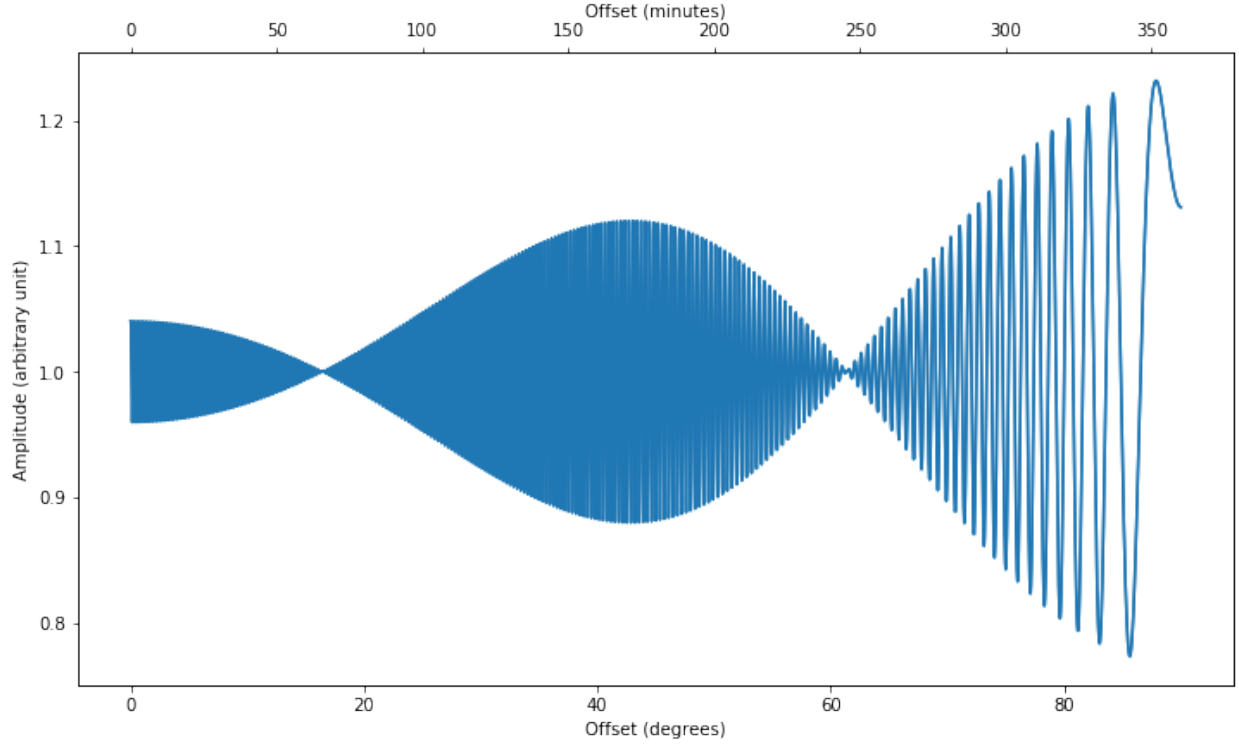


Figure 4: Simulation of a tracking observation of a square Sun for a baseline of 6 m. The observation starts when the Sun is at its highest point, which is at an offset of 0 from azimuth in u , and it ends when the Sun goes below the horizon, at an offset of 90 degrees.

Three things can be seen clearly from Figure 4: the fringes near the horizon are more widely spaced, the amplitude is larger near the horizon, and the amplitude goes to zero at an angle of about 15 degrees and near 60 degrees. The spacing is a direct consequence of the fact that the path difference changes less in time at observation angles. The amplitude is a consequence of the fact that the attenuation by both the bandwidth and the visibility are very small at short projected baselines. However, the amplitude dips are very interesting, since these are the angle for which the visibility goes to zero, giving us the projected resolving baselines and thereby the size of the Sun.

We can also plot the fringes for a fixed dish observation. In this case the attenuation from the dish comes into play. A series of such an observations can be seen in Figure 5.

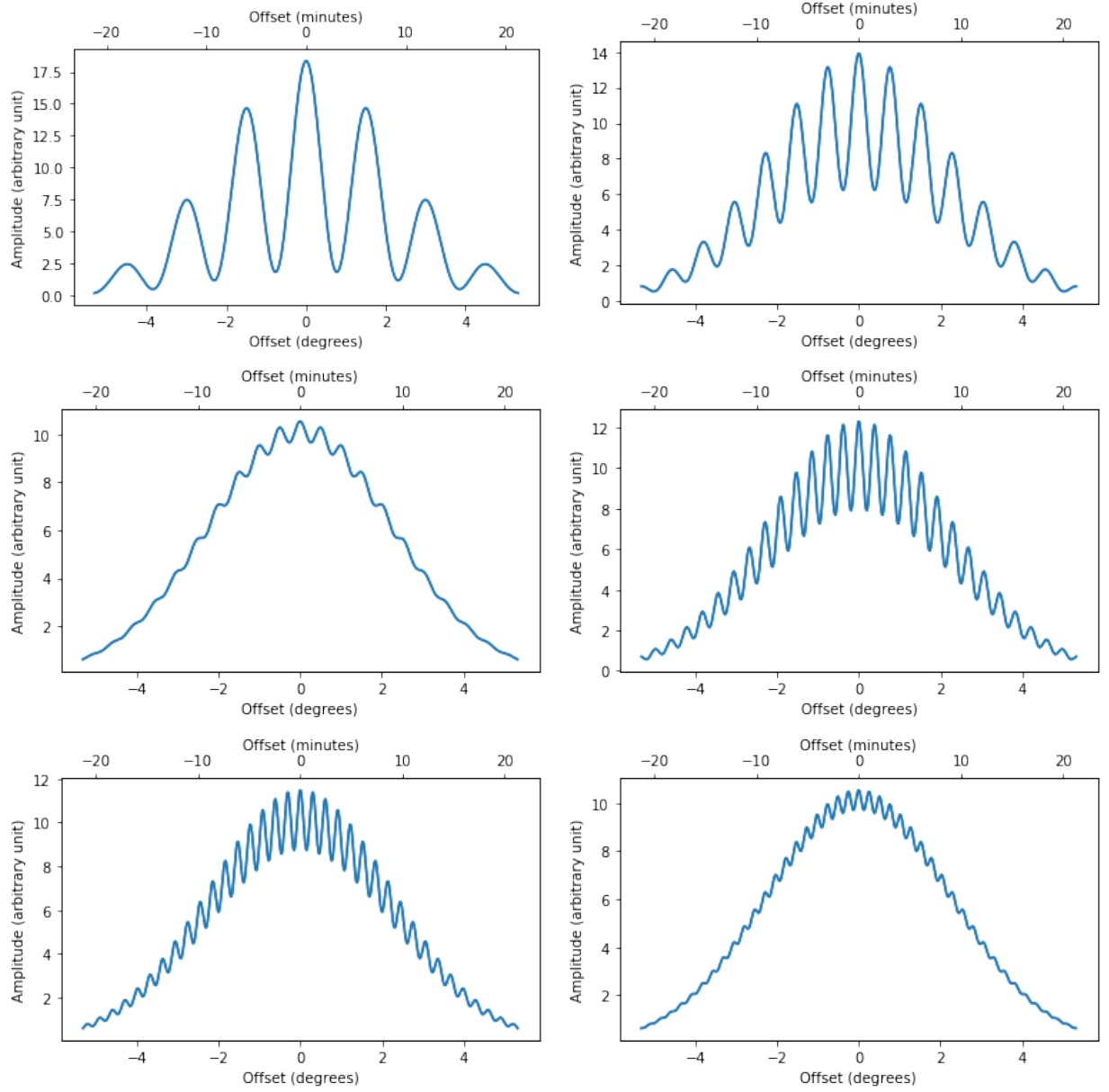


Figure 5: Simulations of fixed dish observations for various baselines. The observations are centered at an offset of 0 from azimuth in u , and contain 2 FWHM of the dishes. The baselines are integer values in meters, and when read from left to right increase from 1 to 6 m.

These observations in itself are not as useful as a tracking observation, since only one data point can be obtained per observation, i.e. the visibility at this specific baseline. This is done by calculating the visibility for P_{\max} being the highest peak and P_{\min} being the highest trough. Then, when measurements at multiple baseline lengths are performed, a numerical fit can be performed to obtain the complete visibility function, and thereby the resolving baseline length.

In Figure 5 the behavior of the visibility can already be seen: at very short baselines the am-

plitude of the fringes is very high, this drops to very small fringes near the resolving baseline, and then goes back up in the secondary peak. Furthermore the time between fringes decreases as the baseline becomes longer, which is as expected since the delay per angle change of the Sun's position is larger for a larger baseline.

Now we can check whether the model actually works and returns the proper size. The model seen in Figure 2 uses a square with sides equal to the diameter of the Sun, being 0.009298 radians (or 0.5327 degrees), which is the answer that should come out of the simulations and the observations. Numerically calculating the minimal value of the visibility between 0λ and 200λ gives a resolving baseline of 107.5λ , so the "observed" diameter of the Sun from the simulated square Sun can be calculated using (13) to be

$$\theta \approx \frac{\lambda}{107.5\lambda} = 0.009\,301\,\text{rad} = 0.5329\,\text{deg}, \quad (16)$$

meaning that the code calculates the visibility accurate to 0.04%.

3.3.2 Disk

The square model is obviously not the most accurate model, since in reality the Sun is observed to be spherical. A model that has a bit more physical meaning is a uniformly illuminated disk. Its Fourier transform is also well defined mathematically, so it can also easily be used to fit the actual observations. In Figure 6 the disk model and its Fourier transform can be seen.

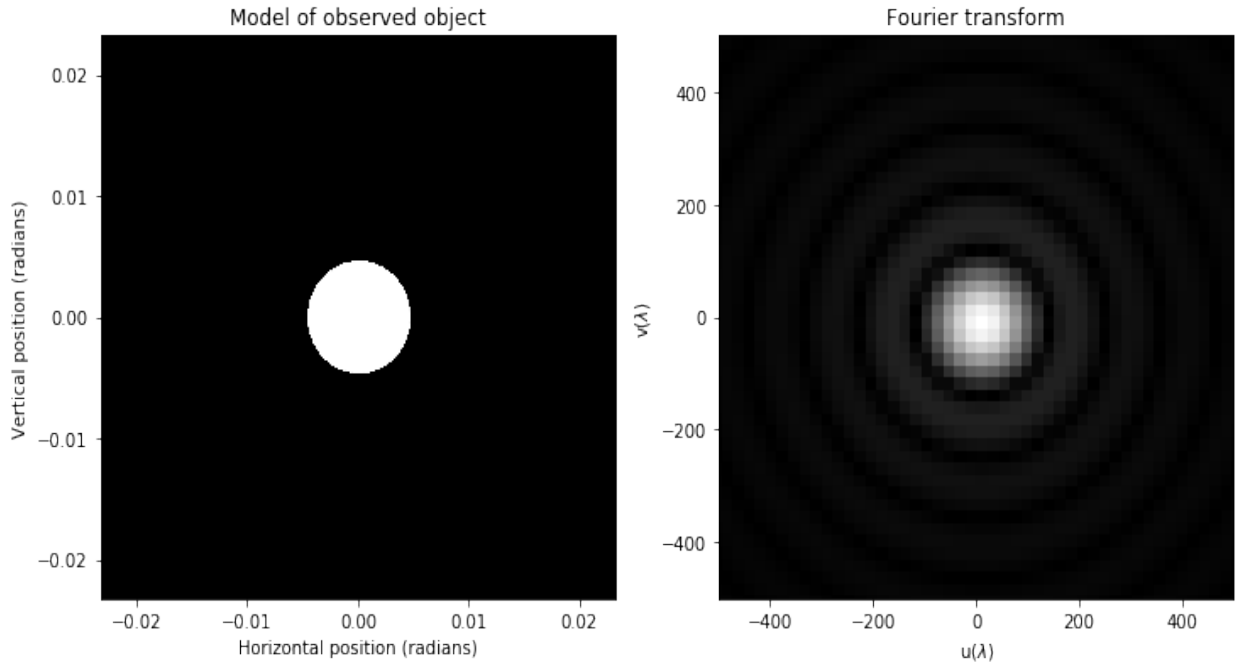


Figure 6: Left: An image of a uniformly illuminated disk. Right: The Fourier transform of the image.

Such an image or "function" can also be described mathematically in polar coordinates:

$$T(x, y) = \begin{cases} 1, & \text{if } |r| \leq a \\ 0, & \text{if } |r| \geq a \end{cases} . \quad (17)$$

Just like the square, this disk has a well defined Fourier transform, this being

$$V(u, v) = F(\rho) = \frac{aJ_1(\pi a \rho)}{\rho}, \quad (18)$$

where J_1 is a Bessel function of the first kind and zeroth order [Thompson, Moran, and Swenson 2017]. We can take a slice of this Fourier transform in the same way as for the square model, giving the visibility in u which can be seen in Figure 7.

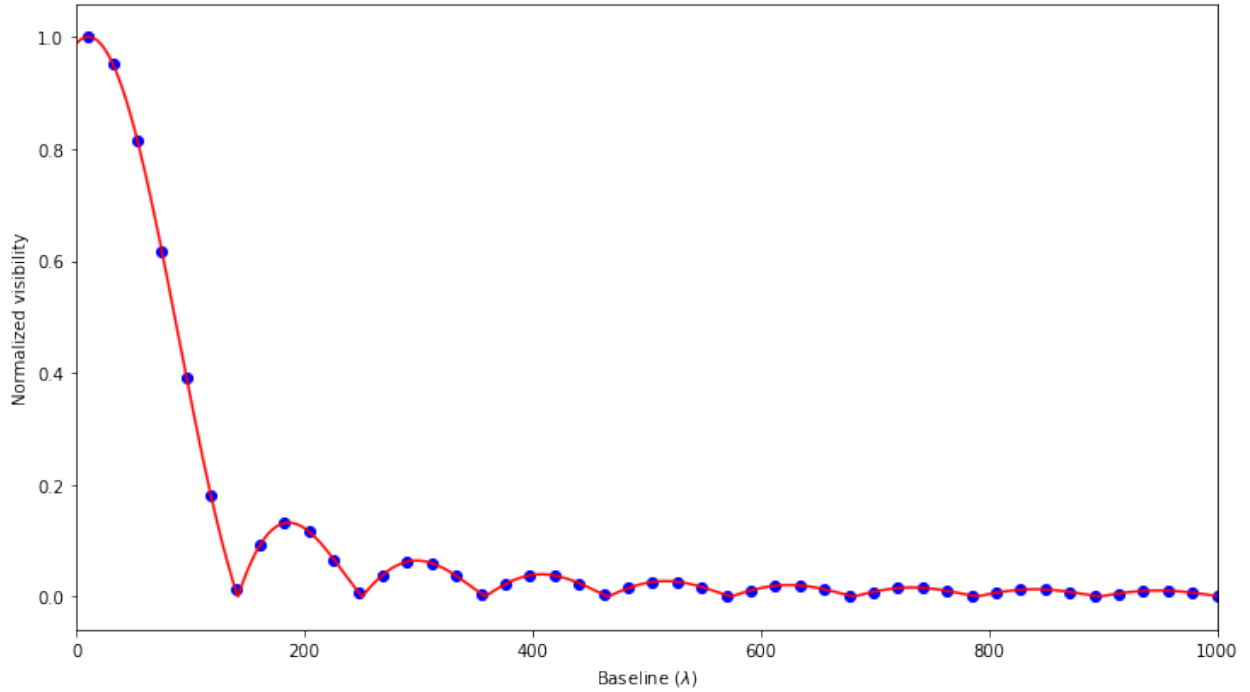


Figure 7: Slice of the Fourier transform of the disk Sun, for $v = 0$.

Two large differences can be seen between Figure 3 and Figure 7: the main peak is wider, and the secondary peaks are more dampened. The wider main peak is a consequence of the fact that a disk shape with a certain diameter has a smaller surface than a square with sides equal to this diameter. The dampening of the side peaks is a consequence of the fact that a square has a sharper cutoff than a circle.

Just like for the square Sun model, a simulation of the fringes for an observation of a disk shaped Sun can be performed. The simulation of a tracking observation can be seen in Figure 8.

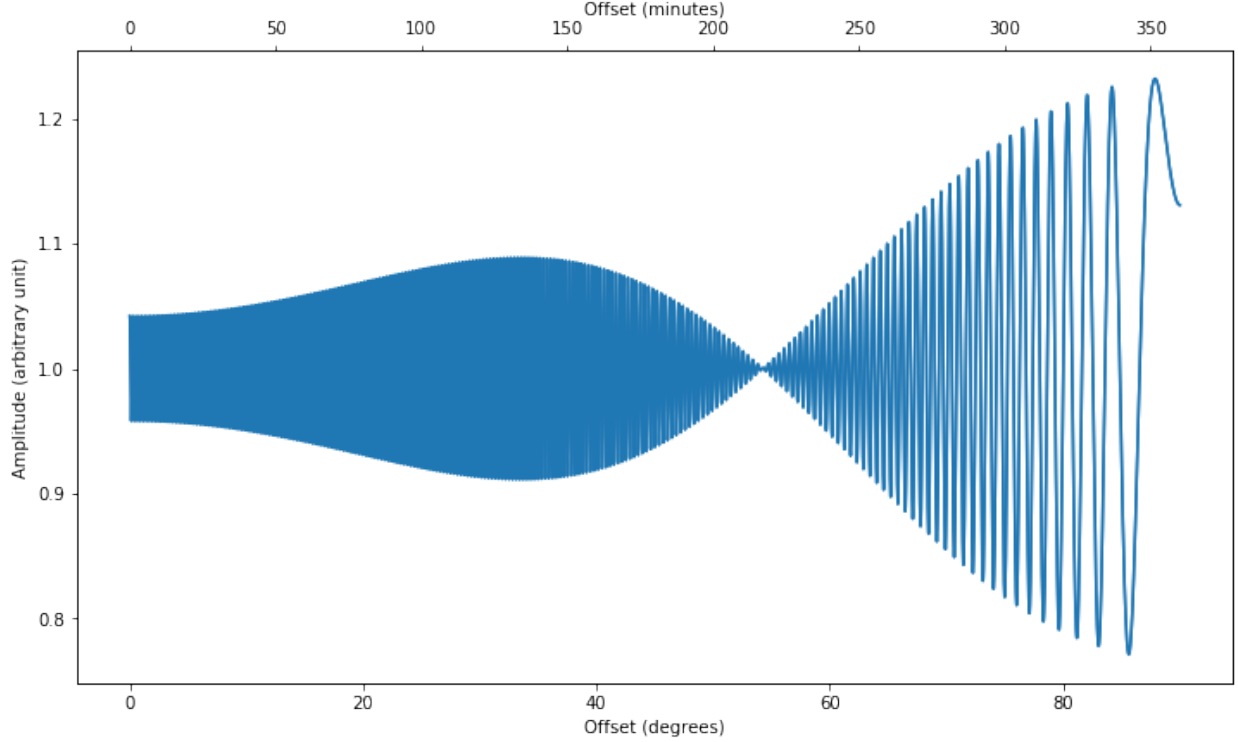


Figure 8: Simulation of a tracking observation of a disk shaped Sun for a baseline of 5 m with. The axes work like the axes in Figure 4.

This observation looks slightly different from the observation of a square Sun observation. We see that the amplitude of the fringes at small angles is smaller than in Figure 4, which is the direct effect of the secondary visibility peaks of the Bessel function being smaller than the those of the sinc function. But more important is the fact that the dip in the amplitude is at a smaller angle. This is a consequence of the fact that the visibility vanishes at a longer baseline than we had for the square Sun, which is reached at a smaller angle, if we have the same ground baseline length.

For this reason the calculation of the resolving resolution, and therefore the size of the Sun, will also be different. If we calculate the size of one pixel using (13) we obtain

$$\theta \approx \frac{\lambda}{131.1\lambda} = 0.007631 \text{ rad} = 0.4372 \text{ deg} \quad (19)$$

which is not equal to the value calculated for a square Sun in (16). This has to do with the zero locations of sinc and Bessel functions. The zero of the function $f(x) = \frac{\sin(\pi x)}{x}$ will have its first zero at $x = 1$, while the first zero of $f(x) = \frac{J_1(\pi x)}{x}$ is at $x \approx 1.22$ [Abramowitz and Stegun 1964]. It can also be visualized as follows: when you slice a disk in thin strips, the outer strips will be smaller than the inner strips. This makes that the interferometer essentially observes a Sun that has a smaller effective size, which in a Fourier transform turns into higher frequencies. Therefore the resolving baseline for a disk shaped Sun is larger than for a square shaped Sun by a factor 1.22.

3.3.3 Donut

The Sun is not necessarily uniformly luminous. In fact, the Sun is never uniformly luminous due to all sorts of effects like sunspots, solar flares, and limb darkening. Since these effects change over time, the Sun's appearance is not static, and therefore all observations of the interferometer will be slightly different. Furthermore, if the interferometer can observe other radio sources on the sky, those observations will be very different from Solar observations since astronomical objects can have all sorts of beautiful and funny shapes.

It would be impossible to simulate the observations for all these effects and shapes. A few interesting ones were picked to see how the visibility changes under the influence of different shapes and structures. The first of these, being a donut shape, does obviously not apply to the Sun, but might be applicable to, for example, the Ring Nebula. An image and the Fourier transform of a simplified donut shape can be seen in Figure 9.

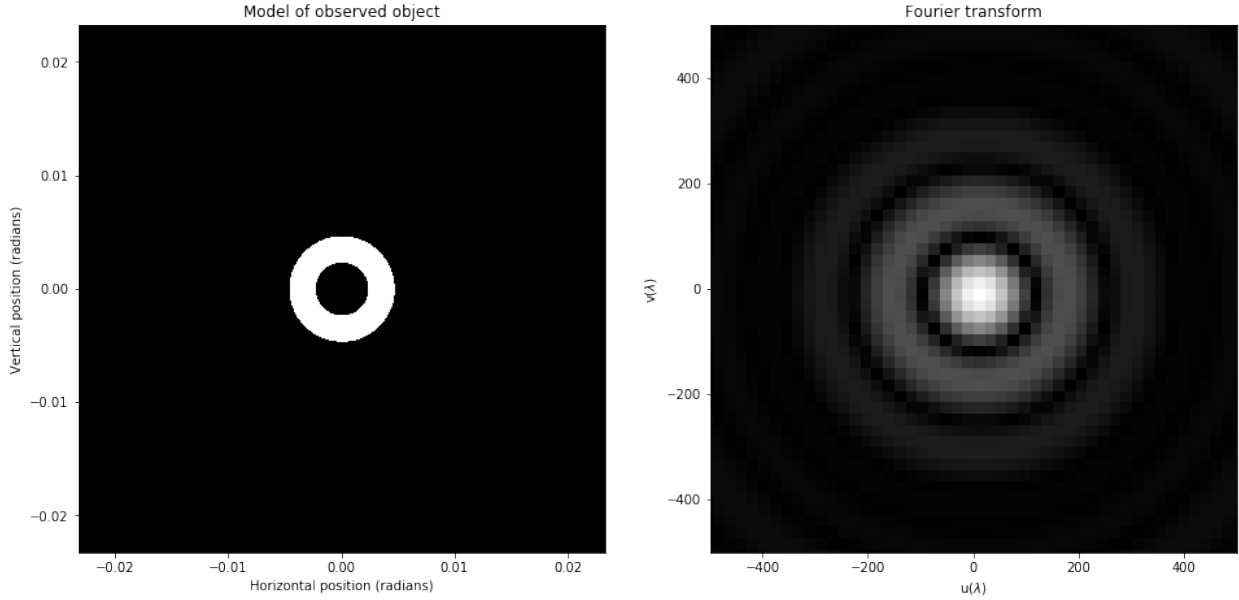


Figure 9: Left: An image of a uniformly illuminated donut. Right: The Fourier transform of the image.

The Fourier transform of the donut looks very similar to the one of the disk. This is explainable, since this donut is just a large uniformly illuminated disk with a smaller disk subtracted. Then, use of the fact that the integral of a sum is a sum of integrals, gives that the Fourier transform will be the difference between the Fourier transform of a large disk and a small disk. This results in a fit as seen in Figure 10.

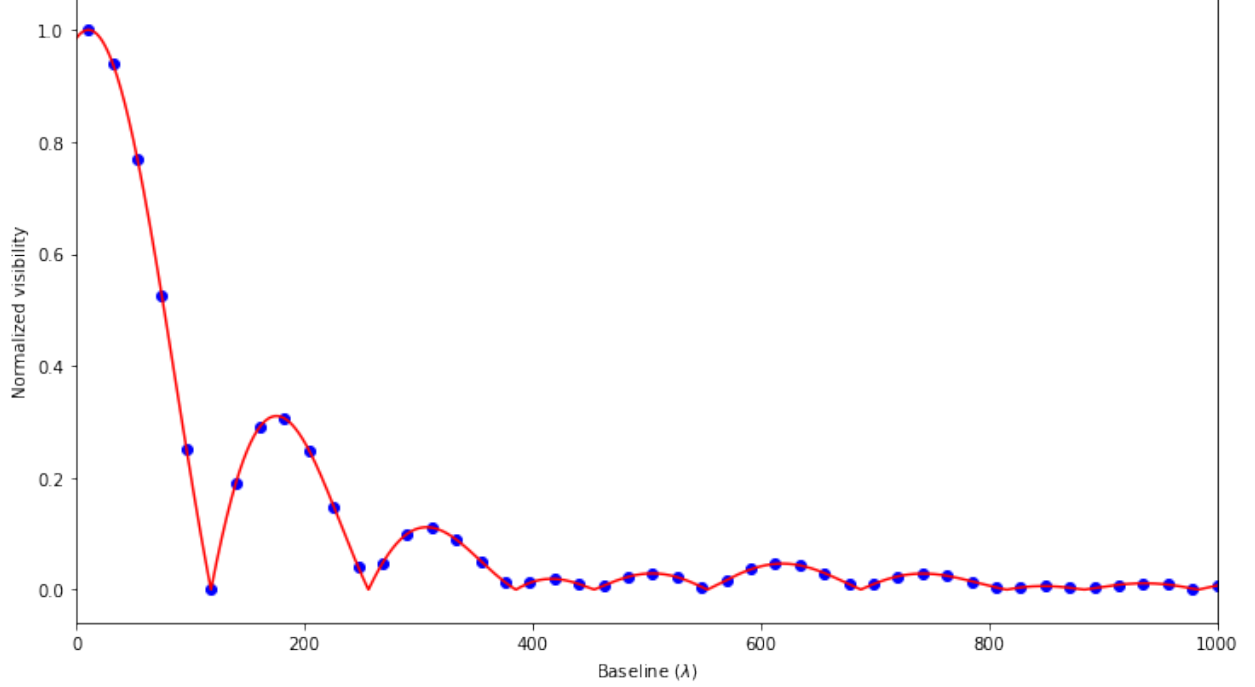


Figure 10: Slice of the Fourier transform of a donut.

A remarkable coincidence is that this visibility goes to zero at approximately the same baseline as for the square Sun. However, really interesting is the fact that the amplitude of the peaks at longer baselines decays differently. The secondary peaks are higher than those of the square and disk Sun, but the quaternary peaks are very small, and from the penitentiary peak the amplitude comes up again. This is the effect of the hole in the middle of the donut, which causes a subtraction of a Bessel function with wider peaks and less amplitude than the Bessel function of the larger circle. The fact that the peaks are wider causes that some peaks do not experience any damping (where the wider Bessel is zero) while other peaks experience much damping (where the wider Bessel has a peak).

3.3.4 Gaussian function

A Gaussian function has a very special property, namely that its Fourier transform is also a Gaussian function [Thompson, Moran, and Swenson 2017]. This is because its wings go to zero very smoothly, meaning no sharp cutoffs, and therefore no harmonics in the Fourier transform. A Gaussian illumination also applies to the Sun to some extent, since it is most luminous in the center, and smoothly becomes less luminous towards the edges. A model of a Gaussian illuminated Sun can be seen in Figure 11. Since there are no sharp edges, there is no resolving baseline either. However, that does not mean that we can say nothing about its size. A calculation of the FWHM of the Fourier transform, which can be seen in Figure 12, tells us that the visibility has dropped to 0.5 at a baseline of 86.37λ , or 2.31 m.

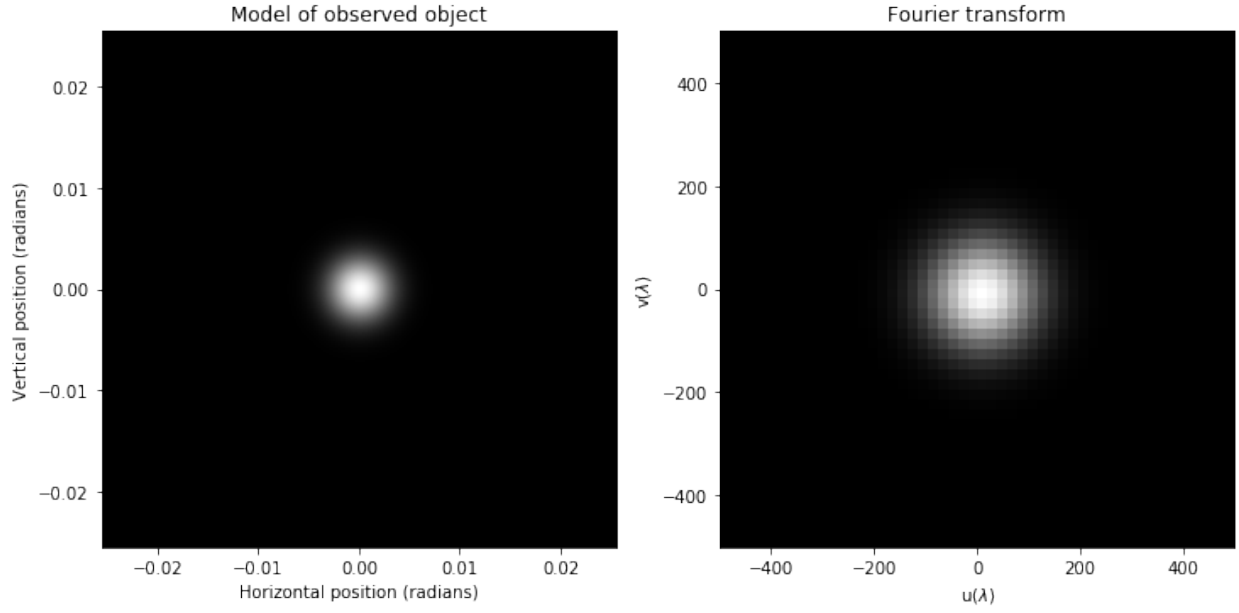


Figure 11: Left: An image of a Gaussian illuminated Sun, with a FWHM equal to the radius of the Sun. Right: The Fourier transform of the image.

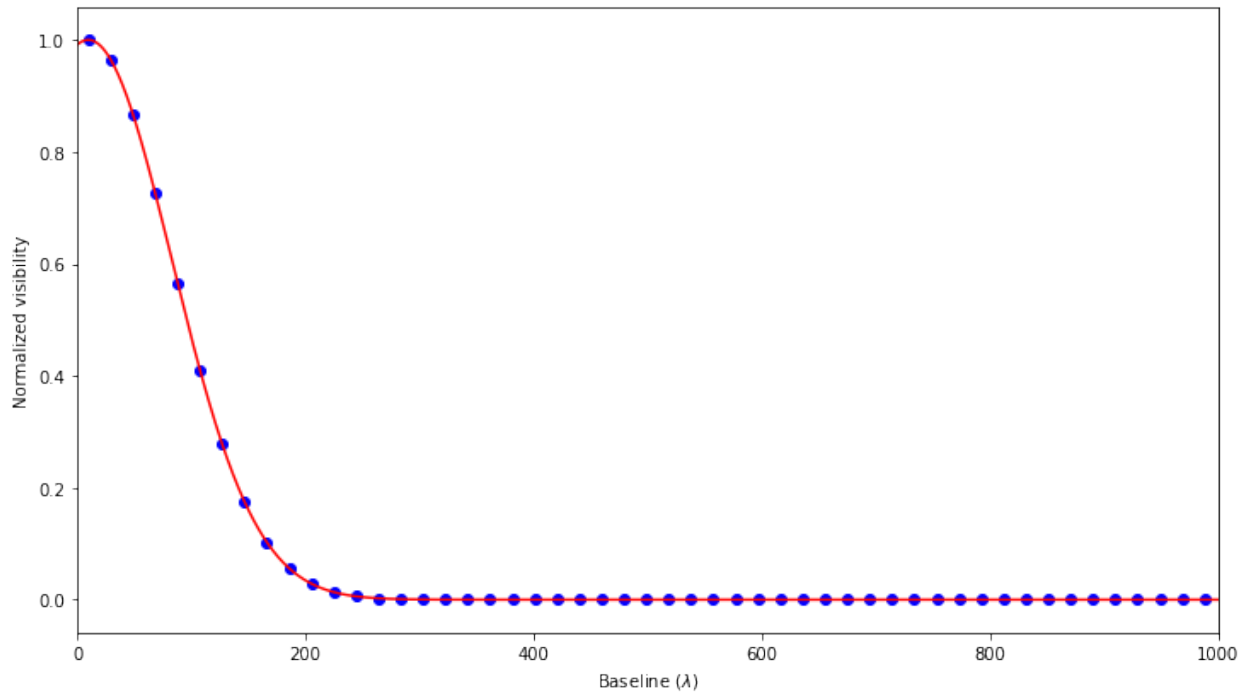


Figure 12: Slice of the Fourier transform of a Gaussian illumination.

3.3.5 Image with sunspots and solar flares

Two phenomena of great interest are sunspots and solar flares, even though their influence on the appearance of the Sun is hard to predict. But what we can do, is take a picture of the Sun, and see how solar flares and sunspots at specific locations affect the visibility. A picture of the Sun and its Fourier transform can be seen in Figure 13.

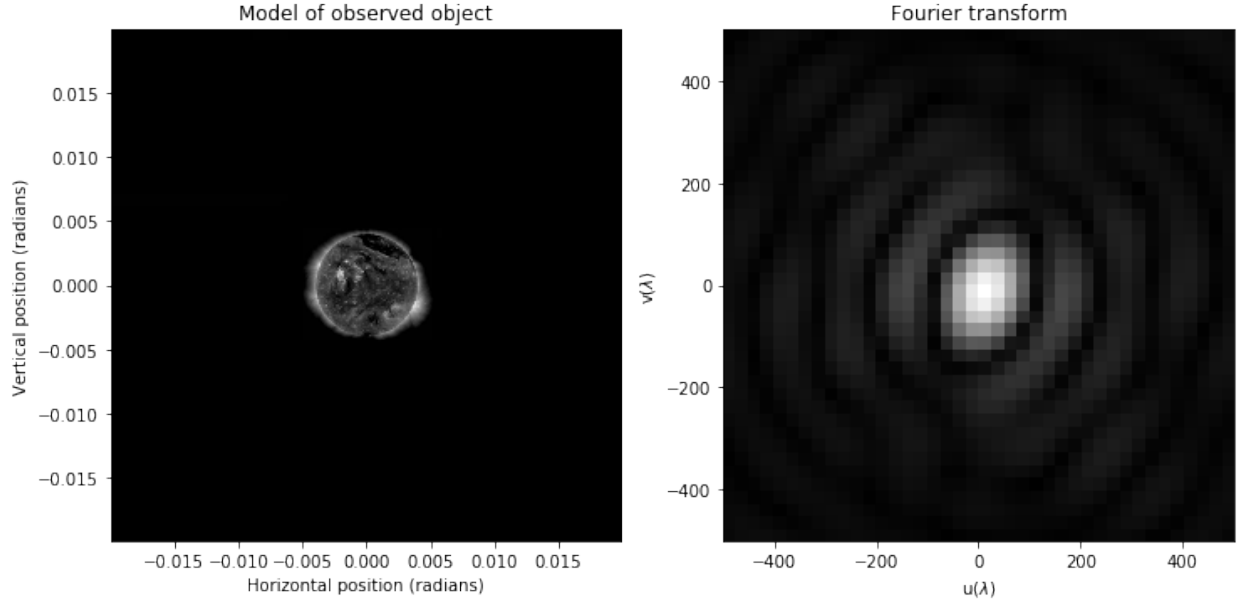


Figure 13: Left: An image of the Sun at radio wavelengths. Sunspots and small solar flares can be seen [NASA 2010]. Right: The Fourier transform of the image.

The effect of the solar flares can be seen very clearly: the most prominent radiation comes from the bottom right of the Sun, while the Fourier transform is stretched towards the top right and bottom left. Keeping in mind that this is in Fourier space, this makes perfect sense, since the long diagonal structure in the image will result in longer waves in the Fourier transform, meaning lower frequencies. The shorter structure in the other diagonal direction with no flares result in shorter waves, or higher frequencies, stretching the Fourier image.

The sunspots and flares that are not located near the outskirts of the surface are harder to distinguish, since they don't contribute in the largest resolvable structure, but instead form a piece of structure by themselves. Therefore, they affect the Fourier transform farther from the center. Furthermore, the bright structure elements are probably saturated on the image, making that they have a smaller impact on the image than they should, and therefore also on the Fourier transform. What is important to keep in mind when doing the observations is that the sunspots will not affect the observations at small baselines, which also makes sense since the sunspots are not resolved. In fact, to resolve a sunspot of one tenth of the Sun's diameter, a baseline of ten times the Sun's resolving baseline is needed, which is already past the limit of what the KISS interferometer can do in its current state.

Fitting this Fourier transform with an analytic function will give a fit that is far from perfect, since it is impossible to take every tiny structure element into account. However, the main structure of the image is still circular, so fitting a Bessel function would not be a bad estimate. This fit can be seen in Figure 14

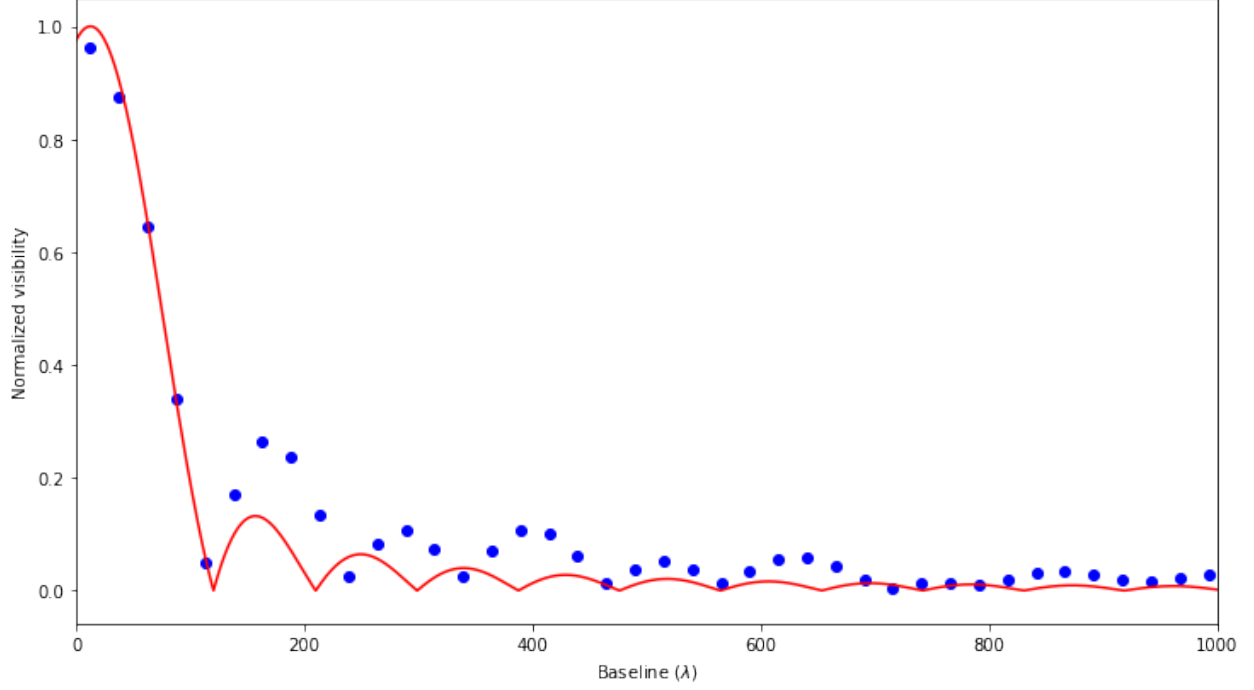


Figure 14: Slice of the Fourier transform of the image of the Sun, and an analytic fit of a Bessel function.

Keeping in mind that we work in Fourier space we can say that the smallest resolving baseline belongs to the largest structure, which in this case can be seen by the fact that the first resolving baseline is smaller, meaning that the largest structure is larger than the Sun’s diameter. Calculating the resolving baseline numerically like with the square and disk Sun, a value of 107.9λ is obtained. Using this baseline and the correction for the disk-like shape found in Section 3.3.2 we find that the angular size of the Sun is

$$\theta \approx 1.22 \frac{\lambda}{107.9\lambda} = 0.01131 \text{ rad} = 0.6479 \text{ deg}, \quad (20)$$

which is significantly larger than what is found in all literature about the angular size of the Sun. It becomes even worse when realizing that the primary peak is still a bit narrower than the fit, so that would mean that the diameter is still slightly larger. This tells us that solar flares that occur on the outskirts of the Sun will have a large impact on the measurements, since it will increase the size of the largest structure.

3.3.6 Hippo

As we said earlier, astronomy has a wide range of interesting objects which all have beautiful and peculiar shapes. We could try imagining that we look at an object in the shape of a hippo with the angular size of the Sun, to see what the visibility will look like for such a structure. An image and a Fourier transform of a hippo-shaped object can be seen in Figure 15.

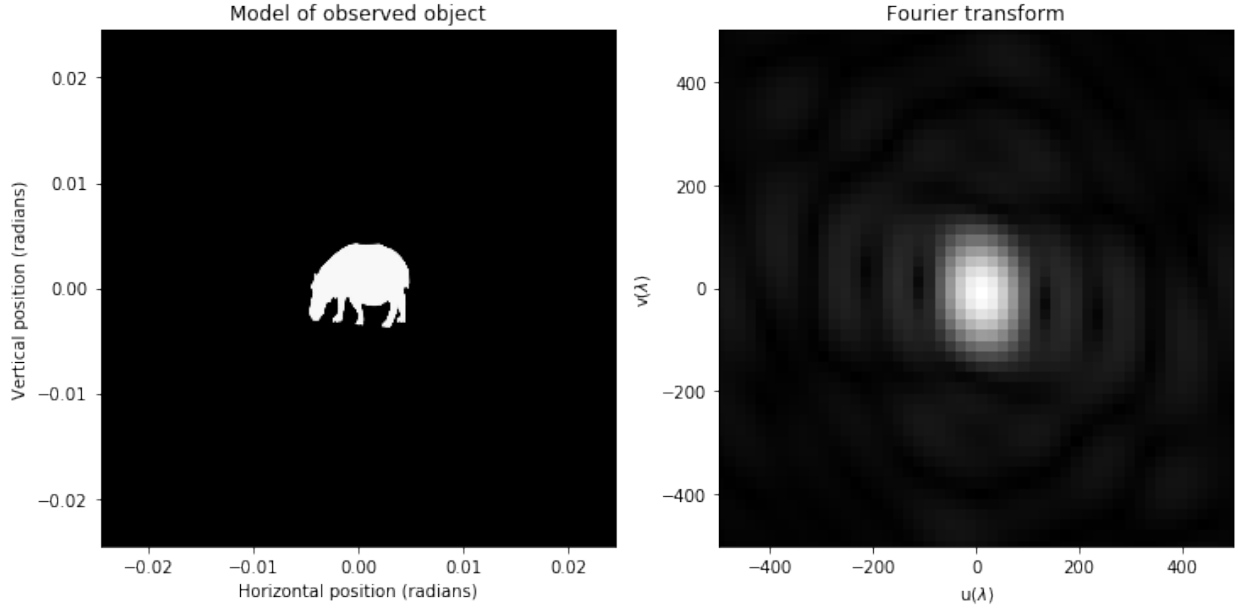


Figure 15: Left: An image of a uniformly illuminated hippo shaped object. Right: The Fourier transform of the image.

We can see in Figure 15 that the small structure elements in the hippo lead to deformation of the Fourier transform, similarly to the realistic image of the Sun in Figure 13. This time, the larger structure is from the bottom left to the top right, since its head is on the left, and therefore the Fourier transform is stretched towards the top left and bottom right. Having a closer look at a slice of the Fourier transform in Figure 16, we see that also this fit is far from perfect, and that small structure elements are still unresolved at small baseline lengths. However, calculating the size of this hippo would still be quite accurate, since it has very sharp edges.

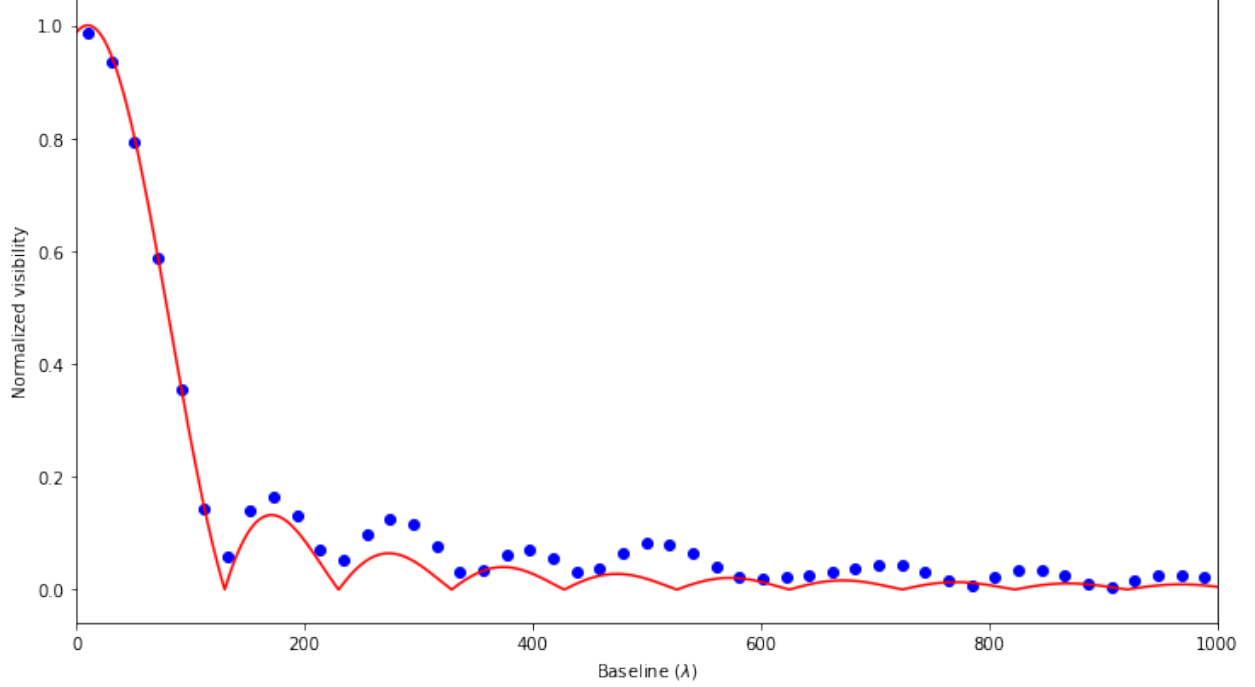


Figure 16: Slice of the Fourier transform of the image of the hippo shaped object, and an analytic fit of a Bessel function.

4 Discussion

From the simulation of the visibilities for different models it can be seen that the outcomes, although they exhibit the same overall behavior, differ a lot in the details. Different shapes and structure elements on the Sun have a huge effect on its Fourier transform, and therefore on the visibility. For many astronomical objects this is usually not a problem, since they are so far away that any changes in appearance are practically unnoticeable. This makes that the visibilities of these objects are static as well, and therefore the reconstruction of the size and location of different structure elements can be done very accurately by interferometric observations. An example of such an object is Cygnus A, a bright radio galaxy with two radio lobes of 160 thousand light years in size, at a distance of 600 million light years from Earth [Strange 2008]. On this scale, significant changes in the appearance of Cygnus A as seen from Earth take thousands of years to be noticed, and therefore can not be seen in observations.

For the Sun the situation is totally different: the Sun is only 8 light minutes from Earth and is half a degree in size on the sky. On this scale details can actually be observed to exist, and also to change over time. The problem that arises from this is that the visibilities of the Sun change all the time, and therefore the reconstruction of the Sun's size and smaller structure elements seems a difficult task. Fortunately, determining the size is not completely impossible: by doing many observations, its angular size can still be seen in the width of the primary peak of the visibility, in which the Sun goes from being unresolved to being resolved.

However, determining the sizes and especially the locations of solar flares and sunspots is a different story. Their existence is already hard to measure, since measurements at higher baselines are required. The simplest example is the donut model in Section 3.3.3. In the visibilities it could be seen fairly clearly how the hole in the center of the disk changed the amplitude of the higher order peaks: a low peak between two higher peaks means that near these baselines a piece of structure is resolved. The next peak with a larger baseline is then higher, since the structure piece then covers part of a second pixel. This could be seen in an even extremer form for the image model in Section 3.3.5 and the hippo in Section 3.3.6, where the behavior of the peak amplitude was even funnier, which is as expected since there were more structure pieces of different size to resolve.

In fact, a series of fixed dish observations has already been performed by the KISS. A total of 12 data points has been obtained at different baselines over the course of a few days. Fitting a sinc and a Bessel function gave a resolving baseline of about 107λ [Farret Jentink 2018]. These observations can be used very well to estimate the overall size of the Sun, since the baselines nearly don't get much further than the second peak, making that the first peaks are known fairly well. Besides, the bias from the structure elements on the edge can be minimized, since the structure might have disappeared between two observation dates. For observing structure they are not really useful, since the structure of the higher order peaks is not observed. Furthermore, the sunspots and flares that existed during the first observation could have been moved or disappeared at the time of the second observation.

In Figure 17 and Figure 18 a comparison of the visibilities for the six models is shown. In Figure 17 it can be clearly seen how the visibility of a disk dies out rather quickly, because there is no smaller structure to resolve. The hole of the donut model can be identified by the fact that the peak around 400λ is very low (where it becomes resolved), and the amplitude then goes back up again. Also the large secondary peak tells us that there still is structure to be resolved. The behavior of the image model even exhibits a more irregular behavior, since there are many irregular structure elements that become resolved at different baselines.

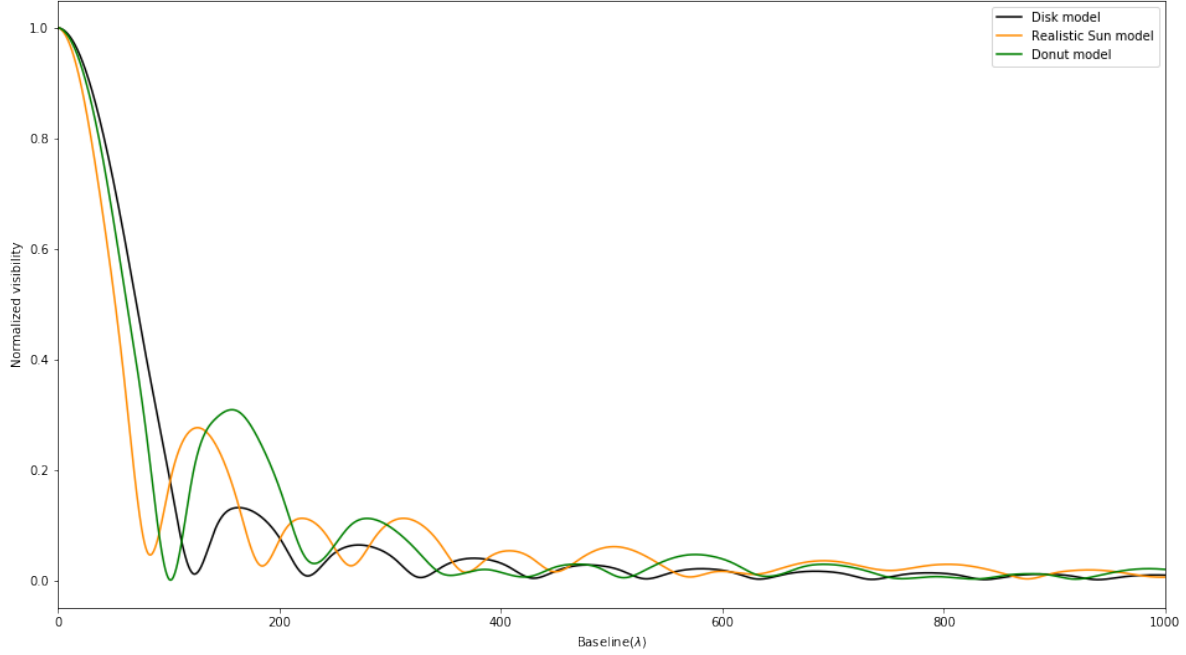


Figure 17: Cubic interpolation of the visibilities to compare the behavior of the disk model, the image of the Sun, and the donut model.

In Figure 18 it is also visible that the hippo has a lot of small unresolved structure by the irregular behavior of its visibility at large baseline lengths. Parts of the square model also take a lot of time to resolve, but this is more periodic since they are the harmonics of the resolving of the full square. The Gaussian model is resolved piece by piece without harmonics, which was expected since it has no sharp cutoffs.

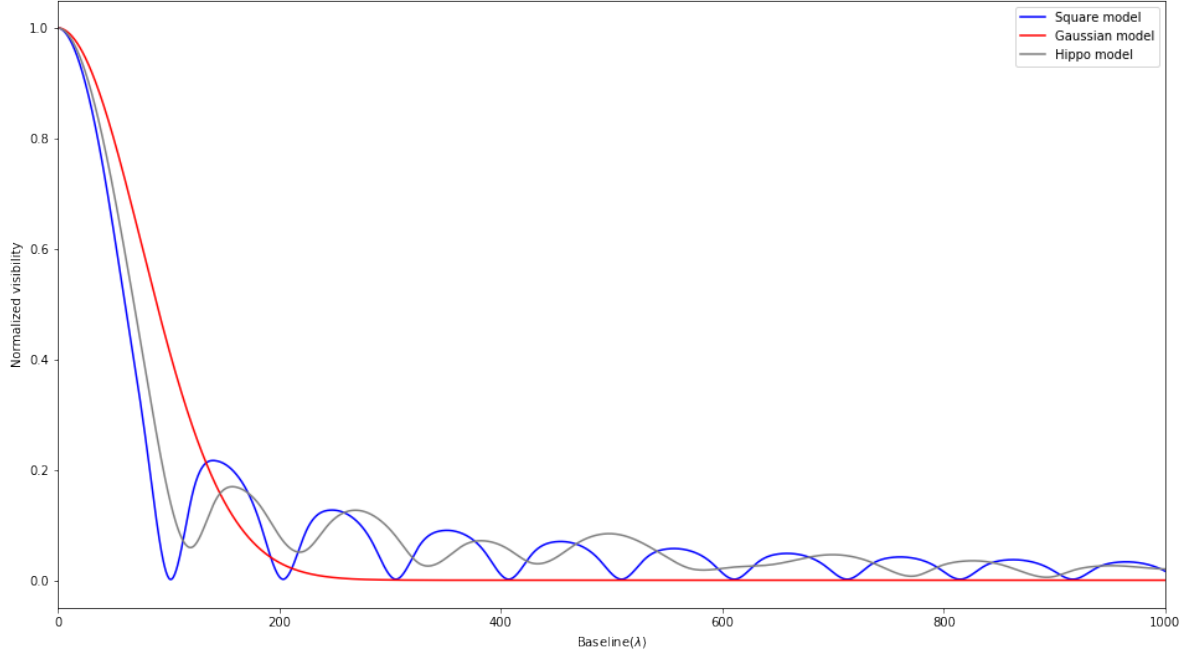


Figure 18: Cubic interpolation of the visibilities to compare the behavior of the square model, the Gaussian model, and the hippo model.

5 Conclusion and recommendation

5.1 Conclusions

The simulations of the observations of the KISS interferometer showed very clearly what is to expect from these observations. The fringes observed by the interferometer get attenuated by an effect called the visibility, which is depending on the Fourier transform of the observed source. The first conclusion of this study is that the visibility and the observations of an object depend heavily on the shapes and structure that can be observed in the object.

Determining the size of the Sun with great precision and accuracy might not be an easy task, since structures on the edge of its surface, like solar flares, influence the size of the large resolvable structure. As a consequence, the primary peak of the Fourier transform of the Sun will become narrower, changing the size of the resolving baseline. It might be that doing many observations at different dates will reduce this effect, since this structure on the Sun changes over time.

Large structure elements, like in the case of the donut model, can be observed relatively well. A periodic behavior can be observed in the amplitude of the higher order peaks of the visibility. At the baseline where the structure becomes resolved, the amplitude of the visibility peak is relatively low compared to the neighbor peaks.

Observing small structures near the center of the Sun's surface can not be done directly with the interferometer in its current state, since these structures only become resolved at baselines

that are larger than the setup. If the structure would be observable in the radio wave regime, the visibility would become less periodic, and also the amplitudes would vary more, because the visibility goes down when a piece of structure becomes resolved. However, the existence of the small structure can be deduced from the fact that the amplitude of the low order peaks in the visibility are relatively large, since at such baselines there is a lot of unresolved structure.

5.2 Recommendations

The number of possible models is infinite, but every model that is simulated adds to the understanding of the behavior of visibilities, and therefore what the observations will look like. Therefore more research on different models will be very helpful.

Adding at least one dish will give an even better understanding of what the interferometer actually observes. The interferometer in its current state only observes the visibility in one dimension. Adding an extra dish will make the correlation more difficult, but will also create the opportunity to obtain two-dimensional visibilities, and therefore it will be possible to reconstruct two-dimensional images from those visibilities.

References

- Abramowitz, Milton and Irene A Stegun (1964). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Vol. 55. Courier Corporation.
- Cittert, Pieter Hendrik van (1934). “Die wahrscheinliche Schwingungsverteilung in einer von einer Lichtquelle direkt oder mittels einer Linse beleuchteten Ebene”. In: *Physica* 1.1-6, pp. 201–210.
- Farret Jentink, Casper (2018). *Constructing the Backend of the Kapteyn Interferometer for Short baseline Solar observations (KISS)*.
- Hecht, Eugene (2014). *Optics*. Pearson.
- Lensen, Mathijn (2018). *Constructing the frontend of the Kapteyn Interferometer and investigating the single dish configuration*.
- Michelson, Albert A and Edward W Morley (1887). “On the Relative Motion of the Earth and of the Luminiferous Ether”. In: *Sidereal Messenger*, vol. 6, pp. 306–310 6, pp. 306–310.
- NASA (2010). *Great Ball of Fire — Nasa*. [Online; accessed Juli 6, 2018]. URL: https://www.nasa.gov/multimedia/imagegallery/image_feature_1732.html.
- NRAO (2017). *NRAO Very Large Array*. [Online; accessed Juli 8, 2018]. URL: <http://www.vla.nrao.edu/>.
- Russell, Randy (2007). *The Multispectral Sun - Windows to the Universe*. [Online; accessed Juli 8, 2018]. URL: https://www.windows2universe.org/?page=/sun/spectrum/multispectral_sun_overview.html.
- Strange, D (2008). *The Radio Galaxy Cygnus A*. [Online; accessed Juli 6, 2018]. URL: <https://web.archive.org/web/20080725125333/http://ourworld.compuserve.com/homepages/dstrange/cygnusa.htm>.
- Thompson, A. R., J. M. Moran, and G. W. Swenson Jr. (2017). *Interferometry and Synthesis in Radio Astronomy, 3rd Edition*. DOI: [10.1007/978-3-319-44431-4](https://doi.org/10.1007/978-3-319-44431-4).
- Wilson, T. L., K. Rohlfs, and S. Hüttemeister (2013). *Tools of Radio Astronomy*. DOI: [10.1007/978-3-642-39950-3](https://doi.org/10.1007/978-3-642-39950-3).
- Zernike, F. (1938). “The concept of degree of coherence and its application to optical problems”. In: *Physica* 5, pp. 785–795. DOI: [10.1016/S0031-8914\(38\)80203-2](https://doi.org/10.1016/S0031-8914(38)80203-2).

A Appendix

A.1 Derivations

A.1.1 Correlation

The correlated signal given in Eq. (4) can be obtained by working out the average of the combined voltages, given by Eq. (3):

$$\begin{aligned}
\langle V_{\text{comb}}(t) \rangle &= \langle V_1^2(t) + V_2^2(t) + 2V_1(t)V_2(t) \rangle \\
&= E^2 \langle \cos^2(2\pi\nu(t - \tau_g)) + \cos^2(2\pi\nu t) + 2\cos(2\pi\nu(t - \tau_g))\cos(2\pi\nu t) \rangle \\
&= E^2 [\langle \cos^2(2\pi\nu(t - \tau_g)) \rangle + \langle \cos^2(2\pi\nu t) \rangle + 2\langle \cos(2\pi\nu(t - \tau_g))\cos(2\pi\nu t) \rangle] \\
&= E^2 \left[\frac{1}{2} + \frac{1}{2} + 2 \cdot \frac{1}{2} \langle \cos(2\pi\nu(t + t - \tau_g)) + \cos(2\pi\nu(t - t + \tau_g)) \rangle \right] \\
&= E^2 [1 + (\langle \cos(2\pi\nu(2t - \tau_g)) \rangle + \langle \cos(2\pi\nu\tau_g) \rangle)] \\
&= E^2 [1 + \langle \cos(2\pi\nu\tau_g) \rangle] \\
&= E^2 [1 + \cos(2\pi\nu\tau_g)].
\end{aligned} \tag{4}$$

A.1.2 Bandwidth effect

The attenuation of the fringes caused by the finite bandwidth given by (6) can be derived by integrating the fringe function given in over the frequency range:

$$F(\theta) = \frac{1}{\Delta\nu} \int_{\nu_0 - \Delta\nu/2}^{\nu_0 + \Delta\nu/2} E^2 \left(1 + \cos\left(\frac{2\pi B \sin \theta}{\lambda}\right) \right) d\nu.$$

Using that $\frac{1}{\lambda} = \frac{\nu}{c}$ and setting $a = \frac{2\pi B \sin \theta}{c}$ this becomes

$$\begin{aligned}
F(\theta) &= \frac{E^2}{\Delta\nu} \int_{\nu_0 - \Delta\nu/2}^{\nu_0 + \Delta\nu/2} (1 + \cos(a\nu)) d\nu \\
&= \frac{E^2}{\Delta\nu} \left[\nu + \frac{\sin(a\nu)}{a} \right]_{\nu_0 - \Delta\nu/2}^{\nu_0 + \Delta\nu/2} \\
&= \frac{E^2}{\Delta\nu} \left[\nu_0 + \Delta\nu/2 - \nu_0 + \Delta\nu/2 + \frac{\sin(a(\nu_0 + \Delta\nu/2))}{a} - \frac{\sin(a(\nu_0 - \Delta\nu/2))}{a} \right] \\
&= \frac{E^2}{\Delta\nu} \left[\Delta\nu + \frac{2}{a} \cos(a\nu_0) \sin\left(\frac{a}{2}\Delta\nu\right) \right] \\
&= E^2 \left[1 + \cos(a\nu_0) \frac{\sin\left(\frac{a}{2}\Delta\nu\right)}{\frac{a}{2}\Delta\nu} \right] \\
&= E^2 \left[1 + \cos\left(\frac{2\pi B \sin \theta}{\lambda_0}\right) \frac{\sin\left(\pi \frac{B \sin \theta}{\lambda_0} \frac{\Delta\nu}{\nu_0}\right)}{\pi \frac{B \sin \theta}{\lambda_0} \frac{\Delta\nu}{\nu_0}} \right].
\end{aligned} \tag{6}$$

A.1.3 Square Fourier transform

The Fourier transform of a square with sides of length a can be calculated as follows:

$$\begin{aligned} V(u, v) &= \int_{-a/2}^{a/2} \int_{-a/2}^{a/2} e^{-2\pi i(ux+vy)} dx dy \\ &= \int_{-a/2}^{a/2} \frac{-1}{2\pi i u} \left(e^{-2\pi i(ua/2+vy)} - e^{-2\pi i(-ua/2+vy)} \right) dy \\ &= \frac{-1}{4\pi^2 uv} \left[e^{-\pi i(ua+va)} - e^{-\pi i(ua-va)} - e^{-\pi i(-ua+va)} + e^{-\pi i(-ua-va)} \right] \\ &= \frac{-1}{4\pi^2 uv} (e^{-\pi iua} - e^{\pi iua}) (e^{-\pi i va} - e^{\pi i va}) \\ &= \frac{\sin(\pi ua)}{\pi u} \frac{\sin(\pi va)}{\pi v} \\ &= a^2 \text{sinc}(\pi ua) \text{sinc}(\pi va). \end{aligned} \tag{15}$$

A.2 Python Code

Asking for an image and model

```
1 import matplotlib.image as mpimg
2 import numpy as np
3 from scipy.misc import imread
4
5 modeltype = input("Model type? (square/disk/donut/sun/pentagram/hippo): ")
6
7 # Loading the correct image
8 if modeltype == "square":
9     load = mpimg.imread('Square.jpg')[:, :, 1]/np.max(mpimg.imread('Square.jpg')
10    [:, :, 1])
11
12 elif modeltype == "disk":
13     load = mpimg.imread('Disk.jpg')[:, :, 1]/np.max(mpimg.imread('Disk.jpg')[:, :, 1])
14
15 elif modeltype == "donut":
16     load = mpimg.imread('Donut.jpg')[:, :, 1]/np.max(mpimg.imread('Donut.jpg')
17    [:, :, 1])
18
19 elif modeltype == "sun":
20     load = mpimg.imread('Sun4.jpg')[:, :, 1]/np.max(mpimg.imread('Sun4.jpg')[:, :, 1])
21
22 elif modeltype == "hippo":
23     load = mpimg.imread('hippo.jpg')[:, :, 1]/np.max(mpimg.imread('hippo.jpg')
24    [:, :, 1])
25
26 elif modeltype == "pentagram":
27     load = mpimg.imread('Pentagram2.jpg')[:, :, 1]/np.max(mpimg.imread('Pentagram2.
28     jpg')[:, :, 1])
29
30 elif modeltype == "crescent":
31     load = mpimg.imread('Crescent.jpg')[:, :, 1]/np.max(mpimg.imread('Crescent.jpg')
32    [:, :, 1])
```

```

29 elif modeltype == "gaussian":
30     xarr = np.linspace(-1500,1500,3000)
31
32     def gaussian(grid,scale):
33         return np.exp(-grid**2/(2*scale**2))
34
35     xgr, ygr = np.meshgrid(xarr,xarr)
36     lenarr = (xgr**2 + ygr**2)**(0.5)
37     load = gaussian(lenarr,300/(2*np.sqrt(2*np.log(2))))
38
39 else:
40     print("Not a valid model")
41
42 # Choosing the right analytic Fourier transform
43 if modeltype == "square":
44     def model(p, x):
45         A, angwidth, offset = p # Absolute jinc model
46         arg = angwidth*(x-offset) # Parameters which are adjusted by kmpfit
47         y = A*np.sinc(arg) # Function
48         return abs(y)
49
50     def residuals(p, data):
51         x, y = data # Residuals function needed by kmpfit
52         return y-model(p,x) # Data arrays is a tuple given by programmer
53
54     paramsinitial = [180000, 0.01, 10]
55
56     zero = 1 # Correction for the first zero of the Fourier transform analytic
57             # function
58 elif modeltype == "disk" or modeltype == "sun" or modeltype == "hippo":
59     def model(p, x):
60         A, angwidth, offset = p # Absolute jinc model
61         arg = angwidth*(x-offset) # Parameters which are adjusted by kmpfit
62         y = A*2*j1(arg)/arg # Function
63         return abs(y)
64
65     def residuals(p, data):
66         x, y = data # Residuals function needed by kmpfit
67         return y-model(p,x) # Data arrays is a tuple given by programmer
68
69     paramsinitial = [100000, 0.036, 10]
70
71     zero = 1.22 # Correction for the first zero of the Fourier transform analytic
72             # function
73 elif modeltype == "donut":
74     def model(p, x):
75         A1, angwidth1, offset1, A2, angwidth2, offset2 = p # Multiple absolute
76         # jinc models
77         arg1 = angwidth1*(x-offset1) # Parameters which are
78         arg2 = angwidth2*(x-offset2) # adjusted by kmpfit
79         y = A1*2*j1(arg1)/arg1+A2*2*j1(arg2)/arg2 # Function
80         return abs(y)

```

```

81     def residuals(p, data):          # Residuals function needed by kmpfit
82         x, y = data                  # Data arrays is a tuple given by programmer
83         return y-model(p,x)
84
85     paramsinitial = [200000, 0.03, 10, -200, 0.01, 10]
86
87     zero = 1 # Correction for the first zero of the Fourier transform analytic
88               function
89 elif modeltype == "gaussian":
90     def model(p,x):                  # Gaussian model
91         A, angwidth, offset = p      # Parameters which are adjusted by
92                                       kmpfit
93         y = A*np.exp(-(x-offset)**2/(2*angwidth**2))
94         return abs(y)                # Function
95
96     def residuals(p, data):          # Residuals function needed by kmpfit
97         x, y = data                  # Data arrays is a tuple given by programmer
98         return y-model(p,x)
99
100    paramsinitial = [200000, 10, 10]

```

Visibility calculator

```

1  %matplotlib inline
2  from scipy.fftpack import *
3  from scipy.constants import *
4  from scipy.ndimage.interpolation import rotate
5  import matplotlib.pyplot as plt
6  import matplotlib.cm as cm
7  import numpy as np
8
9  nu = 11.2e9
10 labda = c/nu # Calculates wavelength
11
12 fig1 = plt.figure(figsize=(12,13.5))
13
14 #=====
15 # Generating the visibility from a chosen model
16 #=====
17
18 # Taking the source as high intensity, and background as low intensity
19 topleft = load[:,int(load.shape[0]/100),:int(load.shape[1]/100)]
20
21 if np.mean(topleft)<0.1:
22     image = load/np.max(load)
23 else:
24     image = 1-load/np.max(load)
25
26 #image = rotate(image,0,reshape=False)
27
28 # Taking a slice through the middle
29 sliceim = image[int(len(image[0])/2)]
30
31 # Converting dimensions
32 angsize = np.radians(0.53274186788) # input in degrees
33 mockrange = np.array(range(len(sliceim)))[sliceim>0.1]

```



```

34 conv = len(sliceim)/2*angsize/(mockrange[-1]-mockrange[0]+1)
35 #conv = len(sliceim)/2*angsize/600
36
37 x = np.linspace(-conv,conv,len(sliceim))
38 coordinates = (-conv,conv,-conv,conv)
39
40 # Plotting model with corrected dimensions
41 ax1 = fig1.add_subplot(121)
42 ax1.imshow(image, cmap=cm.gray, extent=coordinates)
43 ax1.set_title(r"Model of observed object")
44 ax1.set_xlabel(r"Horizontal position (radians)")
45 ax1.set_ylabel(r"Vertical position (radians)")
46
47 # Taking the Fourier transform of the slice
48 Z = fftshift(abs(fft2(image)))
49
50 window = len(x)
51 spacing = (np.max(x)-np.min(x))/len(x)
52
53 # Converting Fourier dimensions
54 fconv = np.max(fftfreq(window,spacing))/np.max(coordinates)*conv
55 fourierconv = (-fconv,fconv,-fconv,fconv)
56
57 # Plotting Fourier transform
58 ax2 = fig1.add_subplot(122)
59 ax2.imshow(Z, cmap=cm.gray, extent=fourierconv)
60 ax2.set_title(r"Fourier transform")
61 ax2.set_xlabel(r"u($\lambda$)")
62 ax2.set_ylabel(r"v($\lambda$)")
63 ax2.set_xlim(-500,500)
64 ax2.set_ylim(-500,500)
65
66 plt.show()

```

Fitting a function through the numerical Fourier transform

```

1 from scipy.special import *
2 from scipy.interpolate import interp1d
3 from kapteyn import kmpfit
4
5 fig2 = plt.figure(figsize=(12,6.75))
6 ax3 = fig2.add_subplot(111)
7
8 # Taking a slice of the Fourier transform
9 fourier_y = Z[np.unravel_index(Z.argmax(), Z.shape)[1]]
10 fourier_x = np.linspace(-fconv,fconv,len(fourier_y))
11
12 xfit = np.linspace(np.min(fourier_x),np.max(fourier_x),400000)
13
14 #=====
15 # Fitting a smooth function using Kapteyn fit
16 #=====
17
18 fitobj = kmpfit.Fitter(residuals=residuals, data=(fourier_x,fourier_y))
19 fitobj.fit(params0=paramsinitial)
20
21 yfit = model(fitobj.params,xfit)

```

```

22
23 amp = np.max(yfit)
24 width = fitobj.params[1]
25 offset = fitobj.params[2]
26
27 # Plotting the Fourier transform with fitted smooth function
28 ax3.scatter(fourier_x, fourier_y/amp, color='blue')
29 ax3.plot(xfit, yfit/amp, color='red')
30 ax3.set_xlim(0, 1000)
31 ax3.set_xlabel(r"Baseline ( $\lambda$ )")
32 ax3.set_ylabel(r"Normalized visibility")
33
34 print(r"Optimal Fourier fit amplitude: {:.10f}".format(amp))
35 print(r"Optimal Fourier fit angular width: {:.10f}".format(width))
36 print(r"Optimal Fourier fit offset: {:.10f}".format(offset))
37
38 # Finding the first zero => resolving baseline
39 stepdist = len(xfit)/(np.max(xfit)-np.min(xfit))
40 firstdip = len(yfit[:int(len(yfit)/2)]+np.argmin(yfit[int(len(yfit)/2):int(len(yfit)/2)+int(200*stepdist)]))
41 resolver = xfit[firstdip]-offset
42 print(r"First zero (wavelengths): {:.10f}".format(resolver))
43 print(r"Resolving baseline (meters): {:.10f}".format((resolver)*labda))
44 print(r"Angular diameter of the Sun (degrees): {:.10f}".format(np.degrees(zero/resolver)))

```

Simulating observed fringes for tracking and fixed observations with a given baseline

```

1 %matplotlib inline
2 from scipy.fftpack import *
3 from scipy.constants import *
4 from scipy.misc import imread
5 from scipy.special import *
6 from kapteyn import kmpfit
7 import matplotlib.pyplot as plt
8 import matplotlib.cm as cm
9 import numpy as np
10
11 #=====
12 # Plotting fringes for tracking observations
13 #=====
14
15 B = float(input("Baseline(m): ")) # Baseline measured on the ground
16
17 D = 0.352
18 nu = 11.2e9
19 deltanu = 40e6
20
21 labda = c/nu # Calculates wavelength
22 hpbw = 1.22*c/(nu*D) # Calculates half power beam width
23 res = 1.22*c/(nu*B) # Calculates resolution
24
25 print("Field of view: {:.6f} degree".format(np.degrees(hpbw)))
26 print("Resolution: {:.6f} degree".format(np.degrees(res)))
27
28 if res > np.radians(np.degrees(zero/resolver)): # Determine whether Sun is
    resolved

```

```

29     print("Sun is not resolved")
30 else:
31     print("Sun is resolved")
32
33 n = 200001
34 x1 = np.linspace(0,pi/2,n)      # Defining the observation angles for tracking
35 x2 = np.linspace(-hpbw,hpbw,n)  # Defining the observation angles for fixed
36
37 def trackingfringes(a,b,d):
38     u = B*np.sin(a)/labda
39     fringe = np.cos(2*pi*u) # Delay causes fringes
40     smearing = np.sinc(u*deltanu/nu) # Finite bandwidth causes smearing
41
42     if len(fitobj.params) == 6:
43         visibility = model((1,fitobj.params[1],0,fitobj.params[3]/fitobj.params
44             [0],fitobj.params[4],0),B*np.cos(a)/labda)
45     else:
46         visibility = model((1,fitobj.params[1],0),B*np.cos(a)/labda)
47
48     f = (1+visibility*fringe*smearing) # Defining the fringe function
49     t = a*4 # Converting degrees to minutes
50     return a, t, f
51
52 trackingfringeplot = trackingfringes(x1,B,D)
53
54 # Plotting tracking fringes
55 fig3 = plt.figure(figsize=(12,6.75))
56 ax4 = fig3.add_subplot(111)
57 ax4.plot(np.degrees(trackingfringeplot[0]),trackingfringeplot[2])
58 #ax4.set_title(r"Fringes at baseline length of {:.2f} m".format(np.float(B)))
59 ax4.set_xlabel(r"Offset (degrees)")
60 ax4.set_ylabel(r"Amplitude (arbitrary unit)")
61 ax5 = ax4.twinx()
62 ax5.plot(np.degrees(trackingfringeplot[1]),trackingfringeplot[2])
63 ax5.set_xlabel(r"Offset (minutes)")
64
65 def primarybeam(x, mu, FWHM):
66     sigma = FWHM/(2*np.sqrt(2*np.log(2)))
67     return np.exp(-1/2*((x-mu)/sigma)**2)/(sigma*np.sqrt(2*pi))
68
69 def fixedfringes(a,b,d):
70     u = B*np.sin(a)/labda
71     fringe = np.cos(2*pi*u) # Delay causes fringes
72     smearing = np.sinc(u*deltanu/nu) # Finite bandwidth causes smearing
73     attenuation = primarybeam(a,0,hpbw)
74
75     if len(fitobj.params) == 6:
76         visibility = model((1,fitobj.params[1],0,fitobj.params[3]/fitobj.params
77             [0],fitobj.params[4],0),B*np.cos(a)/labda)
78     if len(fitobj.params) == 3:
79         visibility = model((1,fitobj.params[1],0),B*np.cos(a)/labda)
80
81     f = attenuation*(1+visibility*fringe*smearing) # Defining the fringe function
82     t = a*4 # Converting degrees to minutes
83     return a, t, f

```

```

83 fixedfringeplot = fixedfringes(x2,B,D)
84
85 # Plotting tracking fringes
86 fig4 = plt.figure(figsize=(12,6.75))
87 ax6 = fig4.add_subplot(111)
88 ax6.plot(np.degrees(fixedfringeplot[0]),fixedfringeplot[2])
89 #ax6.set_title(r"Fringes at baseline length of {:.2f} m".format(np.float(B)))
90 ax6.set_xlabel(r"Offset (degrees)")
91 ax6.set_ylabel(r"Amplitude (arbitrary unit)")
92 ax7 = ax6.twinx()
93 ax7.plot(np.degrees(fixedfringeplot[1]),fixedfringeplot[2])
94 ax7.set_xlabel(r"Offset (minutes)")
95
96 plt.show()

```

Some code to compare interpolated visibilities

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib.image as mpimg
4  from scipy.misc import imread
5
6  im1 = mpimg.imread('Disk.jpg')[:, :, 1]/np.max(mpimg.imread('Disk.jpg')[:, :, 1])
7  im2 = mpimg.imread('Donut.jpg')[:, :, 1]/np.max(mpimg.imread('Donut.jpg')[:, :, 1])
8  im3 = mpimg.imread('Sun4.jpg')[:, :, 1]/np.max(mpimg.imread('Sun4.jpg')[:, :, 1])
9  im4 = mpimg.imread('hippo.jpg')[:, :, 1]/np.max(mpimg.imread('hippo.jpg')[:, :, 1])
10 im5 = mpimg.imread('Square.jpg')[:, :, 1]/np.max(mpimg.imread('Square.jpg')[:, :, 1])
11
12 xarr = np.linspace(-1500,1500,3000)
13
14 def gaussian(grid,scale):
15     return np.exp(-grid**2/(2*scale**2))
16
17 xgr, ygr = np.meshgrid(xarr,xarr)
18 lenarr = (xgr**2 + ygr**2)**(0.5)
19 im6 = gaussian(lenarr,300/(2*np.sqrt(2*np.log(2))))
20
21 fig1 = plt.figure(figsize=(16,9))
22 ax1 = fig1.add_subplot(111)
23 fig2 = plt.figure(figsize=(16,9))
24 ax2 = fig2.add_subplot(111)
25
26 def fftnormalizer(x):
27     return fftshift(abs(fft2(x)))/np.max(fftshift(abs(fft2(x))))
28
29 fourier_x = np.linspace(-fconv-10.75,fconv-10.75,len(fourier_y))
30
31 Z1 = fftnormalizer(1-im1)
32 Z2 = fftnormalizer(1-im2)
33 Z3 = fftnormalizer(im3)
34 Z4 = fftnormalizer(1-im4)
35 Z5 = fftnormalizer(1-im5)
36 Z6 = fftnormalizer(im6)
37
38 fourier_y1 = Z1[np.unravel_index(Z1.argmax(), Z1.shape)[1]]
39 fourier_y2 = Z2[np.unravel_index(Z2.argmax(), Z2.shape)[1]]
40 fourier_y3 = Z3[np.unravel_index(Z3.argmax(), Z3.shape)[1]]

```

```

41 fourier_y4 = Z4[np.unravel_index(Z4.argmax(), Z4.shape)[1]]
42 fourier_y5 = Z5[np.unravel_index(Z5.argmax(), Z5.shape)[1]]
43 fourier_y6 = Z6[np.unravel_index(Z6.argmax(), Z6.shape)[1]]
44
45 xinter = np.linspace(-fconv-10.75,fconv-10.75,100000)
46
47 ax1.plot(xinter, interp1d(fourier_x, fourier_y1, kind="cubic")(xinter), color="
    black", label="Disk model")
48 ax1.plot(xinter, interp1d(fourier_x, fourier_y3, kind="cubic")(xinter), color="
    darkorange", label="Realistic Sun model")
49 ax1.plot(xinter, interp1d(fourier_x, fourier_y2, kind="cubic")(xinter), color="
    green", label="Donut model")
50 ax2.plot(xinter, interp1d(fourier_x, fourier_y5, kind="cubic")(xinter), color="
    blue", label="Square model")
51 ax2.plot(xinter, interp1d(fourier_x, fourier_y6, kind="cubic")(xinter), color = "
    red", label="Gaussian model")
52 ax2.plot(xinter, interp1d(fourier_x, fourier_y4, kind="cubic")(xinter), color = "
    grey", label="Hippo model")
53
54 ax1.set_xlabel(r"Baseline($\lambda$)")
55 ax1.set_ylabel(r"Normalized visibility")
56 ax2.set_xlabel(r"Baseline($\lambda$)")
57 ax2.set_ylabel(r"Normalized visibility")
58
59 ax1.set_xlim(0,1000)
60 ax2.set_xlim(0,1000)
61
62 ax1.legend()
63 ax2.legend()
64 plt.show()

```