UNIVERSITÉ DE GENÈVE

MASTER IN ASTROPHYSICS

Interferometry with TELESTO to measure the diameter of Mars

Student:
Maddalena Bugatti

Supervisors:
François Bouchy
Damien Ségransan
Bruno Chazelas

Academic years 2020/2021

Contents

1 Introduction

The objective of this research project is, first of all, to provide the theoretical notions related to the interferometry and its application to calculate the diameter of stars and planets. Secondly, it is to present the experiment conducted with the TELESTO telescope at the observatory of Geneva. This experiment aims to investigate the possibility of calculating the diameter of Mars using interferometry.
Initially the goal was to calculate the diameter of some of the planets in the solar system, but due to the lack of time it was decided to focus only on the analysis of a single planet. Mars during this winter was very close to the earth and therefore clearly visible, so it was decided to take the latter as reference. Moreover, the trees around the observatory prevent observing objects below 20-25° of altitude (depending on the weather); therefore other bright planets such as Jupiter or Saturn that were low in the sky were discarded for analysis. The report is divided in 4 sections. After a historical excursus, the first part provides the theoretical introduction regarding the particular properties of light and the concept of interferometry and diffraction. In the second part we focus on the experimantal activity, on the procedure of the analysis and on the results.

1.1 Historical excursus

The major problem in astronomical observations is the detection of distant objects and then the improvement in spatial/angular resolution.
For a long time starting from the time of Galileo Galilei (1564-1642), astronomers and astrophysicists have tried to increase the size of telescopes more and more to improve the spatial resolution of observations. In fact, the angle θ at which is seen an astronomical object depends on the telescope diameter D and the wavelength of the light, λ, through the formula: $\theta = \lambda/D$. The image of a light source like a star or a planet given by a telescope is reproduced in Figure 1(a). There is the plane wavefront (in blue) which falls on the telescope of diameter D [9]. The resulting image is a diffraction pattern, with the maximum at the centre. For instance, with a wavelength of 500 nm and a dimeter of 1 m, we should be able to detect an object which an angular size of 100 $milliarcseconds$, as Proxima Centauri, which is the nearest star to our solar system. Starting from this reasoning, one could therefore think that with an ideal telescope with a huge diameter it would be possible to calculate the diameters of all celestial objects. However, for distant objects, there is another difficulty: the turbulent motion of the Earth's atmosphere. This turbulence affects the resolution of the image. The majority of stars are far away from us and even today's largest telescopes such as the Very Large Telescope in Chile (8.2 diameter), are unable to directly resolve their angular dimensions.

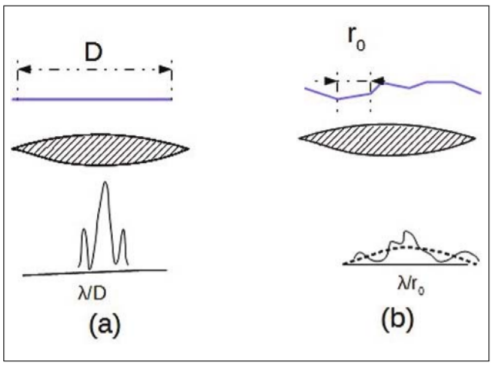

Figure 1: Diffraction pattern generated by the light of a star considering or not the atmospheric turbulence (respectively b and a)

The problem is linked with the diffraction limit [9]. In fact, to obtain a clear image, the light of an observed star has to converge at the focus of the telescope and all the photons arriving to the mirror have to be in phase. Any scratch in the mirror or variation of the refractive index in the atmosphere can cause a phase difference. The reason is quantitatively explained looking at Figure 1(b). If we attribute a size of r_0 (typically 10 cm at visible wavelengths) to the atmospheric disturbances, the error in phase would be around one radian. The obtained image would be smooth and the new wavelength would be equal to λ/r_0 (dashed line). As a result it would be very difficult to find the real size of a distant star. High-resolution observations require to use adaptive optics instruments to correct the effects of atmospheric turbulence.

Hippolyte Fizeau ($1819 - 1896$) was the first who proposed to use interference of light in order to overcome the difficulty to build big telescope [1]. His idea was to cover the top of the telescope with an opaque object, leaving two holes less than 10 cm in size. This technique is called spatial interferometry. Thanks to this technique, Albert A. Michelson and Francis G. Pease obtained the first successful measurement in 1920 [1]. They determined the diameter of α Orionis to be 0.047 $arcsec$. This was an incredible result, because at that time the smallest diameter that was calculated with a simple telescope was around 1 $arcsec$. Nevertheless, spatial interferometry was then abandoned until the seventies, because of technical issues with the mechanical stability at large distances between the pinholes.

In 1974 Antoine Labeyrie had the idea to combine two different telescopes instead of doing spacial interferometry with two pinholes on the same telescope. The results were exactly the same as those obtained with two pinholes, with the advantage that the separation between two indipendent telescopes could be much greater than the distance between two pinholes. The benefit of spatial interferometry is that the resolution depends only on the distance between the two piholes or telescopes, quantity which is called baseline. However, the improvement in resolution thanks to spacial interferometry has some flaws. Firstly, it is needed to do many observations at different baselines and orientations, to achieve a clear

image of faint objects in the space. Secondly, if two large telescopes are used, it is necessary to take into account an adaptive optics to eliminate the seeing problem.
In our experiment, as we will see better in the description of the experimental apparatus, we use pinholes with a diameter of few millimeters and therefore we can ignore the problem of atmospheric turbulence.

1.2 Fundamentals

1.2.1 Intensity pattern given by a two pinholes aperture

Let us consider the intensity distribution given by the light coming from a circular source which passes through a two pinholes aperture placed sufficiently distant from the source. The mathematical formula of the intensity pattern is given by the product between the interference due to the double aperture and the diffraction given by the finite radius of each of them [9].
Firstly, let us define the "Point Spread Function": it is a function which describes the two-dimensional distribution of light in a telescope focal plane for astronomical point sources. For a circular converging aperture the resulting PSF, given by the Fraunhofer diffraction, is called *Airy disk* and its expression is [6]:

$$PSF(\boldsymbol{v}) = \left| \frac{2J_1(\boldsymbol{v})}{\boldsymbol{v}} \right|^2 \text{, where} \quad \boldsymbol{v} = \frac{2\pi}{\lambda} \left(\frac{a}{f} \right) \boldsymbol{r} \tag{1}$$

where J_1 is the first-order Bessel function, v is known as the optical distance and depends on r, which is the radial distance from the optical axis in the focal plane. a is the radius of the aperture, λ is the wavelength of the light, and f the focal length. The representation of the Airy Disk on the focal plane is shown in Figure 2 on the left. As regards the interference

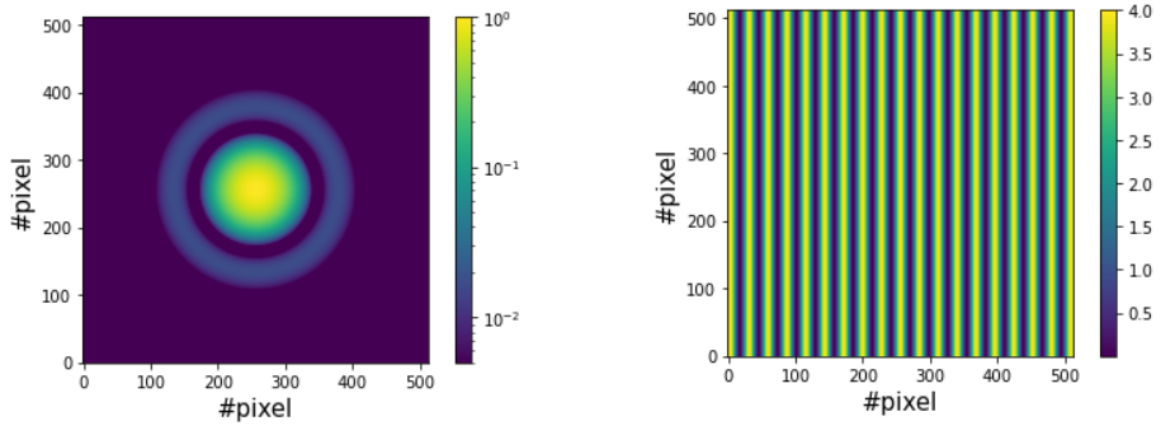

Figure 2: On the left: 2D representation of the Airy Disk. On the right: 2D representation of the interference pattern given by two slits. $\lambda = 879.7 \cdot 10^{-6}\ mm$, $a = 1.5\ mm$, $f = 2280\ mm$, pixscale $= 0.810612\ arcsec/pixel$, $B = 8\ mm$, $I_0 = 1$, $V = 1$, $theta = \frac{\pi}{2}$.

pattern, starting from Young's double slit experiment [1] we have that :

$$I = 2I_0\left\{1 + V\cos\left[\frac{2\pi B}{\lambda f}\cdot\Big(x\cos(\theta) + y\sin(\theta)\Big)\right]\right\} \tag{2}$$

where I_0 is the intensity of each electromagnetic wave coming out of each slits (we assume two identical aperture and then two equal initial intensities I_0): we could re-express I_0 in terms of the maximum and the adjacent minimum intensities resulting from the interference pattern: $I_0 = \frac{I_{max}-I_{min}}{4V}$. θ is the orientation of the slits, B represents the distance between the two slits (baseline) and x and y represents the two axis in the focal plane (the radial distance can be expressed as $\boldsymbol{r} = \sqrt{x^2+y^2}$). V is a very important quantity, called visibility which I will discuss later about. The representation of Eq.(2) on the focal plane is shown in Figure 2 on the right. The total intensity pattern given by two circular apertures is simply given by the product of equations (1) and (2):

$$I = 2I_0\left\{1 + V\cos\left[\frac{2\pi B}{\lambda f}\cdot\Big(x\cos(\theta) + y\sin(\theta)\Big)\right]\right\}\left|\frac{2J_1(\boldsymbol{v})}{\boldsymbol{v}}\right|^2 \tag{3}$$

Taking this equation we obtain Figure 3, which shows the superposition of the airy disk, with the interference fringes.

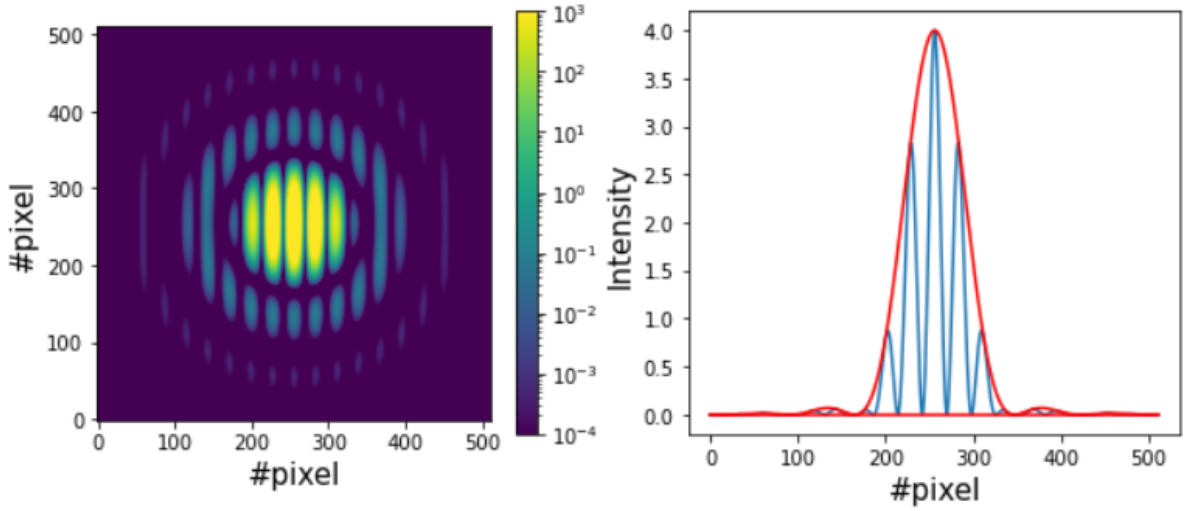

Figure 3: On the left it is shown the 2D representation of the total intensity pattern and on the right the 1D representation as a cut at $y = 256$ *pixel* (the total image measures 512 pixels). The two red lines represent the PSF multiplied by the maximum and the minimum of the intensity and the blue one the total intensity pattern

1.2.2 Visibility

More formally the mathematical expression for the visibility is [1]:

$$V = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{4}$$

where I_{max} and I_{min} represent, respectively, the maximum and the adjacent minimum on the interference pattern. Then V oscillates between 0 and 1. The visibility, for identical apertures, is defined as the contrast of the fringes pattern and it indicates the degree of coherence of the light. A contrast of 1 in a fringes pattern implies a perfect constructive and destructive interference in its maxima and minima. In other words the light waves coming from the two pinholes would be perfectly coherent if the visibility was 1. For instance, a monochromatic plane wave illuminating the screen with the two pinholes will create two waves emerging from the pinholes which are perfectly coherent.
Stars and planets are sources of incoherent light. However, observing them at a great distance (from the earth) they can be approximated to coherent sources thanks to the Van Cittert–Zernike theorem [2]. This latter states that the wave front of an incoherent source measured at a great distance is to be considered coherent. This reasoning can be easily visualized by imagining two stones falling into the center of a pond. Near the center of the pond the waves created by the two stones are very irregular in shape; then, progressively, spreading towards the edge of the pond, they diminish and are almost circular. Similarly, a star or a planet, seen from the earth, appears to be a coherent source of waves.
For a uniform source of light and a circular aperture the visibility results to be [5],[8]:

$$V = 2\left|\frac{J_1(\pi B\alpha/\lambda)}{(\pi B\alpha/\lambda)}\right| \tag{5}$$

where α is the angular size of the source. Changing the distance between the two holes the interference fringes could disappear. Therefore, the easiest way to compute the diameter of the source would be to look at the lowest value of B at which the interference fringes disappear. Looking at Eq.(5) this occurs when $B = \frac{1.22\lambda}{\alpha}$ ($V = 0$). But for a more precise value we need to fit some measured visibility at fixed baselines with Eq.(5). There are two ways to measure experimental visibility: with the analysis in the direct space or with the analysis in the Fourier space.

1.2.3 Direct space analysis

This analysis is clear: we can directly fit the images of Mars with the model in Eq.(3), leaving visibility as a parameter to fit. Once the visibility for different baselines has been extracted, they can be fitted again with Eq.(5) to obtain the angular dimension of Mars.

1.2.4 Fourier space analysis

For simplicity, let us see the theory of frequency spectrum analysis in one dimension. If we consider an optical, non-coherent imaging system, as is our case, the image that is formed on the screen of our telescope is given by the convolution between the object that we are studying (the planet in question) and the PSF of our mask [6]: $Image = Object \otimes PSF$. By calculating the Fourier transform of this equation we obtain: $\hat{Image} = \hat{Object} \cdot \hat{PSF}$, where we have applied the property that the convolution in the frequency space is equivalent to a simple product. If we compute the Fourier transform of Eq.(3) in one dimension we obtain:

$$\hat{I}(f_S) = \int_{-\infty}^{+\infty} I(x) \cdot \exp(-2\pi i f_S \cdot x)\,\mathrm{d}x$$

$$\hat{I}(f_S) = 2I_0 \int_{-\infty}^{+\infty} \left\{ \left| \frac{2J_1(v)}{v} \right|^2 + V \left| \frac{2J_1(v)}{v} \right|^2 \cos\left[\frac{2\pi B}{\lambda f} \cdot x\cos(\theta) \right] \right\} e^{-2\pi i f_S x}\mathrm{d}x$$

Where $v = \frac{2\pi a}{\lambda f}x$. Rewriting $q(x) = \frac{2\pi B x\cos(\theta)}{\lambda f}$ the cosine term becomes $\cos(q(x)) = \frac{e^{iq(x)}+e^{-iq(x)}}{2}$, then we obtain:

$$\hat{I}(f_S) = I_0 \int_{-\infty}^{+\infty} \left\{ 2\left| \frac{2J_1(v)}{v} \right|^2 + V \left| \frac{2J_1(v)}{v} \right|^2 [e^{iq(x)} + e^{-iq(x)}] \right\} e^{-2\pi i f_S x}\mathrm{d}x$$

$$\hat{I}(f_S) = I_0 \{T_0(f_S) + V \cdot [T_{+c}(f_S) + T_{-c}(f_S)]\} \tag{6}$$

The Fourier transform consists of three terms: each of them describes a triangular-shaped peak [4]. The first one, $T_0(f_S)$, which is the Fourier transform of the term with the Bessel function $\left|\frac{2J_1(v)}{v}\right|^2$ (single-pinhole diffraction), describes the central peak around $f_S = 0$. The second and third terms, $T_{\pm c}(f_S)$, describe the spatial frequency shift due to the product between the diffraction and the interference terms. If the two holes are identical, we have that $T_{-c}(f_S) = T_{+c}(f_S) \equiv T_{\pm c}(f_S)$ and if they are small enough (their diameter is less than at least two times the baseline) T_0 and $T_{\pm c}$ can be approximated as two non-overlapping functions [10]:

$$\hat{I}(f_S) \approx \begin{cases} T_0(f_S) & : -c \le f_S \le c \\ T_{\pm c}(f_S) & : f_S < -c \quad \wedge \quad f_S > c \end{cases}$$

where c is a positive real constant. Taking the square module of the Fourier transform and extracting the visibility we obtain that [10]:

$$V = 2\frac{\int_{-c}^{+c} T_{\pm c}(f_S)\,\mathrm{d}f_S}{\int_{-\inf}^{-c} T_0(f_S)\,\mathrm{d}f_S} \tag{7}$$

Figure 4 shows the PSD, namely the square module of the Fourier transform of the intensity pattern of Figure 3. It can be seen that the three integrals refer to three separate cones in the 2D Fourier space.

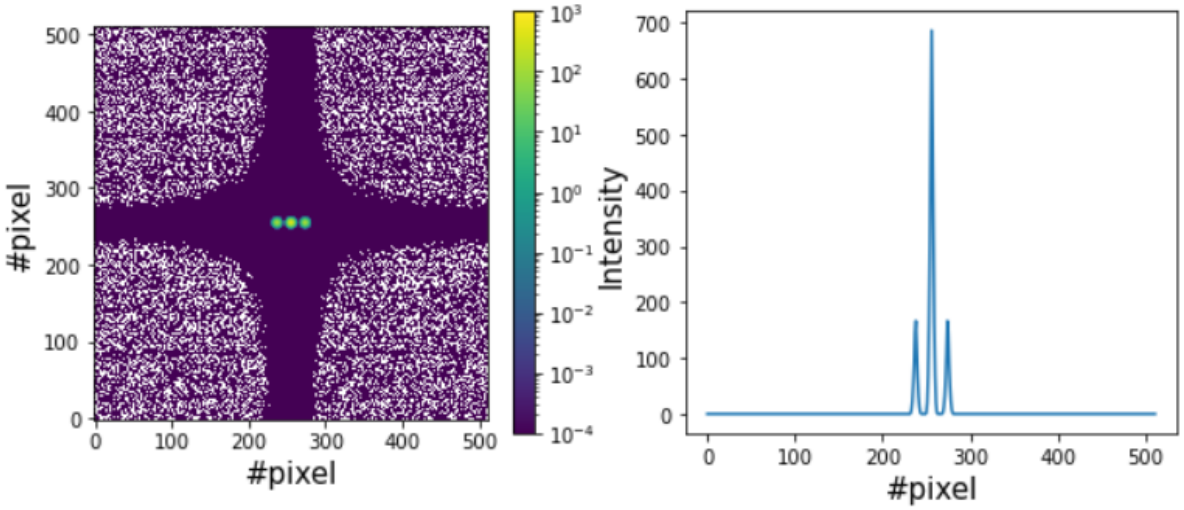

Figure 4: On the left it is shown the 2D PSD of Eq.(3) and on the right its 1D representation as a cut at $y = 256$ pixel. $\lambda = 879.7 \cdot 10^{-6}\ mm$, $a = 1.5\ mm$, $f = 2280\ mm$, pixscale $= |\frac{1}{0.810612 \cdot 512}\ \frac{arcsec^{-1}}{pixel}|^2$, $B = 8\ mm$, $I_0 = 1$, $V = 1$, $theta = \frac{\pi}{2}$.

2 Experimental set-up and observations

To measure the diameter of Mars we used a double-circular-slits mask put at the top of the telescope Telesto (TELescope for Science, Teaching and Outreach).

2.1 Telescope

Telesto is a 60 cm diameter, reflecting telescope with a focal ratio of 3.8. The tracking system is automatic for both the telescope and the dome. This means that once the celestial object is set, the telescope and the dome follow the object for the entire duration of the exposure time. Figure 5 shows the control system of TELESTO, where the red line indicates the path of the light. Firstly the light passes though a motorized flip mirror which decides whether to direct the light towards the eyepiece or to the digital detector (our case). Before the digital detector there is a wheel with the filters for the light. The filters are the Johnson-Cousins filters UV, B, V, R, I. Their curves are shown in the Figure 6. After the filter, the light goes to the CCD detector. It is a FLI PL16803 detector, which has different features:

- Camera reading speed: 1 Mhz or 8 Mhz (minimum reading time $\sim 2.1\ s$)
- 4096 x 4096 pixels array

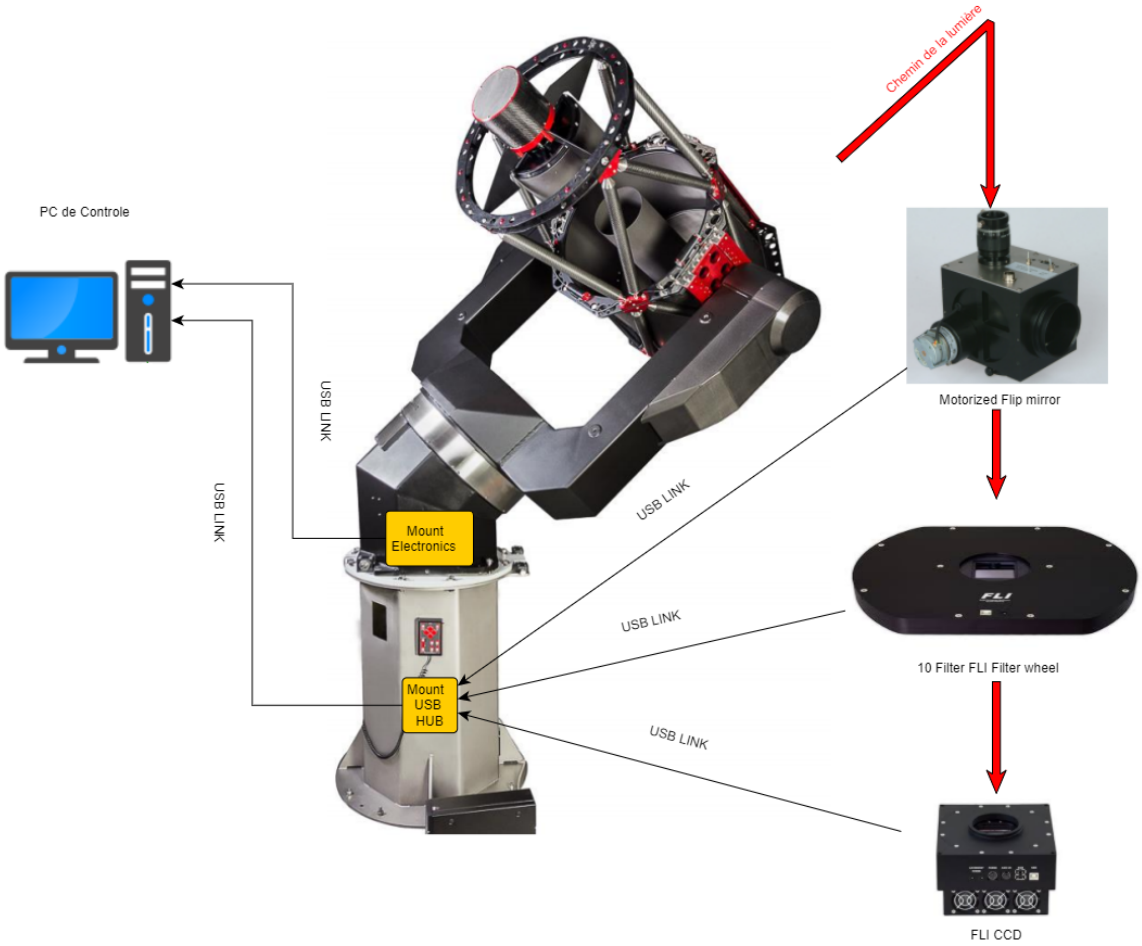

Figure 5: TELESTO control system

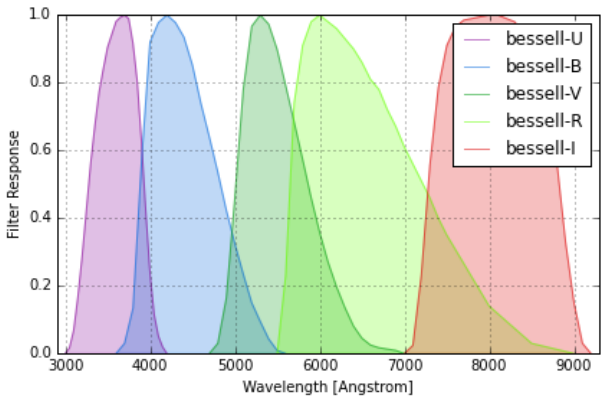

Figure 6: Johnson-Cousins UBVRI filter curves

- Pixel size: 9 μm
- Plate scale: 0.810612 $arcsec/pixel$: calculated through the site *Astrometry.net* [a], which automatically calculates the plate scale of a telescope just analysing an image that contain several stars.

The telescope is mainly controlled eletronically with the Astrometric Telescope Control System (ATCS), which connects the PC to: the motorized flip mirror, the wheel with the filters and the camera. The resulting images are saved in file.fits and the analysis is done through python.

2.2 Masks

Concerning the masks, firstly I built rigid cardboard masks with the help of the workshop of the department. The pinholes have a diameter of 2 and 4 mm and the baselines are: 3.5, 6, 8, 10, 12, 16, 20, 24 and 28 mm. Afterwards, to have more precise masks, we requested to the workshop of the department to build aluminum masks with pinholes of 3 mm as diameter and baselines of: 4.5, 6, 8, 10.5, 13.5, 17, 21 mm. The baselines and the diameters of the masks were chosen by studying the size of Mars a priori and then assuming its visibility curve as known. The Figure 7 shows some of the rigid cardboard masks and the Figure 8 the scheme of the Aluminium masks. Although the first masks were simple

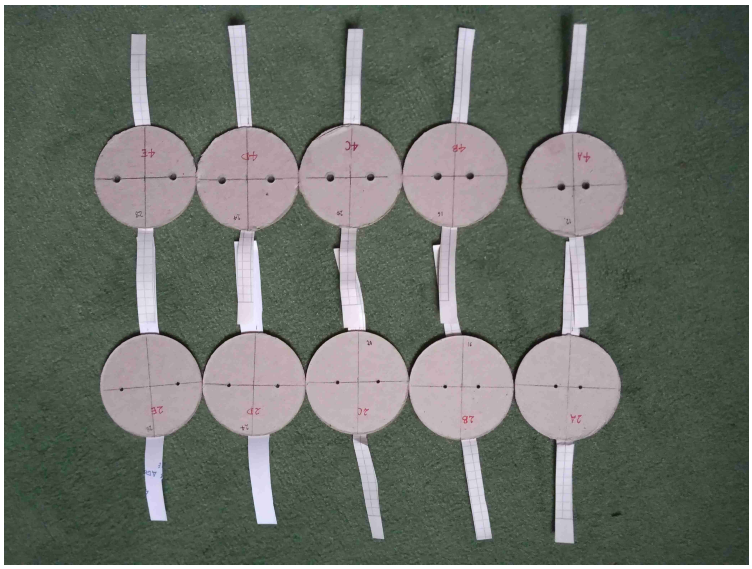

Figure 7: Rigid cardboard masks masks used for the experiment. The two pieces of paper on each mask are meant to help to remove the masks from the telescope lid.

and made by hand with a material not completely opaque to light, the interference pattern was clearly distinguishable in the images that we took. This is incredible considering the simplicity of the setup.

2.3 Observations

Theoretically, it would have been optimal to observe Mars the highest number of nights. Unfortunately, however, in this period of the year the clouds often prevent the observation of Mars. So it was decided to choose nights with few clouds and a clear sky and make the

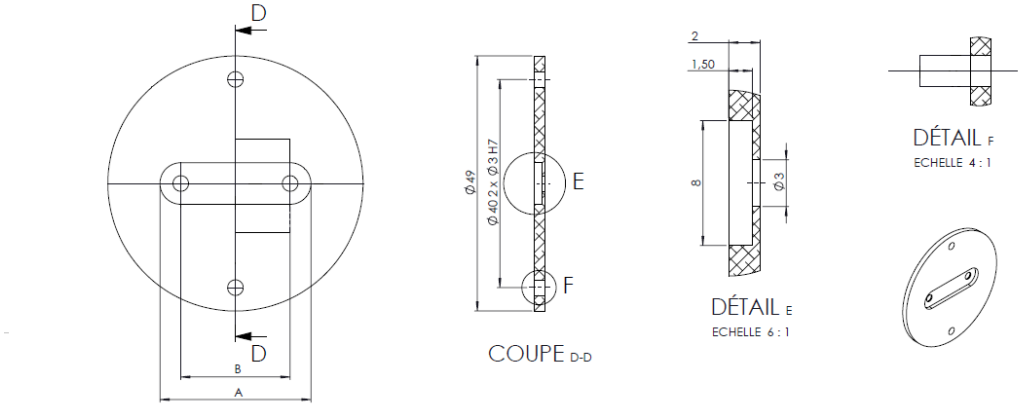

Figure 8: Aluminium masks scheme by the workshop of the department

observations accordingly. Moreover, as already said in the introduction, the trees around the observatory prevent observing objects below $20 - 25°$ of altitude (depending on the weather); therefore this factor also had to be taken into consideration.
We did in total 3 observations. The first was on the 8^{th} of October and we used the cardboard masks. There were some low clouds and Mars had an altitude less than 22°; then, we decided to observe a bright star (Vega) in order to be able to do the analysis on it in preparation for the analysis on the planet. The second observation was made on October 30^{th} and images of Mars and Altair were taken with the cardboard masks. We started the analysis in both the direct space and the Fourier space. On November 13^{th} we did the last observation of Mars and Altair with the aluminum masks.
In these last two observations we collected data from a star like Altair in order to calibrate the visibility of Mars. In fact, to be able to resolve the stars, they would be needed baselines of about ten meters. Then, with our small baselines the visibility of any star should be approximately constant and near to one. Therefore dividing the visibility of Mars with that of Altair for the same baseline would calibrate the results of the planet.

3 Data analysis

As I said before there are two ways to calculate the angular dimension of a planet: through an analysis in the direct space or an analysis in the Fourier space, let us see each of these specifically. The python codes that we used for the data analysis are shown in the appendix.

3.1 Direct space analysis

As already explained in the theoretical part, the analysis in the direct space consists in measuring the visibility for each baseline and then fit the obtained visibility with Eq.(5) to obtain the diameter of the planet. In particular, starting from the initial image of Mars

we identified the center of the image as the pixel with the maximum intensity and cut the image into a square of size 256 x 256 pixels (Figure 9a). Then, we substracted from the image the median of the image itself; considering that Mars occupies a small area of the CCD the median falls certainly into the background noise. So, it is like carrying out a first cleaning of the noise (Figure 9b).
At this stage, two different paths were followed: an analysis through simple fit, i.e. minimizing the chi square, or an analysis through a MCMC algorithm.

3.1.1 Simple fit analysis

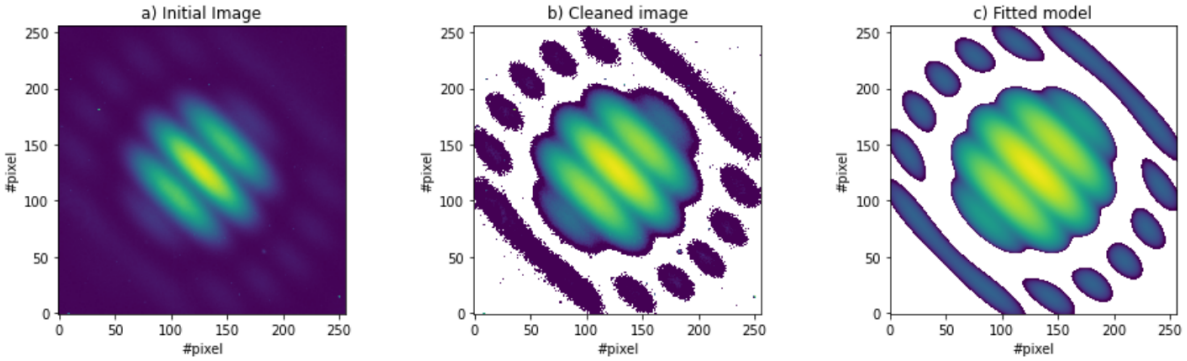

Figure 9: Example of the two initial steps of the analysis in the direct space (a & b) and an example of the simple fit result (c) for a 6 mm baseline with the aluminum mask.

After carrying out the first two steps described above, we continued by fitting the image with the model of Eq.(3) (Figure 9c). As free parameters we took: λ, θ, the visibility, the maximum intensity, the position of the center of the intensity pattern and the background. We have fitted the central position of the image because it could be at a position that is a fraction of a pixel and therefore the approximation that we did would not be exact. In addition we have also fitted the wavelength, although we know the effective wavelengths of our filters. This is because by fitting the image with a fixed wavelength we obtained inaccurate results, then we decided to leave it as free parameter. A possible explanation is given in the "Identified limitations" subsection. To minimize the χ^2, i.e. to find the best parameters of the fit, we took as errors the median absolute deviation (MAD). It is a more robust estimator of scale than the sample variance or standard deviation because it works better with distributions without a mean or variance as in our case. From the literature we know that [3]:

$$MAD = 1.4826 \cdot Median(|data - Median(data)|)$$

where *data* is the list of the values of each pixel. After the minimization of the chi square, we calculated the residuals, i.e. the difference between the observed image and the image obtained with the estimated values. This was done in order to identify eventual bad pixels.

In fact, another problem generally related to the telescope is the presence of random "hot" pixels and the image analysis should be performed ignoring these pixels. This is because these "bad-hot pixels" are not related to the intensity of light that passes through them, but simply to an error in the measurement and therefore counting these pixels as if they reproduced the signal would lead to wrong results. The criterion for selecting bad pixels was the following: if a pixel (in the residuals image) had an intensity greater than 5 times the intensity of the MAD, it was considered as bad. That is a good approximation for a bad pixel, because it is equivalent to considering a point farther than 5 times the standard deviation from the mean. For example, in the case of a Gaussian distribution we know that already at 3 sigma from the mean we include 99.7% of the data, so at 5 sigma we include practically all the signal.
The next step is the re-fitting of the image without taking into account the bad pixels selected previously. Finally we extract the visibility and the fitted wavelengths from this second minimization. Comparing the obtained fit and the starting image, it can be said that our model approximates the observations very well. An example is represented in Figure 10. Here the black curves represent a cut along the diagonal and along the horizontal line at the 128^{th} pixel of the image of Mars taken with the 4.5 mm baseline. The red dashed curves represent our fit, i.e. our model. The two curves are very close to each other, then our model is assumable as correct.

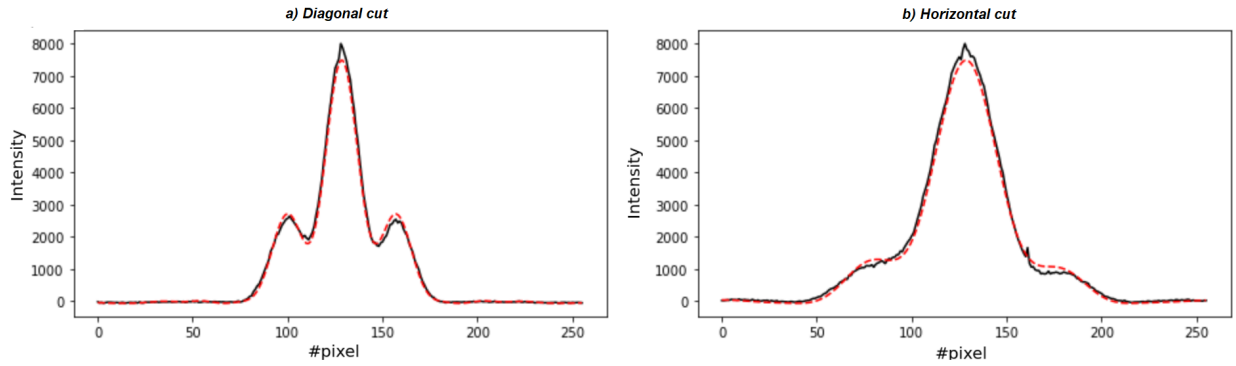

Figure 10: Black curves: observed data along a diagonal (a) or a horizontal cut (b). The data are those of Mars seen with the 4.5 mm baseline. Red curves: our fit

3.1.2 MCMC analysis

To carry out the analysis with the MCMC, a program seen in the course of Astrophysics & Data Science was used: SAM implemented by the professor Jean-Baptiste Delisle. As a first approximation, we took as initial values those obtained from the analysis with the simple fit, with the respective errors. Each chain consisted of 9600 iterations. After the calculation of the first chains, we took the averages and standard deviations of the last 2000

iterations for each of the 7 parameters considered previously and used the latter as initial values to restart the chain. All this is intended to ensure correct convergence of the chain. In Figure 11 is represented an example of the two chains for the visibility. The first chain is shown in Figure 11a and Figure 11b represents the new chain with the changed initial values. The success of convergence is clearly visible only in the second case. Once all the

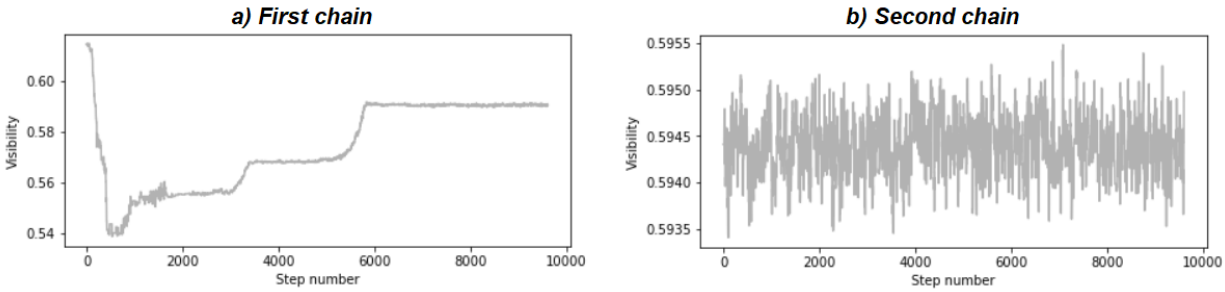

Figure 11: Example of the two chains with the MCMC analysis, for a 6mm baseline with the aluminum mask

converging chains have been obtained, we extracted the mean and standard deviation of the 9600 iterations for each parameter. In fact, all of the medians and means for each of the 7 parameters were practically the same (the difference between the mean and the median was far below one standard deviation). Furthermore, the distribution of the parameters has the form of a Gaussian, so taking the mean and the standard deviation to describe their distribution is a correct procedure. Figure 12 shows an example of the visibility distribution

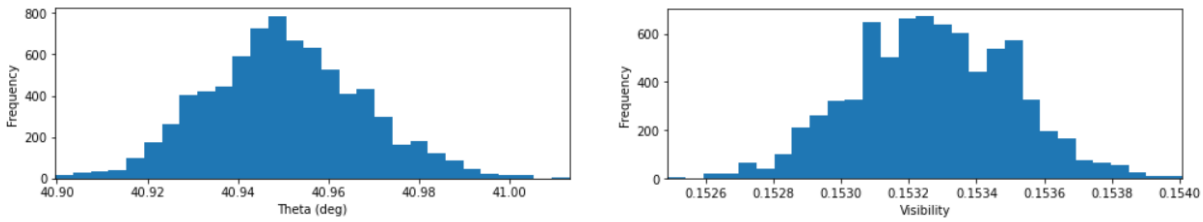

Figure 12: Example of the distribution of the visibility and λ for a 8mm baseline with the MCMC analysis.

and θ distribution for a baseline of 8 mm. It can also be guessed by eye that the two distributions are approximately Gaussians.

3.2 Fourier space analysis

As far as the Fourier space is concerned, the first steps of the analysis are exactly the same as those in the direct space, namely the cut and the cleaning. Then we proceed by calculating the square modulus of the fourier transform of the cleaned image, i.e. the PSD.

There are two methods to extract the visibility from the PSD. I will call them for simplicity: threshold method and constant radius circle method.

3.2.1 Threshold method

We took a threshold for the intensity so that setting to zero all the values of the PSD with less intensities than the latter, three independent figures can be distinguished. The threshold is chosen by eye so that the intensity is the minimum possible when the three cones are distinct figures. Even doubling this threshold, the results of the visibility remain within a 5% and therefore the approximation made by eye is reliable. An example to understand this analysis is shown in Figure 14.

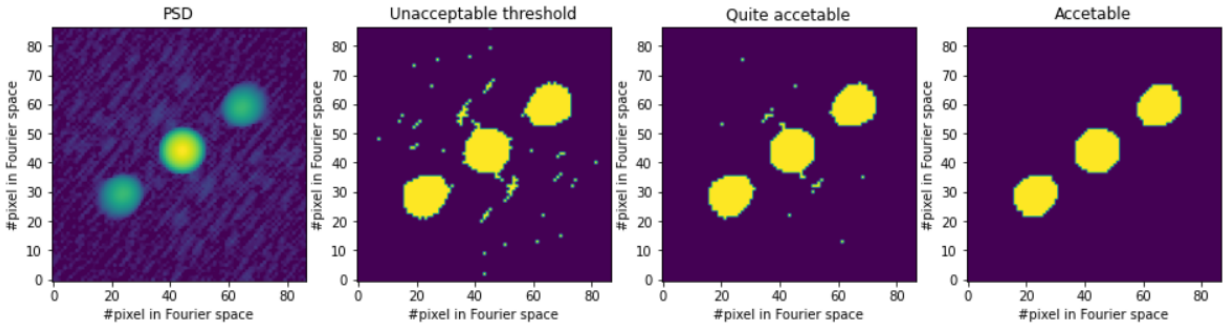

Figure 13: Example to choose the threshold

3.2.2 Constant radius circle method

This method consists first of all on taking a circle that totally covers the central cone. The next step is to reproduce the same circle to determine the area of one of the two external cones (the two external cones are identical). As centers of the two circles we chose the maximum intensity in each cone. An example is shown in Figure 14. The results with

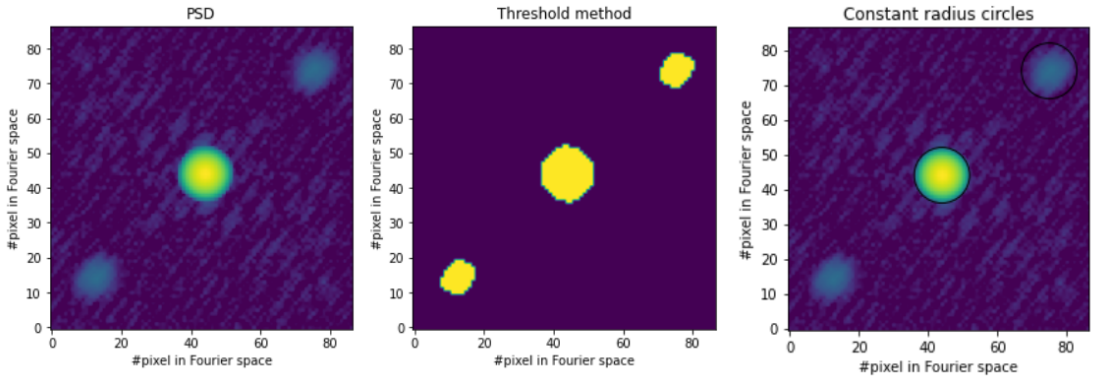

Figure 14: Example of the constant radius circle method

the constant radius circle method gives a visibility between 99% and 100% of the visibility

calculated with the intensity threshold method, therefore the two methods are practically equivalent.

For both methods in the direct space and in the Fourier space, once the visibility has been extracted the last step consists on extracting the diameter of Mars by the fit of Eq.(5).

3.3 Results

I will present the results of this last night of observation as they are the ones we have focused on the most and that have the most accurate masks. In the last observation we used exposure times of 300 seconds for both the star and the planet. Unfortunately Altair turned out to have a very low SNR (Signal to Noise Ratio): the peak of Altair was around 1400 ADU and the background was around 1000 ADU. This compromised the visibility results, which were by no means constant. Therefore it was decided to analyze Mars omitting the calibration with Altair.
Before showing the results it is fair to say that on the night of November 18, Mars measured 16.74 *arcseconds* in the sky from Versoix [b].

3.3.1 Simple Fit analysis

Baseline (mm)	Visibility	σ_V	λ (nm)	σ_λ (nm)	θ (deg)	σ_θ (deg)
4.5	0.5036	0.0094	844.42	3.38	43.803	0.019
6	0.5943	0.0112	818.22	3.28	44.888	0.310
8	0.1508	0.0076	826.53	3.46	40.889	0.362
10.5	0.1729	0.0097	821.10	4.1	53.948	0.470
13.5	0.0326	0.0072	820.75	3.87	48.553	1.503
17	0.0121	0.0081	818.36	4.38	46.369	3.571
21	0.0002	0.008	811.43	4.98	24.190	1.903

Table 1

The results of the visibility, λ and θ through the simple fit analysis are shown in Table 1. To find the angular diameter of Mars, it was used the average wavelength of those find above, i.e. 822.97 nm. Fitting the visibility the result turns out to be: 22.67 ± 0.09 *arcseconds* with a reduced chi square of 124.54. The plot of the visibility obtained for this particular angular diameter is represented in Figure 15a. The "theory" curve represents the visibility of Mars taking its true angular diameter (16.74 *arcseconds*).

3.3.2 MCMC analysis

The results of the visibility, λ and θ through the MCMC analysis are shown in Table 2. Using the MCMC to calculate the angular diameter of Mars we obtained a value of

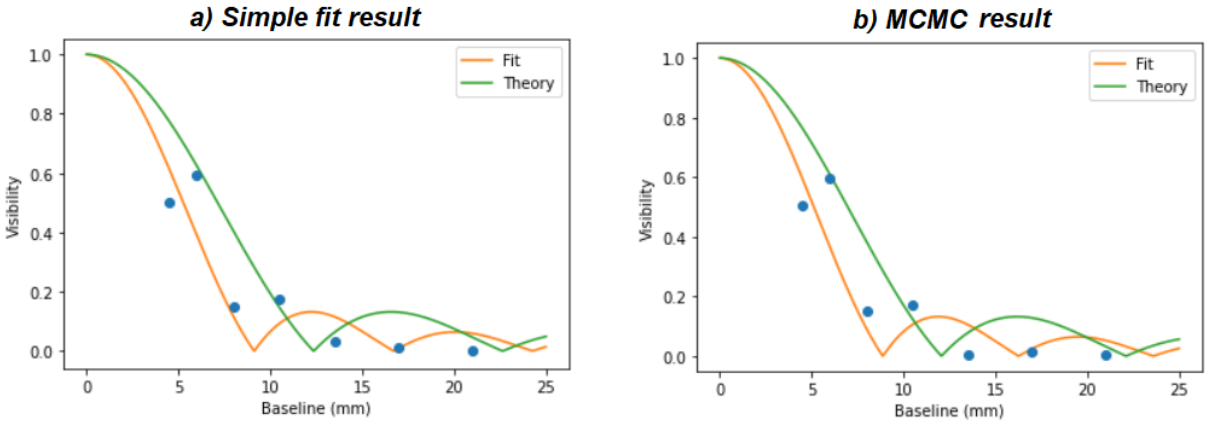

Figure 15: Results of the visibility for the angular diameter calculated with the simple fit method (a) and with the MCMC method (b)

Baseline (mm)	Visibility	σ_V	λ (nm)	σ_λ (nm)	θ (deg)	σ_θ (deg)
4.5	0.5046	0.0003	845.89	0.11	44.110	0.010
6	0.5944	0.0003	817.63	0.08	44.913	0.007
8	0.1533	0.0002	827.80	0.12	40.952	0.018
10.5	0.1739	0.0002	821.32	0.10	54.051	0.012
13.5	0.0022	0.0002	818.89	0.13	-5.381	1.192
17	0.0126	0.0002	817.04	0.12	46.252	0.100
21	0.0052	0.0003	810.69	0.13	50.700	0.190

Table 2

22.7511 ± 0.0033 *arcseconds*. The MCMC chain with 9600 iterations is shown in Figure 16a and the distribution of the angular diameter is shown in Figure 16b. It can be noted that also in this case the distribution is approximately a Gaussian. The representation of the visibility in that case is shown in Figure 15b.

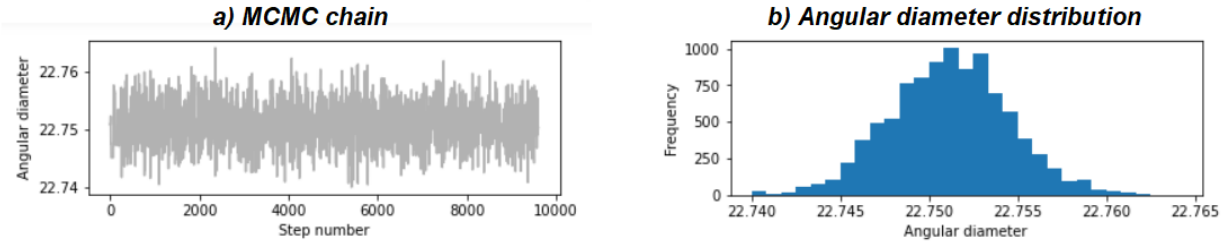

Figure 16: MCMC chain result and angular diameter of Mars distribution

3.3.3 Fourier space

In the Fourier space for the shortest (4.5 mm) and the longest (21 mm) baselines it was impossible to obtain three distinct circles in the fourier space, then we decided to ignore them. Furthermore, it was not correctly established how to identify the wavelength and the

errors. Therefore the results are simply limited to the calculation of the visibility with the threshold method:

Baseline (mm)	6	8	10.5	13.5	17
Visibility	0.597	0.166	0.177	0.038	0.017

4 Conclusions and perspectives

The results of the two direct methods are quite similar except for the 13.5 mm baseline. This can also be seen from the two orange curves in Figure 15, which are practically identical. The results are slightly distant from the theoretical value, but it is nevertheless amazing how spatial interferometry can give these precise results with such small apertures. The results of visibility in the Fourier space are also similar to those obtained in the direct space and this is a confirmation of the correctness of both the analysis.

4.1 Identified limitations

The biggest identified problem is the high reduced chi square in the case of the simple fit analysis. This can be due to different limitations. Let us show them comparing our different results:

- At the largest baseline (21 mm) there is a problem related to the analysis through simple Fit. θ turns out to be completely different from the previous ones. This is quite paradoxical: it is true that the angle was chosen "by eye", but still trying to keep the same orientation. Then 20 degrees of difference are not realistic. Similarly there is a failure of the analysis with regard to theta in the case of the analysis by MCMC for a baseline of 13.5. A possible solution could be to set limits for the fit of the data, preventing the analysis from giving wrong results. This is not as simple as I said and it would require a more in-depth study.

- In reality, the tracking of Telesto is not perfect and for long times of exposure there can be negative consequences on the data taking, due to the non-continuous, but discretized motion of the telescope. These consequences consist of a "trail" of photons following the discrete movement of the telescope. Then, the the Airy disk would turn out to be elongated along an axis. The tracking does not affect the data for a time of exposure of approximately less than one minute. For longer exposure times there is a shift of the interference fringes. This causes a systematic decrease of the visibility and it could explain the fact that most of our visibility results are inferior to theoretical ones (Figure 15).

- We actually considered Mars to be a uniform and circular source of light. This is partly wrong because Mars is not actually a circular and uniform disk. As an example just look at Figure 17. On the top left there is a photo of Mars taken on November 18^{th} with a single aperture of 50 mm in diameter. On the bottom left the yellow figure represents the pixels that have an intensity between 5% and 100% of the maximum intensity of the top left figure. On the right the images show different cuts of Mars (horizontal, vertical and along the diagonals). It can be clearly seen that Mars is not

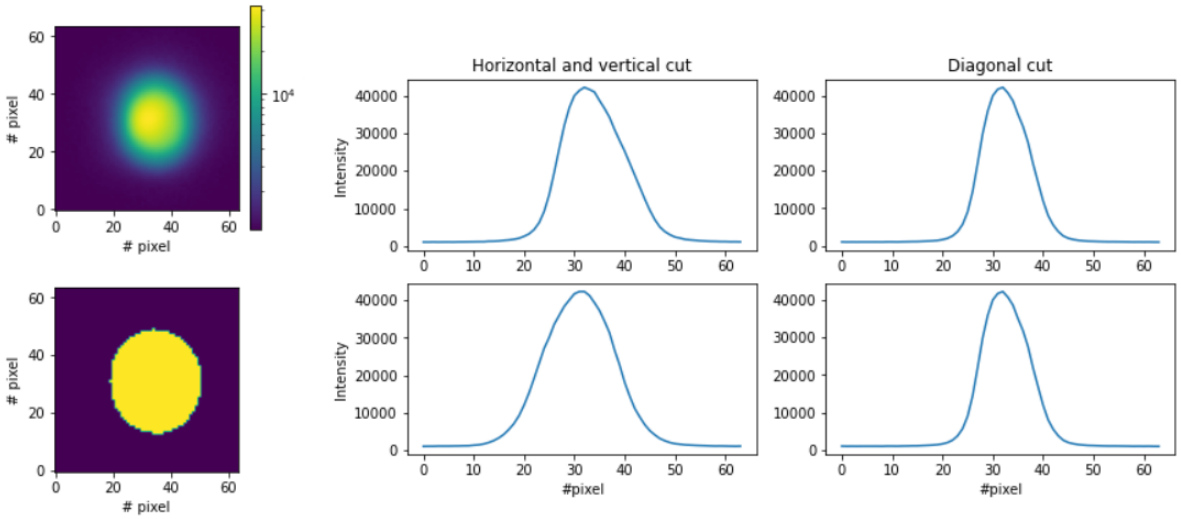

Figure 17: On the top left Mars seen with a single aperture of 50 mm in diameter. On the bottom left is shown the mask with the pixels having an intensity higher than 5% of the intensity peak. On the right the four images represent the intensity for different cuts of Mars

circular and that it does not have a uniform intensity. Therefore our model for the visibility is not completely correct. One method to find a better model is to use a mask with multiple holes, as I will explain in the next subsection.

- Linked to the previous point is the fact that the angle of inclination of the baseline cannot be fixed. If it was possible to fix this latter we could calculate the real diameter of Mars along this angle and then verify whether the experimental result is actually more similar to the real one. One way to fix the angle, for example, could be to draw reference lines in the hole where the masks are placed. In this way we could, with a good approximation, place the baselines always at the same angle.

- A limit, which can be easily overcome, is that of the calibration with a star. In our case, the SNR of Altair was too low, so it would be sufficient to select a brighter star and the problem would be solved.

- One limit concerns the error estimation. They are very low and probably underestimated. In the graphs of Figure 15 the error bars are not visible as their values are

very low. To overcome this impediment it would be necessary to revise the program that was used to fit the parameters. For the MCMC analysis we could chose not to take the error of a Gaussian distribution and change the method of choice.

- The last point concerns the wavelength. It would be helpful to find a way to determine accurately the effective wavelength for each filter. The CCD response should be studied in conjunction with the effective curve of the Jhonson-Cousins filters and the spectral curve of Mars. Furthermore, the wavelengths obtained from the fits and the MCMC analysis are very variable. This may be due to the transmission / absorption properties of the atmosphere. The latter may vary slightly over time causing slight variations in the wavelength.

4.2 Forseen improvements

Besides improving the points listed above there are some other possible works that can be done. We could apply the analysis both in the direct space and in the Fourier space to different planets of the solar system. Another interesting work that could be applied is to make masks with more than two holes. The resulting image would show the interference between each pair of holes. This could be a remedy for the model of Mars. We considered it as a uniform and circular disk, but we have seen in the limitations that this assumption is incorrect. By using a mask with several holes and by implementing some dedicated algorithms it would be possible to reconstruct the image of Mars and to find a better model. Out of curiosity, we tried to build some cardboard masks with three holes, positioned on isosceles or equilateral triangles. We got images like the ones shown in Figure 18. However, we did not dwell on the analysis of these last images, but it would be interesting to study them, especially in the Fourier space. Of course, masks with more than three holes could also be used and the analysis, although if more complicated, would lead to even more accurate results.

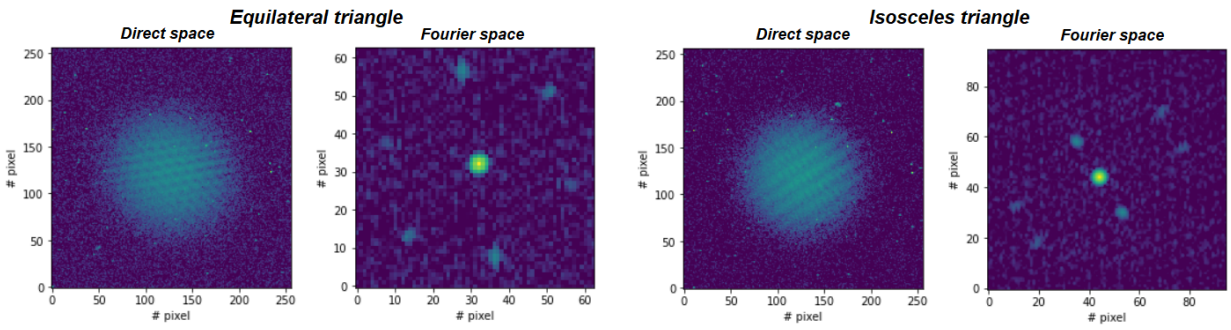

Figure 18: Mars in direct or Fourier space observed with cardboard masks with 3 holes placed on an equilateral triangle (left) or on an isosceles triangle (right)

5 References

Bibliography

[1] A. Glindemann, Introduction to Spatial Interferometry ESO Garching.

[2] A. Richard ThompsonJames M. MoranGeorge W. SwensonJr., Interferometry and Synthesis in Radio Astronomy, Springer International Publishing, 1st ed., 2017

[3] C. Leyset al., Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median, Journal of Experimental Social Psychology, 2013.

[4] D. Rudolf, J. Bussmann,M. Odstrcil,M. Dong, K. Bergmann, S. Danylyuk, L. Juschkin, Interferometric broadband Fourier spectroscopy with a partially coherent gas-discharge extreme ultraviolet light source, Optics Letters, 2015

[5] E. Hecht and A. Zajac, Optics, 4th ed. (Addison-Wesley, Reading, MA, 2003).

[6] F.A. Merchant, A.Bartels, C.BovikKenneth R.Diller, Handbook of Image and Video Processing, 2nd ed., 2005.

[7] J.M. Mariotti, Introduction to fourier optics and coherence, Lecture at the Observatoire de Paris-Meudon.

[8] M. Born and E. Wolf, Principles of Optics, 7th ed., Pergamon Press, 2002.

[9] P.Mazzoldi, M.Nigro, C.Voci, Fisica Vol.II Elettromagnetismo-Onde, 2nd ed., Edises, 2002.

[10] S. Divitt, J. Z. Lapin, L. Novotny, Measuring coherence functions using non-parallel double slits, Optical Society of America, 2014.

Sitography

[*a*] http://astrometry.net/
[*b*] https://theskylive.com/

Appendix

Below are shown the python codes used for data analysis.

```
import scaledAdaptiveMetropolis as sam
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.colors import LogNorm
from scipy import special
from astropy.io import fits
import scipy.optimize as opt
from astropy.nddata import Cutout2D
from scipy.optimize import minimize
```

Simple Fit analysis

```
def PSF1(params,extra):
    lbdEff_mm,theta_deg, V, Imax,dx_as,dy_as,bg_adu = params
    Dhole_mm, Baseline_mm,imSize,pixscale = extra
    theta_rad= theta_deg/180*np.pi
    x = np.arange(imSize)-imSize/2
    x_as = pixscale * x[:,np.newaxis]
    y_as = pixscale * x[np.newaxis,: ]
    r_as = np.sqrt((x_as-dx_as)**2 + (y_as-dy_as)**2)
    fcutHole_asm1 = Dhole_mm/lbdEff_mm * np.pi / (180*3600)
    arg = np.pi*fcutHole_asm1 *r_as
    arg[arg==0]=1e-6#protect against division by zero
    psf =(2*special.jv(1, arg)/arg)**2
    fBaseline_asm1 = Baseline_mm/lbdEff_mm * np.pi / (180*3600)
    du_asm1 = fBaseline_asm1*np.cos(theta_rad)
    dv_asm1 = fBaseline_asm1*np.sin(theta_rad)
    q=2*V/(1+V)
    fringes = 1 + q/2. * (
        np.cos(2*np.pi*((x_as-dx_as)*du_asm1+(y_as-dy_as)*dv_asm1)) -1. )
    image=Imax*psf*fringes+bg_adu
    return image

def chi2(params,noisyImage,sigmaMap,extra, badPixelList=[]):
    model_image = PSF1(params,extra)
```

```
    residualMap = noisyImage-model_image
    squaredNormalizedResidualMap = (residualMap/sigmaMap)**2
    if len(badPixelList)>0:
        squaredNormalizedResidualMap[badPixelList]=0.0
    chi2 = np.sum( squaredNormalizedResidualMap.flatten() )
    return chi2

file_name='3I _Slot 1_Slot 0__300,000secs_NoTarget00016885.fits'
image_data = fits.getdata(file_name)
position = (2065,2060)
imSize = 256
cutout = Cutout2D(image_data, position, (imSize,imSize))
im=cutout.data
#subtract background
im = im - np.median(im)

#estimation of the detector readout noise
#use median absolute deviation
detRon = 1.4826*np.median(np.abs(im-np.median(im)))
extraFirstObs= 3., 4.5,256,0.810612 #Dhole_mm, Baseline_mm,imSize,pixscale
initial = np.array(([845.8e-6,44.110, 0.5045, 7662.93,1.332,-0.497,-63.86]))
    #lbdEff_mm,theta_deg, V, Imax,dx_as,dy_as,bg_adu
print(chi2(initial,im, detRon *np.ones( (imSize,imSize)),extraFirstObs) /
    (imSize*imSize-7))

soln = minimize(chi2, initial, args=(im, detRon*np.ones(
    (imSize,imSize)),extraFirstObs))
imRes = im - PSF1(soln.x,extraFirstObs) #residuals Observed - Modeled

imResMed = np.median(imRes)
imResMad = np.median(np.abs(imRes-imResMed))
madClip=5
good=np.abs(imRes)<=(madClip*1.4826*imResMad)
sigmaEqNormalDist = 1.4826*imResMad
badPixel=np.abs(imRes)>(madClip*sigmaEqNormalDist)
imResCleaned=imRes.copy()
imResCleaned[badPixel]=0.0
plt.imshow(np.abs(imResCleaned)**2,norm=LogNorm(1,10000), origin='lower')

soln = minimize(chi2, initial, args=(im, detRon**2 *np.ones(
    (imSize,imSize)),extraFirstObs, badPixel))
```

```
NbadPix=len(np.where(badPixel)[0])
paramStd=np.sqrt(np.diag(soln.hess_inv))

print('Number of discarded pixels : ',NbadPix)
print('chi2 = %7.2f / %7.0f'%(soln.fun, imSize*imSize-NbadPix-NparamFit))
print('chi2r = %5.2f '%(soln.fun/(imSize*imSize-NparamFit)))
print('')
paramStr=['lambdaEff [mm] =' ,' theta [deg] =','V = ', 'Imax       =','dx
    [arcsec] = ', 'dy [arcsec] = ','background [adu] = ']
for i,param in enumerate(soln.x):
    print('%-25s   %15.10f +- %11.10f'%(paramStr[i],param,paramStd[i]))
```

MCMC analysis

```
def PSF1(params,extra):
    lbdEff_mm,theta_deg, V, Imax,dx_as,dy_as,bg_adu = params
    Dhole_mm, Baseline_mm,imSize,pixscale = extra
    theta_rad= theta_deg/180*np.pi
    x = np.arange(imSize)-imSize/2
    x_as = pixscale * x[:,np.newaxis]
    y_as = pixscale * x[np.newaxis,: ]
    r_as = np.sqrt((x_as-dx_as)**2 + (y_as-dy_as)**2)
    fcutHole_asm1 = Dhole_mm/lbdEff_mm * np.pi / (180*3600)
    arg = np.pi*fcutHole_asm1 *r_as
    arg[arg==0]=1e-6#protect against division by zero
    psf =(2*special.jv(1, arg)/arg)**2
    fBaseline_asm1 = Baseline_mm/lbdEff_mm * np.pi / (180*3600)
    du_asm1 = fBaseline_asm1*np.cos(theta_rad)
    dv_asm1 = fBaseline_asm1*np.sin(theta_rad)
    q=2*V/(1+V)
    fringes = 1 + q/2. * (
        np.cos(2*np.pi*((x_as-dx_as)*du_asm1+(y_as-dy_as)*dv_asm1)) -1. )
    image=Imax*psf*fringes+bg_adu #one also adjust the background
    return image

def computeChi2FlatIm(params,extra, flatImageObs,sigmaMap, badPixelList=[]):
    ImModel = PSF1(params,extra)
    flatResiduals = flatImageObs - ImModel.flatten()
    squaredNormalizedResidualMap = (flatResiduals/sigmaMap)**2
    if len(badPixelList)>0:
```

```
        squaredNormalizedResidualMap[badPixelList]=0.0
    chi2 = np.sum( squaredNormalizedResidualMap.flatten() )
    return chi2

file_name='3I _Slot 1_Slot 0__300,000secs_NoTarget00016885.fits'
image_data = fits.getdata(file_name)
position = (2065,2060)
imSize = 256
cutout = Cutout2D(image_data, position, (imSize,imSize))
im=cutout.data
#subtract background
im = im - np.median(im)

#estimation of the detector readout noise
#use median absolute deviation to do so (avoid outliers)
detRon = 1.4826*np.median(np.abs(im-np.median(im)))
extraFirstObs= 3., 4.5,256,0.810612 #Dhole_mm, Baseline_mm,imSize,pixscale

flatImageObs=im.flatten()
sigmaMap=detRon *np.ones( (imSize,imSize))
flatSigmaMap=sigmaMap.flatten()

def logLikelihood4SAM(theta):
    params=theta[0:]
    chi2=computeChi2FlatIm(params, extraFirstObs, flatImageObs, flatSigmaMap)
    lnLike = -0.5 * chi2 - 0.5* np.sum(np.log(flatSigmaMap))
    return lnLike

#begin parameters
thetaInitAvg=np.array([845.8e-6,44.110, 0.5045, 7662.93,1.332,-0.497,-63.86])
thetaInitStd=np.array([3e-6,1,0.005,50,0.15,0.15,6])

#errors
covMatInit=np.identity(7)
for i in range(7):
    covMatInit[i,i]=thetaInitStd[i]**2

#Launch the MCMC
SamRawSamples=sam.sam(thetaInitAvg,logLikelihood4SAM,nsamples=32*300
    ,cov0=covMatInit)
```

```
SamRawSamplesParams = SamRawSamples[0]
SamRawSamplesLnLike = SamRawSamples[1]['logprob']

#Identify Maximum Likelihood and the corresponding set of parameters
MaxLnlikeSamSamples = np.max(SamRawSamplesLnLike)
MaxLnlikeSamSamplesParams=SamRawSamplesParams[SamRawSamplesLnLike==
     MaxLnlikeSamSamples,:][0,:]
SamCovMatParams=SamRawSamples[1]['cov']

labels = ["lbdEff_mm", "theta_deg", "V",'Imax','dx_as','dy_as','bg_adu']
for i in range(7):
    medVal=np.median(SamRawSamplesParams[:,i])
    meanVal=np.mean(SamRawSamplesParams[:,i])
    stdVal=np.std(SamRawSamplesParams[:,i])
    print('%10s  %18.12f   %18.12f  %18.12f '%(labels[i],medVal,meanVal,stdVal))
```

Fourier analysis

```
lbdEff_mm = 879.7e-6
Baseline_mm = 8
N=512
file_name='8I _Slot 1_Slot 0__300,000secs_Mars00016890.fits'
image_data = fits.getdata(file_name)
position = (2070,2077)
size = (N, N) # pixels
cutout = Cutout2D(image_data, position, size)
cutout_data=cutout.data
cleaned_data=cutout_data-np.median(cutout.data)
STD=np.std(cleaned_data[0:50,0:50])
print("The background of the detector is: {0}".format(STD))

unos=np.ones((N,N))
ftIm=np.fft.fft2(cleaned_data)
psdIm=np.roll(np.roll(np.abs(ftIm)**2,64,axis=0),64,axis=1)
psd_abs=np.absolute(psdIm)
plt.figure()
plt.imshow(psdIm[20:107,20:107],
    norm=LogNorm(np.median(psd_abs[0:10,0:10].flatten()),np.max(psd_abs)),
    origin='lower')
plt.suptitle("PSD of cleaned image", size=20)
```

```
mask=np.ones((N,N))
mask[psdIm<np.median(50*psd_abs[0:10,0:10].flatten())]=0.
mine=mask[20:107,20:107]
plt.figure()
plt.imshow(mine,origin='lower')

position1 = (62,63)
size1 = (25, 25)   #pixels
cutout1 = Cutout2D(mine, position1, size1)
cutout1_psd=Cutout2D(psdIm[20:107,20:107], position1, size1)
new1=cutout1.data*cutout1_psd.data
right_data=new1.flatten()
sum_right=np.sum(right_data)
print("the top right integral value is {0}".format(sum_right))
plt.figure()

position3 = (44,44)
size3 = (16, 16)   #pixels
cutout3 = Cutout2D(mine, position3, size3)
cutout3_psd=Cutout2D(psdIm[20:107,20:107], position3, size3)
new3=cutout3.data*cutout3_psd.data
center_data=new3.flatten()
sum_center=np.sum(center_data)
print("the central integral value is {0}".format(sum_center))
plt.figure()
print("The two ratio are the same and are equal to
    {0}".format(2*np.sqrt(sum_right/sum_center)))
```

Mars diameter calculation

```
#Simple fit method

def Visibility(alpha_arcsec,Baseline_mm, lbdEff_mm):
    arg=(np.pi*Baseline_mm*alpha_arcsec)/(lbdEff_mm*206265)
    V= np.absolute(2*special.jv(1, arg)/arg)
    return V

def chi2(alpha_arcsec,Baseline_mm,lbdEff_mm,Data,sigmaData):
    model_image = Visibility(alpha_arcsec,Baseline_mm,lbdEff_mm)
```

```
    residualMap = Data-model_image
    squaredNormalizedResidualMap = (residualMap/sigmaData)**2
    chi2 = np.sum(squaredNormalizedResidualMap.flatten())
    return chi2

Data=[0.5036,0.5943,0.1508,0.1729,0.0326,0.0121,0.0002]
Baseline_mm=np.array([4.5,6,8,10.5,13.5,17,21])
sigmaData=[0.0094,0.0112,0.0076,0.0097,0.0072,0.0081,0.008]
alpha_arcsec=22
lbdEff_mm=879.7e-6
print(chi2(alpha_arcsec,Baseline_mm,lbdEff_mm,Data,sigmaData)/(6-1))
soln = minimize(chi2, alpha_arcsec, args=(Baseline_mm,lbdEff_mm,Data,sigmaData))
plt.errorbar(Baseline_mm,Data, yerr=sigmaData, fmt='o')
plt.plot(np.linspace(0.0,25,1000),Visibility(soln.x,np.linspace(0.0,25,1000),
    lbdEff_mm),label='Fit')
plt.plot(np.linspace(0.0,25,1000),Visibility(
    16.73873748,np.linspace(0.0,25,1000), lbdEff_mm),label='Theory')
plt.xlabel('Baseline (mm)')
plt.ylabel('Visibility')
plt.legend()
print('chi2 = %5.0f / %7.0f'%(soln.fun, 7-1))
print('chi2r = %5.2f '%(soln.fun/(7-1)))
print('Diameter = %5.2f +- %.2f arcsec'%(soln.x,np.sqrt(soln.hess_inv)))

#MCMC method
def computeChi2FlatIm(alpha_arcsec,Baseline_mm,lbdEff_mm, flatImageObs,sigmaMap):
    VisibilityTrue=np.zeros(7)
    for i in range(7):
        VisibilityTrue[i]=Visibility(alpha_arcsec,Baseline_mm[i],lbdEff_mm[i])
    flatResiduals = flatImageObs - VisibilityTrue
    squaredNormalizedResidualMap = (flatResiduals/sigmaMap)**2
    chi2 = np.sum( squaredNormalizedResidualMap)
    return chi2

flatImageObs=[0.5046,0.5944,0.1533,0.1739,0.0022,0.0126,0.0052]
sigmaMap=[0.0003,0.0003,0.0002,0.0002,0.0002,0.0002,0.0003]
Baseline_mm=np.array([4.5,6,8,10.5,13.5,17,21])
lbdEff_mm=[845.89e-6,817.63e-6,827.80e-6,821.32e-6,818.89e-6,817.04e-6,810.69e-6]
alpha_arcsec=np.array([22.75085825797424])

def logLikelihood4SAM(alpha_arcsec):
```

```
    chi2=computeChi2FlatIm(alpha_arcsec,Baseline_mm,lbdEff_mm,
        flatImageObs,sigmaMap)
    lnLike = -0.5 * chi2 - 0.5* np.sum(np.log(sigmaMap))
    return lnLike

#Launch the MCMC
covMat=np.identity(1)
covMat[0][0]=0.004169446578094459
SamRawSamples=sam.sam(np.array([alpha_arcsec]),logLikelihood4SAM,nsamples=32*300
   ,cov0=covMat)
SamRawSamplesParams = SamRawSamples[0]
SamRawSamplesLnLike = SamRawSamples[1]['logprob']

print(np.std(SamRawSamplesParams[:, 0]),np.mean(SamRawSamplesParams[1000:, 0]))
```