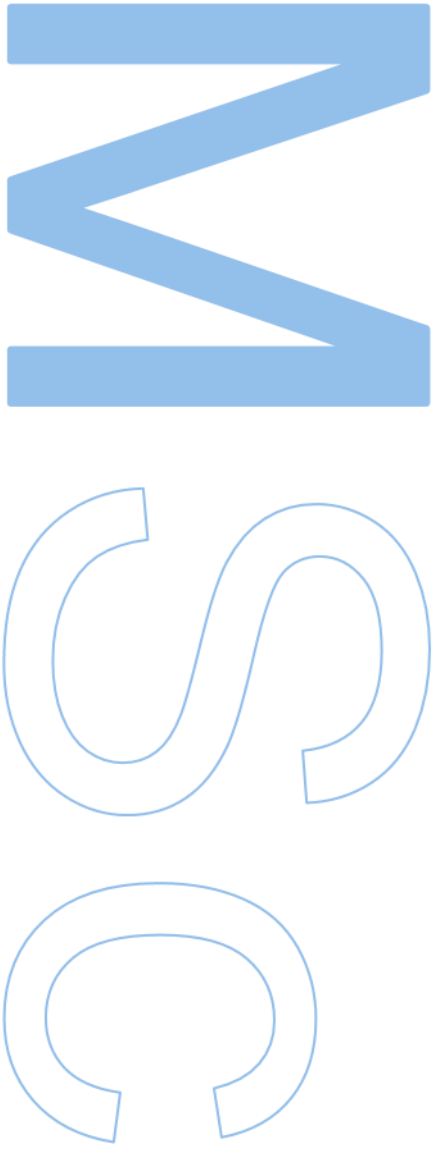
Looking at the Sun, finding other Earths: Identification of Solar Regions

José Luís Lino
Mestrado Integrado em Engenharia Física
Departamento de Física e Astronomia
2023

Supervisor
Nuno Santos, Faculdade de Ciências, Professor Auxiliar




U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

" It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change. "

Charles Darwin

Acknowledgements

Co-funded by the European Union (ERC, FIERCE, 101052347). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

First of all, I would like to thank my supervisor - professor Nuno Santos, for all the help during this thesis, how you guided me, gave me feedback and also for your patience for my questions. Also to my co-supervisor André Silva, thank you for all your help and patience too. You both helped me understand a lot more about this project and I hope you liked having me as your student. Thank you also for the colleagues from Coimbra, Teresa, Ricardo, João and Slava, that helped by providing the code that then I adapted and what approach to follow during the start of this thesis.

Secondly, a big thank you to my family! I want to thank all of you for all your support during this years. Thank you to my parents for how you educated me, for all your love, for my love of football from my dad and for my love with people, caring about them and trying to help everyone when they needed from my mum. I hope I can continue to make you proud of what I am achieving. Thank you also to my brother with your craziness and randomness, to show each other videos and memes to help during the bad times. Thank you also to my aunt, as my parents are living far away from me, you were the closest family member I could rely on. And finally to my godparents that helped me too, also educated me and taught me a lot of things I needed to go through with this thesis and masters.

Now to my friends, thank you all for your support when I needed, thank you for your patience with me, I know I can be a lot sometimes. Thank you to my 4 best friends from Madeira, Luís, Jéssica, Bea and Mouro. Thank you for always believing in me and for our calls playing games and just talking when I was feeling lonely, it helped me a lot. Thank you also for the group that was created on the first day of university with Bernardo, Duarte and Santos, it helped me a lot with my integration and with every difficulty I had during the early days. And finally, thank you to my closest friends from university Jorge, Anselmo and Cristina. I don't really know how to say thank you for everything, for your patience with my problems, for listening to me moan about things that you probably do not even care and for helping me keep pushing when I wanted to give up. I hope I could

make you proud, finishing this masters with everything that has happened. And also thank you Jorge for our endless talks about football since the first days of university, it helped me not think too much about problems I had, making bad days go by easily!

Last but not least, to my best friend Beatriz aka *Engenheira Gomes*, I don't really know how to thank you. You helped me to grow a lot this past years, taught me to care less about less important things and also helped me study a lot to end this masters. Thank you for your patience with me, for your craziness, for the late nights and early mornings working together and for all your company during car drives and lunches. I hope our friendship continues to grow, helping each other and hopefully it lasts a lifetime.

UNIVERSIDADE DO PORTO

Abstract

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

MSc. Engineering Physics

Looking at the Sun, finding other Earths: Identification of Solar regions

by José LINO

This thesis targets to develop a code for analyzing Sun images, identifying regions such as sunspots and faculae. The code complements with the PoET telescope and ESPRESSO spectrograph, collectively observing the Sun and capturing spectra from specific solar disk areas. Utilizing images from the Solar Dynamics Observatory, and converting them to TIFF files, our code can distinguish diverse solar regions, using thresholds. After the identification, a clustering method is used to count sunspots and determine their centroids. These centers go through a conversion between Cartesian to Celestial coordinates and obtains pointing coordinates for the telescope, such as altitude and azimuth. Also, a separation of umbra and penumbra is achieved. After all these identifications, area calculations were made, which validate sunspot areas, matching the combined area of umbrae and penumbrae. Finally, solar rotation velocity is determined, showing around 2 km/s near the Equator and decreasing toward 0 km/s near the Solar Poles. Comprehending more about these solar regions is crucial for understanding their impact on the Sun's spectrum and their role in exoplanet detection.

Keywords: Sun, Sunspots, Umbra, Penumbra, Facula, PoET Telescope, ESPRESSO Spectograph, Celestial Coordinates.

UNIVERSIDADE DO PORTO

Resumo

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

Mestrado Integrado em Engenharia Física

Observar o Sol, à procura de novas Terras: Identificação de Regiões Solares

por José LINO

Esta tese tem como objetivo desenvolver um código para analisar imagens do Sol, identificando regiões como manchas solares e fáculas. O código complementa com o telescópio PoET e o espectrógrafo ESPRESSO, observando coletivamente o Sol e capturando espectros de áreas específicas do disco solar. Utilizando imagens do Observatório Solar Dinâmico e convertendo-as em arquivos TIFF, o nosso código pode distinguir diversas regiões solares, usando limites de intensidade específicos. Após a identificação, um método de *clustering* é utilizado para contar manchas solares e determinar os seus centroides. Esses centros passam por uma conversão de coordenadas cartesianas para celestiais, obtendo coordenadas como altitude e azimute, para o telescópio apontar. Além disso, é feita uma separação entre a umbra e a penumbra. Após todas essas identificações, são feitos cálculos de área, validando as áreas de manchas solares, correspondendo à área combinada de umbra e penumbra. Por fim, a velocidade de rotação do Sol é determinada, obtendo-se cerca de 2 km/s perto do Equador e tendendo para 0 km/s perto dos Pólos Solares. Compreender mais sobre estas regiões solares é crucial para entender o seu impacto no espectro solar e o seu papel na deteção de exoplanetas.

Palavras-Chave: Sol, Manchas Solares, Umbra, Penumbra, Fácula, Telescópio PoET, Espectrógrafo ESPRESSO, Coordenadas Celestes.

Contents

Acknowledgements iii

Abstract v

Resumo vii

Contents ix

List of Figures xi

List of Tables xiii

Glossary xv

1 Introduction 1
- 1.1 Exoplanets and effects of stellar activity . . . 1
  - 1.1.1 Exoplanets . . . 1
  - 1.1.2 Methods to find exoplanets – Radial-Velocity . . . 3
  - 1.1.3 Methods to find exoplanets – Transit . . . 7
  - 1.1.4 Effects and Impacts from stellar activity . . . 8
    - 1.1.4.1 Stellar activity . . . 8
    - 1.1.4.2 Impact on exoplanet detection . . . 10
- 1.2 Our Star – The Sun . . . 12
- 1.3 PoET & ESPRESSO . . . 16
  - 1.3.1 ESPRESSO . . . 16
  - 1.3.2 PoET . . . 17
- 1.4 Thesis Motivation . . . 18

2 Methods 21
- 2.1 Code structure . . . 22
  - 2.1.1 Download of the images from SDO . . . 24
- 2.2 Image conversion . . . 28

3 Identification of Sun regions 31
- 3.1 Sunspots . . . 35
  - 3.1.1 Umbrae . . . 36
  - 3.1.2 Penumbrae . . . 37

3.2 Faculae 41

4 Coordinates transformation **43**
4.1 Converting Cartesian coordinates into celestial coordinates 43
4.2 Parallax correction 48

5 Results **53**
5.1 Identification of Sun regions 53
5.2 Verification of the coordinates transformation 57
5.3 Sun's rotation velocity 60

6 Conclusion **65**

A Appendix containing the codes used **67**

Bibliography **69**

List of Figures

1.1 Artistic representation of the planet Kepler-452b and its host-star. 2
1.2 Number of exoplanets discoveries across the years. 3
1.3 Wobble of a star caused by the gravitational pull of its orbiting planet. . . . 4
1.4 Wave of visible light difference coming from a star that is moving away from (red light) or towards (blue light) a viewer. 6
1.5 Diagram of the exoplanets' mass detected, with the Jupiter's mass as a reference. 6
1.6 Time evolution of the brightness with a drop in brightness caused by a planet passing in front of its host-star. 7
1.7 Aurora borealis created by Coronal Mass Ejections happening on the Sun. . 9
1.8 Representation of Solar Wind being repelled by Earth's magnetic field. . . . 10
1.9 Representation (not on scale) of the Solar System. 13
1.10 Intrinsic brightness vs surface temperature of the different types of stars, containing the Main Sequence where the Sun is part of. 14
1.11 Evolution of a Sun-like star. 14
1.12 Representation of the Sun's structure. 15
1.13 Sun's photosphere where the Umbra and Penumbra are represented. 15
1.14 Cycle of the sunspots with a prediction of the near future. 16

2.1 Image of the Sun with the HMI Intensitygram filter. 22
2.2 Fluxogram of the codes used. 23
2.3 Different available filters of HMI. 24
2.4 Different available resolutions of HMI Intensitygram filter in JPEG format. . 25
2.5 Parameters needed to download data in the FITS format. 26
2.6 FITS format file requested for import, with all the parameters completed. . . 27
2.7 FITS format file requested for import, with all the parameters completed. . . 27
2.8 Comparison of regions identified, umbra+penumbra on the left and umbra on the right. 29

3.1 Image of the attempt to identify the penumbra, with a lot of noise and the umbra mistakenly identified as well. 38
3.2 Clustering method to find the positions of the penumbra, with the umbra represented in green and the penumbra in blue. 40
3.3 Sun with the filter AIA (wavelength of 1700 Åon the left and the Faculae identified on the right. 41

4.1 Scheme used to calculated the Sun's radius angle on the sky, not on scale. . 44

4.2 Scheme of the orientation angle, with the Sun on the images on the left and the Sun on the sky on the right. . . . 47
4.3 Localization of SDO projected orbit, near the VLT observatory. . . . 48
4.4 Angle difference created between different of points of view coming from SDO and PoET. . . . 49
4.5 Angle difference created between different of points of view coming from SDO and PoET. . . . 50

5.1 Identification of sunspots obtained for resolutions 1024x1024 and 4096x4096, up and bottom, respectively. . . . 54
5.2 Each of the 6 individual sunspots that were identified using the image from 23rd of May 2023. . . . 55
5.3 Identification of the Umbra on the left and Penumbra on the right. . . . 55
5.4 Zoom in on an individual sunspot identifying the umbra on the left and the penumbra on the right. . . . 56
5.5 Graphics of Azimuth and Altitude evolution in a day in Porto, with the red line on the higher altitude of the Sun and the blue line in the Azimuth equal to 180º. . . . 58
5.6 Graphics of Azimuth and Altitude evolution in a day in North Pole, with the red line on the higher altitude of the Sun and the blue line in the Azimuth equal to 180º. . . . 58
5.7 Graphics of Azimuth and altitude evolution in a day in Chile, with the red line on the higher altitude of the Sun and the blue line in the Azimuth equal to 180º. . . . 58
5.8 Graphics of the orientation angle in a year in Porto, North Pole and Chile, from left to right. . . . 60
5.9 Evolution of sunspots from 13th to 17th of December 2013. . . . 61
5.10 Velocity of rotation of the Sun calculated and compared with the theoretical curve. . . . 62

List of Tables

2.1 Time taken, in seconds, for the code to process images in the different resolutions. 30

3.1 Parameters chosen for the two resolutions used on the images. 32
3.2 Table comparing how many seconds it takes for both codes to execute. . . . 37

5.1 Table with the characteristics of the sunspots such as x and y coordinates in the Sun and area in pixels. 57

Glossary

AIA	Atmospheric Imaging Assembly
AU	Astronomical Unit
CAUP	Center for Astrophysics of the University of Porto
CME	Coronal Mass Ejections
DEC	Declination
ESPRESSO	Echelle Spectrograph for Rocky Exoplanet and Stable Spectroscopic Observations
FCUP	Faculty of Sciences of University of Porto
FITS	Flexible Image Transport System
FOV	Field of View
GIF	Graphic Interchange Format
GMT	Greenwich Mean Time
HMI	Helioseismic and Magnetic Imager
ID	Identification
JPEG	Joint Photographic Experts Group
JWST	James Webb Space Telescope
NASA	National Aeronautics and Space Administration
PIL	Python Imaging Library
POET	Paranal solar ESPRESSO telescope
RA	Right ascension
RV	Radial Velocity
SDO	Solar Dynamics Observatory

TAI	International Atomic Time
TIFF	Tag Image File Format
UP	University of Porto
VLT	Very Large Telescope

This thesis is dedicated to my family and friends, for their love and support.

Chapter 1

Introduction

The universe has always been something fascinating for human beings from Ancient Greece to modern times. We have been watching the sky for centuries always wondering and asking questions such as "How did the Universe begin and how will it end?", "How did the known structures in the Universe evolve?", "Is the Universe finite?", "Are we alone in the universe?" (Allen and Lidström, 2016). Nowadays a lot of these questions are being worked on or theorized, but that led to even more questions that we are yet to understand. One of the most intriguing questions is that last one, as the immensity of the Universe makes it very likely that there is life on other worlds. So lately, to reply to this important question, we are looking at the sky, searching for planets where life may be present.

1.1 Exoplanets and effects of stellar activity

1.1.1 Exoplanets

An exoplanet (extrasolar planet) is a planet that is outside of the Solar System that orbits another star (or binary star) that is not the Sun. The first exoplanet that was detected orbiting a star similar to the sun happened in 1995 (Mayor and Queloz, 1995). There was an exoplanet that was found in 1988, but only confirmed in 2003 (Schneider et al., 2011). As of this date, there are around 5404 confirmed exoplanets in 3979 planetary systems with 859 with more than one planet[1]. The JWST (James Webb Space Telescope) was created to observe the first stars and the formation of the first galaxies. It also is

[1]Numbers found on this diagram http://exoplanet.eu/diagrams/ and represented as well on the graph from figure 1.1

characterizing the atmosphere of other planets and their environmental conditions, to search for potential of life in such planets (Kasting, 2021). Potential of life in a planet that is similar to Earth means that he checks the following characteristics:

1. Distance to the star and surface gravity of the planet are such that water can exist in the liquid form;
2. Stable environment to have a stable climate;
3. Suitable atmosphere with necessary gases like nitrogen, oxygen, and carbon dioxide;
4. Magnetic field to protect from solar radiation and cosmic rays.

These planets are called Earth-like planets and one of the best candidates is Kepler-452b - quoted to be Earth's Cousin (Jenkins et al., 2015). It was the first near-Earth-size planet (its mass is around 5 times bigger than Earth's, so it is also called Super-Earth) that was found within the habitable zone of a star that is similar to our Sun. In this case, that star is a sun-like star named Kepler-452 that is located in the constellation Cygnus, at around 1,800 light-years[1] from Earth and represented in figure 1.1[2].

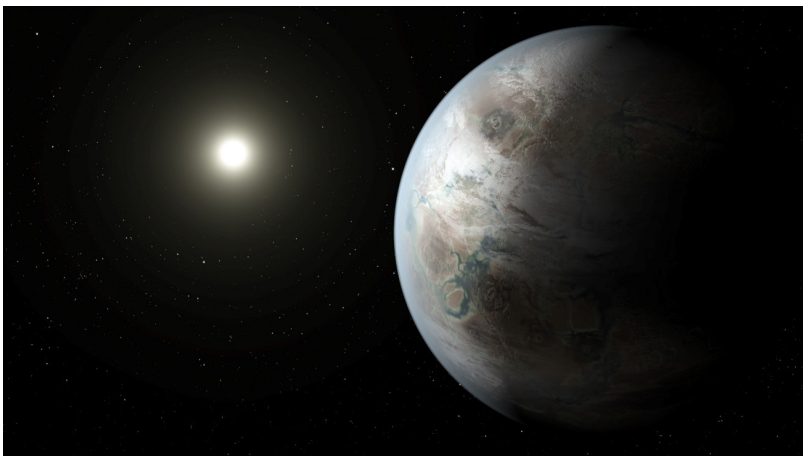


FIGURE 1.1: Artistic representation of the planet Kepler-452b and its host-star.

There are a lot of ways to look for and detect exoplanets, such as the Transit Photometry and the Doppler Spectroscopy, also known as the Radial-Velocity method, but there is also Direct Imaging and Gravitational Microlensing for example. Concerning the number

[1]Value obtained from The Habitable Exoplanets Catalog in https://phl.upr.edu/projects/habitable-exoplanets-catalog (Suárez Mascareño et al., 2022), with Kepler-452b being the first planet on the table Optimistic Sample of Potentially Habitable Exoplanets.

[2]Image credit: NASA Ames/JPL-Caltech/T. Pyle, taken on the 10th of June 2023.

of detections, the Transit and Radial-Velocity methods are the ones that have found the most planets, with the other two methods having smaller numbers, as we can see from figure 1.2[1].

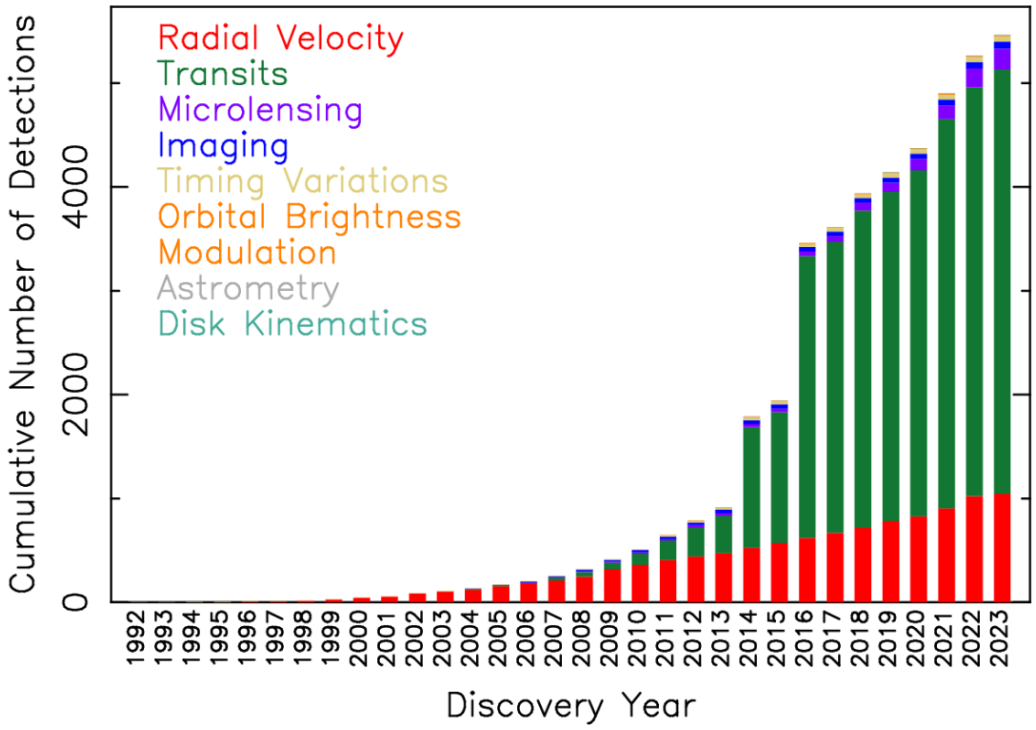


FIGURE 1.2: Number of exoplanets discoveries across the years.

The Radial-Velocity and the Transit Photometry methods, per the NASA numbers [2] as of today, have found 1036 and 4073 exoplanets, respectively.

1.1.2 Methods to find exoplanets – Radial-Velocity

The Radial-Velocity method consists of an indirect method based on the principles of the Doppler effect and gravitational interaction between the star and its orbiting planet (Santos et al., 2020). Although the star is more massive than the planet, there is a gravitational interaction between the two. This is represented by the gravitational force equation:

$$F = \frac{G(m_1 m_2)}{R^2} \tag{1.1}$$

Where G is the gravitation constant, $G = 6.674 \times 10^{-11} m^3 \cdot kg^{-1} \cdot s^{-2}$. The star has a greater gravitational pull, so the planet orbits around it, but the star also orbits around a

[1] Image taken from https://exoplanetarchive.ipac.caltech.edu/exoplanetplots/exo_dischist.png, on the 29th of June 2023.

[2] The number of planets found by these methods are presented in https://exoplanets.nasa.gov/alien-worlds/ways-to-find-a-planet/#.

small orbit in the center of mass of this system, also known as barycenter, as we can see in figure 1.3[1] [2].

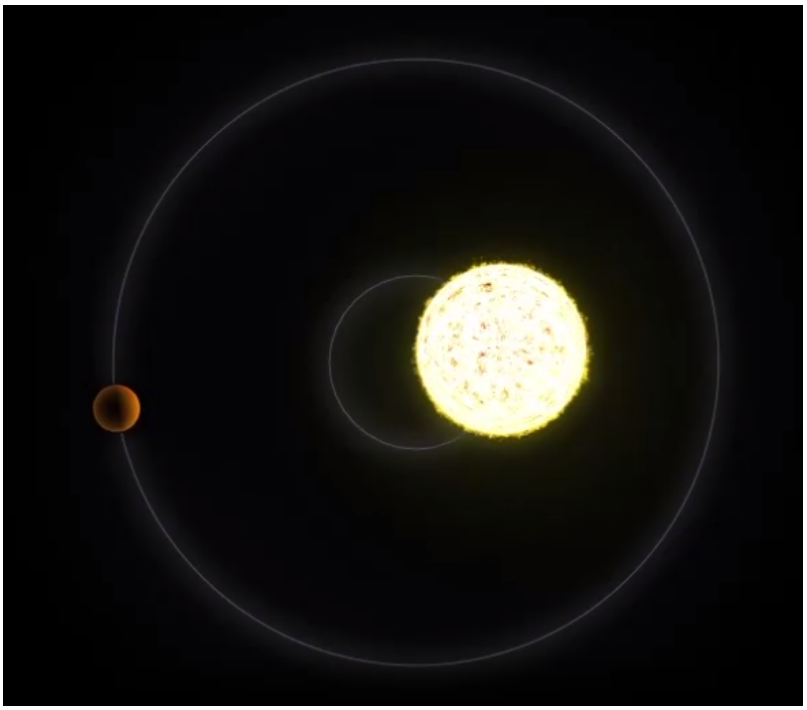


FIGURE 1.3: Wobble of a star caused by the gravitational pull of its orbiting planet.

This representation is exaggerated but if we observe the system along our line of sight, we will notice a wobble in the star, as it either appears to be moving in our direction, or away from us. By measuring this wobble, we can understand if a star has one or more planets and how big and close to the star they are. The radial velocity created by this wobble is given by the following equation:

$$v_{radial}(t) = v_0 + v_* \cdot sin(i) \cdot cos(\Omega_K t + \phi_0) \tag{1.2}$$

This equation considers that the orbit of the planet has an inclination i with respect to the line of sight and, for a circular motion, the semi-major axis of a planet around a star is a. v_0 is the radial velocity of the planetary system with respect to the observer and v_* is the orbital velocity of the star around the barycenter, with v_K being the Kepler velocity of the planet.

[1] Image taken from https://exoplanets.nasa.gov/alien-worlds/ways-to-find-a-planet/#/1, on the 11th of June 2023

[2] This explanation and figure where found in this NASA page https://exoplanets.nasa.gov/alien-worlds/ways-to-find-a-planet/#/1.

$$v_* = \frac{M_p}{M_* + M_p} v_K \tag{1.3}$$

Still regarding equation 1.2, the constant phase is represented by ϕ_0 and Ω_K is the Kepler frequency, described by:

$$\Omega_K = \sqrt{\frac{G(M_* + M_p)}{a^3}} \tag{1.4}$$

The time is represented in equation 1.2 by t, with the orbital period being:

$$P = \frac{2\pi}{\Omega_K} \tag{1.5}$$

For the same star, i.e., M_* is constant, a more massive planet (larger M_p) will have a bigger Kepler frequency and, consequently, will induce a greater radial velocity in its star. Considering this, planets such as Jupiter or Saturn, will have a wobble effect much stronger than Earth or Venus for example, as $M_{Jupiter} > M_{Earth}$. This wobble will vary the star's velocity, as seen in equation 1.2, impacting the frequency of light that reaches us. This impact is given by equation 1.6, explaining numerically the Doppler effect. The Doppler effect is an apparent variation in a wave's frequency in relation to an observer moving relative to the wave source (Giordano, 2012). This motion of the star causes its observed wavelength to shift the spectral lines in the spectrum of the star, according to the Doppler's equation:

$$\frac{\Delta\lambda}{\lambda_0} = \frac{v}{c} \tag{1.6}$$

Where $\Delta\lambda$ is the observed wavelength shift and λ_0 is the reference wavelength, related to the radial velocity of the star, v and speed of light, c. If we measure the wavelength shift in the star's spectral lines, we can determine the radial velocity of the star. This change in wavelength can be detected as a color shift in the spectrum, changing to blue or red depending on if the body moves closer or far away from us, and the wave stretches or squishes, respectively and represented in figure 1.4[1].

To summarize, the radial velocity method utilizes a physics concept - the Doppler effect, measuring the variation in the radial velocity of a star. This variations happen because of the gravitation interactions between the star and the exoplanet, changing the

[1]Image taken from https://upload.wikimedia.org/wikipedia/commons/e/e4/Redshift_blueshift.svg, on the 11th of June 2023

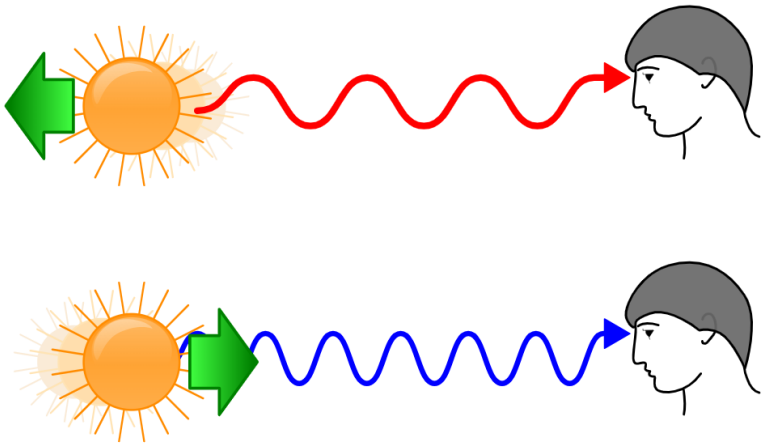


FIGURE 1.4: Wave of visible light difference coming from a star that is moving away from (red light) or towards (blue light) a viewer.

position of the star's spectral lines. This change is also named Doppler shift and it allows to infer the presence of any orbiting companions and also its characteristics.

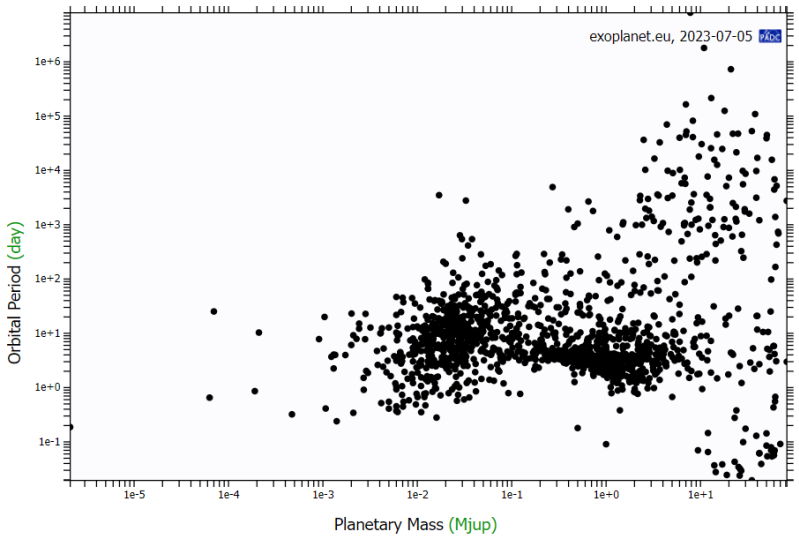


FIGURE 1.5: Diagram of the exoplanets' mass detected, with the Jupiter's mass as a reference.

For curiosity, a Jupiter-like planet measured radial-velocity induced in a star would be around 13 m/s. On the other hand, a Earth-like planet would only have about 9 cm/s of shift in the stellar spectrum (Santos et al., 2020). This is why there are much more exoplanets detected with a mass closer to Jupiter's mass, as we can see from figure 1.5[1]. Measuring this small change is a challenge that is being tackled in order to detect more exoplanets, as discussed further ahead.

[1]Plot obtained using the database from (Schneider et al., 2011), on the 7th of July 2023

1.1.3 Methods to find exoplanets – Transit

This transit method consists of observing the star's emitted light and watching it dim by a measurable amount when a body, in this case a planet, passes directly between the star and the viewer (Charbonneau et al., 1999). This event can be compared to an eclipse, when the Moon passes in front of the Sun, blocking its light, or even when Venus or Mercury pass in front of the Sun. To observe this, the star's light is measured and if a dim happens, that could mean that the star is being orbited by a planet, in this case an exoplanet. This dim can be measured in brightness over time, comprising the so-called light curve, as presented in figure 1.6[1].

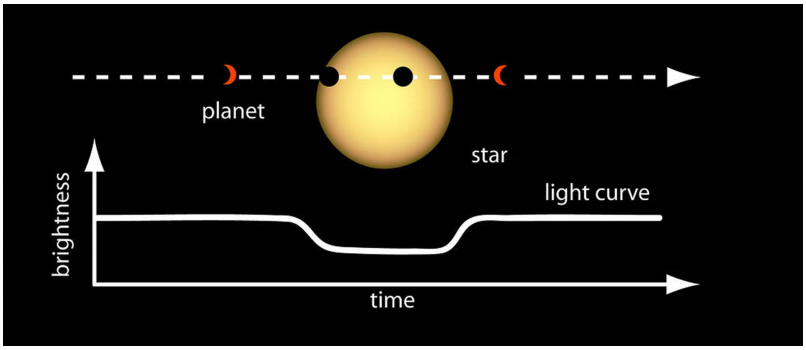


FIGURE 1.6: Time evolution of the brightness with a drop in brightness caused by a planet passing in front of its host-star.

The drop in brightness and time that it takes during that drop provide us with a lot of information about the transiting planet. This allows us to estimate the radius of a planet, given that we know the one of the star. We can also determine the period of rotation of the planet, i.e., the time it takes the planet to orbit its host star. These two values are important to compare this planet to Earth in order to find Earth-like exoplanets. However, the light-curve does not always present a clear dip in brightness, as it can change over time, with stellar activity also appearing in the curve. This stellar effects will be modeled, by adjusting the planet and star's signals, in order to analyse the light curve and consequently measure radial velocities (Baluev, 2015; Damasso and Del Sordo, 2017).

During the transit event, the amount of light blocked by a planet is related to the ratio between the radius of the planet and star. This relation can be described by the following equation:

$$\frac{\Delta L}{L} = \left(\frac{R_p}{R_s}\right)^2 \tag{1.7}$$

[1]Credit: NASA Ames, taken on the 11th of June 2023

Where the radius R_p and R_s correspond to the planet and the star, respectively. ΔL is the change in brightness of the star during transit and the total luminosity of the star is represented by L. This radius and brightness relation provides valuable insights about the size of an exoplanet. For example, considering a Jupiter-like planet orbiting a Sun-like star, the ratio between the radius is around 10%. Using the equation, we can see the ratio of radius is the square of the ratio of brightness, so the brightness would vary about 1%. We can also calculate the change a Earth-like planet would induce on brightness. The expected photometric variation would be around 100 parts per million (Santos et al., 2020). Measuring such small signals is a challenge that new instrumentation can also tackle, however this method is particularly difficult to long-period planets. A shorter orbital period is more likely to transit in front of its host star from our line of sight. Contrarily to that, a longer period planet, for example located at around 1 AU from its star, such as the Earth, is harder to find with this method, as it would require a large number of observations to cover its orbit.

In summary, the transit method detects exoplanets utilizing the principles of light and its interaction with celestial bodies. A periodic change in the brightness of a star is measured during the transit event and allows to determine size, period and other characteristics of an exoplanet. This is the method that has discovered the most exoplanets so far[1]. However, this changes in brightness may mean other things other than a planet orbiting the star, like for example stellar noise such as sunspots or eruptions (Czesla et al., 2009; Forgan, 2013; Aigrain et al., 2004).

1.1.4 Effects and Impacts from stellar activity

1.1.4.1 Stellar activity

There are a lot of phenomena and processes that happen on the atmosphere and surface of the stars. These events can affect and impact not only the star itself, but also its surroundings and consequently bodies that may be orbiting it. We want to study these activities and what consequences they may cause, for example, to an exoplanet nearby or even check if what we think is an exoplanet is a signal produced by stellar physics processes. Let's start by an event that will be more talked about on the next chapters, the sunspots. These darker and cooler regions on the photosphere are caused by magnetic

[1]The number of planets found by this method is shown in https://exoplanets.nasa.gov/alien-worlds/ways-to-find-a-planet/#.

activity (Bray and Loughhead, 1964). These regions may change a star's brightness and also contribute to CME (Coronal Mass Ejections) and stellar flares (solar flares, regarding the Sun), events that eject enormous amounts of energy creating geomagnetic storms that can affect an exoplanet. In the case of our planet, the Earth, it can disrupt satellites operations and also radio communications (Muller, 2014). Another event that can happen on a star is the Coronal Mass Ejections, that are a large-scale of eruptions that release magnetic fields and plasma from the star's corona (Webb and Howard, 2012). In case of the Earth, these CMEs can cause geomagnetic storms that might create the northern and southern lights, or auroras, as we know and can see in figure 1.7[1]. In the matter of stellar flares, these are intense and sudden releases of energy in the form of electromagnetic radiation. On Earth, these can affect our upper atmosphere, triggering ionization (Mitra, 1974) and disturbances or also create risks to spacecrafts and astronauts.



FIGURE 1.7: Aurora borealis created by Coronal Mass Ejections happening on the Sun.

Other events that we can observe in stars are stellar winds. Firstly, Stellar winds are streams of charged particles that are emitted by the stars. If a planet is particularly close to its host star, these winds may cause their atmosphere to strip away and potentially affect the ability to sustain life (Airapetian et al., 2020). In the case of the Earth, our magnetic field protects us from Solar winds as we can see in figure 1.8[2]. Secondly, Stellar outbursts are periodic events that happen to stars known as nova or supernova events.

There is also effects of granulation and plages, for example, that can affect different timescales in evolution of stellar activities. In the case of granulation, it is a convective phenomenon that occurs on the surface of Sun-like stars. This process leads to the formation of small convective cells - granules. On the other hand, we have plages, that are

[1]Image taken from https://upload.wikimedia.org/wikipedia/commons/f/fa/Church_of_light.jpg, on the 11th of June 2023

[2]Image taken from https://www.vectorstock.com/royalty-free-vector/solar-wind-diagram-vector-21409409, on the 11th of June 2023

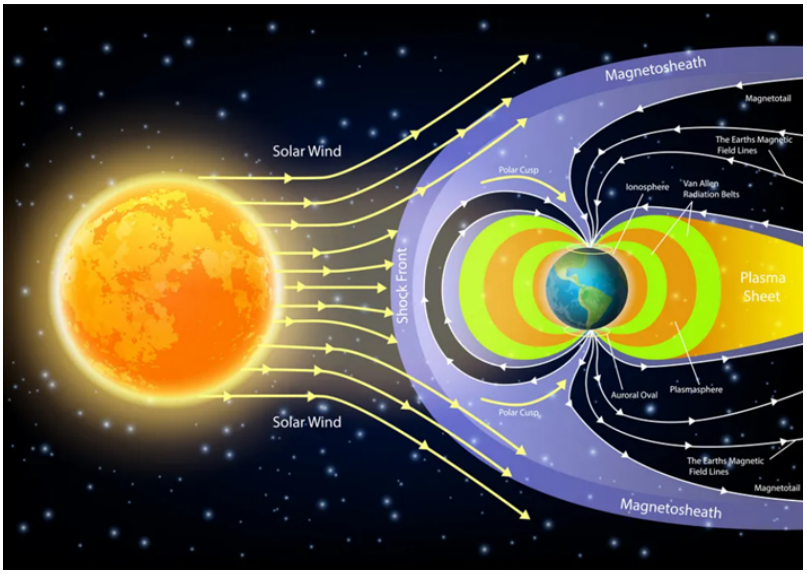


FIGURE 1.8: Representation of Solar Wind being repelled by Earth's magnetic field.

areas of increased brightness and temperature on the surface of a star and are created by magnetic activity.

All these events can be different depending on the star's size, age, temperature and other factors, but understanding them will help understanding stars and even our Sun better and in consequence understand exoplanets and what effects they can create on them.

1.1.4.2 Impact on exoplanet detection

Although these noises can affect stars and their exoplanets, they can also affect the detection of exoplanets. For instance, the characterization and detection of exoplanets are significantly influenced by the presence of stellar activity. These factors present adversities to two of the most used methods to find and characterize exoplanets: Radial Velocity and Transit Methods.

In the case of the Radial Velocity, the method relies on measuring the Doppler shift, caused by the gravitational pull coming from an orbiting planet, in the spectrum of a star. As we saw before, stars can have star spots and stellar flare, and these produce some variations on the position of the lines in the spectrum of the star. This variation makes it challenging to distinguish stellar noise from a signal created by an exoplanet (Serrano et al., 2018; Andersen and Korhonen, 2015; Vanderburg et al., 2016). Consequently, this leads to false detection and even show the presence of smaller planets that in reality might not exist. In the case of granules, they can cause variations in the radial velocity of a

star, affecting the precision of measurements in radial velocity, used to detect exoplanets. Granulation can induce variations in radial velocity from 10 to 400 cm/s depending on the star type and evolutionary stage (Dumusque et al., 2011; Schrijver and Zwaan, 2008), making it difficult to get precise measurements. To mitigate the impact of granulation, long exposure times can help, but low-frequency noise still remains on the spectrum. On the other hand, plages can also induce noise in radial velocity measurements because plages disrupt the flux balance between different parts of the star, causing an apparent Doppler shift that can be similar to radial velocity varying due to the orbit of an exoplanet. This is challenging, because we want to differentiate an actual planetary signal to effects of plages, so there are not any false detections. The presence of plages and convection in active regions contribute to variations of up to 40 cm/s during high activity phases (Dumusque et al., 2011; Meunier et al., 2010). Understanding and accounting the effects of granulation and plages can be crucial when measuring radial velocity data to detect exoplanets. Still regarding the Radial Velocity Method, the appearance and disappearance of sunspots on stars can produce radial velocity (RV) signals that are similar to the ones expected from exoplanets (Santos et al., 2014). The amplitudes of the RV signals imposed by stellar activity are often of the same order of magnitude as the expected signals of a low mass planet. Simulations show that the maximum RV signal induced by sunspots occurs when the spots are approximately close to disk center (around 45º degrees from the "meridian") (Santos et al., 2014). When the spot is at the "meridian", the value of the RV signal will decrease to zero. Therefore, the presence of sunspots can make it difficult to distinguish between RV signals caused by planets and those caused by stellar activity.

Regarding the Transit Method, as referred in 1.1.3, it consists on detecting exoplanets by measuring a slight dim on the brightness of a star when a planet passes in front of it. Stellar noise in the form of starspots and faculae, that are dark and bright regions, can imitate or hide, respectively, a signal from an exoplanet's transit. In addition, it can also make it harder to precisely obtain the planet's orbital period, size and other properties that are key to characterizing exoplanets.

The stellar noise that happens in stars can also cause false positives, where signals wrongly suggest the presence of an exoplanet. It can also produce false negatives, where stellar activity induces a missed presence of an exoplanet. These errors can lead to biases in our understanding of properties of exoplanets and impact their statistical analysis.

1.2 Our Star – The Sun

Since the start of this Section, we have been discussing the search and characterize exoplanets. Also, we have seen that the Stars themselves constitute a significant barrier on our ability to detect and study them. Thus, understanding the stars is critical to advance our analysis of exoplanets. However, it is simpler to study the closest star to Earth – our star, the Sun and by simpler we mean: On one hand, the impact of Radial Velocity that each planet induces in the Sun's spectrum is already known; On the other hand, we can also resolve the effect that small regions produce in the spectrum's lines - one of the objectives of PoET. Although there are a lot of types of stars, with various sizes and temperatures, we can study the Sun to obtain more knowledge on the physics behind the processes occurring on it. Afterwards, we can apply it to Sun-like stars, helping the study of exoplanets.

With this in mind, the Sun is the star at the center of the Solar System that the Earth orbits around, represented in figure 1.9[1]. It is the most important source of energy that allows life at Earth because it radiates that energy mostly as light, infrared and ultraviolet radiation. It is a nearly perfect ball of hot plasma (Fivian et al., 2008) and has nuclear fusion reactions happening in its core that heats to incandescence. The Sun has a radius that is 109 times bigger than Earth's (approximately 695,000 kilometers) and has a mass that is 330,000 times higher, holding about 99.86% of the Solar System's total mass (Woolfson, 2000). This mass consists of 73% of Hydrogen and 25% of Helium and the rest consists of oxygen, carbon, neon, and iron (Basu and Antia, 2008). It is located at an average distance of 1 astronomical unit from the Earth (1.496×108 kilometers) (Pitjeva and Standish, 2009) and was formed around 4.6 billion years ago from a gravitational collapse of matter that came from a large molecular cloud (Bonanno, A. et al., 2002). All of this matter was gathered around in the center of the sphere (the Sun) and the rest flattened into an orbiting disk, creating the Solar System planets, and subsequently other bodies including our planet – the Earth. The Solar System's center became so dense and hot that ultimately it started nuclear fusion in its core, and that is the main process that forms almost all the stars composing the universe (Stahler and Palla, 2008).

The Sun is a yellow dwarf, also known as a G-type main-sequence star (luminosity class V), as we can see in figure 1.10[2]. It has an effective temperature, i.e., the temperature

[1] Illustration by NASA/JPL, taken on the 12th of June 2023
[2] Hertzsprung-Russell Diagram, Credit: ESO, taken on the 12th of June 2023

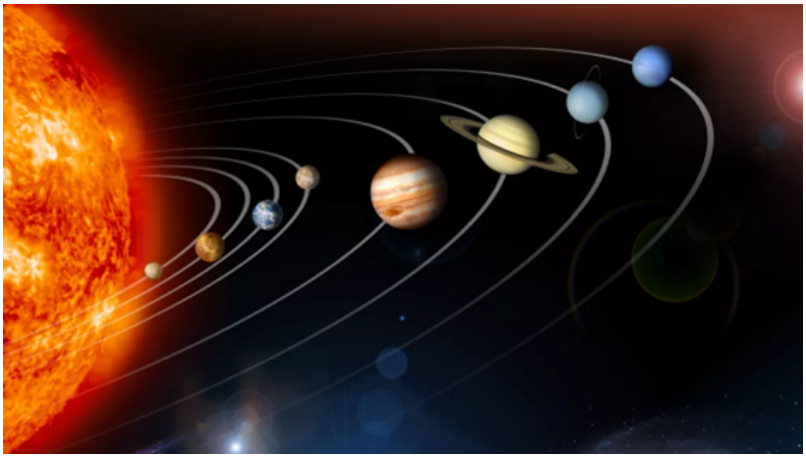


FIGURE 1.9: Representation (not on scale) of the Solar System.

of a black body emitting the same total amount of electromagnetic radiation (Roy and Clarke, 2003) of around 5780 K (Lide, 2004). It fuses in its core around 600 million tons of hydrogen into helium every second (Miramonti et al., 2010), converting in this process almost 4 million tons of matter into energy. This energy is the source of Sun's heat and light, but eventually the hydrogen fusion will be reduced to the point where the Sun is no longer in hydrostatic equilibrium (Wilkinson, 2012). When this happens, the Sun's core will increase in temperature and density, expanding its outer layers, and eventually transforming into a red giant (Laughlin et al., 1997) – as shown in figure 1.11[1]. It's been calculated that when this happens, in approximately 5 billion years, the Sun will absorb the orbits of Mercury and Venus, and the Earth will be inhabitable (Reinders, 2021). After this expansion, its outer layers will get dropped and the Sun will become a white dwarf – a dense type of cooling star, and no longer fuse to produce energy (Mestel, 1952). However it still will give heat and glow to any bodies nearby. Since prehistory that the Sun has a huge influence on Earth, as we can see by the calendars that used based on rotation of the Earth around the Sun and the Sun's observations movements (the days and nights) (Lattis, 2010).

Despite all these stars having different endings to its life's, almost all of them were created by a dust cloud of hydrogen, and that hydrogen fuses into helium in his core as said before. This core is part of the star's structure, and in the case of the Sun, the core also incorporates the likes of the photosphere and the chromosphere as we can observe in figure 1.12[2]. The part of the structure that interest us the most is the photosphere, where

[1]Image taken from Encyclopædia Britannica, Inc., on the 12th of June 2023
[2]Credit: PMF IAS, Physical Geography. Image taken on the 12th of June 2023

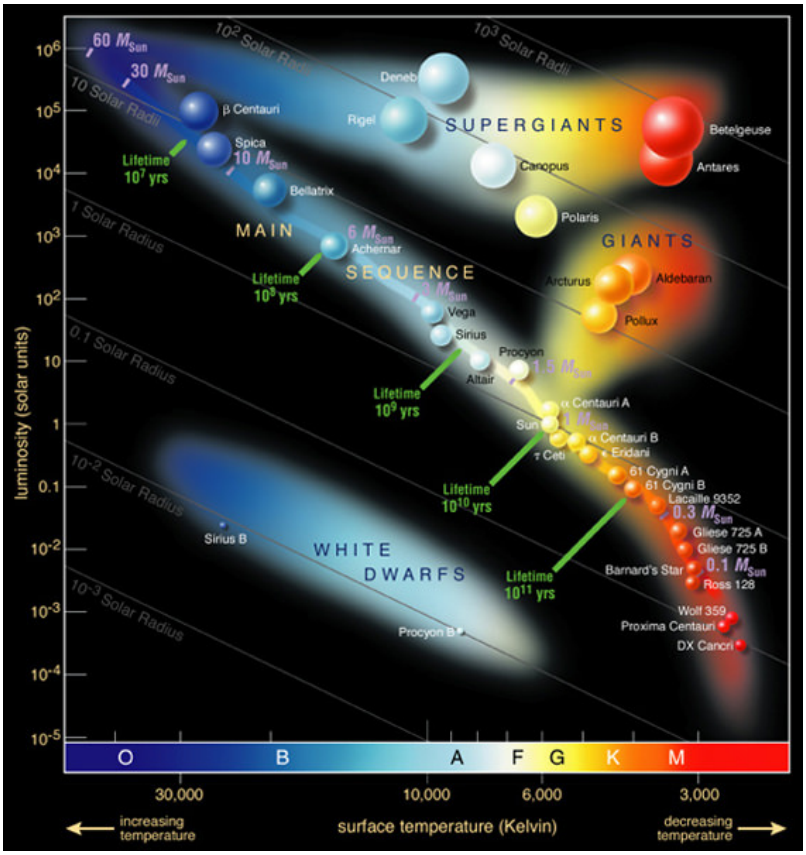


FIGURE 1.10: Intrinsic brightness vs surface temperature of the different types of stars, containing the Main Sequence where the Sun is part of.

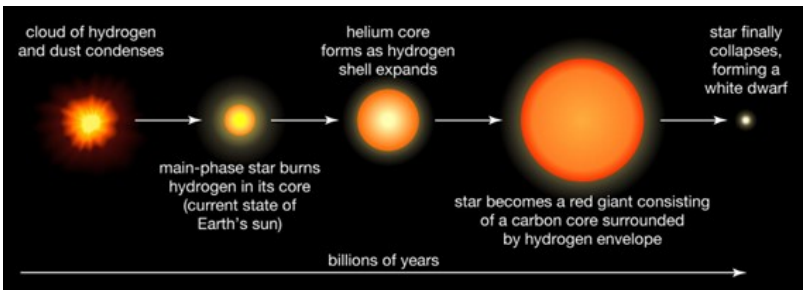


FIGURE 1.11: Evolution of a Sun-like star.

we can find the regions that this project wants to focus on.

The Sun's photosphere is the deepest layer of the star that we can still observe directly and is an outer shell from where the light is radiated. It is in this layer that we can find the sunspots, that can be divided in penumbra and umbra, and faculae – both regions that we want to study. As we can see in the smaller scheme on the top left corner and in figure 1.13[1], the penumbra is the lighter part of the sunspot, and the umbra is darker and usually is inside the penumbra.

[1]Image taken from my own *python* code

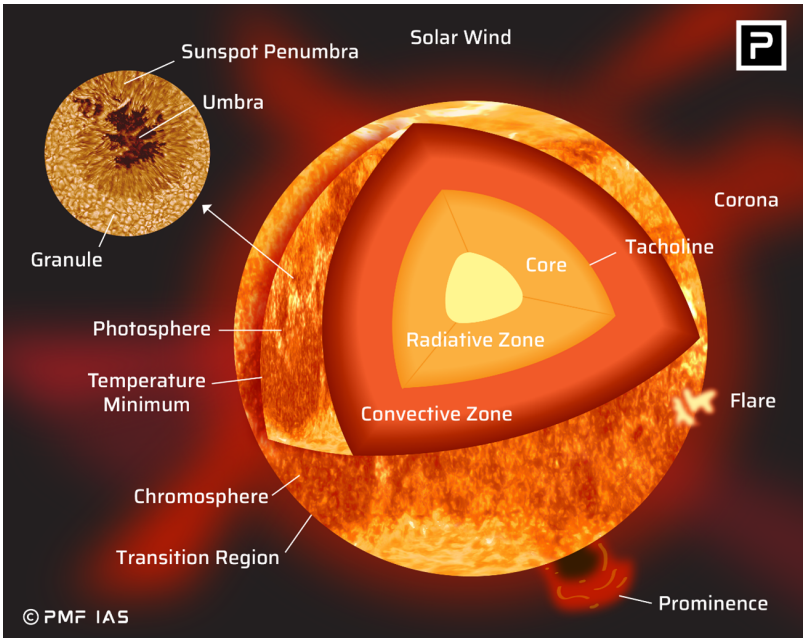


FIGURE 1.12: Representation of the Sun's structure.

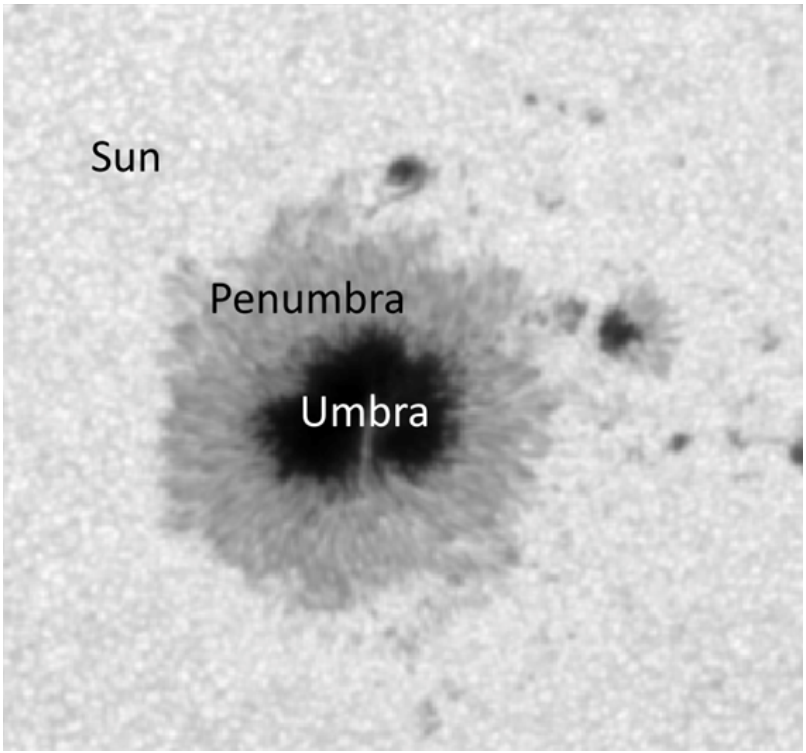


FIGURE 1.13: Sun's photosphere where the Umbra and Penumbra are represented.

These sunspots are regions that are caused by concentrations of magnetic flux inhibited by convection and have a reduced surface temperature. They usually appear in pairs of opposite magnetic polarity, within active regions and they vary according to a 11-year cycle (Labitzke and Van Loon, 1997), as we can see in figure 1.14[1].

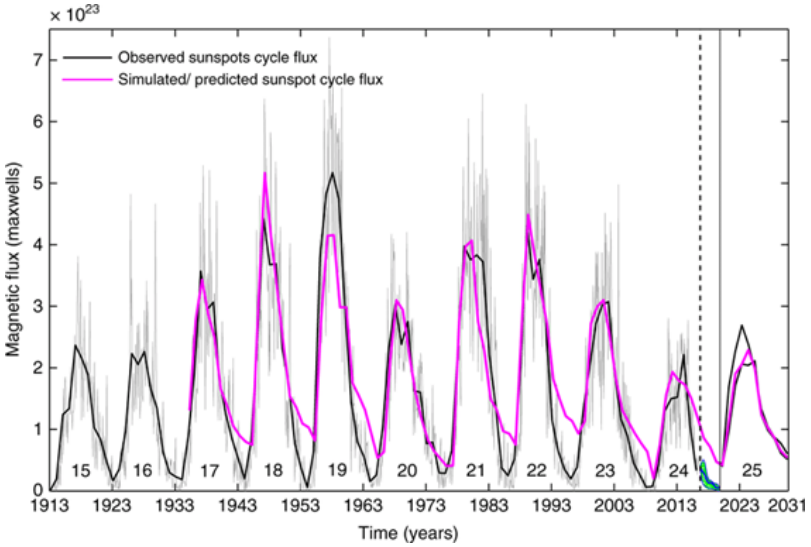


FIGURE 1.14: Cycle of the sunspots with a prediction of the near future.

At this date, we are moving towards a peak of solar activity, so it is a great time with a lot of data and information available to study and understand more about these regions, and subsequently learn more about the Sun and other stars.

1.3 PoET & ESPRESSO

As discussed before, observing and detecting exoplanets can be challenging because of stellar noise, like sunspots and faculae that can affect the lines in the stellar spectrum. So, in order to find and study Earth-like exoplanets, firstly we must study Sun-like stars, observing their spectrum. To do that in a easier way, we can observe the closest star to us - the Sun. With this in mind, there will be in the near future a telescope called PoET that is going to be linked with ESPRESSO to look at the Sun and study its spectrum lines.

1.3.1 ESPRESSO

To work along with PoET, there is a spectrograph called ESPRESSO – Echelle SPectrograph for Rocky Exoplanet and Stable Spectroscopic Observations, which is a high-resolution

[1]Image taken from (Bhomwik and Nandy, 2018), on the 13th of June 2023

spectrograph that is used in astronomical research in order to find and study other planets outside the solar system – exoplanets. It is a third-generational fiber fed, cross-dispersed spectrograph with echelle grating and is located in Chile, in the Very Large Telescope European Southern Observatory (Pepe et al., 2021). It was designed to study exoplanets and stellar physics and optimized to detect and characterize Earth-like exoplanets. Additionally, it can precise measurements of stellar parameters, such as radial-velocity variations that are important to understand if a star has an exoplanet.

ESPRESSO has some key feature such as precise measurements of Radial Velocity and also a high spectral resolution that is achieved using its Spectrograph that is extremely stable. The Precision Radial Velocity Measurements is very important to detect exoplanets because it can detect minute variations in motion of a star (Pepe et al., 2014) caused by the gravitational pull that an exoplanet creates when it orbits his host-star, allowing to find such planets and study its characteristics. This precision can be achieved using the feature Extremely Stable Spectrograph, because it ensures that the performance of the instrument is consistent during long periods of observations. One of the main scientific objectives of ESPRESSO is to achieve a high level of precision in radial velocity measurements around 10 cm/s. This ambitious goal is connected to another one - finding exoplanets using the radial-velocity method, studying stellar activity and characterize the atmospheres and properties of said exoplanets. It represents a big advancement in the field of finding other life on the Universe and also finding more about stellar characteristics. With this in mind, it is important to pair the ESPRESSO with PoET, also because ESPRESSO has a really high resolution, with the instrument covering the optical wavelength band. With this connection, we can look at the Sun to understand more about it (solar physics) and use that knowledge on other stars.

1.3.2 PoET

As mentioned before, exoplanets' research is a field that is rapidly expanding with the goal to find and characterize planets outside the solar system. This has opened up brand new horizons for the comprehension of the formation and evolution of planetary systems. However, such a task is a demanding job that requires sophisticated equipment, instruments, and data analysis techniques. With this in mind, a telescope is being built - PoET (Paranal solar ESPRESSO telescope) which is a project that is going to be linked

with ESPRESSO spectograph in the VLT. The ESPRESSO spectrograph is located in Chile, in the European Southern Observatory of the Very Large Telescope.

When this link is done, PoET will be observing and studying the Sun with the objective to obtain the spectrum of small regions on the solar disk. These regions can be from the quite photosphere, sunspots or faculae, such as referred before. Then, this data will be applied to comprehend how those regions affect lines in the spectrum, alongside used for the determination of the lines that are more sensitive to stellar activity. Following this understanding, we also have the opportunity to learn more about solar physics. The detection and characterization of exoplanets requires precision, due to the tiny wobbles induced in the motion of the stars as seen in 1.1.2. Adding to this precision, the project already achieved spectral resolutions of around R 220,000. This would help to obtain detailed analysis of the spectrum of stars, allowing to understand more the effects of stellar noise on the detection of exoplanets.

1.4 Thesis Motivation

As seen before, PoET will be a great instrument to observe the Sun, working with ESPRESSO. However, it is not efficient to keep the telescope pointing in general at the Sun all times, because it does not interest us to always have observations of the Sun-as-a-star, i.e. integrated light. So, we want to point at specific regions, also to study them and the different kinds of activities happening locally. With this in mind, this project was created with the main objective being using images from the Sun to identify sunspots and faculae, and then point PoET at them. This project will rely on a code to facilitate the use of the telescope, with a software that finds specific parts of the Sun's disk, like sunspots or faculae that are interesting to study. Then it calculates parameters like the Area of that region, its position in the sky or even where that sunspot will be in a few days or where it was in the past. With this parameters, an observer can choose one of these regions and ask the telescope to point at it. This code will receive Sun images from SDO[1] and uses that image to identify where any specific regions are on the solar disk, like sunspots (divided in umbrae and penumbrae) and faculae. The code created to identify this regions was firstly created by researchers from the University of Coimbra (Carvalho et al., 2020) and, afterwards, it was modified in the context of this thesis. The code contained parameters that

[1]This images are taken from https://sdo.gsfc.nasa.gov/data/ and the filter that interests the most for sunspots is the HMI Intensitygram. This website is courtesy of NASA/SDO and the AIA, EVE, and HMI science teams.

can be changed to identify only the umbra or the umbra + penumbra. However, to identify just the penumbra, it is needed a subtraction of images, i.e. the image containing the sunspots is subtracted with the image only containing the penumbra. These regions can also be identified using thresholds. After those identifications, the program can indicate where a sunspot is in real-time on the sky, returning coordinates that can be imported to a telescope to then point it at the selected feature. It can also calculate how many sunspots the Sun has on a determinate day and hour and estimate their area.

In Chapter 2, it will be discussed the methods used in this project and the processing of the images taken from SDO.

In chapter 3, we are going to discuss about the identification of the solar regions, including faculae and sunspots (umbrae + penumbrae).

In Chapter 4, we will talk about the coordinates transformation to correct the azimuth and altitude where we want to point the telescope.

In chapter 5, there is going to be the results of this project.

And finally, in chapter 6, a brief conclusion of the thesis.

Chapter 2

Methods

As discussed earlier, the main goal of this work is to develop a code to identify active regions of the Sun, such as sunspots (umbrae and penumbrae) and faculae in images of the Sun. However, PoET telescope is still under construction, making it so that we must use images from other instruments before its release, but even after that we can still use this images to get an image of the Sun before the telescope is pointed. To do so, we are going to use images from SDO telescope, which are publicly available at https://sdo.gsfc.nasa.gov/data/. The filter that interest us to use is the HMI Intensitygram, as we can see in figure 2.1[1].

HMI means *Helioseismic and Magnetic Imager* and is a SDO instrument that focuses on movement and magnetic properties of the sun's surface. This is the filter of best interest to us because it allows to study the dynamics of convection zones and the origin and evolution of active regions and sunspots (Scherrer et al., 2012). Using the code created by the researchers from Coimbra (Carvalho et al., 2020) and these images, we can identify regions in the Sun, such as sunspots and faculae. These regions will help us study the Sun and its physics, that will be crucial for exploitation of PoET data.

[1]Image taken from https://sdo.gsfc.nasa.gov/assets/img/latest/latest_4096_HMII.jpg, on the 20th of August 2023

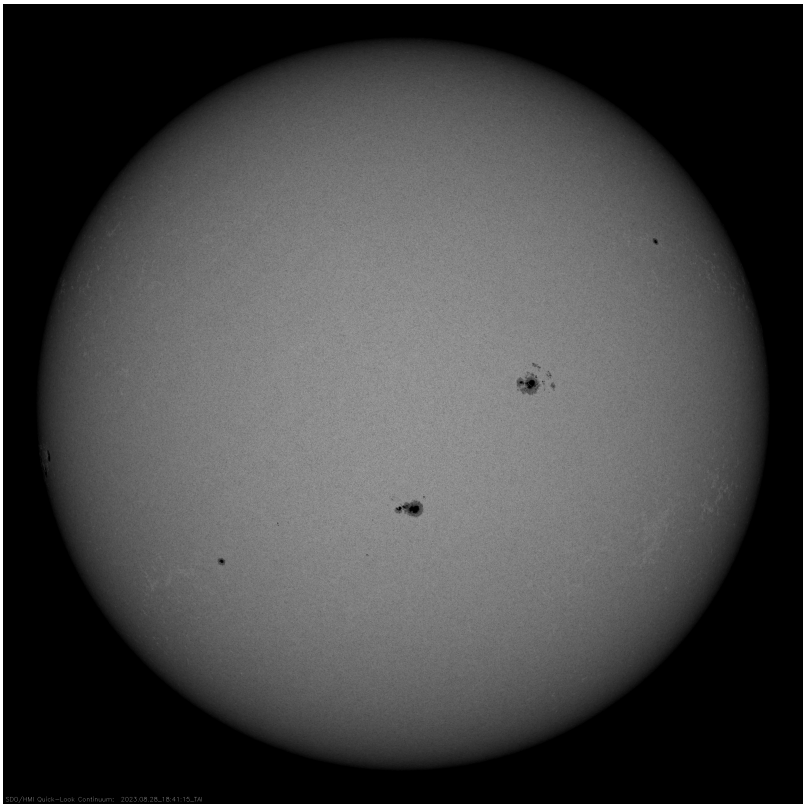


FIGURE 2.1: Image of the Sun with the HMI Intensitygram filter.

2.1 Code structure

As mentioned before, this project was done by the use of a python code originally created by researchers from Coimbra and then adapted to fit the objective of knowing where the regions of the Sun are, additionally to identifying them. To do this, we will follow the fluxogram of the codes shown in figure 2.2.

As we can see, the first part requires to download images. After the download of the image, we need to convert it to a FITS, as the code was made to read those format files.

That image can go into two different codes, the *sunspots.ipynb* that is going to be referenced through this work as the main code, or go into a code named *facula.ipynb*. This second code will only identify faculae, one of the regions present on the Sun, as it returns an image with the faculae identified.

The main part of the project was to identify sunspots and separate them into umbra and penumbra, so we are now going to focus on the center of the fluxogram where the main code is used. That code requires parameters that will help in the processing of the image to get a good identification of regions. After choosing those parameters and

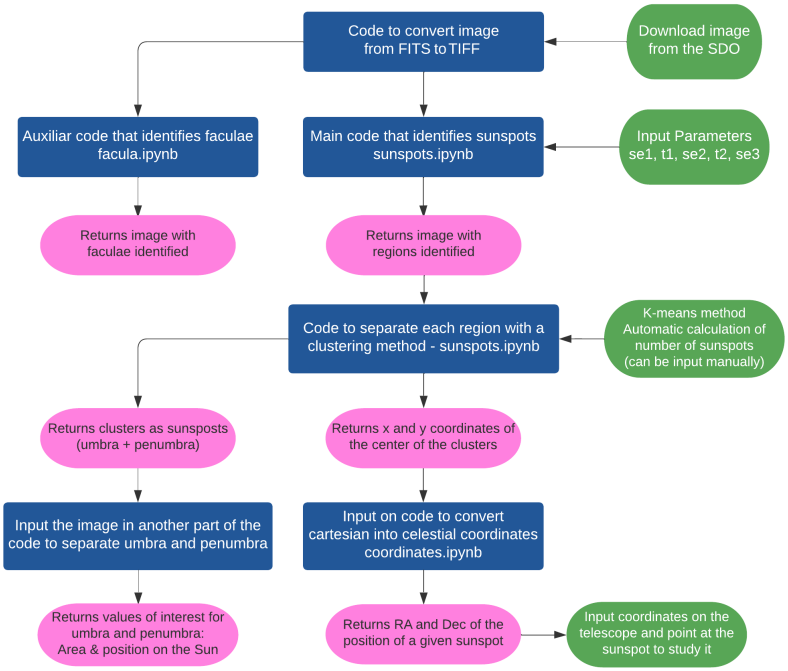


FIGURE 2.2: Fluxogram of the codes used.

inputting them in the code, it will return an image with regions identified. The region will depend on the values of the parameters, but the main focus is to get firstly the sunspots.

After that we have a k-means method that separates each sunspot as a cluster, which is a set of points that are close to each other, that represent a given sunspot. This is made to calculate the centroid of the clusters and to know how many clusters there are. That count can also be done manually, as the automatic calculation is not that well optimized. The calculation of the centroids of the clusters allows to obtain x and y coordinates of the sunspots' centers.

These values of the centers then go into another code called *coordinates.ipynb* that receive them as an input and returns the right ascension and declination of the position of a given sunspot. This is important because we want to get the position of these regions in the sky to point the telescope at them.

Lastly, as we can see on the left side of the fluxogram, the separating of regions into clusters can also help to obtain the umbra and penumbra. Another part of the main code will use the clusters as an input to separate the sunspots into umbra and penumbra and finally return values of interest, such as their area.

2.1.1 Download of the images from SDO

After understanding the structure of the code, we can now start from the top of the fluxogram by downloading images from the SDO and converting them into a FITS format file. These images are available to download from SDO data present in these three different websites:

1. As a GIF - http://jsoc.stanford.edu/data/hmi/images/latest/
2. As a JPEG - https://sdo.gsfc.nasa.gov/data/
3. As a FITS file - http://jsoc.stanford.edu/ajax/exportdata.html

The GIF format is available in the resolutions 1024x1024 or 4096x4096. The website mentioned before allows to download different filters from the latest HMI images, as we can see in figure 2.3[1].

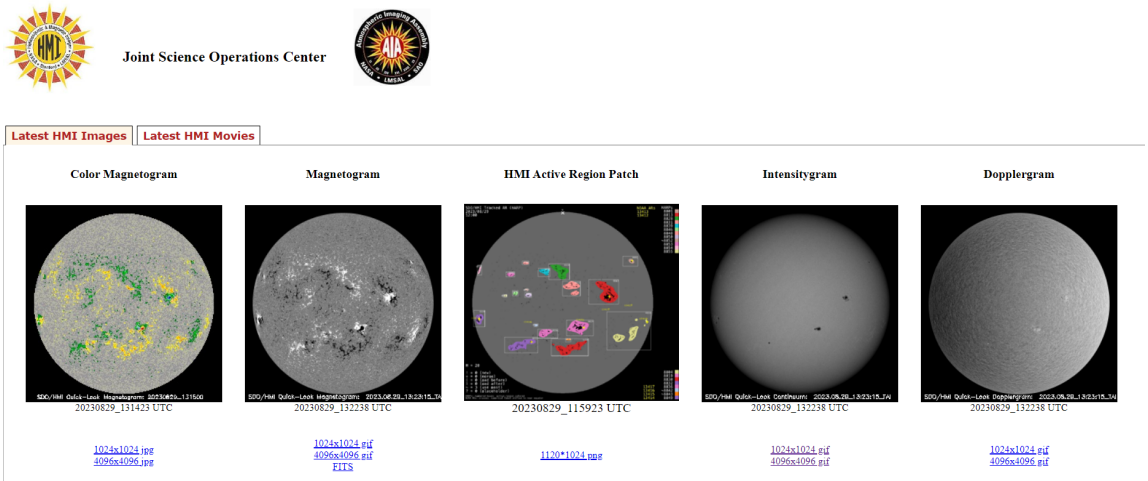


FIGURE 2.3: Different available filters of HMI.

We can see that there is also FITS and JPEG formats available, but using these method to download the image, it is better to choose the GIF format, as it is the Intensitygram. Clicking on the links below the image will open the GIF image in the selected resolution and then right click to save the image on the computer, to use in Python with the code. This process is manual, so it is non sustainable on the long run, especially considering that PoET is projected to have a fully autonomous mode.

The JPEG format is available in the resolutions 4096x4096, 2048x2048, 1024x1024 and 512x512, as shown in figure 2.4[2].

[1] Image taken from http://jsoc.stanford.edu/data/hmi/images/latest/, on the 29th of August 2023
[2] Image taken from http://jsoc.stanford.edu/data/hmi/images/latest/, on the 24th of August 2023

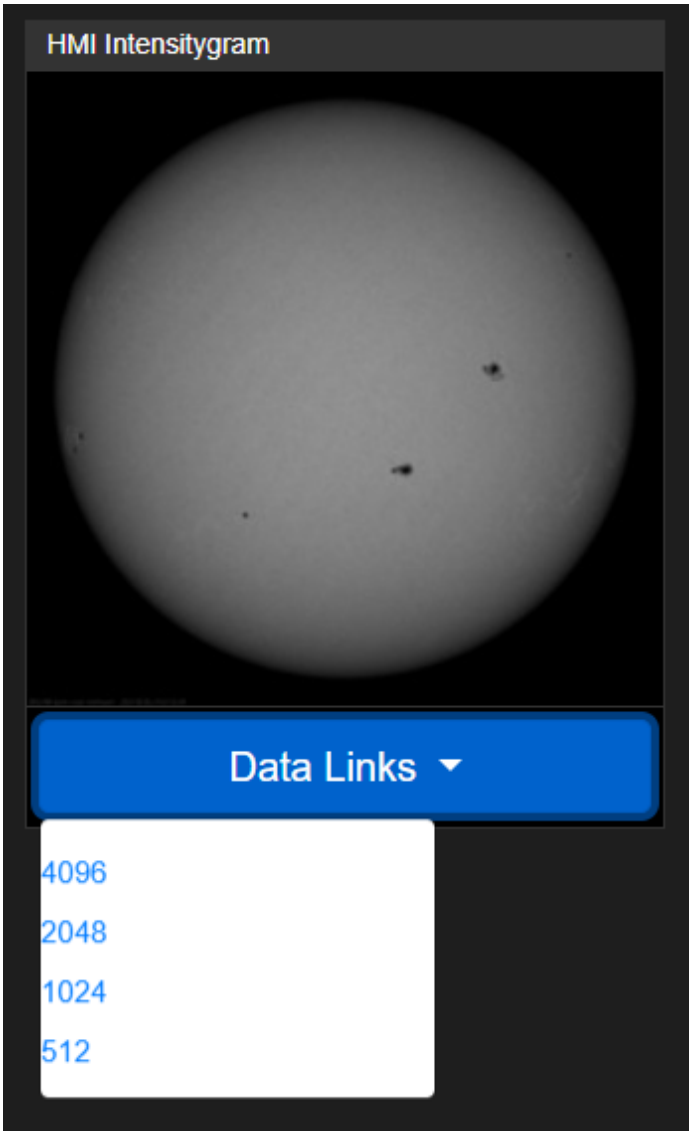


FIGURE 2.4: Different available resolutions of HMI Intensitygram filter in JPEG format.

Clicking those numbers will open the image in the corresponding resolution, after which we can repeat the download to have the input image to our code.

To get the FITS format, the download can be done by providing the information that is requested, as specified in figure 2.5[1]. These parameters also allow access to JPEG format, but that does not interest us for the code used.

To download the image in the FITS format, the parameter named "RecordSet" has to be completed with the following text: "hmi.Ic_45s[2023.05.23_TAI/1d@24h]". The "hmi" represents filter we want, in this case the HMI Intensitygram and the rest before the brackets are series and protocols that the filter has. For example, the one that worked better for the identification of the sunspots was the 45s, which is a series of images taken every 45

[1]Image taken from http://jsoc.stanford.edu/ajax/exportdata.html, on the 25th of August 2023

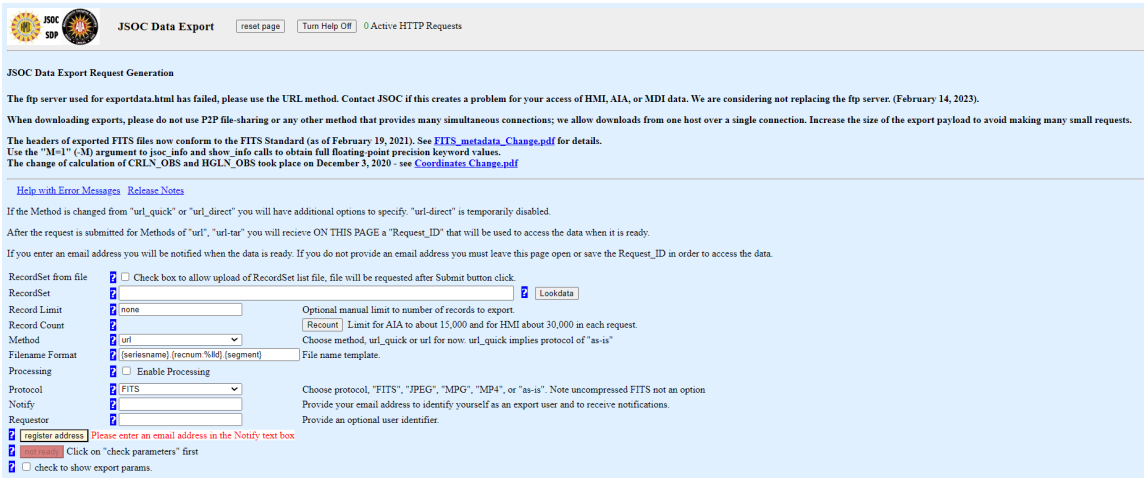


FIGURE 2.5: Parameters needed to download data in the FITS format.

seconds, so we chose that one. After that it is the date of image wanted and the "1d@24h" will get an image from that day. Choosing it to 6h for example will get 4 images from that day, one image every 6 hours. After this the website will calculate how many images that day has, i.e. for example choosing 24h will only get 1 image to download. From there on, choose the protocol FITS, then on the "Notify" box write an email to register the images and finally activate it following the instructions on the mail received. With this, all the parameters will be complete and click on "Check parameters" to then submit the export request by clicking on the "Submit" green box. An ID will appear on the bottom of the screen, as shown in figure 2.6[1].

[1]Image taken from `http://jsoc.stanford.edu/ajax/exportdata.html`, on the 15th of September 2023

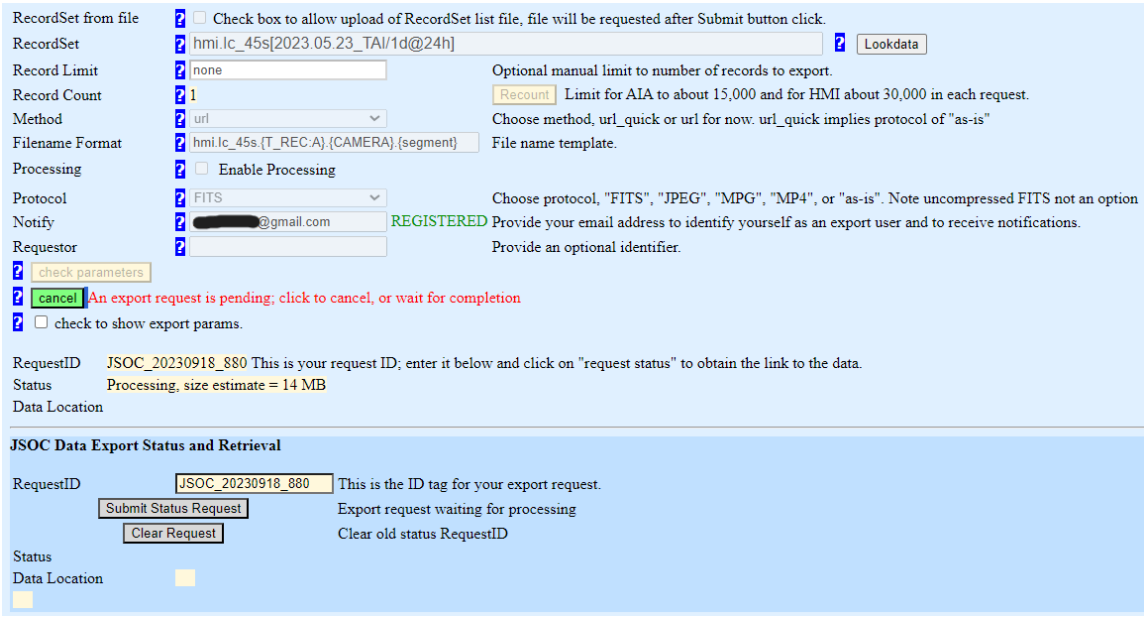


FIGURE 2.6: FITS format file requested for import, with all the parameters completed.

Then just click on "Submit Status Request" and the file for download will appear on the bottom as a blue link, as illustrated in figure 2.7[1]. Clicking on the link will automatically download the file into the computer and then just move it into the folder that contains the Python code.

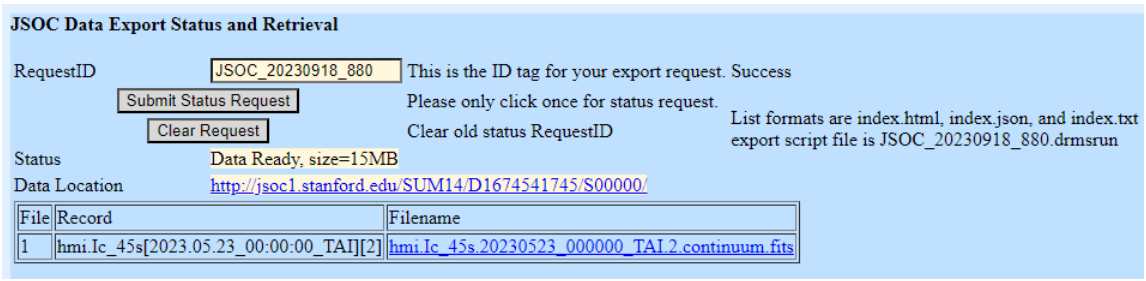


FIGURE 2.7: FITS format file requested for import, with all the parameters completed.

Alternatively, this process can be automated through the drms library, using the following code: (Note that the drms library has to be downloaded with the instructions from https://docs.sunpy.org/projects/drms/en/latest/intro.html#installation (Glogowski et al., 2019).

```
import astropy.io.fits as fits
from PIL import Image
import drms
client = drms.Client()
```

[1]Image taken from http://jsoc.stanford.edu/ajax/exportdata.html, on the 15th of September 2023

```
keys, segments = client.query('hmi.Ic_45s[2023.05.23_TAI/1d@24h]',
                key='T_REC, DATAMEAN, OBS_VR', seg='continuum')
url = 'http://jsoc.stanford.edu' + segments.continuum[0]
sol = fits.getdata(url)

fits.writeto('image.fits', sol, overwrite=True)

# Open the FITS file
hdulist = fits.open('image.fits')

# Extract the data from the first extension as a NumPy array
data = hdulist[0].data

# Close the FITS file
hdulist.close()
```

Running this code will allow to download the image in a FITS format, directly into the computer, which can then be converted into a TIFF file to use in the identifying part of the program.

On one hand, the JPEG and GIF formats are better to download if we want to do a quick test on the code, because those formats only allow to get the latest image taken from SDO. However, the FITS format allows to choose any date that we want to get an image from. The FITS format is a data file, that consists on a matrix of zeros and ones, i.e. it is a binary image with black and white tones. This data can be converted in a TIFF file, that will be further processed to detect the regions of interest. The JPEG and GIF formats can also be changed for to a TIFF file and work on the code. The only difference between them is that you can choose the date on the FITS file, but only use the latest images available in JPEG or GIF formats.

2.2 Image conversion

Although the images can be downloaded in three different formats, the code only can read images in the TIFF format. This happens because, as mentioned before, the TIFF format is a binary image represented in white and black tones, ranging from 0 to 255. To identify the regions present in the Sun, this range is important because we can make a threshold to eliminate its brighter regions. Then, we can obtain the darker regions that

are the sunspots, more precisely the umbra and penumbra. With the threshold we can also separate these two regions, as shown in figure 2.8. The values used for this threshold are going to be explained later in Chapter 3.

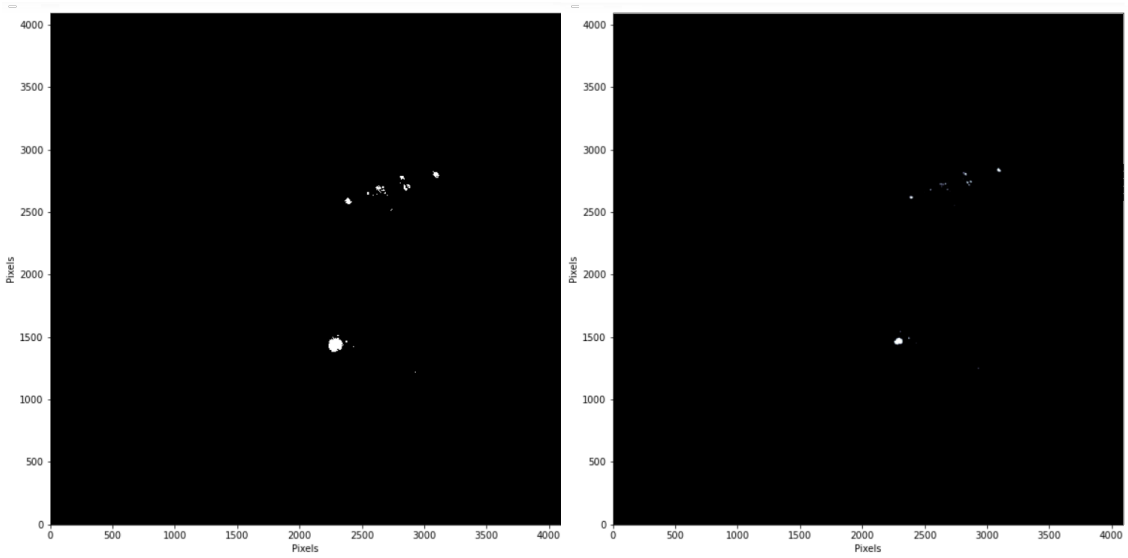


FIGURE 2.8: Comparison of regions identified, umbra+penumbra on the left and umbra on the right.

This separation allows to study the effect just the umbra or the penumbra produce on the spectrum of the Sun. The conversion of the JPEG, GIF or FITS formats into the TIFF format is done by the following code:

```
# JPEG or GIF to TIFF
img = PIL.Image.open("aia_latest_1700.jpg") #name of image downloaded
gray_img = img.convert("L")
gray_img.save('aia.tiff') #name of image saved
#note that this will keep the resolution of the image the same as the image
                                        downloaded


# FITS to TIFF
image = Image.fromarray(data) #data from the code above
image = image.resize((4096,4096), Image.ANTIALIAS) #choose the resolution
                                        of your image
image.save('image.tiff') #name of the image saved
```

Running this code will save a TIFF file on the same folder as the one that contains the other format file. With this format, we can finally use this data to create a binary mask that separates features from the quiet Sun and then identify the regions present in the Sun. As we can see in the end of the code, the data is going to be recorded as a matrix of 4096

rows and 4096 columns, but the resolution in each axis can be decreased to 1024 or 2048. For clarification, the 1024 should be used to make the code run faster, as the matrix that contains the image of the sun is smaller. To get a better resolution, the matrix should be 4096x4096, but the code will run slowly, as we can see in the table 2.1.

Test number	Resolution 4096x4096	Resolution 1024x1024
1	76.4616	8.49077
2	77.2458	8.40563
3	75.6722	8.66671
4	73.7041	8.46258
5	75.7904	8.38855
6	75.0423	9.03601
7	74.5157	8.34422
8	75.6456	8.72535
9	75.8841	8.54298
10	78.0913	8.52286

TABLE 2.1: Time taken, in seconds, for the code to process images in the different resolutions.

So, we can observe that the 1024x1024 resolution runs almost 10 times faster than the 4096x4096. Each was tested 10 times to remove any variability for the way that the operative system runs and processes the code.

Chapter 3

Identification of Sun regions

After downloading and converting the images from the SDO into a TIFF file, we now can identify the regions of interest in the Sun, in this case the sunspots and faculae. As we know, the images can have a resolution of 4096x4096, that is useful to get a better resolution of sunspots and other regions, or 1024x1024, that is better for faster tests, such as experimenting with the parameters. There are 5 parameters on this project:

1. se1: This parameter is a pre-processing that filters the white text present in the image, using a structuring element disk of size, se1. This process will matter when downloading the images as JPEG or GIF, as text will be on the image, but downloading it as a FITS, will not. However, the value of 1 will eliminate any text that appears, no matter the resolution of image;

2. t1: First threshold of the image used to obtain a binary mask of the Sun. This value also should be 1 no matter the resolution and will eliminate any pixel that is higher than one, making it white and the rest of the image remains black. This allows to get a mask of the disk of the Sun;

3. se2: Does a black top hat transformation, that is an operation that highlights small and dark structures or details within an image by computing the difference between the original image and its closing operation using a specified structuring element, in this case se2. This operator dilates the image and then erodes it using the structuring element that is a shape, to differentiate objects from other parts, getting the sunspots. This value will change depending on the resolution, as the higher the resolution, the higher this value needs to be, and will highlight parts of the image with higher intensity, in this case the sunspots;

4. t2: Second threshold value, in this case of the black top hat, that allows to obtain a binarization of the sunspots. This value will also change with resolution. It allows to remove the lower intensity parts that do not contain sunspots, and then make the image black (0) and the sunspots white (255);

5. se3: The final parameter is a filter of open-reconstruction that is a mathematical process where images are smoothed. This removes small objects that are unlikely to be sunspots. This value has to be higher, for higher resolutions, as it removes small dots that can be confused with sunspots. As the resolution gets higher, more details in sunspots will appear on the binary image from before, so it need to be corrected.

After understanding the parameters and observing the values of the pixels in the image, it was concluded that the parameters that optimize the identification are the ones shown in table 3.1.

Parameters	Resolution 4096x4096	Resolution 1024x1024
se1	1	1
t1	1	1
se2	200	100
t2	15000	10000
se3	5	3.5

TABLE 3.1: Parameters chosen for the two resolutions used on the images.

The first two parameters do not change in resolution as they are for removing text that appears on the image and to mask the disk of the Sun. Regarding the other parameters, after many tests, the one that can be changed is the t2, because it adjusts the threshold of intensities in the image, meaning it can identify different regions on the Sun, depending on its value. The se3 is also important, because it removes noise from the identification. This happens when sometimes there are small dots that are identified, but are not sunspots and increasing se3 removes those dots. This parameters will go inside the code represented here: (note that the 'image1' part is the name of the TIFF file that is saved with the image of the Sun)

```
# FITS to TIFF
image = Image.fromarray(data) #data from the code above
image = image.resize((4096,4096), Image.ANTIALIAS) #choose the resolution
                                                   of your image
image.save('image1.tiff') #name of the image saved
```

```
# Open image
img_in = 'image1' #name of the image saved
img = dip.ImageReadTIFF(img_in)
img.Show()
se1 = 1 #parameter
ope = dip.Opening(img,se1)
ope.Show()
t1 = 1 #parameter
bin = dip.FixedThreshold(ope,t1)
bin.Show()
bin = dip.FillHoles(bin)
bin.Show()
contSun = dip.MorphologicalGradientMagnitude(bin)
img_ovr_sun = dip.Overlay(img,contSun)
img_ovr_sun.Show()
se2 = 100 #parameter
BTH = dip.Closing(ope,se2) - ope
BTH.Show()
t2 = 10000 #parameter
BTHbin = dip.FixedThreshold(BTH,t2)
BTHbin.Show()
se3 = 3.5 #parameter
fBTHbin = dip.OpeningByReconstruction(BTHbin,se3)
fBTHbin.Show()
contSPOTS = dip.Dilation(fBTHbin) - fBTHbin
img_ovr_SPOTS = dip.Overlay(img,contSPOTS)
plt.rcParams["figure.figsize"] = (24,20)
fig, ax = plt.subplots(1,2)
ax[0].imshow(img_ovr_SPOTS, cmap='gray');
ax[1].imshow(img, cmap='gray');
plt.rcParams["figure.figsize"] = (12,10)
```

This code will identify the sunspots present in the image of the Sun and then, apply a cluster separation code - in this case the k-means clustering method. This method consists in separating a dataset into K groups of points, where the K corresponds to the number of clusters the algorithm should create, where, in this case, each of the clusters will represent a different sunspot. This code is written bellow and will find how many clusters there are in the image and then separate them, also obtaining the centroid of the sunspots. The part of the code that founds out the number of clusters present in the image is not well optimized for when finding the number of clusters there is on the image. So if the count

of sunspots seem to be very off, after executing the code, the number of clusters can be chosen manually, instead of automatically.

```
points = []
resolution = 4096 #change resolution
for i in range(resolution):
    for j in range(resolution):
        if fBTHbin[i,j] == [True]: #checking point by point if the binary
                                    image is a part of a sunspot and save
                                     it
            points.append([i,resolution-j])

sil = []
kmax = 20 #change this number if clustering is not working well

# dissimilarity would not be defined for a single cluster, thus, minimum
                                    number of clusters should be 2
for k in range(2, kmax+1):
    kmeans = KMeans(n_clusters = k).fit(points)
    labels = kmeans.labels_
    sil.append([silhouette_score(points, labels, metric = 'hamming'),k])

score = [row[0] for row in sil]
k = [row[1] for row in sil]
n_clusters = k[score.index(np.max(score))] #change this number to get from
                                    automatic counting, to manual!

points_x = [row[0] for row in points]
points_y = [row[1] for row in points]
kmeans = KMeans(n_clusters=n_clusters, random_state=0).fit(points)
center_x = [row[0] for row in kmeans.cluster_centers_]
center_y = [row[1] for row in kmeans.cluster_centers_]

#%matplotlib
plt.scatter(points_x,points_y)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1])

for i, j in zip(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1])
                                    :
    plt.text(i+2, j+2, '({})'.format(center_x.index(i)), size = 20)
```

3.1 Sunspots

In the case of the sunspots, the parameters will need change if the resolution is changed between 1024x1024 and 4096x4096. These parameters were found by trial and error and the main one is the t2, because, as previously mentioned, it is the value of intensity that thresholds what parts of the image are going to be identified. The 4096x4096 resolution is better to get high quality images of each of the sunspots identified, but for faster tests, the other resolution is preferable.

As we saw before on the table, when the resolution changes from 1024x1024 to 4096x4096 the se3 increased from 3.5 to 5, as it is a higher resolution so it identifies more noise that are not sunspots. The value of t2 was also increased, as more quality on the image makes the sunspots better shaped and the values of color intensity change. If the identification has any mistakes, it is recommended to do a trial and error on the last 3 values to get a better final image. The parameter se2 can be changed by 100, the t2 by 1000 and se3 by 1, and goes up or down depending, respectively if it identified too much or not on the image. Note that this value of t2 identifies the entire spot, with a need for a change if we are only interested in the umbra. needing a change to get only the umbra, for example.

Although the main code is ideal, there is another way to identify the sunspots by using the threshold code from the openCV library, with a get contours mask function. It will also rely on threshold values, but the image is normalized, so the values of intensities goes from 0 to 255. It is similar to the t2 parameter, i.e. changing the value will identify different regions of the Sun. This adjacent code was used because it is easier to identify just the penumbra and the umbra, and use those identifications to calculate important characteristics such as the area of the regions and their position on the Sun. It is represented here:

```
plt.rcParams["figure.figsize"] = (12,10)
import cv2
from astropy.visualization import simple_norm

def get_contours_mask(matrix):
    b = matrix.copy()
    b[np.isnan(b)] = 0
    info = np.finfo(b.dtype)
    b = b.astype(np.float64) / b.max()
    b = b*255
    b = 255-b
```

```
    img = b.astype(np.uint8)
    ret,thresh = cv2.threshold(img,150,255,0)
    contours,hierarchy = cv2.findContours(thresh, 1, 2)

    cnt = contours[0]
    M = cv2.moments(cnt)
    return contours,thresh

img_ = imagem.transpose()

matrix = data
a = matrix.copy()
contours = get_contours_mask(a)[0]
coint = cv2.drawContours(a, contours, -1, (0,255,0), 1)

norm = simple_norm(img_, 'log')

plt.show()

umbra = get_contours_mask(a)[1]
plt.imshow(umbra, origin = 'lower', cmap = 'bone')
```

As we can see, there is a variable named thresh that uses the openCV library with the cv2 import. Inside the function cv2.threshold there is an image and then two values, in this case, x and 255. The firsts value is the lower threshold of color intensities and everything above until the second value is going to be identified. This allows to get the darker regions as a contour or as a mask and, consequently, obtain the identification of said region. Regarding the values, 0 corresponds to white and 255 to black, so to get the darker regions, the lower threshold needs to be closer to 255. Again with trial and error, so if some identification has some mistakes, shifting the value up or down may help to get a better final image. This will help to get only the umbra and then the penumbra as we are going to see now.

3.1.1 Umbrae

The umbra is the darker region of the sunspot and is inside the penumbra. So, to identify it, it is necessary to increase the values of the thresholds used from before, as the umbra is darker than the penumbra. To do that, the only thing that needs to change is the value

of t2, on the main code, and, on the alternative code, change the x between 0 and 255 as talked about above.

As seen before, using $t2 = 15000$ on the main code will identify the sunspot (umbra + penumbra), so rising that value to around $t2 = 30000$ works to identify just the umbra. Note that this value can be tweaked up or down, with trial and error again, to get a better identification of the umbra. Regarding the alternative code, the value that identifies just the umbra is around $x = 150$ and this value can be tweaked as well. Comparing the two codes, although the main one can have a better processed image, both of them will get a final image with the same resolution - 4096x4096. Additionally, the alternative code is much faster than the main one, as we can see from the table 3.2.

Tests	Resolution 4096x4096	Resolution 1024x1024
1	76.4616	3.47913
2	77.2458	3.31365
3	75.6722	3.84769
4	73.7041	3.03712
5	75.7904	3.61424
6	75.0423	3.50072
7	74.5157	3.53145
8	75.6456	3.98569
9	75.8841	3.70426
10	78.0913	3.37699

TABLE 3.2: Table comparing how many seconds it takes for both codes to execute.

Using the alternative code will get the same results as the other one, with less time, so it is recommended to use it. This part is also important for the identification of the penumbra, because the penumbra will be a subtraction of the umbra inside the sunspot. So if only the penumbra wants to be identified, it is easier to use the alternative code to get a faster identification of the umbra to remove it from the sunspot. The sunspot will always have to be identified with the main code, because lowering the value of the threshold from 150, will get too much noise to work with, as we are going to see now.

3.1.2 Penumbrae

The penumbra is the brighter region of the sunspot, involving all the umbra inside its area. As it is brighter than the umbra, lowering the threshold values would identify the penumbra, but it is almost similar to the rest of the disk's color. This will create a lot

of noise when trying to identify the penumbra, as shown in figure 3.1. Also, the umbra should not be identified, but as we can see it appears in the image.

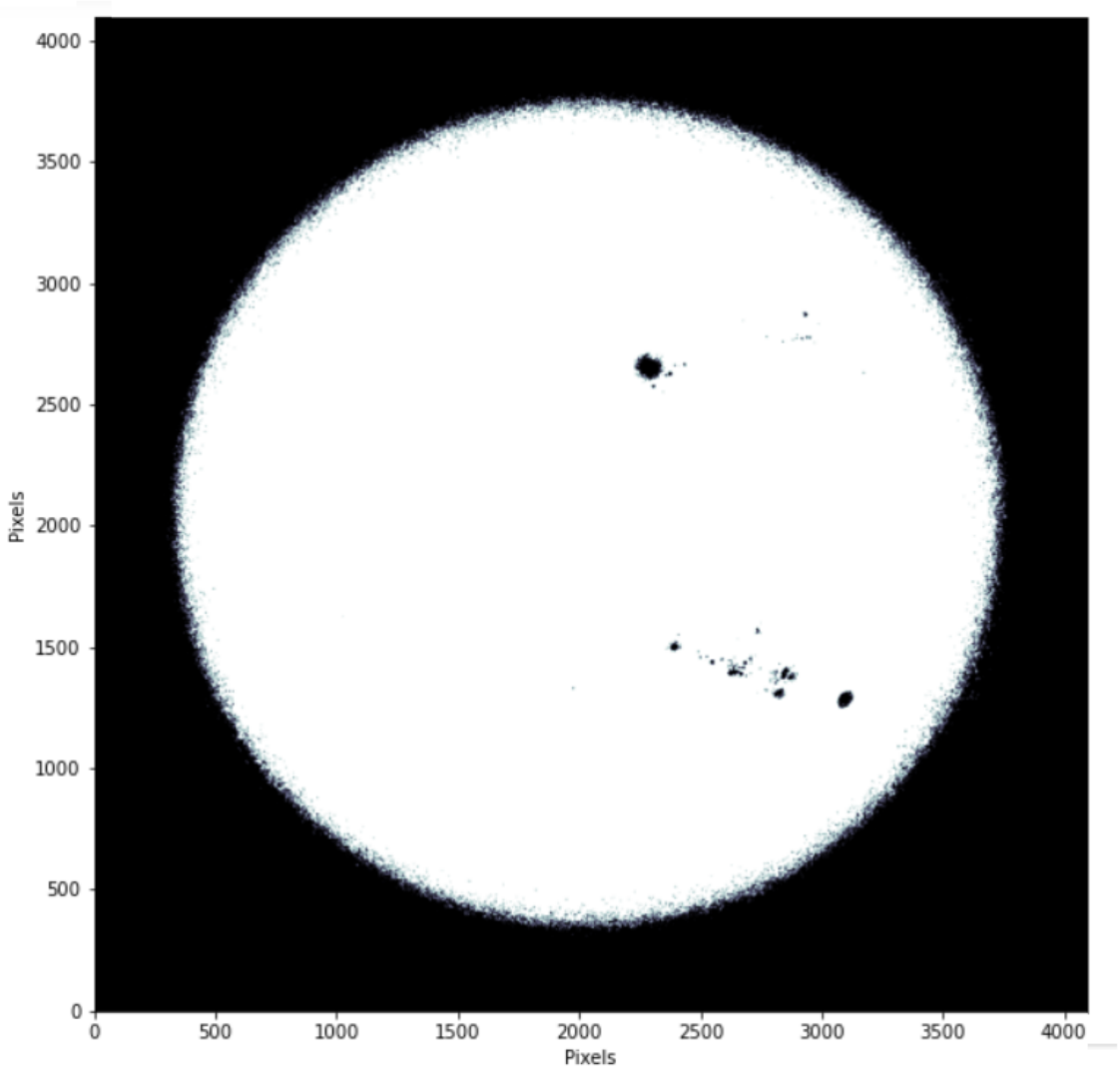


FIGURE 3.1: Image of the attempt to identify the penumbra, with a lot of noise and the umbra mistakenly identified as well.

This image was obtained using the alternative code from before, with values of 100 and 255 on the lower and higher thresholds, respectively. Decreasing the lower threshold will get the brighter regions identified, however the higher threshold value will always be capped at 255, and changing the number will not affect the darker regions. So using the alternative code is not the best option, as it will identify the penumbra + umbra, as the umbra is darker and the higher threshold can not be capped at 150 for example. The noise also does not help to identify the penumbra, as we can see on figure 3.1 with the circumference of the Sun being imperfectly defined.

To fix this problem, we then used the main code that identifies the sunspots and the alternative code that identifies the umbra and subtract the two images. As we know, the penumbra is outside of the umbra and the sunspot is umbra + penumbra, so subtracting the umbra to the sunspot will get the penumbra. Obviously the images have to be of the same resolution, otherwise it will not subtract correctly. The code made for this part was the following:

```
for j in range(4096):
    for i in range(4096):
        if (i-2048)**2 + (j-2048)**2 >= 1845**2 :
            umbra[i,j] = 0
##other part of the code to separate umbra and penumbra
image_1 = Image.fromarray(data)
image_1.save('image_another.tiff')
# Open image
img_in_1 = 'image_another'
img_1 = dip.ImageReadTIFF(img_in_1)
print('Done!')
ope_1 = dip.Opening(img_1,1)
print('Done!')
bin_1 = dip.FixedThreshold(ope_1,1)
print('Done!')
bin_1 = dip.FillHoles(bin_1)
print('Done!')
contSun_1 = dip.MorphologicalGradientMagnitude(bin_1)
img_ovr_sun_1 = dip.Overlay(img_1,contSun_1)
print('Done!')
BTH_1 = dip.Closing(ope_1,200) - ope_1
print('Done!')
BTHbin_1 = dip.FixedThreshold(BTH_1,15000)
print('Done!')
fBTHbin_1 = dip.OpeningByReconstruction(BTHbin_1,5)
print('Done!')

#get the penumbra
penumbra = np.zeros([4096,4096])
spots = np.zeros([4096,4096])
for i in range(4096):
    for j in range(4096):
        if fBTHbin_1[i,j] == [True]:
            spots[4096-j,i] = 255
```

```
            penumbra[4096-j,i] = 255 - umbra[j,i] #subtraction of sunspot -
                                                  umbra

plt.imshow(penumbra, origin = 'lower', cmap = 'bone')
```

As we can see, inside the for cycle there is a switch in coordinates because the two images are rotated 90 degrees. The x has to change to y and vice-versa in one of the images so the subtraction of them works. After this, there was also used the code to find the clusters of just the penumbra to see it they match the same position as the sunspots and umbra. As we can see in figure 3.2, the empty spots inside the penumbra match with the umbra, so the subtraction worked well. Note that this figure has the pixels reduced as it is a zoom in only the regions identified, the rest is just the quiet Sun.

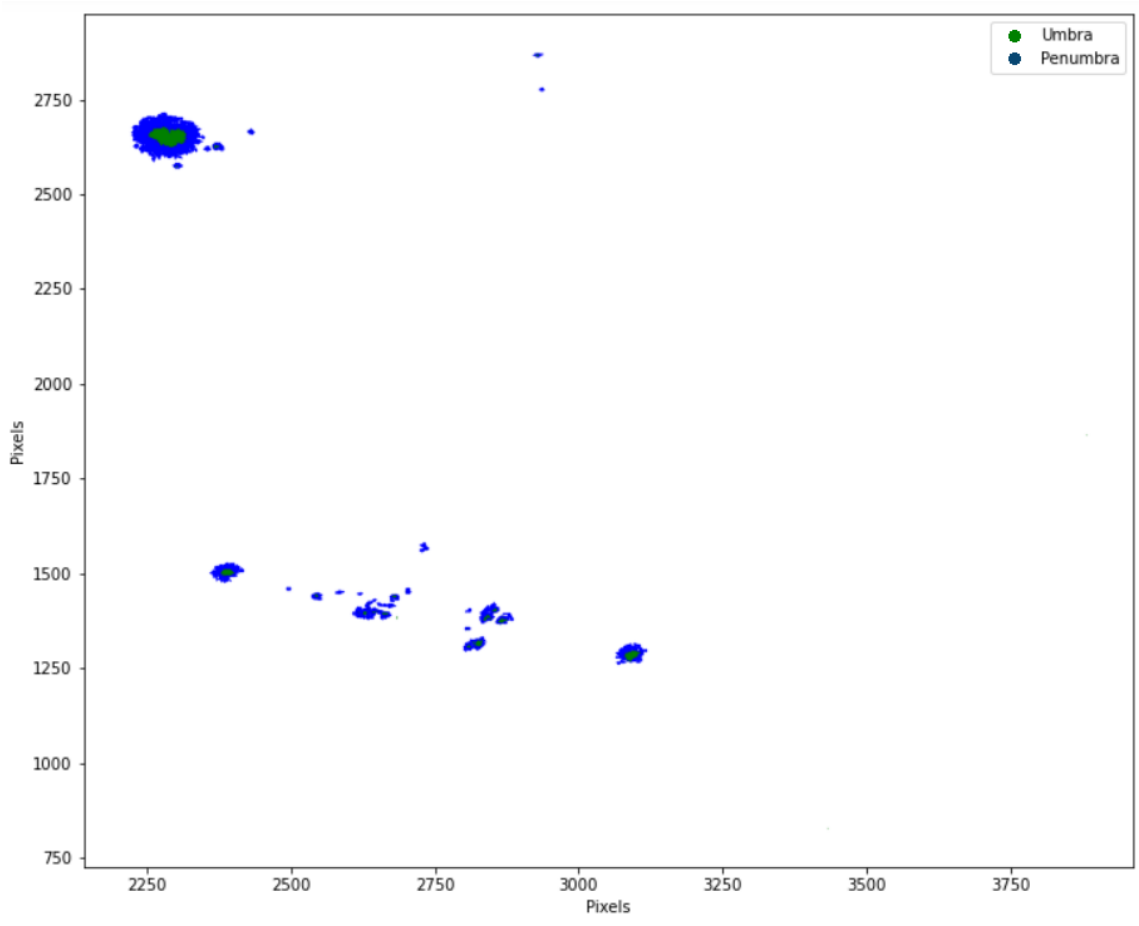


FIGURE 3.2: Clustering method to find the positions of the penumbra, with the umbra represented in green and the penumbra in blue.

After these regions are identified, we can know obtain the location of the regions' centers and convert the corresponding coordinates to determine the position at which the telescope must be pointed to.

3.2 Faculae

The facula is a brighter and hotter region of the Sun that forms on the valleys between granules, created by concentrations of magnetic field lines. As the region is brighter, the value of thresholds is changed again, but the alternative code that was used for this remains the same. The only thing that changes is the way to get the image from SDO, as another filter is needed to get the faculae. The filter needed is the AIA, which means *Atmospheric Imaging Assembly* and contains images of the solar atmosphere in diverse wavelengths to link surface to interior changes. The wavelength that interest us for the faculae is 1700 Å, represented in figure 3.3, on the left, and the image can again have different resolutions.

In this case, the code is relatively fast, because it is the alternative code from before, so the resolution used was 4096x4096, but the 1024x1024 resolution can be used again for faster tests. The only problem was to download an image from a specific hour and day, so it was decided to use the latest image available and work with it. The value found for the threshold was 140, but it can tweaked just as before. Note as well that the part of the code that contains the equation $b = 255 - b$ was removed to invert the colors of the image, as we only want the brighter regions to be identified. The final identification of the faculae is represented in figure 3.3, on the right.

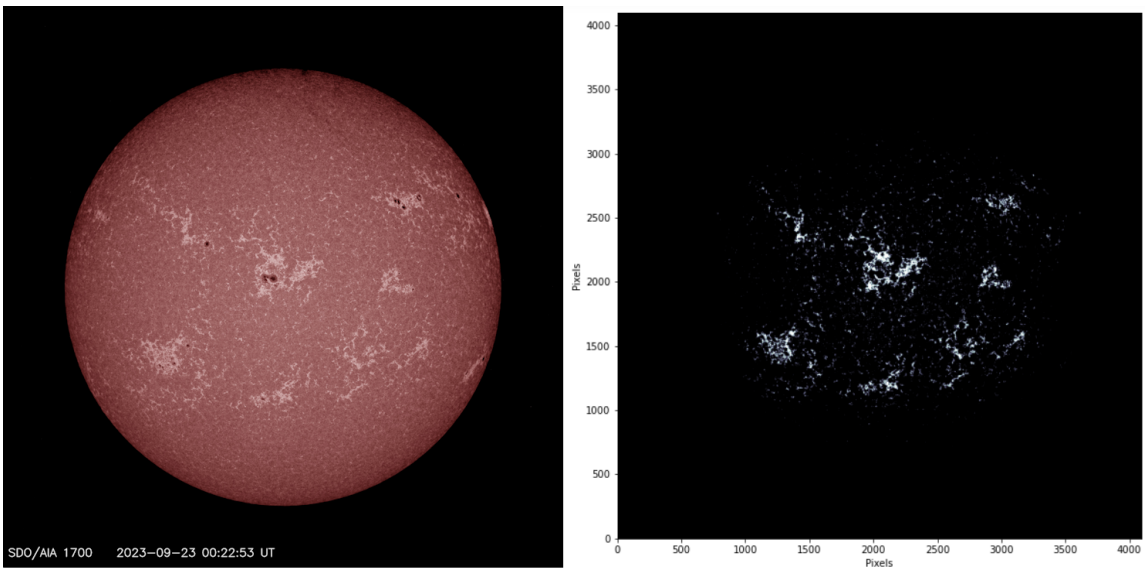


FIGURE 3.3: Sun with the filter AIA (wavelength of 1700 Åon the left and the Faculae identified on the right.

The final identification could be better, as there are some faculae on the edges of the

disk that have different intensities and will not be identified by the threshold of 140. Additionally to this, the area of the faculae could be calculated. However, as the identification is not optimal, this was left for future work.

Chapter 4

Coordinates transformation

Additionally to the main code of identifying the sunspots, there are coordinates transformations that are crucial to the functionality of the telescope. As referred in Chapter 1, PoET will be installed in Chile and SDO images are coming from a satellite that is orbiting Earth. This difference in position has to be corrected, so when the telescope points at the Sun, it finds the right spot containing the region in interest. Adding to this, it is also needed to convert Cartesian coordinates represented on the images into celestial coordinates in the celestial sky. This is important because the center of the regions on the Sun will have a given x and y on the image, but in the sky we need to get the right ascension α and declination δ (and, ultimately, azimuth and altitude) to point the telescope at them. Furthermore, we are going to see that the images have the Sun's equator on an horizontal line, but on the sky that line rotates a certain angle.

4.1 Converting Cartesian coordinates into celestial coordinates

As mentioned before, the images obtained from SDO have sunspots in the Cartesian coordinates, i.e. x and y. However, when we look at the sky and want to point a telescope, the bodies have celestial coordinates called right ascension α and declination δ. Right ascension is the angular distance measured along the celestial equator, beginning on the position of the Sun at the March equinox and going eastward to a particular point. Declination is the angle measured perpendicular to the celestial equator from the horizon to north until the point in question. Combining these two coordinates, allows to get a specific location of a point on the celestial sphere. As our main goal is to be able to point PoET to the detected features, we need their coordinates into the celestial system. The Cartesian

coordinates are provided in pixels that can go from 0 to 4096 (the best resolution possible was used for the conversion to avoid errors in calculations). Adding to that conversion, it is also needed to consider that the equator of the Sun is horizontal on SDO images. But, when we look at the sky, the equator may be rotated an angle. The inclination of the Sun can be obtained from a sunpy module, using a code in python and depends on the location and the time of observation.

After the desired features were identified, we can then proceed to select the number of a sunspot that we are interested in, for example if 6 sunspots were detected, our choice was restricted to the [0, 5] closed interval. In this case, the k-means clustering method found 6 sunspots, so we can choose a number from 0 to 5. The sunspot chosen was the one marked with the number 0 and using the variables center_x and center_y, the Cartesian coordinates of the center of sunspots can be obtained. As the 1024x1024 grid was used to speed up the codes, the coordinates must be multiplied by 4 to transform them to the position in the original grid - 4096x4096.

Now, with the obtained values of x and y it is needed to convert the size of a pixel to a value of right ascension, α, and declination, δ. To do this, we need to calculate the angular size of the diameter of the Sun on the sky, and that can be obtained according to the scheme on figure 4.1.

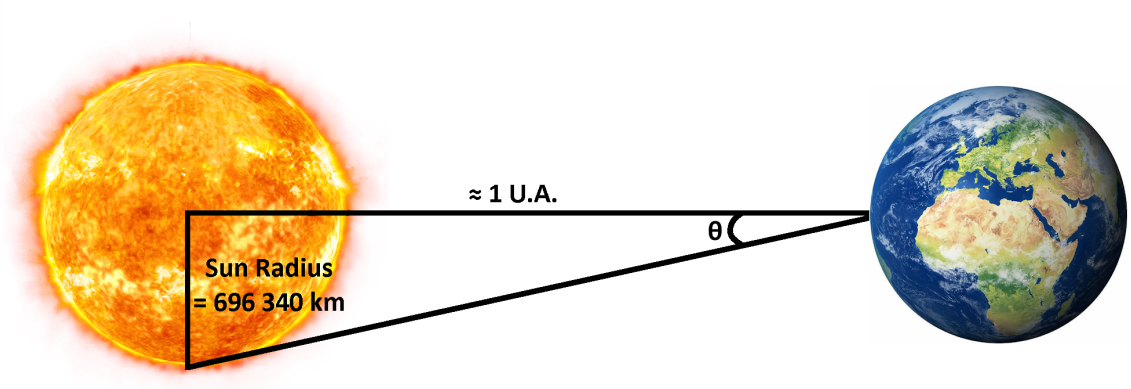


FIGURE 4.1: Scheme used to calculated the Sun's radius angle on the sky, not on scale.

The θ is the angle that the radius of the Sun appears on the sky. Usually it is around 30 minutes of arc or 0.5 degrees, depending on the distance between the Sun and the Earth. This value is around 1 U.A. or $\approx 1.49 \times 10^8$ km, , changing as Earth moves closer to the aphelion (January) and the perihelion (July). The distance can be obtained using the sunpy module, with the following code: (note that you can change the day wanted by replacing 'now' with another date)

```
distance, vazio = re.split(' ', str(sunpy.coordinates.sun.earth_distance(
                                        time='now')))
distance = float(distance)*1.495978707e8
```

After that, using the triangle from figure 4.1 and equation 4.1, we can calculate θ and double it to get the angle of the diameter of the Sun's disk on the sky.

$$\theta = \tan^{-1}\left(\frac{R_{Sun}}{d}\right) \tag{4.1}$$

Where d is the distance between the Earth and the Sun according to the day in question and $R_{Sun} = 696340$ km is the radius of the Sun. Following this, even though the images are 4096x4096 pixels, the diameter of the Sun is not equal to 4096, but rather to 3746 (175 of pixels from both sides do not correspond to the Sun). Then, 2θ corresponds to 3746 pixels and, consequently, 1 pixel can be converted into an angle in degrees. With this in mind, we can convert x and y into angles through the equation 4.2, where x can be replaced by y to get the declination instead of the right ascension.

$$\text{Right ascension} = (x - 2048) \times \frac{2\theta}{3746} \tag{4.2}$$

The way that the images are obtained from the SDO, the Sun is always centred at the $(2048, 2048)$ point, but it is easier if it was centred at $(0, 0)$. So the values of x and y were subtracted by 2048, before being converted to angles. As we know the location of the center of the Sun in the sky at a given time and place, the subtraction allows to obtain a difference of value, such as a $\Delta\alpha$ and $\Delta\delta$ that will be added to the values of the center of the Sun, with the center of the sunspot being a translation from it. It is easier to understand values of altitude and azimuth for the position of the Sun and, declination and right ascension can be converted into them. The altitude corresponds to the declination and the altitude to the right ascension. So, the values of the center of the Sun were obtained in azimuth and altitude, using this code:

```
location = astropy.coordinates.EarthLocation(lat = 41.17749, lon = -8.59466
                                        )
time = Time(datetime(2023, 4, 13, hour=19, minute=13, second=0))
sun = astropy.coordinates.get_sun(time)
sun_coordinates = astropy.coordinates.AltAz(location = location, obstime =
                                        time)
```

```
sun_altitude = str(sun.transform_to(sun_coordinates).alt)
sun_azimuth = str(sun.transform_to(sun_coordinates).az)

deg, minutes, secs, vazio =  re.split('[a-z]', sun_altitude)
if math.copysign(1, float(deg)) == -1:
    sun_alt = float(deg)-float(minutes)/60-float(secs)/3600
else:
    sun_alt = float(deg)+float(minutes)/60+float(secs)/3600

deg, minutes, secs, vazio =  re.split('[a-z]', sun_azimuth)
if math.copysign(1, float(deg)) == -1:
    sun_azi = float(deg)-float(minutes)/60-float(secs)/3600
else:
    sun_azi = float(deg)+float(minutes)/60+float(secs)/3600

print(sun_alt, sun_azi)
```

This code can also be changed to account for a specific Earth location and date and there were tests done to ensure that those values are realistic, as further discussed in Chapter 5.

Lastly, we still have to consider the rotation of the equator of the Sun, depending on the Earth position. For instance, the image from SDO has the equator horizontally, but on the sky it is rotated an angle (as Chile is in the South Hemisphere, the Sun appears to be upside down). This angle can be obtained from a sunpy module using the following code:

```
#sun diameter on the sky
distance, vazio = re.split(' ', str(sunpy.coordinates.sun.earth_distance(
                                        time='now')))
distance = float(distance)*1.495978707e8
angle = 2*math.degrees(np.arctan(1392700/(2*distance)))
#rotation code
location = astropy.coordinates.EarthLocation(lat = lat_Chile, lon =
                                        lon_Chile)
time = datetime(2023, 3, 20, hour=12, minute=0, second=0)
orientation = sunpy.coordinates.sun.orientation(location, time)
orientation = str(orientation)
deg, minutes, secs, vazio =  re.split('[a-z]', orientation)
if math.copysign(1, float(deg)) == -1:
    orient = float(deg)-float(minutes)/60-float(secs)/3600
```

```
else:
    orient = float(deg)+float(minutes)/60+float(secs)/3600

print(orient)
```

The location and time can be changed for any place or date of interest, but it is important to have them set to those from the place and time of observation. Running that code will return the orientation angle for the Sun from a specified Earth location and time. This is the angle between the meridian that passes through local zenith and solar north, measured eastward from local zenith, as shown on the scheme from figure 4.2. In this case the angle is always going to be higher than 90°, because Chile is in the South Hemisphere, so the Sun is inverted (North Pole on the bottom) compared to how it is in Porto.

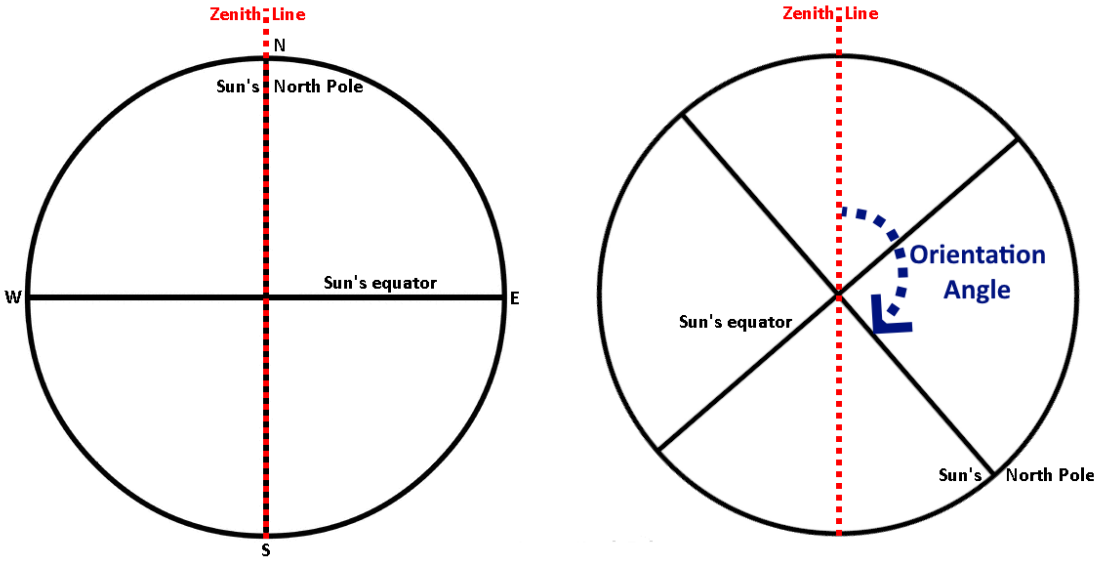


FIGURE 4.2: Scheme of the orientation angle, with the Sun on the images on the left and the Sun on the sky on the right.

Therefore, we can using this angle to rotate the right ascension and declination of the center of the sunspot, as provided in equation 4.3:

$$[\Delta RA, \Delta DEC] = [x\cos(\theta) - y\sin(\theta), x\sin(\theta) + y\cos(\theta)] \tag{4.3}$$

Where x and y is the right ascension and declination of the center of the sunspot, respectively and θ is the orientation angle of the Sun's equator calculated before. Finally, adding these 2 values to the right ascension and declination of the center of the Sun, we can point the telescope at the sunspot in interest.

4.2 Parallax correction

After all this, we still need to consider the fact that SDO and PoET are in different locations. This correction will be minor, as the distance between the Earth and the Sun is much larger than the distance between SDO orbit and Chile. However, it is still important, consequence of the small FOV (Field of View) of the telescope, where a small error in coordinates values can make the telescope point at the wrong direction.

As we can see in figure 4.3[1] PoET will be located where the VLT is and SDO has a projected orbit shaped like a figure-8 form. This happens because SDO is in an inclined geosynchronous orbit around Earth, i.e. its orbital period is equal to the Earth's rotation (23 hours and 56 minutes) (Pesnell et al., 2012).

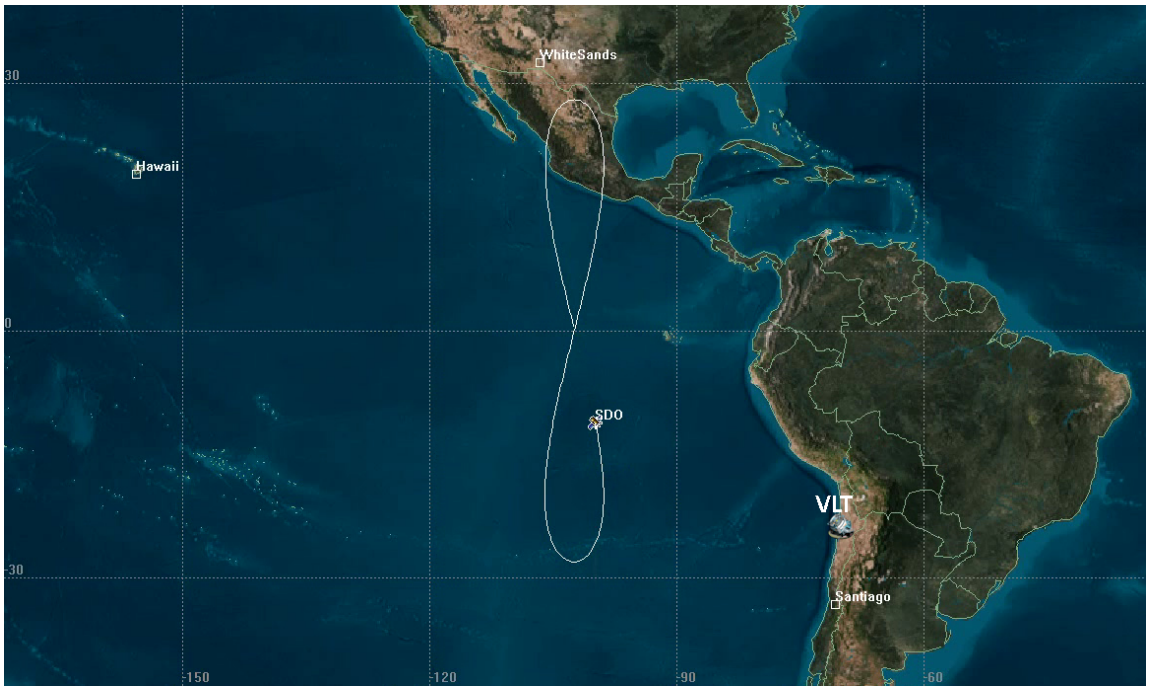


FIGURE 4.3: Localization of SDO projected orbit, near the VLT observatory.

From the figure we can also get the Latitude and Longitude of the projection of SDO on the Earth's surface, such as the coordinates of the location where PoET is going to be. The main calculation to do consists in the most and least distance between the two objects, so we are going to consider the top and bottom of the 8-shaped form. With this in mind, the Longitude of SDO is then 102 ºW and the Latitudes are 27 ºN and ºS. In case of PoET, as we know, it is located in the Atacama Desert on the coordinates 24.63 ºS and 70.4 ºW. Using these coordinates, we can calculate the straight line difference between the two

[1] Image adapted from https://sdo.gsfc.nasa.gov/gallery/main/item/68, on the 30th of August 2023

locations and, with that, the angle difference of where the Sun is in the sky at the same time. This calculation is going to be made for the maximum and minimum difference in distance between SDO and the VLT location, using equation 4.4 that comes from the scheme in figure 4.4.

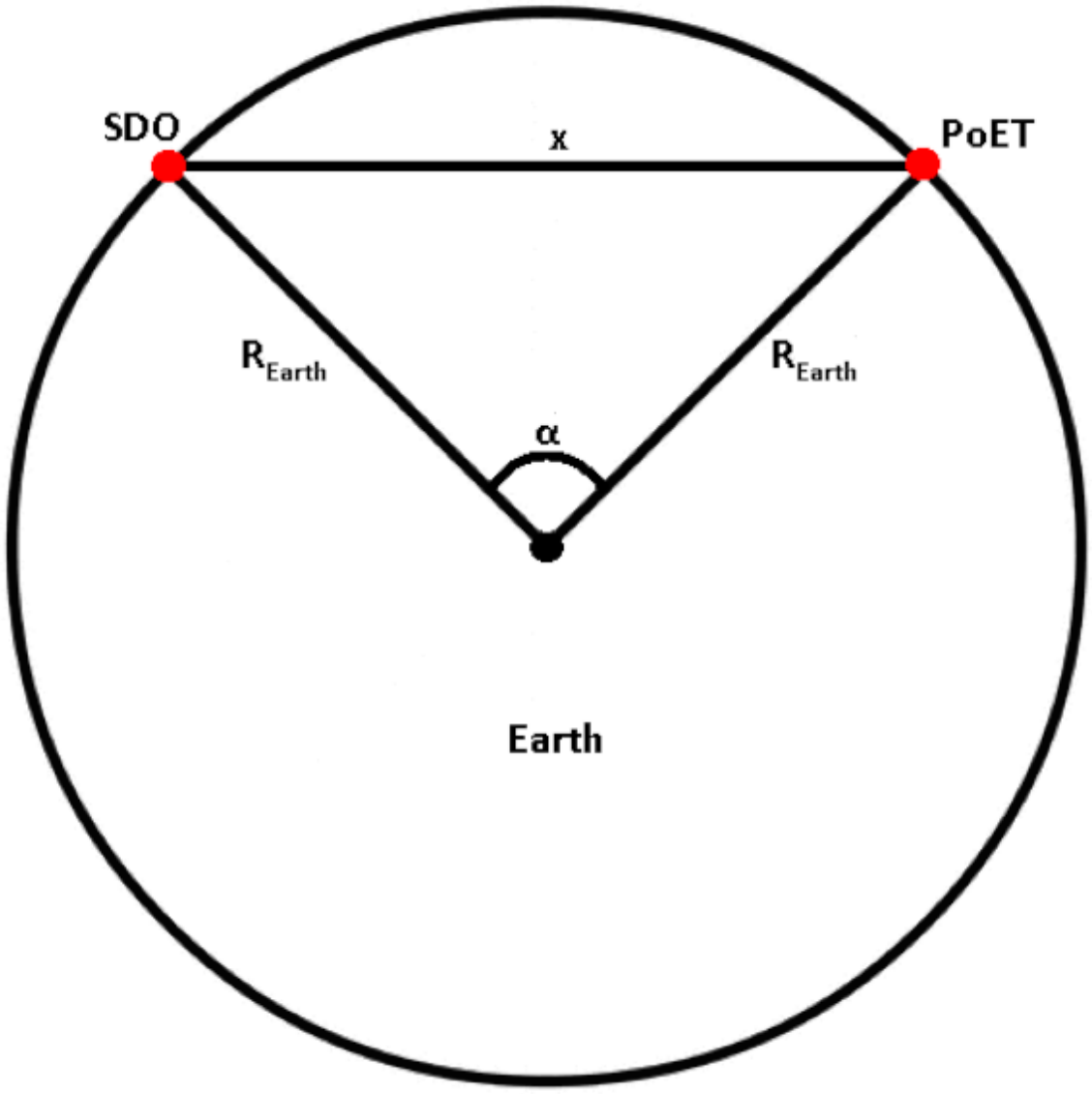


FIGURE 4.4: Angle difference created between different of points of view coming from SDO and PoET.

We want to find the value of x and the angle α that corresponds to the latitude or longitude, depends on which calculation is made using equation 4.4. This equation was done with the cosine law and finds the straight line distance for both latitude and longitude.

$$\begin{aligned} x^2 &= R^2 + R^2 - 2R^2 \cos\alpha \\ x &= R\sqrt{2(1-\cos\alpha)} \end{aligned} \tag{4.4}$$

Then, to find the straight line distance between SDO and PoET, the Pythagorean theorem is used in equation 4.5. The cathetus are the straight line distances in the latitude and longitude (the x from 4.4) and the distance and the distance between the two locations in kilometers is the hypotenuse.

$$\text{Straight line distance}^2 = x^2_{latitude} + y^2_{latitude} \tag{4.5}$$

With this straight line difference, the difference of angle between the two points of view and that is done by using the scheme in figure 4.5.

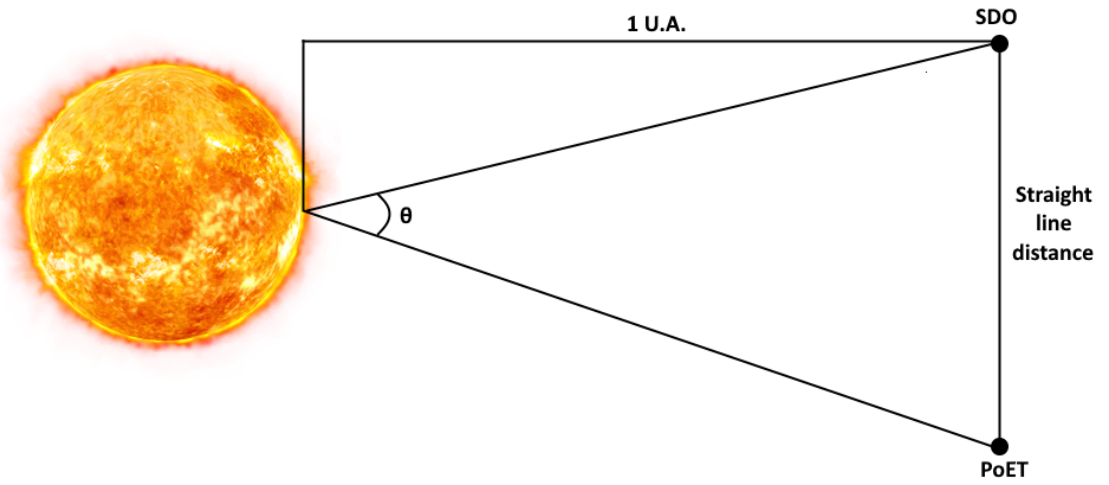


FIGURE 4.5: Angle difference created between different of points of view coming from SDO and PoET.

This figure allows to calculate the angle θ using equation 4.6.

$$\theta = 2\tan^{-1}\left(\frac{\text{Straight line distance}/2}{1\text{ U.A.}}\right) \tag{4.6}$$

The values obtained for this where maximum distance $= 6543.616km$ and minimum distance $= 3478.966km$. Therefore, the difference of angles obtained where maximum angle $= 9.02231$ arc-seconds and minimum angle $= 4.79678$ arc-seconds, with a difference between them of difference angle $= 4.22552$ arc-seconds. One effect that was not explored in this work, and left for future analysis, is the impact of the aberration of the atmosphere, which can

affect the quality of observations, due to distortions in apparent position of celestial objects.

After these transformations, we can now finish the project with verification of calculations such as graphics of the changes of azimuth and altitude over one or two years and also determination of the velocity of the rotation of the Sun, to check if the sunspots are well identified on a period of 5-10 days.

Chapter 5

Results

After doing all the research, this part consists in presenting the results of the code and confirming what was obtained earlier, such as the identification of the Sun regions and also calculating the velocity of rotation of the Sun. Also, when transforming the coordinates, there were values of azimuth, altitude and orientation angle that were obtained from a python module. We need to check if these values are correct, otherwise all the transformations made will be miscalculated and lead to a wrong pointing of the telescope.

5.1 Identification of Sun regions

As seen before, the main objective of this project was to identify regions in the Sun, regions such as sunspots and faculae. Sunspots consist of umbrae and penumbrae, and that separation was also made possible with the introduction of thresholds that would change considering what we wanted to identify. Sunspots per se were identified using the main code of the project, with the changing of parameters from the code that came from another project - (Carvalho et al., 2020). This identification was made for the resolution 1024x1024, that was used for faster tests, and 4096x4096, that was used to get the better quality in the images, as we can see in figure 5.1.

As we can see, there are not many differences between the images. The resolution 1024x1024 seems to have more defined contours but that just happens because the image has lower resolution, so the contour is thicker than the ones on the resolution 4096x4096. This shows that for a quick run to just test the parameters or see where the sunspots are on a specific day, the resolution 1024x1024 is better, because the result is almost the same, but it is much faster as seen on Chapter 2. On the other hand, if we want a better

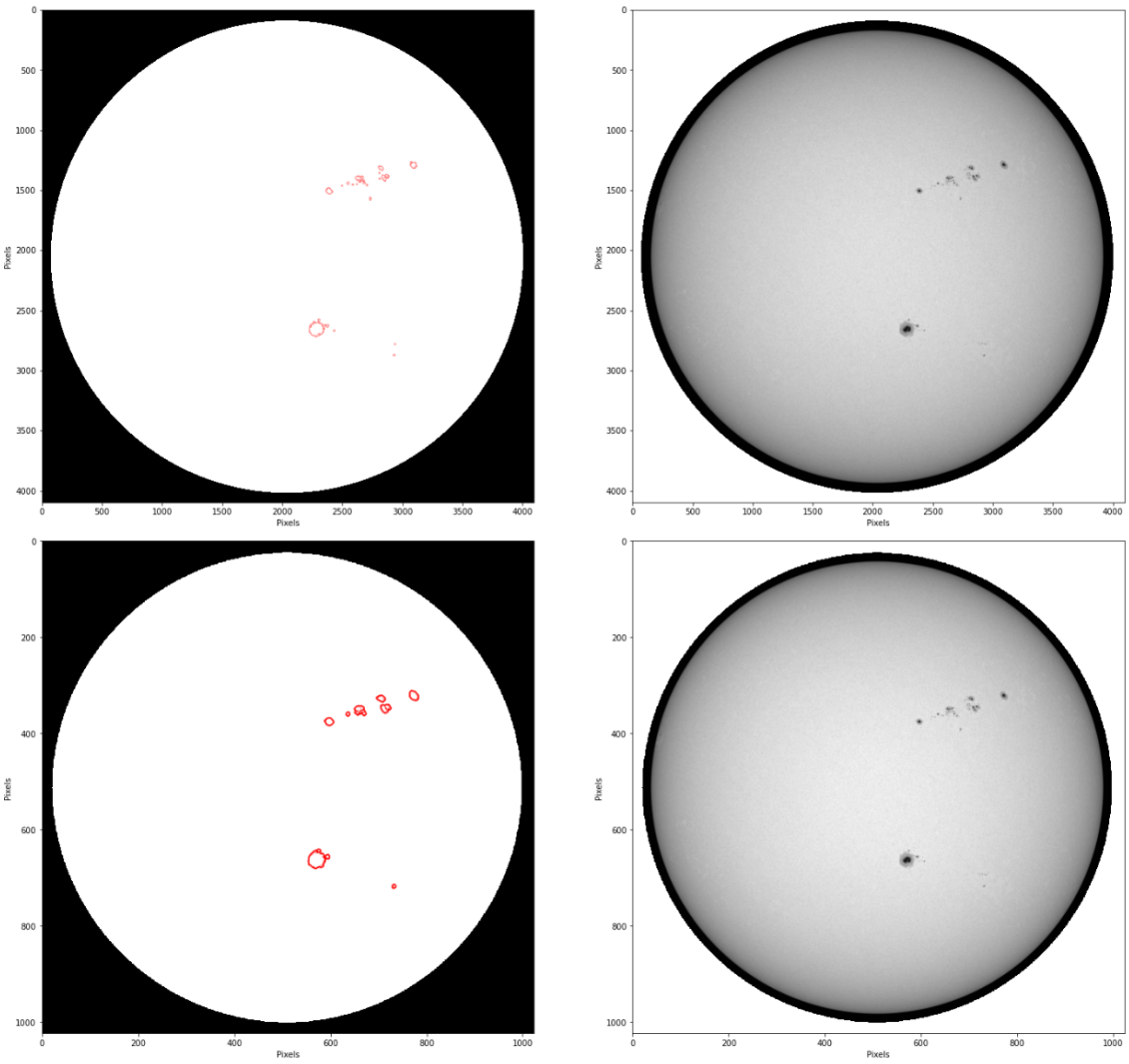


FIGURE 5.1: Identification of sunspots obtained for resolutions 1024x1024 and 4096x4096, up and bottom, respectively.

resolution, the 4096x4096 image would be better to choose, as there is a part of the code where a sunspot can be picked to get a closer image of what it looks like. The data from the FITS file is in a matrix of 4096x4096, so to cut the image where a sunspot was identified only works using that resolution. In figure 5.2 is shown each individual sunspot that was identified in figure 5.1.

As we can see, the sunspots are well identified, but now we can separate the umbra and penumbra and just identify each one. This was done by changing the thresholds as referred in Chapter 3 and the results obtained where the ones shown in figure 5.3.

If we zoom in on the first sunspot, the identification seems correct, with the smaller part being identified for the umbra, and the outside part being identified for the penumbra. As seen before, the penumbra is the sunspot with the umbra subtracted, so it makes

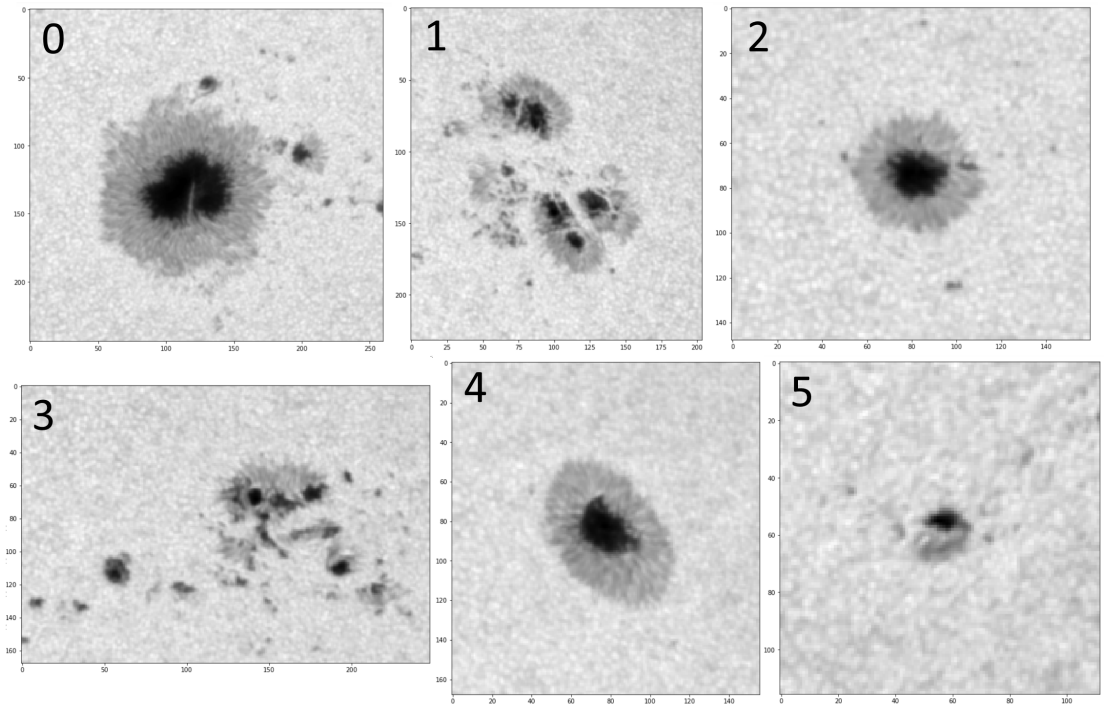


FIGURE 5.2: Each of the 6 individual sunspots that were identified using the image from 23rd of May 2023.

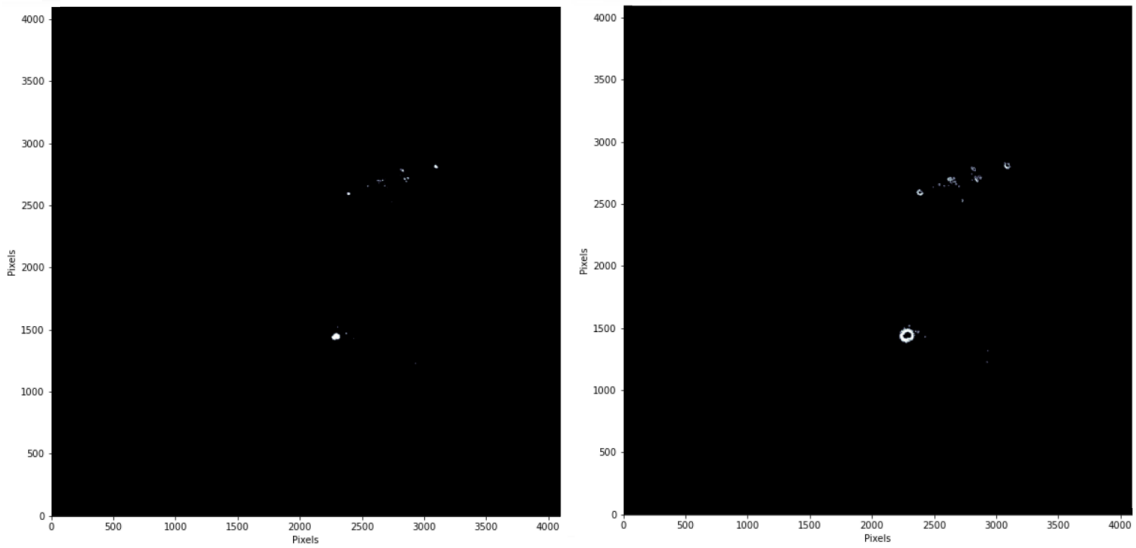


FIGURE 5.3: Identification of the Umbra on the left and Penumbra on the right.

sense that the inside part of the sunspot is not identified. In figure 5.4 we can check this zoom, with the umbra identified on the left and the penumbra on the right, and on the bottom the real image from the TIFF file.

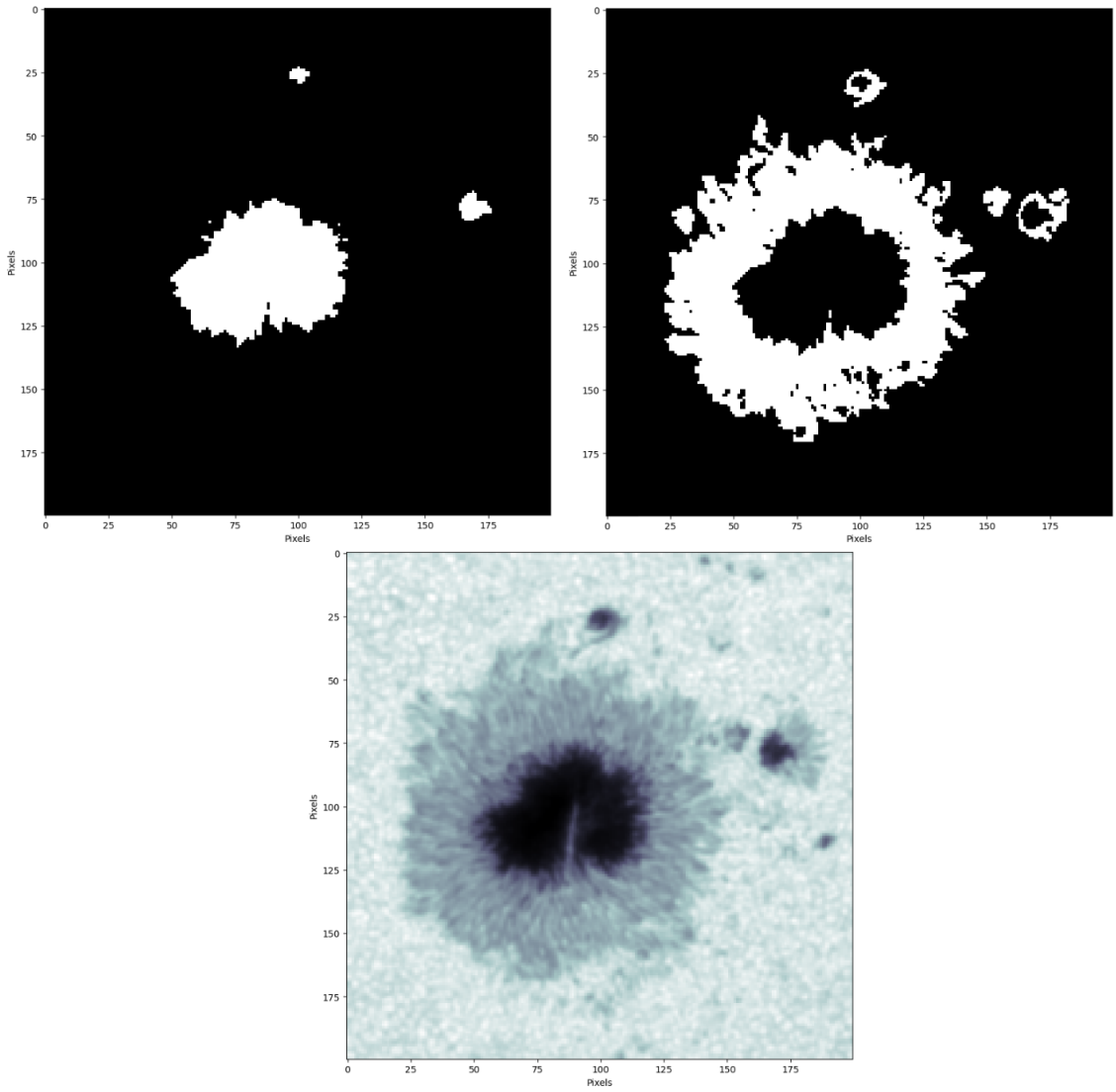


FIGURE 5.4: Zoom in on an individual sunspot identifying the umbra on the left and the penumbra on the right.

As we can see the left part corresponds to the darker region on the sunspot, and the brighter region to the right part. This means that the identification was a success, although it could have been even better. The umbra should fit inside the penumbra, but that is not the case. This could have happened because of a wrong value of threshold and, even tho the better resolution we can get is 4096x4096, if it was higher it could help with the identification. Also, there are pixels that are really close in the value that separates the umbra from the penumbra, so some of them can be lost to one image and some to another.

Despite of that, the identification still worked well and now we can calculate the area in pixels of each of the sunspot. These values are all shown in table 5.1. The index of each of the sunspot corresponds to the numbers shown in figure 5.2.

Index of the Spot	Coordinate x of the Center	Coordinate y of the Center	Area of the Spot	Area of Umbra + Penumbra	Area of the Umbra	Area of the Penumbra
0	2286.37	1443.03	10498	10498	2754	7744
1	2838.16	2732.07	2211	2211	750	1461
2	2389.4	2591.26	1616	1616	406	1210
3	2641.03	2670	2304	2308	435	1873
4	3089.11	2806.05	1627	1627	574	1053
5	2930.01	1255.55	120	120	32	88

TABLE 5.1: Table with the characteristics of the sunspots such as x and y coordinates in the Sun and area in pixels.

As expected, the areas obtained for each sunspot is equal to the area obtained for the umbra + penumbra, except one of them. This happens because the identification can have some errors in some of the pixels detected, so the areas do not add up to be the same, but the errors are relatively small, in this case it failed for 4 pixels.

5.2 Verification of the coordinates transformation

After the identification of the sunspots were done, we then wanted to automate the pointing of the telescope at them. This was done by getting the location of the center of a sunspot in pixels from 0 to 4096 and converting that to right ascension and declination. It is easier for the telescope to point at an object knowing its altitude and azimuth. There is a module on astropy named *SkyCoord* that converts right ascension and declination into azimuth and altitude. After this conversion of the position of the sunspots, we can then obtain the values of azimuth and altitude of the Sun using another module of astropy, as previously explained. It is also more intuitive to check if these values are correct, as we know how the altitude and position relative to North of the Sun evolves during a day. To do that, a graph was made for the evolution of these angles in a day's time, in three different locations - Porto (coordinates of CAUP), the North Pole (coordinates (90º,0º)), and Chile (coordinates of the VLT). The days chosen were the Equinox of March (20th of March of 2023) and both of the Solstices, as the altitude of the Sun will change from higher on the Summer Solstice (21st of June of 2023) to lower on the Winter Solstice (21st

of December of 2022), considering the North Hemisphere. The Solstices change around, to Winter in June and Summer in December, when considering the location at Chile. All of the graphics obtained are represented in figures 5.5, 5.6 and 5.7, corresponding respectively to Porto, North Pole and Chile, and with the order Winter Solstice, March Equinox and Summer Solstice from left to right. The September Equinox was excluded from here, because it is the same as the March Equinox in terms of altitude and azimuth.

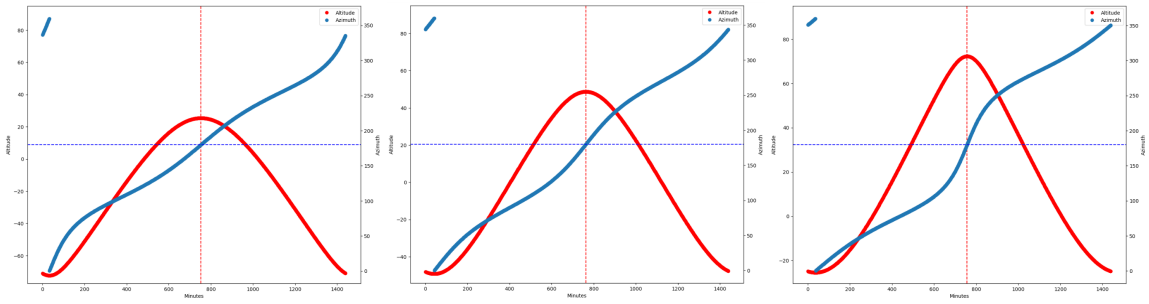


FIGURE 5.5: Graphics of Azimuth and Altitude evolution in a day in Porto, with the red line on the higher altitude of the Sun and the blue line in the Azimuth equal to 180º.

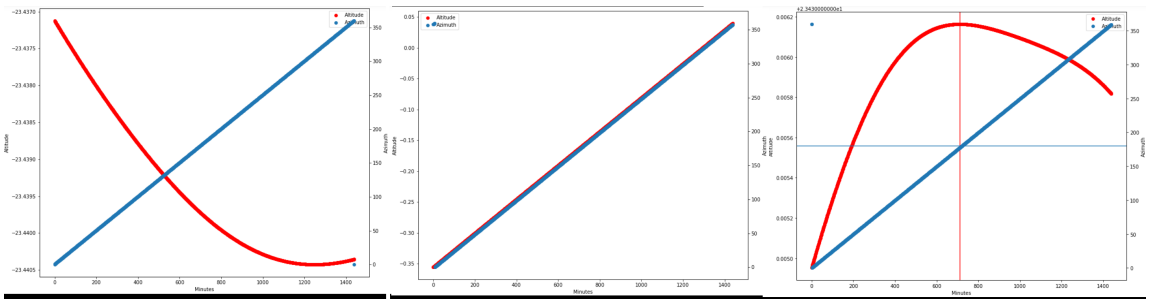


FIGURE 5.6: Graphics of Azimuth and Altitude evolution in a day in North Pole, with the red line on the higher altitude of the Sun and the blue line in the Azimuth equal to 180º.

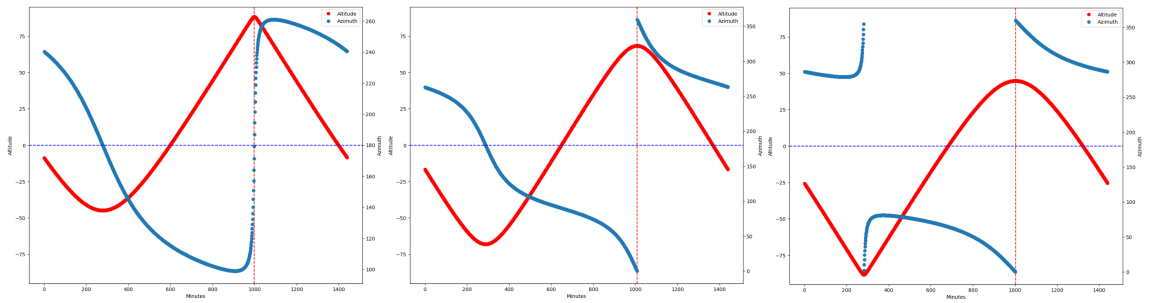


FIGURE 5.7: Graphics of Azimuth and altitude evolution in a day in Chile, with the red line on the higher altitude of the Sun and the blue line in the Azimuth equal to 180º.

Note that the dotted red and blue lines do not appear in the Winter Solstice and March Equinox at the North Pole. This happens because the maximum altitude during those

days do not interest us, as the Sun is below the horizon on the first case and at the horizon on the second case. It appears on the third image as it is the Summer Solstice, so the Sun has a maximum altitude at the solar noon, even though its altitude is almost constant.

About the graphs in general, the x axis starts starts counting the minutes from midnight on the said day and ends at 23h59. This means that the middle of the day is around the 700 minutes mark. Just note that Chile is 4 or 5 hours behind Portugal's time, so the solar noon there happens at around 1000 minute mark, considering GMT0. As we can see in the graphics that correspond to the location of CAUP in Porto, the maximum height of the Sun (marked with the red line) happens around those values and the maximum corresponds to the solar noon. Comparing each of the graphs, we can see that the height of the Sun also changes from winter to summer, as expected. There is also a horizontal blue line that marks the Azimuth position as 180º. As the Azimuth angle is measured starting North, this value being 180º indicates where the South is on the sky (when we look at the North, South is in the back of our heads). So, as we can see, when the height is maximum, the Azimuth is equal to 180º. In the case of the North Pole, even though the Sun's height does not change, it still is at the maximum height at Solar Noon and when Azimuth is 180º in the Summer Solstice. We can also see that the Summer Solstice has a positive height, as the Sun is in the sky for the entire day and, on the other hand, the Sun has a negative height during the Winter Solstice, as it is night during a few months. In case of Chile, as it is on the Southern Hemisphere, the Sun gets higher from June to December, because the Summer Solstice in there happens in December. The rest of the graphs look more or less the same, even with the change in location. All of this confirms that these values are obtained correctly from the astropy module, so we can use them for our calculations in the coordinates transformation.

There is also another angle that was considered in the transformation of coordinates, that was the orientation angle, as explained in Chapter 4. To check if this angle was correct, it was done an evolution of this value in a year (2022), in the same three locations, as represented in figure 5.8.

As we can see, in the North Pole the Sun rotates around the values of -23,6º and +23,6º, during the year. This happens because the tilt of the Earth's axis is around this value and note that they are obtained during the Summer and Winter Solstices. On the Equinoxes, this value reaches zero because during those days, the Earth's axis is aligned with the Sun, so the equator of the Sun is aligned with Earth's horizon. In the case of Porto and Chile,

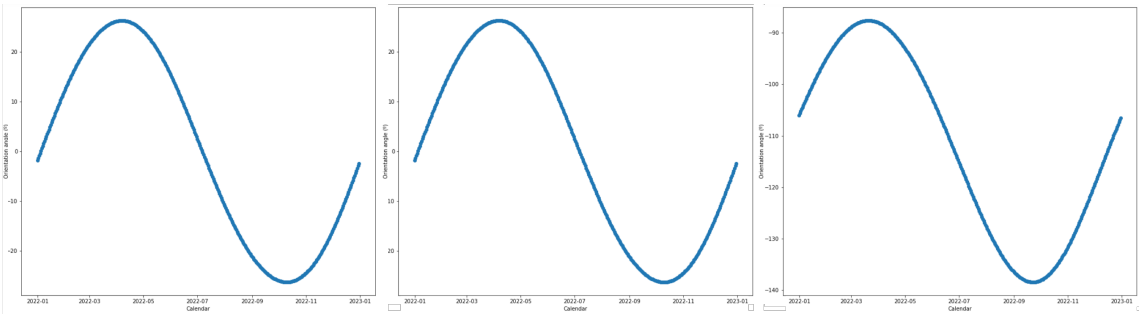


FIGURE 5.8: Graphics of the orientation angle in a year in Porto, North Pole and Chile, from left to right.

the value still changes from -23,6º to +23,6º, but with a translation for both cases. As we saw before, Chile is in the Southern Hemisphere, so sky is viewed from another perspective, almost as if it was upside down when compared to how we see it in Porto. With that in mind, we can check that the orientation angle calculated for Chile is more than 90 degrees, which is consistent with the position of the Sun's North Pole when observing it from Chile. With all these graphs it is safe to conclude that the values obtained from python modules are correct, so the coordinates transformations are well-made as well.

5.3 Sun's rotation velocity

The identification of sunspots allows to get images from any given day and, with that knowledge, calculate where a sunspot will be in the next few days. This was done with the calculation of the velocity of the Sun's rotation, using sunspots identification from 5 consecutive days. The dates chosen where when the Sun had a good solar activity and were from 13th to 17th of December 2013. These 5 days were made just to understand the evolution of the sunspots as we can see in figure 5.9.

To make the calculations, there were used other 3 sets of 5 dates that were 15th to 19th of August 2022, 15th to 19th of January 2023 and 11th to 15th of February 2023. These dates were chosen because they allowed to get sunspots in different latitudes of the Sun, as the velocity of rotation of the Sun changes with latitude according to equation 5.1 (Snodgrass and Ulrich, 1990).

$$w = A + B\sin^2(\varphi) + C\sin^4(\varphi) \tag{5.1}$$

Where w is the angular velocity in degrees per day and φ is the latitude in the Sun. A is the equator's angular velocity, B and C are constants controlling the decrease in velocity

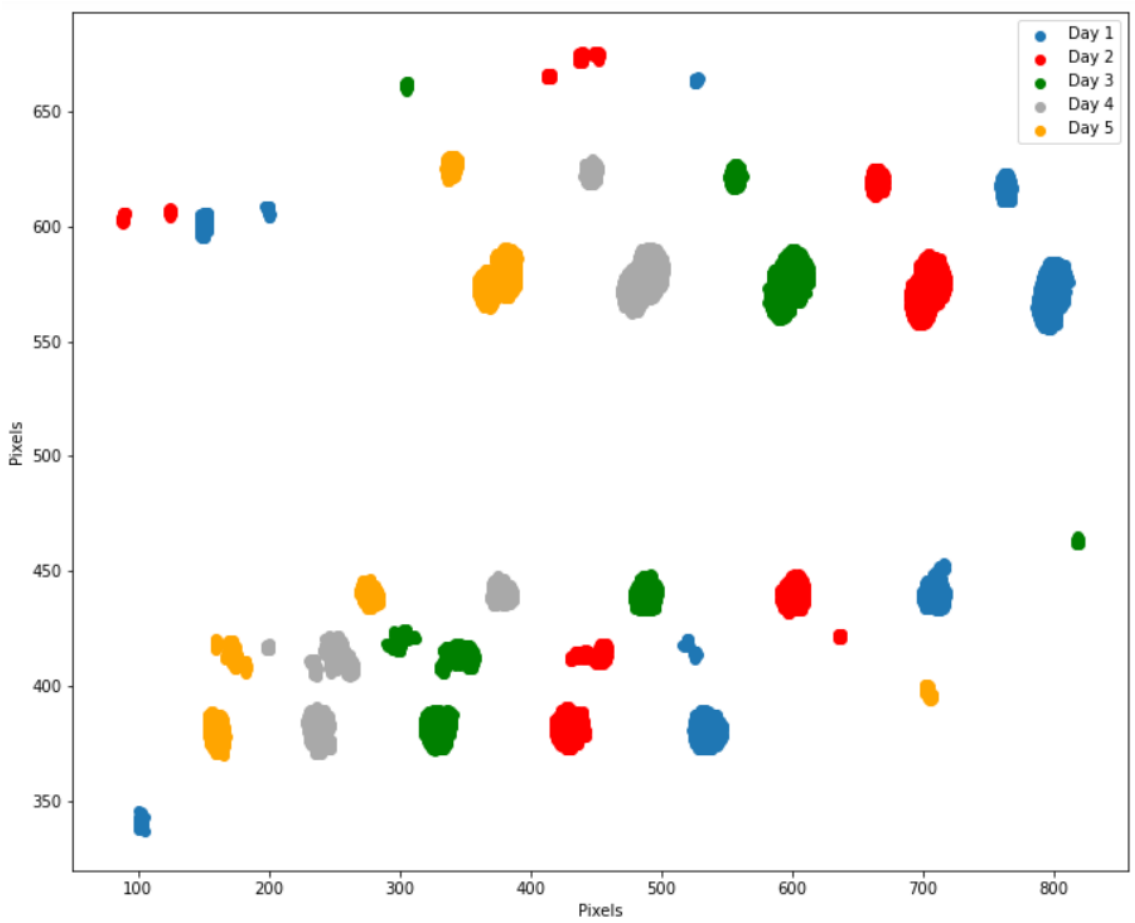


FIGURE 5.9: Evolution of sunspots from 13th to 17th of December 2013.

with increasing latitude, as the velocity of rotation of the Sun on the poles ($\pm$ 90º) is zero. The values of A, B and C can change depending on the measurements techniques used and period of time studied (Beck, 2000). A set of values accepted as of now is (Snodgrass and Ulrich, 1990):

1. A = 14.713 º/day;
2. B = -2.396 º/day;
3. C = -1.787 º/day.

After this we can calculate the velocity of the rotation of the Sun in kilometers per second, instead of degrees per day. To do that, we need to convert a day into seconds and the degrees to kilometers using equation 5.2.

$$v = \frac{w\pi}{180} \frac{R_{Sun} \cos(\varphi)}{24 \times 60 \times 60} \tag{5.2}$$

Where w and φ are the same as equation 5.1 and $R_{Sun} = 696340$ km. The $24 \times 60 \times 60$ converts a day into a second and we want to get the velocity of a section with a given latitude, so that is done by multiplying its cosine with the radius of the Sun. With this we have the velocity of rotation of the Sun in kilometers per second which is around 2 km/s in the equator and approaches to 0 km/s near the solar Poles.

Using the three dates mentioned above, we calculated the velocity of the Sun's rotation, by calculating the distance between consecutive sunspots and dividing by the time between them which is 24 hours. This calculations were made for different sets of latitudes and compared with the expected curve, as we can observe in figure 5.10.

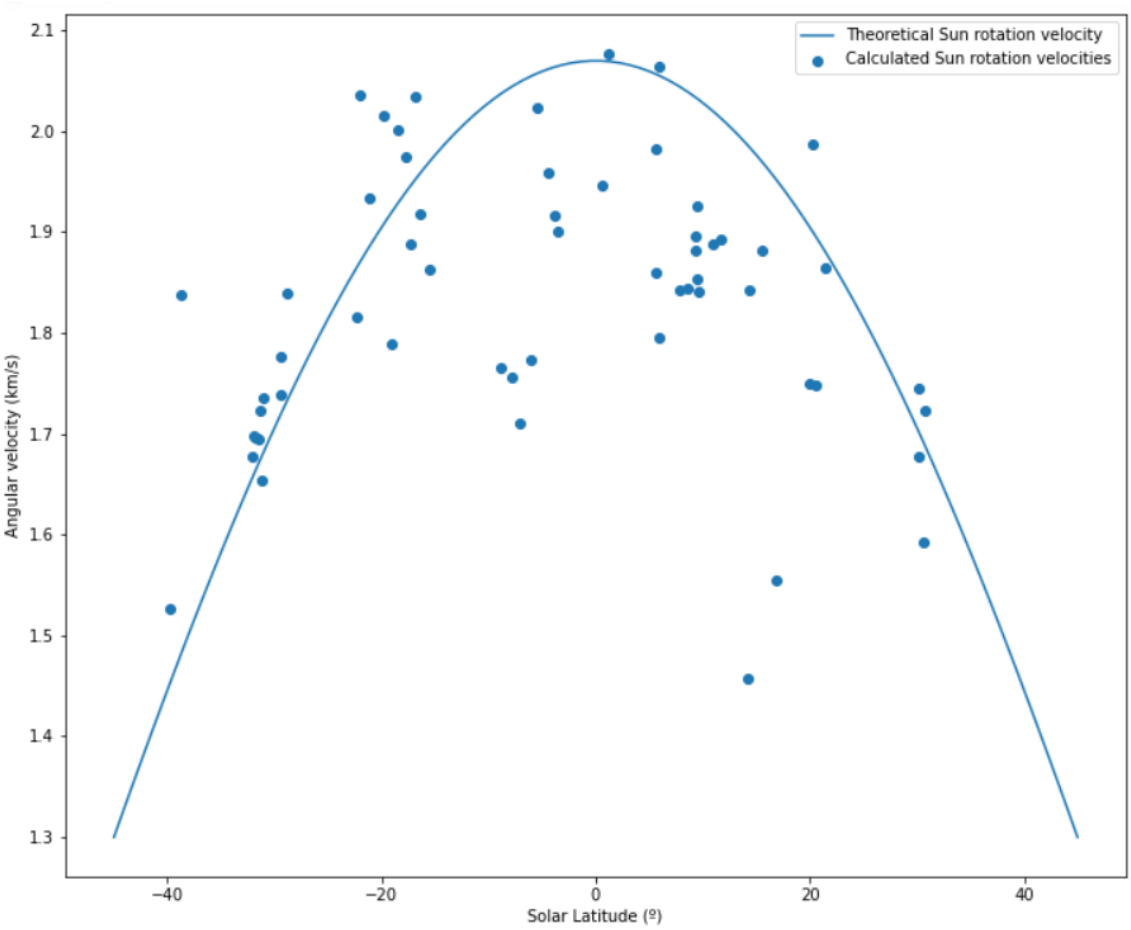


FIGURE 5.10: Velocity of rotation of the Sun calculated and compared with the theoretical curve.

As we can note, the velocities for latitudes between -15º and +15º do not follow the expected curve that well. This may happen because of the way the distance between sunspots is calculated. When the sunspots are closer to the equator, they will change more in position each day, as the velocity of the rotation of the Sun is higher when getting

far away from the Poles. However, the distances are more difficult to measure when the sunspot is closer to the equator, so the value obtain does not reflect on the curve.

Chapter 6

Conclusion

In conclusion, the main goal of this project was accomplished, as receiving an image of the Sun allows to obtain its regions, such as sunspots (umbra + penumbra) and faculae. This identification is crucial for the work of the PoET telescope that is under construction and requires the position of said regions to be able to study them and understand more about the Sun. This study is very important for the future work in astronomy, as we want to know more about other Earths, looking at the Sun as a basis to understand stellar physics and separate phenomena that happen on its atmosphere and surface from the finding of exoplanets. For example the effect of a sunspot on the Sun's spectrum may be confused with a discovery of a exoplanet, so this work will allow us to understand more of those effects on planet detection.

Firstly, we needed a way to get images from the Sun to start this research. We were able to use images from the SDO, an instrument that is studying solar activity and takes images from the Sun. Using the filter HMI and a code initially created by researchers from Coimbra, we were able to obtain a successful identification of regions present in the Sun. In this case we managed to obtain identifications for sunspots, separate them into umbra and penumbra, identifying each of them and also identify faculae, a brighter and hotter region of the Sun. However, the identification of the faculae still needs further research, as the code is not well optimized and we could not use images from previous days as done for the sunspots.

After finding out where these regions are in an image, we used a clustering method that still needs to be optimized in order to count how many sunspots identified. However, that number can be input manually to ease the clustering method and then identify where is the centroid of each sunspot. After those centroids coordinates where obtained,

there was made a conversion from the Cartesian to the celestial, as we want to point the telescope at the sunspots. That conversion was done by trigonometry and allowed to obtain the declination and right ascension of the sunspot. Those two coordinates are very helpful to know where an object is in the sky at a given time and location. However, it is easier for a telescope to point using altitude and azimuth, that correspond respectively to the declination and right ascension. We did this conversion and then finding out where the Sun is located, we can add the values of the sunspot as a translation from the center of the Sun. With this we were able to conclude where a sunspot is on the sky and, when PoET is built, point him at the regions of interest to study. These values of altitude and azimuth are well calculated, as they check out with the expected, i.e. the altitude of the Sun is higher on the Summer Solstice and lower on the Winter Solstice, considering the North Hemisphere. We also did that calculation in the North Pole, obtaining the Sun above the horizon for the full day during the Summer, and bellow the horizon (night) during the Winter. In the case of the azimuth, we can conclude that it was also right, as the Sun was at to 180^{o} during the Solar Noon, as the Sun is located South in the North Hemisphere, and the azimuth is the angle between the North and the position of the object.

To finalize, after separating the sunspot into umbra and penumbra, we made a calculation of their area. The areas checked out with the umbra+penumbra being equal to the area of the corresponding sunspot. Also, using the identification of the sunspots part, we can predict where a sunspot is going to be in a few days. To check if this part works, there was made an evolution of five consecutive days and the calculation of the Sun's rotation velocity. The velocity obtained followed the expected, near the equator it is the maximum, at around 2 km/s and near the poles it approaches to 0 km/s. With this values, knowing the velocity of the Sun in a given latitude and the position of a sunspot, we can calculate where it is going to be evolving to in the next days, easing the study of the Sun's physics.

Appendix A

Appendix containing the codes used

Access to the full codes used during this project: https://gitfront.io/r/SilvaLino201/vjVGeS239h38/Tese-Mestrado/

Bibliography

R. E. Allen and S. Lidström, "Life, the universe, and everything—42 fundamental questions," *Physica Scripta*, vol. 92, no. 1, p. 012501, nov 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1804.08730 [Cited on page 1.]

M. Mayor and D. Queloz, "A Jupiter-mass companion to a solar-type star," , vol. 378, no. 6555, pp. 355–359, Nov. 1995. [Cited on page 1.]

J. Schneider, C. Dedieu, P. Le Sidaner, R. Savalle, and I. Zolotukhin, "Defining and cataloging exoplanets: the exoplanet. eu database," *Astronomy & Astrophysics*, vol. 532, p. A79, 2011. [Cited on pages 1 and 6.]

J. F. Kasting, "Exoplanetary habitability," in *ExoFrontiers: Big questions in exoplanetary science*. IOP Publishing Bristol, UK, 2021, pp. 22–1. [Cited on page 2.]

J. M. Jenkins, J. D. Twicken, N. M. Batalha, D. A. Caldwell, W. D. Cochran, M. Endl, D. W. Latham, G. A. Esquerdo, S. Seader, A. Bieryla, E. Petigura, D. R. Ciardi, G. W. Marcy, H. Isaacson, D. Huber, J. F. Rowe, G. Torres, S. T. Bryson, L. Buchhave, I. Ramirez, A. Wolfgang, J. Li, J. R. Campbell, P. Tenenbaum, D. Sanderfer, C. E. Henze, J. H. Catanzarite, R. L. Gilliland, and W. J. Borucki, "Discovery and validation of kepler-452b: A 1.6 r super earth exoplanet in the habitable zone of a g2 star," *The Astronomical Journal*, vol. 150, no. 2, p. 56, jul 2015. [Online]. Available: https://dx.doi.org/10.1088/0004-6256/150/2/56 [Cited on page 2.]

A. Suárez Mascareño, M. Damasso, N. Lodieu, A. Sozzetti, V. Béjar, S. Benatti, M. Zapatero Osorio, G. Micela, R. Rebolo, S. Desidera *et al.*, "Rapid contraction of giant planets orbiting the 20-million-year-old star v1298 tau," *Nature Astronomy*, vol. 6, no. 2, pp. 232–240, 2022. [Cited on page 2.]

N. C. Santos, S. C. Barros, O. D. Demangeon, and J. P. Faria, "Detection and characterization methods of exoplanets," in *Oxford Research Encyclopedia of Planetary Science*, 2020. [Cited on pages 3, 6, and 8.]

N. Giordano, *College physics: reasoning and relationships*. Cengage Learning, 2012. [Cited on page 5.]

D. Charbonneau, T. M. Brown, D. W. Latham, and M. Mayor, "Detection of planetary transits across a sun-like star," *The Astrophysical Journal*, vol. 529, no. 1, p. L45, 1999. [Cited on page 7.]

R. V. Baluev, "Enhanced models for stellar doppler noise reveal hints of a 13-year activity cycle of 55 cancri," *Monthly Notices of the Royal Astronomical Society*, vol. 446, no. 2, pp. 1493–1511, 2015. [Cited on page 7.]

M. Damasso and F. Del Sordo, "Proxima centauri reloaded: Unravelling the stellar noise in radial velocities," *Astronomy & Astrophysics*, vol. 599, p. A126, 2017. [Cited on page 7.]

S. Czesla, K. Huber, U. Wolter, S. Schröter, and J. Schmitt, "How stellar activity affects the size estimates of extrasolar planets," *Astronomy & Astrophysics*, vol. 505, no. 3, pp. 1277–1282, 2009. [Cited on page 8.]

D. H. Forgan, "On the possibility of detecting class a stellar engines using exoplanet transit curves," *arXiv preprint arXiv:1306.1672*, 2013. [Cited on page 8.]

S. Aigrain, F. Favata, and G. Gilmore, "Characterising stellar micro-variability for planetary transit searches," *Astronomy & Astrophysics*, vol. 414, no. 3, pp. 1139–1152, 2004. [Cited on page 8.]

R. J. Bray and R. E. Loughhead, "Sunspots," 1 1964. [Online]. Available: https://www.osti.gov/biblio/5642144 [Cited on page 9.]

C. Muller, "The carrington solar flares of 1859: consequences on life," *Origins of Life and Evolution of Biospheres*, vol. 44, no. 3, pp. 185–195, 2014. [Cited on page 9.]

D. F. Webb and T. A. Howard, "Coronal mass ejections: Observations," *Living Reviews in Solar Physics*, vol. 9, no. 1, pp. 1–83, 2012. [Cited on page 9.]

A. P. Mitra, *Ionospheric effects of solar flares*. Springer, 1974, vol. 46. [Cited on page 9.]

V. Airapetian, R. Barnes, O. Cohen, G. Collinson, W. Danchi, C. Dong, A. Del Genio, K. France, K. Garcia-Sage, A. Glocer *et al.*, "Impact of space weather on climate and habitability of terrestrial-type exoplanets," *International Journal of Astrobiology*, vol. 19, no. 2, pp. 136–194, 2020. [Cited on page 9.]

L. Serrano, S. Barros, M. Oshagh, N. Santos, J. Faria, O. Demangeon, S. Sousa, and M. Lendl, "Distinguishing the albedo of exoplanets from stellar activity," *Astronomy & Astrophysics*, vol. 611, p. A8, 2018. [Cited on page 10.]

J. M. Andersen and H. Korhonen, "Stellar activity as noise in exoplanet detection–ii. application to m dwarfs," *Monthly Notices of the Royal Astronomical Society*, vol. 448, no. 4, pp. 3053–3069, 2015. [Cited on page 10.]

A. Vanderburg, P. Plavchan, J. A. Johnson, D. R. Ciardi, J. Swift, and S. R. Kane, "Radial velocity planet detection biases at the stellar rotational period," *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 4, pp. 3565–3573, 2016. [Cited on page 10.]

X. Dumusque, S. Udry, C. Lovis, N. C. Santos, and M. Monteiro, "Planetary detection limits taking into account stellar noise-i. observational strategies to reduce stellar oscillation and granulation effects," *Astronomy & Astrophysics*, vol. 525, p. A140, 2011. [Cited on page 11.]

C. J. Schrijver and C. Zwaan, *Solar and stellar magnetic activity*. Cambridge University Press, 2008, vol. 34. [Cited on page 11.]

N. Meunier, M. Desort, and A.-M. Lagrange, "Using the sun to estimate earth-like planets detection capabilities-ii. impact of plages," *Astronomy & Astrophysics*, vol. 512, p. A39, 2010. [Cited on page 11.]

N. Santos, A. Mortier, J. Faria, X. Dumusque, V. Z. Adibekyan, E. Delgado-Mena, P. Figueira, L. Benamati, I. Boisse, D. Cunha *et al.*, "The harps search for southern extra-solar planets-xxxv. the interesting case of hd 41248: stellar activity, no planets?" *Astronomy & Astrophysics*, vol. 566, p. A35, 2014. [Cited on page 11.]

M. D. Fivian, H. S. Hudson, R. P. Lin, and H. J. Zahid, "A large excess in apparent solar oblateness due to surface magnetism," *Science*, vol. 322, no. 5901, pp. 560–562, 2008. [Cited on page 12.]

M. Woolfson, "The origin and evolution of the solar system," *Astronomy Geophysics*, vol. 41, no. 1, pp. 1.12–1.19, 02 2000. [Online]. Available: https://doi.org/10.1046/j.1468-4004.2000.00012.x [Cited on page 12.]

S. Basu and H. Antia, "Helioseismology and solar abundances," *Physics Reports*, vol. 457, no. 5, pp. 217–283, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157307004565 [Cited on page 12.]

E. Pitjeva and E. Standish, "Proposals for the masses of the three largest asteroids, the moon-earth mass ratio and the astronomical unit," *Celestial Mechanics and Dynamical Astronomy*, vol. 103, pp. 365–372, 2009. [Cited on page 12.]

Bonanno, A., Schlattl, H., and Paternò, L., "The age of the sun and the relativistic corrections in the eos," *A&A*, vol. 390, no. 3, pp. 1115–1118, 2002. [Online]. Available: https://doi.org/10.1051/0004-6361:20020749 [Cited on page 12.]

S. W. Stahler and F. Palla, *The formation of stars*. John Wiley & Sons, 2008. [Cited on page 12.]

A. E. Roy and D. Clarke, *Astronomy*. Hilger, 2003. [Cited on page 13.]

D. R. Lide, *CRC handbook of chemistry and physics*. CRC press, 2004, vol. 85. [Cited on page 13.]

L. Miramonti *et al.*, "Solar neutrinos: from their production to their detection," *Proceedings of science 4th school on cosmic rays and astrophysics*, pp. 25–04, 2010. [Cited on page 13.]

J. Wilkinson, "The sun and stars," in *New Eyes on the Sun*. Springer, 2012, pp. 219–236. [Cited on page 13.]

G. Laughlin, P. Bodenheimer, and F. C. Adams, "The end of the main sequence," *The Astrophysical Journal*, vol. 482, no. 1, p. 420, 1997. [Cited on page 13.]

L. Reinders, "Stellar processes and quantum mechanics," *Sun in a Bottle?... Pie in the Sky! The Wishful Thinking of Nuclear Fusion Energy*, pp. 13–17, 2021. [Cited on page 13.]

L. Mestel, "On the theory of white dwarf stars: I. the energy sources of white dwarfs," *Monthly Notices of the Royal Astronomical Society*, vol. 112, no. 6, pp. 583–597, 1952. [Cited on page 13.]

J. M. Lattis, *Between Copernicus and Galileo: Christoph Clavius and the collapse of ptolemaic cosmology*. University of Chicago Press, 2010. [Cited on page 13.]

K. Labitzke and H. Van Loon, "The signal of the 11-year sunspot cycle in the upper troposphere-lower stratosphere," *Space Science Reviews*, vol. 80, no. 3-4, pp. 393–410, 1997. [Cited on page 16.]

P. Bhomwik and D. Nandy, "Prediction of the strength and timing of sunspot cycle 25 reveal decadal-scale space environmental conditions," *Nature Communications*, vol. 9, 12 2018. [Cited on page 16.]

F. Pepe, S. Cristiani, R. Rebolo, N. C. Santos, H. Dekker, A. Cabral, P. Di Marcantonio, P. Figueira, G. Lo Curto, C. Lovis, M. Mayor, D. Mégevand, P. Molaro, M. Riva, M. R. Zapatero Osorio, M. Amate, A. Manescau, L. Pasquini, F. M. Zerbi, V. Adibekyan, M. Abreu, M. Affolter, Y. Alibert, M. Aliverti, R. Allart, C. Allende Prieto, D. Álvarez, D. Alves, G. Avila, V. Baldini, T. Bandy, S. C. C. Barros, W. Benz, A. Bianco, F. Borsa, V. Bourrier, F. Bouchy, C. Broeg, G. Calderone, R. Cirami, J. Coelho, P. Conconi, I. Coretti, C. Cumani, G. Cupani, V. D'Odorico, M. Damasso, S. Deiries, B. Delabre, O. D. S. Demangeon, X. Dumusque, D. Ehrenreich, J. P. Faria, A. Fragoso, L. Genolet, M. Genoni, R. Génova Santos, J. I. González Hernández, I. Hughes, O. Iwert, F. Kerber, J. Knudstrup, M. Landoni, B. Lavie, J. Lillo-Box, J. L. Lizon, C. Maire, C. J. A. P. Martins, A. Mehner, G. Micela, A. Modigliani, M. A. Monteiro, M. J. P. F. G. Monteiro, M. Moschetti, M. T. Murphy, N. Nunes, L. Oggioni, A. Oliveira, M. Oshagh, E. Pallé, G. Pariani, E. Poretti, J. L. Rasilla, J. Rebordão, E. M. Redaelli, S. Santana Tschudi, P. Santin, P. Santos, D. Ségransan, T. M. Schmidt, A. Segovia, D. Sosnowska, A. Sozzetti, S. G. Sousa, P. Spanò, A. Suárez Mascareño, H. Tabernero, F. Tenegi, S. Udry, and A. Zanutta, "ESPRESSO at VLT. On-sky performance and first results," , vol. 645, p. A96, Jan. 2021. [Cited on page 17.]

F. Pepe, P. Molaro, S. Cristiani, R. Rebolo, N. Santos, H. Dekker, D. Mégevand, F. Zerbi, A. Cabral, P. Di Marcantonio *et al.*, "Espresso: The next european exoplanet hunter," pp. 8–20, 2014. [Cited on page 17.]

S. Carvalho, S. Gomes, T. Barata, A. Lourenço, and N. Peixinho, "Comparison of automatic methods to detect sunspots in the coimbra observatory spectroheliograms," *Astronomy and Computing*, vol. 32, p. 100385, 2020. [Cited on pages 18, 21, and 53.]

P. H. Scherrer, J. Schou, R. Bush, A. Kosovichev, R. Bogart, J. Hoeksema, Y. Liu, T. Duvall, J. Zhao, A. Title *et al.*, "The helioseismic and magnetic imager (hmi) investigation for the solar dynamics observatory (sdo)," *Solar Physics*, vol. 275, pp. 207–227, 2012. [Cited on page 21.]

K. Glogowski, M. G. Bobra, N. Choudhary, A. B. Amezcua, and S. J. Mumford, "drms: A python package for accessing hmi and aia data," *Journal of Open Source Software*, vol. 4, no. 40, p. 1614, 2019. [Online]. Available: https://doi.org/10.21105/joss.01614 [Cited on page 27.]

W. D. Pesnell, B. J. Thompson, and P. Chamberlin, *The solar dynamics observatory (SDO)*. Springer, 2012. [Cited on page 48.]

H. B. Snodgrass and R. K. Ulrich, "Rotation of Doppler Features in the Solar Photosphere," , vol. 351, p. 309, Mar. 1990. [Cited on pages 60 and 61.]

J. G. Beck, "A comparison of differential rotation measurements–(invited review)," *Solar physics*, vol. 191, no. 1, pp. 47–70, 2000. [Cited on page 61.]