

1 Introduction

This project is concerned with the identification and studying of the solar active regions, which are concentrations of relatively high magnetic field on the solar surface. One of the types of active regions is a sun-spot. Sun-spots (when they are present) are seen as the isolated dark areas on the solar disk when the latter is observed in the visible or IR wavelengths. The other type is faculae which are seen in the magnetic field images of the Sun. In actuality, there are more types of active regions, but for the purposes of the project this basic division into spots and faculae will suffice.

In this project we will work with the high-resolution (4096×4096 pixels) intensity images and magnetograms of the solar disk obtained with the Helioseismic and Magnetic Imager (HMI) onboard Solar Dynamics Observatory (SDO). An example of these data is shown in Fig. 1.1. Each pixel of the intensity image contains the intensity (in arbitrary units) at

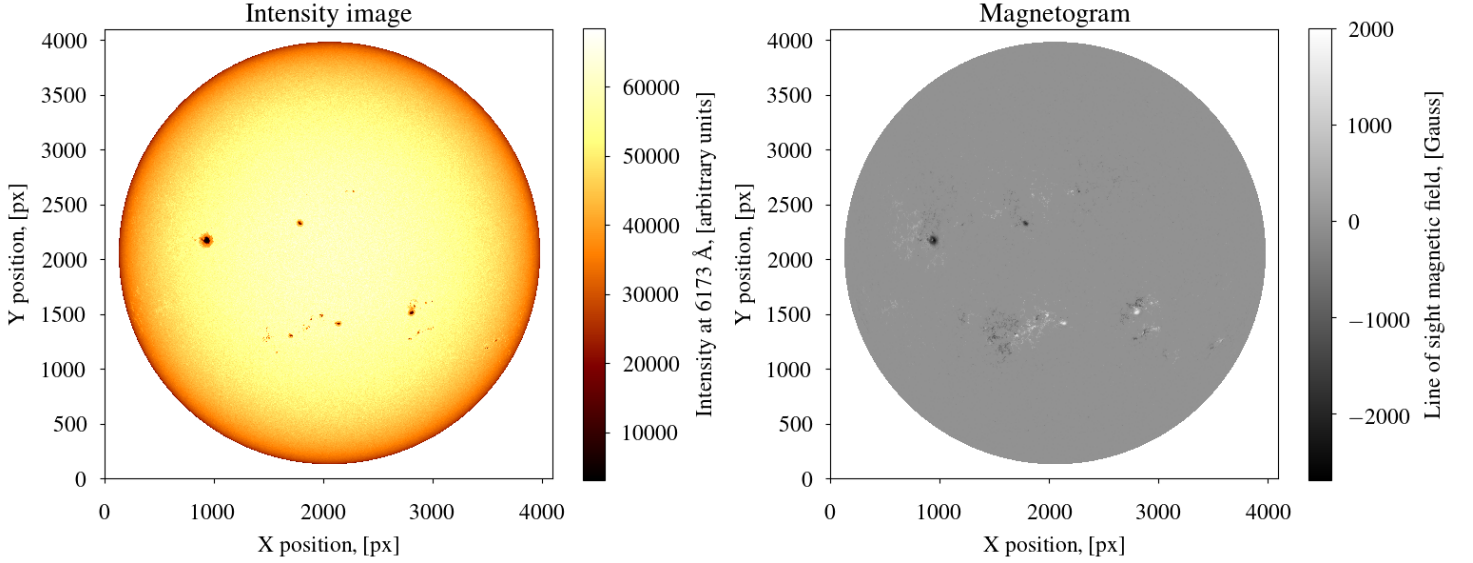


Figure 1.1: The HMI/SDO intensity image (left panel) and magnetogram (right panel) for 16.11.2013.

6173 Å wavelength and each pixel of the magnetogram contains the line-of-sight component of magnetic field in Gauss. Since both spots and faculae manifest themselves in magnetograms, but only spots show the darkening in the intensity images, analyzing these two types of data allows to identify spots and faculae on the solar disk.

The core of this project can be boiled down to the following goals:

- see how sun-spots are different from facular structures in terms of their morphology;
- find out what the strengths of typical magnetic fields in faculae and the inner parts of sun-spots are;
- find out what the typical sizes of sun-spots are;

In what follows, the main aspects of the analysis will be demonstrated using the images shown in Fig. 1.1. Where needed the references to `python 3.6.0` functions will be made. It should be mentioned at the outset that we will restrict ourselves by not considering the outer parts of the solar disk ($r/R_{\odot} < 0.953$), the so-called limb, because the analysis is greatly hindered there by the projection effect. The identification of active regions in the images starts with finding the image pixels in which we reliably measure the signal.

2 Intensity images

2.1 Center-to-Limb Variation

Fig. 2.1 shows the scan of the intensity image through the biggest sun-spot within $r = 0.953R_{\odot}$ circumference. The intensity values are stored in a `fits` format file. The access to them as well as to the coordinates of the solar disk center (x_0, y_0) and solar radius R_{\odot} , which we will need in the analysis, can be carried out in the following way:

```
>>> from astropy.io import fits

>>> hdulist_int = fits.open('/path/to/intensity/fits/file')

>>> hdu_int = hdulist_int[0]

>>> I = hdu_int.data # hdu_int.data is a 2D-array
```

```
>>> x0 = hdu_int.header['X0']; y0 = hdu_int.header['Y0']

>>> rsun = hdu_int.header['R_SUN']
```

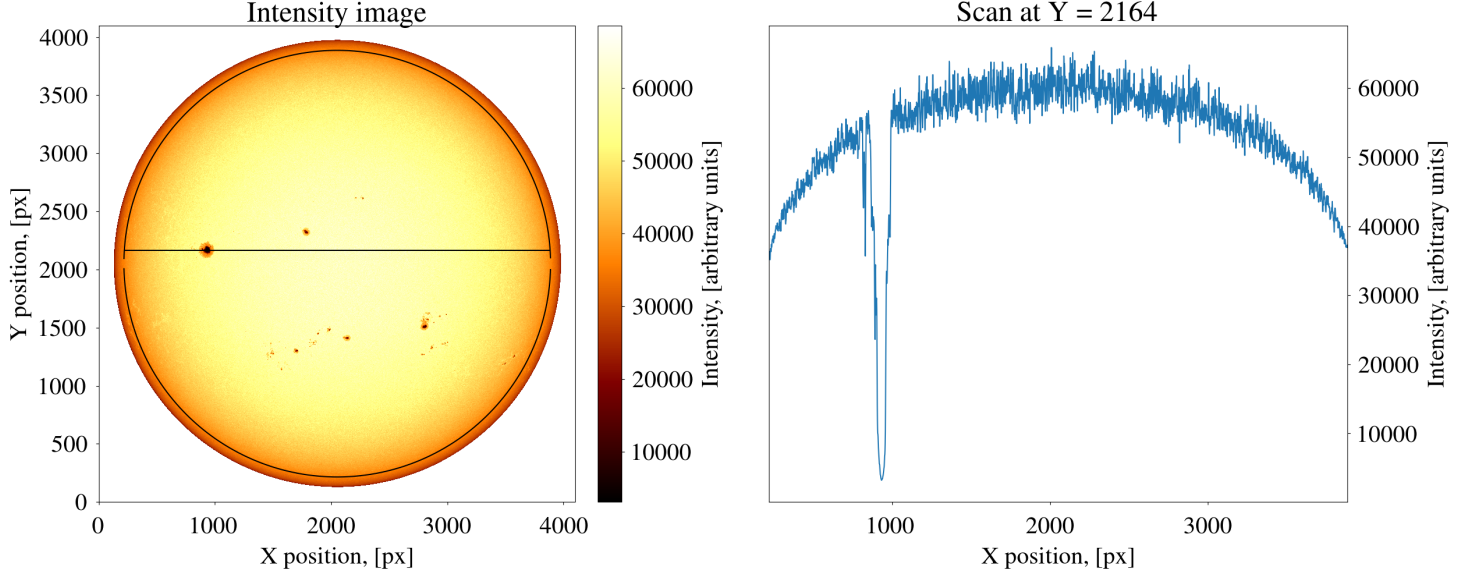


Figure 2.1: Scan of the intensity image (see left panel of Fig. 1.1) within $r = 0.953R_{\odot}$ circumference at the Y-position of the biggest sun-spot.

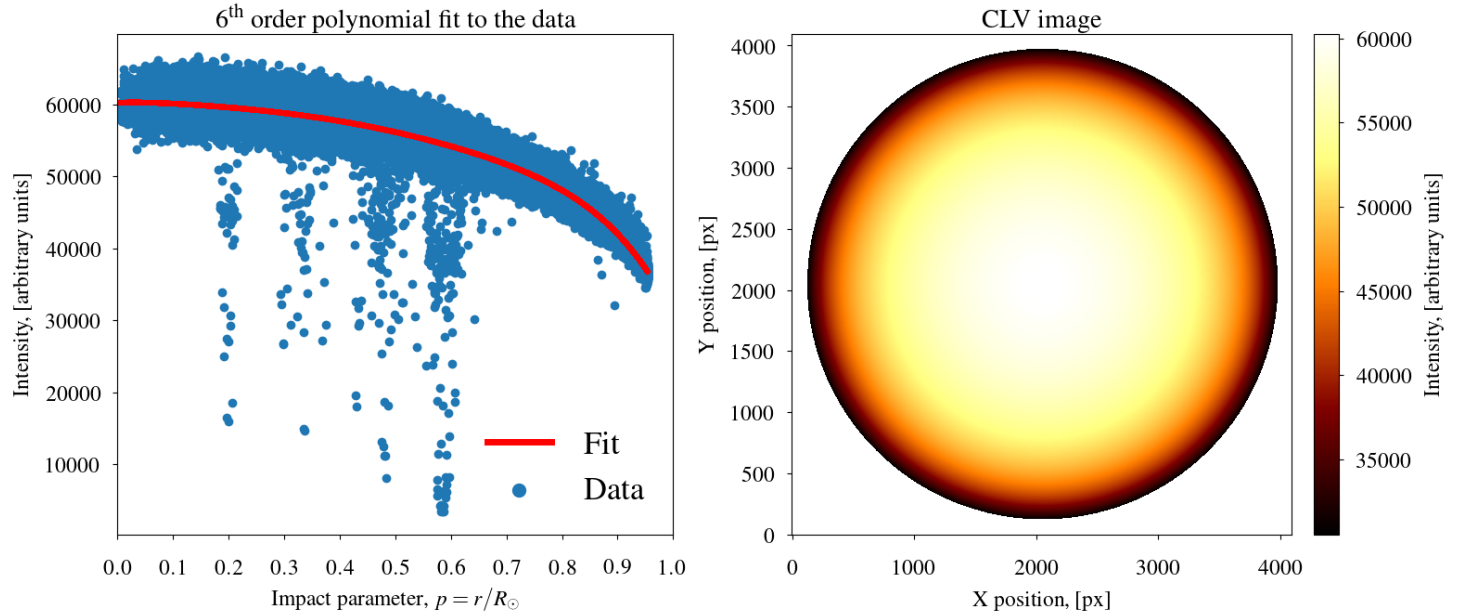


Figure 2.2: **Left panel:** the cloud of data points in the intensity image in the left panel of Fig. 1.1 as a function of the impact parameter and the 6th order polynomial fit to this cloud. **Right panel:** the corresponding CLV image constructed from the fit.

It is seen from Fig. 2.1 that the background intensity decreases as the distance from the disk center increases. This decrease is called Center-to-Limb Variation (CLV) or limb-darkening. In the solar atmosphere the radiation at 6137 \AA originates below the temperature minimum. When we look at the regions of the solar disk closer to the limb we see this radiation coming from higher layers (see Fig. 2.3), but still from somewhere below the temperature minimum. Therefore, since the temperature at the formation height (see Sect. 8.3) of radiation is lower as we move towards the disk limb the emerging radiation undergoes the corresponding decrease in intensity.

The CLV effect has to be removed in order for the analysis of the intensity image to be facilitated, or in other words the intensity image has to be normalized with respect to the CLV background. To do this we first map the (x, y) dependence of the image intensity into its radial dependence:

$$I(x, y) \rightarrow I(r(x, y)), \quad (2.1)$$

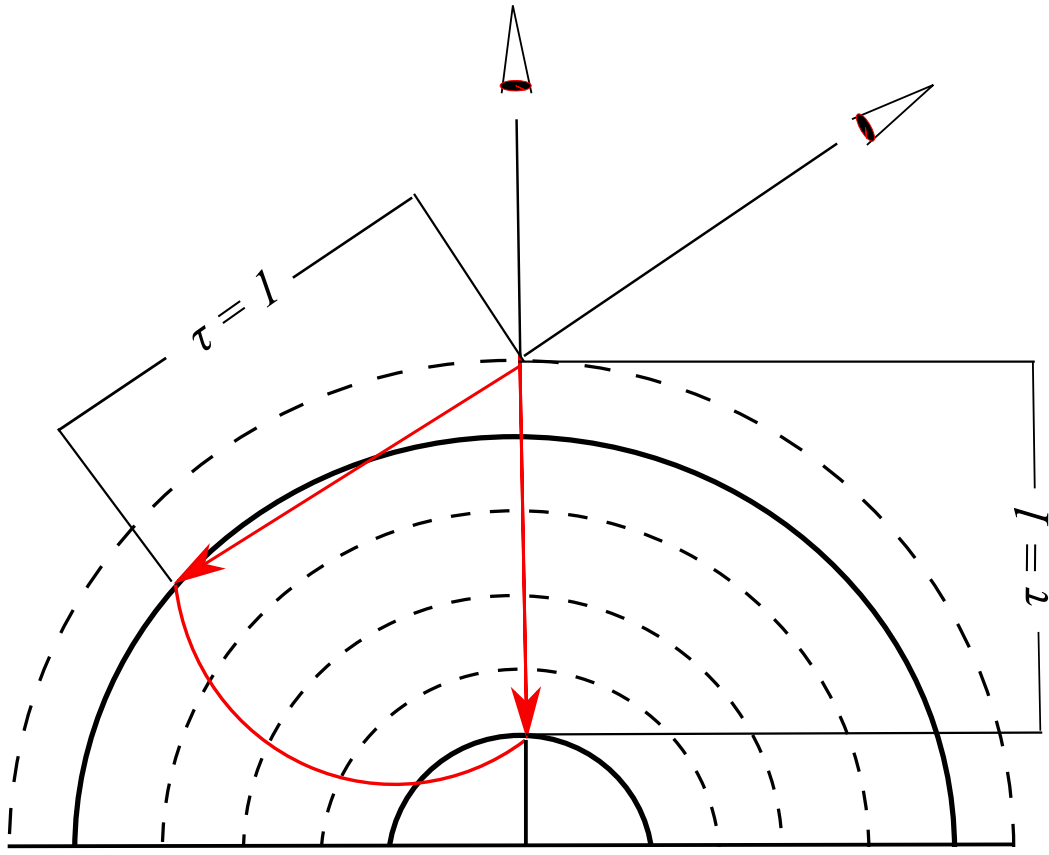


Figure 2.3: The schematic representation of the CLV-effect. An observer always sees radiation forming at the layer with $\tau = 1$ (τ is the optical depth, see Appendix D for details). When the line of sight differs from the direction perpendicular to the solar disk the $\tau = 1$ level is reached at a higher level.

where

$$r(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2}. \quad (2.2)$$

This mapping gives the cloud of points shown in the left panel of Fig. 2.2, for which we then find the best-fit 6th order polynomial:

```
>>> import numpy as np

>>> import numpy.polynomial.polynomial as poly

>>> c = poly.polyfit(r, Ir, 6)
```

where c is the array of polynomial coefficients. Now, we construct the CLV image (see the right panel of Fig. 2.2):

$$I_{\text{clv}}(x, y) = I_{\text{fit}}(r(x, y)) = \sum_{i=0}^6 c_i r^i(x, y). \quad (2.3)$$

The procedure building a CLV image for a given intensity image is given in Appendix A.

When we have built the CLV image we might want to save it in binary (.npz) format in order for it to be faster accessible at the following stages of the analysis (for example, plotting). The procedure for saving any array structure in python in binary format is as follows:

```
>>> import numpy as np

>>> np.savez(filename, array1 = array1, array2 = array2, array3 = array3)

>>> arrays = np.load(filename + '.npz') # pay heed to the "+ '.npz'" part
                                         # variable "arrays" works as a dictionary

>>> array1 = arrays['array1']
>>> array2 = arrays['array2']
>>> array3 = arrays['array3']
```

2.2 Image normalization and signal/noise threshold

Having calculated the CLV image, we obtain the normalized intensity image as:

$$I_n(x, y) = \frac{I(x, y)}{I_{\text{clv}}(x, y)}, \quad (2.4)$$

where $I(x, y)$ is the original image (see the left panel of Fig. 2.1) and $I_{\text{clv}}(x, y)$ is the CLV image (see the right panel of Fig. 2.2). The normalized image is shown in the left panel of Fig. 2.4.

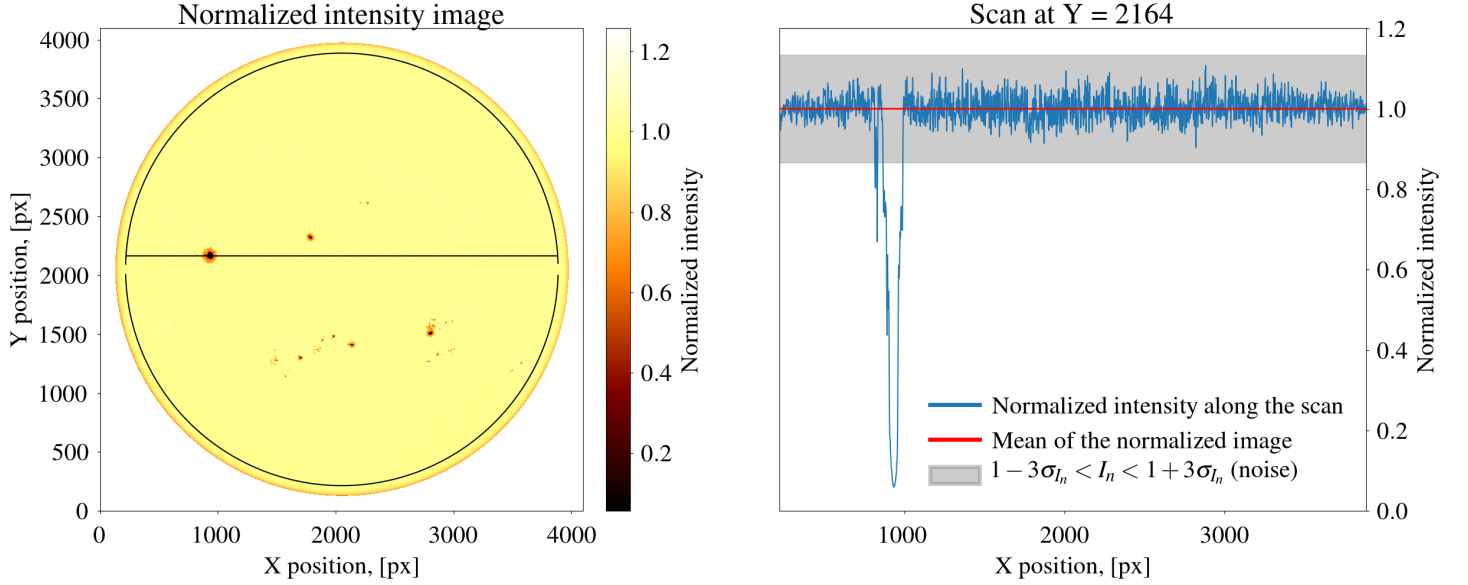


Figure 2.4: **Left panel:** Normalized intensity image (i.e. that in the left panel of Fig. 2.1 divided by the CLV image in the right panel of Fig. 2.2) **Right panel:** scan of the normalized intensity image within $r = 0.953R_{\odot}$ circumference at the Y-position of the biggest sun-spot.

Analogously to the right panel of Fig. 2.1 the right panel of Fig. 2.4 shows the scan of the normalized image at the Y-position of the biggest spot. The red line indicates the mean value of the normalized image. This mean value should be equal to one if the normalization is done properly. The way normalization has been done here suffers from the presence of the outliers in the scatter plot in the left panel of Fig. 2.2, therefore the mean of the normalized intensity we actually obtain is slightly lower than one (0.9937). A more consistent approach is required to remedy this. We will not get into the details of it here. However, this might be one of the possible further developments of the project.

Meanwhile, we will force the mean value of the normalized image to be equal to one and calculate the standard deviation of the normalized image as:

$$\sigma_{I_n} = \sqrt{\frac{1}{N_{\odot}} \sum_{x,y} (I_n(x, y) - 1)^2}. \quad (2.5)$$

Here, the summation is carried over all image pixels satisfying the condition:

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} \leq R_{\odot} \quad (2.6)$$

and N_{\odot} is the number of these pixels. We have to take into account that the values of the data in the original image and CLV image corresponding to the pixels not satisfying condition (2.6) are equal to NaN, therefore in the normalized image they are also equal to NaN and we have to exclude them with the appropriate logical operators:

```
>>> import numpy as np
```

```
>>> sigma = np.sqrt(np.mean((1 - In[np.where(np.logical_not(np.isnan(In)))]**2))
```

The $r < 0.953R_{\odot}$ restriction has been ignored here as it does not affect this stage of the analysis. At the further stages we will come back to this restriction again.

The gray area in the left panel of Fig. 2.4 designates the range

$$1 - 3\sigma_{I_n} < I_n(x, y) < 1 + 3\sigma_{I_n}. \quad (2.7)$$

All the intensity values in the original image satisfying condition (2.7) are considered to contain the noise. The pixels satisfying the condition

$$1 + 3\sigma_{I_n} < I_n(x, y) < 1 - 3\sigma_{I_n} \quad (2.8)$$

are considered to contain the signal. The $3\sigma_{I_n}$ value of the threshold ensures that 99.9% of the noise is excluded. For the image considered here $\sigma_{I_n} = 0.044623$.

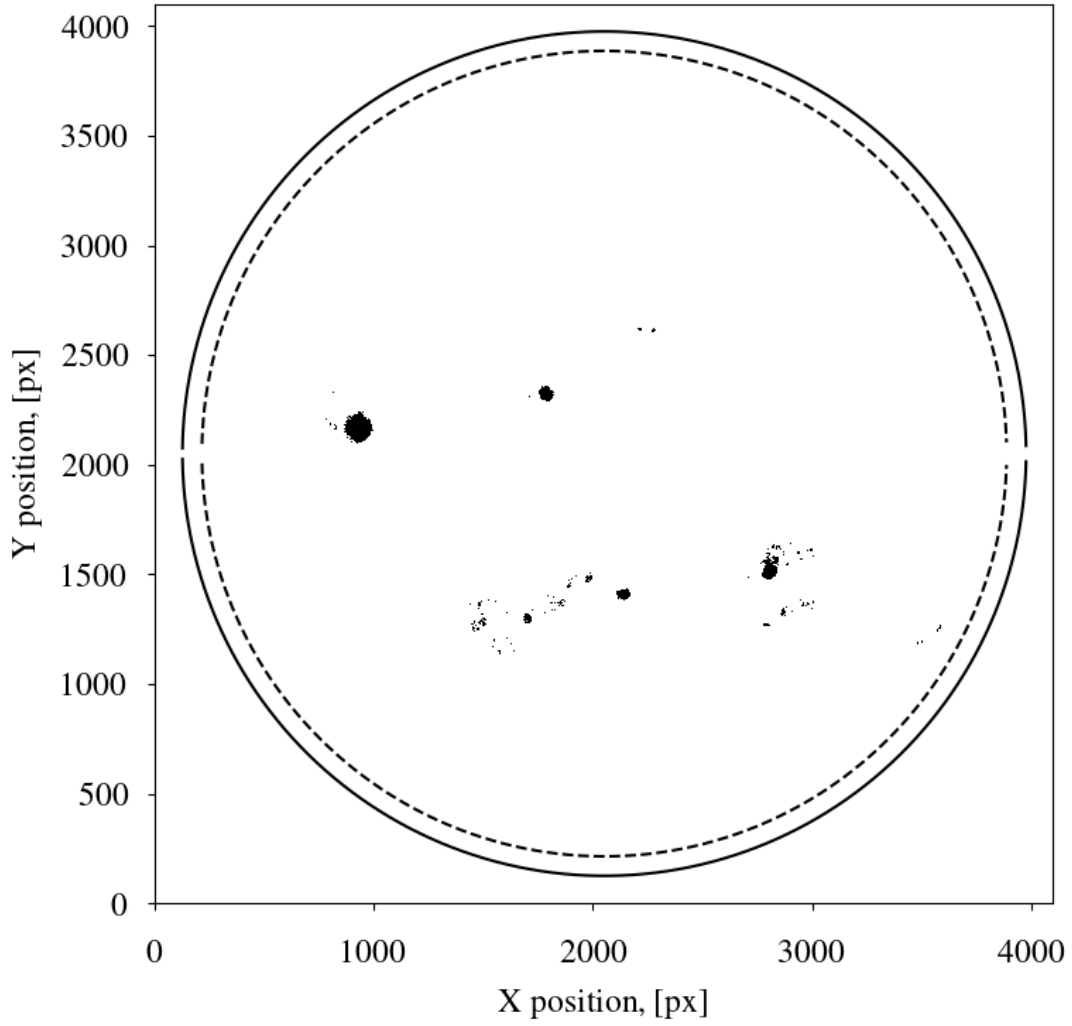


Figure 2.5: Intensity image mask given by Eq. (2.9). The pixels outside $r = 0.953R_{\odot}$ circumference (dashed line) have been disregarded.

2.3 Intensity image mask

Having set the signal criterion we can now build the intensity image mask:

$$M_I(x, y) = \begin{cases} 1, & \text{if } I_n(x, y) < 1 - 3\sigma_{I_n} \text{ (signal)} \\ 0, & \text{otherwise (noise).} \end{cases} \quad (2.9)$$

The mask is concerned only with the intensity signal below $1 - 3\sigma_{I_n}$ level because our ultimate purpose is the identification of sun-spots which as we know are dark.

Fig. 2.5 shows the intensity mask build from the image in the left panel of Fig. 1.1 according to Eq. (2.9). As we can see the size of the identified objects varies. Only the biggest ones are actually spots. The smaller ones are the so-called pores, which we will not study here.

As can be seen from the top panels of Fig. 2.6 the spots are composed of two parts. The outer brighter part is called penumbra and the inner darker one is called umbra. The drop of intensity in the spot is caused by the local suppression of convection by the spot's magnetic field. Since the main mechanism of energy transport below the solar photosphere¹ is convection this suppression entails the lower temperature in the spot with respect to its surroundings. In penumbral parts of the spot some forms of magneto-convection maintain their higher temperature with respect to umbra. The $1 - 3\sigma_{I_n}$ threshold set above corresponds to the outer border of penumbra in the intensity image. The transition from penumbra to umbra occurs at about $10\sigma_{I_n}$ level.

Having that in mind we now build the intensity image mask taking into account the internal division of spots into umbra

¹photosphere, in turn, is the region in the solar atmosphere below the local minimum of its temperature distribution, this is the region where the radiation at 6173 Å comes from

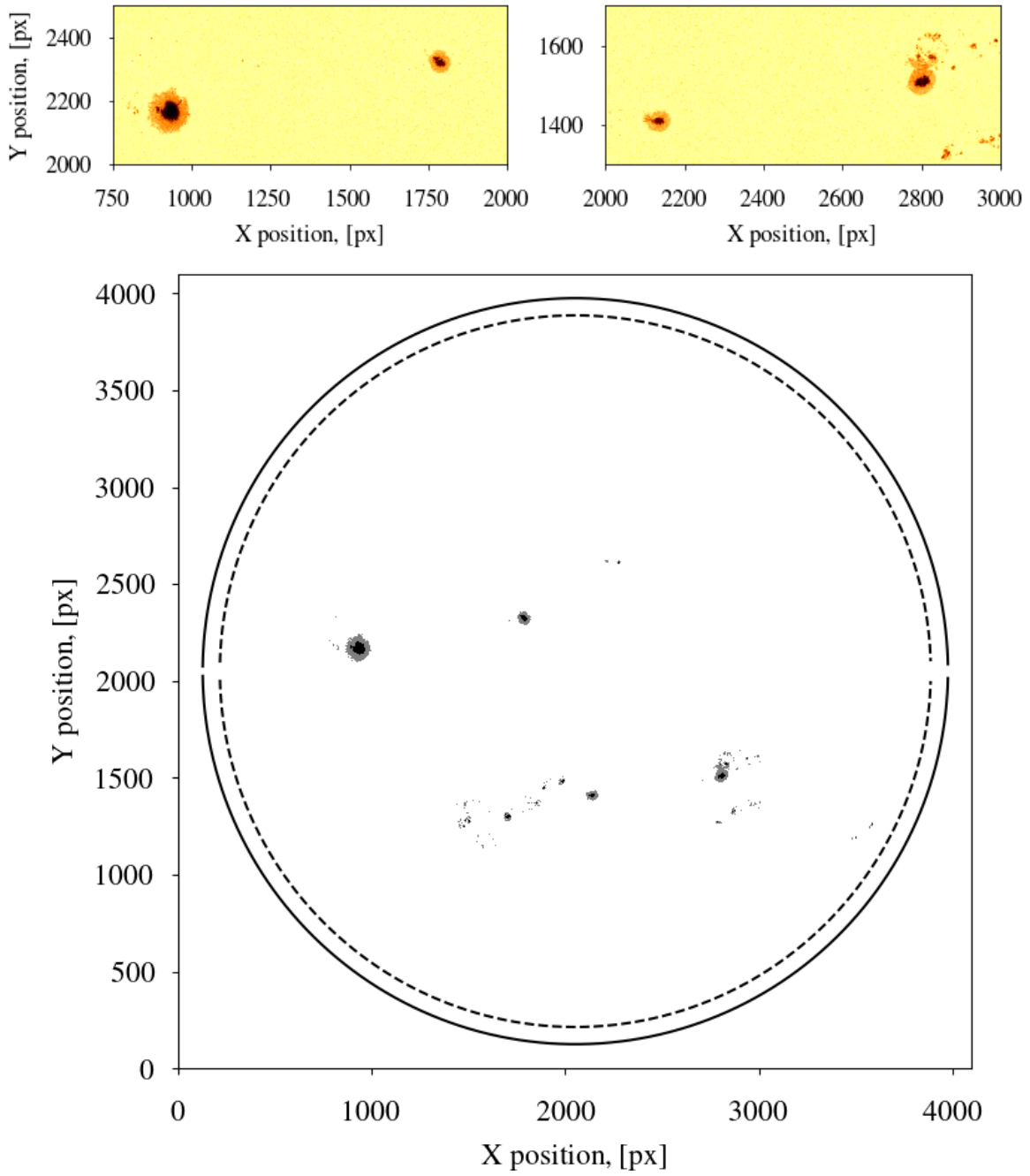


Figure 2.6: **Top panels:** parts of the normalized intensity image (see the left panel of Fig. 2.4) zoomed in to better show the internal division of the spots into umbra and penumbra. **Bottom panel:** intensity image mask given by Eq. (2.10). The pixels outside $r = 0.953R_{\odot}$ circumference (dashed line) have been disregarded.

and penumbra:

$$M_I^{\text{pu}}(x, y) = \begin{cases} 1, & \text{if } I_n(x, y) < 1 - 10\sigma_{I_n} \text{ (umbra signal)} \\ 0.5, & \text{if } 1 - 10\sigma_{I_n} < I_n(x, y) < 1 - 3\sigma_{I_n} \text{ (penumbra and other signal)} \\ 0, & \text{otherwise (noise).} \end{cases} \quad (2.10)$$

The result is shown in the bottom panel of Fig. 2.6. The procedure for building the intensity masks according to Eqs. (2.9) and (2.10) is given in Appendix B.

3 Magnetograms

3.1 Flux tubes

In terms of spots and faculae identification, finding pixels with magnetic signal is complementary to finding the dark pixels, since what is magnetic and dark enough is a spot-like object and what is magnetic and not dark is a facula. This, in turn, is because topologically spots and faculae are the same. Roughly, they can be represented as so-called magnetic flux tubes. Fig. 3.1 shows a sketch of the vertical cross-section of a magnetic flux tube. The difference between spot and facula is mainly in the radius of its horizontal cross-section.

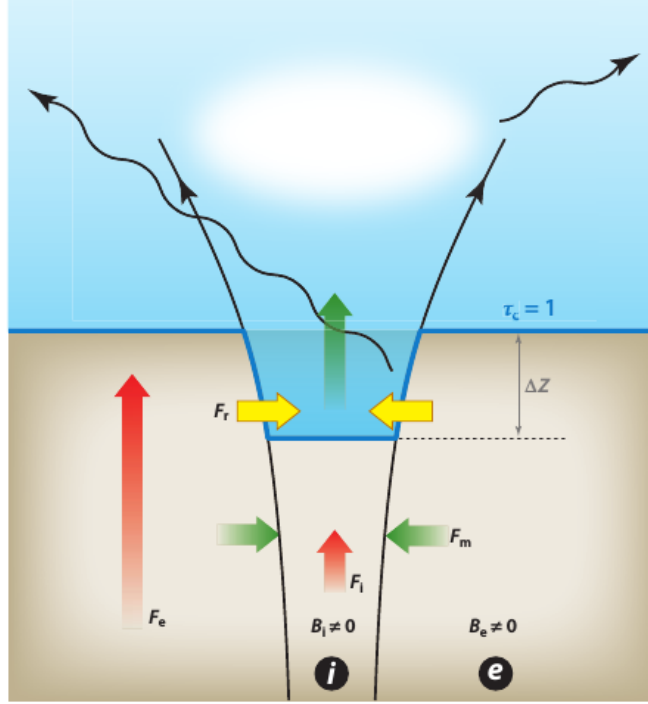


Figure 3.1: Adopted from [3]. Sketch of the vertical cross section through a magnetic flux tube. The red arrows show the vertical convective and radiative energy transport inside the flux tube (designated with i) and in the external medium (designated with e). The yellow arrows indicate the horizontal radiative energy flow into the flux tube. The rest of the designations are not important.

As has been mentioned above, the convection in a spot is suppressed by the magnetic field. It is also suppressed in a facular flux tube, but because the latter has a smaller radius the loss of heat resulting from the suppression of convection is compensated by the radiative heating coming from the hotter external medium through the walls of the flux tube. As the radius of the flux tube increases the amount of matter that has to be heated up grows as $\sim R^2$ and the heat flux into the tube grows as $\sim R$. Therefore, the radiative inflow through the walls is insufficient to re-heat the matter inside a large flux tube and it remains cold and dark and is seen as a spot.

3.2 Finding the signal pixels

We now proceed to identifying the pixels containing magnetic signal. Analogously to Fig. 2.1, the magnetogram scan is shown in Fig. 3.2, in which the values of magnetic field represent the line-of-sight (LOS) component of magnetic field B_μ . The LOS component is denoted with the μ subscript, which is the standard designation for the cosine of observational angle θ in solar physics. This angle is formed by the intersection of LOS corresponding to a certain position on the solar disk and the local normal to the solar surface (see Fig. 3.3). Its cosine is used in solar physics as a measure of radial distance from the solar disk center. Alternatively, the impact parameter p can be used (see Fig. 2.2, left panel). Assuming that the local magnetic field is predominantly perpendicular to the solar surface the relationship between LOS magnetic field B_μ and the actual magnetic field can be expressed in terms of μ (see Fig. 3.3). Below we will discuss when this assumption breaks down.

The principle of magnetic signal identification is the same as for the intensity images, except that no normalization has to be performed. However, in the case of magnetograms, proper calculation of σ_{B_μ} is quite complicated. We will not get into the details of it here and adopt the value of $\sigma_{B_\mu} = 10G$ for the noise level (the references to this number will be given a bit later). The mean magnetic signal is zero as is seen from the right panel of Fig. 3.2. Hence, the magnetic mask is given by:

$$M_B(x, y) = \begin{cases} 1, & \text{if } |B_\mu(x, y)| > 3\sigma_{B_\mu} = 30G \text{ (signal)} \\ 0, & \text{otherwise (noise).} \end{cases} \quad (3.1)$$

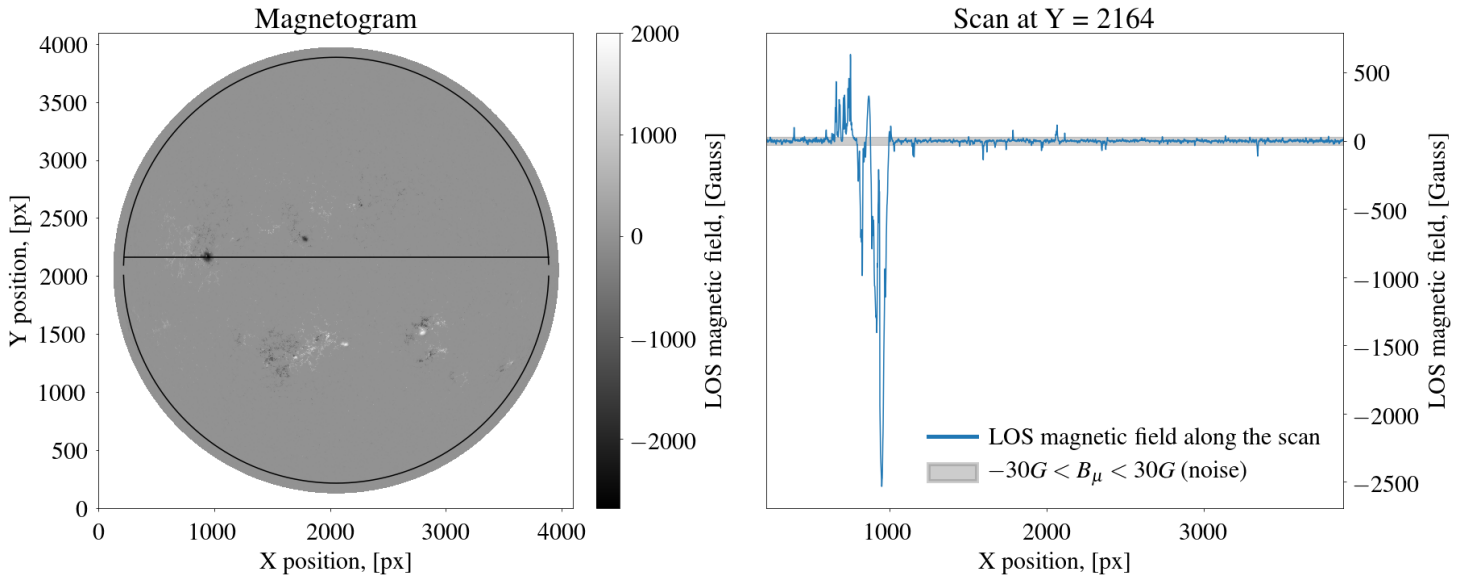


Figure 3.2: Magnetogram scan (see the right panel of Fig. 1.1) within $r = 0.953R_{\odot}$ circumference at the Y -position of the biggest sun-spot.

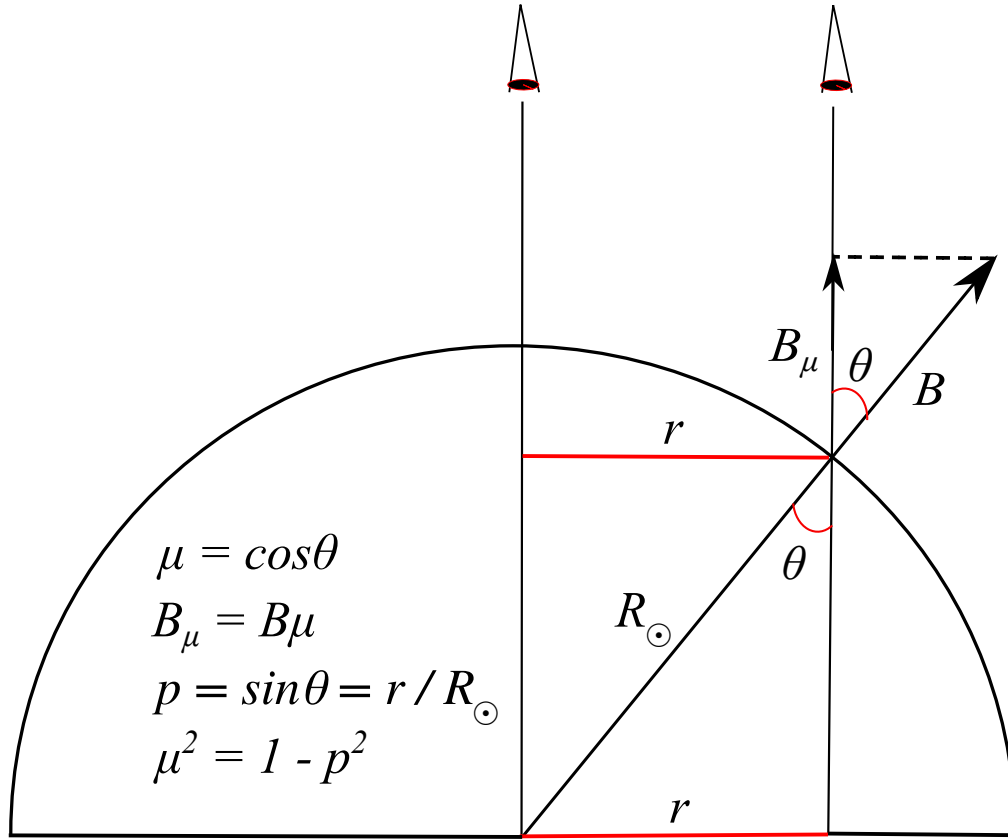


Figure 3.3: Geometry of the observational angle and impact parameter. Lines of sight corresponding to different positions on the solar disk can be considered parallel, because the observer is far away from the Sun. The constraint $r < 0.953R_{\odot}$ or equivalently $p < 0.953$ introduced in the beginning corresponds to $\mu > 0.3$.

A procedure building the magnetic mask is given in Appendix B. In accordance with the discussion in Sect. 3.1 the facular mask is calculated as:

$$M_F(x, y) = M_B(x, y) - M_I(x, y). \quad (3.2)$$

The magnetic and facular masks for the images in Fig. 1.1 are shown in Fig. 3.4 from which we can see the morphological difference between the spot and facular coverages of the solar disk. Faculae, being much smaller than spots (see Sect 3.1), can cover an area on the solar disk which is smaller than the area of one pixel at the disk center ². They group together and

²the area covered by one pixel depends on μ because of the projection effect

form the facular structures seen in the right panel of Fig. 3.4. In that sense (as opposed to the topological sense, discussed in Sect. 3.1) faculae are different from spots because the latter represent singular spatially resolved objects and the former — a conglomeration of (mostly) spatially unresolved objects.

The magnetic field lines in a flux tube diverge. The horizontal component of the magnetic field grows with the radial distance from the center of the horizontal cross-section of the flux tube. In the penumbral parts the field becomes more and more horizontal (see Fig. 3.5). That is why we see the -1 values in the facular mask. These values correspond to the outer parts of some spots (i.e. penumbral) where the magnetogram values are the actual signal, but because the LOS component of the field is so low they turned out to be below the 30G LOS magnetic field threshold (see Eq. 3.1) and were counted as noise. Therefore

$$\text{if } M_F(x, y) = -1 \Rightarrow M_F(x, y) \equiv 0. \quad (3.3)$$

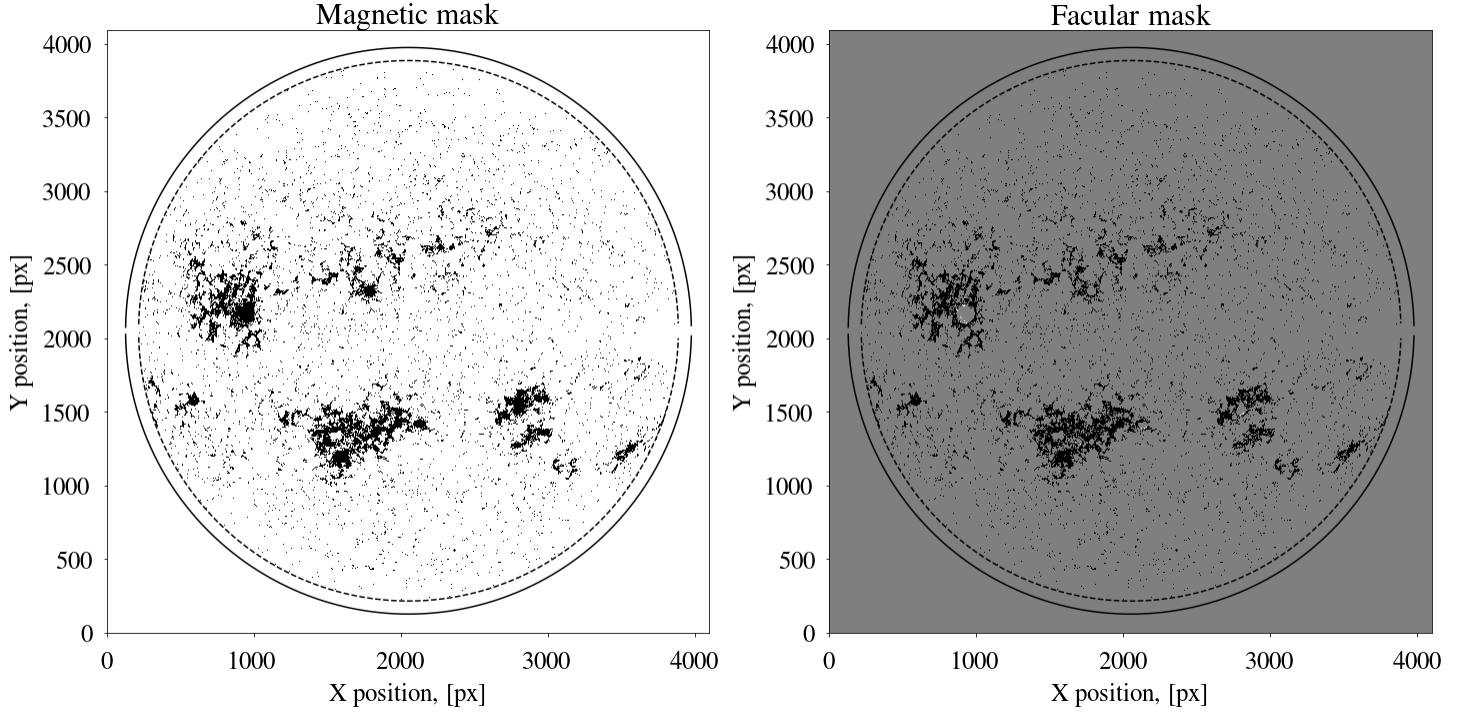


Figure 3.4: Magnetic and facular masks for the images in Fig. 1.1 built according to Eqs. (2.9), (3.1) and (3.2). Color correspondence in the left panel: white is 0, black is 1. Color correspondence in the right panel: white is -1 , gray is 0, black is 1. The pixels outside $r = 0.953R_{\odot}$ circumference have been disregarded.

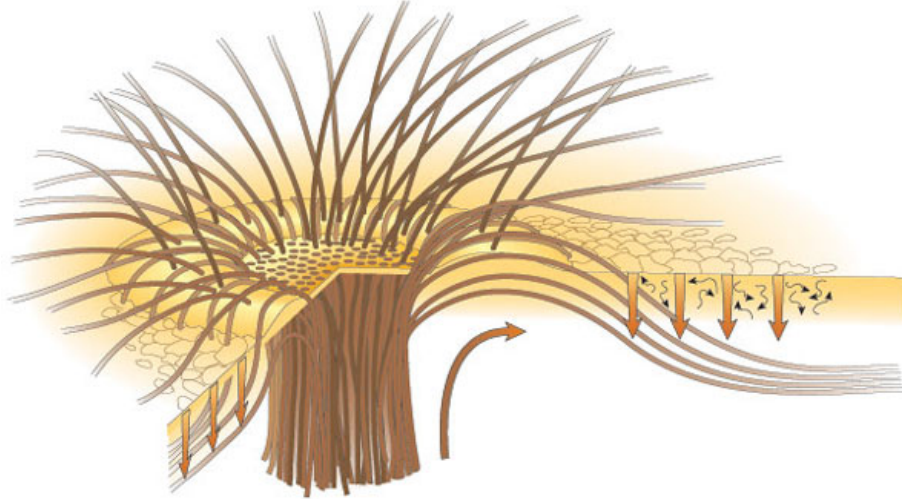


Figure 3.5: Adopted from [4]. Sketch of 3D sun-spot magnetic field structure showing almost horizontal magnetic field in the penumbral regions of a sun-spot.

4 Facular and umbral magnetic fields and spot sizes

Having the intensity and magnetic masks we can now look at the typical facular and spot magnetic fields and also calculate the typical sizes of sun-spots. Because of the predominantly horizontal alignment of the magnetic field in the penumbra (see Fig. 3.5), the determination of penumbral magnetic fields is obstructed. When penumbra is seen at the limb its magnetic field is aligned with LOS, otherwise it is predominantly out of alignment and there is no simple way to obtain the actual magnetic field from the penumbral LOS magnetic field component. Hence, we will only concern ourselves with facular and umbral magnetic fields which are predominantly perpendicular to the local solar surface (as was explained above) and therefore the actual magnetic field can be obtained from the LOS magnetic field by means of the observational angle correction (see Fig. 3.3). The determination of the sizes of faculae is, in turn, obstructed by their smallness and conglomeration.

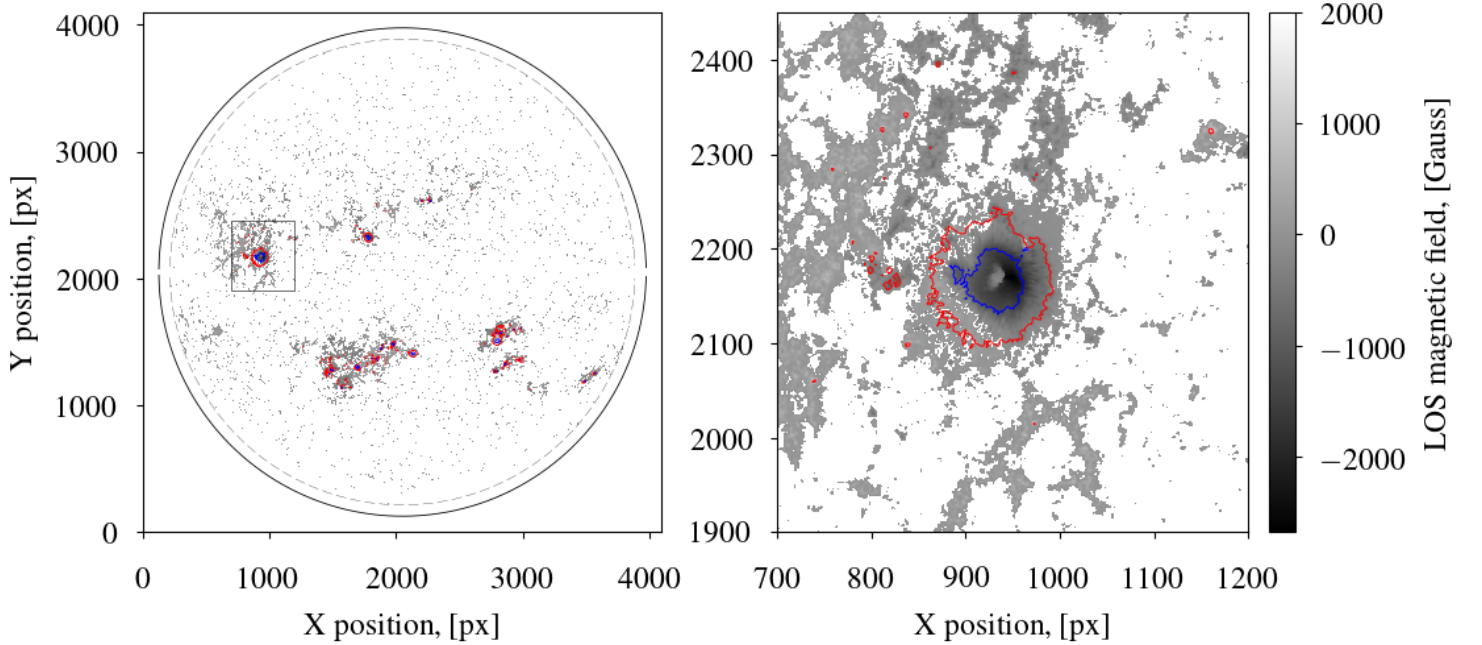


Figure 4.1: **Left panel:** Magnetic mask built according to Eq. (4.1). The umbral and penumbral contours are shown in blue and red respectively. The pixels outside $r = 0.953R_{\odot}$ circumference have been disregarded. **Right panel:** Zoom-in of the box in the left panel.

Fig. 4.1 shows the magnetic mask

$$M'_B(x, y) = \begin{cases} B_{\mu}(x, y), & \text{if } |B_{\mu}(x, y)| > 3\sigma_{B_{\mu}} = 30G \\ \text{NaN}, & \text{otherwise,} \end{cases} \quad (4.1)$$

with the values of LOS magnetic field from the original magnetogram along with the penumbral and umbral contours. The typical values of facular and umbral magnetic fields can be obtained by making a scatter plot of all magnetic field values outside the red contours and inside the blue contours, respectively, as a function of radius (or any other measure of radial distance) or X -position or Y -position or pixel number (though in the last case it would not be a scatter plot). The observational angle correction of magnetic field (see Fig. 3.3) has to be implemented before the plotting³.

The typical spot sizes can be obtained by labeling spots in Fig. 2.5:

```
>>> from scipy.ndimage import measurements
```

```
>>> labeled_spot_mask, number_of_spots = measurements.label(original_spot_mask)
```

and summing the areas of individual spot pixels, having implemented the observational angle correction to the said areas⁴.

Both tasks can be quite easily performed using the intensity, magnetogram and facular masks in combination with applying the `numpy.where` function. In some cases `numpy.logical_not` and `numpy.isnan` functions might be needed. The procedure for plotting the umbral and penumbral contours (see Fig. 4.1) is given in Appendix C.

³the observational angle has to be calculated for each pixel of the magnetogram

⁴the correction of pixel area is analogous to the magnetic field correction

5 Appendix A: procedure for building the CLV image

```
1 import numpy as np
2
3 import numpy.polynomial.polynomial as poly
4
5 from tqdm import tqdm # this is just a package for making nice progress bars
6
7 def clv(int_img, x0, y0, r, rsun): # r = 0.953 * rsun
8
9     rad = its = np.array([])      # rad and its are 1D-arrays here
10
11     y1 = int(np.ceil(y0 - r))
12     y2 = int(np.ceil(y0 + r))
13
14     for ys in tqdm(range(y1, y2), desc = 'Building I(r)'):
15
16         if ys % 2 != 0: continue
17
18         x = np.arange(4096)
19
20         x1 = x0 - np.sqrt(r**2 - (ys - y0)**2)
21         x2 = x0 + np.sqrt(r**2 - (ys - y0)**2)
22
23         idx_scan = np.where((x >= x1) & (x <= x2) & (x % 2 == 0))
24
25         xs = x[idx_scan]
26
27         rs = np.sqrt((xs - x0)**2 + (ys - y0)**2)
28
29         ints = int_img[ys, idx_scan[0]]
30
31         its = np.concatenate((its, ints))
32
33         rad = np.concatenate((rad, rs))
34
35     c = poly.polyfit(rad, its, 6)
36
37     ld_img = np.zeros((4096, 4096))
38
39     y1 = int(np.ceil(y0 - rsun))
40     y2 = int(np.floor(y0 + rsun))
41
42     for ys in tqdm(range(y1, y2), desc = 'Building CLV image'):
43
44         x = np.arange(4096)
45
46         x1 = x0 - np.sqrt(rsun**2 - (ys - y0)**2)
47         x2 = x0 + np.sqrt(rsun**2 - (ys - y0)**2)
48
49         idx_scan = np.where((x >= x1) & (x <= x2))
50
51         xs = x[idx_scan]
52
53         rs = np.sqrt((xs - x0)**2 + (ys - y0)**2)
54
55         ld_img[ys, idx_scan[0]] = poly.polyval(rs, c)
56
57     return ld_img
```

6 Appendix B: procedure for building the spot/umbra/penumbra/magnetic masks

```
1 import numpy as np
2
3 from tqdm import tqdm # this is just a package for making nice progress bars
4
5 def rad(x0, y0):
6
7     rad = np.zeros((4096, 4096))
8
9     for x in tqdm(range(4096), desc = 'Calculating r(x, y)'):
10
11         for y in range(4096):
12
13             rad[y, x] = (x - x0)**2 + (y - y0)**2
14
15     rad = np.sqrt(rad)
16
17     return rad
18
19 def intensity_mask(I, r, rad): # r = 0.953 * rsun # rad is calculated as a 2D-array above
20
21     sigma = np.sqrt(np.mean((1 - I[np.where(np.logical_not(np.isnan(I)))]**2)))
22
23     I[np.where(np.isnan(I))] = 0.0
24
25     Ip = 1 - 3 * sigma
26
27     Iu = 1 - 10 * sigma
28
29     mask_s = np.zeros((4096, 4096))
30     mask_u = np.zeros((4096, 4096))
31     mask_p = np.zeros((4096, 4096))
32
33     mask_s[np.where((rad <= r) & (I <= Ip) & (I != 0.0))] = 1.0
34
35     mask_u[np.where((rad <= r) & (I <= Iu) & (I != 0.0))] = 1.0
36
37     mask_p[np.where((rad <= r) & (I > Iu) & (I <= Ip) & (I != 0.0))] = 0.5
38
39     return mask_s, mask_u, mask_p
40
41 def magnetic_mask(B, r, rad): # r = 0.953 * rsun # rad is calculated as a 2D-array above
42
43     B[np.where(np.isnan(B))] = 0.0
44
45     sigma = 10.0
46
47     mask = np.zeros((4096, 4096))
48
49     mask[np.where((rad <= r) & ((B <= -3.0 * sigma) | (B >= 3.0 * sigma)) & (B != 0.0))] = 1.0
50
51     return mask
```

7 Appendix C: procedure for deriving and plotting umbral and penumbral contours

```
1 import numpy as np
2
3 import cv2
4
5 def get_cnt_coord(cnt):
6
7     x = np.array([])
8     y = np.array([])
9
10    for elem in cnt:
11
12        x = np.concatenate((x, [elem[0][0]]))
13        y = np.concatenate((y, [elem[0][1]]))
14
15    first_elem = cnt[0]
16
17    x = np.concatenate((x, [first_elem[0][0]]))
18    y = np.concatenate((y, [first_elem[0][1]]))
19
20    return x, y
21
22 shapes_p = cv2.inRange(mask_p, 0.4, 0.6)
23 shapes_u = cv2.inRange(mask_u, 0.9, 1.1)
24
25 # w1 and w2 stand for "whatever1" and "whatever2", we do not need these outputs
26 # get the info on the second output to understand how the function get_cnt_coord works
27
28 w1, cnts_p, w2 = cv2.findContours(shapes_p.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
29 w1, cnts_u, w2 = cv2.findContours(shapes_u.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
30
31 # ... some stuff here ...
32
33 for cnt in cnts_p:
34
35     x_cnt, y_cnt = get_cnt_coord(cnt)
36
37     plt.plot(x_cnt, y_cnt, '-', color = 'r')
38
39 for cnt in cnts_u:
40
41     x_cnt, y_cnt = get_cnt_coord(cnt)
42
43     plt.plot(x_cnt, y_cnt, '-', color = 'b')
44
45 # ... some more stuff here ...
```

8 Appendix D: Radiative transfer equation

8.1 Derivation of the general form

The main quantity of interest in the theory of radiative transfer is the intensity of radiation. The intensity is defined as the amount of energy carried by the photons within a frequency interval $d\nu$, through area $d\sigma$, over time dt , within solid angle $d\omega$ around the direction \mathbf{n} :

$$dE_\nu(\mathbf{r}, \mathbf{n}, t) = I_\nu(\mathbf{r}, \mathbf{n}, t) d\sigma dt d\nu d\omega. \quad (8.1)$$

Here, \mathbf{r} is the radius-vector of the point in space where the intensity is measured.

The transport of radiation through a medium is mathematically formulated with the equation of radiative transfer which describes the evolution of intensity along direction \mathbf{n} . The physical meaning of this equation is conservation of energy, i.e. that the intensity along the ray \mathbf{n} does not change unless photons are added into the beam or taken out of it. To write the conservation of energy we need the quantities characterizing the loss and gain of photons in the beam.

The gain of energy of frequency ν is quantified with the emissivity $\eta_\nu(\mathbf{r}, \mathbf{n}, t)$ which is the amount of energy dE_ν added by the volume dV into the beam of photons with frequencies within $d\nu$ and directions within $d\omega$ around vector \mathbf{n} per time dt :

$$dE_\nu^+(\mathbf{r}, \mathbf{n}, t) = \eta_\nu(\mathbf{r}, \mathbf{n}, t) dV dt d\nu d\omega. \quad (8.2)$$

The loss of energy is characterized by the absorption coefficient (or opacity) χ_ν defined so that the fraction of intensity subtracted from the beam within the length interval ds along the ray \mathbf{n} is $\chi_\nu ds$. Therefore the corresponding decrease of the beam energy is

$$dE_\nu^-(\mathbf{r}, \mathbf{n}, t) = \chi_\nu I_\nu(\mathbf{r}, \mathbf{n}, t) ds d\sigma dt d\nu d\omega. \quad (8.3)$$

We can now write the equation of radiative transfer using simple energy conservation considerations [1, 2]. The amount by which the energy of the beam at frequency ν changes as it passes the length interval ds is:

$$dE_\nu^+ - dE_\nu^- = (I_\nu(\mathbf{r} + d\mathbf{r}, \mathbf{n}, t + dt) - I_\nu(\mathbf{r}, \mathbf{n}, t)) d\sigma dt d\nu d\omega \quad (8.4a)$$

$$= (\eta_\nu(\mathbf{r}, \mathbf{n}, t) - \chi_\nu I_\nu(\mathbf{r}, \mathbf{n}, t)) ds d\sigma dt d\nu d\omega, \quad (8.4b)$$

where $ds = |d\mathbf{r}|$ and we took into account that $dV = ds d\sigma$. In Eq. (8.4a) we used the definition of intensity (Eq. 8.1) and when transitioning from Eq. (8.4a) to Eq. (8.4b) we used the definitions (8.2) and (8.3). Taking into account that $dt = ds/c$, where c is the speed of light, we write the difference of intensities as:

$$I_\nu(\mathbf{r} + d\mathbf{r}, \mathbf{n}, t + dt) - I_\nu(\mathbf{r}, \mathbf{n}, t) = \left(\frac{1}{c} \frac{\partial I_\nu}{\partial t} + \frac{\partial I_\nu}{\partial s} \right) ds = \left(\frac{1}{c} \frac{\partial I_\nu}{\partial t} + \mathbf{n} \nabla I_\nu \right) ds. \quad (8.5)$$

Here, we wrote the directional derivative $\partial/\partial s$ along vector \mathbf{n} in terms of the ∇ operator. Substituting the derivative (8.5) into Eq. (8.4) we arrive at the general form of the equation of radiative transfer:

$$\left(\frac{1}{c} \frac{\partial}{\partial t} + \mathbf{n} \nabla \right) I_\nu(\mathbf{r}, \mathbf{n}, t) = \eta_\nu(\mathbf{r}, \mathbf{n}, t) - \chi_\nu(\mathbf{r}, \mathbf{n}, t) I_\nu(\mathbf{r}, \mathbf{n}, t). \quad (8.6)$$

8.2 Plain-parallel geometry

Let us consider now the simplest geometrical case of a static non-expanding medium that is comprised of plane-parallel slabs. In each slab the properties of the medium remain the same both in space and time. We introduce the Cartesian system of coordinates with axis z perpendicular to the slabs, along which the properties of the medium change (see Fig. 8.1), and the observational angle θ with respect to axis z . In the Cartesian system of coordinates the directional derivative $\mathbf{n} \nabla$ is:

$$\mathbf{n} \nabla = n_x \frac{\partial}{\partial x} + n_y \frac{\partial}{\partial y} + n_z \frac{\partial}{\partial z}. \quad (8.7)$$

In the case of plain-parallel geometry $\partial/\partial x = 0$, $\partial/\partial y = 0$. Therefore, taking into account that the medium is static ($\partial/\partial t = 0$), the radiative transfer equation takes the form:

$$\mu \frac{dI_\nu(z, \mu)}{dz} = \eta_\nu(z) - \chi_\nu(z) I_\nu(z, \mu), \quad (8.8)$$

where $\mu = \cos \theta = n_z$. Dividing both sides of the last equation by $-\chi_\nu$ and designating $S_\nu = \eta_\nu/\chi_\nu$ we write:

$$\mu \frac{dI_\nu(\tau_\nu, \mu)}{d\tau_\nu} = I_\nu(\tau_\nu, \mu) - S_\nu(\tau_\nu), \quad (8.9)$$

where

$$d\tau_\nu = -\chi_\nu(z) dz \quad (8.10)$$

is increment of the so-called optical depth or length, which is the “effective distance” for photons. The physical meaning of optical depth is that in the process of radiative transfer across a certain distance, not the distance itself, but the strength of absorption of the radiation by the medium determines how “hard” it is for photons to travel across this distance. It is convenient to measure the optical depth in the direction opposite to that of z axis in Fig. 8.1 (hence the minus sign in Eq. 8.10). The quantity $S_\nu(\tau_\nu)$ is called the source function. It describes the influence of the medium on the intensity of the beam.

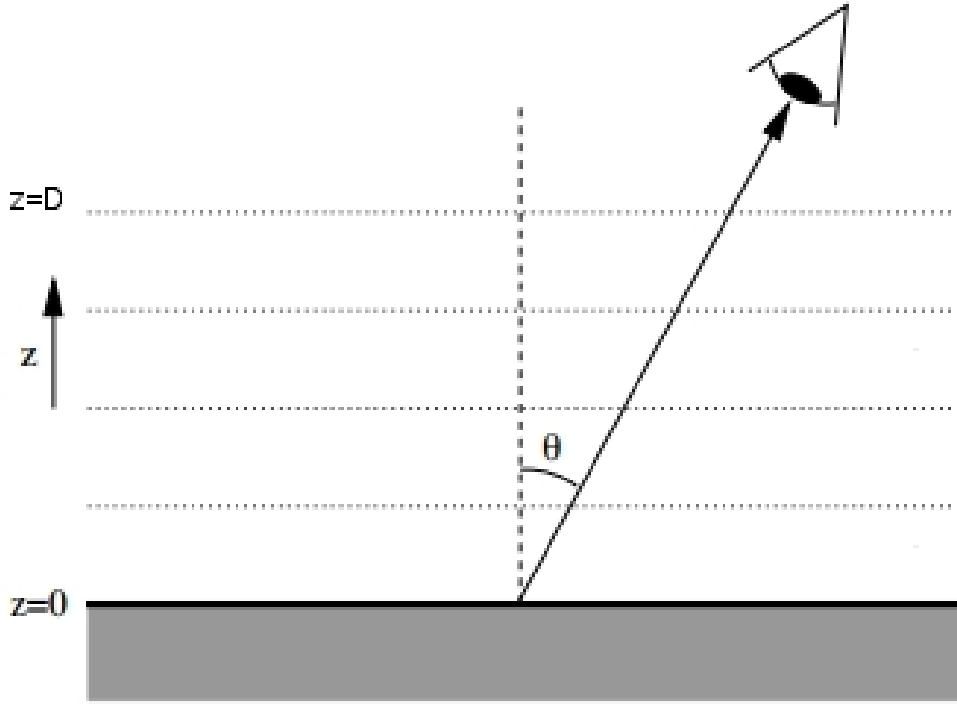


Figure 8.1: Plane-parallel geometry: the simplest case for which the radiative transfer equation can be written. The z axis is perpendicular to the parallel slabs of the medium in which its properties remain the same. The observer sees radiation coming at angle θ with respect to axis z .

8.3 Eddington-Barbier approximation

Eq. (8.9) can be solved analytically for inward directed intensity ($\mu < 0$) and outward directed intensity ($\mu > 0$) [2, p. 17]:

$$I_\nu(\tau_\nu, \mu) = \begin{cases} - \int_{\tau_\nu}^{\tau_\nu} S_\nu(t_\nu) e^{-(t_\nu - \tau_\nu)/\mu} dt_\nu / \mu, & \mu < 0, \\ + \int_{\tau_\nu}^{\infty} S_\nu(t_\nu) e^{-(t_\nu - \tau_\nu)/\mu} dt_\nu / \mu, & \mu > 0. \end{cases} \quad (8.11)$$

Solution given by Eq. (8.11) is called formal solution, since it expresses the intensity $I_\nu(\tau_\nu, \mu)$ in terms of the source function $S_\nu(\tau_\nu)$ which is not always known in advance.

According to the formal solution the emergent ($\tau_\nu = 0$) outward intensity as seen by the observer in Fig. 8.1 is:

$$I_\nu^+(\tau_\nu = 0, \mu) = \int_0^\infty S_\nu(t_\nu) e^{-t_\nu/\mu} dt_\nu / \mu. \quad (8.12)$$

Substituting the Taylor expansion of the source function

$$S_\nu(\tau_\nu) = \sum_{n=0}^{\infty} a_n \tau_\nu^n \quad (8.13)$$

into Eq. (8.12) and using the integral representation of the factorial

$$n! = \int_0^\infty x^n e^{-x} dx, \quad (8.14)$$

we obtain the Taylor expansion of the emergent intensity in terms of the cosine of the observational angle μ :

$$I_\nu^+(\tau_\nu = 0, \mu) = \sum_{n=0}^{\infty} n! a_n \mu^n. \quad (8.15)$$

Comparing Eq. (8.13) and Eq. (8.15) we arrive at the Eddington-Barbier approximation:

$$I_\nu^+(\tau_\nu = 0, \mu) \approx S_\nu(\tau_\nu = \mu). \quad (8.16)$$

Hence, the radiation coming in the direction μ forms at the layer with $\tau_\nu \approx \mu$ ($\tau \approx 1$ for the radiation coming along the direction perpendicular to the slabs of the medium). Thus, the notion of the formation height arises. The formation height of the radiation

observed along the direction μ is defined as the height of the layer in the medium with $\tau_\nu = \mu$. Using the definition of optical depth (Eq. 8.10) we can write the condition from which the formation height h_f , as seen from the direction μ , can be found:

$$\mu = - \int_D^{h_f} \chi_\nu dz, \quad (8.17)$$

where D is the height of the topmost layer of the medium (see Fig. 8.1).

References

- [1] D. Mihalas. *Stellar atmospheres, 2nd edition*. 1978.
- [2] R. J. Rutten. *Radiative Transfer in Stellar Atmospheres*. May 2003.
- [3] S. K. Solanki, N. A. Krivova, and J. D. Haigh. “Solar Irradiance Variability and Climate”. In: *Annual Review of Astronomy and Astrophysics* 51 (Aug. 2013), pp. 311–351. DOI: 10.1146/annurev-astro-082812-141007. arXiv: 1306.2770 [astro-ph.SR].
- [4] J. H. Thomas et al. “Downward pumping of magnetic flux as the cause of filamentary structures in sunspot penumbrae”. In: *Nature* 420 (Nov. 2002), pp. 390–393. DOI: 10.1038/nature01174.