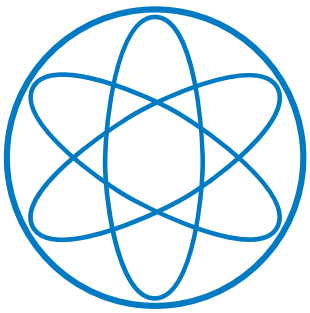
Physik

Technische Universität München

Bachelor's Thesis

Information Field Theory with INTEGRAL/SPI data

Andreas Wagner



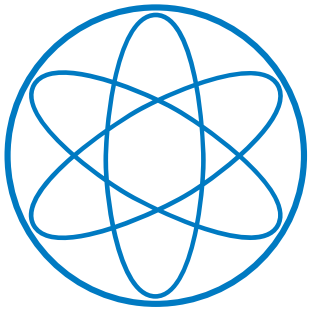
Physik

Technische Universität München

Bachelor's Thesis

Information Field Theory with INTEGRAL/SPI data

Author:	Andreas Wagner
Supervisor:	Prof. Dr. rer. nat. (apl.) Roland Diehl
Submission Date:	August 22th, 2017



Contents

Chapter 1

Introduction

Since its very first days, mankind has observed the sky and tried to learn something about the world around us by interpreting these observations. At first, only the light visible to the human eye was observed with telescopes, which improved the visual perception. In more recent times other wavelengths have become accessible, and help to get a deeper understanding of astrophysical phenomena: microwaves in the lower energy region and gamma-rays for high energy processes. These energy regimes require more complex telescopes to acquire experimental data. This information in its raw form is unsuitable for humans and has to be transformed into a more direct visualization. The most natural format are celestial skymaps, which are projections of the sky onto a flat surface, similar to maps of the Earth surface. The construction of these maps is – especially for gamma-rays – far from trivial and poses a hard challenge in the field of data analysis.

The goal of this bachelor's thesis is to investigate an information-based method to construct these sky maps with a specific algorithm, using a Bayesian inference framework. Here, we will try to create a map of the 511 keV line, with the experimental data gathered by the INTEGRAL/SPI gamma-ray telescope.

Previous work. The results of this thesis are a continuation of the work in Mahsa Ghaempanah's doctoral thesis [Gha17]. There she explores, how the data of INTEGRAL/SPI can be processed by algorithms derived through **i**nformation **f**ield **t**heory (IFT) [EFK09]. She further examines how the generalized Wiener filter and a modified version of the D^{3}PO-algorithm [SE15] perform with data from SPI, which was developed for "**D**enoising, **D**econvolving, and **D**ecomposing **P**hoton **O**bservations".

This thesis extends her work and investigates how a modified D^{3}PO-algorithm behaves, if the background, which dominates the SPI gamma-ray data, is handled differently. For this purpose, the code was modified to increase the performance and to allow for easy modifications. We adapted the original D^{3}PO-codebase for our needs and relied on it as a reference.

Outline. We will start in Chapter 2 with a short introduction to the SPI telescope. Since we want to focus more on data analysis, we will present the fundamentals to understand how a gamma-ray telescope works.

Everything we measure which is not part of our sky signal is called background. This background poses the main difficulty in the image analysis for gamma-rays, since it has the same order of magnitude as the data, whereas the sky signal is very weak. To handle the background-problem properly, we require a mathematical description of our measurement process. This will be the topic in Chapter 3. Since we will later use IFT for our inference process, we will use its language and formalism for our problem formulation. Hence we will give a general introduction to IFT in Sec. 3.1 and discuss how IFT applies to the INTEGRAL/SPI problem in Sec. 3.2. In Sec. 3.3, we will describe our telescope in terms of a linear operator. We would apply the same procedure to the background, but while the signal operator is well understood, the background response operator has a different appearance and must be treated differently. Hence we will try to understand the background in terms of reappearing patterns in Sec. 3.4, and formulate

a very general operator in Sec. 3.5, which lacks a discretization with respect to time. We can create such a discretization by partitioning our observations, which will be the topic of Chapter 4.

A simple inversion of these operators to get estimates of the sky and background signal is not possible. Not only because the experimentally measured data includes poissonian noise, but also because the problem is ill-conditioned. Consequently, we illustrate the image reconstruction through an inference algorithm in Chapter 5. At first, we will formulate mathematical basis in Sec. 5.1, in order to rederive a simplified version of the D^3PO-algorithm (Sec. 5.2), and point out its modifications (Sec. 5.3).

In Chapter 6, we will apply it to data of the SPI telescope. We will start with carefully analyzing how it handles simulated data in Sec. 6.1. After that, we will apply it on "real" measured data in Sec. 6.2.

Finally in Chapter 7 we will summarize our results and give a small outlook into possible future modifications and improvements of the algorithm.

A short summary of our main optimizations in the implementation of the algorithm is given in Appendix A.

Chapter 2

The INTEGRAL/SPI telescope

The goal of this chapter is to give a short overview of the INTEGRAL/SPI spectrometer. INTEGRAL stands for **Inte**rnational **G**amma-**R**ay **A**strophysics **L**aboratory and is a ESA satellite orbiting Earth, whose payload consists of several instruments, specialized on analyzing gamma-rays. This thesis focusses on SPI, the **SP**ectrometer on **I**ntegral, which is a coded-mask gamma-ray telescope. We will concentrate on the aspects of the telescope, which are important for our data analysis, and refer to [KBS06] for a more general introduction to gamma-ray telescopes.

A high-energy telescope measures the energy of the incoming photons and the direction from which they originate. In astrophysics it is useful to have very fine spectrometers for detailed analysis of specific energy ranges. Hence the detectors have to be comparatively large and since the size of the instrument is limited, only a few detectors are built in, which results in a rather coarse image resolution. Especially for gamma-ray telescopes, the spectral lines we want to detect are typically very narrow. SPI has an energy resolution of $\approx 2.5\,\mathrm{keV}$ at $1.3\,\mathrm{MeV}$ and an angular resolution $2.5°$ within a field of view of $16°$ [Ved+03].

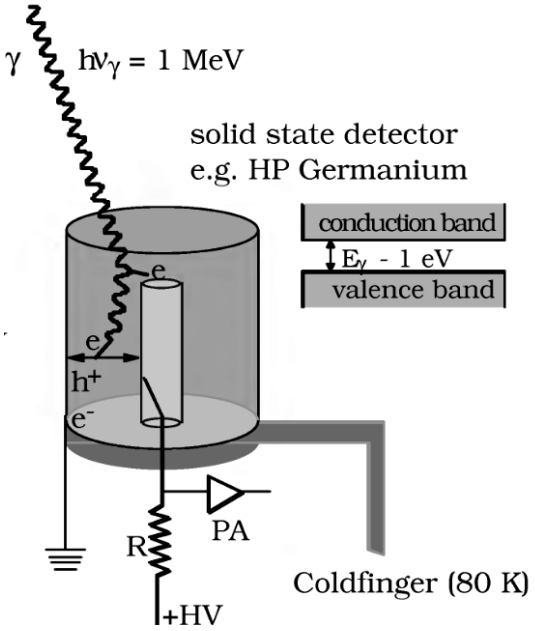


Figure 2.1: How a Ge-Detector works. Picture taken from [KBS06].

2.1 Detecting gamma-rays

A gamma-ray telescope needs a way to detect photons, as e.g. a typical CCD would not work, since most of them would only penetrate the sensor and continue their flight undisturbed. To avoid this SPI uses 19 high purity germanium semiconductor detectors, with a size of $\approx 6 \times 6 \times 6\mathrm{cm}^3$ per detector. The large detector size increases the pathlength of the photons through the detector, which improves its efficiency.

How does a semiconductor detector work? If a gamma-ray hits a semiconductor, it lifts one or several electrons from the valence band into the conduction band and generates *secondary electrons*. Along the tracks of these secondary electrons, further electron hole pairs are created, and their number is proportional to the energy of the secondary electrons. Therefore, by counting the electrons and holes, we can make conclusions about the energy of

Composition	Density [g cm^{-3}]	Band gap [eV]
Ge	5.32	0.74
Si	2.33	1.12
CdTe	6.2	1.6
Cd(Zn)Te	6.0	1.6
HgI_2	6.36	2.15

Table 2.1: Band gap and density of common semiconductor materials. Table taken from [KBS06, p.144].

the secondary electron, and thus the incident photon.

These pairs are now separated through a high voltage field (see Fig. 2.1) before they recombine. The electrons then drift to the anode and the holes to the cathode and produce an electric pulse, which we can measure and whose integral over time is proportional to the number of electrons.

Why is germanium used? On the one hand, a small band gap increases the efficiency of the detector. On the other hand, a high density of the material increases the probability that photons interact with it. Germanium has a band gap of 0.74 eV and a density of 5.32 g cm^3. If we compare this to other semiconductors (see Table 2.1), we see that Germanium is in both respects a good compromise.

Using high purity germanium, sounds counterintuitive since we know from doping that impurities can decrease the band gap. But these additional levels, "in the middle of the gap" trap the electrons and holes long enough, that they cannot contribute to the signal. Therefore the sensitivity is reduced.

Cooling The small band gap however, also brings problems: Thermal excitations become more numerous and create a *leakage current*, which interferes with any signal. We therefore have to cool down the detector to $\sim$ 90 K to increase the gap again.

2.2 Placing a gamma-ray telescope

The Earth atmosphere is opaque for gamma-rays (Fig. 2.2), thus the SPI-telescope is mounted on the INTEGRAL satellite, orbiting the Earth.

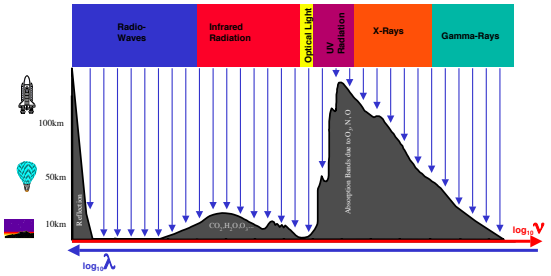


Figure 2.2: Atmospheric transparency for photons. Picture taken from [Die01, p. 3].

The orbit is chosen such that the exposure is maximized by avoiding the van Allen radiation belts, consisting of particles trapped by the magnetic field of the Earth. No observations are taken in these intervals.

2.3 Imaging of gamma-rays

The directions from which gamma-rays originate are most difficult to determine. There exist several possibilities:

1. coded aperture systems,
2. Compton telescopes and
3. focusing instruments.

Since SPI uses a coded aperture we will concentrate on that.

The idea behind a coding aperture is not to look at the sky directly but at the shadow it casts. To achieve this, a mask is put on top of the telescope, which in the cse of SPI consists of hexagonal tungsten cells, either filled or empty. Light rays will pass through the empty cells more easily and therefore cast a shadow pattern on the Germanium detector array, which depends directly on the direction of the light. This process is illustrated in Fig. 2.3 for two light sources. Both cast a distinct shadow pattern, and, if the patterns are independent enough, we are able to separate their superposition.

For detecting the shadow pattern on the camera, we use the technique of Sec. 2.1. SPI has 19 high purity germanium detectors on the detection plane. A photo of this detector array is shown in Fig. 2.4, a photo of the mask in Figure 2.5.

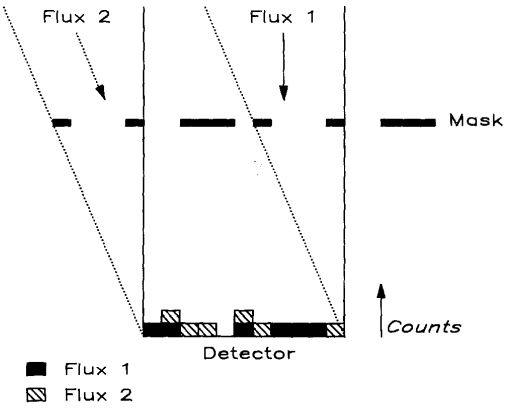


Figure 2.3: Basic functional principle of a coded mask telescope. Picture taken from [Car+87, p. 3].

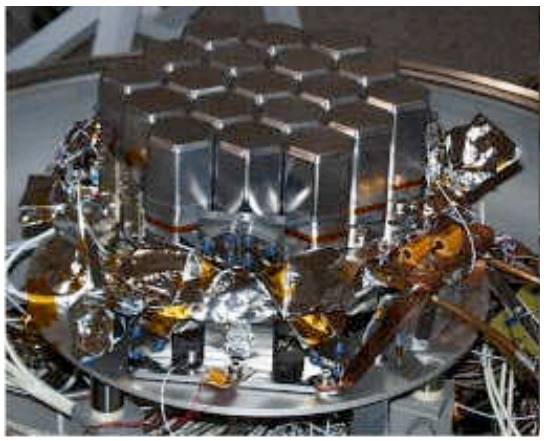


Figure 2.4: Photo of the detector array. Picture taken from [Ved+03, p. 4].

The creation of such a mask is far from trivial. Every sky position has to be encoded in a possibly unique way and the shadow patterns should be independent of each other. We do not have to rely on the spatial modulation through the mask alone, but can add temporal modulation. This leads to the *dithering strategy* of SPI: We vary the observation angle every $\sim$ 30 min by $\sim 2.1°$, to increase the visual information.

2.4 SPI gamma-ray data characteristics

2.4.1 Detectors

Failure

SPI detects gamma-rays with 19 high purity germanium detectors. In the years since 2002, four detectors failed due to unknown reasons. The lifetime of these affected detectors is set to zero.

After a detector failure, the sky response operators of Sec. 3.3 have to be recalculated.

Furthermore, they drastically change the background, since photons which hit the "dead" detectors are scattered to the neighboring detectors and increase the number of events in those.

Detector degradation

The performance of detectors gets worse over time through cosmic bombardment, which creates defects in our Germanium-detectors. This leads in general to fewer events being detected, since electrons and holes can be trapped. Therefore, in regular intervals, the detectors are annealed, which fixes the crystal structure. These degradation effects will have to be incorporated in our background model (Sec. 3.4), since the instrumental resolution will get worse over time, which leads to broader lines.

2.4.2 The background

The background are all the events we measure, which do not originate from a celestial source. If we want to observe the sky, then all the gamma-rays from somewhere else, for instance the satellite, would constitute the background. Our data is therefore a superposition of the sky signal, which we want to know, and the background signal.

Background is present in every measurement. The special challenge in gamma-ray astrophysics is, that the data is dominated by background. Just about 1 % of the measured photons are due to the sky. We will try to overcome this challenge by introducing a background model in Sec. 3.4.

The INTEGRAL satellite and its instruments are hit permanently by *cosmic rays*. These mostly consist of protons, which interact with the material of

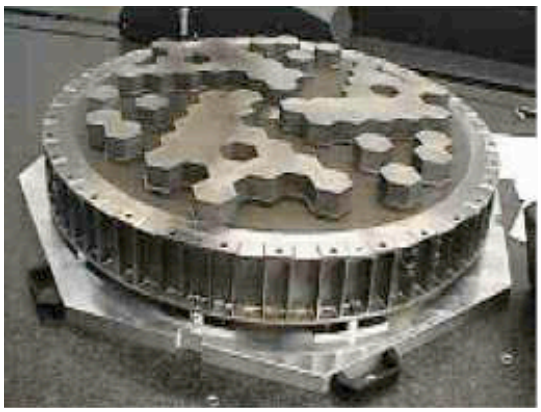


Figure 2.5: Photo of the mask. Picture taken from [Ved+03, p. 4].

the satellite. These interactions either leave behind nuclei in excited states, which then emit gamma-ray photons, or it leads to the creation of neutrons and radioactive nuclei. These decay again into an excited state, which then produces secondary photons. All these photons from excited states are measured by our Germanium-detectors.

Chapter 3

Mathematical problem formulation

Our final goal is to reconstruct the sky and therefore we have to define a mathematical description of our measurement problem. Since the inference algorithm, which we will derive in Chapter 5, is based on methods of information field theory (IFT) [EFK09], we will first give a short introduction to that theory in Sec. 3.1. This will lead to an understanding that the whole measurement process can be described by a linear response operator. If we apply this to SPI in Sec. 3.2, it is natural to split the linear operator into two independent parts: one describing the impact of the sky signal on the measured data and the other one describing the background. In Sec. 3.3, we will explain how the sky operator can be constructed. The construction of a background response operator is not straight-forward and not unique. We will analyse in Sec. 3.4, how we can extract so called "patterns" from the background, which we will use to construct a very general operator in Sec. 3.5, which only lacks a proper discretization with respect to time.

3.1 Formulation in IFT

A field $\boldsymbol{\rho}$ in physics is a function over a continuous domain. The underlying function space depends on the problem under consideration. For example, the temperature in a room could be described by a scalar function, which maps coordinates of the room to the temperature values, i.e.

$$\boldsymbol{\rho} : \text{“Room”} \subset \mathbb{R}^3 \to \mathbb{R}.$$

In our abstract setting we will denote the unknown domain as Ω, such that

$$\boldsymbol{\rho} : \Omega \to \mathbb{R}.$$

We want to determine $\boldsymbol{\rho}$ from our measured data $\boldsymbol{d}$. This is in practice impossible since we only have a finite amount of measurements at our disposal. IFT tries to approach this problem by introducing a probabilistic Bayesian framework, in which we can incorporate additional prior knowledge, constraints and assumptions in a controlled way. One such example would be the smoothness of our fields.

For a probabilistic framework, we need a probability space for the objects of interest, which are functions. Let us consider a finite domain $\Omega^N = \{x_1, \dots x_N\} \subset \Omega$. Then our field could be described by an N dimensional vector

$$\boldsymbol{\rho} = \begin{pmatrix} \boldsymbol{\rho}(x_1) \\ \vdots \\ \boldsymbol{\rho}(x_N) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\rho}_1 \\ \vdots \\ \boldsymbol{\rho}_N \end{pmatrix}, \quad x_i \in \Omega^N$$

and our configuration space

$$C^N = \left\{ (\boldsymbol{\rho}_1, \dots \boldsymbol{\rho}_N)^T \mid \boldsymbol{\rho} : \Omega^N \to \mathbb{R} \right\}$$

would be an N dimensional vector space. We can then define a probability density $p(\boldsymbol{\rho}) : C^N \to \mathbb{R}$ and calculate the expectation value of an arbitrary function $f(\boldsymbol{\rho})$ depending on our signal by

$$\begin{aligned} \langle f(\boldsymbol{\rho}) \rangle_{(\boldsymbol{\rho})} &= \int d\boldsymbol{\rho}\, p(\boldsymbol{\rho})\, f(\boldsymbol{\rho}) \\ &= \prod_{i=1}^{N} \left(\int d\boldsymbol{\rho}_i \right) p(\boldsymbol{\rho}) f(\boldsymbol{\rho}) \end{aligned}$$

All field theories now require that a unique limit for $N \to \infty$ exists, and that we therefore get a probability measure over a function space, through the finite approximations of our domain. To date, a mathematical rigorous description, which contains all use cases, is still an open research problem[Enß].

We have to find a relation between the signal of interest $\boldsymbol{\rho}$ and the measured data $\boldsymbol{d}$. We assume they are linked by a linear operator $\widehat{\boldsymbol{R}}$ and an unknown noise field $\boldsymbol{n}$, such that

$$\boldsymbol{d} = \widehat{\boldsymbol{R}}\boldsymbol{\rho} + \boldsymbol{n}.$$

The operator $\widehat{\boldsymbol{R}}$ describes the deterministic parts of our measurement process and encodes everything we know about it. It produces an expectation value for the data generated by the signal

$$\langle \boldsymbol{d} \rangle_{(\boldsymbol{\rho})} = \widehat{\boldsymbol{R}}\boldsymbol{\rho}.$$

We further assume that $\widehat{\boldsymbol{R}}$ is given by an integral kernel, such that

$$\langle \boldsymbol{d} \rangle_i = \int_x dx\, R(i,x)\, \boldsymbol{\rho}(x).$$

We hereby exclude certain unbounded operators, like $\widehat{\boldsymbol{R}} = {}^{d}/_{dx}$, which are not completely contrived, if we think about differentiator circuits in electronics.

The noise field $\boldsymbol{n}$ on the other hand, models everything non-deterministic. It is defined by

$$\boldsymbol{n} = \boldsymbol{d} - \langle \boldsymbol{d} \rangle = \boldsymbol{d} - \widehat{\boldsymbol{R}}\boldsymbol{\rho}.$$

In the above example of measuring the room temperature, we could measure twice at the same spatial position $y \in \Omega$ and assume that the temperature does not change during the measurement. $\widehat{\boldsymbol{R}}$ would then have the form

$$\boldsymbol{R}(x) = \begin{pmatrix} \delta(x-y) \\ \delta(x-y) \end{pmatrix},$$

and we would expect to measure

$$\widehat{\boldsymbol{R}}\boldsymbol{\rho} = \begin{pmatrix} \int_x dx \delta(x-y)\, \boldsymbol{\rho}(x) \\ \int_x dx \delta(x-y)\, \boldsymbol{\rho}(x) \end{pmatrix} = \boldsymbol{\rho}(y) \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

In practice we will not measure the same value twice, but

$$\boldsymbol{d} = \widehat{\boldsymbol{R}}\boldsymbol{\rho} + \boldsymbol{n} = \begin{pmatrix} \boldsymbol{\rho}(y) + n_1 \\ \boldsymbol{\rho}(y) + n_2 \end{pmatrix},$$

due to noise. It will be our goal in Chapter 5 to develop an algorithm to filter out the noise field.

3.2 Formulation for SPI

For the SPI-problem, data originates from both the sky and the background. Therefore we write

$$\boldsymbol{d} = \widehat{\boldsymbol{R}}^{(s)}\boldsymbol{\rho}^{(s)} + \widehat{\boldsymbol{R}}^{(b)}\boldsymbol{\rho}^{(b)} + \boldsymbol{n},$$

where $\boldsymbol{d}$ is the measured data, $\boldsymbol{\rho}^{(s)}$ our sky image, $\boldsymbol{\rho}^{(b)}$ a description of the background and $\boldsymbol{n}$ the poissonian noise field. The operators $\widehat{\boldsymbol{R}}^{(s)}$ and $\widehat{\boldsymbol{R}}^{(b)}$ describe the measurement process of the telescope, and the behaviour of our background, respectively. The sky response operator is explained in Sec. 3.3, a description of the background is given in Sec. 3.4 and 3.5. We assume that the operators are linear, or that we can approximate them as linear in the given energy range at about 511 keV. This is intuitive sense, since doubling the signal strength, should double the number of counts. As before we can get rid of the noise by taking the expectation value

$$\langle \boldsymbol{d} \rangle = \widehat{\boldsymbol{R}}^{(s)}\boldsymbol{\rho}^{(s)} + \widehat{\boldsymbol{R}}^{(b)}\boldsymbol{\rho}^{(b)}.$$

3.3 The signal response operator

3.3.1 Construction

The signal response operator models the measurement process of the telescope, here mainly the coded-mask technique. It consists of three parts, which can be constructed independently:

1. One part has to incorporate the orientation of the telescope. The telescope has a field of view of 16°. We want to observe the full sky, which is a unit sphere with 4π sr. Hence we need a transformation of the locally observed sky patch to the global sky. This is done by axis rotations.

2. Another part has to describe the shadow pattern on the detector plane, which can be created by ray tracing.

3. Finally, we need to know the behaviour of the single detectors which is determined from Monte-Carlo simulations.

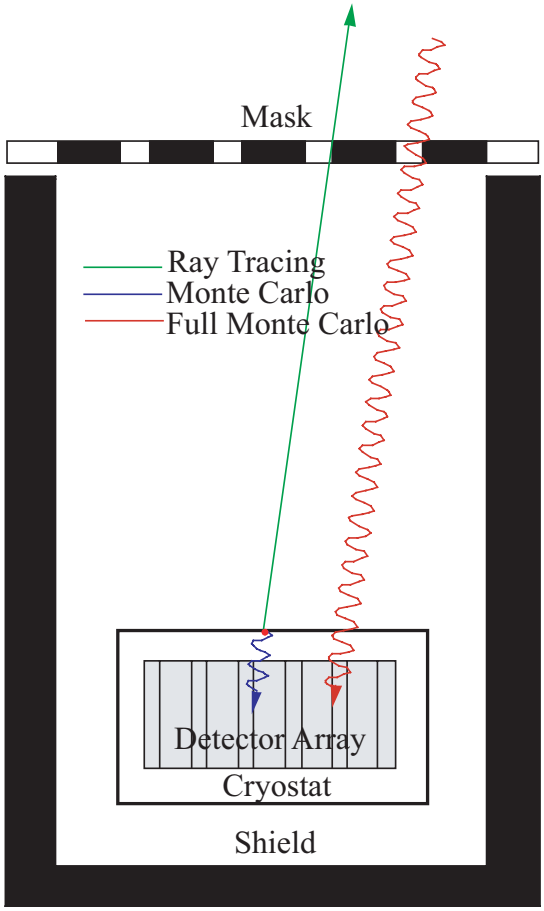


Figure 3.1: Illustration of the simulation process for the signal response operator. Picture taken from [Stu+03].

Parts 2 and 3 are illustrated in Fig. 3.1. They have to be redone, if the hardware of the telescope changes, for example in case of a detector failure. Only part 1 has to be recalculated for every observation, which points to a new spot on the sky hemisphere.

The execution of these steps yields the response operator for INTEGRAL/SPI

$$\widehat{\boldsymbol{R}}^{(s)} = \left(\widehat{\boldsymbol{R}}^{(s)}_{\mathrm{pderc}}\right),$$

which is a five dimensional array. Here the index p specifies the pointing, d the detector and e the energy bin. The indices r and c are the row and column index of our discretized picture.

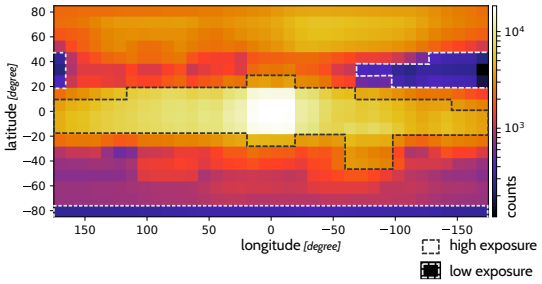


Figure 3.2: Exposure map for our signal operator.

Since we are only interested in the 511 keV line, we only use events in a small energy range of 509.5 keV to 512.5 keV, which covers the majority of the line photons. Hence we can neglect the energy dependence and omit its index,

$$\widehat{\boldsymbol{R}}^{(s)} = \left(\widehat{\boldsymbol{R}}^{(s)}_{\mathrm{pdrc}}\right).$$

In practice p, d, r and c are integer numbers, where p ranges from 0 to 73589, d from 0 to 18 and r and c depend on the image resolution, i.e. the number of pixels. For example a 18×36 pixel resolution was used in this thesis for testing purposes, so that r ranges from 0 to 17 and c from 0 to 35, corresponding to pixel sizes of $10° \times 10°$.

3.3.2 Exposure map

This thesis uses screened data, ranging from revolution 21 to revolution 1279, and consists of 73590 pointings in total. In Figure 3.2 we have an exposure map of our data set. Such a map can be constructed by counting how often $\widehat{\boldsymbol{R}}^{(s)}$ contains a non zero entry, i.e.

$$(\text{exposure map})_{\mathrm{rc}} = \left(\text{count where } \widehat{\boldsymbol{R}}^{(s)}_{\mathrm{pdrc}} \neq 0\right).$$

This determines, in which areas we have a high observation time and consequently contribute much to the total data. We note that most of the observations are at the galactic centre, or follow the galactic disk (see the black dotted line in Fig. 3.2). There are very few observations at the bottom and two small gaps in the upper right, and upper left (see the white dotted line in Fig. 3.2). This will result in imprecise results in these neglected regions,

as

$$\text{significance} \propto \sqrt{\text{observation time}}.$$

This effect will be visible in our simulations, see Sec. 6.1.

3.4 Background patterns

In this section we will take the first step towards constructing a response operator for the background. We put all the information and all our assumptions about the behaviour of the background into this operator.

These assumptions are generally called the "background model" and the construction process requires building up an "instrumental database".

3.4.1 Background Model

We will use a background model first introduced by Thomas Siegert [Sie13]. It is published in [Sie+16] and thoroughly explained in [Sie17]. The model is based on the assumption, that the intensity pattern due to the background on the detector array stays constant, as opposed to the pattern due to the sky, which changes with every pointing. We will discuss this in detail in the following sections.

3.4.2 Detector patterns

Recall how the sky casts a shadow on the bottom of the satellite, see Figure 2.3. We expect that the shadow pattern highly depends on the direction in which the telescope is pointing. Even small changes in the pointing direction should have an effect on the shape of the sky pattern.

On the other hand, the background also produces some kind of pattern, but we would expect it to be nearly independent of where the telescope is pointed. It should therefore, at least for a while, be constant, as the telescope scans a region on the sky with small steps. Another way to formulate this is to say that the "detector ratios" r_d for the dth-detector stay constant, that means

$$r_\mathrm{d} = \frac{d_{\mathrm{p}\,\mathrm{d}} \cdot t_{\mathrm{d,p}}^{-1}}{\sum_\mathrm{d} d_{\mathrm{p}\,\mathrm{d}} \cdot t_{\mathrm{d,p}}^{-1}} \cdot \text{ number of detectors } \approx \text{const},$$

where $t_{\mathrm{d,p}}$ is the lifetime of the detector for that pointing. We describe the full construction of these patterns r_d in Sec. 3.4.3 and for now assume that we already have them at our disposal. Then we have to scale these ratios with the correct coefficients b_p, and thereby fit it to the data, i.e.

$$d_{\mathrm{p}\,\mathrm{d}}^{(b)} = r_\mathrm{d} \cdot b_\mathrm{p} + \text{"sky and noise"}.$$

This procedure is illustrated in Figure 3.3. Of course also events from the sky signal contribute, but smear out due to an average of many pointings, see Sec. 3.4.3. We assume that these sky contributions fit nicely into the "left over" parts, which are not explained by our background data. This is illustrated in Figure 3.4 for a single pointing.

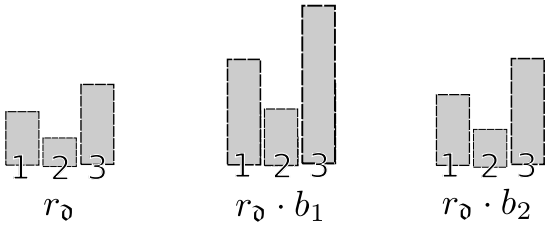


Figure 3.3: A telescope with 3 detectors and without noise. The background pattern on the left is given by $r = (2, 1, 3)$. It is scaled in pointing 1 with $b_1 = 2$, s.t. $r \cdot b_1 = (4, 2, 6)$. In pointing 2 a different coefficient $b_2 = 1.5$ was determined, s.t. $r \cdot b_2 = (3, 1.5, 4.5)$.

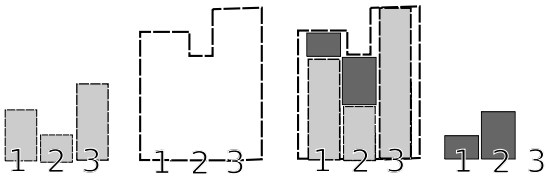


Figure 3.4: A telescope with 3 detectors. The background pattern is shown on the left. The "measured data" is displayed in the second image. The third image shows the background pattern fitted to our data together with sky contributions. In the last picture the sky contributions are isolated.

The procedure is in practice more complicated, since the patterns also depend on p, the pointing index, such that we get patterns of the form $r_{\mathrm{p}\,\mathrm{d}}$. This "change over time" is due to the detector failures introduced in Sec. 2.4.1 and the detector degradation of Sec. 2.4.1.

3.4.3 Creating the pattern

We now discuss, how the background patterns, introduced in Section 3.4.2, are generated from the full INTEGRAL/SPI mission database. We will try to introduce here the main ideas and reference to [Sie13] and [Sie17] for details.

Studying the background

How are we able to obtain any information about the background from our data? Recall that the sky only contributes 1% to the total data, and let us assume that the background is independent of the mask. If we add up the data of several pointings we expect that

1. the background dominates the summed data and

2. the events due to the sky have been smeared out and can be neglected.

Typically, the pointings of one revolution[1] of the INTEGRAL satellite are added up. Since we expect that crossing the van Allen radiation belts significantly changes the performance of our detectors, it is not reasonable to add up more data, except when more statistics are required.

We denote with

$$\bar{d}_{\mathrm{r\,d\,e}} = \sum_{\mathrm{p}\in\{\text{rth revolution}\}} d_{\mathrm{p\,d\,e}},$$

the summed data of revolution r over all pointings p belonging to that revolution, for a fixed detector d and a fixed energy bin e. We have reintroduced the suppressed index e since spectral properties are important in the used model.

Separating continuum and line spectrum

In Figure 3.5 we show a spectrum over the pointings of the first revolution for detector 1. We can see that the spectrum can be split up into a line $L(E)$ and a continuum $C(E)$ part. Both can be

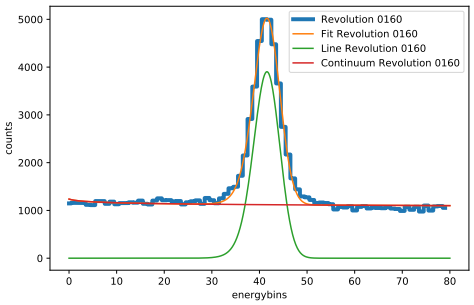


Figure 3.5: The spectrum of the 511 keV line of the 160th revolution, split up into a possible line and continuum part, with a least squares fit.

approximated by functions of the form

$$\begin{aligned} C(E) &= C_{0\,\mathrm{d}} \left(\frac{E}{511\mathrm{keV}}\right)^{\alpha_\mathrm{d}}, \\ G(E) &= A_{0\,\mathrm{d}} \exp\left(-\frac{(E-E_{0\,\mathrm{d}})^2}{2\sigma_\mathrm{d}^2}\right), \\ T(E) &= {}^1\!/_{\tau_\mathrm{d}} \exp\left(-\frac{E}{\tau_\mathrm{d}}\right) \text{ and} \\ L(E) &= (G \otimes T)\,(E)\,, \end{aligned}$$

where $C_{0\,\mathrm{d}}$, α_d, $A_{0\,\mathrm{d}}$, $E_{0\,\mathrm{d}}$, σ_d and τ_d are parameters, which are fitted to our data with a Markov-Chain-Monte-Carlo-method.

Separating the background into line and continuum part, is important, since both parts show a different temporal behaviour. The line part tends to degenerate significantly faster than the continuum part.

We can now do this for every revolution and get a family of parameters $(C_{0\,\mathrm{r\,d}}, \alpha_{\mathrm{r\,d}}, A_{0\,\mathrm{r\,d}}, E_{0\,\mathrm{r\,d}}, \sigma_{\mathrm{r\,d}}, \tau_{\mathrm{r\,d}})$, which describe our background.

Extracting the pattern

In the last step we extend the background patterns to single pointings. We do this by scaling the background

$$\begin{aligned} B_{\mathrm{p\,d\,e}} = \theta_{\mathrm{p\,d\,e}} &\cdot C(E_\mathrm{e}; C_{0\,\mathrm{r\,d}}, \alpha_{\mathrm{r\,d}}) \\ &+ \vartheta_{\mathrm{p\,d,e}} \cdot L(E_\mathrm{e}; A_{0\,\mathrm{r\,d}}, E_{0\,\mathrm{r\,d}}, \sigma_{\mathrm{r\,d}}, \tau_{\mathrm{r\,d}}) \end{aligned}$$

[1] One revolution takes roughly three days.

to the measured data $d_{\mathrm{p\,d}}$ by calculating the coefficients $\theta_{\mathrm{p\,d}}$ and $\vartheta_{\mathrm{p\,d}}$ for every pointing, using a maximum likelihood method. This corresponds to explaining the measured data by only using the background and completely neglecting the sky, and also accounts for the just mentioned different temporal behaviours of line and continuum.

We eliminate the energy dependence by summing up over a small energy range around the 511 keV bin

$$r_{\mathrm{p\,d}} = \sum_{\mathrm{e}} B_{\mathrm{p\,d\,e}}, \tag{3.1}$$

as explained in Sec. 3.3.1.

3.5 The background operator

3.5.1 Motivation

Now we have an adequate approximation of the background patterns available, given by $r_{\mathrm{p\,d}}$. Since we constructed $r_{\mathrm{p\,d}}$ by explaining all our measured data with the background, it also contains data from the sky and hence overestimates the background.

Nevertheless, we can use $r_{\mathrm{p\,d}}$ as an approximation for the background. Then we must assume that we can neglect the sky contributions and that the sky signal still fits in the areas, which cannot be explained by the background pattern[Gha17, Sec. 5.1].

On the other hand, we do not know the amplitude of the patterns, and introduce a free parameter for the amplitude of each pointing [Gha17, Sec. 5.3]. The disadvantage is, that we then add a lot of free parameters, even more than we use for our sky signal. To see this consider a typical sky image image of $360° \times 180°$, with a pixel size of $10° \times 10°$. This image consists of $36 \cdot 18 = 648$ pixels. The background on the other hand has 73590 pointings in the used dataset. We then have ~ 100-times more degrees-of-freedom in the background, than in the sky image. Hence the background can be flexibly fitted to the measurements and thus it can also explain data originating from the sky signal.

Both approaches are extreme cases, where we either heavily rely on the precomputed background and completely neglect the sky contributions, or deliberately destroy information. Here in this thesis we will try to find an intermediate approach: Instead of introducing a free parameter for every pointing, we will scale several pointings with the same parameter. We will refer to this group of pointings as a *chunk*. To achieve this, we will introduce a *background response operator*. The *chunk size* will then be the number of pointings belonging to the chunk. All the chunks will form a *partition* of the pointings. The number of chunks are our *degrees-of-freedom* with respect to the background.

The question remains, which pointings should be scaled with the same amplitude, i.e. which partition we should use. Naively we would expect both a *spatial* and *temporal* dependence: If we point the telescope to the same region in the sky, we are exposed to the same sky flux. Hence the magnitude of the events originating from the sky should be equal. If the background flux is unchanged the ratio between sky and background will be similar for those observations thus it seems legit to use the same amplitude for the background pattern. On the other hand we know that the performance of the detector changes over time, by the eleven-year solar cycle, detector failures, passing the Van-Allen radiation belt or simple detector degradation. Therefore we expect that we can use the same amplitude for successive pointings, while the amplitudes of two measurements more than one year apart might differ significantly.

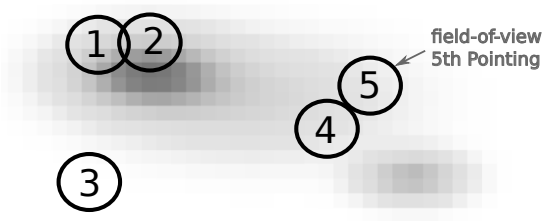
3.5.2 Definition

We will define the background response operator $\widehat{\boldsymbol{R}}^{(b)}$ in its most general form, independent of the underlying partition. For this, we need a mapping from the pointings to the chunks to which they belong, which basically contains all the information for our partition. We will denote this *partition mapping* by

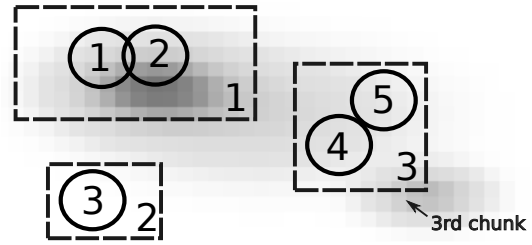
$$\boldsymbol{q} : \{1, \dots N_{\mathrm{p}}\} \to \{1, \dots N_{\mathrm{c}}\}, \tag{3.2}$$

where N_{c} is the number of chunks and N_{p} the number of pointings. The mapping is illustrated in Fig. 3.6. If $\boldsymbol{\rho}^{(b)} \in \mathbb{R}^{N_{\mathrm{c}}}$ is now a vector containing the amplitudes of the chunks, then the background response operator $\widehat{\boldsymbol{R}}^{(b)} : \mathbb{R}^{N_{\mathrm{c}}} \to \mathbb{R}^{N_{\mathrm{p}}}$ can be defined as

$$\left\langle \boldsymbol{d}^{(b)} \right\rangle_{\mathrm{p\,d}} = \left(\widehat{\boldsymbol{R}}^{(b)} \boldsymbol{\rho}^{(b)} \right)_{\mathrm{p\,d}} := r_{\mathrm{pd}} \cdot \boldsymbol{\rho}^{(b)}_{\boldsymbol{q}(\mathrm{p})}.$$



(a) An artificial sky with 5 pointings, whose field-of-view is denoted by circles.



(b) Here we have a partition of the pointings into three chunks. Chunks one and three have a chunk size of two, while the second chunk has a chunk size of one. The partition mapping of Eq. 3.2 would be

$$\boldsymbol{q}(\mathrm{p}) = \begin{cases} 1, & \text{for } \mathrm{p} \in \{1,2\} \\ 2, & \text{for } \mathrm{p} = 3 \\ 3, & \text{for } \mathrm{p} \in \{4,5\} \end{cases} .$$

Figure 3.6: Partitioning pointings in chunks.

At this point it is unclear how $\boldsymbol{q}$ should be constructed, i.e. which pointings should depend on the same parameter and how many parameters we should introduce. A few approaches how to partition the pointings will be introduced in Chapter 4, and their performance will be discussed in Chapter 6.

3.5.3 Physical interpretation of the background signal

We interpret the sky $\boldsymbol{\rho}^{(s)}$ as a field of the photon flux from the sky. $\widehat{\boldsymbol{R}}^{(s)}$ is the mathematical description of our telescope and $\widehat{\boldsymbol{R}}^{(s)}\boldsymbol{\rho}^{(s)}$ are the photon counts due to the sky. The physical interpretation for $\boldsymbol{\rho}^{(b)}$ and $\widehat{\boldsymbol{R}}^{(b)}$ is more abstract, whereas $\widehat{\boldsymbol{R}}^{(b)}\boldsymbol{\rho}^{(b)}$ are still the photon counts due to the background.

In [Gha17] the background patterns in Eq. 3.1 were converted into rates by considering the detector lifetimes and normalized to the intensity seen in the 13th detector. Hence $\boldsymbol{\rho}^{(b)}$ could be interpreted as the background signal at the 13th detector. The intensity of the background signal was deliberately destroyed, so that it could be inferred again from the data. This reconstruction was then taken as an indicator for the correctness of the algorithm [Gha17, p. 91].

Such a simple normalization is not possible in our case: If we use chunks, consisting of several pointings, the 13th detector would always have the same value inside the chunk. An appropriate generalization would be to normalize the whole chunk, such that the 13th detector has a mean of one inside the chunk.

We refrain from applying such a normalization, since the constructed background already contains a good approximation of the intensity of the background signal. Deleting that information does not seem sensible. Hence also our interpretation of $\boldsymbol{\rho}^{(b)}$ has to change. By construction $r_{\mathrm{p\,d}}$ $r_{\mathrm{p\,d}}$ explains the photon counts just with the background pattern. If we consider a chunk size of 1, $\boldsymbol{\rho}_{\mathrm{p}}^{(b)} = 1$ would mean that the data at the pth pointing consists mostly of background and $\boldsymbol{\rho}_{\mathrm{p}}^{(b)} = 0$ that no background is present. Thus we can interpret $\boldsymbol{\rho}_{\mathrm{p}}^{(b)}$ as the amount of background, which is present in the data and hence our $\boldsymbol{\rho}^{(b)}$ should therefore have similar characteristics as the background sig-

nal used in [**masha**], although it is more abstract,
but closer to the physics in the satellite.

Chapter 4

Background partitions

The goal of this chapter is to construct partitions of pointings into chunks, which we can use for the background response operator introduced in Sec. 3.4. We will start by considering only the temporal component and subdivide the pointings into chunks with equal chunk size in Sec. 4.1. After that we will consider a more advanced approach, which also includes spatial properties in Sec. 4.2.

4.1 Equal subdivisions for pointings

The most naïve way to create a partition would be to group successive pointings into chunks of fixed size. We then scale all background patterns in one chunk with the same factor of our background image. We can write down this algorithm as

$$\boldsymbol{d}_{\mathrm{p\,d}}^{(b)} = \widehat{\boldsymbol{R}}_{\mathrm{p\,d}}^{(b)} \boldsymbol{\rho}_{\lfloor \mathrm{p}/\text{chunk size} \rfloor}^{(b)},$$

where we already plugged in our partition mapping $\lfloor \mathrm{p}/\text{chunk size} \rfloor$. This approach intuitively makes sense: Many adjacent pointings will be only apart by a short spatial distance and timespan. We do not expect that the background amplitude changes much in this case. The spatial proximity of adjacent pointings gets further increased by the dithering strategy of SPI. The algorithm is illustrated in Fig. 4.1 for a chunk size of three.

Of course there are situations, where this method will fail: Consider for instance if an annealing phase falls directly into a chunk. Then the background for the pointings at the beginning of the chunk will be significantly different than for those at the end and we would construct large error, if we scale them with the same factor.

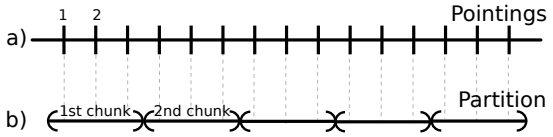


Figure 4.1: Illustration for the naive equal subdivision algorithm. Adjacent pointings are grouped into chunks of size three.

That is why we have to incorporate the revolution boundaries into our subdivision process. We do this by moving a chunk boundary, which is directly before a revolution boundary to the revolution boundary, as illustrated in Fig. 4.2. This respects the revolution bounds, but in practice gives similar results as the naïve equal subdivision algorithm.

The exact algorithm is given in Alg. 1 for reference. The approach here is expected to work well, if the chunk size is bigger than the number of pointings per revolution. However if it gets smaller, we expect that spatial properties should play a more dominant role, i.e. pointings at the same position should be scaled with the same factor. That is the reason why we develop a more advanced algorithm in Sec. 4.2, which is able to change the size of the chunks and their boundaries, depending on the spatial positions of the pointings.

4.2 K-Means algorithm

The chunks of Sec. 4.1 only incorporate temporal information. The idea now is to construct groups by combining the available temporal and spatial

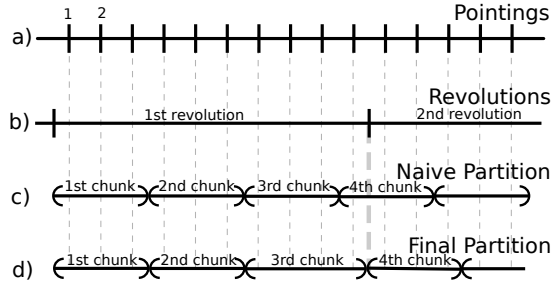


Figure 4.2: Illustration for the an equal subdivision algorithm respecting the revolution boundaries. The right boundary of the third chunk is shifted to the revolution boundary and thus becomes a chunk of size four. Note that the fourth chunk has again a chunk size of tree.

information of the pointings. For this we will apply the K-Means algorithm [Mac67] on our pointings, which will group our pointings into clusters. All background patterns whose pointings are in the same cluster will then be scaled by the same background parameter.

4.2.1 Classic K-Means algorithm

We will repeat here the main points of the algorithm and point to [Mac03, Sec. 20.1] for a more thorough introduction.

The K-Means algorithm partitions a set of N points $\{x^{(n)}\} \subset \mathbb{R}^d$ with $n = 1 \dots N$ into K groups, so-called clusters. Every cluster of points has a mean $m^{(k)}$, which can be calculated by

$$m^{(k)} = \frac{1}{N_k} \sum_{x^{(i)} \in C^{(k)}} x^{(i)}, \tag{4.1}$$

where $C^{(k)}$ is the set of all points in the kth-cluster and $N_k = |C^{(k)}|$ its cardinality.

The algorithm itself is given in Alg. 2.

Convergence: The algorithm always converges, as can be seen from a *physical* argument as follows: Imagine that every point in a cluster is connected with a spring to the mean point of its cluster. In the *assignment step*, both the cluster points and mean points cannot move, they stay fixed. A point changes its assigned cluster, if the distance to another mean point is shorter. This corresponds to a shorter spring. Therefore we minimize the energy of the system in the assignment step.

Then in the *update step*, if we release the mean points, the springs fixed to the points in the cluster will pull them to the position in which the potential energy is minimal. This corresponds to the mean position of the cluster, which we recalculate in the update step. Therefore, the update step also minimizes the energy.

The whole process is illustrated in Figure 4.3.

Input Let N_c be the optimal chunk size and N_p the total number of pointings.
Output Let $\boldsymbol{q} \in \mathbb{R}^{N_p}$ the partition mapping.

1. Calculate the revolution boundaries $\boldsymbol{r}$, such that $\boldsymbol{r}_i$ returns the index of the first pointing, which belongs to the $i+1$ revolution.
2. Let $b_{\text{last}} := 0$ be the last boundary of a chunk, initially set to the 0th pointing, $c_{\text{index}} := 0$ be the index of the current chunk, initially set to the 0th chunk.
3. Repeat until $b_{\text{last}} + N_c > N_p$:
   (a) Set $b_{\text{next}} = b_{\text{last}} + N_c$.
   (b) Calculate the nearest boundary in $\boldsymbol{r}$ with a higher pointing index as b_{next}, i.e.
   $$r_{\text{nearest}} := \underset{\boldsymbol{r}_i}{\operatorname{argmin}} \{\boldsymbol{r}_i - b_{\text{next}} | \boldsymbol{r}_i > b_{\text{next}}\}.$$
   (c) If the distance to r_{nearest} is less than the chunk size c, correct it to the revolution boundary
   (d) We update the mapping
   $$\boldsymbol{q}_i := c_{\text{index}}, \quad \forall i \text{ with } b_{\text{last}} \le i < b_{\text{next}}.$$
   (e) We start the next iteration with $b_{\text{last}} := b_{\text{next}}$ and $c_{\text{index}} := c_{\text{index}} + 1$.

Algorithm 1: Partitions consecutive pointings into chunks, which are as equal in size as possible, if we respect the revolution boundaries.

Input Let K be the number of chunks, $\{x^{(n)} \,|\, n = 1...N\}$ the positions of our points.
Output Let $\{C^{(k)} \,|\, k = 1 \dots K\}$ be a partition of our points into sets.

1. Initialize the means $\{m^{(k)}\}$ by setting them to random points of $\{x^{(n)}\}$.
2. Initialize the clusters $C^{(k)}$ by distributing the points $\{x^{(n)}\}$ between them randomly.
3. Repeat the following step until no point changes its cluster and no mean changes its position:
   (a) *Assignment-Step:* Assign every point $x^{(i)}$ to the cluster, such that the distance to the mean $m^{(k)}$ is smallest, or more precise

$$x^{(i)} \in C^{(k)} \quad \Updownarrow \quad \operatorname{dist}\left(x^{(i)}, m^{(k)}\right) = \min_{l=1\dots K} \operatorname{dist}\left(x^{i}, m^{(l)}\right)$$

   (b) *Update-Step:* Recalculate the means by Eq. 4.1.

Algorithm 2: K-Mean

The energy of a configuration is now monotonically decreasing sequence, bounded below by 0 and therefore converges.

4.2.2 Necessary adaptions

We now want to adapt the K-Means algorithm as described above to our problem of grouping the pointings with respect to their position in space and time. Consider the kth-pointing, taken at time[1] $t^{(k)}$, looking at the point in the sky with galactic longitude $l^{(k)}$ and galactic latitude $b^{(k)}$. We will group the spatial component in the vector $\boldsymbol{s}^{(k)} = (l^{(k)}, b^{(k)})$. A point in our data space would then take the form

$$\boldsymbol{x}^{(k)} = (t^{(k)}, l^{(k)}, b^{(k)}) = (t^{(k)}, \boldsymbol{s}^{(k)}) \in \mathbb{R} \times S^2.$$

We can now define a metric

$$\left\|\boldsymbol{x}^{(k)}\right\| = w_t \cdot \left|t^{(k)}\right| + w_{S^2} \cdot \left\|\boldsymbol{s}^{(k)}\right\|_{S^2},$$

where $|t|$ is the absolute value of the time coordinate, $\|\cdot\|_{S^2}$ is the default norm on the sphere, and w_t as well as w_{S^2} are weights, which allow us to give more weight to either the temporal or the spatial dimension.

However, the K-Means algorithm described above in general only works in an euclidean space, since only there, the arithmetic mean of Eq. 4.1 and the barycenter (which is used in the convergence proof) are equal, as can be seen in Fig. 4.4. We can generalize it to arbitrary manifolds by replacing Eq. 4.1 with the Frechet-Mean [2]

$$\boldsymbol{m}^{(k)} = \operatorname*{arg\,min}_{\boldsymbol{m} \in \mathbb{R} \times S^2} \sum_{\boldsymbol{x}^{(i)} \in C^{(k)}} \left\|\boldsymbol{m} - \boldsymbol{x}^{(i)}\right\|,$$

which requires more computational effort, than the arithmetic mean.

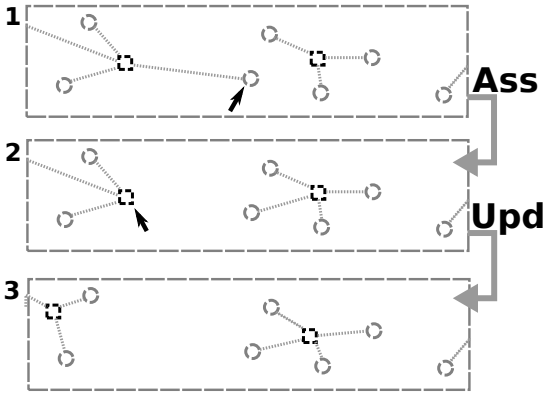


Figure 4.3: Illustration of the K-Means algorithm. The points are depicted as circles and the means as squares. Every circle is connected to its midpoint. In the assignment-step the point marked with an arrow changes its cluster, and therefore its spring is connected to its new mean point. In the subsequent update-step the means move to their new positions.

[1] More precise $t^{(k)}$ is the mean of the beginning and end of our observation. This is sufficiently accurate in practice.

[2] It can be shown that if the points are sufficiently close to each other, the Frechet-Mean is unique. This condition should always be fulfilled for our use case.

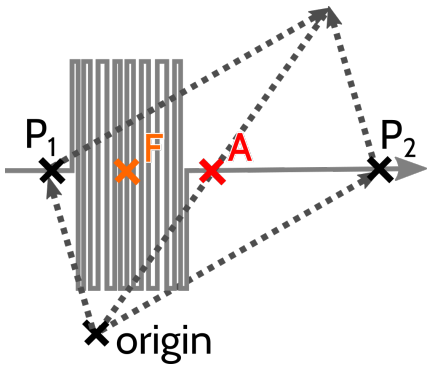


Figure 4.4: Illustrates the difference of the Frechet-Mean and the arithmetic mean on manifolds. Here the number line was embedded into two dimensional space, such that it is densely folded in a certain area. The Frechet-Mean F of two points P_1 and P_2 is equally far apart from both points, and therefore somewhere in the folded region, since there is, by construction, the majority of the curve length. The arithmetic mean does not respect the structure and lies halfway on a straight line between the points.

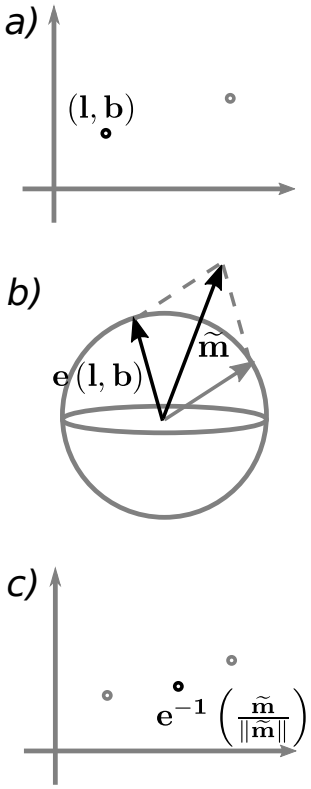


Figure 4.5: Illustration how the Frechet-Mean can be calculated efficiently with the arithmetic mean. A point (l, b) in the 2-dimensional plane of Fig. a) is mapped to $e(l, b)$ in 3-dimensional space (Fig. b)). With other embedded points we calculate the arithmetic mean $\widetilde{m}$, which we map back into the 2-dimensional space (Fig. c)).

However, it can be shown, that if there exists an *isometric embedding* of our 'manifold' into an euclidean space, then its possible to calculate the mean in the embedding and project it down onto the manifold [Pan+13, Sec. 3.1]. The result will be equal to the Frechet-Mean. In our case the embedding $\widehat{\boldsymbol{e}}$ is just the coordinate transformation of longitude l and latitude b to the sphere in 3D-space. We can therefore calculate the arithmetic mean

$$\tilde{\boldsymbol{m}}^{(k)} = \frac{1}{N_k} \sum_{\boldsymbol{x}^{(i)} \in C^{(k)}} \widehat{\boldsymbol{e}} \left(l^{(i)}, b^{(i)} \right),$$

project it back on the sphere by normalizing and reverse the coordinate transformation

$$(l_m^{(k)}, b_m^{(k)}) = \widehat{\boldsymbol{e}}^{-1} \left(\frac{\tilde{\boldsymbol{m}}^{(k)}}{\left\| \tilde{\boldsymbol{m}}^{(k)} \right\|} \right).$$

The whole process is illustrated in Fig. 4.5 on page 18.

Since $\mathbb{R}$ is an euclidean space, with the absolute value as a metric, the time coordinate can simply be calculated by

$$\boldsymbol{m}_t^{(k)} = \frac{1}{N_k} \sum_{\boldsymbol{x}^{(i)} \in C^{(k)}} \boldsymbol{x}_t^{(i)}.$$

Spatial Simulation Figure 4.6 shows how the adapted K-Means algorithm works with only spatial test data. We note that there are single clusters at the top and at the bottom. This is due to the fact, that the angular distances in these regions are small, even though the distances look large in the Mercator projection. We also note that the blue cluster wraps around the image, as we would expect.

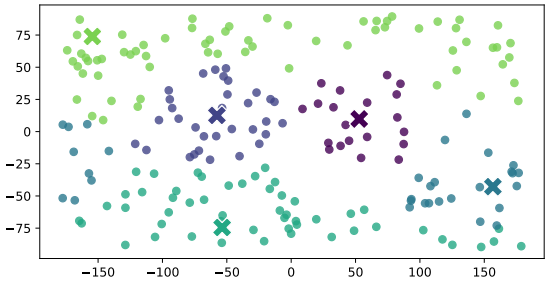


Figure 4.6: Simulation of the K-Means algorithm for 200 points distributed in 5 clusters. The points are depicted as circles and the means as crosses. Every cluster has its own unique color.

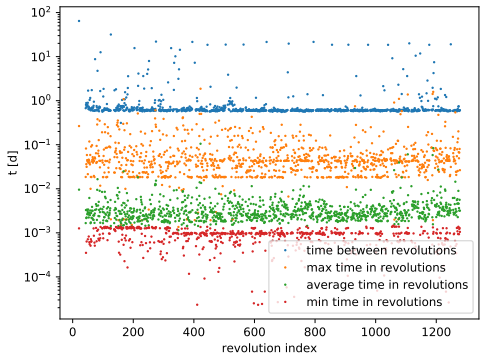


Figure 4.7: Plot of the time delay between to *neighbouring* revolutions as well as the maximum, average and minimum delay *inside* the revolution.

Means The end result of the K-Mean depends a lot on the starting values of the means, since in practice the means cannot move too much, because they cannot escape out of local energy minimum. This is a nuisance for many applications, but sufficient for our use case. Recall that our main goal is to correct the chunks of the equal subdivision approach by a small amount. If we therefore choose the initial values for our means uniformly from all pointings, the first iteration will produce something very similar to the equal subdivision. Subsequent iterations will then (more or less) shift the position of the chunks and their boundaries.

Weights The main use of the weights is to remove the physical units of our quantities, and to give a parameter for fine tuning whether temporal or spatial distances should be weigthed stronger. We typically used

$$1 = w_t \cdot 5 \text{ hours}$$
$$1 = w_{S^2} \cdot 2 \text{ degree}.$$

4.2.3 Revolution boundaries

In Sec. 4.1, we had to modify the subdivision algorithm, to agree with the revolution boundaries. As can be seen from Fig. 4.7 this is not compulsory for the K-Means algorithm, since the delay between adjacent pointings of different revolutions is larger than the average delay between pointings inside of a revolution. Therefore, as long as the weight for the time component w_t is sufficiently big, the distance of adjacent pointings of different revolutions will be large. Hence it is likely that these points will be separated, even though it is not explicitly enforced and might fail.

We can enforce the revolution boundaries, if we apply the K-Means algorithm just on pointings belonging to the same revolution. The algorithm, which we will use is given in Alg. 3.

Input Let N_m be the number of chunks per revolution.
Output Let $\boldsymbol{q} \in \mathbb{R}^{N_p}$ the mapping introduced in the previous section, with the total number of pointings N_p.

1. Calculate the revolution boundaries $\boldsymbol{r}$, such that $\boldsymbol{r}_i$ returns the index of the first pointing, which belongs to the $i+1$ revolution.
2. Repeat for revolution $i = 1 \dots |\boldsymbol{r}|$:
   (a) Extract the pointings of the ith revolution. Let p_{first} be the index of the first pointing belonging to the revolution and N_r the revolutions number of pointings.
   (b) Apply the K-Means algorithm with N_m means on the extracted pointings, to get a mapping m, between pointings and clusters, such that the kth pointing of the revolution is in the m_kth cluster.
   (c) Update the mapping $\boldsymbol{q}$ such that all pointings of the same cluster are scaled by the same factor, i.e.

$$\boldsymbol{q}_{k+p_{\text{first}}} := m_k + i \cdot N_m \quad \forall k.0 \leq k < N_r,$$

   where $k+p_{\text{first}}$ is the global index the kth pointing of the revolution and $m_k + i \cdot N_m$ shifts the cluster numbers of the revolution, such that they are unique globally.

Algorithm 3: Modified K-Mean

Chapter 5

Image reconstruction

5.1 Mathematical Basis

5.1.1 Bayesian inference

The main goal of image reconstruction is to determine the expectation value of an unknown signal $\langle s \rangle$, where s can be a scalar, vector or field. If we were given the probability density $p(s|d)$ – the probability that s is our signal under the condition that we measured d – then we could easily calculate the mean by

$$\langle s \rangle = \int ds\, s \cdot p(s|d).$$

$p(s|d)$ is referred to as the *posterior probability density*, which is the desired value of the experiment, for which we have no direct expression. Switching s and d gives the *likelihood* $p(d|s)$, for which we often know a formula from theoretical considerations, but which is not the desired quantity, since s is unknown. Bayes theorem allows us to find an expression for the posterior in terms of the likelihood, i.e.

$$p(s|d) = \frac{p(s,d)}{p(d)} = \frac{p(d|s)p(s)}{p(d)}, \tag{5.1}$$

where $p(d)$ is the *evidence* and $p(s)$ the *prior probability*. The evidence $p(d)$ contains no explicit information about our signal, and can, for many applications, be neglected. The prior probability $p(s)$ contains all the assumptions we make about the signal, before we measure any data. Its choice is oft highly subjective and open for discussion.

5.1.2 Maximum a posteriori estimate

If we want to determine $\langle s \rangle$, and if our distribution is sufficiently symmetric, the maximum of the posterior distribution $p(s|d)$ is a often good estimate. That means

$$\langle s \rangle \approx s_{\max}, \quad \text{where} \quad s_{\max} := \operatorname{argmax} p(s|d).$$

This approximation is called the ***m**aximum **a** **p**osteriori estimate* (MAP). The estimate is exact for symmetric distributions, for instance the Gaussian shown in Figure 5.1. It fails in case of the highly unsymmetrical distributions, e.g. the Inverse-Gamma function in Figure 5.2, where mean and maximum are apart. In the following we will use the maximum a posteriori probability estimate for our inference problem.

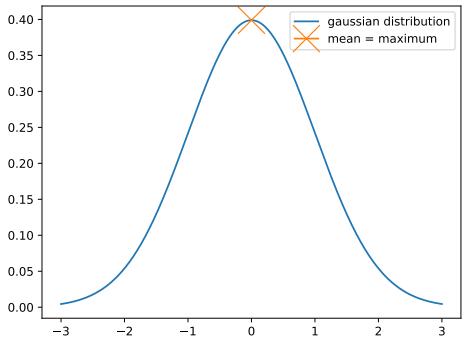


Figure 5.1: Illustration for a symmetric distribution, to which the MAP estimate would apply, since here the maximum and mean are identical.

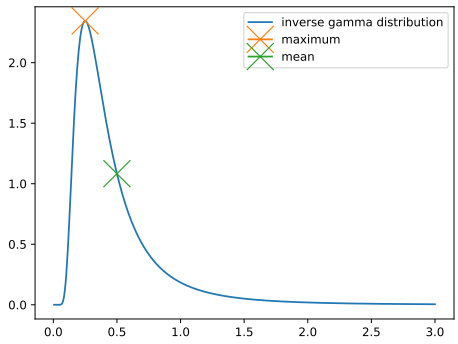


Figure 5.2: Illustration for an unsymmetrical distribution, where the MAP estimate fails, since mean and maximum are different.

Since the gradient of a function is zero at its maximum, i.e.

$$\left.\frac{\partial p(s|d)}{\partial s}\right|_{s=s_{max}} = 0, \tag{5.2}$$

we can use this necessary condition to calculate the maximum.

5.1.3 Probability Hamiltonians

The negative log-probability of the joint probability density $p(d, s)$ of d and s is often referred to as information Hamiltonian

$$H(s, d) := -\log p(d, s).$$

This definition allows us to express the posterior probability similar to the probability in thermodynamics, since

$$p(s|d) = \frac{p(d, s)}{p(d)} = \frac{1}{Z} e^{-H(s,d)},$$

where the evidence becomes the partition function

$$Z = p(d) = \int ds\, p(d, s) = \int ds\, e^{-H(s,d)}.$$

Although, this analogy only holds for the joint probability density, we will extend the use of the terminology and notation to every log-probability.

We often prefer the negative log-probability over traditional probability if we work in a maximum a posteriori setting, because the logarithm does not change the position of our maxima and minima, since it is a strictly monotonic increasing function.

Working with the log-probability further ensures, that the probabilities are positive, since

$$p = \exp(-H) \geq 0.$$

On the other hand the product of probabilities for independent events becomes a sum. For instance

$$p(A, B, C) = p(A)p(B)p(C)$$

becomes

$$H(A, B, C) = H(A) + H(B) + H(C).$$

Applied to the Bayes formula of Eq. 5.1, we get

$$H(s|d) = H(s, d) - H(d) = H(d|s) + H(s) - H(d).$$

This is especially convenient if we differentiate the Hamiltonian for s, i.e.

$$\frac{\partial H(s|d)}{\partial s} = \frac{\partial H(d|s)}{\partial s} + \frac{\partial H(s)}{\partial s},$$

since all multiplicative terms not depending on s vanish and therefore calculating the MAP-estimate by Eq. 5.2 is simpler.

Since we are aiming for the MAP estimate, we will often omit all the terms not depending on s from our equation and therefore use $\propto$ instead of the equality sign.

5.1.4 Uncertainty estimate

In this section we want find an error estimate for the MAP estimate. The main idea will be to approximate our posterior with a Gaussian distribution.

Recall that for a one dimensional Gaussian distribution

$$p(x|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma}\right)$$

with mean μ, a random event x will be in the range of $[\mu - \sigma, \mu + \sigma]$ with a probability of 68.3%. If we increase the range to $\pm 2\sigma$, we get a 95.4% probability. This can be generalized to higher dimensions,

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = \frac{1}{\sqrt{|2\pi\widehat{\boldsymbol{D}}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \widehat{\boldsymbol{D}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right),$$

where $|2\pi\widehat{\boldsymbol{D}}|$ is the determinant of the linear operator.

We now want to approximate our prior as a Gaussian distribution. To do this we do a Taylor Expansion around the maximum $\boldsymbol{s}_{\max}$

$$\begin{aligned} H(s|d) =& H(\boldsymbol{s}_{\max}) + \left.\frac{\partial H}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}_{\max}} (\boldsymbol{s}-\boldsymbol{s}_{\max}) \\ &+ \frac{1}{2}\cdot(\boldsymbol{s}-\boldsymbol{s}_{\max})^\dagger \left.\frac{\partial H}{\partial \boldsymbol{s}\partial \boldsymbol{s}}\right|_{\boldsymbol{s}_{\max}} (\boldsymbol{s}-\boldsymbol{s}_{\max}). \end{aligned}$$

Conveniently the gradient disappears at the maximum $\frac{\partial H}{\partial \boldsymbol{s}}|_{\boldsymbol{s}_{\max}} = 0$ and $H(\boldsymbol{s}_{\max})$ is just a constant. We can now plug in $H(\boldsymbol{s}|\boldsymbol{d}) = -\log p(\boldsymbol{s}|\boldsymbol{d})$ and solve for $p(\boldsymbol{s}|\boldsymbol{d})$. We get

$$p(s|d) \approx A \exp\left(-\frac{1}{2}(\boldsymbol{s}-\boldsymbol{s}_{\max})^\dagger \left.\frac{\partial H}{\partial \boldsymbol{s}\partial \boldsymbol{s}}\right|_{s_{\max}} (\boldsymbol{s}-\boldsymbol{s}_{\max})\right),$$

where A is some normalization constant.

We can now compare our approximation to the multidimensional Gaussian. Thus we can identify

$$D_{xy} = \left(\frac{\partial H}{\partial s_x \partial s_y}\right)^{-1}_{\boldsymbol{s}_{\max}}.$$

Note that D_{xy} denotes the correlation of the pixel at position x with the pixel at position y. Often we are only interested in the diagonal part of the operator D_{xx}, which corresponds to the uncertainty of the pixel. To represent a pixel value with its uncertainty we can in principle write $s_x \pm \sqrt{D_{xx}}$. That means, the "real" image value of the pixel x is within these bounds with a probability of 68.3%.

It is often convenient, that we only have to calculate the diagonal part of the operator, since calculating the full inverse of the Hamiltonian is costly.

5.2 Diffuse D^{3}PO-algorithm

We will now try to solve the equation

$$\boldsymbol{d} = \widehat{\boldsymbol{R}}\boldsymbol{\rho} + \boldsymbol{\mu} + \boldsymbol{n}$$

for the *diffuse* signal $\boldsymbol{\rho}$, where $\boldsymbol{d}$ is the measured data, $\widehat{\boldsymbol{R}}$ the response operator, $\boldsymbol{n}$ the poissonian noise and $\boldsymbol{\mu}$ a known offset. Note that all quantities except $\boldsymbol{\rho}$ and $\boldsymbol{n}$ are known to us.

This problem formulation is general enough for our purposes, as we will see in Sec. 5.3.

We will repeate the derivation of D^{3}PO[SE15], but omit the point-like signal and add the known offset.

5.2.1 Enforcing positive signals

We assume that $\boldsymbol{\rho}$ should be strictly positive, since a negative photon flux is not physically correct. To enforce this constraint, we can write it as a log-normal field, i.e.

$$\boldsymbol{\rho} = \rho_0 e^{\boldsymbol{w}},$$

where $\rho_0 > 0$ is a constant containing the physical units and $\boldsymbol{w}$ is the logarithm of $\boldsymbol{\rho}/\rho_0$. Note that $e^{\cdot}$ means element-wise exponentiation.

For convenience we can choose ρ_0 such that $\langle \boldsymbol{w} \rangle \approx 0$. To see that this is always possible consider

$$\langle \boldsymbol{w} \rangle = \left\langle \log \frac{\boldsymbol{\rho}}{\rho_0} \right\rangle = \langle \log \boldsymbol{\rho} \rangle - \rho_0.$$

If we demand $\langle \boldsymbol{w} \rangle \stackrel{!}{=} 0$ and solve for ρ_0 we get

$$\rho_0 = \langle \log \boldsymbol{\rho} \rangle .$$

For testing, we can always calculate the exact value of ρ_0. For real data we should always be able to make a good guess about the value of ρ_0 by considering previously published results.

From now on we will try to solve

$$\boldsymbol{d} = \widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu} + \boldsymbol{n},$$

for $\boldsymbol{w}$, where we absorbed ρ_0 into $\widehat{\boldsymbol{R}}$.

5.2.2 Poissonian likelihood

The Poisson distribution, is defined as

$$p(d|\lambda) = e^{-\lambda}\frac{\lambda^d}{d!},$$

where d is the number of detected photons, and $\lambda = \langle d \rangle$ is its mean.

If we assume that the noise of all our data points is independent and follows a Poisson distribution we have

$$p(\boldsymbol{d}|\boldsymbol{\lambda}) = \prod_k p(d_k|\lambda_k) = \prod_k e^{-\lambda_k}\frac{\lambda_k^{d_k}}{d_k!}.$$

Rewriting this as a Hamiltonian probability density by taking the logarithm yields

$$\begin{aligned} H(\boldsymbol{d}|\boldsymbol{\lambda}) &= -\sum_k \left(-\lambda_k + d_k \log \lambda_k - \log d_k!\right) \\ &\propto \mathbf{1}^\dagger \boldsymbol{\lambda} - \boldsymbol{d}^\dagger \log \boldsymbol{\lambda}, \end{aligned}$$

where $\mathbf{1}$ is the unital vector and $\log \boldsymbol{\lambda}$ is meant element wise. Since $\boldsymbol{\lambda} = \langle \boldsymbol{d} \rangle$ we can now insert $\langle \boldsymbol{d} \rangle = \widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu}$ to obtain

$$\begin{aligned} H(\boldsymbol{d}|\boldsymbol{w}) \propto \mathbf{1}^\dagger &\left(\widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu}\right) \\ &- \boldsymbol{d}^\dagger \log \left(\widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu}\right). \end{aligned} \tag{5.3}$$

5.2.3 Diffuse prior

The next step in our derivation will be the prior for $\boldsymbol{w}$. We will assume that the field $e^{\boldsymbol{w}}$ will follow a log-normal distribution, which is the *maximum entropy probability distribution* for a strictly positive random variable with given mean and variance of its logarithm. Note that we can assume that $\boldsymbol{w}$ has zero mean, by appropriately choosing ρ_0, as discussed in Sec. 5.2.1, and although we do not know the covariance operator $\widehat{\boldsymbol{S}}$ of $\boldsymbol{w}$ yet, we will be able to infer it later. By definition of the log-normal distribution $\boldsymbol{w}$ is normally distributed. Hence the signal can be described by a Gaussian prior, i.e.

$$\begin{aligned} &p\left(\boldsymbol{w}|\widehat{\boldsymbol{S}}\right) = \mathcal{G}\left(\boldsymbol{w}, \widehat{\boldsymbol{S}}\right) \text{ with} \\ &\mathcal{G}\left(\boldsymbol{w}, \widehat{\boldsymbol{S}}\right) = \frac{1}{\det(2\pi\widehat{\boldsymbol{S}})} \exp\left(-\frac{1}{2}\boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w}\right). \end{aligned}$$

We now have to estimate the covariance operator $\widehat{\boldsymbol{S}}$. Since $\widehat{\boldsymbol{S}}$ is positive semi-definite, its always possible to diagonalize it by Mercer's theorem, although we do not know its eigenspaces. If the signal is *statistical homogeneous* and *isotropic*, which we will assume from now on, one can show that $\widehat{\boldsymbol{S}}$ is diagonal with respect to the harmonic basis. Because of positive semi-definiteness, the eigenvalues of $\widehat{\boldsymbol{S}}$ are positive. Thus we can write

$$\widehat{\boldsymbol{S}} = \sum_k e^{\tau_k} \widehat{\boldsymbol{S}}_k,$$

where $\widehat{\boldsymbol{S}}_k$ are projection operators onto linear subspaces U_k, called the *spectral bands* of the operator, and $\boldsymbol{\tau} = \{\tau_k\}$ are its *spectral parameters*. Here the spectral bands correspond to "eigenspaces" and the spectral parameters to the logarithm of the "eigenvalues" in finite dimensions. The designations come from operator theory and are unrelated to the energy spectrum of the 511 keV line under consideration. By writing the eigenvalues as e^{τ_k}, we again enforce positivity, if we infer them. Hence we get a prior depending on $\boldsymbol{\tau}$, i.e.

$$\widehat{\boldsymbol{H}}(\boldsymbol{w}|\boldsymbol{\tau}) = \log\det\left(2\pi\widehat{\boldsymbol{S}}\right) + {}^1\!/\!_2 \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w} \tag{5.4}$$

Have now reduced the inference of $\widehat{\boldsymbol{S}}$ to its power spectrum $\boldsymbol{\tau}$, which still has to be infered from the data. Hence we have to construct a prior for $\boldsymbol{\tau}$.

Magnitude

In a first step we will now neglect the correlation between the τ_k and concentrate on constructing a non informative prior, which does not assume anything about the order of magnitude. We will justify why the *inverse-Gamma distribution* achieves this goal and construct a prior for the τ_k from it.

Inverse-Gamma distribution: Assume we have a quantity v, whose order of magnitude is unknown. It therefore seems reasonable to assume, that its prior is uniform on a logarithmic scale[1] that is,

$$p\left(\log(v)\right) = 1.$$

Back transformation would give

$$p\left(\log(v)\right) d\log(v) = p\left(v\right) \frac{d\log(v)}{dv} dv = \frac{1}{v},$$

which cannot be normalized for $-\infty < v < +\infty$ and is therefore improper.

We can approximate this prior by an inverse-Gamma distribution $\mathcal{I}$, which can be written as

$$p(v|\alpha, q) = \mathcal{I}(v; \alpha, q) = \frac{1}{q \cdot \Gamma(\alpha)} \left(\frac{v}{q}\right)^{-\alpha} \exp\left(-\frac{q}{v}\right),$$

where Γ is the gamma function, α and q parameters controlling the mean and variance of $\mathcal{I}$. This distributions converges point wise to ${}^1\!/\!_v$ if we let $q \to 0$ and $\alpha \to 1$.

[1] This is equivalent to demand that our measure is invariant w.r.t multiplication, which would naturally lead to the Haar-Measure $\mu(S) = \int \frac{dv}{v}$.

We will later see that the singularity at $q = 0$, $\alpha = 1$ in the inverse-Gamma distribution will disappear during our derivation. Hence we can explicitly set $q = 0$ and $\alpha = 1$ in our final algorithm, if we want.

Applied on τ_k: We now assume that the τ_k are independent and distributed according to the inverse-Gamma distribution, i.e.

$$\begin{aligned} p(e^{\boldsymbol{\tau}}|\boldsymbol{\alpha}, \boldsymbol{q}) &= \prod_k \mathcal{I}(e^{\tau_k}; \alpha_k, q_k) \\ &= \prod_k \frac{1}{q_k \cdot \Gamma(\alpha_k)} \left(\frac{e^{\tau_k}}{q_k}\right)^{-\alpha_k} \exp\left(-q_k \cdot e^{-\tau_k}\right) \\ &\propto \prod_k \exp\left(-\alpha_k \tau_k - q_k \cdot e^{-\tau_k}\right), \end{aligned}$$

where we omitted all multiplicative terms in the last step not depending on $\boldsymbol{\tau}$, as announced in Sec. 5.1.3, since they will disappear in the MAP estimate. If we transform the density from e^{τ_k} to τ_k we get

$$\begin{aligned} p(\tau|\alpha, q) &= \prod_k p(\tau_k|\alpha, q) \left|\frac{de^{\tau_k}}{d\tau_k}\right| \\ &\propto \prod_k \exp\left(-\alpha_k \tau_k - q_k \cdot e^{-\tau_k} + \tau_k\right) \\ &\propto \exp\left(-(\boldsymbol{\alpha} - \mathbf{1})^\dagger \boldsymbol{\tau} - \boldsymbol{q} \cdot e^{-\boldsymbol{\tau}}\right). \end{aligned}$$

Smoothness

Our prior probability still does not incorporate any correlations between the τ_k. We now want to introduce a prior which favours a (eigen-)spectrum following a power law.

Such a prior is not unique, since we can choose any functional which takes on its maximum value for functions of the form $k \to k^\gamma$, which are called power functions.

One such choice is the *spectral smoothness prior* introduced in [Opp+13], which has the form

$$p(x|\sigma) \propto \exp\left(-\frac{1}{2} \int d(\log k) \frac{1}{\sigma_k^2} \left(\frac{\partial^2 \log x(k)}{\partial(\log k)^2}\right)^2\right). \tag{5.5}$$

This has its maximum for $x(k) = A \cdot k^\gamma$, where $A, \gamma \in \mathbb{R}$ are arbitrary constants. This property can be seen as follows: Consider the differential equation

$$\frac{\partial^2 \log x(k)}{\partial(\log k)^2} = 0, \tag{5.6}$$

which we solve by integrating twice over $\log k$

$$\log x(k) = \gamma \log k + \beta \qquad \gamma, \beta \in \mathbb{R}$$

and exponentiating to get

$$x(k) = \beta' \cdot k^\gamma \qquad \gamma \in \mathbb{R}, \quad \beta' = e^\beta,$$

which is its *unique* solution.

Therefore power functions fulfill Eq. 5.6 and consequently maximize Eq. 5.5. Now consider a function $\hat{x}$, which is not a power function but maximizes the functional Eq. 5.5. Obviously still Eq. 5.6 must hold, but this contradicts the uniqueness of the solutions, which are the power functions. Therefore, the power functions must be the only functions which maximize Eq. 5.5.

If we now plug in e^{τ_k} for $x(k)$ and introduce the operator $\widehat{\boldsymbol{T}}$ to make the notation more concise then we get the following prior

$$\begin{aligned} p(\boldsymbol{\tau}|\boldsymbol{\sigma}) &= \exp\left(-1/2 \cdot \boldsymbol{\tau}^\dagger \widehat{\boldsymbol{T}} \boldsymbol{\tau}\right) \\ \boldsymbol{\tau}^\dagger \widehat{\boldsymbol{T}} \boldsymbol{\tau} &= \int d(\log k) \frac{1}{\sigma_k^2} \left(\frac{\partial^2 \tau_k}{\partial(\log k)^2}\right)^2 \end{aligned}$$

Combining priors

We now combine both derived priors. If we assume that the parameters are independent we have to multiply the previously derived probabilities, i.e.

$$p(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{q}, \boldsymbol{\sigma}) = p(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{q}) \cdot p(\boldsymbol{\tau}|\boldsymbol{\sigma}),$$

or written down in terms of Hamiltonian functions

$$\begin{aligned} H(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{q}, \boldsymbol{\sigma}) &= H(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{q}) + H(\boldsymbol{\tau}|\boldsymbol{\sigma}) \\ &= -(\boldsymbol{\alpha} - \mathbf{1})^\dagger \boldsymbol{\tau} - \boldsymbol{q}^\dagger e^{-\boldsymbol{\tau}} - 1/2 \boldsymbol{\tau}^\dagger \widehat{\boldsymbol{T}} \boldsymbol{\tau}. \end{aligned} \tag{5.7}$$

5.2.4 Applying the MAP estimate

We now find a posterior probability distribution by using Bayes theorem and combining the poissonian likelihood with all the priors for the diffuse signals introduced in the previous sections.

$$p(\boldsymbol{w}, \boldsymbol{\tau}|\boldsymbol{d}) \propto p(\boldsymbol{d}|\boldsymbol{w}) \cdot p(\boldsymbol{w}|\boldsymbol{\tau}) \cdot p(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{q}, \boldsymbol{\sigma}).$$

Rewriting this in terms of Hamiltonians yields

$$H(\boldsymbol{w}, \boldsymbol{\tau}|\boldsymbol{d}) \propto H(\boldsymbol{d}|\boldsymbol{w}) + H(\boldsymbol{w}|\boldsymbol{\tau}) + H(\boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{q}, \boldsymbol{\sigma}).$$

If we plug in for the Hamiltonians Eqs. 5.3, 5.4 and 5.7 we get

$$\begin{aligned} H(\boldsymbol{w}, \boldsymbol{\tau}|\boldsymbol{d}) \propto \mathbf{1}^\dagger \left(\widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu}\right) - \boldsymbol{d}^\dagger \log\left(\widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu}\right) \\ + {}^1\!/_2 \log\det \widehat{\boldsymbol{S}} + {}^1\!/_2 \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w} \quad (5.8) \\ - (\boldsymbol{\alpha} - \mathbf{1})^\dagger \boldsymbol{\tau} - \boldsymbol{q}^\dagger e^{-\boldsymbol{\tau}} - {}^1\!/_2 (\boldsymbol{\tau})^\dagger \widehat{\boldsymbol{T}} \boldsymbol{\tau}, \end{aligned}$$

which is the final information Hamiltonian for our problem and contains all the results up to now.

For the maximum a posteriori principle we now have to differentiate for $\boldsymbol{w}$ and $\boldsymbol{\tau}$ and set it to zero. We will just state the results and refer to Appendix B for the intermediate calculation steps. Differentiation w.r.t $\boldsymbol{w}$ gives

$$\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{w}} = (\mathbf{1} - \boldsymbol{d}/\boldsymbol{l})^\dagger \widehat{\boldsymbol{R}} * e^{\boldsymbol{w}} + \left(\widehat{\boldsymbol{S}}^*\right)^{-1} \boldsymbol{w}, \quad (5.9)$$

with

$$\boldsymbol{l} = \widehat{\boldsymbol{R}} e^{\boldsymbol{w}} + \boldsymbol{\mu} \quad (5.10)$$

$$\widehat{\boldsymbol{S}}^* = \sum_k e^{\tau_k^*} \widehat{\boldsymbol{S}}_k, \quad (5.11)$$

where $*$ stands for element wise multiplication.

The expression $(\mathbf{1} - \boldsymbol{d}/\boldsymbol{l})^\dagger \widehat{\boldsymbol{R}}^{(s)} * e^{\boldsymbol{s}}$ has to be understood in the way, that we create a transposed vector $(\mathbf{1} - \boldsymbol{d}/\boldsymbol{l})^\dagger$, which we then multiply with $\widehat{\boldsymbol{R}}^{(s)}$ to get a vector. This resulting vector is now multiplied per component with $e^{\boldsymbol{s}}$.

Differentiation w.r.t. $\boldsymbol{\tau}$ yields

$$\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{\tau}} = -e^{-\boldsymbol{\tau}} \left(\boldsymbol{q} + {}^1\!/_2 \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w}\right) + \boldsymbol{\gamma} + \widehat{\boldsymbol{T}} \boldsymbol{\tau}, \quad (5.12)$$

with

$$\boldsymbol{\gamma} = (\boldsymbol{\alpha} - 1) + {}^1\!/_2 \boldsymbol{\rho} \quad (5.13)$$

where ρ_k measures the degrees of freedom of the range of $\widehat{\boldsymbol{S}}_k$. These are the dimension of the eigenspace in finite dimension. In infinite dimensions it becomes a path integral over the eigenfunctions.

Setting Eq. 5.12 to zero and solving for e^τ yields

$$e^{\boldsymbol{\tau}} = \frac{\boldsymbol{q} + \frac{1}{2}\left(\mathrm{tr}\left[\boldsymbol{s}\boldsymbol{s}^\dagger \widehat{\boldsymbol{S}}_k^{-1}\right]\right)_k}{\boldsymbol{\gamma} + \widehat{\boldsymbol{T}} \boldsymbol{\tau}} \quad (5.14)$$

For an error estimate we calculate the covariance matrix by evaluating the second derivative

$$\begin{aligned} \widehat{\boldsymbol{D}}_{xy}^{-1} =& \frac{\partial \widehat{\boldsymbol{H}}}{\partial w(x) w(y)} \\ =& \sum_i \left(1 - \frac{d_i}{l_i}\right) \widehat{\boldsymbol{R}}_{ix} e^{w(x)} \delta_{xy} + \left(\widehat{\boldsymbol{S}}_{xy}^*\right)^{-1} \\ &+ \sum_i \frac{d_i}{l_i^2} \left(\widehat{\boldsymbol{R}}_{iy} e^{w(y)}\right) \left(\widehat{\boldsymbol{R}}_{ix} e^{w(x)}\right). \quad (5.15) \end{aligned}$$

We want to stress that $\widehat{\boldsymbol{D}}$, the covariance of $e^{\boldsymbol{w}}$, is different from $\widehat{\boldsymbol{S}}$, the covariance of $\boldsymbol{w}$. In practice we are only interested in the diagonal entries $\widehat{\boldsymbol{D}}_{xx}$ of the matrix. This is still very costly, since calculating them involves inverting a large matrix.

5.2.5 The full algorithm

From a mathematical point of view, we are now finished. We only have to

1. find the root $(\boldsymbol{w}^*, \boldsymbol{\tau}^*)$ of $\left(\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{w}}, \frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{\tau}}\right)$,
2. calculate the uncertainty $\widehat{\boldsymbol{D}}_{xy}^{-1}$ for the fields $(\boldsymbol{w}^*, \boldsymbol{\tau}^*)$ and
3. check whether we have found the correct maximum, through our domain specific knowledge.

In practice, minimizing $\boldsymbol{w}$ and $\boldsymbol{\tau}$ at the same time is too costly and complicated to implement. It is more feasible to minimize $\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{w}}$ and $\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{\tau}}$ *separately.* This will certainly not give the correct result immediately, but if done iteratively, this scheme will converge. Hence we can stop if the root-mean-square deviation of the signals compared to the previous iteration is below a certain threshold, since we are then close to a fixed point, i.e. the minimum. Typically we set the threshold for this thesis to 10^{-7}.

For the minimization we use the *steepest-descent-method.* Its suitable because in every iteration, we just need to calculate the gradient of $\widehat{\boldsymbol{H}}$ and not the Hessian matrix as e.g. in Newton's method. Aside from the fact that building the Hessian is

Input Let $\widehat{\boldsymbol{R}}$ be our response operator and $\boldsymbol{d}$ the data.
Output Let $\boldsymbol{w}$ be the inferred signal and $\widehat{\boldsymbol{D}}_{xx}^{-1}$ its uncertainty.
Initialize Set $\boldsymbol{w}_0$ and $\boldsymbol{\tau}$ to 0.
Loop $i = 1...$maxiter:

1. minimize
$$\boldsymbol{w}_i \rightarrow \left.\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}_i, \boldsymbol{\tau}_{i-1}}$$
with the steepest-descent method for $\boldsymbol{w}_i$.
2. minimize
$$\boldsymbol{\tau}_i \rightarrow \left.\frac{\partial \widehat{\boldsymbol{H}}}{\partial \boldsymbol{\tau}}\right|_{\boldsymbol{w}_i, \boldsymbol{\tau}_i}$$
with the steepest-descent method for $\boldsymbol{\tau}_i$.
3. If $\|\boldsymbol{w}_i - \boldsymbol{w}_{i-1}\| < 10^{-7}$ stop.

Calculate the uncertainty by inverting $\widehat{\boldsymbol{D}}_{xx}$ by diagonal probing or directly inverting $\widehat{\boldsymbol{D}}^{-1}$.

Algorithm 4: Diffuse D^{3}PO-Algorithm.

very costly, it would be ill-conditioned for our response operator and hence the Newton iteration without preconditioning would most likely fail.

We will use the zero function as an *initial guess* of $\boldsymbol{w}$. This is reasonable, since $\langle \boldsymbol{w} \rangle = 0$, as we have discussed in Sec. 5.2.1, by appropriately setting ρ_0 in $\widehat{\boldsymbol{R}}$.

To quantify the *uncertainty* in our calculation we have to determine $\widehat{\boldsymbol{D}}_{xy}^{-1}$. In the original D^{3}PO-implementation, this is done by diagonal probing [SOE12], which is part of NIFTY, the "**N**umerical **I**nformation **F**ield **T**heor**y**" library. . We can re-use this implementation for our modification in Sec. 5.3.1, where we work with a fixed background, but this is not possible if we try to infer two diffuse signals - the background and the sky - in Sec. 5.3.2. Hence, as a workaround, we build up the whole matrix $\widehat{\boldsymbol{D}}$ and invert it.

We summarize the discussed algorithm in Alg. 4.

5.3 Modified D^{3}PO

In the previous section we derived the D^{3}PO-algorithm for a diffuse signal $\boldsymbol{w}$ and a constant offset $\boldsymbol{\mu}$. In this section we will perform the small modifications which are necessary for our problem with SPI data.

5.3.1 D^{3}PO with fixed background

The most basic algorithm would be to take the sky signal as our diffuse background $\boldsymbol{w} = \boldsymbol{s}$, and to take for the background a constant vector $\boldsymbol{\mu} = \boldsymbol{b}$, which was calculated by another algorithm [Sie+16]. Then we would get $\widehat{\boldsymbol{R}} = \widehat{\boldsymbol{R}}^{(s)}$ for the response operator and $\widehat{\boldsymbol{S}} = \widehat{\boldsymbol{S}}^{(s)}$ for the covariance matrix.

We can then directly apply Alg.4. We will discuss the results of this modification applied on test data in Sec. 6.1.3.

5.3.2 D^{3}PO with free background

A more sophisticated algorithm can be constructed by allowing both, the sky and background signal to be determined by the algorithm. In this case our diffuse signal $\boldsymbol{w}$ would be a combination of the sky $\boldsymbol{s}$ and background $\boldsymbol{b}$ signal

$$\boldsymbol{w} = \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{b} \end{pmatrix}$$

and thus be part of the product space of all possible skies with all background parameters. In this case we can set the constant offset to zero

$$\boldsymbol{\mu} = \boldsymbol{0}.$$

Our response operator would be

$$\widehat{\boldsymbol{R}} = \begin{pmatrix} \widehat{\boldsymbol{R}}^{(s)} & \widehat{\boldsymbol{R}}^{(b)} \end{pmatrix},$$

such that

$$\widehat{\boldsymbol{R}} e^{\boldsymbol{w}} = \widehat{\boldsymbol{R}}^{(s)} e^{\boldsymbol{s}} + \widehat{\boldsymbol{R}}^{(b)} e^{\boldsymbol{b}}.$$

If we assume that $\boldsymbol{s}$ and $\boldsymbol{b}$ are uncorrelated, we get the block diagonal correlation matrix

$$\widehat{\boldsymbol{S}} = \begin{pmatrix} \widehat{\boldsymbol{S}}^{(s)} & \\ & \widehat{\boldsymbol{S}}^{(b)} \end{pmatrix}.$$

The last assumption is an approximation and cannot rigorously be justified. It is certainly sensible to assume that the "sky signal" and the "background" are uncorrelated, since we already use this assumption, if we construct the background patterns. In the original implementation in [Gha17], the inferred background signal $\boldsymbol{b}$ was the intensity of the 13th detector, thus uncorrelatedness should apply there. Here $\boldsymbol{b}$ is the ratio of background present in the measured data. Thus if $\boldsymbol{s}$ increases, $\boldsymbol{b}$ has to decrease, hence we expect them to be *anti-correlated*. With our assumption above, we neglect these correlations.

Correct prior for b

Note that if we define $\boldsymbol{w}$ as above, we implicitly assume that both $\boldsymbol{\rho}^{(\boldsymbol{s})}$ and background $\boldsymbol{\rho}^{(\boldsymbol{b})}$ are "diffuse" signals following a log-normal distribution. This is clear for $\boldsymbol{\rho}^{(\boldsymbol{s})}$ but controversial for $\boldsymbol{\rho}^{(\boldsymbol{b})}$. In [Gha17] the background-patterns is normalized with respect to the 13th detector. Thus $\boldsymbol{\rho}^{(\boldsymbol{b})}$ can be interpreted as the background intensity for the 13th detector, which should not vary too much from pointing to pointing and therefore should be "diffuse". In contrast, already the results in [Gha17, p. 91] suggest that the inferred background is not very smooth. This will not change if our background operator modifies several pointings at once with one parameter. And since we refrained from normalizing the background, the prior here becomes even more questionable. Nevertheless we do not expect dramatic changes in the final results.

Algorithmic modifications

It would be possible to minimize $\boldsymbol{s}$ and $\boldsymbol{b}$ simultaneously, though tedious to implement, since the current version of the underlying NIFTY-package does not support the disjoint union of spaces. Therefore, we will do the same as we did in Sec. 5.2.5 with $\boldsymbol{w}$ and $\boldsymbol{\tau}$: We will iteratively minimize $\boldsymbol{s}$ and $\boldsymbol{b}$ until the mean-squared-deviation is smaller than the threshold of 10^{-7}.

Note that contrary to the correlation matrix $\widehat{\boldsymbol{S}}$, the uncertainty matrix $\widehat{\boldsymbol{D}}^{-1}$ is *not* block diagonal. This is in contrast to the original D^{3}PO-implementation, where we also determine two signals - one point-like and one diffuse - but the $\widehat{\boldsymbol{D}}^{-1}$-matrix still factors into two independent blocks. Hence we can not rely on the original D^{3}PO-implementation to determine the uncertainty with diagonal probing. Instead we will assemble $\widehat{\boldsymbol{D}}^{-1}$ and invert it manually. This will not always lead to usable uncertainty estimates, as we will discover in Sec. 6.1.2.

We will discuss the results of this modification applied to test data in Sec. 6.1.4.

Chapter 6

Application to multi-year data and the 511 keV line emission

6.1 Simulations

In this section we will apply the algorithm discussed in the previous chapters to SPI-like data. We will introduce a convenient test case from a pre-defined (artificial) sky in Sec. 6.1.1 and simulate the measured data, by applying the response operator and adding poissonian noise.

Then we can apply our algorithm on the simulated data, infer a sky and compare it to our test sky. If both are similar they should have a good cross-correlation factor and hence our algorithm has succeeded.

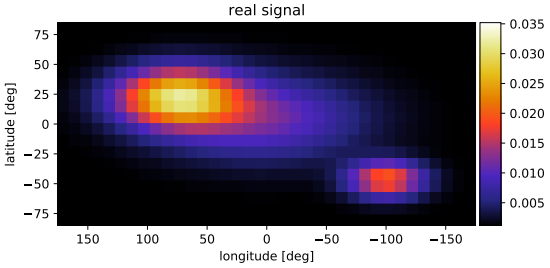
At first we will start with very simple problems and increase the difficulty gradually. In Sec. 6.1.3 we will start with a fixed background and just infer the sky signal. In Sec. 6.1.4 we also introduce degrees-of-freedom in the background.

6.1.1 Artificial test sky

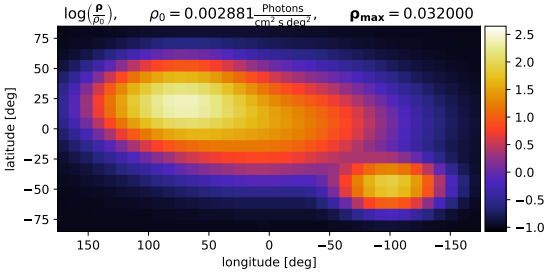
Our test sky is depicted in Figure 6.1a and consists of three Gaussian functions, whose properties are summarized in Tab. 6.1. They are superimposed such that

$$\text{“test sky”} = \frac{A_{\max}}{N} \left(0.4 \cdot G_0 + 1 \cdot G_1 + 0.7 \cdot G_2\right),$$

where $N = \max\left(0.4 \cdot G_0 + 1 \cdot G_1 + 0.7 \cdot G_2\right)$ is a normalization constant and $A_{\max}$ is the maximum value of our test sky. We will vary $A_{\max}$ to test the performance of our algorithm for different signal strengths.



(a) The simulated sky we use for testing.



(b) The logarithm of the simulated sky.

	μ_x [°]	μ_y [°]	σ_x [°]	σ_y [°]
G_0	0	0	60	51
G_1	72	17	36	17
G_2	-108	-50	23	11

Table 6.1: Properties for Gaussian functions G_0, G_1 and G_2.

$A_{\max}$	$\frac{\text{sky counts}}{\text{total counts}}$ [%]	$\frac{\text{sky counts}}{\text{noise}}$ [%]
0.032	3	78
0.0032	0.3	7.9
0.00128	0.12	3.1
0.00064	0.05	1.6

Table 6.2: Properties of our test sky for different signal strengths.

Signal strength To get meaningful results from our tests, the test signal should have the same order of magnitude, as the final image, which we want to infer. Hence we scale it to a range similar to the diffuse part of the end results of the D^{3}PO-reconstruction in [Gha17, Section 5.1, p. 74]. We will use $A_{\max} = 0.032\,\text{Ph/cm}^2\text{/s/ deg}^2$, as an optimistic guess for the signal strength and resort to $A_{\max} = 0.0032\,\text{Ph/cm}^2\text{/s/ deg}^2$ and $A_{\max} = 0.00128\,\text{Ph/cm}^2\text{/s/ deg}^2$ to study the behaviour for weaker signals.

We summarize how much the sky contributes to the total data and the ratio of the sky counts to the noise in Tab. 6.2. The ratio $\frac{\text{sky counts}}{\text{total counts}}$ indicates how much the sky contributes to the total data, while $\frac{\text{sky counts}}{\text{noise}}$ quantifies the difficulty in the separation of the sky signal and noise.

Signal shape The position of the Gaussian functions were chosen, such that the test image differs significantly from the exposure map. Hence we cannot accidentally fit the data to the exposure map. This can happen, if the background signal is inferred as too weak and thus the data is explained mainly from the sky signal. Note that G_2 is in an area of low exposure and hence an indicator, how our algorithm performs there.

Logarithmic view In a lot of sky maps, we see strong point sources and weak diffuse signals, which makes it hard to discern changes in the diffuse part. It is thus useful to view them on a logarithmic scale. If we normalize them by their mean first, as we already do in the D^{3}PO-algorithm, we can even compare the quality of signals with different signal strengths. Hence we will often depict $\boldsymbol{s}$, instead of the real signal $\boldsymbol{\rho_s}$. A logarithmic view of our test signal is shown in Fig. 6.1b for reference.

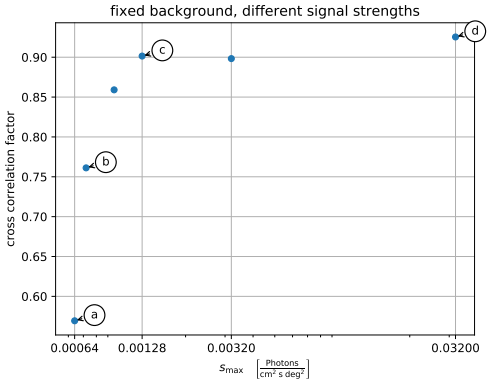


Figure 6.2: Plot of the cross-correlation, if we fix the background, just determine the sky but vary the signal strength. The final inferred images for the marked points are in Fig. 6.3.

6.1.2 Image comparison

We use the *normalized cross-correlation* for quantifying, how "similar" an inferred image is compared to the original one. This measure ranks the similarity with a number between -1 and 1. If the images are identical, the cross-correlation is 1.

Alternatively we could calculate χ^2 by

$$\chi^2 = \frac{1}{\#\text{pixels}} \sum_{\text{pixels}} \left(\frac{\text{original} - \text{inferred}}{\text{uncertainty}} \right)^2,$$

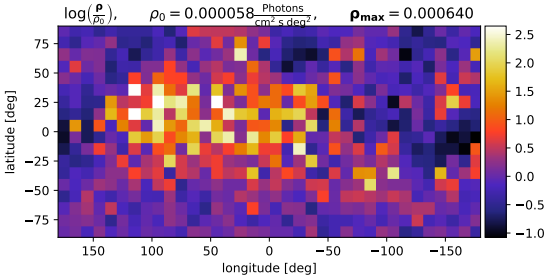
which should have a value near 1. This approach was not used, because the uncertainties were questionable.

6.1.3 Fixed background

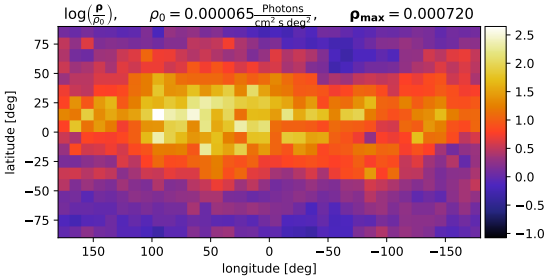
For a first test we want to study the inference of the sky if we already know the background, and keep it constant. This allows us to study, how different signal strengths affect the outcome.

In Fig. 6.2 we have plotted the cross-correlation for signals ranging from $0.00064\,\text{Ph/cm}^2\text{/s/ deg}^2$ to $0.032\,\text{Ph/cm}^2\text{/s/ deg}^2$. We note that the quality of our reconstruction decreases quickly below $0.0128\,\text{Ph/cm}^2\text{/s/ deg}^2$. It is likely that at this point the signal is so weak, that it cannot be distinguished from the noise.

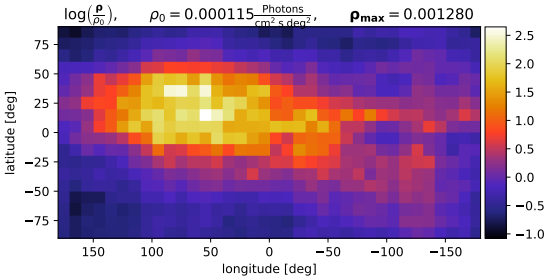
To get an intuition of the different cross-correlation values, we have plotted in Fig. 6.3 the



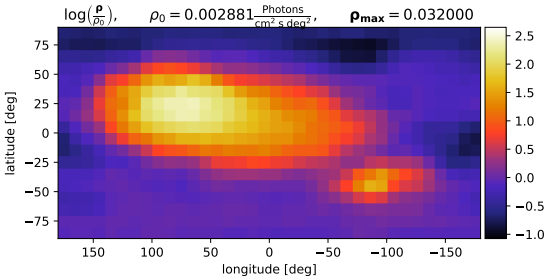
(a) $\boldsymbol{\rho}_{\max} = 0.00064 \text{Ph/cm}^2\text{/s/ deg}^2$



(b) $\boldsymbol{\rho}_{\max} = 0.00072 \text{Ph/cm}^2\text{/s/ deg}^2$



(c) $\boldsymbol{\rho}_{\max} = 0.00128 \text{Ph/cm}^2\text{/s/ deg}^2$



(d) $\boldsymbol{\rho}_{\max} = 0.032 \text{Ph/cm}^2\text{/s/ deg}^2$

Figure 6.3: Inferred sky maps for different signal strengths. The sky maps correspond to the marked points in Fig. 6.2.

sky maps for the marked points of Fig. 6.2. The inferred signal loses its clear shape and becomes more noisy, with decreasing signal strength. The small Gaussian function G_2, is only visible for our strong signal (Fig. 6.3d) and almost disappears in Fig. 6.3c, 6.3b and 6.3a, due to its low exposure time. In all Figures we can discern the rough shape of the superposition of G_0 and G_1. In Fig. 6.3b and 6.3a we can see additional artifacts appearing all over the image, due to fitting noise into our image.

6.1.4 Varying background

In this section, we allow the algorithm to change the amplitude of our background patterns by constructing the background response operator. This significantly increases the degrees-of-freedom and complicates the reconstruction process.

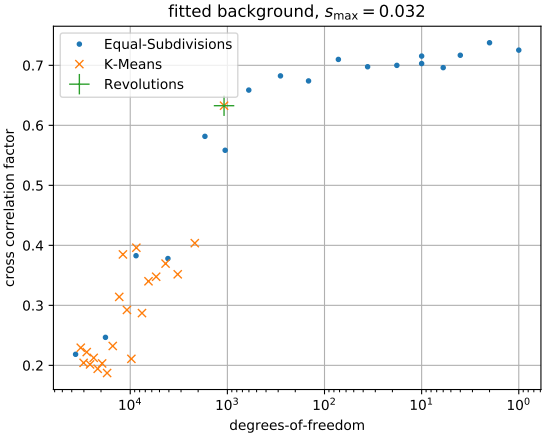
As in the simulation before we add the test sky to the calculated SPI background and add noise, to obtain our test data. When we constructed the background patterns in Sec. 3.4.3, we did this by explaining all the measured data with the help of the background patterns. Hence we expect that the background patterns will have to be scaled down most of the time. To simulate this effect, we fit the SPI background for every pointing to the test data, with a least squares approach, and take this as our new background pattern.

The results of the inference process are depicted in Fig. 6.4, for different signal strengths. We choose signals, which had a cross-correlation greater than 0.9 in Fig. 6.2.

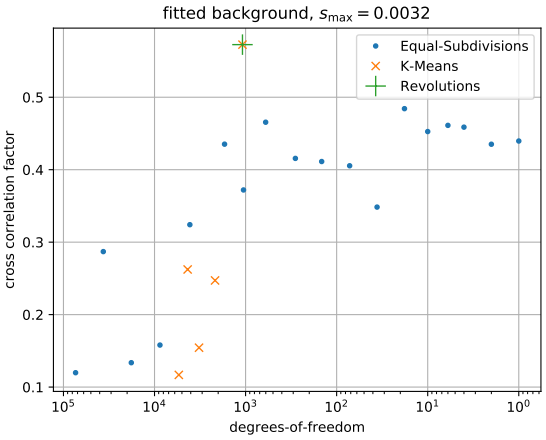
We have plotted the degrees-of-freedom against the cross-correlation factor, which allows us to directly compare the effect of different subdivision strategies. Since applying the K-Mean algorithm only make sense inside a revolution we only use it for a large number of degrees of freedom.

We note that adding a single degree-of-freedom to the background decreases our cross-correlation by 20% for our strong signal and by 50% and 65% for the weaker signals. This is not unexpected, since small changes to the background amplitude significantly changes the data counts used to infer the sky signal and hence its shape.

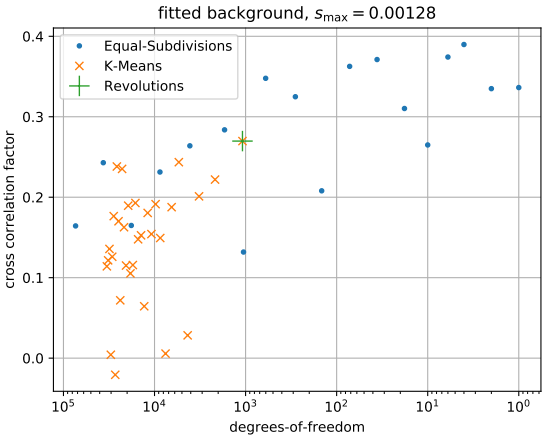
In all of the plots we can observe, that the cross-correlation drops if we add too many degrees-of-freedom. Then we start to explain the data counts due to the sky with only the background. This



(a)



(b)



(c)

Figure 6.4: Inferred sky maps for different signal strengths. The sky maps correspond to the marked points in Fig. 6.2.

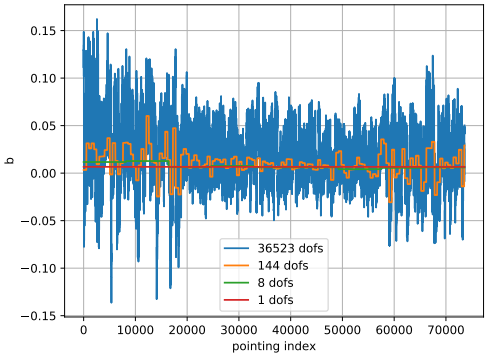


Figure 6.5: The values of $\boldsymbol{b}$ for the strong signal.

claim is supported by Fig. 6.5, where we have plotted the value of $\boldsymbol{b}$ for different degrees-of-freedom. There the amplitudes of the scaling factors increases with the degrees-of-freedom.

We can see that subdividing the background into more chunks does not change much, until we start subdividing the revolutions. Then the quality of our results drops at once for all signal strengths. This is in accordance to Fig. 6.5, where we can see that the amplitudes for a small number of degrees-of-freedom are very similar.

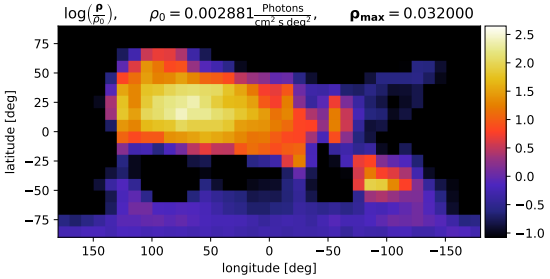
For every signal strength the equal subdivision and K-Mean algorithms perform similarly, hence adding spatial information does not seem to improve the quality of our subdivisions. The temporal order seems to be dominant effect on the image quality.

We have plotted the inferred sky map with the best cross-correlation factor for every signal strength in Fig. 6.6.

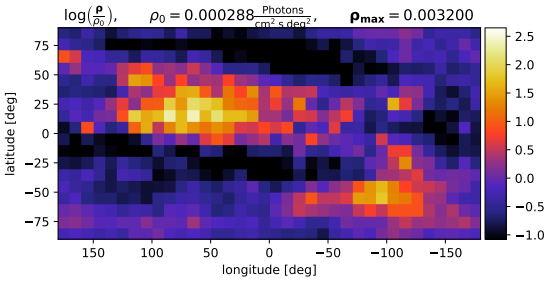
The shape of the strongest signal in Fig. 6.6a is clearly visible, though areas with a small signal strength are underestimated. In Fig. 6.6b areas with a signal are still recognizable, but the clear shape has dissolved. For the weakest signal in Fig. 6.6b, the large number of artifacts makes it difficult to identify the real signal.

6.2 Real data

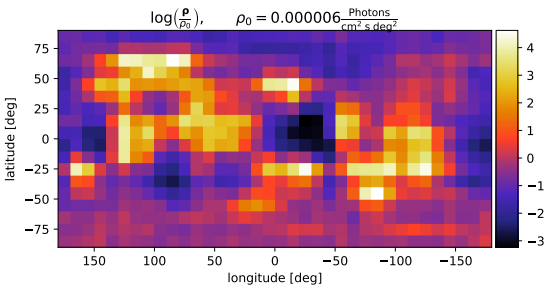
In this section we apply the algorithm to the measured SPI data. To estimate ρ_0, we the take the image of the D^{3}PO-reconstruction for both back-



(a) $A_{\max} = 0.032 \frac{\text{Photons}}{\text{cm}^2\,\text{s}\,\text{deg}^2}$



(b) $A_{\max} = 0.0032 \frac{\text{Photons}}{\text{cm}^2\,\text{s}\,\text{deg}^2}$



(c) $A_{\max} = 0.00128 \frac{\text{Photons}}{\text{cm}^2\,\text{s}\,\text{deg}^2}$

Figure 6.6: The inferred logarithmic sky maps with the best cross-correlation factor.

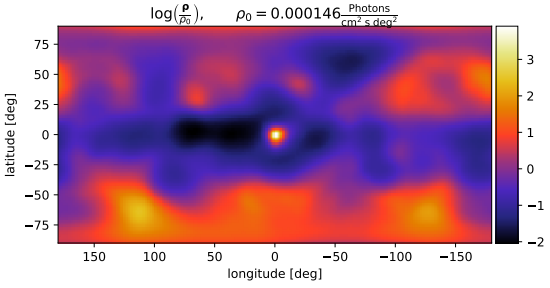


Figure 6.7: The end result of [Gha17, p. 87], with the colour scheme used in this thesis.

ground and sky from [Gha17, p. 87], depicted in Fig. 6.7 for comparison. We transform it to our pixel size and calculate the exact ρ_0 for that approximation.

We start in Sec. 6.2.1, with the coarse $10° \times 10°$-resolution, which allows a direct comparison with our simulation results. Later, we will switch in Sec. 6.2.2, to a fine $2° \times 2°$-resolution also used in [Gha17].

6.2.1 Coarse resolution imaging

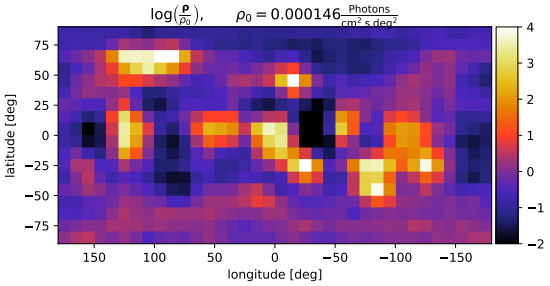
We have plotted the results for different subdivision schemes in Fig. 6.8. In Fig. 6.8a we only used one parameter for the background, in Fig. 6.8b we subdivided per revolution and in Fig. 6.8c we used one parameter per pointing, as in [Gha17].

Qualitatively the algorithm infers a bright spot in the galactic centre, which is in accordance with previous observations. The diffuse components, occurring in Fig. 6.7 at the top and bottom of the sky map, are no longer visible, probably because they are too weak for our coarse resolution.

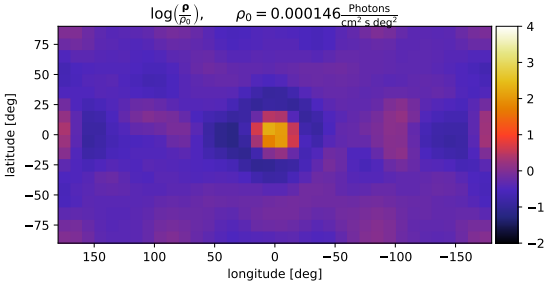
In Fig. 6.8a - in addition to the point like signal - artifacts appear. These have a similar shape as the artifacts in our simulation of the weakest signal in Fig. 6.6c. Hence they seem to be independent of the sky signal, which suggests, that in both cases the sky signal tries to explain background data.

Very prominent are the exponential-negative, black coloured areas around the central signal, which were already present in the original reconstruction (Fig. 6.7). These are reconstruction flaws, which appear if a point like signal is inferred by a diffuse signal, and are well documented in the D^{3}PO-documentation [Sel], as depicted in Fig. 6.9 for the D^{3}PO-demo signal.

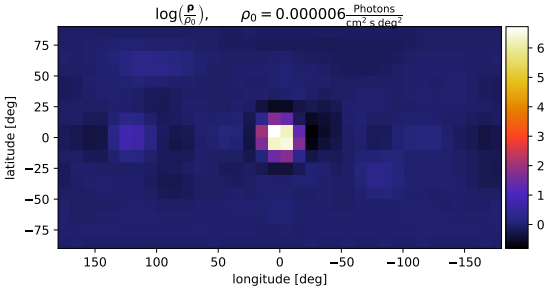
Quantitatively, we can compare the bright spot in the centre of the sky, with the results obtained by Mahsa Ghaempanah, by integrating the respective areas. Due to the coarse resolution of our images, the bright spot occupies an area of $20° \times 20°$ and thus the results are inaccurate. For the one-parameter case (Fig. 6.8a) we get eight times the flux of 6.7. For one parameter per revolution (Fig. 6.8b), we get twice the flux and for one parameter per pointing we roughly get the same flux as inferred before.



(a) One parameter for all pointings.



(b) Per Revolution subdivision.



(c) One parameter per pointing.

Figure 6.8: The inferred logarithmic sky maps for the real signal data.

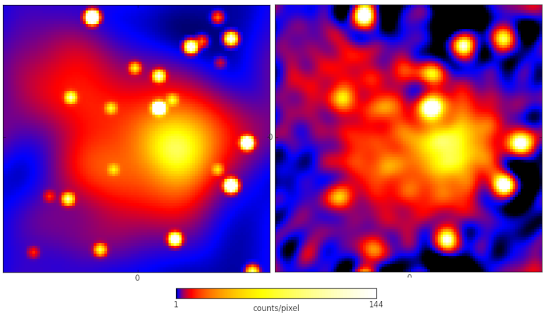


Figure 6.9: The reconstruction artifacts of the D^{3}PO-demo, due to using only a diffuse signal. The signal on the left is explained with a point like and a diffuse signal, the one on the right just with a diffuse signal. Picture take from [Sel].

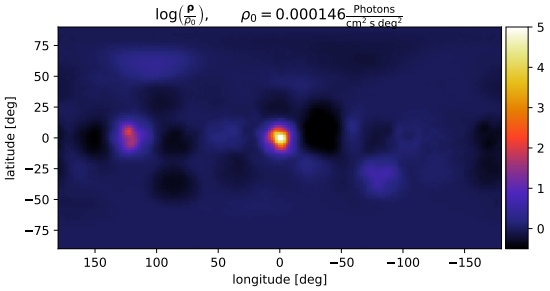
6.2.2 Fine resolution imaging

In Fig. 6.10 we use a finer discretization of 2° × 2° sized pixels in our response operator and fix the other parameters.

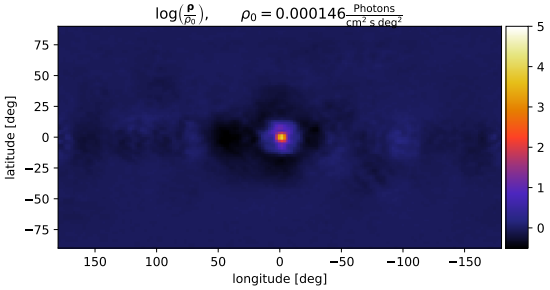
The results are similar to the ones we obtained for the coarser resolution. If we compare the coarse and fine signal for one background parameter (Fig. 6.8a and Fig. 6.10a), we notice the same dark and bright artifacts in both pictures, although the bright artifacts seem attenuated for the finer resolution. As we noticed before the signal gets weaker, if we introduce more degrees-of-freedom for the background. To quantify this effect, we calculate the the flux in the central region and plot it against the number of background parameters in Fig. 6.11. The original value from [Gha17] is indicated by a horizontal red line, the number of revolutions by vertical green line. We have also included the values from the coarser grid, to allow a direct comparison.

For the signal on the fine grid we get in general less flux than for the one on the coarser grid. We get the same flux as [Gha17], if we use one degree-of-freedom per revolution, though this is likely a coincidence.

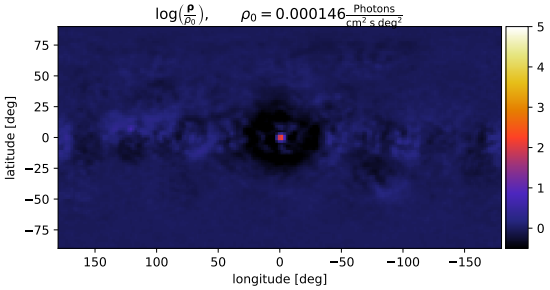
For the fine resolution we can directly compare the inferred signals with the reference signal of Fig. 6.7, by calculating the cross-correlation factor, which is depicted in Fig. 6.12. Since we lack the diffuse signals at the top and bottom of the sky map, the results are mediocre at best. We get the best coincidence if we partition the background per



(a) One parameter for all pointings.



(b) Per Revolution subdivision.



(c) One parameter per pointing.

Figure 6.10: The inferred logarithmic sky maps for the real signal data for a finer resolution.

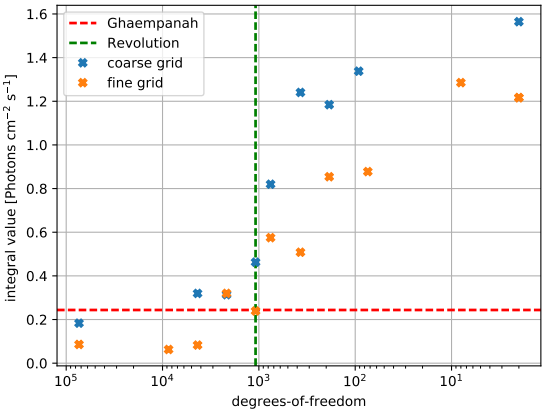


Figure 6.11: Integral over a $20° \times 20$ degree area. The result from Ghaempanah [Gha17] is marked with a red line. The point where we have more degrees-of-freedom than revolutions is marked by a green line.

revolution.

Background signal behaviour. Finally we will discuss the background characteristics. We have plotted the range of the values for $\boldsymbol{b}$ in Fig. 6.13. If we introduce more degrees-of-freedom, also the value range increases. As expected the mean is roughly zero.

For a better understanding how the background behaves on average, we show the mean value of $\boldsymbol{b}$ on a smaller value scale in Fig. 6.14. The mean of $\boldsymbol{b}$ is always negative, and thus $\langle e^{\boldsymbol{b}} \rangle$ is smaller than one. We expected this, since we constructed our background, by explaining all the data with it. Hence our algorithm would not decrease the background amplitude on average, no sky signal would fit into our data.

The background amplitude for many degrees-of-freedom is much larger, than if we only have a few. If we compare Fig. 6.11 with Fig. 6.14 we see that the mean of the background decreases as the flux of the signal increases. This strengthens our assumption, that the artifacts in Fig. 6.6c, Fig. 6.8a and Fig. 6.10a are due to background data explained by the sky signal.

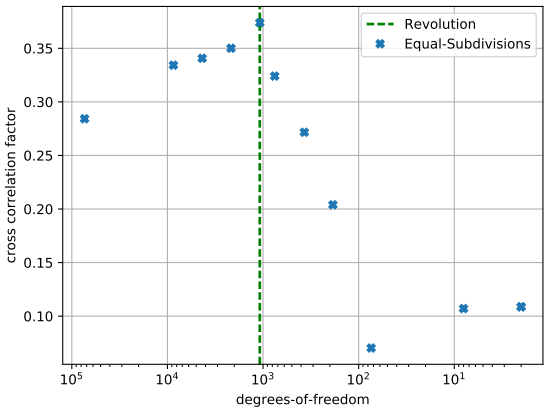


Figure 6.12: The cross-correlation factor with respect to the result from Ghaempanah [Gha17].

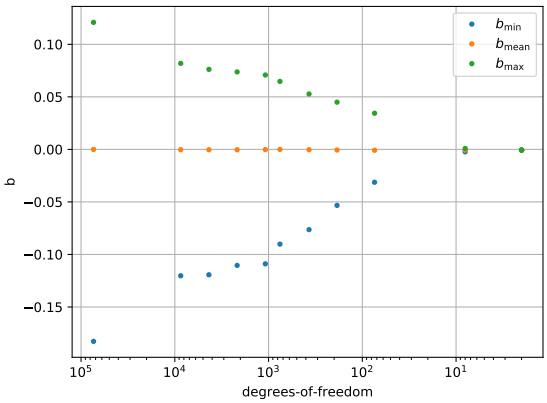


Figure 6.13: The range of $\boldsymbol{b}$ for different degree-of-freedom.

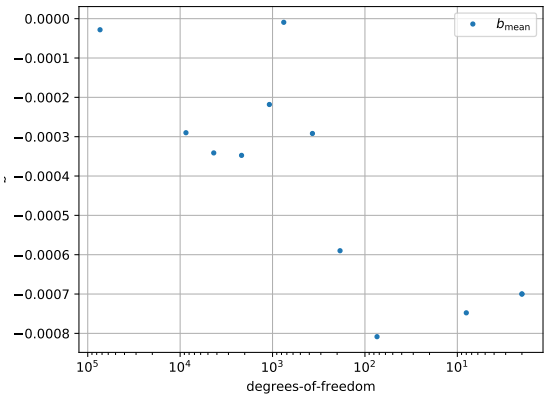


Figure 6.14: Mean value of $\boldsymbol{b}$ for different degree-of-freedom.

Chapter 7

Conclusion

7.1 Summary

The main focus of this thesis was a sensitivity study of gamma-ray imaging for the 511 keV line and its dependence on background treatment. For this, we constructed different *background response operators* for the SPI dataset and studied their effects on the D^{3}PO-algorithm.

If we relied only on the temporal information of the pointings, the Equal-Subdivisions algorithm seemed most appropriate. To includ spatial information we devised the K-Means algorithm. We expected that the K-Means algorithm should outperform the Equal-Subdivisions algorithm, if we use a fine subdivision of the pointings, because spatial information should then become relevant.

After introducing the modified D^{3}PO-algorithm, we started to test our algorithm. We noticed, that the K-Means algorithm did not provide any advantage compared to the Equal-Subdivisions algorithm. This indicates, that the temporal succession is dominant and that the spatial variations can mainly be neglected.

If we introduced too many degrees-of-freedom in the background operator, the results got poorer. This started to happen, if we exceeded the number of revolutions, hence we conclude, that this might be the right point to stop our refinement of pointings. Since the background at other energies differs significantly this will only be valid in the energy range around 511 keV.

When we tested weak sky-signals we observed artifacts , because the algorithm underestimated the background.

Finally, we applied the algorithm on the SPI dataset. We were able to reconstruct the point like signal in the centre, but failed in reconstructing the expected diffuse components around it. This is probably due to the signal being too weak and because of reconstruction artefacts introduced by the point like signals.

The reconstructed central signal seems reasonable for a large number of degrees-of-freedom in the background, though the intensity of the flux is larger than expected. If we have very few, the algorithm underestimates the background signal and artifacts start to appear and become stronger.

There is a general consensus that there should be a diffuse signal throughout the disk of the Milky Way [Sie+16], which we were not able to reconstruct.

7.2 Outlook

There are many major and minor modifications we could apply to our algorithm and which would probably improve the results. We will introduce a few here, which we find especially interesting.

Artifacts around central source. To avoid the artifacts from fitting a point like signal into a diffuse signal, the D^{3}PO-documentation suggests applying a mask to the image, which filters out the point like areas. Then we would have a chance to infer only the diffuse signal around the point source. For this approach to work we would have to estimate the point like signal beforehand.

Another approach might be to reintroduce the point like signal, which plays a mayor role in the D^{3}PO-algorithm, but was removed in [Gha17], due to problems with convergence. Since we already know, that the point like source appears in the galactic centre, we could try to circumvent this

problem, by restricting the point like signal to a tiny area in the centre. That means, the domain of the diffuse signal would still range from $-180°$ to $180°$, while the point like signal could only be considered from $-5°$ to $5°$.

One background signal per epoch. Currently we model the background amplitudes as one large one-dimensional signal ranging over all the pointings. The correlation between amplitudes of different pointings is described in the frequency domain by the power spectrum. Hence we assume, that the amplitude of pointings belonging to different epochs are somehow correlated. This assumption might be wrong. It would be more sensible to divide the large signal into several signals, with one signal per epoch. Each of these signals could have a log-normal prior for which we infer the covariance through the power spectrum, but these would be independent of the other signals.

This would be a manageable extension to our algorithm. Currently it infers two diffuse signals: the sky flux and the background amplitudes. Nothing prevents us from generalizing the algorithm by adding more diffuse signals.

Background averaging during the image. Currently the background patterns originate from data, which explained the measured photon counts only with the background. We did not apply a normalization, because it did not seems sensible to destroy the inferred amplitudes.

In the derivation of the D^{3}PO-algorithm we assume, that the logarithmic signals have mean zero, by a suitable choice of ρ_0. This condition is violated for our non normalized patterns, since we know that the mean must be slightly smaller than zero, otherwise we would not have measured any data from the sky.

Hence it would be nice to transform the background signal to more physical quantity, for which we have a good estimate of its mean.

One possible idea would be to take the normalization in [Gha17], but omit the conversion into rates through the detector-lifetimes. Hence we would normalize the chunks, such that the means of the 13th detector for one chunk are zero. We would then infer the mean value of the 13th detector for the background. We could estimate the mean value from the given data and previous approximations of the sky signal.

Gibbs free energy. One variation of the D^{3}PO-algorithm uses the Gibbs free energy to infer the signals, instead of the maximum a posterior approach. This is computational more expensive and hence was not used originally.

We did experience a huge performance gain by rewriting our response operators in cython[Beh+11] and applying **O**pen**MP** (OMP) for a simple shared memory parallelization (Appendix A). Extending this to a full distributed memory parallelization with the **m**essage **p**assing **i**nterface (MPI) would certainly be doable. The thus gained performance could make the Gibbs-approach feasible again. The main difficulty would be, to reimplement the diagonal sampling method in NIFTY, which uses its own process based parallelization, which would be incompatible. Since forked processes are already incompatible with some OMP implementations [Yli], this might still be the right way to proceed.

Soft background partitions. In this thesis we always assumed, that the amplitude for the background at one pointing is controlled by exactly one free parameter. This is certainly the simplest approach, but there is no reason not to use several parameters.

For instance, instead of K-Means clustering we could use a soft clustering algorithm. Such an algorithm assigns a pointing to several clusters at once. Every pointing belongs with a certain probability to a cluster. These probabilities are called weights. If every cluster has a background amplitude, we could calculate the weighted amplitude for each pointing and use it to scale the background pattern.

Another approach would be to use different ansatz functions for the background. Currently the mapping of a pointing to its background amplitude has the form of a discontinuous step function, since the background amplitude is constant inside a chunk. There is no reason why we could not use at least piecewise linear functions, or even more exotic shape-functions.

Appendix A

Numerical performance optimizations

In this chapter we will give a short overview, how we improved the performance of the algorithm, compared to its original version [Gha17]. This is a great example, how minor modifications can have a great impact on the overall performance and thus facilitate scientific work.

Since the original algorithm only worked on images with a $2^\circ \times 2^\circ$ resolution, all comparisons will be done with respect to this resolution.

A.1 Response operators

The main bottle neck of our algorithm is the evaluation of our sky response operator. This is the case since we maximize our posterior probabilities with a steepest descent method. Such an algorithm multiplies a vector with the operator and its adjoint in every iteration step at least once, sometimes more often, if the Wolfe-Conditions cannot be satisfied right away. This is also true for the background response operator, but since it can be easily formulated as the component wise multiplication of two vectors, it is easy to optimize. In the original implementation the sky response operator needs $\approx$ 40-times longer than the background response operator.

Our response operators are stored in a sparse format. This is necessary, since the a dense matrix of the sky response operator would need approximately 90 GB of main memory in floating point representation.

Sky response operator The original implementation was written in pure python, hence the main challenge there was to use as many matrix-vector-multiplications as possible, since these can be evaluated efficiently by the numpy package. On the other hand the matrices should not be too big, to avoid an explosion of memory. It is very difficult to find a satisfying compromise between these mutually exclusive goals.

We tried a simpler approach, by rewriting the multiplication in cython[Beh+11], starting out from an assembly algorithm for dense matrices by Xiao-Ling Zhang. We reformulated it to a sparse multiplication algorithm, translated it to cython, applied OpenMP for the parallelization, interchanged loops where appropriate to improve the cache performance and applied reductions where necessary to overcome loop dependencies. We refrained from using advanced blocking techniques to improve the cache performance even further, due to time constraints. The performance increase is plotted in Fig. A.1.

Simply by switching to cython, we get a speed-up of two, for both the the normal multiplication and the adjoint multiplication. As expected scales the speed-up linear with the number of processors. The slopes are not optimal, but still good for strong-scaling.

Background response operator We can apply the same optimizations to the background response operator. The only difficulty is, that we have to take into account the partitioning of our background into chunks, to avoid loop dependen-

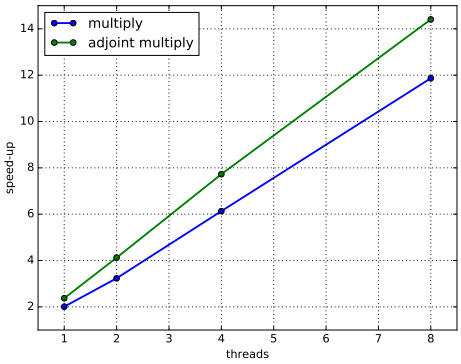


Figure A.1: Speed-up of the sky response operator, compared to the original implementation in [Gha17].

cies. This requires some preprocessing, but since this only has to be done once for a given partitioning, it does not impact the overall performance of our operator. The results are depicted in Fig. A.2.

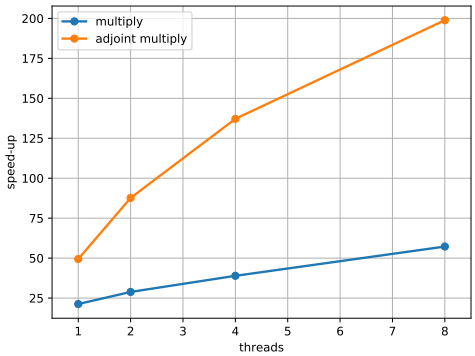


Figure A.2: Speed-up of the background response operator, compared to the original implementation in [Gha17].

Using cython improves the performance by a factor of 20 for the multiplication operation and even 50 for the adjoint multiplication. The decreasing slope of our speed-up might be due to the overhead from calling cython code from python, as predicted by Amdahl's law.

A.2 Memory

Main memory is seldom an issue in programming, but since we are working here with large datasets and huge matrices, we should still try to decrease our memory footprint.

The original implementation needed 54 GB of main memory. By simply freeing intermediate results and eliminating redundant data we were able to decrease it to 6.4 GB.

This allows us to run the algorithm on arbitrary machines and not only on special hardware with a lot of main memory.

A.3 Conclusion

All in all scripting languages like python can only take us up to a certain point in scientific computing. If we work with large data sets, and use operators, which cannot be easily expressed by conventional matrices, more efficient languages become obligatory, for optimizing the hotspots.

Appendix B

Derivatives

In this appendix we give the steps to calculate the derivatives for $\boldsymbol{w}$ and $\boldsymbol{\tau}$ in Eq. 5.9 and Eq. 5.12 from the Hamiltonian of Eq. 5.8.

B.1 Derivatives for $\boldsymbol{w}$

To calculate $\partial\widehat{\boldsymbol{H}}/\partial w(x)$ only the integrals have to be written out and differentiated. We calculate

$$\frac{\partial}{\partial w(x)}\left(\mathbf{1}^\dagger(\widehat{\boldsymbol{R}}e^{\boldsymbol{w}}+\boldsymbol{\mu})\right)=\sum_y \widehat{\boldsymbol{R}}_{yx}e^{s(x)} \tag{B.1}$$

and

$$\begin{aligned}\frac{\partial}{\partial w(x)}&\left(\boldsymbol{d}^\dagger\log\left(\widehat{\boldsymbol{R}}e^{\boldsymbol{w}}+\boldsymbol{\mu}\right)\right)\\ &=\sum_y d_y\frac{\widehat{\boldsymbol{R}}_{yx}e^{w(x)}}{\sum_z\widehat{\boldsymbol{R}}_{yz}e^{w(z)}+\mu_y}\\ &=\sum_y\frac{d_y}{l_y}\widehat{\boldsymbol{R}}_{yx}e^{w(x)},\end{aligned} \tag{B.2}$$

with

$$l_y=\sum_z\widehat{\boldsymbol{R}}_{yz}e^{w(z)}+\mu_y$$

Finally

$$\frac{\partial}{\partial w(x)}\left(\tfrac{1}{2}\boldsymbol{w}^\dagger\widehat{\boldsymbol{S}}^{-1}\boldsymbol{w}\right)=\sum_y\left(\widehat{\boldsymbol{S}}^{-1}\right)^*_{xy}w_y \tag{B.3}$$

holds since $\widehat{\boldsymbol{S}}^{-1}$ is symmetric but not hermitian.

Since all other terms of Eq. 5.8 do not depend on $\boldsymbol{w}$ adding up Eqs. B.1, B.2 and B.3 gives the full derivative of Eq. 5.9.

B.2 Derivatives for $\boldsymbol{\tau}$

Calculating $\partial\widehat{\boldsymbol{H}}/\partial w(x)$ needs more work. We calculate

$$\frac{\partial}{\partial\tau(k)}\left((\boldsymbol{\alpha}-\mathbf{1})^\dagger\boldsymbol{\tau}\right)=\alpha_k-1, \tag{B.4}$$

$$\frac{\partial}{\partial\tau(k)}\left(\boldsymbol{q}^\dagger e^{-\boldsymbol{\tau}}\right)=q_k e^{-\tau_k} \tag{B.5}$$

and

$$\frac{\partial}{\partial\tau(k)}\left(\tfrac{1}{2}\boldsymbol{\tau}^\dagger\widehat{\boldsymbol{T}}\boldsymbol{\tau}\right)=\sum_l\widehat{\boldsymbol{T}}_{kl}\tau_l^*, \tag{B.6}$$

by assuming that $\boldsymbol{\alpha}$, $\boldsymbol{q}$ are real valued vectors and $\widehat{\boldsymbol{T}}$ is a real valued symmetric operator. We are not finished yet, since $\widehat{\boldsymbol{S}}^{-1}$ depends by definition on $\boldsymbol{\tau}$.

For differentiating $\log\det\widehat{\boldsymbol{S}}$ recall that

$$\log\det\widehat{\boldsymbol{S}}=\operatorname{tr}\log\widehat{\boldsymbol{S}}$$

with the logarithm defined by the series

$$\log\widehat{\boldsymbol{S}}=\sum\frac{1}{n}\widehat{\boldsymbol{S}}^n.$$

$\widehat{\boldsymbol{S}}$ can be written as the sum of projection operators $\widehat{\boldsymbol{S}}_k$, which are idempotent and have a disjoint range, i.e.

$$\widehat{\boldsymbol{S}}_k\widehat{\boldsymbol{S}}_l=\delta_{kl}\widehat{\boldsymbol{S}}_k.$$

Hence

$$\widehat{\boldsymbol{S}}^n=\left(\sum_k e^{\tau_k}\widehat{\boldsymbol{S}}_k\right)=\sum_k e^{n\tau_k}\widehat{\boldsymbol{S}}_k,$$

and further

$$\begin{aligned}\log \widehat{\boldsymbol{S}} &= \sum_n \frac{1}{n}(\widehat{\boldsymbol{S}})^n \\ &= \sum_{n\,k} \frac{1}{n} e^{n\tau_k} \widehat{\boldsymbol{S}}_k \\ &= \sum_k \log e^{\tau_k} \widehat{\boldsymbol{S}}_k \\ &= \sum_k \tau_k \widehat{\boldsymbol{S}}_k.\end{aligned}$$

Since the trace is linear we get

$$\mathrm{tr} \log \widehat{\boldsymbol{S}} = \sum_k \tau_k \mathrm{tr} \widehat{\boldsymbol{S}}_k.$$

The projections are idempotent and therefore act as an identity on their respective range. Since in finite dimensional spaces, the trace of a projection operator corresponds to the dimension of the projected space, which is a direct consequence of the trace being the sum of eigenvalues. For infinite dimensions we would need the path integral

$$\mathrm{tr} \widehat{\boldsymbol{S}}_m = \int D\boldsymbol{k}\, \boldsymbol{k}^\dagger \widehat{\boldsymbol{S}}_m \boldsymbol{k},$$

where $\boldsymbol{k}$ are the eigenfunctions of $\widehat{\boldsymbol{S}}$.

Differentiating with respect to $\tau(k)$ we get

$$\frac{\partial}{\partial \tau(k)} \left(\mathrm{tr} \log \widehat{\boldsymbol{S}} \right) = \mathrm{tr} \widehat{\boldsymbol{S}}_k = \rho_k, \tag{B.7}$$

where ρ_k denotes the number of degrees-of-freedom. The original D^{3}PO-Paper uses the notation $\rho_k = \mathrm{tr} \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{S}}_k^{-1}$, which we find confusing, since $\widehat{\boldsymbol{S}}_k^{-1}$ is not surjective.

Finally we will differentiate $1/2 \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w}$. Note that

$$\widehat{\boldsymbol{S}}^{-1} = \left(\sum_k e^{\tau_k} \widehat{\boldsymbol{S}}_k \right)^{-1} = \sum_k e^{-\tau_k} \widehat{\boldsymbol{S}}_k$$

and hence

$$\frac{\partial}{\partial \tau(k)} \left(1/2 \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w} \right) = -\frac{1}{2} e^{-\tau_k} \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}_k \boldsymbol{w}.$$

Using the general identity $\boldsymbol{x}^\dagger \widehat{\boldsymbol{A}} \boldsymbol{x} = \mathrm{tr} \left(\boldsymbol{x}\boldsymbol{x}^\dagger \widehat{\boldsymbol{A}} \right)$

$$\frac{\partial}{\partial \tau(k)} \left(1/2 \boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}^{-1} \boldsymbol{w} \right) = -\frac{1}{2} e^{-\tau_k} \mathrm{tr} \left[\boldsymbol{w}\boldsymbol{w}^\dagger \widehat{\boldsymbol{S}}_k \right] \tag{B.8}$$

we get the last missing term.

Since all other terms of Eq. 5.8 do not depend on $\boldsymbol{\tau}$ adding up Eqs. B.4, B.5, B.6, B.7 and B.8 gives the full derivative of Eq. 5.12.

Bibliography

[Beh+11] S. Behnel et al. "Cython: The Best of Both Worlds". In: *Computing in Science Engineering* 13.2 (2011), pp. 31–39. ISSN: 1521-9615. DOI: 10.1109/MCSE.2010.118.

[Car+87] E Caroli et al. "Coded aperture imaging in X- and gamma-ray astronomy". In: *Space Science Reviews* 45.3 (1987), pp. 349–403. ISSN: 1572-9672. DOI: 10.1007/BF00171998. URL: http://dx.doi.org/10.1007/BF00171998.

[Die01] Roland Diehl. "Gamma-ray production and absorption processes". In: *The Universe in Gamma Rays*. Springer, 2001, pp. 9–25.

[EFK09] Torsten A Enßlin, Mona Frommert, and Francisco S Kitaura. "Information field theory for cosmological perturbation reconstruction and nonlinear signal analysis". In: *Physical Review D* 80.10 (2009), p. 105005.

[Enß] Torsten Enßlin. *Why IFT is a field theory*. URL: http://wwwmpa.mpa-garching.mpg.de/ift/Why_IFT_is_a_field_theory.html.

[Gha17] Mahsa Ghaempanah. "Information Field Theory with INTEGRAL/SPI data". PhD thesis. MPA Garching, 2017.

[KBS06] Steven M. Kahn, Peter Ballmoos, and Rashid A. Sunyaev. "High-Energy Spectroscopic Astrophysics: 30 (Saas-Fee Advanced Course)". In: (Mar. 2006).

[Mac03] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[Mac67] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297. URL: http://projecteuclid.org/euclid.bsmsp/1200512992.

[Opp+13] Niels Oppermann et al. "Reconstruction of Gaussian and log-normal fields with spectral smoothness". In: *Physical Review E* 87.3 (2013), p. 032136.

[Pan+13] Daniele Panozzo et al. "Weighted averages on surfaces". In: *ACM Transactions on Graphics (TOG)* 32.4 (2013), p. 60.

[SE15] Marco Selig and Torsten A Enßlin. "Denoising, deconvolving, and decomposing photon observations-Derivation of the D3PO algorithm". In: *Astronomy & Astrophysics* 574 (2015), A74.

[Sel] Marco Selig. *D3PO-Demonstration*. URL: https://web.archive.org/web/20170803094944/http://wwwmpa.mpa-garching.mpg.de/ift/d3po/demo.html (visited on 08/03/2017).

[Sie+16] Thomas Siegert et al. "Gamma-ray spectroscopy of positron annihilation in the Milky Way". In: *Astronomy & Astrophysics* 586 (2016), A84.

[Sie13] Thomas Siegert. “High-precision cosmic gamma-ray line spectroscopy - Spectral response and background modeling”. MA thesis. MPE Garching, 2013.

[Sie17] Thomas Siegert. “Positron-Annihilation Spectroscopy throughout the Milky Way”. PhD thesis. MPE Garching, 2017.

[SOE12] Marco Selig, Niels Oppermann, and Torsten A Enßlin. “Improving stochastic estimates with inference methods: Calculating matrix diagonals”. In: *Physical Review E* 85.2 (2012), p. 021134.

[Stu+03] SJ Sturner et al. “Monte Carlo simulations and generation of the SPI response”. In: *Astronomy & Astrophysics* 411.1 (2003), pp. L81–L84.

[Ved+03] Vedrenne, G. et al. “SPI: The spectrometer aboard INTEGRAL”. In: *A&A* 411.1 (2003), pp. L63–L70. DOI: 10.1051/0004-6361:20031482. URL: https://doi.org/10.1051/0004-6361:20031482.

[Yli] Joel Yliluoma. *Shortcomings: OpenMP and fork()*. URL: https://web.archive.org/web/20170327041455/http://bisqwit.iki.fi/story/howto/openmp/#OpenmpAndFork (visited on 03/27/2017).